

# **Science Qtech Employee Performance Mapping**

## **Problem Statement:**

**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 the project that the employees have undertaken, to analyse the employee database and extract specific data based on different requirements.

## **Objectives:**

- To find the maximum salary of the employees and ensure that all jobs meet the organization's profile standard
- To calculate the bonuses to find the extra cost of the expenses

## **Prerequisites:**

- ER diagram
- Working of database
- Working of tables

- SQL views
- SQL functions
- SQL queries

## Industry Relevance

- **ER diagram:** It is used to visualize the structure of a table as well as the relationships between logically related tables.
- **Database:** It is a collection of tables that store a specific set of structured data.
- **Tables:** It is a database object that contains all the data within it.
- **SQL functions:** Several built-in functions are available in SQL to calculate data.
- **SQL queries:** A query is a request for data or information from a database table or combination of tables. As a result of the structured query language (SQL), this data may be displayed as pictorials, graphs, or complex results, such as trend analyses from data mining tools.

## Dataset Description

**Description** **EMP\_ID** - ID of the employee **FIRST\_NAME** - First name of the employee **LAST\_NAME** - Last name of the employee **GENDER** - Gender of the employee **ROLE** - Post of the employee **DEPT** - Field of the employee **Dataset Description Variable** - **Description** **EXP** - Years of experience the employee has **COUNTRY** - The country in which the employee is presently living **CONTINENT** - Continent in which the country is **SALARY**

- Salary of the employee **EMP\_RATING** - Performance rating of the employee **MANAGER\_ID** - The manager under which the employee is assigned **PROJ\_ID** - The project on which the employee is working or has worked on

**Dataset Description**

**Proj\_table:** It contains information about the projects. Variable - **Description** **PROJECT\_ID** - ID for the project **PROJ\_Name** - Name of the project **DOMAIN** - Field of the project **START\_DATE** - The day the project began **CLOSURE\_DATE** - Day the project was or will be completed **DEV\_QTR** - Quarter in which the project was scheduled **STATUS** - Current status of the project

**Dataset Description** **Data\_science\_team:** It contains information about all the employees in the Data Science team.

Variable - **Description** **EMP\_ID** - ID of the employee **FIRST\_NAME** - First name of the employee **LAST\_NAME** - Last name of the employee **GENDER** - Gender of the employee **ROLE** - Post of the employee **DEPT** - Field of the employee **EXP** - Years of experience the employee has **COUNTRY** - Country in which the employee is presently living **CONTINENT** - Continent in which the country is.

# Tasks to Perform

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

The screenshot shows the MySQL Workbench interface. In the left sidebar, under 'Schemas', there is a tree view with 'Local instance MySQL80'. Under 'Local instance MySQL80', there is a 'Tables' node which contains 'emp\_record\_table' and 'proj\_table'. Under 'Tables', there is a 'Employee' node which contains 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'Tables' node has 'emp' as its child. The 'emp' node has 'Columns' as its child, which lists columns such as EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY, and EMP\_RATING. The 'emp' node also has 'Indexes', 'Foreign Keys', 'Triggers', 'Stored Procedures', and 'Functions'. The 'Employee' node also has 'Indexes', 'Foreign Keys', 'Triggers', 'Stored Procedures', and 'Functions'. The 'Information' node is also present. In the center pane, there are two tabs: 'SQL File 3\*' and 'SQL File 4\*'. The 'SQL File 4\*' tab contains the following SQL code:

```
1 * Create Database employee;
2 * Use employee;
```

Below the tabs, there is an 'Output' section titled 'Action Output' with a table showing the results of the database creation and switching:

#	Time	Action	Message	Duration / Fetch
1	19:51:04	Create Database employee	1 row(s) affected	0.016 sec
2	19:53:19	Use employee	0 row(s) affected	0.000 sec

The screenshot shows the MySQL Workbench interface. In the left sidebar, under 'Schemas', there is a tree view with 'Local instance MySQL80'. Under 'Local instance MySQL80', there is a 'Tables' node which contains 'emp\_record\_table'. Under 'Tables', there is a 'Employee' node which contains 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'Tables' node has 'emp' as its child. The 'emp' node has 'Columns' as its child, which lists columns such as EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY, and EMP\_RATING. The 'emp' node also has 'Indexes', 'Foreign Keys', 'Triggers', 'Stored Procedures', and 'Functions'. The 'Employee' node also has 'Indexes', 'Foreign Keys', 'Triggers', 'Stored Procedures', and 'Functions'. The 'Information' node is also present. In the center pane, there are two tabs: 'SQL File 3\*' and 'SQL File 4\*'. The 'SQL File 4\*' tab contains the following SQL code:

```
1 * select * from emp_record_table;
```

Below the tabs, there is a 'Result Grid' tab showing the data imported from 'emp\_record\_table'. The data consists of 15 rows of employee records. The columns are: EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY, and EMP\_RATING. The data includes various roles like SENIOR DATA SCIENTIST, JUNIOR DATA SCIENTIST, ASSOCIATE DATA SCIENTIST, MANAGER, and RETAIL, across different departments like AUTOMOTIVE, FINANCE, HEALTHCARE, and RETAIL, in countries like USA, Canada, Germany, France, India, China, and Colombia, and continents like North America, Europe, and South America. The salary ranges from 7000 to 16500. The 'Output' section at the bottom shows the results of the 'select' query:

#	Time	Action	Message	Duration / Fetch
1	19:58:01	select * from emp_record_table LIMIT 0, 500	15 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

SCHEMAS

- employee
  - Tables
    - data\_science\_team
    - emp\_record\_table
    - proj\_table
  - Views
  - Stored Procedures
  - Functions
- sys

SQL File 3\* SQL File 4\*

```
1 • select * from proj_table;
```

Result Grid | Filter Rows: Export: Wrap Cell Content: 1

PROJ_ID	PROJ_NAME	DOMAIN	START_DATE	CLOSURE_DATE	DEV_QTR	STATUS
P003	Drug Discovery	HEALTHCARE	04-05-2021	6/20/2021	Q1	DONE
P005	Fraud Detection	FINANCE	04-11-2021	6/25/2021	Q1	DONE
P008	Algorithmic Trading	FINANCE	01/16/2022	3/27/2022	Q4	YTS
P009	Market Basket Analysis	RETAIL	04-12-2021	6/30/2021	Q1	DELAYED
P024	Supply Chain Management	AUTOMOTIVE	07/15/2021	9/28/2021	Q2	WIP
P006	Customer Sentiment Analysis	RETAIL	07-09-2021	9/24/2021	Q2	WIP
P002	Early Detection of Lung Cancer	HEALTHCARE	10-08-2021	12/18/2021	Q3	YTS
P001	Self Driving Cars	AUTOMOTIVE	01-12-2022	3/30/2022	Q4	YTS

Administration Schemas Information

Table: proj\_table

Columns:

- PROJ\_ID text
- PROJ\_NAME text
- DOMAIN text
- START\_DATE text
- CLOSURE\_DATE text
- DEV\_QTR text
- STATUS text

proj\_table 5 x

Output

Action Output

- # Time Action
- 1 19:30:00 selected \* from proj\_table LIMIT 0, 500

Message 0 row(s) returned Duration / Fetch 0.000 sec / 0.000 sec

Object Info Session

Query Completed

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

SCHEMAS

- employee
  - Tables
    - data\_science\_team
    - emp\_record\_table
    - proj\_table
  - Views
  - Stored Procedures
  - Functions
- sys

SQL File 3\* SQL File 4\*

```
1 • select * from data_science_team;
```

Result Grid | Filter Rows: Export: Wrap Cell Content: 1

BIP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	BIP	COUNTRY	CONTINENT
E000	Roy	Collins	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA
E005	Nan	Zhen	M	SENIOR DATA SCIENTIST	RETAIL	6	CHINA	ASIA
E010	Karina	Allen	F	JUNIOR DATA SCIENTIST	RETAIL	2	INDIA	ASIA
E040	Jenifer	Jones	F	JUNIOR DATA SCIENTIST	RETAIL	1	COLOMBIA	SOUTH AMERICA
E403	Steve	Hoffman	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA
E004	Karenne	Novak	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE
E057	Dorothy	Vilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA
E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE
E478	David	Smith	M	ASSOCIATE DATA SCIENTIST	RETAIL	3	COLOMBIA	SOUTH AMERICA
E005	Eric	Huffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA
E052	Donna	Vilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA
E005	Chad	Vilson	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	3	CANADA	NORTH AMERICA
E032	Olare	Brennan	F	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	3	GERMANY	EUROPE

