

Project Title: ScienceQtech Employee Performance Mapping

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.

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

Problem Statement and Motivation:

As a DBA (Database Administrator), 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 the extra cost for expenses.

What motivates the company to do this?

This will raise the overall performance of the organization by ensuring that all required employees receive training.

Task and Solutions:

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.

Answer:

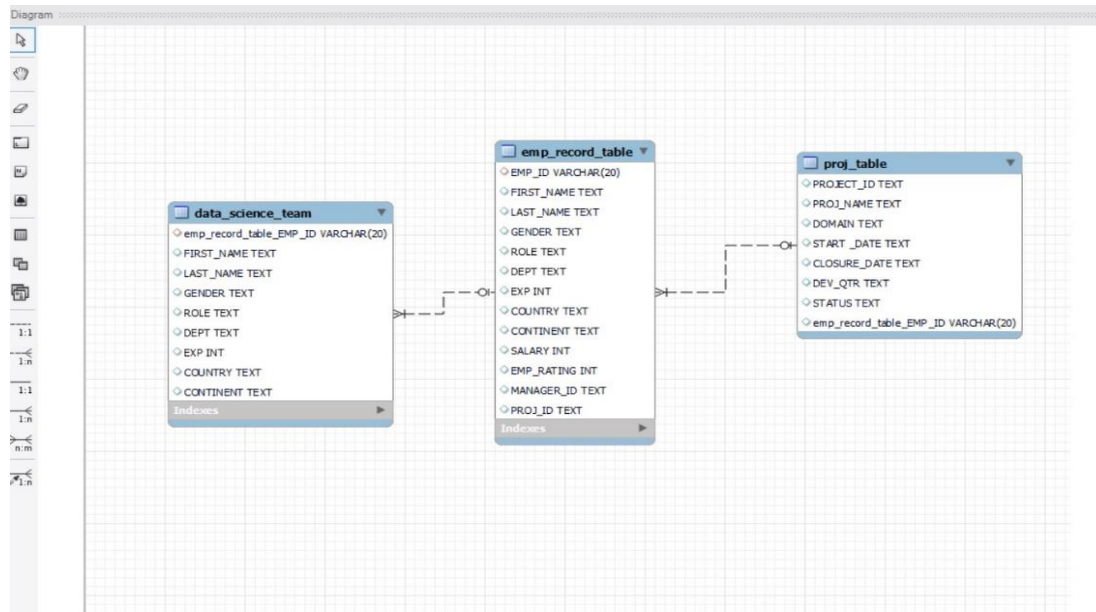
.CREATE DATABASE employee;

The screenshot shows a database management tool interface. The left sidebar displays a tree view with 'default' and 'employee' schemas. The 'employee' schema is selected, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The main window shows a SQL editor with the command 'create database employee;'. The right sidebar contains a context help panel. The bottom output window displays the execution results of the SQL commands.

#	Time	Action	Message	Duration / Fetch
19	00:52:01	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0.000 sec
20	00:52:01	SHOW TABLES FROM 'employee' like 'proj_table'	OK	0.000 sec
21	00:52:03	CREATE TABLE 'employee'.'proj_table' ('PROJECT_ID' text, 'PROJ_NAME' text, 'DOMAIN' text, 'START_DA...	OK	0.000 sec
22	00:52:03	PREPARE stmt FROM 'INSERT INTO 'employee'.'proj_table' ('PROJECT_ID','PROJ_NAME','DOMAIN','STA...	OK	0.000 sec
23	00:52:03	DEALLOCATE PREPARE stmt	OK	0.000 sec

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

Answer:



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.

Answer:

```

SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
FROM emp_record_table;
  
```

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the query: `SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT FROM emp_record_table;`. The Results Grid shows the following data:

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

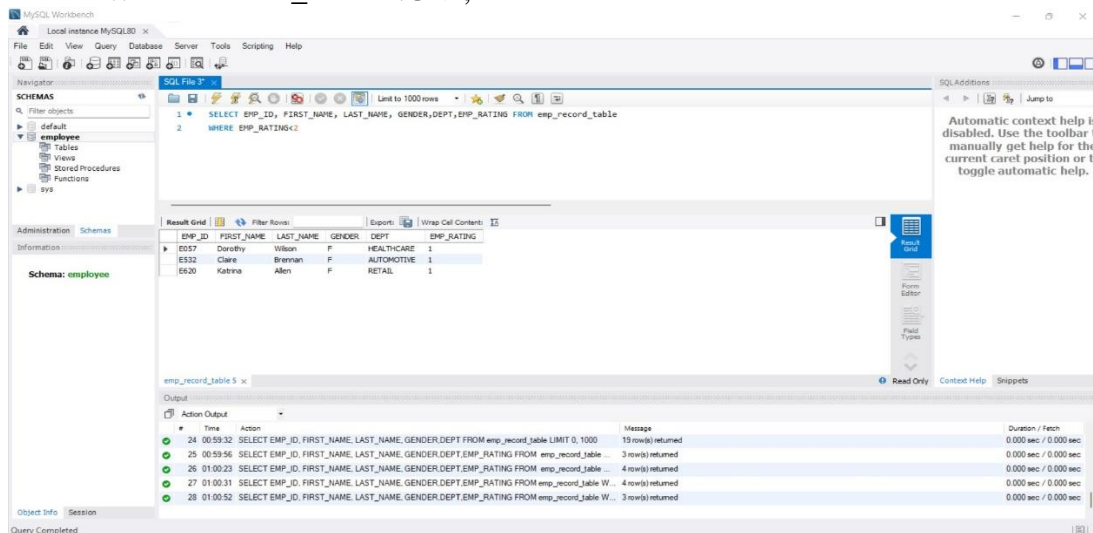
The Action Output pane shows the execution log, including the query execution and the message: "19 row(s) returned".

