

ScienceQtech 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 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.

Note: You must download the dataset from the course resource section in LMS and create a table to perform the above objective.

Dataset description:

emp_record_table: It contains the information of all the employees.

- EMP_ID – ID of the employee
- FIRST_NAME – First name of the employee
- LAST_NAME – Last name of the employee
- GENDER – Gender of the employee
- ROLE – Post of the employee
- DEPT – Field of the employee
- EXP – Years of experience the employee has
- COUNTRY – Country in which the employee is presently living
- CONTINENT – Continent in which the country is
- SALARY – Salary of the employee
- EMP_RATING – Performance rating of the employee
- MANAGER_ID – The manager under which the employee is assigned
- PROJ_ID – The project on which the employee is working or has worked on

Proj_table: It contains information about the projects.

- PROJECT_ID – ID for the project
- PROJ_Name – Name of the project
- DOMAIN – Field of the project
- START_DATE – Day the project began
- CLOSURE_DATE – Day the project was or will be completed
- DEV_QTR – Quarter in which the project was scheduled
- STATUS – Status of the project currently

Data_science_team: It contains information about all the employees in the Data Science team.

- EMP_ID – ID of the employee
- FIRST_NAME – First name of the employee
- LAST_NAME – Last name of the employee
- GENDER – Gender of the employee
- ROLE – Post of the employee
- DEPT – Field of the employee
- EXP – Years of experience the employee has
- COUNTRY – Country in which the employee is presently living
- CONTINENT – Continent in which the country is

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.
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
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.
 6. Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President).
 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.
 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.
 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.
 10. Write a query to assign ranks to each employee based on their experience. Take data from the employee 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.
 12. Write a nested query to find employees with experience of more than ten years. Take data from the employee 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.
 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',

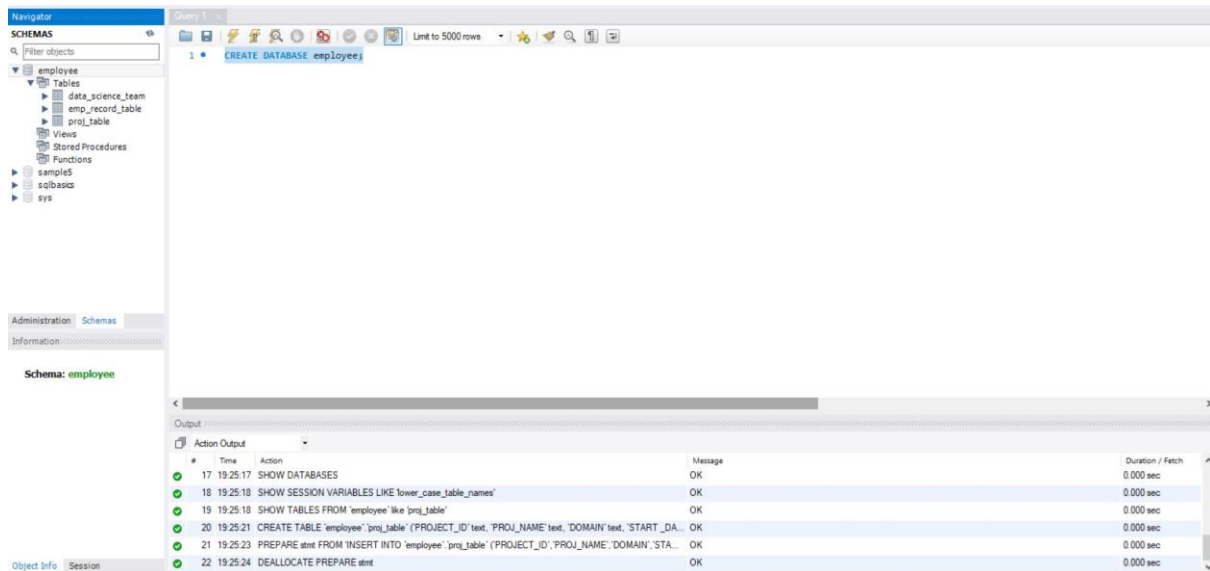
For an employee with the experience of 10 to 12 years assign 'LEAD DATA SCIENTIST',

For an employee with the experience of 12 to 16 years assign 'MANAGER'.

