

Science Qtech Employee Performance Mapping SQL Project.

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 **employee**;

Import

data_science_team.csv

emp_record_table.csv

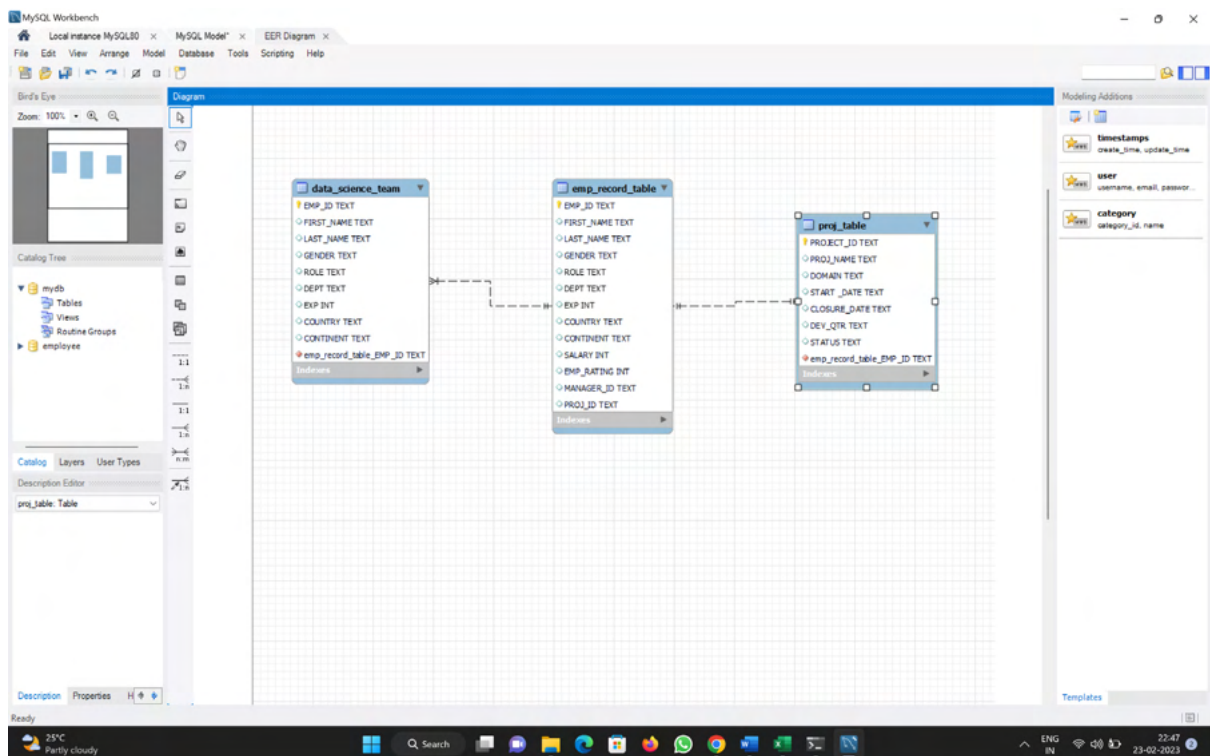
proj_table.csv into the **employee** database from the given resources.

>>>Right click on Table Tab >>click Table Data Import Wizard>>Select File Path>>Import Data.

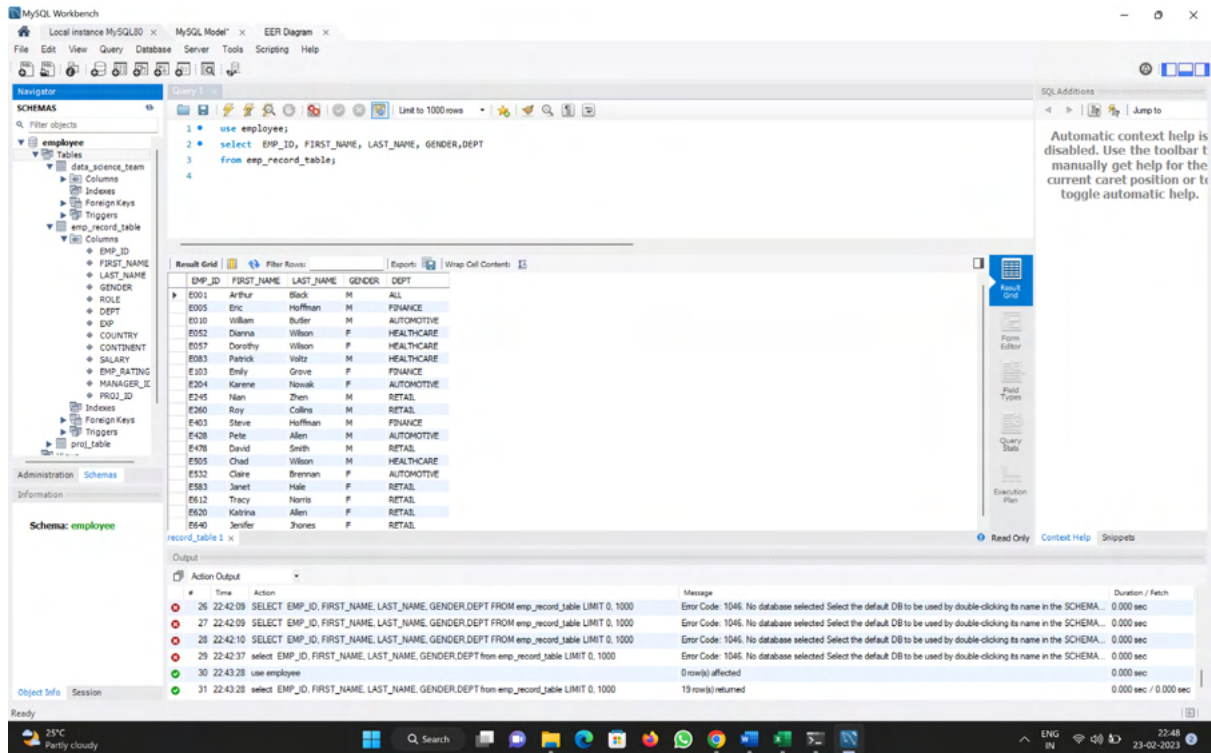
Then we have to refresh the Schema and use command

