

# ScienceQtech Employee Performance Mapping.

Course-end Project 1

## 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:

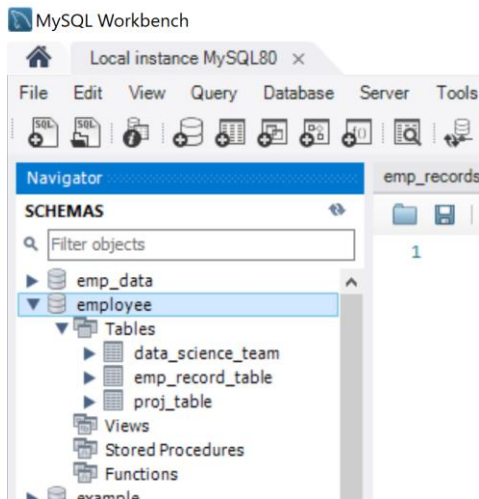
To facilitate a better understanding, managers have provided ratings for each employee which will help the HR department to finalize the employee performance mapping. As a DBA, you should find the maximum salary of the employees and ensure that all jobs are meeting the organization's profile standard.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

### The task to be performed:

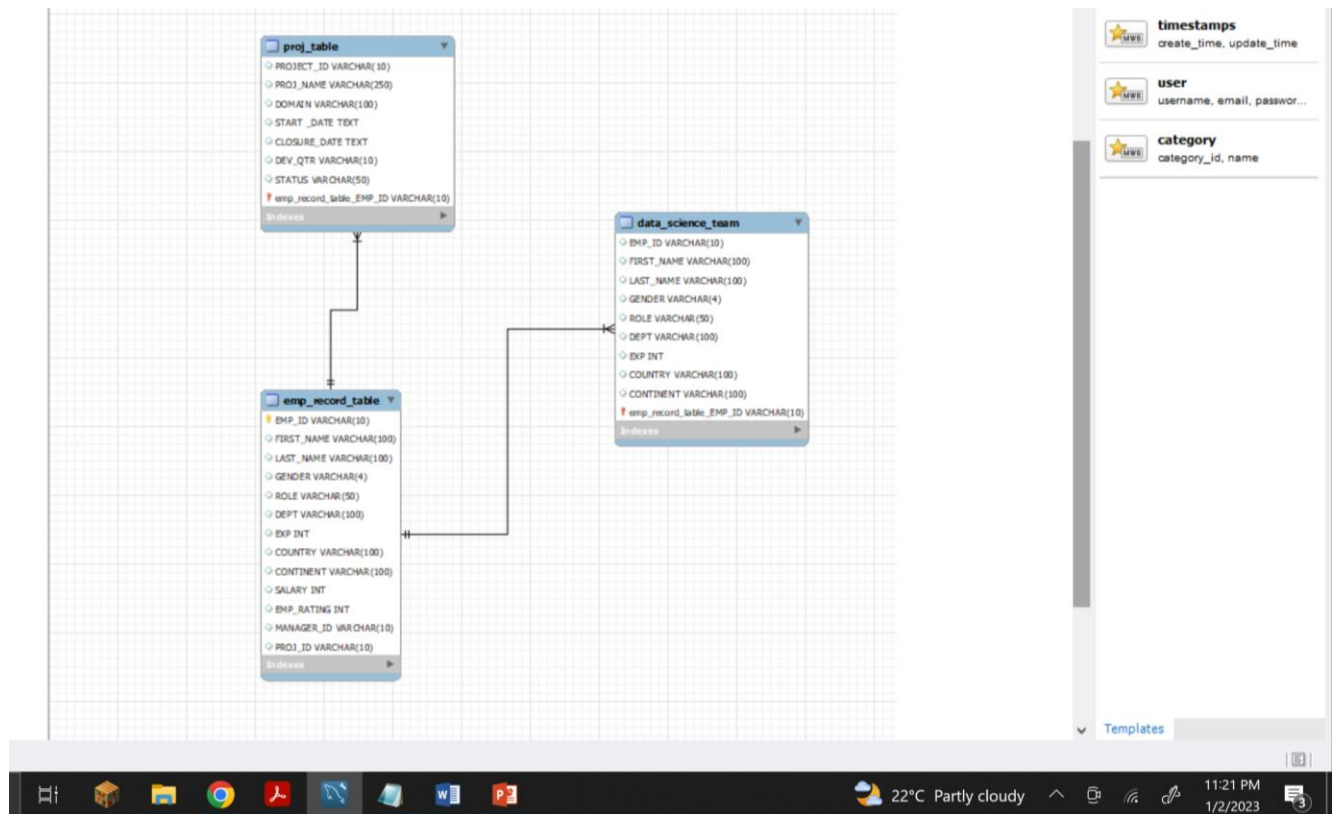
1. Create a database named **employee**, then import **data\_science\_team.csv**, **proj\_table.csv** and **emp\_record\_table.csv** into the **employee** database from the given resources.

