

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.

MySQL Workbench

employee_performance_map... x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employee

- Tables
- Views
- Stored Procedures
- Functions

sql_intro

sys

Query 1

```
1 • create database if not exists employee;  
2
```

Limit to 1000 rows

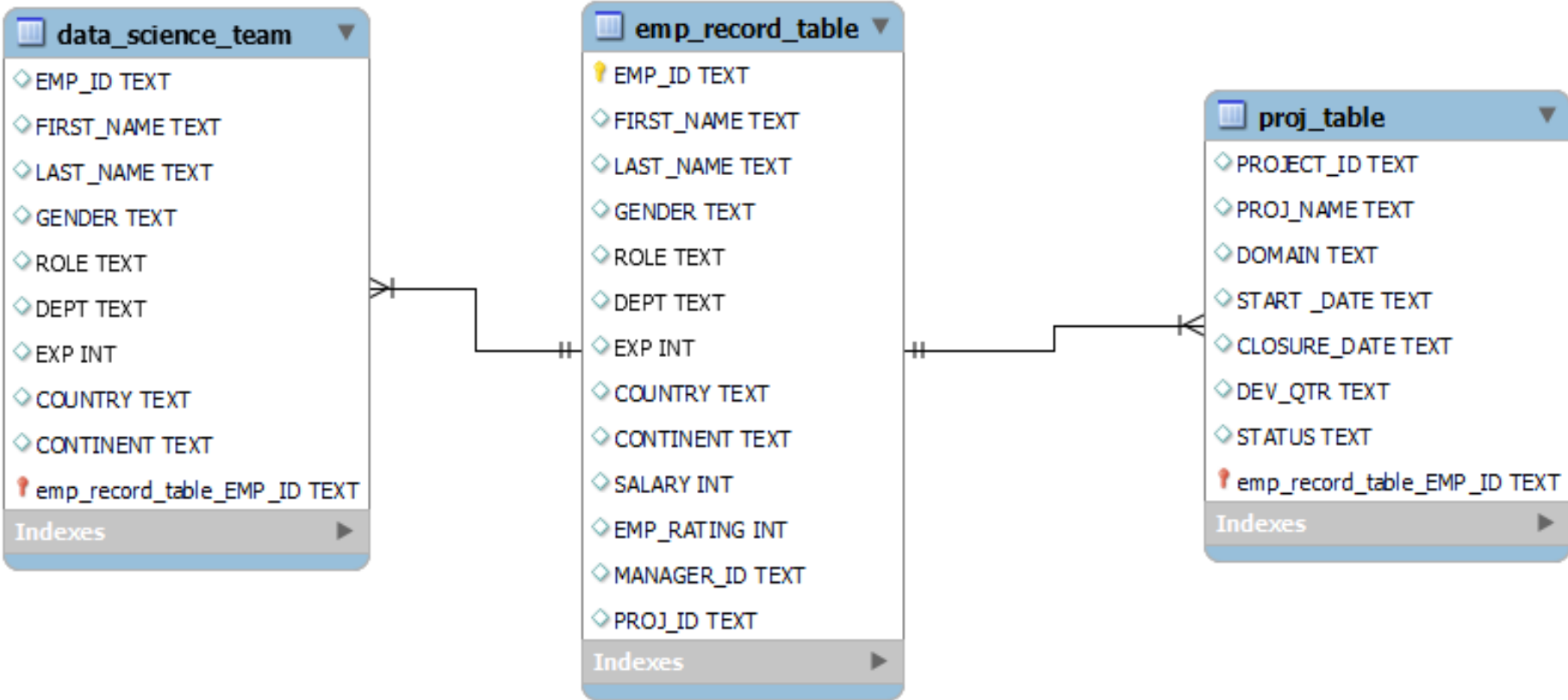
Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	21:30:34	create database if not exists employee	1 row(s) affected	0.156 sec

Schema: 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.

