

# ScienceQtech Employee Performance Mapping.

## DESCRIPTION

ScienceQtech is a startup that works in the Data Science field. ScienceQtech has worked on fraud detection, market basket, self-driving cars, supply chain, algorithmic early detection of lung cancer, customer sentiment, and the drug discovery field. With the annual appraisal cycle around the corner, the HR department has asked you (Junior Database Administrator) to generate reports on employee details, their performance, and on the project that the employees have undertaken, to analyze the employee database and extract specific data based on different requirements.

## Objective:

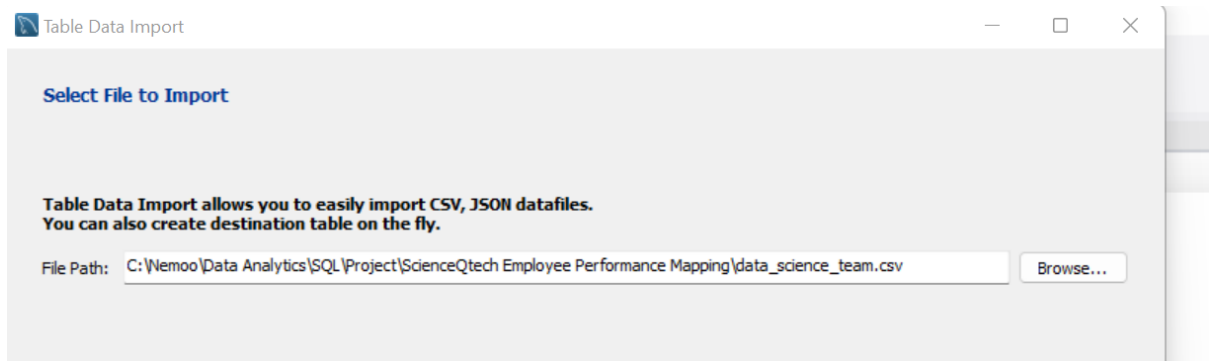
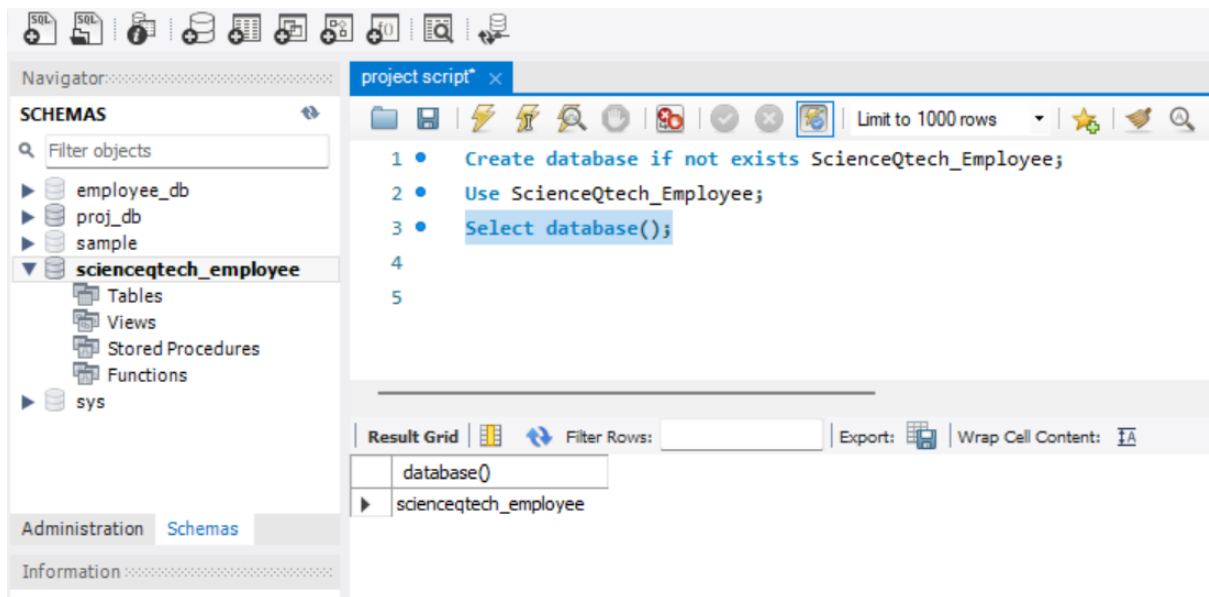
To facilitate a better understanding, managers have provided ratings for each employee which will help the HR department to finalize the employee performance mapping. As a DBA, you should find the maximum salary of the employees and ensure that all jobs are meeting the organization's profile standard. You 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.