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

Answer:

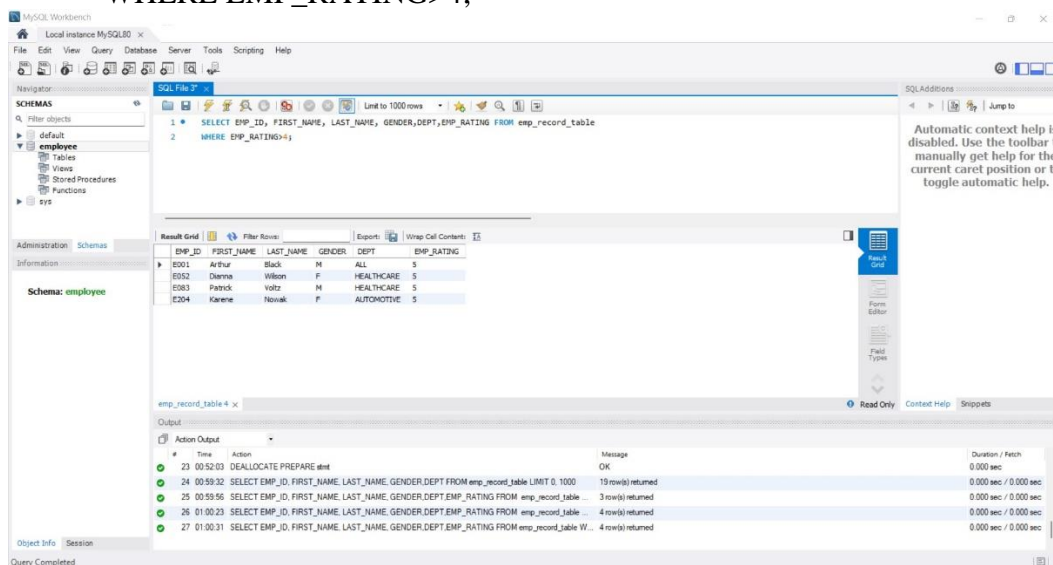
- less than two

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



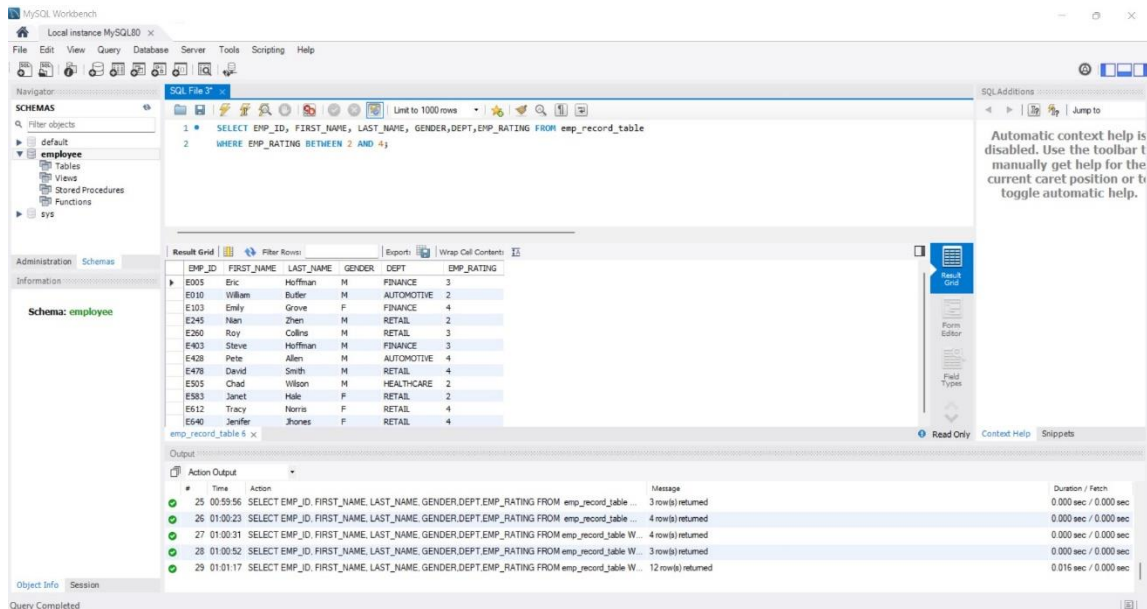
- greater than four

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



- between two and four

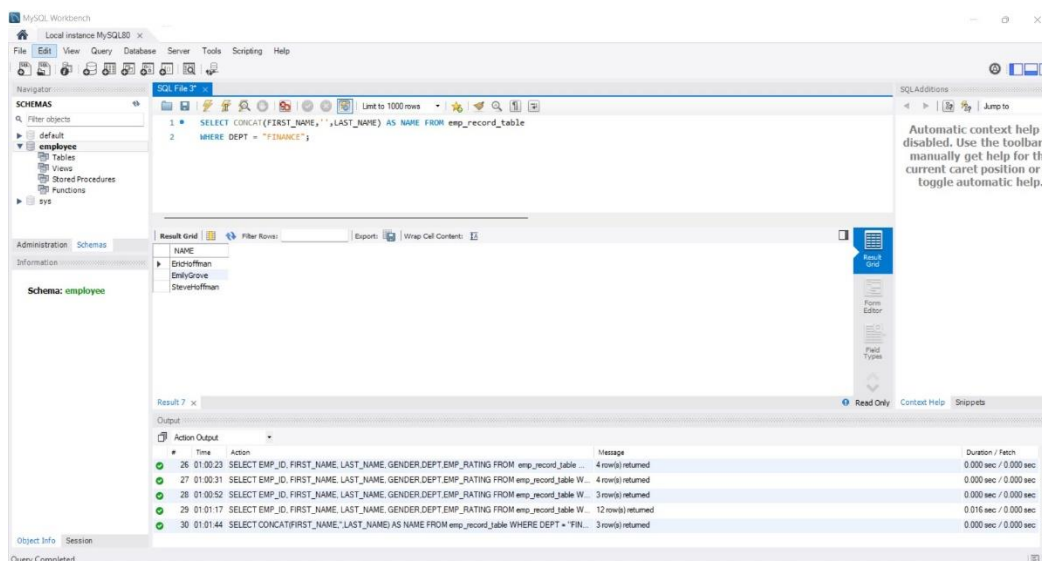
```
SELECT EMP_ID, FIRST_NAME, LAST_NAME,
GENDER,DEPT,EMP_RATING FROM emp_record_table
WHERE EMP_RATING BETWEEN 2 AND 4;
```



- 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.