CREATE DATABASE IF DOES NOT EXISTS **employee**;

Screenshot of imported tables:



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



- 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 x MySQL Model x EER Diagram x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- emp\_data
  - employee
    - Tables
      - data\_science\_team
      - emp\_record\_table
        - Columns
          - EMP\_ID
          - FIRST\_NAME
          - LAST\_NAME
          - GENDER
          - ROLE
          - DEPT
          - EXP
          - COUNTRY
          - CONTINENT
          - SALARY
          - EMP\_RATING
          - MANAGER\_ID
          - PROJ\_ID

Administration Schemas

Information

Schema: employee

emp\_records SQL File 10\*

Limit to 1000 rows

```

1 • USE employee;
2 • SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
3   FROM emp_record_table;
4

```

Result Grid

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
E103	Emily	Grove	F	FINANCE
E204	Karene	Nowak	F	AUTOMOTIVE
E245	Nian	Zhen	M	RETAIL
E260	Roy	Collins	M	RETAIL
E403	Steve	Hoffman	M	FINANCE
E428	Pete	Allen	M	AUTOMOTIVE
E478	David	Smith	M	RETAIL
E505	Chad	Wilson	M	HEALTHCARE
E532	Claire	Brennan	F	AUTOMOTIVE
E583	Janet	Hale	F	RETAIL
E612	Tracy	Norris	F	RETAIL
E620	Katrina	Allen	F	RETAIL
E640	Jenifer	Jhones	F	RETAIL

record\_table 1 x

Read Only

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

a. less than two

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'employee' schema selected. The central SQL editor contains the following query:

```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
FROM emp_record_table
WHERE EMP_RATING < 2;
```

The 'Result Grid' at the bottom displays the following data:

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT
E057	Dorothy	Wilson	F	HEALTHCARE
E532	Claire	Brennan	F	AUTOMOTIVE
E620	Katrina	Allen	F	RETAIL

b. greater than four

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'employee' schema selected. The central SQL editor contains the following query:

```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
FROM emp_record_table
WHERE EMP_RATING > 4;
```

The 'Result Grid' at the bottom displays the following data:

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT
E001	Arthur	Black	M	ALL
E052	Dianna	Wilson	F	HEALTHCARE
E083	Patrick	Voltz	M	HEALTHCARE
E204	Karene	Nowak	F	AUTOMOTIVE

c. between two and four

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'employee' schema selected. The central SQL editor contains the following query:

```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
FROM emp_record_table
WHERE EMP_RATING BETWEEN 2 AND 4;
```

The 'Result Grid' at the bottom displays the following data:

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT
E005	Eric	Hoffman	M	FINANCE
E010	William	Butler	M	AUTOMOTIVE
E103	Emily	Grove	F	FINANCE
E245	Nian	Zhen	M	RETAIL
E260	Roy	Collins	M	RETAIL
E403	Steve	Hoffman	M	FINANCE
E428	Pete	Allen	M	AUTOMOTIVE
E478	David	Smith	M	RETAIL
E505	Chad	Wilson	M	HEALTHCARE
E583	Janet	Hale	F	RETAIL
E612	Tracy	Norris	F	RETAIL
E640	Jenifer	Jones	F	RETAIL

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. The left sidebar displays the 'SCHEMAS' tree with the 'employee' schema selected. The main editor window shows a SQL query in the 'SQL File 10\*' tab:

```
16
17 • SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS 'NAME'
18 FROM emp_record_table
19 WHERE DEPT = 'FINANCE';
20
```

Below the query editor, the 'Result Grid' is visible, showing the results of the query:

NAME
Eric Hoffman
Emily Grove
Steve Hoffman

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

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'employee' schema selected. The main editor window shows a SQL query in the 'SQL File 10\*' tab:

```
20
21 • SELECT e.EMP_ID, e.FIRST_NAME, e.LAST_NAME, e.ROLE, e.DEPT, COUNT(e.EMP_ID) AS 'NO_OF_REPORTERS'
22 FROM emp_record_table AS e
23 INNER JOIN emp_record_table AS n
24 ON e.EMP_ID = n.MANAGER_ID
25 GROUP BY e.EMP_ID
26 ORDER BY e.EMP_ID;
27
28
29
```

Below the query editor, the 'Result Grid' is visible, showing the results of the query:

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	NO_OF_REPORTERS
E001	Arthur	Black	PRESIDENT	ALL	5
E083	Patrick	Voltz	MANAGER	HEALTHCARE	3
E103	Emily	Grove	MANAGER	FINANCE	2
E428	Pete	Allen	MANAGER	AUTOMOTIVE	3
E583	Janet	Hale	MANAGER	RETAIL	3
E612	Tracy	Norris	MANAGER	RETAIL	2

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.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a tree view of the database structure, including tables like 'emp\_record\_table' and columns like 'EMP\_ID', 'FIRST\_NAME', 'LAST\_NAME', 'ROLE', 'DEPT', 'EMP\_RATING', 'MANAGER\_ID', and 'PROJ\_ID'. The main editor window contains a SQL query that uses the UNION operator to combine data from the 'emp\_record\_table' for the 'HEALTHCARE' and 'FINANCE' departments. The query is as follows:

```
27
28 SELECT e.EMP_ID,e.FIRST_NAME, e.LAST_NAME, e.ROLE, e.DEPT
29 FROM emp_record_table AS e
30 WHERE DEPT = 'HEALTHCARE'
31 UNION
32 SELECT n.EMP_ID,n.FIRST_NAME, n.LAST_NAME, n.ROLE, n.DEPT
33 FROM emp_record_table AS n
34 WHERE DEPT = 'FINANCE';
35
36
```

The 'Result Grid' at the bottom displays the results of the query, showing a list of employees from both departments. The columns are EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, and DEPT. The results are as follows:

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT
E052	Dianna	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE
E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE
E083	Patrick	Voltz	MANAGER	HEALTHCARE
E505	Chad	Wilson	ASSOCIATE DATA SCIENTIST	HEALTHCARE
E005	Eric	Hoffman	LEAD DATA SCIENTIST	FINANCE
E103	Emily	Grove	MANAGER	FINANCE
E403	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	FINANCE

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.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a tree view of the database structure, including tables like 'emp\_record\_table' and columns like 'EMP\_ID', 'FIRST\_NAME', 'LAST\_NAME', 'ROLE', 'DEPT', 'EMP\_RATING', 'MANAGER\_ID', and 'PROJ\_ID'. The main editor window contains a SQL query that uses a window function to calculate the maximum employee rating for each department. The query is as follows:

```
35
36 SELECT EMP_ID,FIRST_NAME, LAST_NAME, ROLE, DEPT,EMP_RATING,
37 MAX(EMP_RATING) OVER(PARTITION BY DEPT) AS 'MAX_RATING'
38 FROM emp_record_table
39 ORDER BY DEPT;
40
41
42
```

The 'Result Grid' at the bottom displays the results of the query, showing a list of employees grouped by department. The columns are EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, DEPT, EMP\_RATING, and MAX\_RATING. The results are as follows:

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EMP_RATING	MAX_RATING
E001	Arthur	Black	PRESIDENT	ALL	5	5
E010	William	Butler	LEAD DATA SCIENTIST	AUTOMOTIVE	2	5
E204	Karene	Nowak	SENIOR DATA SCIENTIST	AUTOMOTIVE	5	5
E428	Pete	Allen	MANAGER	AUTOMOTIVE	4	5
E532	Claire	Brennan	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	1	5
E005	Eric	Hoffman	LEAD DATA SCIENTIST	FINANCE	3	4
E103	Emily	Grove	MANAGER	FINANCE	4	4
E403	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	FINANCE	3	4
E052	Dianna	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	5	5
E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	1	5
E083	Patrick	Voltz	MANAGER	HEALTHCARE	5	5
E505	Chad	Wilson	ASSOCIATE DATA SCIENTIST	HEALTHCARE	2	5
E245	Nian	Zhen	SENIOR DATA SCIENTIST	RETAIL	2	4
E260	Roy	Collins	SENIOR DATA SCIENTIST	RETAIL	3	4
E478	David	Smith	ASSOCIATE DATA SCIENTIST	RETAIL	4	4
E583	Janet	Hale	MANAGER	RETAIL	2	4
E612	Tracy	Norris	MANAGER	RETAIL	4	4
E620	Katrina	Allen	JUNIOR DATA SCIENTIST	RETAIL	1	4
E640	Jenifer	Jhones	JUNIOR DATA SCIENTIST	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.