MySQL Workbench

employee_performance_map... x MySQL Model* x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 SQL File 3* x

SCHEMAS

Filter objects

employee

- Tables
 - data_science_team
 - emp_record_table
 - proj_table
- Views
- Stored Procedures
- Functions

sql_intro

- Tables
- Views
- Stored Procedures
- Functions

sys

Administration Schemas

Information

Schema: employee

Query 1

```
1 • select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT from emp_record_table;
```

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
	E500	Chad	Wilson	M	HEALTHCARE

emp_record_table 2 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:

- Less than two
- Greater than four
- Between two and four

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left shows the 'employee' schema with tables 'data_science_team', 'emp_record_table', and 'proj_table'. The 'Query' editor shows a SQL query to fetch employee details where the rating is less than 2. The 'Result Grid' shows the results of the query.

MySQL Workbench

employee_performance_map... x MySQL Model* x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 SQL File 3* SQL File 4* x emp_record_table

Schemas

Filter objects

employee

- Tables
 - data_science_team
 - emp_record_table
 - proj_table
- Views
- Stored Procedures
- Functions

sql_intro

- Tables
- Views
- Stored Procedures
- Functions

sys

Administration Schemas

Information

Schema: employee

```
1 -- fetching employee which is having rating less than two
2 • select EMP_ID,FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record_table
3 where EMP_RATING < 2;
```

Result Grid

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

record_table 1 x

Read Only

MySQL Workbench

employee_performance_map... x MySQL Model* x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employee

- Tables
 - data_science_team
 - emp_record_table
 - proj_table
- Views
- Stored Procedures
- Functions

sql_intro

- Tables
- Views
- Stored Procedures
- Functions

sys

Administration Schemas

Information

Schema: employee

Query 1 SQL File 3* SQL File 4* SQL File 5* x

Limit to 2000 rows

```
1 -- fetching employee which is having rating greater than four
2 • select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record_table
3 where EMP_RATING > 4;
```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT	EMP_RATING
E001	Arthur	Black	M	ALL	5
E052	Dianna	Wilson	F	HEALTHCARE	5
E083	Patrick	Voltz	M	HEALTHCARE	5
E204	Karene	Nowak	F	AUTOMOTIVE	5

record_table 1 x

Read Only

MySQL Workbench

employee_performance_map... x MySQL Model* x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employee

- Tables
 - data_science_team
 - emp_record_table
 - proj_table
- Views
- Stored Procedures
- Functions

sql_intro

- Tables
- Views
- Stored Procedures
- Functions

sys

Administration Schemas

Information

Schema: employee

Query 1 SQL File 3* SQL File 4* SQL File 5* SQL File 6* x

Limit to 2000 rows

```
1 -- fetching employee which is having rating between two and four
2 • select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record_table
3 where EMP_RATING between 2 and 4;
```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT	EMP_RATING
E005	Eric	Hoffman	M	FINANCE	3
E010	William	Butler	M	AUTOMOTIVE	2
E103	Emily	Grove	F	FINANCE	4
E245	Nian	Zhen	M	RETAIL	2
E260	Roy	Collins	M	RETAIL	3
E403	Steve	Hoffman	M	FINANCE	3
E428	Pete	Allen	M	AUTOMOTIVE	4
E478	David	Smith	M	RETAIL	4
E505	Chad	Wilson	M	HEALTHCARE	2

record_table 1 x

Read Only

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 'SQL File 7*':

```
1 • select concat(FIRST_NAME, " ", LAST_NAME) AS "NAME" from emp_record_table
2 where DEPT="FINANCE";
```

Below the query editor, the 'Result Grid' is visible, showing the output of the query. The grid has a single column named 'NAME' with three rows of data:

NAME
Eric Hoffman
Emily Grove
Steve Hoffman

The bottom status bar indicates 'Schema: employee' and 'Result 3 x'.

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' database selected. The main editor shows a SQL query in 'SQL File 8*' that counts the number of reporters for each manager. The 'Result Grid' at the bottom displays the query results.