Answer:

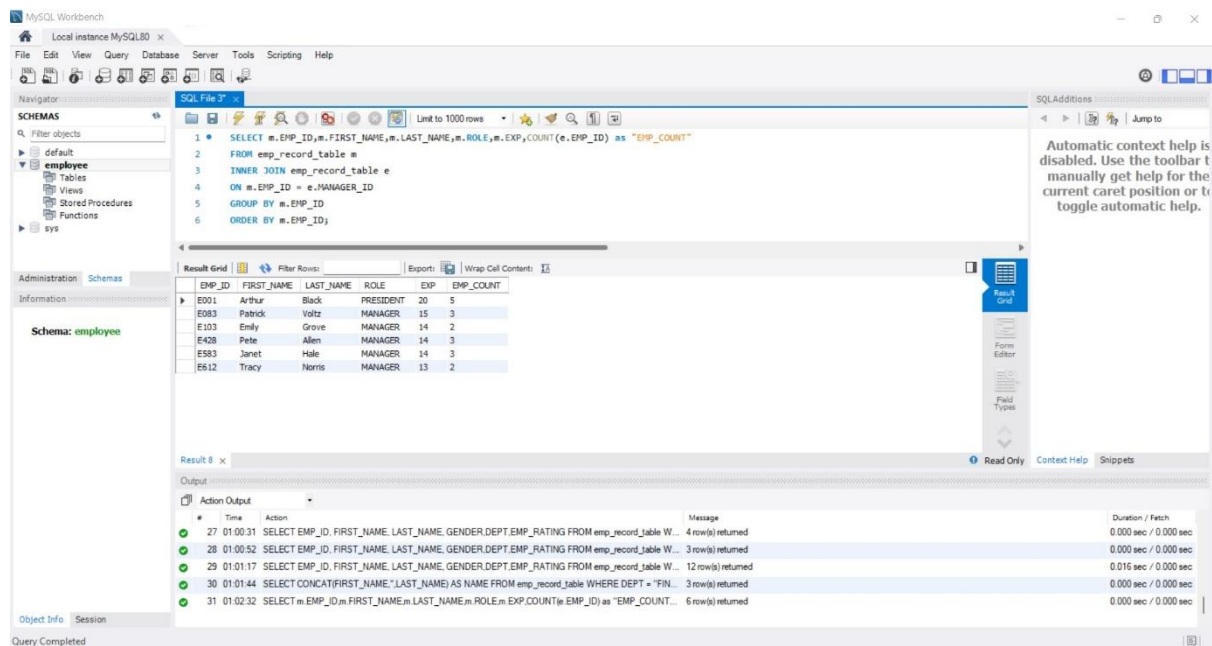
```
SELECT CONCAT(FIRST_NAME,"",LAST_NAME) AS NAME FROM
emp_record_table
WHERE DEPT = "FINANCE";
```



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

Answer:

```
SELECT m.EMP_ID,m.FIRST_NAME,m.LAST_NAME,m.ROLE,  
  
m.EXP,COUNT(e.EMP_ID) as "EMP_COUNT"  
  
FROM emp_record_table m  
  
INNER JOIN emp_record_table e  
  
ON m.EMP_ID = e.MANAGER_ID  
  
GROUP BY m.EMP_ID  
  
ORDER BY m.EMP_ID;
```



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.

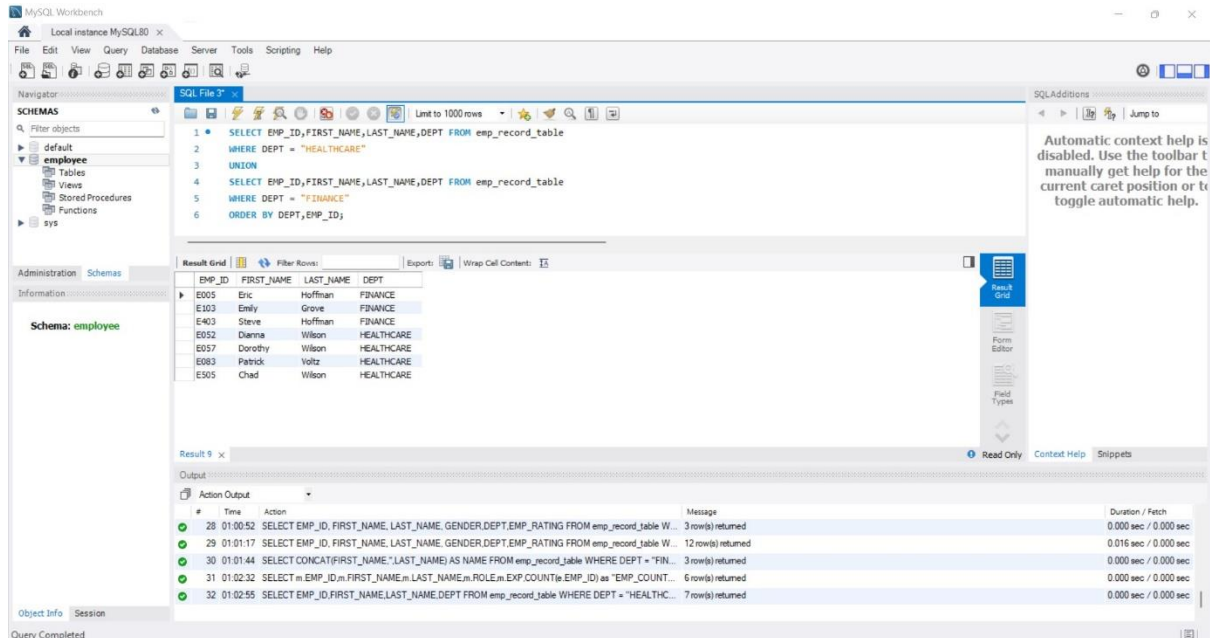
Answer:

```
SELECT EMP_ID,FIRST_NAME,LAST_NAME,DEPT FROM emp_record_table  
  
WHERE DEPT = "HEALTHCARE"  
  
UNION
```

```
SELECT EMP_ID,FIRST_NAME,LAST_NAME,DEPT FROM emp_record_table

WHERE DEPT = "FINANCE"

ORDER BY DEPT,EMP_ID;
```



