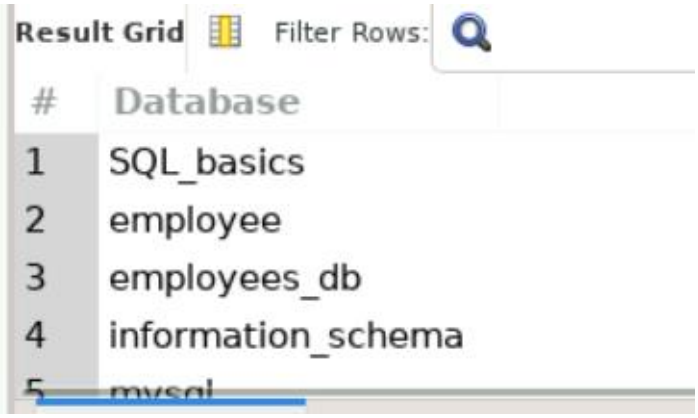
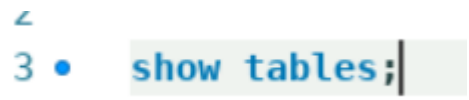
SQL Assessment Project -1 submitted by Sujit Sonar:

ScienceQtech Employee Performance Mapping.

The task to be performed:

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.



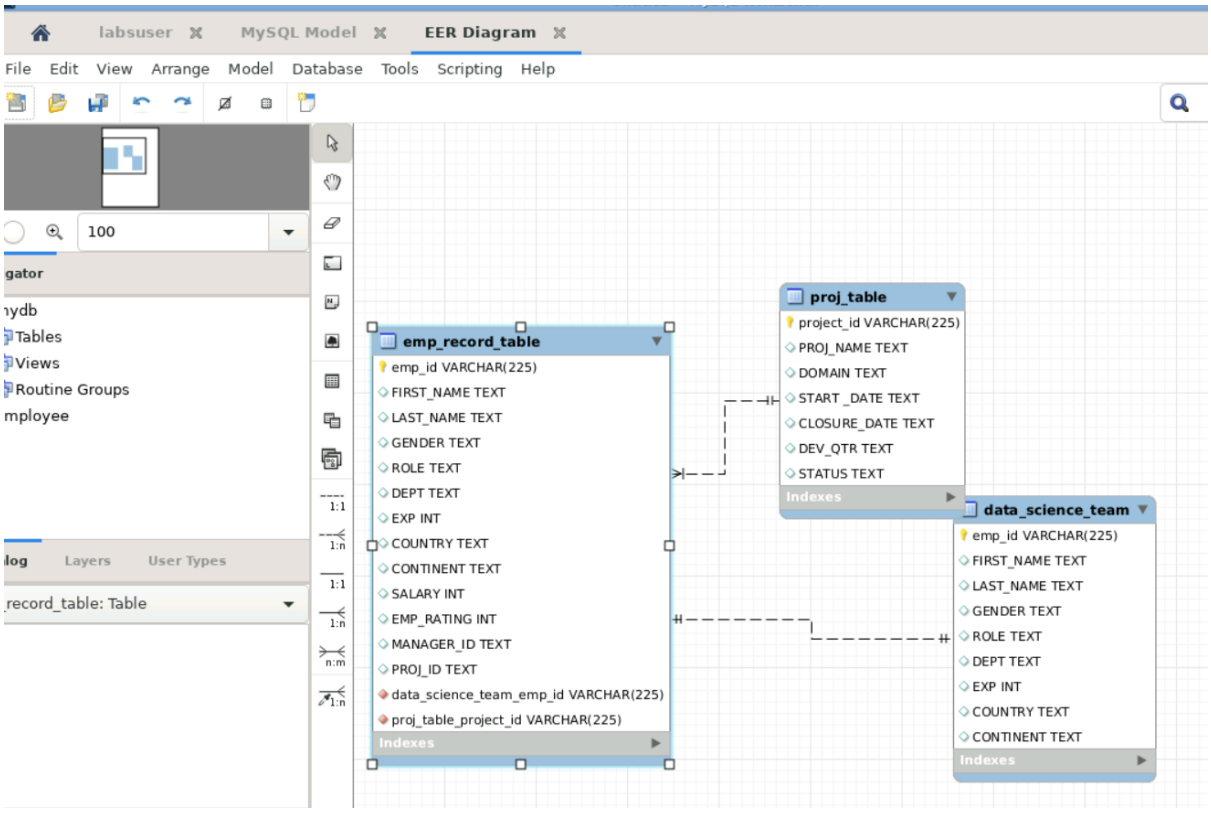


2. Create an ER diagram for the given **employee** database.

First altering the table to define Primary key:

```
1  alter table data_science_team
2  modify emp_id varchar(225) not null;
3
4  alter table data_science_team
5  add primary key (emp_id);
6
7
8  alter table emp_record_table
9  modify emp_id varchar(225) not null;
10
11 alter table emp_record_table
12 add primary key (emp_id);
13
14 alter table proj_table
15 modify project_id varchar(225) not null;
16
```

Then using the Mysql work bench , created the below ER diagram



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.

```
20 •  select emp_id, first_name, last_name, gender, dept
21    from emp_record_table
22    order by dept;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

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

emp_record_table 6

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

```
24 •  select emp_id, first_name, last_name, gender, dept, emp_rating
25    from emp_record_table
26    where emp_rating < 2 ;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

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
NULL	NULL	NULL	NULL	NULL	NULL

- greater than four

```
24 •  select emp_id, first_name, last_name, gender, dept, emp_rating
25    from emp_record_table
26    where emp_rating >4 ;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

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

- between two and four

```
24 •  select emp_id, first_name, last_name, gender, dept, emp_rating
25    from emp_record_table
26    where emp_rating between 2 and 4 ;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