Use **employee**;

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.



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
 - less than two

MySQL Workbench

Local instance MySQL80 x MySQL Model x EER Diagram x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

employee

Tables

data_science_team

Columns

Indexes

ForeignKeys

Triggers

emp_record_table

Columns

EMP_ID

FIRST_NAME

LAST_NAME

GENDER

ROLE

DEPT

EMP

COUNTRY

CONTINENT

SALARY

EMP_RATING

MANAGER_ID

PROJ_ID

Indexes

ForeignKeys

Triggers

proj_table

Administration Schemas

Information

Schema: employee

Query 1

```

1 select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
2 where EMP_RATING > 4
3

```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT	EMP_RATING
E057	Dorothy	Wilson	F	HEALTHCARE	1
E532	Clare	Brennan	F	AUTOMOTIVE	1
E620	Katrina	Allen	F	RETAIL	1

emp_record_table 2 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
27	22:42:09	SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT FROM emp_record_table LIMIT 0, 1000	Error Code: 1046. No database selected. Select the default DB to be used by double-clicking its name in the SCHEMA...	0.000 sec
28	22:42:10	SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT FROM emp_record_table LIMIT 0, 1000	Error Code: 1046. No database selected. Select the default DB to be used by double-clicking its name in the SCHEMA...	0.000 sec
29	22:42:37	select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT from emp_record_table LIMIT 0, 1000	Error Code: 1046. No database selected. Select the default DB to be used by double-clicking its name in the SCHEMA...	0.000 sec
30	22:43:28	use employee	0 row(s) affected	0.000 sec
31	22:43:28	select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT from emp_record_table LIMIT 0, 1000	19 row(s) returned	0.000 sec / 0.000 sec
32	22:50:45	select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record_table where EMP_...	3 row(s) returned	0.000 sec / 0.000 sec

Query Completed

25°C Partly cloudy

- greater than four

MySQL Workbench

Local instance MySQL80 x MySQL Model x EER Diagram x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

employee

Tables

data_science_team

Columns

Indexes

ForeignKeys

Triggers

emp_record_table

Columns

EMP_ID

FIRST_NAME

LAST_NAME

GENDER

ROLE

DEPT

EMP

COUNTRY

CONTINENT

SALARY

EMP_RATING

MANAGER_ID

PROJ_ID

Indexes

ForeignKeys

Triggers

proj_table

Administration Schemas

Information

Schema: employee

Query 1

```

1 select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
2 where EMP_RATING > 4
3

```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT	EMP_RATING
E001	Arthur	Black	M	ALL	5
E052	Dianne	Wilson	F	HEALTHCARE	5
E083	Patrick	Volz	M	HEALTHCARE	5
E204	Karen	Nowak	F	AUTOMOTIVE	5

emp_record_table 3 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
28	22:42:10	SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT FROM emp_record_table LIMIT 0, 1000	Error Code: 1046. No database selected. Select the default DB to be used by double-clicking its name in the SCHEMA...	0.000 sec
29	22:42:37	select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT from emp_record_table LIMIT 0, 1000	Error Code: 1046. No database selected. Select the default DB to be used by double-clicking its name in the SCHEMA...	0.000 sec
30	22:43:28	use employee	0 row(s) affected	0.000 sec
31	22:43:28	select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT from emp_record_table LIMIT 0, 1000	19 row(s) returned	0.000 sec / 0.000 sec
32	22:50:45	select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record_table where EMP_...	3 row(s) returned	0.000 sec / 0.000 sec
33	22:51:43	select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record_table where EMP_...	4 row(s) returned	0.000 sec / 0.000 sec

Query Completed

25°C Partly cloudy

- between two and four

MySQL Workbench interface showing a query execution. The query is:

```
1 select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record_table
2 where EMP_RATING between 2 and 4
3
```

The result grid displays the following data:

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	Nan	Zhen	M	RETAIL	2
E260	Roy	Collins	M	RETAIL	3
E463	Steve	Hoffman	M	FINANCE	3
E428	Pete	Allen	M	AUTOMOTIVE	4
E478	David	Smith	M	RETAIL	4
E505	Chad	Wilson	M	HEALTHCARE	2
E583	Janet	Hale	F	RETAIL	2
E612	Tracy	Norris	F	RETAIL	4
E640	Jennifer	Jones	F	RETAIL	4

The Action Output pane shows the execution steps and messages:

#	Time	Action	Message	Duration / Fetch
29	22:42:37	select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT from emp_record_table LIMIT 0, 1000	Error Code: 1046. No database selected. Select the default DB to be used by double-clicking its name in the SCHEMA.	0.000 sec
30	22:43:28	use employee	0 row(s) affected	0.000 sec
31	22:43:28	select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT from emp_record_table LIMIT 0, 1000	19 row(s) returned	0.000 sec / 0.000 sec
32	22:50:45	select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record_table where EMP...	3 row(s) returned	0.000 sec / 0.000 sec
33	22:51:43	select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record_table where EMP...	4 row(s) returned	0.000 sec / 0.000 sec
34	22:52:54	select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record_table where EMP...	12 row(s) returned	0.000 sec / 0.000 sec

- 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.