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.

Answer:

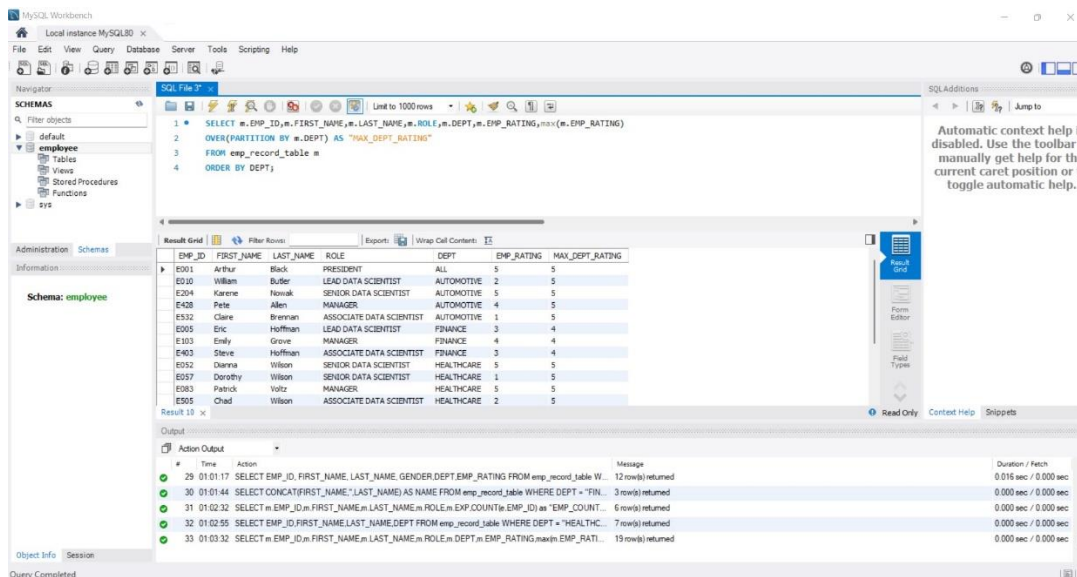
```
SELECT
m.EMP_ID,m.FIRST_NAME,m.LAST_NAME,m.ROLE,m.DEPT,m.EMP_RATING
,max(m.EMP_RATING)

OVER(PARTITION BY m.DEPT)

AS "MAX_DEPT_RATING"

FROM emp_record_table m

ORDER BY DEPT;
```

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

Answer:

```

SELECT EMP_ID, FIRST_NAME, LAST_NAME, ROLE, MAX(SALARY),
MIN(SALARY)

```

```

FROM emp_record_table

```

```

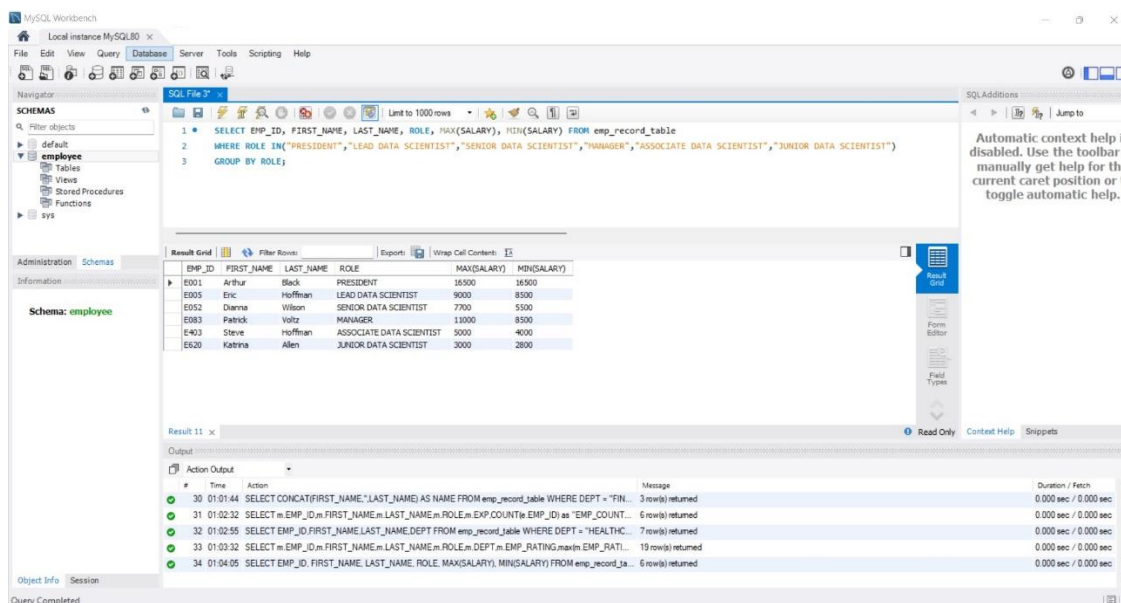
WHERE ROLE IN("PRESIDENT","LEAD DATA SCIENTIST","SENIOR DATA
SCIENTIST","MANAGER","ASSOCIATE DATA SCIENTIST","JUNIOR DATA
SCIENTIST")