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 if not exists ScienceQtech\_Employee;

Use ScienceQtech\_Employee;

Select database();



## Select Destination

### Select destination table and additional options.

- ☐ Use existing table:
- ☒ Create new table:
- ☐ Drop table if exists

## Configure Import Settings

Detected file format: csv 

Encoding:

### Columns:

<input checked="" type="checkbox"/> Source Column	Field type
<input checked="" type="checkbox"/> EMP_ID	<input type="text" value="text"/>
<input checked="" type="checkbox"/> FIRST_NAME	<input type="text" value="text"/>
<input checked="" type="checkbox"/> LAST_NAME	<input type="text" value="text"/>
<input checked="" type="checkbox"/> GENDER	<input type="text" value="text"/>
<input checked="" type="checkbox"/> ROLE	<input type="text" value="text"/>
<input checked="" type="checkbox"/> DEPT	<input type="text" value="text"/>

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT
E005	Eric	Hoffman	M	LEAD DATA...	FINANCE	11	USA	NORTH AM...
E010	William	Butler	M	LEAD DATA...	AUTOMOTI...	12	FRANCE	EUROPE
E052	Dianna	Wilson	F	SENIOR DA...	HEALTHCARE	6	CANADA	NORTH AM...
E057	Dorothy	Wilson	F	SENIOR DA...	HEALTHCARE	9	USA	NORTH AM...
E204	Karene	Nowak	F	SENIOR DA...	AUTOMOTI...	8	GERMANY	EUROPE

< Back

Next >

Cancel

## Import Data

The following tasks will now be performed. Please monitor the execution.

- ☒ Prepare Import
- ☒ Import data file

Finished performing tasks. Click [Next >] to continue.

### Message Log

- Data import  
- Data import  
- Data import  
- Data import  
- Data import  
- Data import  
- Data import  
- Data import  
- Data import  
- Data import  
- Data import  
- Data import  
- Data import  
- Import finished  
Import data file finished  
Finished performing tasks.

Hide Logs

< Back

Next >

Cancel

## Table Data Import

### Import Results

File C:\Nemoo\Data Analytics\SQL\Project\ScienceQtech Employee Performance Mapping\data\_science\_team.csv was imported in 0.198 s

Table scienceqtech\_employee.data\_science\_team was created

13 records imported

### SCHEMAS

Filter objects

- employee\_db
- proj\_db
- sample
- scienceqtech\_employee
  - Tables
    - data\_science\_team
  - Views
  - Stored Procedures
  - Functions

1 • Create database if not exists ScienceQtech\_Employee;  
2 • Use ScienceQtech\_Employee;  
3 • Select database();  
4  
5

Applying the same procedure as above we will import the **proj\_table.csv** and **emp\_record\_table.csv** to **proj\_table** and **emp\_record\_table**.