data\_science\_team 4 x

Output

Action Output

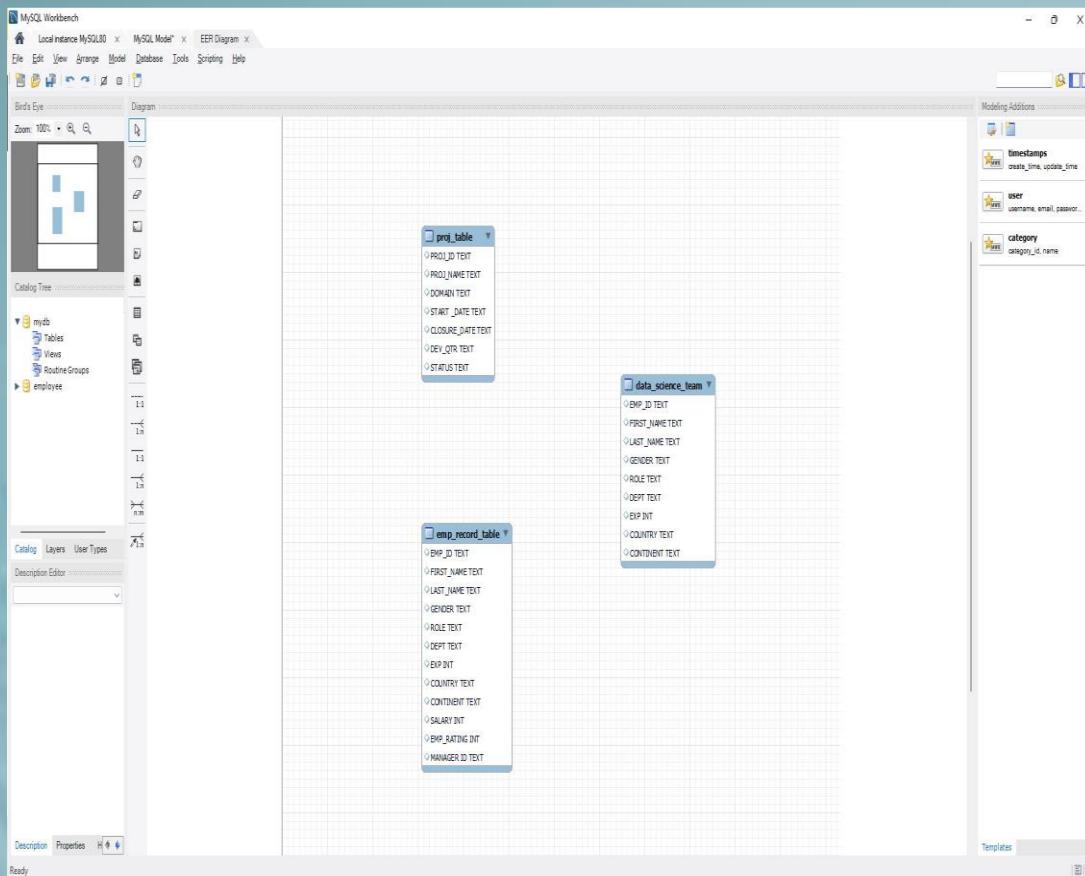
- # Time Action
- 1 19:25:38 selected \* from data\_science\_team LIMIT 0, 500

Message 13 row(s) returned Duration / Fetch 0.000 sec / 0.000 sec

Object Info Session

Query Completed

## 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**

```

MySQL Workbench - Local instance MySQL80 - MySQL Model* - EER Diagram *
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Schemas
Filter objects
employee
+-- data_science_team
+-- emp_record_table
    +-- Columns
        +-- EMP_ID
        +-- FIRST_NAME
        +-- LAST_NAME
        +-- GENDER
        +-- ROLE
        +-- DEPT
        +-- SALARY
        +-- COUNTRY
        +-- EMP_RATING
        +-- MANAGER_ID
    +-- Indexes
    +-- Foreign Keys
    +-- Triggers
    +-- Constraints
    +-- Views
Administration Schemas Information
Tables: emp_record_table
Columns:
EMP_ID text
FIRST_NAME text
LAST_NAME text
GENDER text
ROLE text
DEPT text
SALARY text
COUNTRY text
COUNTRY text
SALARY text
RATING int
MANAGER_ID text
Table: emp_record_table 6 < x>
Output
Action Output
1 19:38:59 select EMP_ID, FIRST_NAME, LAST_NAME, GENDER,DEPT from emp_record_table LIMIT 0, 500
Message
19 row(s) returned
Duration / Fetch
0.000 sec / 0.000 sec
Object Info Session
Query Completed

```

**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

The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```
1 • select EMP_ID, FIRST_NAME, LAST_NAME, GENDER,DEPT,EMP_RATING from emp_record_table where EMP_RATING <2 or EMP_RATING>4 or EMP_RATING between 2 and 4;
```

The results grid displays 15 rows of employee data. The columns are: EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, and EMP\_RATING. The data includes various employees like Roy Colins, Nan Chen, etc., across different departments like RETAIL, FINANCE, and AUTOMOTIVE.

**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.**

The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```
1 • select concat(FIRST_NAME, ' ', LAST_NAME) as NAME from emp_record_table where DEPT='FINANCE';
```

The results grid displays 3 rows of concatenated employee names. The columns are: NAME. The data includes Steve Hoffman, Brian Hoffman, and Emily Grove.

## 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 e1.manager_id,
       count(*)
  FROM emp_record_table e1,
       emp_record_table e2
 WHERE e1.manager_id = e2.emp_id
 GROUP BY e1.manager_id
 ORDER BY e1.manager_id ASC;
    
```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** Local instance MySQL80, employee schema selected.
- Tables:** emp\_record\_table selected.
- Query Editor:**

```

SELECT e1.manager_id,
       count(*)
  FROM emp_record_table e1,
       emp_record_table e2
 WHERE e1.manager_id = e2.emp_id
 GROUP BY e1.manager_id
 ORDER BY e1.manager_id ASC;
    
```
- Result Grid:**

manager_id	count(*)
E001	6
E003	2
E103	2
E439	3
E583	3
E512	2
- Message:** 6 row(s) returned.
- Execution Plan:** Read Only.

## 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 from emp_record_table where dept='FINANCE'
union
select emp_id from emp_record_table where dept='HEALTHCARE';
    
```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** Local instance MySQL80, employee schema selected.
- Tables:** emp\_record\_table selected.
- Query Editor:**

```

select emp_id from emp_record_table where dept='FINANCE'
union
select emp_id from emp_record_table where dept='HEALTHCARE';
    
```
- Result Grid:**

emp_id
E400
E005
E103
E557
E525
E583
- Message:** 7 row(s) returned.
- Execution Plan:** Read Only.