```

```

GROUP BY ROLE;

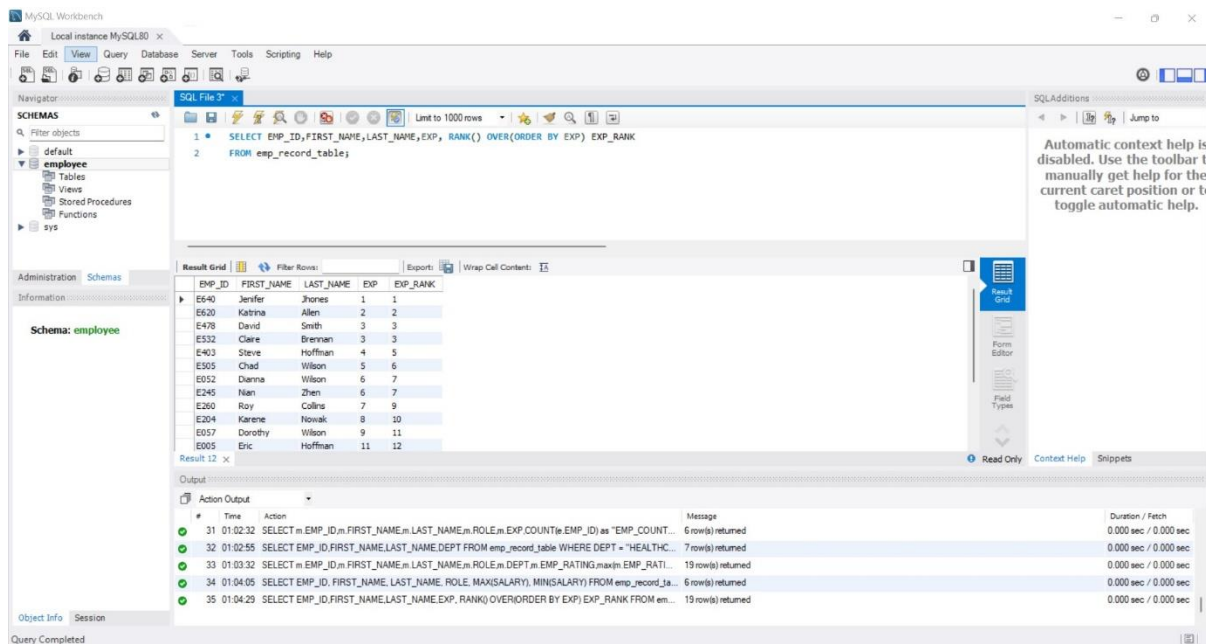
```



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

Answer:

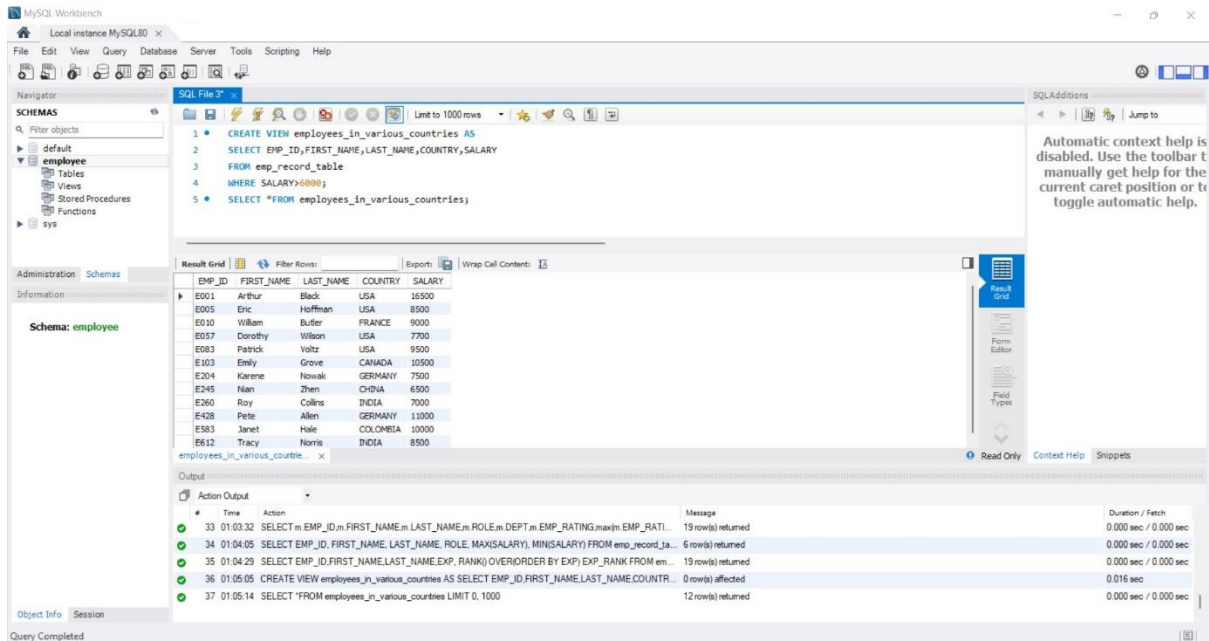
```
SELECT EMP_ID,FIRST_NAME,LAST_NAME,EXP,  
  
RANK() OVER(ORDER BY EXP) EXP_RANK  
  
FROM emp_record_table;
```



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.