MySQL Workbench interface showing a query execution. The query is:

```
1 select DEPT, CONCAT(FIRST_NAME, ' ', LAST_NAME) as
2 name from emp_record_table
3 where DEPT = 'FINANCE'
4
5
```

The result grid displays the following data:

DEPT	name
FINANCE	EricHoffman
FINANCE	EmilyGrove
FINANCE	SteveHoffman

The Action Output pane shows the execution steps and messages:

#	Time	Action	Message	Duration / Fetch
30	22:43:28	use employee	0 row(s) affected	0.000 sec
31	22:43:28	select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT from emp_record_table LIMIT 0, 1000	19 row(s) returned	0.000 sec / 0.000 sec
32	22:50:45	select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record_table where EMP...	3 row(s) returned	0.000 sec / 0.000 sec
33	22:51:43	select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record_table where EMP...	4 row(s) returned	0.000 sec / 0.000 sec
34	22:52:54	select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record_table where EMP...	12 row(s) returned	0.000 sec / 0.000 sec
35	22:55:21	select DEPT, CONCAT(FIRST_NAME, ' ', LAST_NAME) as name from emp_record_table where DEPT = 'FINANCE' ...	3 row(s) returned	0.000 sec / 0.000 sec

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

MySQL Workbench interface showing a query to join employee and manager tables. The query is:

```

1 select employee.EMP_ID, concat(employee.FIRST_NAME, ' ', employee.LAST_NAME)
2 as Employee_Name, manager.MANAGER_ID, concat(manager.FIRST_NAME, ' ', manager.LAST_NAME)
3 as Manager_Name, manager.ROLE as ROLE
4 from emp_record_table employee
5 join emp_record_table manager on employee.MANAGER_ID = manager.EMP_ID
6

```

The result grid displays the following data:

EMP_ID	Employee_Name	MANAGER_ID	Manager_Name	ROLE
E612	Tracy Norris		Arthur Black	PRESIDENT
E583	Janet Hale		Arthur Black	PRESIDENT
E428	Pete Allen		Arthur Black	PRESIDENT
E103	Emily Grove		Arthur Black	PRESIDENT
E083	Patrick Voltz		Arthur Black	PRESIDENT
E505	Chad Wilson	E001	Patrick Voltz	MANAGER
E057	Dorothy Wilson	E001	Patrick Voltz	MANAGER
E052	Dianna Wilson	E001	Patrick Voltz	MANAGER
E403	Steve Hoffman	E001	Emily Grove	MANAGER
E005	Eric Hoffman	E001	Emily Grove	MANAGER
E532	Claire Brennan	E001	Pete Allen	MANAGER
E204	Karen Novak	E001	Pete Allen	MANAGER
E010	William Butler	E001	Pete Allen	MANAGER
E478	David Smith	E001	Janet Hale	MANAGER
E260	Roy Collins	E001	Janet Hale	MANAGER
E245	Nan Zhen	E001	Janet Hale	MANAGER
E640	Jennifer Jones	E001	Tracy Norris	MANAGER
E620	Katrina Allen	E001	Tracy Norris	MANAGER

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

MySQL Workbench interface showing a query using UNION to combine employees from healthcare and finance departments. The query is:

```

1 select EMP_ID, FIRST_NAME, LAST_NAME, DEPT from emp_record_table
2 where DEPT = "HEALTHCARE"
3 union
4 select EMP_ID, FIRST_NAME, LAST_NAME, DEPT from emp_record_table
5 where DEPT = "FINANCE"
6 order by DEPT, EMP_ID
7

```

The result grid displays the following data:

EMP_ID	FIRST_NAME	LAST_NAME	DEPT
E005	Eric	Hoffman	FINANCE
E103	Emily	Grove	FINANCE
E403	Steve	Hoffman	FINANCE
E052	Dianna	Wilson	HEALTHCARE
E057	Dorothy	Wilson	HEALTHCARE
E083	Patrick	Voltz	HEALTHCARE
E505	Chad	Wilson	HEALTHCARE

- 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.

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

employee

Tables

data_science_team

Columns

Indexes

ForeignKeys

Triggers

emp_record_table

Columns

EMP_ID

FIRST_NAME

LAST_NAME

DEPT

EMP_RATING

MAX_DEPT_RATING

Information

Schema: employee

Query 1

```

1 select m.EMP_ID,m.FIRST_NAME,m.LAST_NAME,m.ROLE,m.DEPT,m.EMP_RATING,max(m.EMP_RATING)
2 over(partition by m.DEPT)
3 as "MAX_DEPT_RATING"
4 from emp_record_table m
5 order by DEPT;