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

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.

```
29 •  select first_name, last_name, concat(first_name," ", last_name) as NAME
30    from emp_record_table;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	first_name	last_name	NAME
1	Arthur	Black	Arthur Black
2	Eric	Hoffman	Eric Hoffman
3	William	Butler	William Butler
4	Dianna	Wilson	Dianna Wilson
5	Dorothy	Wilson	Dorothy Wilson

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

```
32 •  select a.emp_id, a.full_name, a.role, b.emp_reporting from
33    (select emp_id, concat(first_name," ",last_name)as full_name, role
34    from emp_record_table
35    where emp_id in (select manager_id from emp_record_table))a
36    left join
37    (select manager_id, count(emp_id) as emp_reporting from emp_record_table
38    group by manager_id) b
39    on a.emp_id = b.manager_id
40    order by b.emp_reporting desc;
```

Result Grid | Filter Rows: | Export:

#	emp_id	full_name	role	emp_reportin
1	E001	Arthur Black	PRESIDENT	5
2	E428	Pete Allen	MANAGER	3
3	E083	Patrick Voltz	MANAGER	3
4	E583	Janet Hale	MANAGER	3
5	E103	Emily Grove	MANAGER	2
6	E612	Tracy Norris	MANAGER	2

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

```
45 •  select * from emp_record_table
46    where dept = 'Finance'
47    union select * from emp_record_table
48    where dept='Healthcare';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	emp_id	FIRST_NAM	LAST_NAM	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP
	E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3
	E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	10500	4
	E403	Steve	Hoffman	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA	5000	3
	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
	E505	Chad	Wilson	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	CANADA	NORTH AMERICA	5000	2