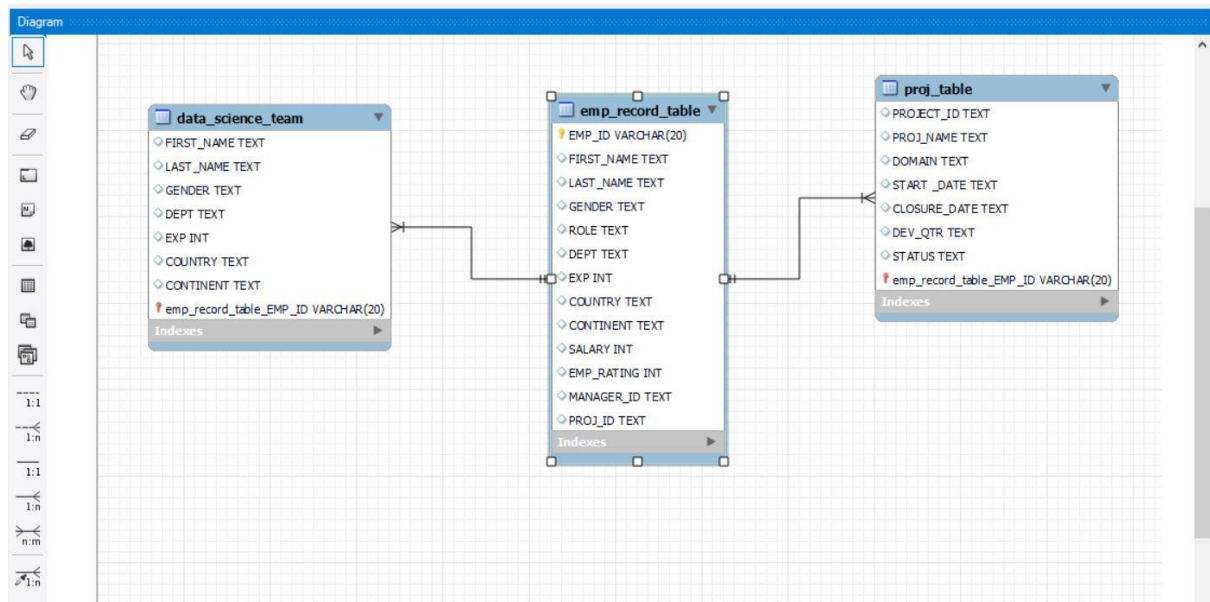
15. Create an index to improve the cost and performance of the query to find the employee whose FIRST_NAME is 'Eric' in the employee table after checking the execution plan.
16. Write a query to calculate the bonus for all the employees, based on their ratings and salaries (Use the formula: 5% of salary * employee rating).
17. Write a query to calculate the average salary distribution based on the continent and country. Take data from the employee record table.

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.

1A. CREATE DATABASE employee;



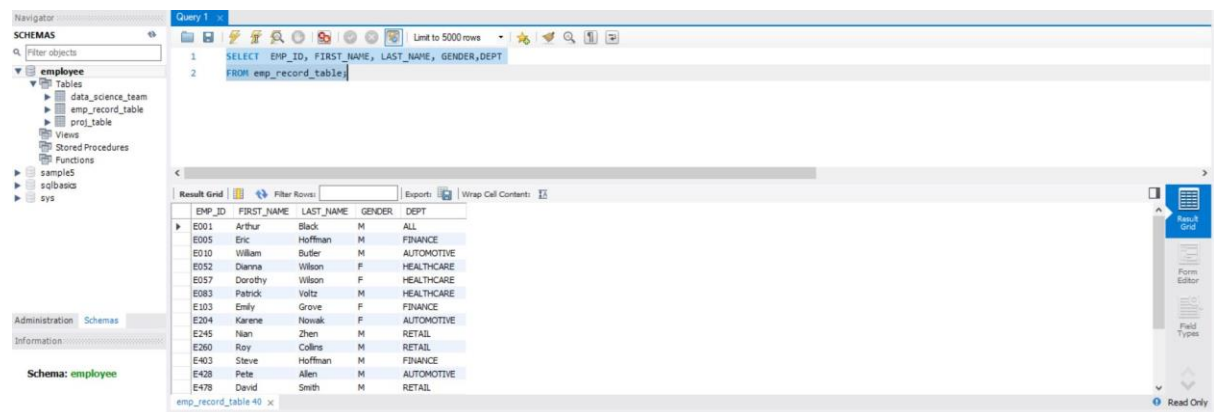
2. Create an ER diagram for the given **employee** database.