```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EMP_RATING	MAX_DEPT_RATING
E001	Arthur	Black	PRESIDENT	ALL	5	5
E010	William	Butler	LEAD DATA SCIENTIST	AUTOMOTIVE	2	5
E204	Karen	Novak	SENIOR DATA SCIENTIST	AUTOMOTIVE	5	5
E408	Pete	Allen	MANAGER	AUTOMOTIVE	4	5
E532	Claire	Brennan	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	1	5
E005	Eric	Huffman	LEAD DATA SCIENTIST	FINANCE	3	4
E103	Emily	Grove	MANAGER	FINANCE	4	4
E403	Steve	Huffman	ASSOCIATE DATA SCIENTIST	FINANCE	3	4
E052	Danna	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	Nan	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	Jennifer	Jones	JUNIOR DATA SCIENTIST	RETAIL	4	4

Output

Action Output

Time Action Message Duration / Fetch

46 23:22:04 select employee EMP_ID, concat(employee FIRST_NAME, "employee LAST_NAME) as Employee_Name, manager... Error Code: 1054. Unknown column 'employee Manager_Name' in field list 0.000 sec

47 23:24:38 select employee EMP_ID, concat(employee FIRST_NAME, "employee LAST_NAME) as Employee_Name, manager... 10 row(s) returned 0.000 sec / 0.000 sec

48 23:31:01 select e EMP_ID, e FIRST_NAME, e LAST_NAME, e ROLE, count(e EMP_ID) as Reporters from emp_record_table e... Error Code: 1055. Expression #2 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'em... 0.000 sec

49 23:31:45 select e EMP_ID, e FIRST_NAME, e LAST_NAME, e ROLE, count(e EMP_ID) as Reporters from emp_record_table e... Error Code: 1055. Expression #2 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'em... 0.000 sec

50 23:37:19 select EMP_ID, FIRST_NAME, LAST_NAME, DEPT from emp_record_table where DEPT = "HEALTHCARE" union se... 7 row(s) returned 0.000 sec / 0.000 sec

51 23:40:07 select m EMP_ID, m FIRST_NAME, m LAST_NAME, m ROLE, m DEPT, m EMP_RATING, max(m EMP_RATING) over... 19 row(s) returned 0.000 sec / 0.000 sec

Query Completed

25°C Party cloudy

23:40 23-02-2023

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.

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

employee

Tables

data_science_team

Columns

Indexes

ForeignKeys

Triggers

emp_record_table

Columns

EMP_ID

FIRST_NAME

LAST_NAME

DEPT

EMP_RATING

MAXIMUM_SALARY

Information

Schema: employee

Query 1

```

1 select EMP_ID, FIRST_NAME, LAST_NAME, ROLE, SALARY, min(SALARY) over (partition by ROLE) as Minimum_Salary,
2 max(SALARY) over (partition by ROLE) as Maximum_Salary
3 from emp_record_table
4 order by SALARY;

```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	SALARY	Minimum_Salary	Maximum_Salary
E640	Jennifer	Jones	JUNIOR DATA SCIENTIST	2800	2800	3000
E620	Katrina	Allen	JUNIOR DATA SCIENTIST	3000	2800	3000
E478	David	Smith	ASSOCIATE DATA SCIENTIST	4000	4000	5000
E532	Claire	Brennan	ASSOCIATE DATA SCIENTIST	4300	4000	5000
E403	Steve	Huffman	ASSOCIATE DATA SCIENTIST	5000	4000	5000
E005	Eric	Huffman	LEAD DATA SCIENTIST	8500	8500	9000
E052	Danna	Wilson	SENIOR DATA SCIENTIST	5500	5500	7700
E245	Nan	Zhen	SENIOR DATA SCIENTIST	6500	5500	7700
E260	Roy	Collins	SENIOR DATA SCIENTIST	7000	5500	7700
E204	Karen	Novak	SENIOR DATA SCIENTIST	7500	5500	7700
E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	7700	5500	7700
E612	Tracy	Norris	MANAGER	8500	8500	11000
E010	William	Butler	LEAD DATA SCIENTIST	9000	8500	9000
E083	Patrick	Voltz	MANAGER	9500	8500	11000
E583	Janet	Hale	MANAGER	10000	8500	11000
E103	Emily	Grove	MANAGER	10500	8500	11000
E408	Pete	Allen	MANAGER	11000	8500	11000
E001	Arthur	Black	PRESIDENT	16500	16500	16500

Output

Action Output

Time Action Message Duration / Fetch

49 23:31:45 select e EMP_ID, e FIRST_NAME, e LAST_NAME, e ROLE, count(e EMP_ID) as Reporters from emp_record_table e... Error Code: 1055. Expression #2 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'em... 0.000 sec

50 23:37:19 select EMP_ID, FIRST_NAME, LAST_NAME, DEPT from emp_record_table where DEPT = "HEALTHCARE" union se... 7 row(s) returned 0.000 sec / 0.000 sec

51 23:40:07 select m EMP_ID, m FIRST_NAME, m LAST_NAME, m ROLE, m DEPT, m EMP_RATING, max(m EMP_RATING) over... 19 row(s) returned 0.000 sec / 0.000 sec

52 23:44:22 select EMP_ID, FIRST_NAME, LAST_NAME, ROLE, MAX(SALARY), MIN(SALARY) from emp_record_table where ... Error Code: 1055. Expression #1 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'em... 0.000 sec

53 23:44:39 select EMP_ID, FIRST_NAME, LAST_NAME, ROLE, MAX(SALARY), MIN(SALARY) from emp_record_table where ... Error Code: 1055. Expression #1 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'em... 0.000 sec

54 23:48:32 select EMP_ID, FIRST_NAME, LAST_NAME, ROLE, SALARY, min(SALARY) over (partition by ROLE) as Minimum... 19 row(s) returned 0.000 sec / 0.000 sec