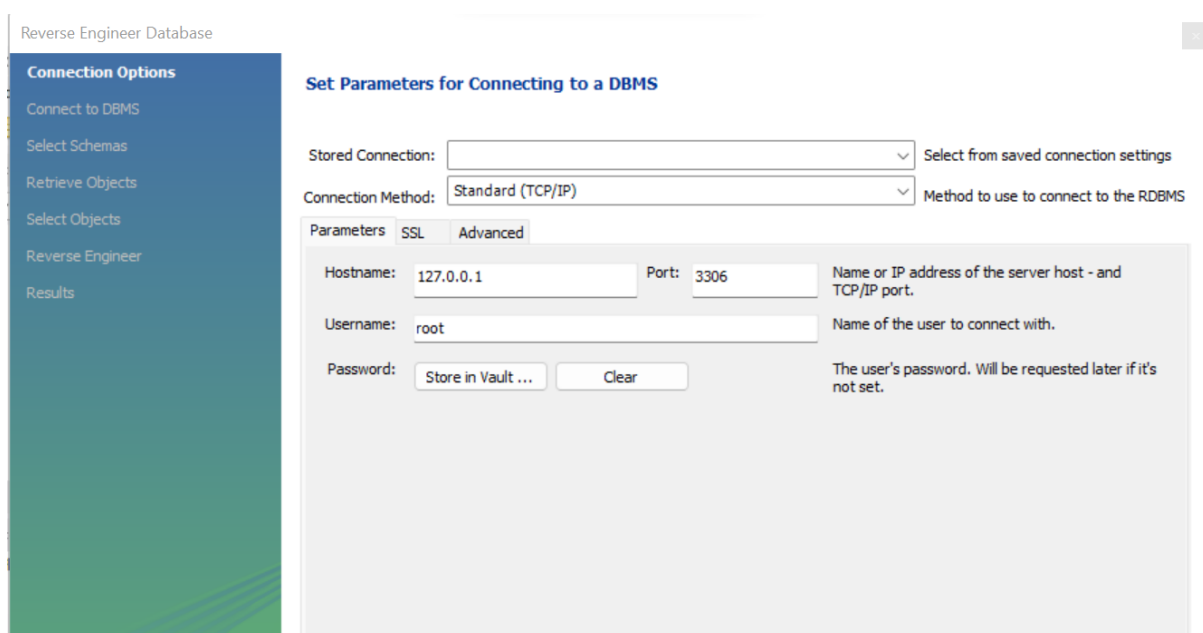


Now all three tables are imported to the MYSQL.

### Create an ER diagram for the given employee database.

Entity-Relationship (ER) model is a visual representation of the table's structure and the relationships between logically related tables.

Click on **Database -> Reverse Engineer**.



Reverse Engineer Database

**Connection Options**

- Connect to DBMS
- Select Schemas
- Retrieve Objects
- Select Objects
- Reverse Engineer
- Results

**Set Parameters for Connecting to a DBMS**

Stored Connection:  Select from saved connection settings

Connection Method:  Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname:  Port:  Name or IP address of the server host - and TCP/IP port.

Username:  Name of the user to connect with.

Password:  Store in Vault ... Clear The user's password. Will be requested later if it's not set.

## Connection Options

Connect to DBMS

Select Schemas

Retrieve Objects

Select Objects

Reverse Engineer

Results

## Set Parameters for Connecting to a DBMS

Stored Connection: **Local instance MySQL80** Select from saved connection settings

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: localhost Port: 3306 Name or IP address of the server host - and TCP/IP port.

Username: root Name of the user to connect with.

Password:   The user's password. Will be requested later if it's not set.

## Connection Options

## Connect to DBMS

Select Schemas

Retrieve Objects

Select Objects

Reverse Engineer

Results

## Connect to DBMS and Fetch Information

The following tasks will now be executed. Please monitor the execution.  
Press Show Logs to see the execution logs.

- ☒ Connect to DBMS
- ☒ Retrieve Schema List from Database
- ☒ Check Common Server Configuration Issues

Execution Completed Successfully

Fetch finished.

Select the scienceqtech\_employee schema as we need to create the ER diagram for this schema's tables.

## Connection Options

Connect to DBMS

## Select Schemas

Retrieve Objects

Select Objects

Reverse Engineer

Results

## Select Schemas to Reverse Engineer

**Select the schemas you want to include:**

- ☐ employee\_db
- ☐ proj\_db
- ☐ sample
- ☒ **scienceqtech\_employee**

Connection Options

Connect to DBMS

Select Schemas

**Retrieve Objects**

Select Objects

Reverse Engineer

Results

## Retrieve and Reverse Engineer Schema Objects

The following tasks will now be executed. Please monitor the execution. Press Show Logs to see the execution logs.

- ☒ Retrieve Objects from Selected Schemas
- ☒ Check Results

Retrieval Completed Successfully  
Finished.