6. 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.

```
50 •  select emp_id, first_name,last_name, role, dept,emp_rating, max(emp_rating) over (partition by dept) as
51    max_emp_rating from emp_record_table;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	emp_id	first_name	last_name	role	dept	emp_rating	max_emp_rating
1	E001	Arthur	Black	PRESIDENT	ALL	5	5
2	E010	William	Butler	LEAD DATA SCIENTIST	AUTOMOTIVE	2	5
3	E204	Karene	Nowak	SENIOR DATA SCIENTIST	AUTOMOTIVE	5	5
4	E428	Pete	Allen	MANAGER	AUTOMOTIVE	4	5
5	E532	Claire	Brennan	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	1	5
6	E005	Eric	Hoffman	LEAD DATA SCIENTIST	FINANCE	3	4
7	E103	Emily	Grove	MANAGER	FINANCE	4	4
8	E403	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	FINANCE	3	4
9	E052	Dianna	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	5	5
10	E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	1	5
11	E083	Patrick	Voltz	MANAGER	HEALTHCARE	5	5
12	E505	Chad	Wilson	ASSOCIATE DATA SCIENTIST	HEALTHCARE	2	5
13	E640	Jenifer	Jhones	JUNIOR DATA SCIENTIST	RETAIL	4	4

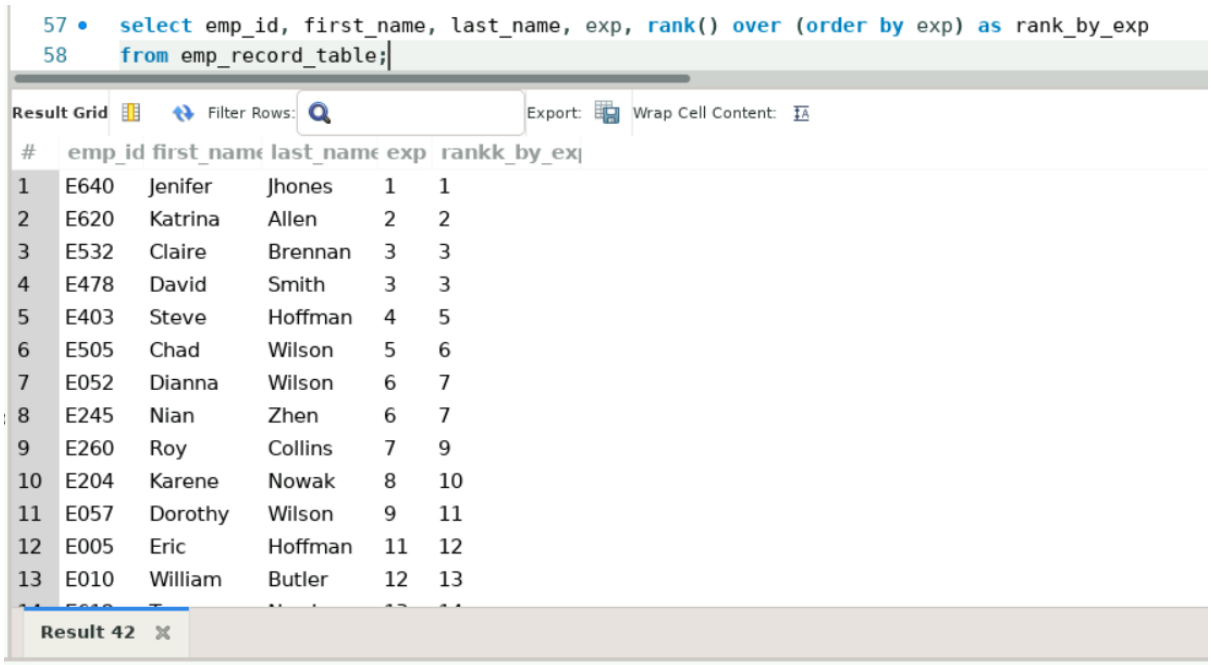
7. Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.

```
53 •  select role, salary, min(salary) over (partition by role) as min_salary,
54    max(salary) over (partition by role) as max_salary
55    from emp_record_table;
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

#	role	salary	min_salary	max_salary
1	ASSOCIATE DATA SCIENTIST	4300	4000	5000
2	ASSOCIATE DATA SCIENTIST	5000	4000	5000
3	ASSOCIATE DATA SCIENTIST	4000	4000	5000
4	ASSOCIATE DATA SCIENTIST	5000	4000	5000
5	JUNIOR DATA SCIENTIST	2800	2800	3000
6	JUNIOR DATA SCIENTIST	3000	2800	3000
7	LEAD DATA SCIENTIST	8500	8500	9000
8	LEAD DATA SCIENTIST	9000	8500	9000
9	MANAGER	10500	8500	11000
10	MANAGER	11000	8500	11000
11	MANAGER	9500	8500	11000
12	MANAGER	10000	8500	11000
13	MANAGER	8500	8500	11000