Query Completed

25°C Party cloudy

23:49 23-02-2023

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 query editor contains the following SQL code:

```
1 select EMP_ID, FIRST_NAME, LAST_NAME, EXP,
2 rank() over(order by EXP) EXP_RANK
3 from emp_record_table;
4
```

The Results grid displays the following data:

EMP_ID	FIRST_NAME	LAST_NAME	EXP	EXP_RANK
E640	Jennifer	Phones	1	1
E620	Katrina	Allen	2	2
E478	David	Smith	3	3
E532	Claire	Brennan	3	3
E463	Steve	Huffman	4	5
E505	Chad	Wilson	5	6
E052	Danna	Wilson	6	7
E245	Nan	Zhen	6	7
E260	Roy	Collins	7	9
E264	Karen	Nowak	8	10
E057	Dorothy	Wilson	9	11
E005	Eric	Huffman	11	12
E010	William	Butler	12	13
E612	Tracy	Norris	13	14
E103	Emily	Grove	14	15
E408	Pete	Allen	14	15
E583	Janet	Hale	14	15
E083	Patrick	Voltz	15	18
E001	Arthur	Black	20	19

The Action Output pane shows the execution of the query, indicating that 19 rows were returned.

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 query editor contains the following SQL code:

```
1 drop view if exists employee_salary;
2 create view employee_salary as
3 select EMP_ID, FIRST_NAME, LAST_NAME, ROLE, COUNTRY, SALARY from emp_record_table
4 where SALARY > 6000;
5 select * from employee_salary
6 order by SALARY;
```

The Results grid displays the following data:

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	COUNTRY	SALARY
E245	Nan	Zhen	SENIOR DATA SCIENTIST	CHINA	6500
E260	Roy	Collins	SENIOR DATA SCIENTIST	INDIA	7000
E264	Karen	Nowak	SENIOR DATA SCIENTIST	GERMANY	7500
E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	USA	7700
E005	Eric	Huffman	LEAD DATA SCIENTIST	USA	8500
E612	Tracy	Norris	MANAGER	INDIA	8500
E010	William	Butler	LEAD DATA SCIENTIST	FRANCE	9000
E083	Patrick	Voltz	MANAGER	USA	9500
E583	Janet	Hale	MANAGER	COLOMBIA	10000
E103	Emily	Grove	MANAGER	CANADA	10500
E408	Pete	Allen	MANAGER	GERMANY	11000
E001	Arthur	Black	PRESIDENT	USA	16500

The Action Output pane shows the execution of the query, indicating that 12 rows were returned.

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 query editor contains the following SQL statement:

```
1 select EMP_ID, FIRST_NAME, LAST_NAME, EXP, ROLE from (select * from emp_record_table where EXP > 10) as Exp10;
```

The result grid displays the following data:

EMP_ID	FIRST_NAME	LAST_NAME	EXP	ROLE
E001	Arthur	Black	20	PRESIDENT
E005	Eric	Hoffman	11	LEAD DATA SCIENTIST
E010	William	Butler	12	LEAD DATA SCIENTIST
E083	Patrick	Volzt	15	MANAGER
E103	Emily	Grove	14	MANAGER
E408	Pete	Allen	14	MANAGER
E583	Janet	Hale	14	MANAGER
E612	Tracy	Norris	13	MANAGER

The output pane shows the execution of the query, with a message indicating that the query was successful and returned 8 rows.

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 query editor contains the following SQL statement:

```
1 DELIMITER $$
2 create procedure get_experience_details()
3 begin
4 select EMP_ID, FIRST_NAME, LAST_NAME, EXP from emp_record_table where EXP > 3;
5 end $$
6 call get_experience_details();
```

The result grid displays the following data:

EMP_ID	FIRST_NAME	LAST_NAME	EXP
E001	Arthur	Black	20
E005	Eric	Hoffman	11
E010	William	Butler	12
E052	Dianna	Wilson	6
E057	Dorothy	Wilson	9
E083	Patrick	Volzt	15
E103	Emily	Grove	14
E204	Karen	Novak	8
E245	Nan	Zhen	6
E260	Roy	Collins	7
E403	Steve	Hoffman	4
E408	Pete	Allen	14
E505	Chad	Wilson	5
E583	Janet	Hale	14
E612	Tracy	Norris	13

The output pane shows the execution of the stored procedure, with a message indicating that the procedure was successful and returned 15 rows.

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 'Query' window contains the following SQL code:

```
1 DELIMITER $$
2 create function Employee_ROLE(
3     EXP int
4 )
5     returns varchar(40)
6     deterministic
7     begin
8         declare Employee_ROLE varchar(40);
9         if EXP > 12 and 16 then
10             set Employee_ROLE = "MANAGER";
11         elseif EXP > 10 and 12 then
12             set Employee_ROLE = "LEAD DATA SCIENTIST";
13         elseif EXP > 5 and 10 then
14             set Employee_ROLE = "SENIOR DATA SCIENTIST";
15         elseif EXP > 2 and 5 then
16             set Employee_ROLE = "ASSOCIATE DATA SCIENTIST";
17         elseif EXP <= 2 then
18             set Employee_ROLE = "JUNIOR DATA SCIENTIST";
19         end if;
20         return (Employee_ROLE);
21     end $$
22
23 select EXP, Employee_ROLE(EXP)
24 from data_science_team;
25
```

The 'Output' window shows the execution results of the query. The first four rows show the creation of the function, each with an error message: "Error Code: 1304. FUNCTION Employee_ROLE already exists". The fifth row shows the execution of the query, which returns 13 rows. The last row shows the execution of the function, which returns the same error message.