The screenshot shows the MySQL Workbench interface. The 'SCHEMAS' pane on the left displays the database structure, including the 'emp\_record\_table' with columns like EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, DEPT, EXP, SALARY, EMP\_RATING, MANAGER\_ID, and PROJ\_ID. The 'SQL File 10\*' editor contains the following query:

```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EXP,
       MAX(SALARY) AS MAX_SALARY, MIN(SALARY) AS MIN_SALARY
FROM emp_record_table
GROUP BY ROLE
ORDER BY EXP DESC;
```

The 'Result Grid' pane shows the output of the query, displaying columns EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, DEPT, EXP, MAX\_SALARY, and MIN\_SALARY. The data is grouped by role and ordered by experience (EXP) in descending order.

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EXP	MAX_SALARY	MIN_SALARY
E001	Arthur	Black	PRESIDENT	ALL	20	16500	16500
E083	Patrick	Voltz	MANAGER	HEALTHCARE	15	11000	8500
E005	Eric	Hoffman	LEAD DATA SCIENTIST	FINANCE	11	9000	8500
E052	Dianna	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	6	7700	5500
E403	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	FINANCE	4	5000	4000
E620	Katrina	Allen	JUNIOR DATA SCIENTIST	RETAIL	2	3000	2800

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. The 'SCHEMAS' pane on the left displays the database structure, including the 'emp\_record\_table' with columns like EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, DEPT, EXP, SALARY, EMP\_RATING, MANAGER\_ID, and PROJ\_ID. The 'SQL File 10\*' editor contains the following query:

```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, SALARY, ROLE, EXP,
       RANK() OVER (ORDER BY EXP) EMP_RANK
FROM emp_record_table
GROUP BY EMP_ID
ORDER BY EXP DESC;
```

The 'Result Grid' pane shows the output of the query, displaying columns EMP\_ID, FIRST\_NAME, LAST\_NAME, SALARY, ROLE, EXP, and EMP\_RANK. The data is ordered by experience (EXP) in descending order, and each employee is assigned a rank based on their experience.

EMP_ID	FIRST_NAME	LAST_NAME	SALARY	ROLE	EXP	EMP_RANK
E001	Arthur	Black	16500	PRESIDENT	20	15
E083	Patrick	Voltz	9500	MANAGER	15	14
E103	Emily	Grove	10500	MANAGER	14	13
E612	Tracy	Norris	8500	MANAGER	13	12
E010	William	Butler	9000	LEAD DATA SCIENTIST	12	11
E005	Eric	Hoffman	8500	LEAD DATA SCIENTIST	11	10
E057	Dorothy	Wilson	7700	SENIOR DATA SCIENTIST	9	9
E204	Karene	Nowak	7500	SENIOR DATA SCIENTIST	8	8
E260	Roy	Collins	7000	SENIOR DATA SCIENTIST	7	7
E052	Dianna	Wilson	5500	SENIOR DATA SCIENTIST	6	6
E505	Chad	Wilson	5000	ASSOCIATE DATA SCIENTIST	5	5
E403	Steve	Hoffman	5000	ASSOCIATE DATA SCIENTIST	4	4
E478	David	Smith	4000	ASSOCIATE DATA SCIENTIST	3	3
E620	Katrina	Allen	3000	JUNIOR DATA SCIENTIST	2	2
E640	Jenifer	Jhones	2800	JUNIOR DATA SCIENTIST	1	1



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. The left sidebar displays the 'SCHEMAS' tree with 'data\_science\_team' and 'emp\_record\_table' expanded. The 'Columns' list for 'emp\_record\_table' includes EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY, EMP\_RATING, MANAGER\_ID, and PROJ\_ID. The 'Information' tab shows the definition for 'MANAGER\_ID' as 'MANAGER\_ID varchar(10)'. The main editor shows a SQL query in the 'SQL File 10\*' tab:

```
52
53 • SELECT EMP_ID, FIRST_NAME, LAST_NAME, SALARY, ROLE, COUNTRY
54 FROM emp_record_table
55 WHERE SALARY > 6000
56 ORDER BY SALARY DESC;
57
58
59
```

The 'Result Grid' at the bottom displays the following data:

EMP_ID	FIRST_NAME	LAST_NAME	SALARY	ROLE	COUNTRY
E001	Arthur	Black	16500	PRESIDENT	USA
E428	Pete	Allen	11000	MANAGER	GERMANY
E103	Emily	Grove	10500	MANAGER	CANADA
E583	Janet	Hale	10000	MANAGER	COLOMBIA
E083	Patrick	Voltz	9500	MANAGER	USA
E010	William	Butler	9000	LEAD DATA SCIENTIST	FRANCE
E005	Eric	Hoffman	8500	LEAD DATA SCIENTIST	USA
E612	Tracy	Norris	8500	MANAGER	INDIA
E057	Dorothy	Wilson	7700	SENIOR DATA SCIENTIST	USA
E204	Karene	Nowak	7500	SENIOR DATA SCIENTIST	GERMANY
E260	Roy	Collins	7000	SENIOR DATA SCIENTIST	INDIA
E245	Nian	Zhen	6500	SENIOR DATA SCIENTIST	CHINA