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



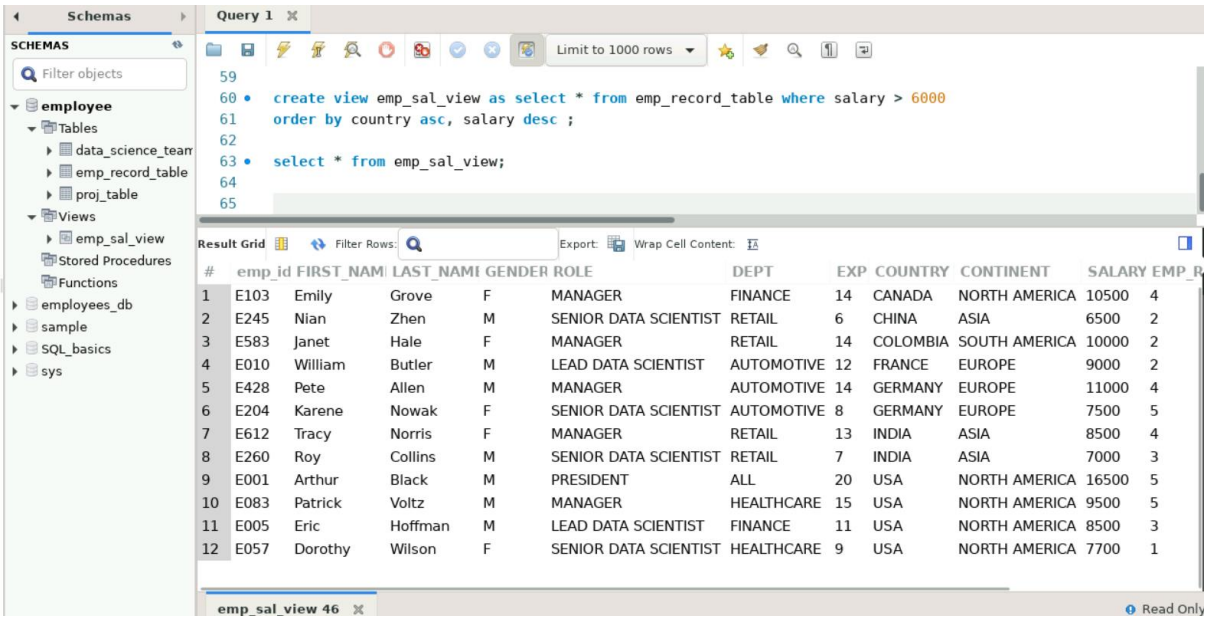
```
select emp_id, first_name, last_name, exp, rank() over (order by exp) as rank_by_exp
from emp_record_table;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	emp_id	first_name	last_name	exp	rankk_by_exp
1	E640	Jenifer	Jhones	1	1
2	E620	Katrina	Allen	2	2
3	E532	Claire	Brennan	3	3
4	E478	David	Smith	3	3
5	E403	Steve	Hoffman	4	5
6	E505	Chad	Wilson	5	6
7	E052	Dianna	Wilson	6	7
8	E245	Nian	Zhen	6	7
9	E260	Roy	Collins	7	9
10	E204	Karene	Nowak	8	10
11	E057	Dorothy	Wilson	9	11
12	E005	Eric	Hoffman	11	12
13	E010	William	Butler	12	13

Result 42

9. 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 emp_sal_view as select * from emp_record_table where salary > 6000
order by country asc, salary desc ;

select * from emp_sal_view;
```

Schemas: employee (Tables: data_science_team, emp_record_table, proj_table; Views: emp_sal_view; Stored Procedures; Functions), employees_db, sample, SQL_basics, sys

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	emp_id	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_R
1	E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	10500	4
2	E245	Nian	Zhen	M	SENIOR DATA SCIENTIST	RETAIL	6	CHINA	ASIA	6500	2
3	E583	Janet	Hale	F	MANAGER	RETAIL	14	COLOMBIA	SOUTH AMERICA	10000	2
4	E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9000	2
5	E428	Pete	Allen	M	MANAGER	AUTOMOTIVE	14	GERMANY	EUROPE	11000	4
6	E204	Karene	Nowak	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE	7500	5
7	E612	Tracy	Norris	F	MANAGER	RETAIL	13	INDIA	ASIA	8500	4
8	E260	Roy	Collins	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA	7000	3
9	E001	Arthur	Black	M	PRESIDENT	ALL	20	USA	NORTH AMERICA	16500	5
10	E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500	5
11	E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3
12	E057	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA	7700	1