**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.**

```
1 • select EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EMP_RATING, max(EMP_RATING) over(partition by dept) from emp_record_table ;
```

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EMP_RATING	max(EMP_RATING) over(partition by dept)
E001	Arthur	Black	PRESIDENT	AU	5	5
E201	Karen	Nosik	SENIOR DATA SCIENTIST	AUTOMOTIVE	5	5
E101	Willan	Budden	LEAD DATA SCIENTIST	AUTOMOTIVE	2	5
E301	Celine	Brennan	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	1	5
E401	Pete	Allison	MANAGER	AUTOMOTIVE	4	5
E005	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	FINNANCE	3	4
E005	Eric	Hoffman	LEAD DATA SCIENTIST	FINNANCE	3	4
E205	Darryl	Groves	MANAGER	FINNANCE	4	4
E017	Dorothy	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	1	5
E052	Dianne	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	5	5
E053	Chad	Wilson	ASSOCIATE DATA SCIENTIST	HEALTHCARE	2	5
E058	Willie	Wilson	MANAGER	HEALTHCARE	3	5
E260	Roy	Collins	SENIOR DATA SCIENTIST	RETAIL	3	4
E245	Nan	Zhen	SENIOR DATA SCIENTIST	RETAIL	2	4
E620	Katrina	Allen	JUNIOR DATA SCIENTIST	RETAIL	1	4
E640	Travis	Allen	ASSOCIATE DATA SCIENTIST	RETAIL	4	4
E478	David	Smith	ASSOCIATE DATA SCIENTIST	RETAIL	4	4
E583	Janet	Hale	MANAGER	RETAIL	2	4
E612	Tracy	Norris	MANAGER	RETAIL	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**

```
1 • select min(salary),max(salary) from emp_record_table group by role;
```

min(salary)	max(salary)
5500	7700
5500	8000
4000	5000
8500	9000
8500	11000
18000	18000

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

The screenshot shows the MySQL Workbench interface with a query window containing the following SQL code:

```
1 select emp_id,first_name,role,exp,rank() over(order by exp) emp_rank from emp_record_table;
```

The results grid displays the following data:

emp_id	first_name	role	exp	emp_rank
E640	Jenifer	JUNIOR DATA SCIENTIST	1	2
E620	Kathy	JUNIOR DATA SCIENTIST	2	2
E478	David	ASSOCIATE DATA SCIENTIST	3	3
E479	Chris	ASSOCIATE DATA SCIENTIST	3	3
E403	Steve	ASSOCIATE DATA SCIENTIST	4	5
E505	Chad	ASSOCIATE DATA SCIENTIST	5	6
E245	Han	SENIOR DATA SCIENTIST	6	7
E504	David	SENIOR DATA SCIENTIST	7	7
E360	Ray	SENIOR DATA SCIENTIST	7	9
E304	Karen	SENIOR DATA SCIENTIST	8	10
E557	Dorothy	SENIOR DATA SCIENTIST	9	11
E510	Willam	LEAD DATA SCIENTIST	11	13
E512	Tracy	MANAGER	12	13
E583	Janet	MANAGER	14	14
E428	Mike	MANAGER	14	15
E429	Pete	MANAGER	14	15
E983	Patrick	MANAGER	15	18
E801	Arthur	PRESIDENT	20	29

The status bar at the bottom indicates "Duration / Fetch 0.000 sec / 0.000 sec".

## 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.

The screenshot shows the MySQL Workbench interface with a query window containing the following SQL code:

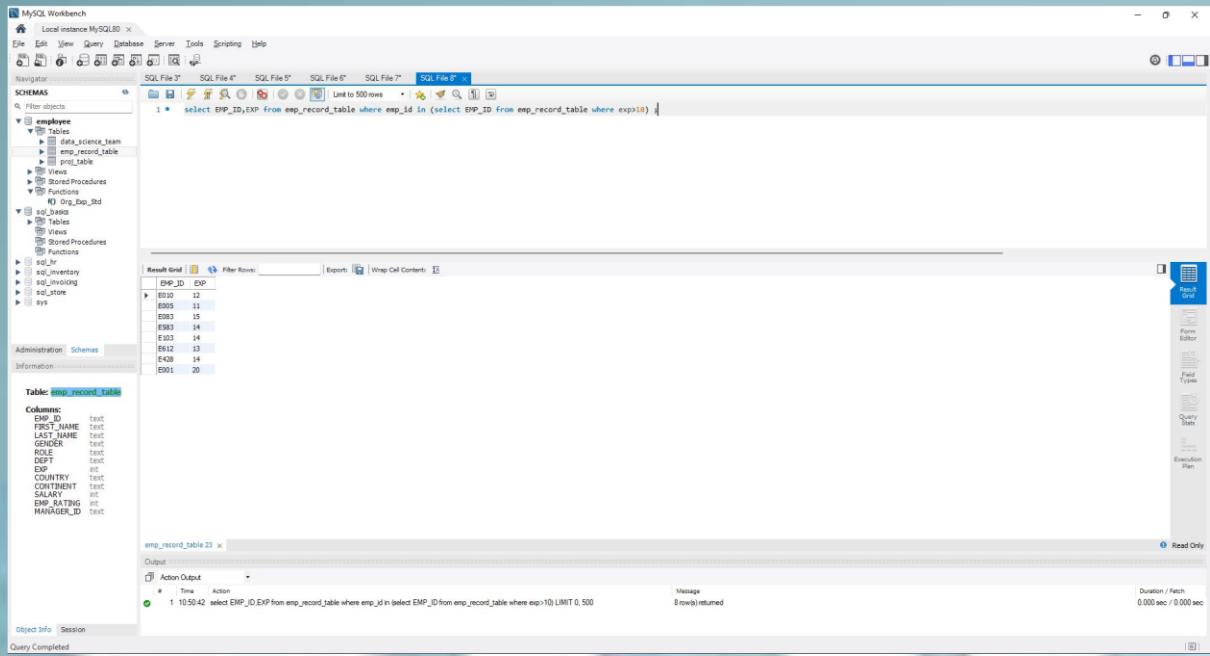
```
1 Create view emp_country as select * from emp_record_table where salary > 6000;
```