Connection Options

Connect to DBMS

Select Schemas

Retrieve Objects

**Select Objects**

Reverse Engineer

Results

## Select Objects to Reverse Engineer

☒ Import MySQL Table Objects

3 Total Objects, 3 Selected

Show Filter

Modeling /



MWB

MWB

MWB

Connection Options

Connect to DBMS

Select Schemas

Retrieve Objects

Select Objects

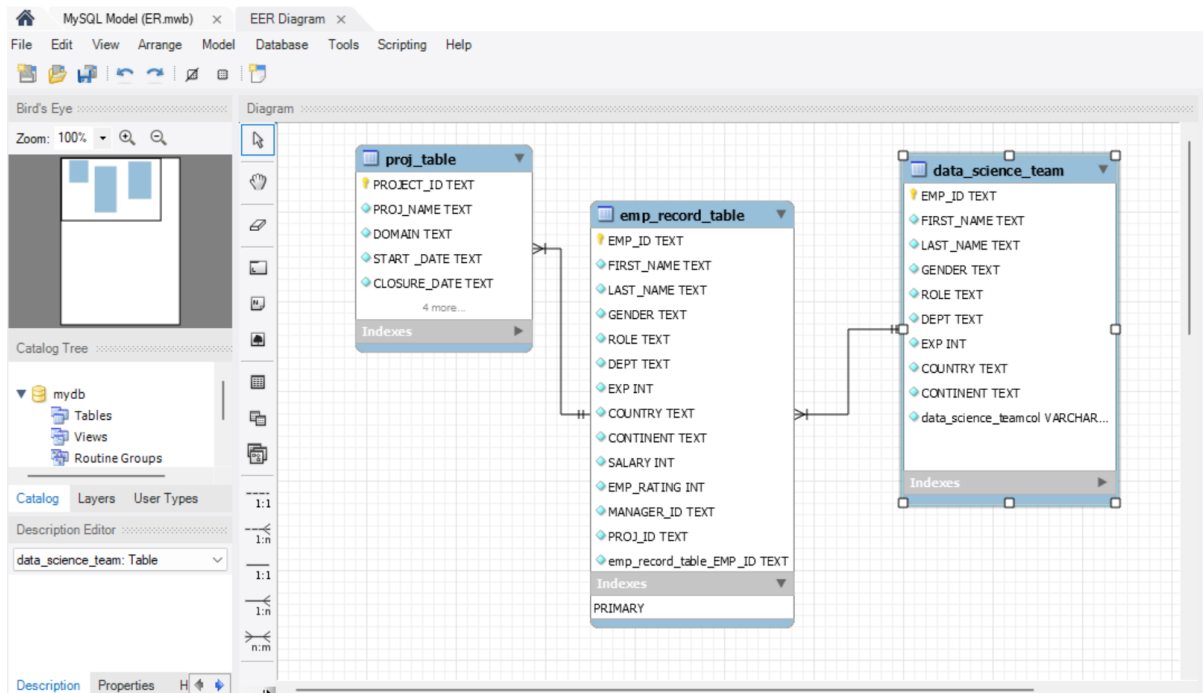
Reverse Engineer

**Results**

## Reverse Engineering Results

Summary of Reverse Engineered Objects:

- 3 tables from schema 'scienceqtech\_employee'



**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.**

Create View emp\_record\_table\_view1 as

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

from emp\_record\_table;

select \* from emp\_record\_table\_view1; → to view the emp\_record\_table\_view1

```

6 • Create View emp_record_table_view1 as
7   select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
8   from emp_record_table;
9
10 • select * from emp_record_table_view1;
11

```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT
E001	Arthur	Black	M	ALL
E005	Eric	Hoffman	M	FINANCE
E010	William	Butler	M	AUTOMOTIVE
E052	Dianna	Wilson	F	HEALTHCARE
E057	Dorothy	Wilson	F	HEALTHCARE
E083	Patrick	Voltz	M	HEALTHCARE

emp\_record\_table\_view1 2 x



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

Create View emp\_record\_table\_view2 as





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

from emp\_record\_table


where EMP\_RATING <2;

select \* from emp\_record\_table\_view2;

```
14 • Create View emp_record_table_view2 as
15   select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
16   from emp_record_table
17   where EMP_RATING <2;
18
19 • select * from emp_record_table_view2;
20
```

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

	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

emp\_record\_table\_view2 4 x 

- **greater than four**

Create View emp\_record\_table\_view3 as

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

from emp\_record\_table

where EMP\_RATING > 4;

select \* from emp\_record\_table\_view3;

```

22 • Create View emp_record_table_view3 as
23     select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
24     from emp_record_table
25     where EMP_RATING > 4;
26
27 • select * from emp_record_table_view3;
28

```

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**

Create View emp\_record\_table\_view4 as

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

from emp\_record\_table

where EMP\_RATING

between 2

and 4;

select \* from emp\_record\_table\_view4;

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
E403	Steve	Hoffman	M	FINANCE	3

**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';
```

The screenshot shows a SQL IDE window titled 'project script x emp\_record\_table'. The query editor contains the following SQL code:

```
37 • select * from emp_record_table_view4;
38
39 ##concatenate FIRST_NAME and the LAST_NAME of employees in the Finance department
40 • select concat(FIRST_NAME," ",LAST_NAME) as Name
41 from emp_record_table
42 Where DEPT = 'FINANCE';
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query. The results are as follows:

Name
Eric Hoffman
Emily Grove
Steve Hoffman

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

```
SELECT MANAGER_ID, COUNT(EMP_ID) as EMP_COUNT
```

```
FROM emp_record_table
```

```
GROUP BY MANAGER_ID
```

```
ORDER BY EMP_COUNT DESC;
```

```
45 • SELECT MANAGER_ID, COUNT(EMP_ID) as EMP_COUNT
46 FROM emp_record_table
47 GROUP BY MANAGER_ID
48 ORDER BY EMP_COUNT DESC;
```

The screenshot shows a SQL IDE window displaying the results of a query. The 'Result Grid' tab is active, showing the following data:

MANAGER_ID	EMP_COUNT
E001	5
E428	3
E083	3
E583	3
E103	2
E612	2

At the bottom of the window, it says 'Result 4 x'.