emp_sal_view 46 | Read Only

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

```
65 •  select t.emp_id, t.first_name, t.last_name, t.exp from
66    (select * from emp_record_table
67    where exp > 10)t;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	emp_id	FIRST_NAM	LAST_NAM	EXP
1	E001	Arthur	Black	20
2	E005	Eric	Hoffman	11
3	E010	William	Butler	12
4	E083	Patrick	Voltz	15
5	E103	Emily	Grove	14
6	E428	Pete	Allen	14
7	E583	Janet	Hale	14
8	E612	Tracy	Norris	13

11. 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.

```
70    delimiter &&
71 •  create procedure get_emp_exp()
72    begin
73    select * from emp_record_table where exp > 3 order by exp;
74    end &&
75
76 •  call get_emp_exp;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	emp_id	FIRST_NAM	LAST_NAM	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EM
1	E403	Steve	Hoffman	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA	5000	3
2	E505	Chad	Wilson	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	CANADA	NORTH AMERICA	5000	2
3	E052	Dianna	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	5500	5
4	E245	Nian	Zhen	M	SENIOR DATA SCIENTIST	RETAIL	6	CHINA	ASIA	6500	2
5	E260	Roy	Collins	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA	7000	3
6	E204	Karene	Nowak	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE	7500	5
7	E057	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA	7700	1
8	E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3
9	E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9000	2
10	E612	Tracy	Norris	F	MANAGER	RETAIL	13	INDIA	ASIA	8500	4
11	E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	10500	4
12	E428	Pete	Allen	M	MANAGER	AUTOMOTIVE	14	GERMANY	EUROPE	11000	4
13	E583	Janet	Hale	F	MANAGER	RETAIL	14	COLOM...	SOUTH AMERICA	10000	2

Result 50

12. 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'.

```
use employee;
drop procedure if exists get_role;
delimiter &&
create procedure get_role(
in expx int,
out role_check varchar(225),
out experiance int)
begin
declare emp_exp int;
declare emp_role varchar(225);
select role, exp into emp_role, emp_exp
from data_science_team where exp = expx;
set experiance = emp_exp;
case
when emp_exp <=2 then set role_check = 'JUNIOR DATA SCIENTIST';
when emp_exp = 3 then set role_check = 'ASSOCIATE DATA SCIENTIST';
when emp_exp between 2 and 5 then set role_check = 'ASSOCIATE DATA SCIENTIST';
when emp_exp between 5 and 10 then set role_check = 'SENIOR DATA SCIENTIST';
when emp_exp = 6 then set role_check = 'SENIOR DATA SCIENTIST';
when emp_exp between 10 and 12 then set role_check = 'LEAD DATA SCIENTIST';
when emp_exp between 12 and 16 then set role_check = 'MANAGER';
else
begin
set role_check = 'no employe with given experiance';
end;
end case;
```

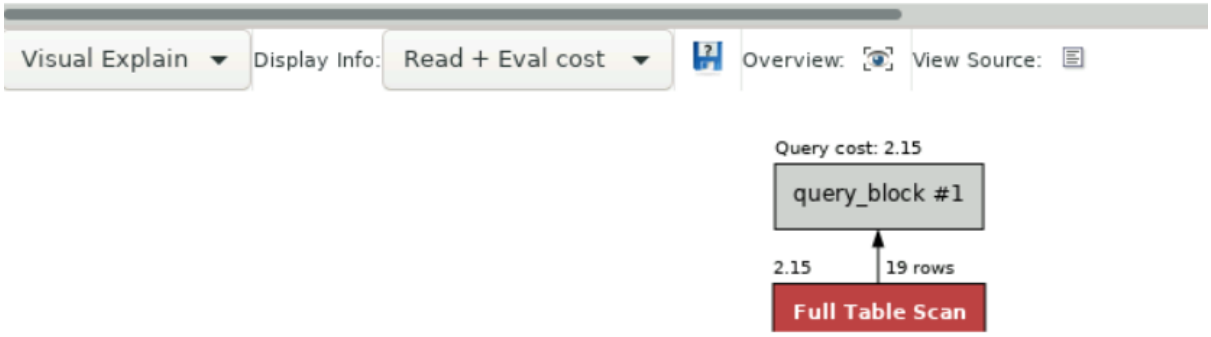
```
end &&
delimiter ;

call get_role(3, @role_check, @experiance);
select @role_check,@experiance;
```

Note: The stored procedure works for all values of experience, except for 3 and 6, the query is getting interrupted, and I am not able to find out why. Please review and provide feedback on this one what am I doing wrong.