The results grid displays the following data:

emp_id	first_name	last_name	role	dept	country	city	salary
E640	Jenifer	Jenifer	JUNIOR DATA SCIENTIST	DATA SCI	USA	NEW YORK	50000
E620	Kathy	Kathy	JUNIOR DATA SCIENTIST	DATA SCI	USA	NEW YORK	50000
E478	David	David	ASSOCIATE DATA SCIENTIST	DATA SCI	USA	NEW YORK	60000
E479	Chris	Chris	ASSOCIATE DATA SCIENTIST	DATA SCI	USA	NEW YORK	60000
E403	Steve	Steve	ASSOCIATE DATA SCIENTIST	DATA SCI	USA	NEW YORK	60000
E505	Chad	Chad	ASSOCIATE DATA SCIENTIST	DATA SCI	USA	NEW YORK	60000
E245	Han	Han	SENIOR DATA SCIENTIST	DATA SCI	USA	NEW YORK	70000
E504	David	David	SENIOR DATA SCIENTIST	DATA SCI	USA	NEW YORK	70000
E360	Ray	Ray	SENIOR DATA SCIENTIST	DATA SCI	USA	NEW YORK	70000
E304	Karen	Karen	SENIOR DATA SCIENTIST	DATA SCI	USA	NEW YORK	70000
E557	Dorothy	Dorothy	SENIOR DATA SCIENTIST	DATA SCI	USA	NEW YORK	70000
E510	Willam	Willam	LEAD DATA SCIENTIST	DATA SCI	USA	NEW YORK	80000
E512	Tracy	Tracy	MANAGER	DATA SCI	USA	NEW YORK	90000
E583	Janet	Janet	MANAGER	DATA SCI	USA	NEW YORK	90000
E428	Mike	Mike	MANAGER	DATA SCI	USA	NEW YORK	90000
E429	Pete	Pete	MANAGER	DATA SCI	USA	NEW YORK	90000
E983	Patrick	Patrick	MANAGER	DATA SCI	USA	NEW YORK	90000
E801	Arthur	Arthur	PRESIDENT	DATA SCI	USA	NEW YORK	120000

The status bar at the bottom indicates "Duration / Fetch 0.015 sec / 0.000 sec".

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



The screenshot shows the MySQL Workbench interface with a query results window. The query is:`1 • select EMP_ID, EXP from emp_record_table where emp_id in (select EMP_ID from emp_record_table where exp>10)`

The result grid displays the following data:

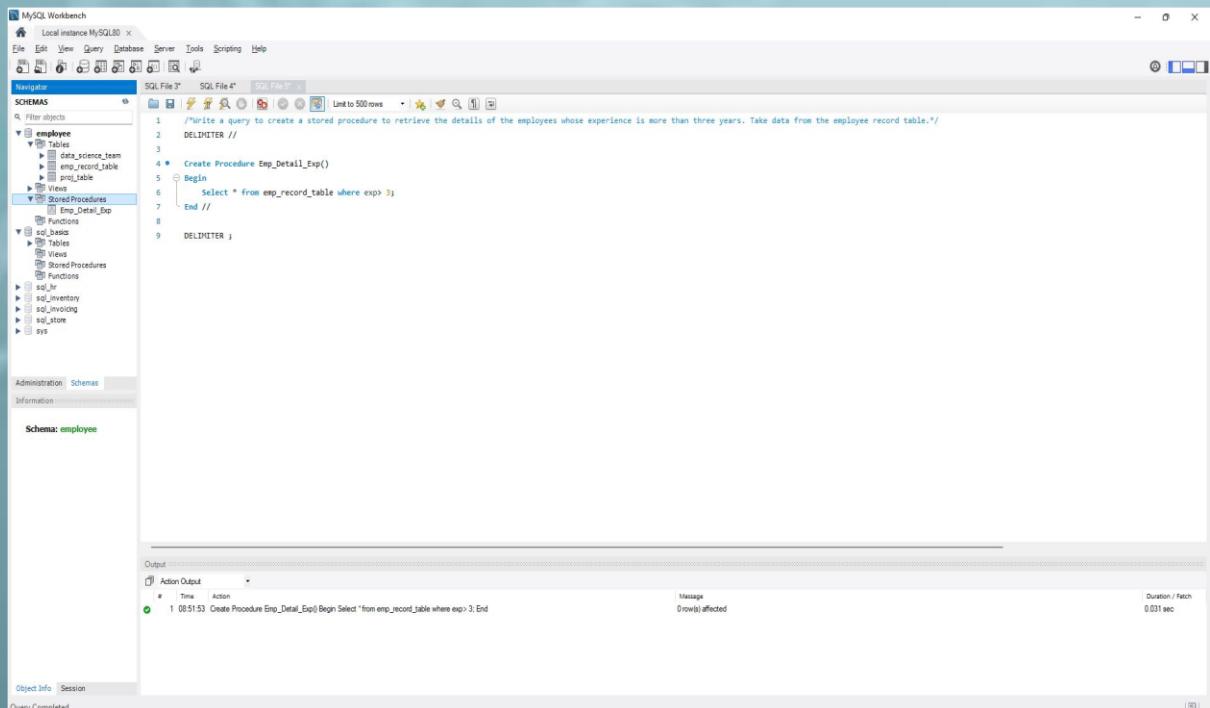
EMP_ID	EXP
E010	12
E005	11
E003	15
E083	14
E103	14
E612	13
E420	14
E001	20

Below the results, the schema browser shows the `emp_record_table` structure:

```
Table: emp_record_table
Columns:
EMP_ID          text
FIRST_NAME       text
LAST_NAME        text
GENDER           text
MIDDLE_NAME      text
DEPT             text
EXP              int
COUNTRY          text
CONTINENT        text
SALARY           int
EMP_RATING      int
MANAGER_ID       text
```

The status bar indicates "Query Completed".

**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.**



The screenshot shows the MySQL Workbench interface with a query results window. The query is:`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.*/
2 DELIMITER //
3
4 * Create Procedure Emp_Detail_Exp()
5 Begin
6 Select * from emp_record_table where exp>3;
7 End //
8
9 DELIMITER ;`

The result grid displays the message:

```
1 08:51:53 Create Procedure Emp_Detail_Exp() Begin Select * from emp_record_table where exp>3; End
Message
0 rows affected
Duration / Fetch
0.031 sec
```

The status bar indicates "Query Completed".

```

MySQL Workbench
File Edit View Query Database Server Tools Help
File SQL File 4*
Local instance MySQL80
Limit to 500 rows
1 DELIMITER //
2
3
4 * Create Procedure Emp_Detail_Exp()
5
6     Begin
7         Select * From emp_record_table where exp > 3;
8     End //
9
10 DELIMITER ;
11
12 Call Emp_Detail_Exp();

```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	COUNTRY	CONTINENT	SALARY	EMP_RATING	MANAGER_ID	
E060	Roy	Colins	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA	7000	3	E981
E245	Nan	Zhen	M	SENIOR DATA SCIENTIST	RETAIL	6	CHINA	ASIA	6000	2	E981
E024	Osiris	Holloway	M	ASSOCIATE DATA SCIENTIST	FINANCIAL	4	USA	NORTH AMERICA	5000	3	E981
E004	Karlene	Novakoli	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE	7500	5	E428
E057	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA	7700	1	E083
E008	Patricia	Burnham	M	LEAD DATA SCIENTIST	HEALTHCARE	10	FRANCE	EUROPE	8500	4	E428
E005	Eric	Huffman	M	LEAD DATA SCIENTIST	FINANCIAL	11	USA	NORTH AMERICA	8500	3	E103
E052	Diane	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	5500	5	E083
E002	Chris	Willard	M	ASSOCIATE DATA SCIENTIST	FINANCIAL	12	CANADA	NORTH AMERICA	5000	3	E981
E083	Patrิก	Valtz	M	MANAGER	HEALTHCARE	13	USA	NORTH AMERICA	9500	5	E001
E083	Jonet	Hole	F	MANAGER	RETAIL	14	COLOMBIA	SOUTH AMERICA	12000	2	E001
E083	Erin	Shaffer	M	MANAGER	FINANCIAL	15	CANADA	NORTH AMERICA	10000	3	E981
E012	Tracy	Norris	F	MANAGER	RETAIL	13	INDIA	ASIA	8500	4	E001
E028	Pete	Allen	M	MANAGER	AUTOMOTIVE	14	GERMANY	EUROPE	12000	4	E001
E001	Arthur	Black	M	PRESIDENT	ALL	20	USA	NORTH AMERICA	18500	5	E001

Result 7:

Action	Time	Message	Duration / Fetch
Action Output	1 08:51:53	Create Procedure Emp_Detail_Exp() Begin Selected * from emp_record_table where exp > 3; End	0 row(s) affected 0.031 sec 0.000 sec / 0.000 sec
2 08:53:57	Call Emp_Detail_Exp()	15 row(s) returned	

**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. The standard is given as follows:**

- Employee with experience less than or equal to 2 years, assign 'JUNIOR DATA SCIENTIST'
- Employee with experience of 2 to 5 years, assign 'ASSOCIATE DATA SCIENTIST'
- Employee with experience of 5 to 10 years, assign 'SENIOR DATA SCIENTIST'
- Employee with experience of 10 to 12 years, assign 'LEAD DATA SCIENTIST',
- Employee with experience of 12 to 16 years, assign 'MANAGER'

MySQL Workbench

Local instance MySQL00 X

File Edit View Database Server Tools Scripting Help

Schemas

employee

Tables

Views

Stored Procedures

Functions

Org\_Exp\_Std

sql\_hr

sql\_inventory

sql\_invoicing

sql\_store

sys

Administration Schemas Information

```

1  DELIMITER //
2  • Create Function Org_Exp_Std( exp_year int)
3   Returns Varchar(25)
4   Deterministic
5   Begin
6     Declare role_found varchar(25);
7     If exp_year<=2 Then
8       Select distinct role into role_found from emp_record_table where exp >exp_years;
9     Elseif (exp_year >2 And exp_year<=5) Then
10      Select distinct role into role_found from emp_record_table where exp >exp_years;
11    Elseif (exp_year >5 And exp_year<=10) Then
12      Select distinct role into role_found from emp_record_table where exp >exp_years;
13    Elseif (exp_year >10 And exp_year<=12) Then
14      Select distinct role into role_found from emp_record_table where exp >exp_years;
15    Elseif (exp_year >12 And exp_year<=15) Then
16      Select distinct role into role_found from emp_record_table where exp >exp_years;
17    Elseif (exp_year >15 And exp_year<=18) Then
18      Select distinct role into role_found from emp_record_table where exp >exp_years;
19    End If;
20    Return (role_found);
21  End //
22
23  DELIMITER $$
```

Output

Action Output

# Time Action Message Duration / Fetch

1 09:55:44 Create Function Org\_Exp\_Std(exp\_year int) Returns Varchar(25) Deterministic Begin Declare role\_found varchar(25); If exp\_year <=2 Then Select d... 0 row(s) affected 0.016 sec

Object Info Session Query Completed

MySQL Workbench

Local instance MySQL00 X

File Edit View Database Server Tools Scripting Help

Schemas

employee

Tables

Views

Stored Procedures

Functions

Org\_Exp\_Std

sql\_hr

sql\_inventory

sql\_invoicing

sql\_store

sys

Administration Schemas Information

```

3   Returns Varchar(25)
4   Deterministic
5   Begin
6     Declare role_found varchar(25);
7     If exp_year<=2 Then
8       Select distinct role into role_found from emp_record_table where exp >exp_years;
9     Elseif (exp_year >2 And exp_year<=5) Then
10      Select distinct role into role_found from emp_record_table where exp >exp_years;
11    Elseif (exp_year >5 And exp_year<=10) Then
12      Select distinct role into role_found from emp_record_table where exp >exp_years;
13    Elseif (exp_year >10 And exp_year<=12) Then
14      Select distinct role into role_found from emp_record_table where exp >exp_years;
15    Elseif (exp_year >12 And exp_year<=15) Then
16      Select distinct role into role_found from emp_record_table where exp >exp_years;
17    End If;
18    Return (role_found);
19  End //
20
21  DELIMITER $$
```

Result Grid

Table: emp\_record\_table

Columns:

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_RATING	MANAGER_ID

Result 4 <

Output

Action Output

# Time Action Message Duration / Fetch

1 09:55:44 Create Function Org\_Exp\_Std(exp\_year int) Returns Varchar(25) Deterministic Begin Declare role\_found varchar(25); If exp\_year <=2 Then Select d... 0 row(s) affected 0.016 sec

2 09:58:08 Call Org\_Exp\_Std()

3 09:59:59 select Org\_Exp\_Std(exp) from emp\_record\_table LIMIT 0..500 Error Code: 1305. PROCEDURE employee.Org\_Exp\_Std does not exist 0.000 sec / 0.000 sec

Object Info Session Query Completed

## 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.

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'File', 'Edit', 'View', 'Query', 'Database', 'Server', 'Tools', 'Scripting', and 'Help' are visible. Below the menu is a toolbar with various icons. The 'Schemas' pane on the left lists databases like 'Local instance MySQL80', 'employee', 'data\_science\_team', 'emp\_record\_table', 'prod\_table', 'views', 'emp\_country', 'Stored Procedures', 'Functions', 'sys', 'sql\_hr', 'sql\_inventory', 'sql\_invoicing', 'sql\_store', and 'sys'. The main area shows a SQL editor with the following code:

```
1 Create_Index F_Name_INDEX on emp_record_table(first_name(30))
```