Query 1 SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 8* x

```
1 • select MANAGER_ID,COUNT(EMP_ID) AS "Number_of_Reporter"
2   from emp_record_table
3  group by MANAGER_ID;
```

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

MANAGER_ID	Number_of_Reporter
NULL	1
E103	2
E428	3
E083	3
E001	5
E583	3
E612	2

Administration Schemas

Information

No object selected

Result 1 x 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.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'employee' database selected, showing tables like 'data_science_team', 'emp_record_table', and 'proj_table'. The main editor window shows a SQL query in 'Query 1' using the 'SQL File 9*' tab. The query is as follows:

```
1 • select EMP_ID,concat(FIRST_NAME," ",LAST_NAME) as EMP_NAME,DEPT from emp_record_table
2   where DEPT="HEALTHCARE"
3   union
4   select EMP_ID,concat(FIRST_NAME," ",LAST_NAME) as EMP_NAME,DEPT from emp_record_table
5   where DEPT="FINANCE";
```

Below the query editor, the 'Result Grid' is displayed, showing the results of the query. The grid has columns for EMP_ID, EMP_NAME, and DEPT. The results are as follows:

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

The bottom status bar indicates 'Result 6' and '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.

MySQL Workbench

employee_performance_map ... x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Filter objects

employee

- Tables
 - data_science_team
 - emp_record_table
 - proj_table
- Views
- Stored Procedures
- Functions

sql_intro

sys

Query 1 SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* x

Limit to 2000 rows

```
1 -- using window function
2 select EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EMP_RATING,
3 max(EMP_RATING) over(partition by DEPT) as "MAXIMUM_EMP_RATING_IN_DEPT"
4 from emp_record_table;
```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EMP_RATING	MAXIMUM_EMP_RATING_IN_DEPT
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

Result 3 x

Read Only

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

employee_performance_map... x

File Edit View Query Database Server Tools Scripting Help

Navigator:.....

SCHEMAS

Filter objects

▼ **employee**

- ▼ Tables
 - data_science_team
 - emp_record_table
 - proj_table
- Views
- Stored Procedures
- Functions

► sql_intro

► sys

Query 1 SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* x

Limit to 2000 rows

```
1 • select ROLE,MIN(SALARY) as "MINIMUM_SALARY",MAX(SALARY) as "MAXIMUM_SALARY" from emp_record_table
2 group by ROLE;
```

Result Grid

ROLE	MINIMUM_SALARY	MAXIMUM_SALARY
PRESIDENT	16500	16500
LEAD DATA SCIENTIST	8500	9000
SENIOR DATA SCIENTIST	5500	7700
MANAGER	8500	11000
ASSOCIATE DATA SCIENTIST	4000	5000
JUNIOR DATA SCIENTIST	2800	3000

Administration Schemas

Information

No object selected

Result 5 x

Read Only

Output

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

MySQL Workbench

employee_performance_map... x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* x

SCHEMAS

Filter objects

- employee
 - Tables
 - data_science_team
 - emp_record_table
 - proj_table
 - Views
 - Stored Procedures
 - Functions
- sql_intro
- sys

Administration Schemas

Information

No object selected

```
1 -- using window ranking function ,considering higher experience as a first rank
2 • select EMP_ID,CONCAT(FIRST_NAME," ",LAST_NAME) AS "NAME" ,EXP,
3 rank() over(order by EXP desc) as "RANK_OF_EMPLOYEE"
4 from emp_record_table;
```

Result Grid

	EMP_ID	NAME	EXP	RANK_OF_EMPLOYEE
▶	E001	Arthur Black	20	1
	E083	Patrick Voltz	15	2
	E103	Emily Grove	14	3
	E428	Pete Allen	14	3
	E583	Janet Hale	14	3
	E612	Tracy Norris	13	6
	E010	William Butler	12	7
	E005	Eric Hoffman	11	8
	E057	Dorothy Wilson	9	9
	E204	Karene Nowak	8	10
	E260	Roy Collins	7	11
	E052	Dianna Wilson	6	12
	E245	Nian Zhen	6	12
	E005	Shad Wilson	5	14

Result 3 x

Read Only

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 the 'employee' schema selected. The main editor shows a SQL query to create a view named 'employee_in_various_countries' and then select from it. The result grid below the query shows the data for the view.

```
1  -- creating view
2  • create view employee_in_various_countries as
3  select EMP_ID,concat(FIRST_NAME," ",LAST_NAME) AS "NAME",COUNTRY,SALARY
4  from emp_record_table
5  where SALARY>6000;
6  -- displaying view
7  • select * from employee_in_various_countries;
```

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

employee_in_various_countries 3 x

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. On the left, the 'SCHEMAS' pane displays a tree view with 'employee' expanded, showing tables like 'emp_record_table'. The main editor contains a SQL query using a nested subquery to filter employees by experience. Below the query, the 'Result Grid' shows the output of the query, listing employee names and their experience in years.