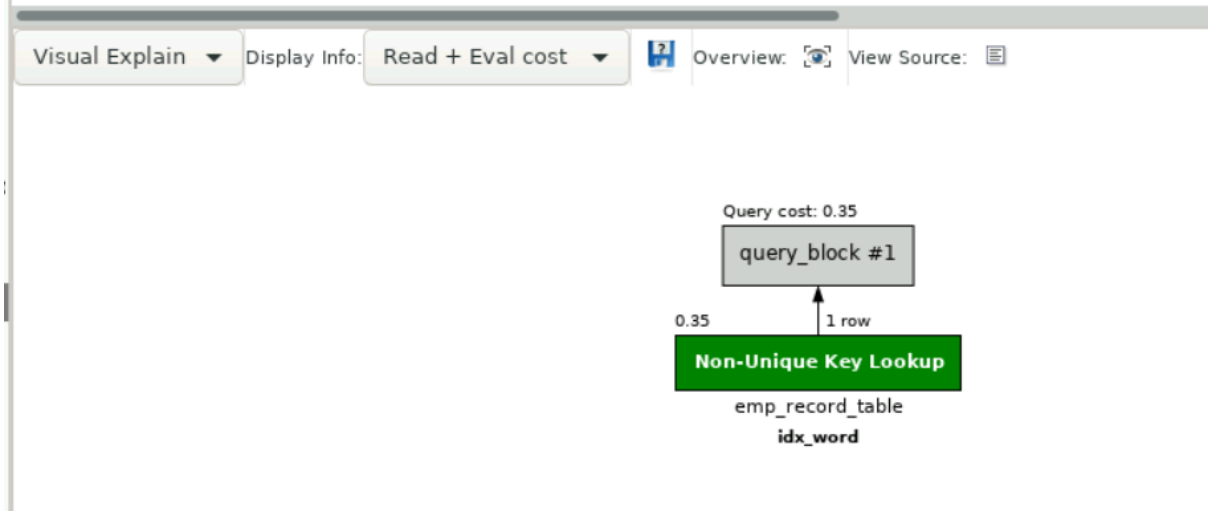
13. Create an index to improve the cost and performance of the query to find the employee whose FIRST_NAME is ‘Eric’ in the employee table after checking the execution plan.

```
12
13 •  select * from employee.emp_record_table
14    where first_name = 'Eric';
```



After creating index on text column

```
18 •  create index idx_word on emp_record_table(FIRST_NAME(50));
19 •  select * from employee.emp_record_table
20    where first_name = 'Eric';
```



14. 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).

```
24 •  select emp_id, salary, emp_rating, round((5*salary)/100 * emp_rating,2) as Bonus
25    from emp_record_table;
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

#	emp_id	salary	emp_ratin	Bonus
1	E001	16500	5	4125.00
2	E005	8500	3	1275.00
3	E010	9000	2	900.00
4	E052	5500	5	1375.00
5	E057	7700	1	385.00
6	E083	9500	5	2375.00
7	E103	10500	4	2100.00
8	E204	7500	5	1875.00
9	E245	6500	2	650.00
10	E260	7000	3	1050.00
11	E403	5000	3	750.00
12	E428	11000	4	2200.00

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

```
27 •  select continent, country, avg(salary) avg_sal_distribution
28    from emp_record_table
29    group by continent,country
30    order by continent, country;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	continent	country	avg_sal_distributic
1	ASIA	CHINA	6500.0000
2	ASIA	INDIA	6166.6667
3	EUROPE	FRANCE	9000.0000
4	EUROPE	GERMANY	7600.0000
5	NORTH AMERICA	CANADA	7000.0000
6	NORTH AMERICA	USA	9440.0000
7	SOUTH AMERICA	COLOMBIA	5600.0000

########## END##########