Time	Action	Message	Duration / Fetch
03 00:25:20	create FUNCTION Employee_ROLE(EXP int) RETURNS VARCHAR(40) DETERMINISTIC BEGIN DECLARE Empl...	Error Code: 1304. FUNCTION Employee_ROLE already exists	0.000 sec
04 00:25:21	create FUNCTION Employee_ROLE(EXP int) RETURNS VARCHAR(40) DETERMINISTIC BEGIN DECLARE Empl...	Error Code: 1304. FUNCTION Employee_ROLE already exists	0.000 sec
05 00:25:25	create FUNCTION Employee_ROLE(EXP int) RETURNS VARCHAR(40) DETERMINISTIC BEGIN DECLARE Empl...	Error Code: 1304. FUNCTION Employee_ROLE already exists	0.000 sec
06 00:25:30	create FUNCTION Employee_ROLE(EXP int) RETURNS VARCHAR(40) DETERMINISTIC BEGIN DECLARE Empl...	Error Code: 1304. FUNCTION Employee_ROLE already exists	0.000 sec
07 00:25:36	SELECT EXP, Employee_ROLE(EXP) FROM data_science_team LIMIT 0, 1000 ;	13 row(s) returned	0.000 sec / 0.000 sec
08 13:52:51	create FUNCTION Employee_ROLE(EXP int) returns varchar(40) deterministic begin declare Employee_ROLE varchar(4...	Error Code: 1304. FUNCTION Employee_ROLE already exists	0.000 sec

MySQL Workbench

Local instance MySQL80 x MySQL Model x EER Diagram x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

Indexes

ForeignKeys

Triggers

emp_record_table

Columns

EMP_ID

FIRST_NAME

LAST_NAME

GENDER

ROLE

DEPT

EMP

COUNTRY

CONTINENT

SALARY

EMP_RATING

MANAGER_ID

PROJ_ID

Indexes

ForeignKeys

Triggers

proj_table

Views

StoredProcedures

Functions

sys

Administration Schemas

Information

Schema: employee

Query 1

end if;

return (Employee_ROLE);

end if;

select EXP, Employee_ROLE(EXP)

from data_science_team;

Result Grid

Filter Rows

Exports

Wrap Cell Contents

Result 21

Output

Action Output

#	Time	Action	Message	Duration / Fetch
82	00:25:00	SELECT EXP, Employee_ROLE(EXP) FROM data_science_team LIMIT 0, 1000;	13 row(s) returned	0.000 sec / 0.000 sec
83	00:25:00	create FUNCTION Employee_ROLE(EXP vt) RETURNS VARCHAR(40) DETERMINISTIC BEGIN DECLARE Emp...	Error Code: 1304. FUNCTION Employee_ROLE already exists	0.000 sec
84	00:25:00	create FUNCTION Employee_ROLE(EXP vt) RETURNS VARCHAR(40) DETERMINISTIC BEGIN DECLARE Emp...	Error Code: 1304. FUNCTION Employee_ROLE already exists	0.000 sec
85	00:25:00	create FUNCTION Employee_ROLE(EXP vt) RETURNS VARCHAR(40) DETERMINISTIC BEGIN DECLARE Emp...	Error Code: 1304. FUNCTION Employee_ROLE already exists	0.000 sec
86	00:25:00	create FUNCTION Employee_ROLE(EXP vt) RETURNS VARCHAR(40) DETERMINISTIC BEGIN DECLARE Emp...	Error Code: 1304. FUNCTION Employee_ROLE already exists	0.000 sec
87	00:25:36	SELECT EXP, Employee_ROLE(EXP) FROM data_science_team LIMIT 0, 1000;	13 row(s) returned	0.000 sec / 0.000 sec

Query Completed

24°C Haze

ENG IN

00:31

24-02-2023

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 MySQL Model x EER Diagram x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

Tables

data_science_team

Columns

EMP_ID

FIRST_NAME

LAST_NAME

GENDER

ROLE

DEPT

EMP

COUNTRY

CONTINENT

SALARY

EMP_RATING

MANAGER_ID

PROJ_ID

Indexes

ForeignKeys

Triggers

proj_table

Views

StoredProcedures

Functions

sys

Administration Schemas

Information

Schema: employee

Query 1

SQL File 3

emp_record_table

Limit to 1000 rows

1 SELECT * FROM employee.emp_record_table;

Visual Explain

Display Info

Read

Eval cost

Overview

View Source

Query cost: 2.15

2.15 19 rows

Full Table Scan

emp_record_table

Access Type: ALL

Full Table Scan

Cost Hint: Very High - very costly for large tables (not so much for small ones).

No usable indexes were found for the table and the optimizer must search every row. This could also mean the search range is so broad that the index would be useless.