MySQL Workbench

employee_performance_map... x

File Edit View Query Database Server Tools Scripting Help

Navigator:.....

SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* x

Limit to 2000 rows

```
1 -- using nested query/subquery finding Employee which is having experience more than Ten year
2 select concat(FIRST_NAME," ",LAST_NAME) as "EMP_NAME",EXP as "EXPERIENCE" from emp_record_table
3 where EXP in (select EXP from emp_record_table where EXP>10);
4
5
6
```

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

EMP_NAME	EXPERIENCE
Arthur Black	20
Eric Hoffman	11
William Butler	12
Patrick Voltz	15
Emily Grove	14
Pete Allen	14
Janet Hale	14
Tracy Norris	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 left sidebar displays the 'SCHEMAS' tree with the 'employee' database selected. Under 'Stored Procedures', the 'emp_exp' procedure is listed. The main editor window shows the SQL script for creating and calling the stored procedure. The script is as follows:

```
1  -- creating stored procedure for employee which is having experience more than three year
2  delimiter &&
3  • create procedure emp_exp()
4  begin
5      select EMP_ID,concat(FIRST_NAME," ",LAST_NAME) as "EMP_NAME",EXP
6      from emp_record_table
7      where EXP>3;
8  end &&
9  -- setting again default delimiter
10 delimiter ;
11 • -- calling stored procedure
12 call emp_exp();
```

Below the script, the 'Result Grid' shows the output of the stored procedure call. It displays a table with three columns: EMP_ID, EMP_NAME, and EXP. The data is as follows:

EMP_ID	EMP_NAME	EXP
E001	Arthur Black	20
E005	Eric Hoffman	11
E010	William Butler	12
E052	Dianna Wilson	6
E057	Dorothy Wilson	9
E083	Patrick Voltz	15
E103	Emily Grove	14
E204	Karene Nowak	8

The bottom status bar indicates 'Result 3' and 'Read Only'.

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

employee_performance_map... x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Filter objects

employee

- Tables
 - data_science_team
 - emp_record_table
 - proj_table
- Views
 - employee_in_various_c
- Stored Procedures
 - emp_exp
- Functions
 - f() emp_job_profile

Administration Schemas

Information

Schema: employee

SQL File 5* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16* x

Limit to 2000 rows

```
1 -- creating stored function to assign job profile based on their experience
2 delimiter $$
3 • create function emp_job_profile(EXP int)
4 returns varchar(30) deterministic
5 begin
6 declare job_profile varchar(30) default "";
7 if EXP<=2 then set job_profile="JUNIOR DATA SCIENTIST";
8 elseif (EXP>=2 and EXP<=5) then set job_profile="ASSOCIATE DATA SCIENTIST";
9 elseif (EXP>=5 and EXP<=10) then set job_profile="SENIOR DATA SCIENTIST";
10 elseif(EXP>=10 and EXP<=12) then set job_profile="LEAD DATA SCIENTIST";
11 elseif(EXP>=12 and EXP<=16) then set job_profile="MANAGER";
12 end if;
13 return job_profile;
14 end $$
15 delimiter ;
16 • -- calling stored function
17 select concat(FIRST_NAME, " ", LAST_NAME) as "EMP_NAME", emp_job_profile(EXP) as "JOB_PROFILE" from data_science_team;
```

Result Grid

EMP_NAME	JOB_PROFILE
Eric Hoffman	LEAD DATA SCIENTIST
William Butler	LEAD DATA SCIENTIST
Dianna Wilson	SENIOR DATA SCIENTIST
Dorothy Wilson	SENIOR DATA SCIENTIST
Karene Nowak	SENIOR DATA SCIENTIST

Result 2 x Read Only

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. The left sidebar displays the 'SCHEMAS' tree with the 'employee' database selected. The main editor shows a SQL query in 'SQL File 17':

```
1 -- checking execution plan before creating index on table
2 • select * from emp_record_table
3 where FIRST_NAME="Eric";
```

The 'Visual Explain' tab is active, showing the execution plan for the query. The plan consists of a single block, 'query_block #1', which performs a 'Full Table Scan' on the 'emp_record_table'. The query cost is 2.15, and it returns 19 rows.