Answer:

```
CREATE VIEW employees_in_various_countries AS  
  
SELECT EMP_ID,FIRST_NAME,LAST_NAME,COUNTRY,SALARY  
  
FROM emp_record_table  
  
WHERE SALARY>6000;  
  
SELECT *FROM employees_in_various_countries;
```

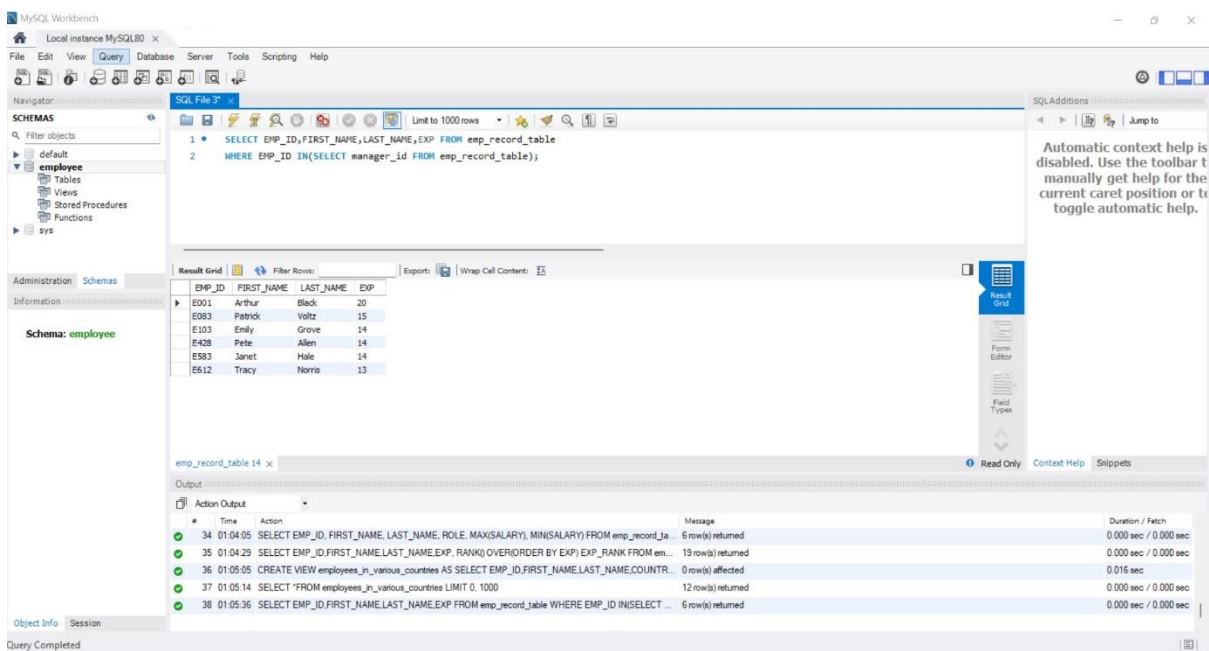
12. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

Answer:

```

SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP FROM emp_record_table
WHERE EMP_ID IN (SELECT manager_id FROM emp_record_table);

```



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.

Answer:

DELIMITER &&

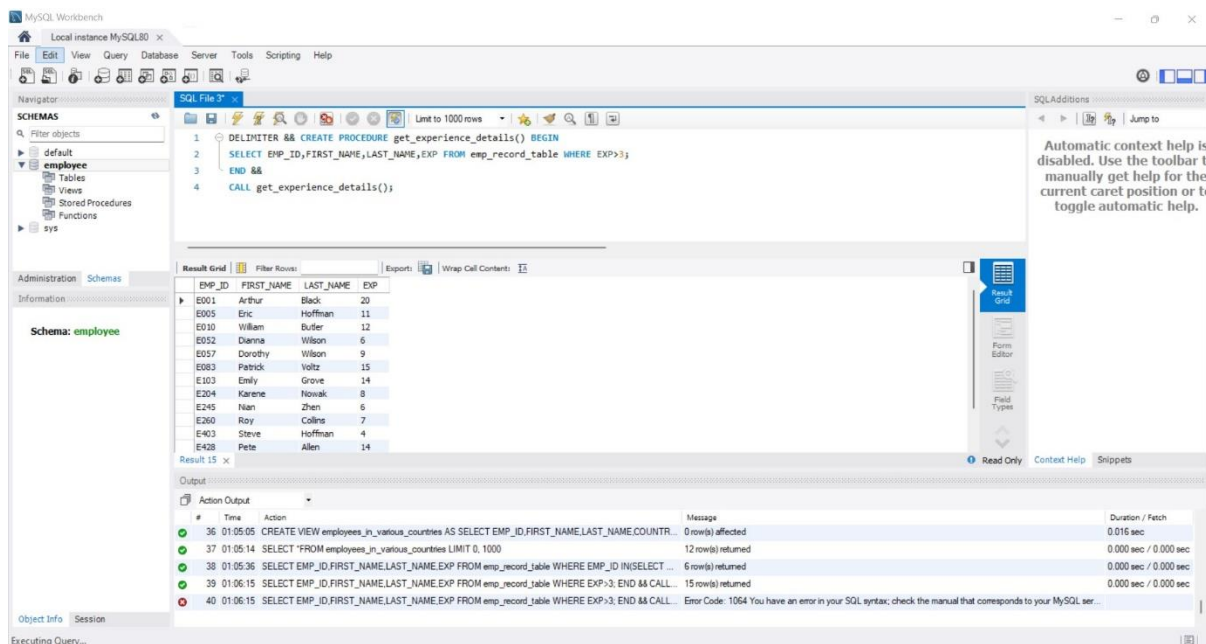
CREATE PROCEDURE get_experience_details()

BEGIN

SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP FROM emp_record_table
WHERE EXP > 3;

END &&

CALL get_experience_details();



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'.