Used Columns: EMP_ID, FIRST_NAME, LAST_NAME, GENDER, ROLE, DEPT, EMP, COUNTRY, CONTINENT, SALARY, EMP_RATING, MANAGER_ID, PROJ_ID

Key/Indexes:

Rows Examined per Scan: 19

Rows Produced per Join: 19

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: 1.90

Prefix: 2.15

Data Read: 2K

emp_record_table 3

Output

Action Output

#	Time	Action	Message	Duration / Fetch
148	14:47:19	SELECT * FROM employee.emp_record_table WHERE FIRST_NAME = "Eric" LIMIT 0, 1000		0.000 sec / 0.000 sec
149	14:47:23	EXPLAIN SELECT * FROM employee.emp_record_table WHERE FIRST_NAME = "Eric"		0.000 sec
150	14:47:23	EXPLAIN FORMAT=JSON SELECT * FROM employee.emp_record_table WHERE FIRST_NAME = "Eric"		0.000 sec
151	14:48:02	SELECT * FROM employee.emp_record_table LIMIT 0, 1000		0.000 sec / 0.000 sec
152	14:48:05	EXPLAIN SELECT * FROM employee.emp_record_table		0.000 sec
153	14:48:05	EXPLAIN FORMAT=JSON SELECT * FROM employee.emp_record_table		0.000 sec

Query Completed

33°C Sunny

ENG IN

14:48

24-02-2023

MySQL Workbench

Local instance MySQL80 x MySQL Model x EER Diagram x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

Indexes

ForeignKeys

Triggers

emp_record_table

Columns

EMP_ID

FIRST_NAME

LAST_NAME

GENDER

ROLE

DEPT

EMP

COUNTRY

CONTINENT

SALARY

EMP_RATING

MANAGER_ID

PROJ_ID

Views

StoredProcedures

Functions

sys

Administration Schemas

Information

Schema: employee

Query 1

```

1 create index idx_first_name
2 on emp_record_table(FIRST_NAME(20));
3 select * from emp_record_table
4 where FIRST_NAME='Eric';

```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EMP	COUNTRY	CONTINENT	SALARY	EMP_RATING	MANAGER_ID	PROJ_ID
E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3	E103	P105

emp_record_table 22

Output

Action Output

#	Time	Action	Message	Duration / Fetch
85	00:25:25	create FUNCTION Employee_ROLE(EMP INT) RETURNS VARCHAR(40) DETERMINISTIC BEGIN DECLARE Emp...	Error Code: 1304. FUNCTION Employee_ROLE already exists	0.000 sec
86	00:25:30	create FUNCTION Employee_ROLE(EMP INT) RETURNS VARCHAR(40) DETERMINISTIC BEGIN DECLARE Emp...	Error Code: 1304. FUNCTION Employee_ROLE already exists	0.000 sec
87	00:25:36	SELECT EMP.Employee_ROLE(EMP) FROM data_science_team LIMIT 0, 1000 :	13 row(s) returned	0.000 sec / 0.000 sec
88	13:52:51	create FUNCTION Employee_ROLE(EMP INT) RETURNS VARCHAR(40) DETERMINISTIC BEGIN DECLARE Employee_ROLE varchar(40);	Error Code: 1304. FUNCTION Employee_ROLE already exists	0.000 sec
89	13:55:27	create index idx_first_name on emp_record_table(FIRST_NAME(20))	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.062 sec
90	13:55:27	select * from emp_record_table where FIRST_NAME='Eric' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Query Completed

33°C Sunny

ENG IN

24-02-2023

MySQL Workbench

Local instance MySQL80 x MySQL Model x EER Diagram x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

Indexes

ForeignKeys

Triggers

emp_record_table

Columns

EMP_ID

FIRST_NAME

LAST_NAME

GENDER

ROLE

DEPT

EMP

COUNTRY

CONTINENT

SALARY

EMP_RATING

MANAGER_ID

PROJ_ID

Views

StoredProcedures

Functions

sys

Administration Schemas

Information

Table: emp_record_table

Columns:

EMP_ID text

FIRST_NAME text

LAST_NAME text

GENDER text

ROLE text

DEPT int

COUNTRY text

CONTINENT text

SALARY int

EMP_RATING int

MANAGER_ID text

Object Info Session

Query 1

```

1 create index idx_first_name
2 on emp_record_table(FIRST_NAME(20));
3 select * from emp_record_table
4 where FIRST_NAME='Eric';

```

Visual Explain

Display Info Read Eval cost Overview View Source

Query cost: 0.35

0.35

1 row

Non-Unique Key Lookup

emp_record_table

idx_first_name

Access Types ref

Access Types 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, EMP, 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, idx1_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

Reads: 0.25

Eval: 0.10

Prefix: 0.35

Data Read: 120

emp_record_table 24

Output

Action Output

#	Time	Action	Message	Duration / Fetch
129	14:28:23	EXPLAIN FORMAT=JSON select * from emp_record_table where FIRST_NAME='Eric'		0.000 sec
130	14:29:01	create index idx_first_name on emp_record_table(FIRST_NAME(20))		0.000 sec
131	14:29:04	create index idx_first_name on emp_record_table(FIRST_NAME(20))		0.000 sec
132	14:29:33	create index idx_first_name on emp_record_table(FIRST_NAME(20))		0.000 sec
133	14:29:42	select * from emp_record_table where FIRST_NAME='Eric' LIMIT 0, 1000		0.000 sec / 0.000 sec
134	14:29:48	EXPLAIN select * from emp_record_table where FIRST_NAME='Eric'		0.000 sec

Query Completed

33°C Sunny

ENG IN

24-02-2023

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

employee

data_science_team

Columns

Indexes

Foreign Keys

Triggers

emp_record_table

Columns

EMP_ID

FIRST_NAME

LAST_NAME

GENDER

ROLE

DEPT

EMP

COUNTRY

CONTINENT

SALARY

EMP_RATING

MANAGER_ID

PROJ_ID

Indexes

Foreign Keys

Triggers

proj_table

Administration Schemas

Information

Schema: employee

Query 1 x SQL File 3 emp_record_table

Limit to 1000 rows

1 select EMP_ID, FIRST_NAME, LAST_NAME, SALARY, EMP_RATING, round((0.05 * SALARY * EMP_RATING),0) as Bonus