**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 e.EMP_ID,  
  
concat(e.FIRST_NAME," ",e.LAST_NAME) As Full_Name,  
  
e.DEPT  
  
from emp_record_table e  
  
Where (e.DEPT = 'HEALTHCARE' or e.DEPT = 'FINANCE')  
  
UNION  
  
select p.EMP_ID,  
  
concat(p.FIRST_NAME," ",p.LAST_NAME) As Full_Name,  
  
p.DEPT  
  
From data_science_team p;
```

The screenshot shows a SQL IDE interface with a query editor and a results grid. The query editor contains the following SQL code:

```
54 from emp_record_table e  
55 Where (e.DEPT = 'HEALTHCARE' or e.DEPT = 'FINANCE')  
56 UNION  
57 select p.EMP_ID,  
58 concat(p.FIRST_NAME," ",p.LAST_NAME) As Full_Name,  
59 p.DEPT  
60 From data_science_team p;
```

The results grid displays the following data:

EMP_ID	Full_Name	DEPT
E005	Eric Hoffman	FINANCE
E052	Dianna Wilson	HEALTHCARE
E057	Dorothy Wilson	HEALTHCARE
E083	Patrick Voltz	HEALTHCARE
E103	Emily Grove	FINANCE
E403	Steve Hoffman	FINANCE

The results grid also includes a "Filter Rows" input field, an "Export" button, and a "Wrap Cell Content" checkbox.

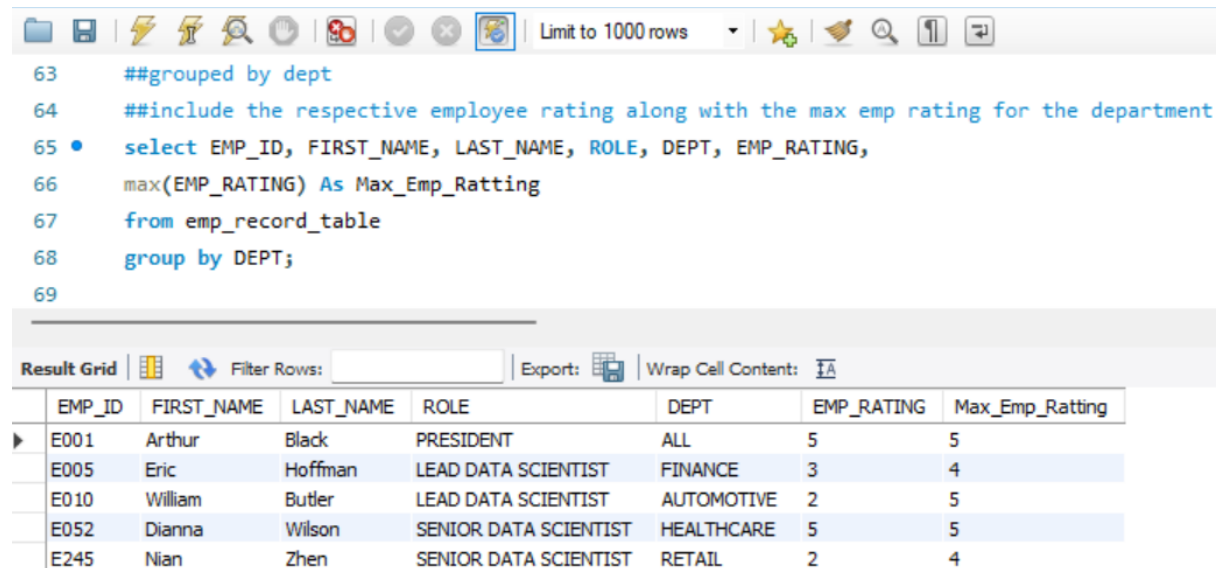
**Write a query to list down employee details such as EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, DEPARTMENT, and EMP\_RATING grouped by dept. Also include the respective employee rating along with the max emp rating for the department.**

```
select EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EMP_RATING,
```

```
max(EMP_RATING) As Max_Emp_Ratting
```

```
from emp_record_table
```

```
group by DEPT;
```



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```

63  ##grouped by dept
64  ##include the respective employee rating along with the max emp rating for the department
65  • select EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EMP_RATING,
66      max(EMP_RATING) As Max_Emp_Ratting
67  from emp_record_table
68  group by DEPT;
69

```

Below the query, the results are displayed in a table with the following columns: EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, DEPT, EMP\_RATING, and Max\_Emp\_Ratting.