Answer:

DELIMITER &&

CREATE FUNCTION Employee_ROLE(

EXP int

)

RETURNS VARCHAR(40)

DETERMINISTIC

BEGIN

DECLARE Employee_ROLE VARCHAR(40);

IF EXP>12 AND 16 THEN

SET Employee_ROLE="MANAGER";

ELSEIF EXP>10 AND 12 THEN

SET Employee_ROLE="LEAD DATA SCIENTIST";

ELSEIF EXP>5 AND 10 THEN

SET Employee_ROLE="SENIOR DATA SCIENTIST";

ELSEIF EXP>2 AND 5 THEN

SET Employee_ROLE="ASSOCIATE DATA SCIENTIST";

ELSEIF EXP<=2 THEN

SET Employee_ROLE="JUNIOR DATA SCIENTIST";

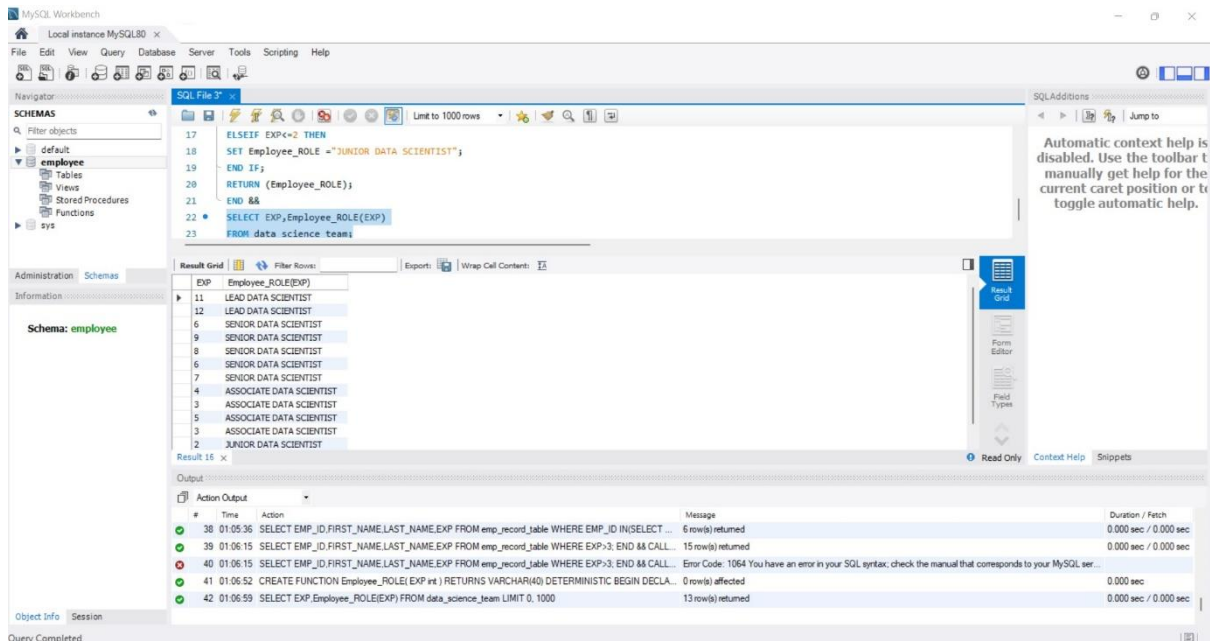
END IF;

RETURN (Employee_ROLE);

END &&

SELECT EXP,Employee_ROLE(EXP)

FROM data_science_team;



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.

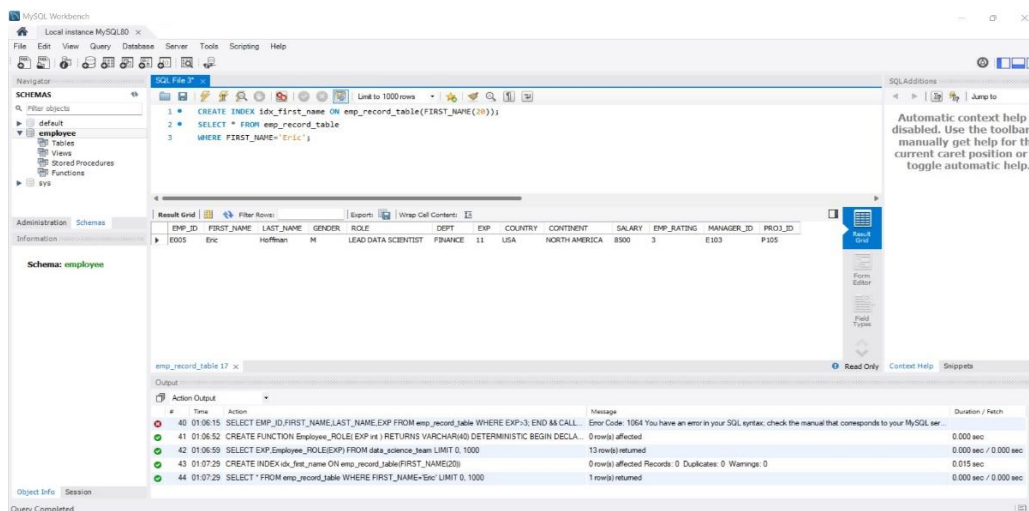
Answer:

CREATE INDEX idx_first_name

ON emp_record_table(FIRST_NAME(20));

SELECT * FROM emp_record_table

WHERE FIRST_NAME='Eric';



toggle automatic help.