2 from emp_record_table

3 order by Bonus desc

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	SALARY	EMP_RATING	Bonus
E001	Arthur	Black	10500	5	4125
E083	Patrick	Voltz	9500	5	2375
E428	Pete	Allen	10000	4	2000
E103	Emily	Grove	10500	4	2100
E204	Karen	Nowak	7500	5	1875
E612	Tracy	Norris	8500	4	1760
E052	Dianna	Wilson	5500	5	1375
E005	Eric	Huffman	8500	3	1275
E260	Roy	Collins	7000	3	1050
E583	Janet	Hale	10000	2	1000
E010	William	Butler	9000	2	900
E478	David	Smith	4000	4	800
E403	Steve	Huffman	5000	3	750
E245	Nan	Zhen	6500	2	650
E640	Jennifer	Jones	2800	4	560
E505	Chad	Wilson	5000	2	500
E057	Dorothy	Wilson	7700	1	385
E532	Claire	Brennan	4300	1	215
E620	Katrina	Allen	3000	1	150

Result 37 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
149	14:47:23	EXPLAIN SELECT * FROM employee emp_record_table where FIRST_NAME = "Eric"	OK	0.000 sec
150	14:47:23	EXPLAIN FORMAT=JSON SELECT * FROM employee emp_record_table where FIRST_NAME = "Eric"	OK	0.000 sec
151	14:48:02	SELECT * FROM employee emp_record_table LIMIT 0, 1000	19 row(s) returned	0.000 sec / 0.000 sec
152	14:48:05	EXPLAIN SELECT * FROM employee emp_record_table	OK	0.000 sec
153	14:48:05	EXPLAIN FORMAT=JSON SELECT * FROM employee emp_record_table	OK	0.000 sec
154	14:53:09	select EMP_ID, FIRST_NAME, LAST_NAME, SALARY, EMP_RATING, round((0.05 * SALARY * EMP_RATING),0) as Bonus	19 row(s) returned	0.000 sec / 0.000 sec

Query Completed

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 MySQL Model x EER Diagram x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employee

data_science_team

Columns

Indexes

Foreign Keys

Triggers

emp_record_table

Columns

EMP_ID

FIRST_NAME

LAST_NAME

GENDER

ROLE

DEPT

EMP

COUNTRY

CONTINENT

SALARY

EMP_RATING

MANAGER_ID

PROJ_ID

Indexes

Foreign Keys

Triggers

proj_table

Administration Schemas

Information

Schema: employee

Query 1 x SQL File 3 emp_record_table

Limit to 1000 rows

1 select EMP_ID, FIRST_NAME, LAST_NAME, COUNTRY, CONTINENT,

2 round(avg(SALARY) over (partition by CONTINENT order by COUNTRY),0) as Average_Salary

3 from emp_record_table

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	COUNTRY	CONTINENT	Average_Salary
E245	Nan	Zhen	CHINA	ASIA	6500
E260	Roy	Collins	INDIA	ASIA	6290
E612	Tracy	Norris	INDIA	ASIA	6290
E620	Katrina	Allen	INDIA	ASIA	6290
E010	William	Butler	FRANCE	EUROPE	9000
E204	Karen	Nowak	GERMANY	EUROPE	7950
E428	Pete	Allen	GERMANY	EUROPE	7950
E532	Claire	Brennan	GERMANY	EUROPE	7950
E052	Dianna	Wilson	CANADA	NORTH AMERICA	7000
E103	Emily	Grove	CANADA	NORTH AMERICA	7000
E001	Arthur	Black	USA	NORTH AMERICA	8525
E005	Eric	Huffman	USA	NORTH AMERICA	8525
E057	Dorothy	Wilson	USA	NORTH AMERICA	8525
E083	Patrick	Voltz	USA	NORTH AMERICA	8525
E403	Steve	Huffman	USA	NORTH AMERICA	8525
E478	David	Smith	COLOMBIA	SOUTH AMERICA	5600
E583	Janet	Hale	COLOMBIA	SOUTH AMERICA	5600
E640	Jennifer	Jones	COLOMBIA	SOUTH AMERICA	5600

Result 38 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
150	14:47:23	EXPLAIN FORMAT=JSON SELECT * FROM employee emp_record_table where FIRST_NAME = "Eric"	OK	0.000 sec
151	14:48:02	SELECT * FROM employee emp_record_table LIMIT 0, 1000	19 row(s) returned	0.000 sec / 0.000 sec
152	14:48:05	EXPLAIN SELECT * FROM employee emp_record_table	OK	0.000 sec
153	14:48:05	EXPLAIN FORMAT=JSON SELECT * FROM employee emp_record_table	OK	0.000 sec
154	14:53:09	select EMP_ID, FIRST_NAME, LAST_NAME, SALARY, EMP_RATING, round((0.05 * SALARY * EMP_RATING),0) as Bonus	19 row(s) returned	0.000 sec / 0.000 sec
155	14:57:29	select EMP_ID, FIRST_NAME, LAST_NAME, COUNTRY, CONTINENT, round(avg(SALARY) over (partition by CO...	19 row(s) returned	0.000 sec / 0.000 sec

Query Completed