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. The left sidebar displays the 'SCHEMAS' tree with 'data\_science\_team' and 'emp\_record\_table' expanded. The 'Columns' list for 'emp\_record\_table' includes EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY, EMP\_RATING, MANAGER\_ID, and PROJ\_ID. The 'Information' tab shows the definition for 'MANAGER\_ID' as 'MANAGER\_ID varchar(10)'. The main editor shows a SQL query in the 'SQL File 10\*' tab:

```
52
53 • SELECT EMP_ID, FIRST_NAME, LAST_NAME, SALARY, ROLE, EXP
54 FROM emp_record_table
55 WHERE ROLE IN ('PRESIDENT', 'MANAGER') AND EXP > 10
56 ORDER BY EXP DESC;
57
58
59
```

The 'Result Grid' at the bottom displays the following data:

EMP_ID	FIRST_NAME	LAST_NAME	SALARY	ROLE	EXP
E001	Arthur	Black	16500	PRESIDENT	20
E083	Patrick	Voltz	9500	MANAGER	15
E103	Emily	Grove	10500	MANAGER	14
E428	Pete	Allen	11000	MANAGER	14
E583	Janet	Hale	10000	MANAGER	14
E612	Tracy	Norris	8500	MANAGER	13

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. The SQL editor contains the following code:

```
130 DELIMITER &&
131 CREATE PROCEDURE EMP_EXP()
132 BEGIN
133     SELECT *
134     FROM emp_record_table
135     WHERE EXP > 3;
136 END &&
137
138 CALL EMP_EXP();
```

The result grid displays the following data:

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_RATING	MANAGER_ID	PROJ_ID
E001	Arthur	Black	M	PRESIDENT	ALL	20	USA	NORTH AMERICA	16500	5	E001	NA
E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3	E103	P105
E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9000	2	E428	P204
E052	Dianne	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	5500	5	E083	P103
E057	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA	7700	1	E083	P302
E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500	5	E001	NA
E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	10500	4	E001	NA
E204	Karene	Nowak	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE	7500	5	E428	P204
E245	Nian	Zhen	M	SENIOR DATA SCIENTIST	RETAIL	6	CHINA	ASIA	6500	2	E583	P109
E260	Roy	Collins	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA	7000	3	E583	NA
E403	Steve	Hoffman	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA	5000	3	E103	P105
E428	Pete	Allen	M	MANAGER	AUTOMOTIVE	14	GERMANY	EUROPE	11000	4	E001	NA
E505	Chad	Wilson	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	CANADA	NORTH AMERICA	5000	2	E083	P103
E583	Janet	Hale	F	MANAGER	RETAIL	14	COLOMBIA	SOUTH AMERICA	10000	2	E001	NA
E612	Tracy	Norris	F	MANAGER	RETAIL	13	INDIA	ASIA	8500	4	E001	NA

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.

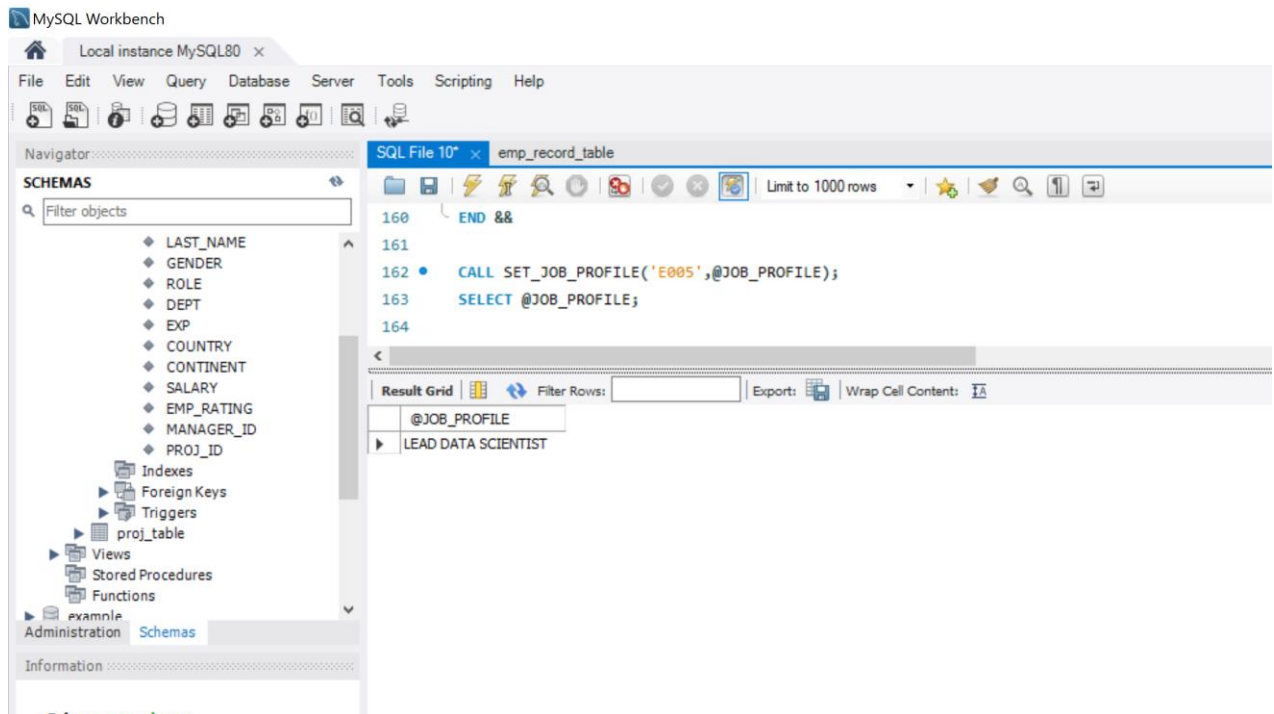
The standard being:

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'.

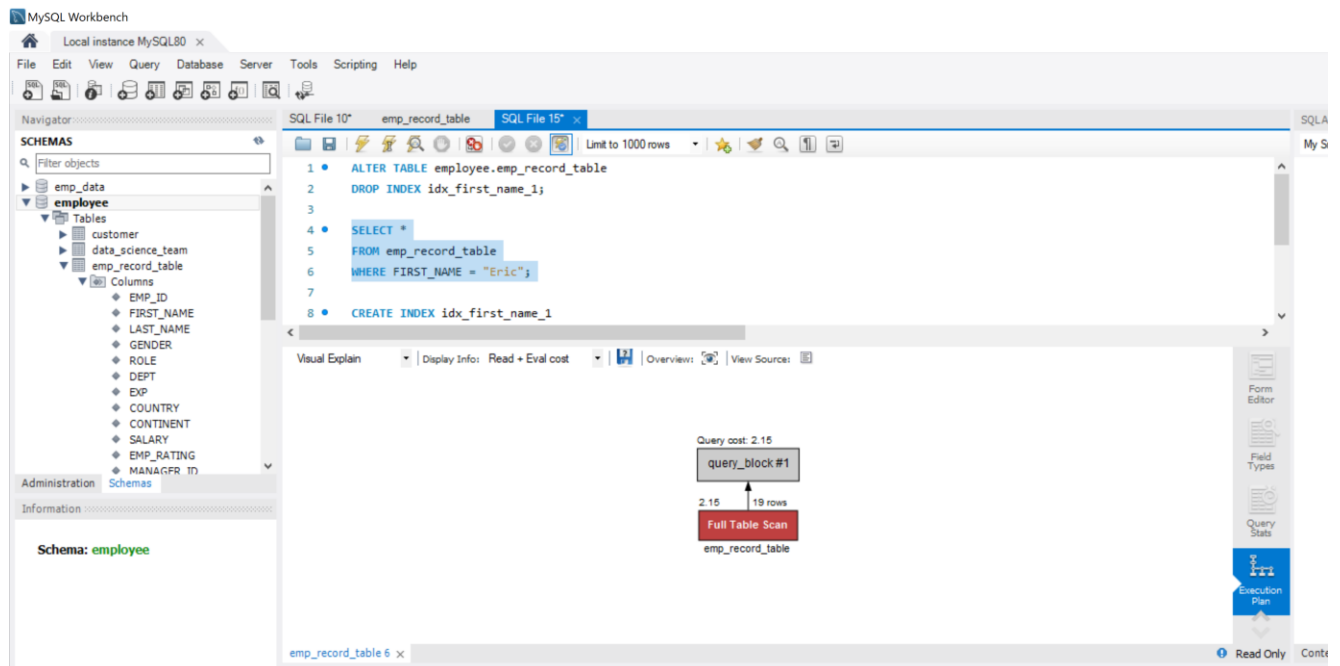
The screenshot shows the MySQL Workbench interface. The SQL editor contains the following code:

```
140 DROP PROCEDURE IF EXISTS SET_JOB_PROFILE
141 DELIMITER &&
142 CREATE PROCEDURE SET_JOB_PROFILE(IN EID VARCHAR(10), OUT JOB_PROFILE VARCHAR(100))
143 BEGIN
144     DECLARE YEARS_EXP INT;
145     SELECT EXP INTO YEARS_EXP FROM emp_record_table
146     WHERE EMP_ID = EID;
147     IF YEARS_EXP <= 2 THEN
148         SET JOB_PROFILE = 'JUNIOR DATA SCIENTIST';
149     ELSEIF YEARS_EXP BETWEEN 2 AND 5 THEN
150         SET JOB_PROFILE = 'ASSOCIATE DATA SCIENTIST';
151     ELSEIF YEARS_EXP BETWEEN 5 AND 10 THEN
152         SET JOB_PROFILE = 'SENIOR DATA SCIENTIST';
153     ELSEIF YEARS_EXP BETWEEN 10 AND 12 THEN
154         SET JOB_PROFILE = 'LEAD DATA SCIENTIST';
155     ELSEIF YEARS_EXP BETWEEN 12 AND 16 THEN
156         SET JOB_PROFILE = 'MANAGER';
157     ELSE
158         SET JOB_PROFILE = 'INVALID';
159     END IF;
160 END &&
```