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EMP_RATING	Max_Emp_Ratting
E001	Arthur	Black	PRESIDENT	ALL	5	5
E005	Eric	Hoffman	LEAD DATA SCIENTIST	FINANCE	3	4
E010	William	Butler	LEAD DATA SCIENTIST	AUTOMOTIVE	2	5
E052	Dianna	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	5	5
E245	Nian	Zhen	SENIOR DATA SCIENTIST	RETAIL	2	4

**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,SALARY, max(SALARY) As max_salary, min(SALARY) As min_salary
```

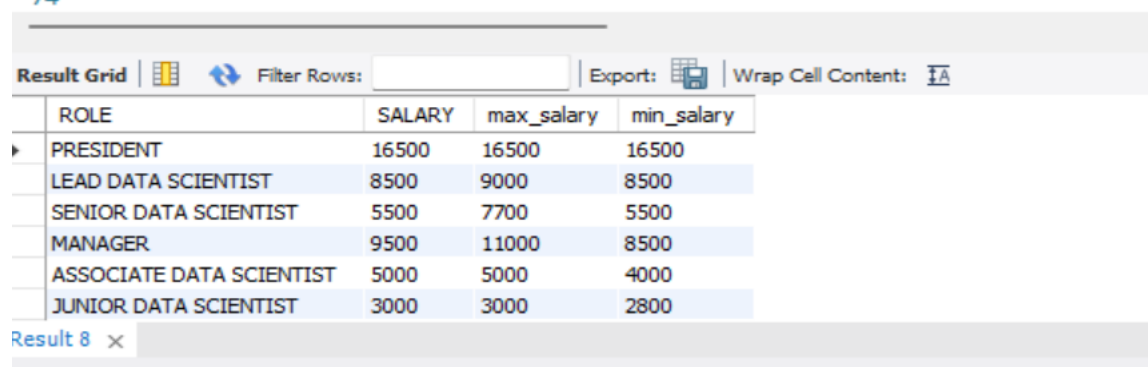
```
from emp_record_table
```

```
group by ROLE;
```

```

71  • select ROLE,SALARY, max(SALARY) As max_salary, min(SALARY) As min_salary
72  from emp_record_table
73  group by ROLE;
74

```



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```

71  • select ROLE,SALARY, max(SALARY) As max_salary, min(SALARY) As min_salary
72  from emp_record_table
73  group by ROLE;
74

```

Below the query, the results are displayed in a table with the following columns: ROLE, SALARY, max\_salary, and min\_salary.

ROLE	SALARY	max_salary	min_salary
PRESIDENT	16500	16500	16500
LEAD DATA SCIENTIST	8500	9000	8500
SENIOR DATA SCIENTIST	5500	7700	5500
MANAGER	9500	11000	8500
ASSOCIATE DATA SCIENTIST	5000	5000	4000
JUNIOR DATA SCIENTIST	3000	3000	2800

Result 8 x

**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 desc)
```

```
From emp_record_table;
```

```
76 • select EMP_ID, FIRST_NAME, LAST_NAME, EXP,  
77      rank() Over (order by EXP desc)  
78      From emp_record_table;  
79
```

Result Grid					
		Filter Rows:		Export:	Wrap Cell Content:
	EMP_ID	FIRST_NAME	LAST_NAME	EXP	rank() Over (order by EXP desc)
▶	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

Result 15 x

**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 Salary\_View1 as

```
select EMP_ID, FIRST_NAME, LAST_NAME, COUNTRY, SALARY
```

```
from emp_record_table
```

```
where SALARY >6000;
```

```
select * from Salary_View1;
```

```
81 • Create View Salary_View1 as  
82   select EMP_ID, FIRST_NAME, LAST_NAME, COUNTRY, SALARY  
83   from emp_record_table  
84   where SALARY >6000;  
85  
86 • select * from Salary_View1;
```

Result Grid					
		Filter Rows:		Export:	Wrap Cell Content:
	EMP_ID	FIRST_NAME	LAST_NAME	COUNTRY	SALARY
▶	E001	Arthur	Black	USA	16500
	E005	Eric	Hoffman	USA	8500
	E010	William	Butler	FRANCE	9000
	E057	Dorothy	Wilson	USA	7700
	E083	Patrick	Voltz	USA	9500
	E103	Emily	Grove	CANADA	10500

Salary\_View1 17 x

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