Visual Explain Display Info Read + Eval cost Overview View Source

Query cost: 0.35

query_block #1

0.35

1 row

Non-Unique Key Lookup

emp_record_table
idx_first_name

emp_record_table ref
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/Indexes: idx_first_name
Ref.: const
Used Key Parts: FIRST_NAME
Possible Keys: idx_first_name

Attached Conditions
(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: 120

emp_record_table 18 x

Output

Action Output

#	Time	Action
44	01:07:29	SELECT * FROM emp_record_table WHERE FIRST_NAME='Eric' LIMIT 0, 1000
45	01:07:57	CREATE INDEX idx_first_name ON emp_record_table(FIRST_NAME(20))
46	01:08:21	CREATE INDEX idx_first_name ON emp_record_table(FIRST_NAME(20))
47	01:08:29	SELECT * FROM emp_record_table WHERE FIRST_NAME='Eric' LIMIT 0, 1000
48	01:08:42	EXPLAIN SELECT * FROM emp_record_table WHERE FIRST_NAME='Eric'

Duration / Fetch
0.000 sec / 0.000 sec
0.000 sec
0.000 sec
0.000 sec / 0.000 sec
0.000 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).

Answer:

update emp_record_table set salary=(select salary +(select salary*.05*EMP_RATING))

SELECT *FROM emp_record_table;

MySQL 8.0.17

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

default

employee

Tables

Views

Stored Procedures

Functions

sys

Administration Schemas

Information

Schema: employee

SQL File 3*

Limit to 1000 rows

1 • update emp_record_table set salary=(select salary +(select salary*.05*EMP_RATING))

2 • SELECT *FROM emp_record_table

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_RATING	MANAGER_ID	PROJ_ID
E001	Arthur	Bock	M	PRESIDENT	ALL	20	USA	NORTH AMERICA	40500	5	E000	
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	Dianna	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	
E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	10500	4	E001	
E204	Karen	Nowak	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE	7500	5	E428	P204
E245	Nan	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	

emp_record_table 19 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
47	01:08:29	SELECT * FROM emp_record_table WHERE FIRST_NAME='Eric' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
48	01:08:42	EXPLAIN SELECT * FROM emp_record_table WHERE FIRST_NAME='Eric'	OK	0.000 sec
49	01:08:42	EXPLAIN FORMAT=JSON SELECT * FROM emp_record_table WHERE FIRST_NAME='Eric'	OK	0.000 sec
50	01:10:00	update emp_record_table set salary=(select salary +(select salary*.05*EMP_RATING)) SELECT *FROM emp_record_table	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'SELECT *FROM emp_record_table' at line 1	0.000 sec
51	01:10:24	SELECT *FROM emp_record_table LIMIT 0, 1000	19 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Read Only Context Help Snippets

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

Answer:

```
SELECT EMP_ID,FIRST_NAME,LAST_NAME,SALARY,COUNTRY,CONTINENT,
AVG(salary)OVER(PARTITION BY COUNTRY)AVG_salary_IN_COUNTRY,
AVG(salary)OVER(PARTITION BY CONTINENT)AVG_salary_IN_CONTINENT,
COUNT(*)OVER(PARTITION BY COUNTRY)COUNT_IN_COUNTRY,
COUNT(*)OVER(PARTITION BY CONTINENT)COUNT_IN_CONTINENT
FROM emp_record_table;
```

The screenshot displays the MySQL Workbench interface. The SQL Editor at the top contains a query that calculates average salaries by country and continent, along with counts. The query is as follows:

```
1 SELECT EMP_ID,FIRST_NAME,LAST_NAME,SALARY,COUNTRY,CONTINENT,
2 AVG(salary)OVER(PARTITION BY COUNTRY)AVG_salary_IN_COUNTRY,
3 AVG(salary)OVER(PARTITION BY CONTINENT)AVG_salary_IN_CONTINENT,
4 COUNT(*)OVER(PARTITION BY COUNTRY)COUNT_IN_COUNTRY,
5 COUNT(*)OVER(PARTITION BY CONTINENT)COUNT_IN_CONTINENT
6 FROM emp_record_table;
```

Below the editor, the 'Result Grid' shows the output of the query. The columns are: EMP_ID, FIRST_NAME, LAST_NAME, SALARY, COUNTRY, CONTINENT, AVG_salary_IN_COUNTRY, AVG_salary_IN_CONTINENT, COUNT_IN_COUNTRY, and COUNT_IN_CONTINENT. The results are as follows:

EMP_ID	FIRST_NAME	LAST_NAME	SALARY	COUNTRY	CONTINENT	AVG_salary_IN_COUNTRY	AVG_salary_IN_CONTINENT	COUNT_IN_COUNTRY	COUNT_IN_CONTINENT
E245	Nian	Zhen	6500	CHINA	ASIA	6500.0000	6250.0000	1	4
E260	Roy	Collins	7000	INDIA	ASIA	6166.6667	6250.0000	3	4
E612	Tracy	Norris	8500	INDIA	ASIA	6166.6667	6250.0000	3	4
E620	Katrina	Allen	3000	INDIA	ASIA	6166.6667	6250.0000	3	4
E010	William	Burter	9000	FRANCE	EUROPE	9000.0000	7950.0000	1	4
E204	Karen	Nemak	7500	GERMANY	EUROPE	7600.0000	7950.0000	3	4
E428	Pete	Allen	11000	GERMANY	EUROPE	7600.0000	7950.0000	3	4
E532	Clare	Brennan	4300	GERMANY	EUROPE	7600.0000	7950.0000	3	4
E052	Dianna	Wilson	5500	CANADA	NORTH AMERICA	7000.0000	8525.0000	3	8
E103	Emily	Grove	10500	CANADA	NORTH AMERICA	7000.0000	8525.0000	3	8
E505	Chad	Wilson	5000	CANADA	NORTH AMERICA	7000.0000	8525.0000	3	8
E001	Arthur	Black	16500	USA	NORTH AMERICA	9440.0000	8525.0000	5	8

The 'Output' pane at the bottom shows the execution log, including the query execution time and the number of rows returned (19 rows).