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.



MySQL Workbench

Local instance MySQL80 x

Edit View Query Database Server Tools Scripting Help

Navigator

HEMAS

Filter objects

emp\_data

emp\_employee

Tables

customer

data\_science\_team

emp\_record\_table

Columns

EMP\_ID

FIRST\_NAME

LAST\_NAME

GENDER

ROLE

DEPT

EXP

COUNTRY

CONTINENT

SALARY

EMP\_RATING

MANAGER\_ID

Administration Schemas

Information

Schema: employee

SQL File 10" emp\_record\_table SQL File 15"

Limit to 1000 rows

1 CREATE INDEX idx\_first\_name\_1

2 ON emp\_record\_table(FIRST\_NAME);

3

4 SELECT \*

5 FROM emp\_record\_table

6 WHERE FIRST\_NAME = "Eric";

Visual Explain

Display Info: Read + Eval cost

Overview: View Sources

Query cost: 0.35

query\_block#1

0.35 1 row

Non-Unique Key Lookup

emp\_record\_table

idx\_first\_name

emp\_record\_table

Access Type: ref

Non-Unique Key Lookup

Cost Hint: Low-medium - Low if number of matching rows is small, higher as the number of rows increases.

Used Columns: EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY, EMP\_RATING, MANAGER\_ID, PROJ\_ID

Key/Index: idx\_first\_name

Ref.: const

Used Key Parts: FIRST\_NAME

Possible Keys: idx\_first\_name, idx\_first\_name\_1

Attached Condition:

("employee"."emp\_record\_table"."FIRST\_NAME" = "Eric")

Rows Examined per Scan: 1

Rows Produced per Join: 1

Filtered (ratio of rows produced per rows examined): 100.00%

Hint: 100% is best, <= 1% is worst

A low value means the query examines a lot of rows that are not returned.

Cost Info

Read: 0.25

Eval: 0.10

Prefix: 0.35

Data Read: 2K

record\_table 1 x

Output

Action Output

# Time Action

512 11:38:16 EXPLAIN SELECT \* FROM emp\_record\_table WHERE FIRST\_NAME = "Eric"

513 11:38:16 EXPLAIN FORMAT=JSON SELECT \* FROM emp\_record\_table WHERE FIRST\_NAME = "Eric"

Object Info Session

Query Completed

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).

MySQL Workbench

Local instance MySQL80 x MySQL Model" x EER Diagram x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

data\_science\_team

emp\_record\_table

Columns

EMP\_ID

FIRST\_NAME

LAST\_NAME

GENDER

ROLE

DEPT

EXP

COUNTRY

CONTINENT

SALARY

EMP\_RATING

MANAGER\_ID

PROJ\_ID

Indexes

Foreign Keys

Triggers

Administration Schemas

Information

Column: MANAGER\_ID

Collation: utf8mb4\_0900\_ai\_ci

Definition:

MANAGER\_ID varchar(10)

emp\_records SQL File 10" x emp\_record\_table

Limit to 1000 rows

57

58 SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, SALARY,

59 (0.05\*SALARY \* EMP\_RATING) AS BONUS, (SALARY + (0.05\*SALARY \* EMP\_RATING)) AS NEW\_SALARY

60 FROM emp\_record\_table

61 ORDER BY BONUS DESC;

62

Result Grid

Filter Rows:

Exports Wrap Cell Contents

EMP_ID	FIRST_NAME	LAST_NAME	SALARY	BONUS	NEW_SALARY
E001	Arthur	Black	16500	4125.00	20625.00
E083	Patrick	Voltz	9500	2375.00	11875.00
E428	Pete	Allen	11000	2200.00	13200.00
E103	Emily	Grove	10500	2100.00	12600.00
E204	Karene	Nowak	7500	1875.00	9375.00
E612	Tracy	Norris	8500	1700.00	10200.00
E052	Dianna	Wilson	5500	1375.00	6875.00
E005	Eric	Hoffman	8500	1275.00	9775.00
E260	Roy	Collins	7000	1050.00	8050.00
E583	Janet	Hale	10000	1000.00	11000.00
E010	William	Butler	9000	900.00	9900.00
E478	David	Smith	4000	800.00	4800.00
E403	Steve	Hoffman	5000	750.00	5750.00
E245	Nian	Zhen	6500	650.00	7150.00
E640	Jenifer	Jhones	2800	560.00	3360.00
E505	Chad	Wilson	5000	500.00	5500.00
E057	Dorothy	Wilson	7700	385.00	8085.00
E532	Claire	Brennan	4300	215.00	4515.00
E620	Katrina	Allen	3000	150.00	3150.00

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

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SQL File 10\*

Limit to 1000 rows

```
161
162 SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT, SALARY, COUNTRY, CONTINENT,
163 AVG(SALARY) OVER (PARTITION BY COUNTRY) AS AVG_SALARY_COUNTRY,
164 AVG(SALARY) OVER (PARTITION BY CONTINENT) AS AVG_SALARY_CONTINENT,
165 COUNT(*) OVER (PARTITION BY COUNTRY) AS DISTRIBUTION_PER_COUNTRY,
166 COUNT(*) OVER (PARTITION BY CONTINENT) AS DISTRIBUTION_PER_CONTINENT
167 FROM emp_record_table;
```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	DEPT	SALARY	COUNTRY	CONTINENT	AVG_SALARY_COUNTRY	AVG_SALARY_CONTINENT	DISTRIBUTION_PER_COUNTRY	DISTRIBUTION_PER_CONTINENT
E245	Nian	Zhen	RETAIL	6500	CHINA	ASIA	6500.0000	6250.0000	1	4
E260	Roy	Collins	RETAIL	7000	INDIA	ASIA	6166.6667	6250.0000	3	4
E612	Tracy	Norris	RETAIL	8500	INDIA	ASIA	6166.6667	6250.0000	3	4
E620	Katrina	Allen	RETAIL	3000	INDIA	ASIA	6166.6667	6250.0000	3	4
E010	William	Butler	AUTOMOTIVE	9000	FRANCE	EUROPE	9000.0000	7950.0000	1	4
E204	Karene	Nowak	AUTOMOTIVE	7500	GERMANY	EUROPE	7600.0000	7950.0000	3	4
E428	Pete	Allen	AUTOMOTIVE	11000	GERMANY	EUROPE	7600.0000	7950.0000	3	4
E532	Claire	Brennan	AUTOMOTIVE	4300	GERMANY	EUROPE	7600.0000	7950.0000	3	4
E052	Dianna	Wilson	HEALTHCARE	5500	CANADA	NORTH AMERICA	7000.0000	8525.0000	3	8
E103	Emily	Grove	FINANCE	10500	CANADA	NORTH AMERICA	7000.0000	8525.0000	3	8
E505	Chad	Wilson	HEALTHCARE	5000	CANADA	NORTH AMERICA	7000.0000	8525.0000	3	8
E001	Arthur	Black	ALL	16500	USA	NORTH AMERICA	9440.0000	8525.0000	5	8
E005	Eric	Hoffman	FINANCE	8500	USA	NORTH AMERICA	9440.0000	8525.0000	5	8
E057	Dorothy	Wilson	HEALTHCARE	7700	USA	NORTH AMERICA	9440.0000	8525.0000	5	8
E083	Patrick	Voltz	HEALTHCARE	9500	USA	NORTH AMERICA	9440.0000	8525.0000	5	8
E403	Steve	Hoffman	FINANCE	5000	USA	NORTH AMERICA	9440.0000	8525.0000	5	8
E478	David	Smith	RETAIL	4000	COLOMBIA	SOUTH AMERICA	5600.0000	5600.0000	3	3
E583	Janet	Hale	RETAIL	10000	COLOMBIA	SOUTH AMERICA	5600.0000	5600.0000	3	3
E640	Jenifer	Jhones	RETAIL	2800	COLOMBIA	SOUTH AMERICA	5600.0000	5600.0000	3	3

Schema: employee