We can see in above screenshot, here Query Cost is high.

The screenshot shows the MySQL Workbench interface after creating an index. The main editor shows a SQL query in 'SQL File 18':

```
1 -- creating index to speedup searches/query and improve the cost and performance
2 • create index first_name_index on
3 emp_record_table(FIRST_NAME(30));
4 -- executing queries
5 • select * from emp_record_table
6 where FIRST_NAME="Eric";
```

The 'Visual Explain' tab is active, showing the execution plan for the query. The plan consists of a single block, 'query_block #1', which performs a 'Non-Unique Key Lookup' on the 'emp_record_table'. The query cost is 0.35, and it returns 1 row.

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 left sidebar displays the 'SCHEMAS' tree with the 'employee' database selected. The 'Tables' section is expanded, showing 'data_science_team', 'emp_record_table', and 'proj_table'. The 'Views' section shows 'employee_in_various_c'. The 'Stored Procedures' section shows 'emp_exp'. The 'Functions' section shows 'f() emp_job_profile'. The 'sys' database is also visible.

The main editor window shows a SQL query in 'SQL File 19*':

```
1 • select EMP_ID, CONCAT(FIRST_NAME, " ", LAST_NAME) as "EMP_NAME", SALARY, EMP_RATING, convert(((0.05*SALARY)*EMP_RATING), decimal(10,0)) as BONUS_AMOUNT
2 from emp_record_table;
```

The query results are displayed in the 'Result Grid' at the bottom. The grid shows 8 rows of data with columns: EMP_ID, EMP_NAME, SALARY, EMP_RATING, and BONUS_AMOUNT.

EMP_ID	EMP_NAME	SALARY	EMP_RATING	BONUS_AMOUNT
E001	Arthur Black	16500	5	4125
E005	Eric Hoffman	8500	3	1275
E010	William Butler	9000	2	900
E052	Dianna Wilson	5500	5	1375
E057	Dorothy Wilson	7700	1	385
E083	Patrick Voltz	9500	5	2375
E103	Emily Grove	10500	4	2100

The bottom status bar indicates 'Result 8' and 'Read Only'.

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

employee_performance_map...

File Edit View Query Database Server Tools Scripting Help

Navigator: SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16* SQL File 17* SQL File 18* SQL File 19* SQL File 20* x

SCHEMAS

Filter objects

- employee
 - Tables
 - data_science_team
 - emp_record_table
 - proj_table
 - Views
 - employee_in_various_c
 - Stored Procedures
 - emp_exp
 - Functions
 - emp_job_profile
- sql_intro
- sys

```
1 -- using window function
2 • select * from emp_record_table;
3 • select EMP_ID,concat(FIRST_NAME," ",LAST_NAME) as "EMP_NAME",SALARY,CONTINENT,COUNTRY,
4   avg(SALARY) over (partition by CONTINENT) as "CONTINENT_AVG_SALARY",
5   avg(SALARY) over (partition by COUNTRY) as "COUNTRY_AVG_SALARY"
6   from emp_record_table;
```

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

	EMP_ID	EMP_NAME	SALARY	CONTINENT	COUNTRY	CONTINENT_AVG_SALARY	COUNTRY_AVG_SALARY
▶	E052	Dianna Wilson	5500	NORTH AMERICA	CANADA	8525.0000	7000.0000
	E103	Emily Grove	10500	NORTH AMERICA	CANADA	8525.0000	7000.0000
	E505	Chad Wilson	5000	NORTH AMERICA	CANADA	8525.0000	7000.0000
	E245	Nian Zhen	6500	ASIA	CHINA	6250.0000	6500.0000
	E478	David Smith	4000	SOUTH AMERICA	COLOMBIA	5600.0000	5600.0000
	E583	Janet Hale	10000	SOUTH AMERICA	COLOMBIA	5600.0000	5600.0000
	E640	Jenifer Jhones	2800	SOUTH AMERICA	COLOMBIA	5600.0000	5600.0000

Result 5 x

Read Only