Below the SQL editor is a table definition for 'Table: emp\_record\_table' with columns: EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY, RATING, and MANAGER\_ID. The 'Output' section shows the execution of the command:

```
1 11:21:40 Create Index F_Name_INDEX on emp_record_table(first_name(30))
```

Message: 0 rows affected Records: 0 Duplicates: 0 Warnings: 0 Duration / Fetch: 0.110 sec

## 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).

The screenshot shows the MySQL Workbench interface. The top navigation bar and toolbar are identical to the previous screenshot. The 'Schemas' pane lists the same databases. The main area shows a SQL editor with the following query:

```
1 /*5% of salary * employee rating*/
2 *
3 * select salary, emp_rating,round(((S * Salary * Emp_rating)/100),2) as Bonus from emp_record_table;
```

Below the SQL editor is a table definition for 'Table: emp\_record\_table' with the same columns as before. The 'Output' section shows the results of the query:

salary	emp_rating	Bonus
7000	3	1050.00
6500	2	650.00
5500	3	825.00
2800	4	560.00
5000	3	750.00
7500	5	1275.00
5500	4	995.00
9000	2	900.00
4000	4	800.00
5500	3	825.00
5000	5	1275.00
5000	2	500.00
4500	1	225.00
10000	2	1000.00
10500	4	2100.00
12000	3	1200.00
13000	4	2200.00
18500	5	4125.00

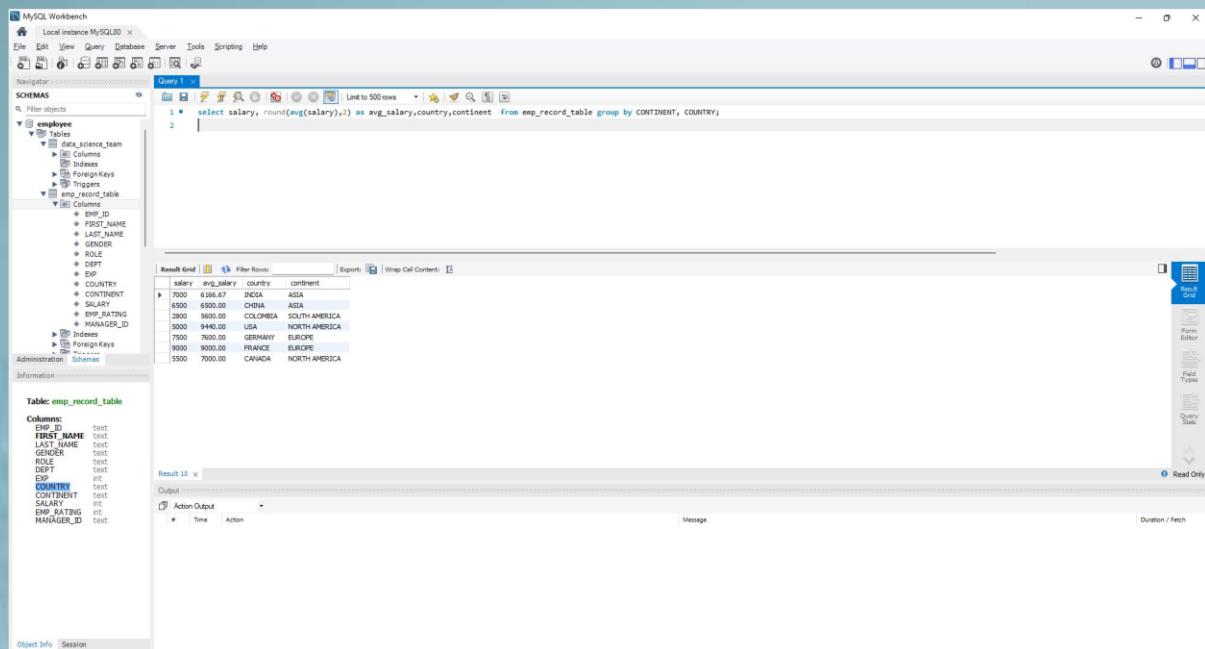
Result 29 x

Output:

```
1 11:34:51 select salary, emp_rating,round(((S * Salary * Emp_rating)/100),2) as Bonus from emp_record_table LIMIT 0,500
```

Message: 19 rows returned Duration / Fetch: 0.000 sec / 0.000 sec

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



The screenshot shows the MySQL Workbench interface with a query editor window. The query is:`1 • select salary, round(avg(salary),2) as avg_salary,country,continent from emp_record_table group by CONTINENT, COUNTRY;`

The result grid displays the following data:

salary	avg_salary	country	continent
7000	6366.67	INDIA	ASIA
6500	6500.00	CHINA	ASIA
2800	2800.00	SRI LANKA	ASIA
5000	9440.00	USA	NORTH AMERICA
7500	7600.00	GERMANY	EUROPE
9000	9000.00	FRANCE	EUROPE
5500	7000.00	CANADA	NORTH AMERICA

## Project Outcome:

- This project's goals are to determine the employees' maximum pay and confirm that every position satisfies the organization's profile criterion.
- This project involves calculating bonuses by determining the additional costs of the expenses.



**THANK YOU**