```
select e.FIRST_NAME, e.LAST_NAME, e.EXP,
```


```
(select count(distinct p.EMP_ID) from emp_record_table p ) as EXP1
```

```
From emp_record_table e
```

```
where e.EXP > 10;
```

```
88  ##Nested query to find employees with experience of more than ten years
89  •  select e.FIRST_NAME, e.LAST_NAME, e.EXP,
90      (select count(distinct p.EMP_ID) from emp_record_table p ) as EXP1
91      From emp_record_table e
92      where e.EXP > 10;
93
```

Result Grid    Filter Rows: <input type="text"/>   Export:    Wrap Cell Content: 				
	FIRST_NAME	LAST_NAME	EXP	EXP1
▶	Arthur	Black	20	19
	Eric	Hoffman	11	19
	William	Butler	12	19
	Patrick	Voltz	15	19
	Emily	Grove	14	19
	Pete	Allen	14	19

Result 2 x 

**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 Get_Exp()
```

```
begin
```

```
select * from emp_record_table where EXP > 3;
```

```
end &&
```

```
call Get_Exp;
```

```

95  ##more than three years
96  DELIMITER &&
97  • Create procedure Get_Exp()
98  begin
99      select * from emp_record_table where EXP > 3;
100 end &&
101 • call Get_Exp;

```

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

	EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY
▶	E001	Arthur	Black	M	PRESIDENT	ALL	20	USA	NORTH AMERICA	16500
	E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500
	E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9000
	E052	Dianna	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	5500
	E057	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA	7700

Result 3 x

**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',

DELIMITER &&

Create procedure Job\_Profile()

begin

select \* from data\_science\_team

Where EXP <= 2 and

ROLE = "JUNIOR DATA SCIENTIST"

end &&

call Job\_Profile

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

	EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT
▶	E620	Katrina	Allen	F	JUNIOR DATA SCIENTIST	RETAIL	2	INDIA	ASIA
	E640	Jenifer	Jhones	F	JUNIOR DATA SCIENTIST	RETAIL	1	COLOMBIA	SOUTH AMERICA



## experience of 2 to 5 years assign 'ASSOCIATE DATA SCIENTIST',

##experience of 5 to 10 years assign 'SENIOR DATA SCIENTIST',

##experience of 10 to 12 years assign 'LEAD DATA SCIENTIST',

##experience of 12 to 16 years assign 'MANAGER'.

DELIMITER &&

create procedure Job\_Profile1()

begin

select \* from data\_science\_team

Where EXP between 2 and 5

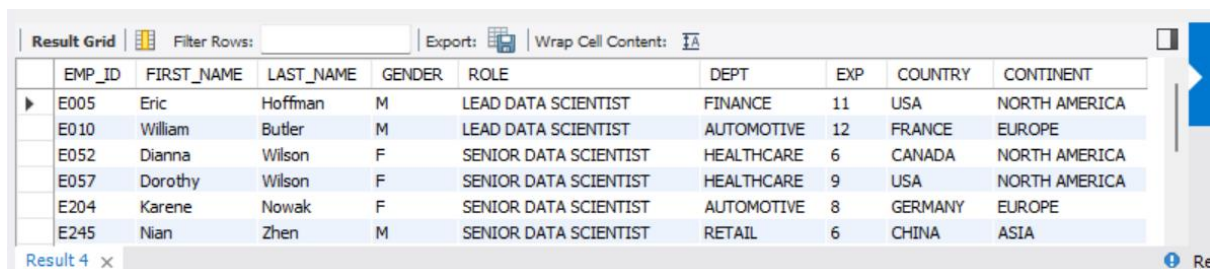
or EXP between 5 and 10

or EXP between 10 and 12

or EXP between 12 and 16;

end &&

call Job\_Profile1



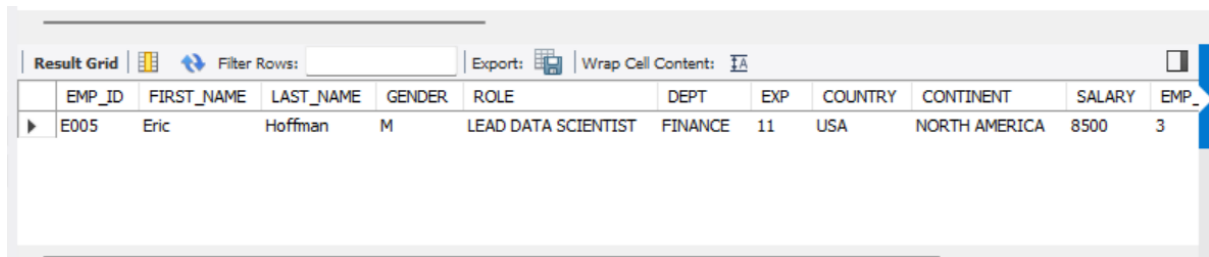
The screenshot shows a SQL query result grid with the following columns: EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, ROLE, DEPT, EXP, COUNTRY, and CONTINENT. The data is as follows:

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

**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**

```
create index idx_word on emp_record_table(FIRST_NAME);
```

```
select *from emp_record_table where FIRST_NAME ='Eric';
```



The screenshot shows a database query result grid. The grid has columns: EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY, and EMP\_. The first row contains the data for employee Eric Hoffman, who is a LEAD DATA SCIENTIST in the FINANCE department, with 11 years of experience, located in the USA, North America, with a salary of 8500 and an employee ID of E005.

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_
E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3

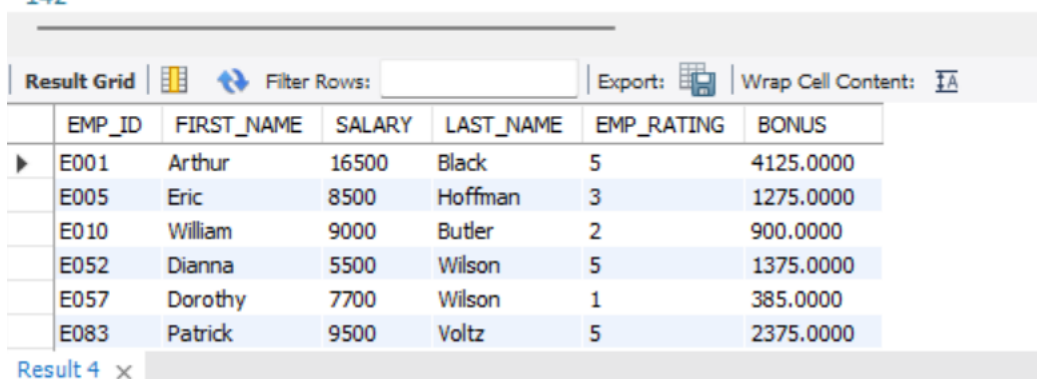
**Write a query to calculate the bonus for all the employees, based on their ratings and salaries (Use the formula: 5% of salary \* employee rating).**

```
select EMP_ID, FIRST_NAME, SALARY, LAST_NAME, EMP_RATING,
```

```
(SALARY * 5/100)*(EMP_RATING) as BONUS
```

```
from emp_record_table;
```

```
---
139     select EMP_ID, FIRST_NAME, SALARY, LAST_NAME, EMP_RATING,
140     (SALARY * 5/100)*(EMP_RATING) as BONUS
141     from emp_record_table;
142
```



The screenshot shows a database query result grid with columns: EMP\_ID, FIRST\_NAME, SALARY, LAST\_NAME, EMP\_RATING, and BONUS. The grid contains six rows of data for employees E001, E005, E010, E052, E057, and E083. The BONUS column shows the calculated bonus for each employee based on the formula: (SALARY \* 5/100) \* EMP\_RATING.

EMP_ID	FIRST_NAME	SALARY	LAST_NAME	EMP_RATING	BONUS
E001	Arthur	16500	Black	5	4125.0000
E005	Eric	8500	Hoffman	3	1275.0000
E010	William	9000	Butler	2	900.0000
E052	Dianna	5500	Wilson	5	1375.0000
E057	Dorothy	7700	Wilson	1	385.0000
E083	Patrick	9500	Voltz	5	2375.0000

Result 4 x

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





Select COUNTRY, CONTINENT,

avg(SALARY) as Avg\_salary

from emp\_record\_table

group by CONTINENT;

```
145  select COUNTRY, CONTINENT,  
146  avg(SALARY) as Avg_salary  
147  from emp_record_table  
148  group by CONTINENT;  
149
```

Result Grid   Filter Rows: <input type="text"/>				Export: 	Wrap Cell Content: 
	COUNTRY	CONTINENT	Avg_salary		
▶	USA	NORTH AMERICA	8525.0000		
	FRANCE	EUROPE	7950.0000		
	CHINA	ASIA	6250.0000		
	COLOMBIA	SOUTH AMERICA	5600.0000		

Result 7 ×