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.

3A.

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



The screenshot shows the SQL Developer interface. The 'Schemas' pane on the left shows the 'employee' schema. The 'Query' window contains the following SQL query:

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

The 'Result Grid' shows the following data:

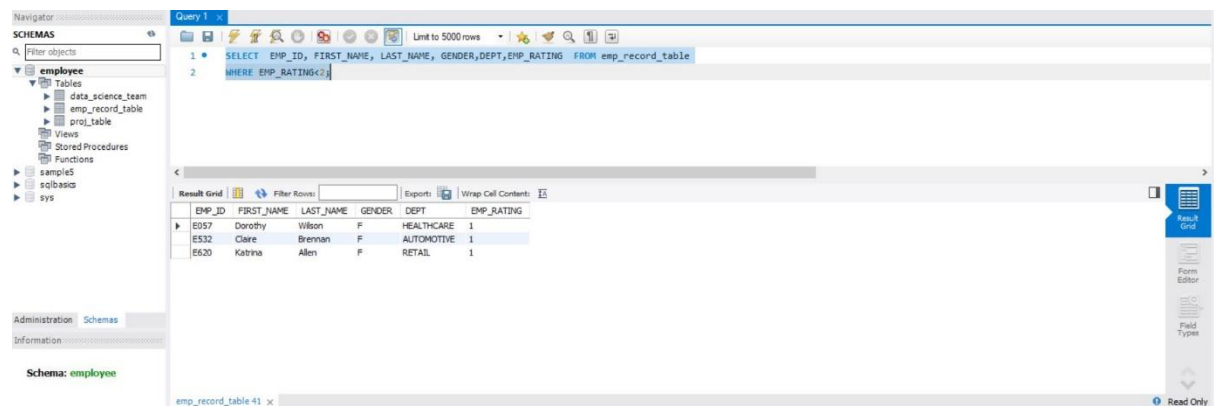
EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT
E001	Arthur	Black	M	ALL
E005	Eric	Hoffman	M	FINANCE
E010	William	Butler	M	AUTOMOTIVE
E052	Dianna	Wilson	F	HEALTHCARE
E057	Dorothy	Wilson	F	HEALTHCARE
E083	Patrick	Voltz	M	HEALTHCARE
E103	Emily	Grove	F	FINANCE
E204	Karen	Novak	F	AUTOMOTIVE
E245	Nan	Zhen	M	RETAIL
E260	Roy	Collins	M	RETAIL
E403	Steve	Hoffman	M	FINANCE
E428	Pete	Allen	M	AUTOMOTIVE
E478	David	Smith	M	RETAIL

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

4A. less than two

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



The screenshot shows the SQL Developer interface. The 'Schemas' pane on the left shows the 'employee' schema. The 'Query' window contains the following SQL query:

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

The 'Result Grid' shows the following data:

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

4B.

greater than four

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

```
WHERE EMP_RATING>4;
```

The screenshot shows the SQL Developer interface. The 'Navigator' pane on the left displays the 'employee' schema with tables 'data_science_team', 'emp_record_table', and 'proj_table'. The 'Query' pane shows the following SQL query:

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

The 'Result Grid' displays the following data:

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT	EMP_RATING
E001	Arthur	Black	M	ALL	5
E032	Dennis	Wilson	F	HEALTHCARE	5
E083	Patrick	Voltz	M	HEALTHCARE	5
E204	Karen	Nowak	F	AUTOMOTIVE	5

4C.

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;
```

The screenshot shows the SQL Developer interface. The 'Navigator' pane on the left displays the 'employee' schema with tables 'data_science_team', 'emp_record_table', and 'proj_table'. The 'Query' pane shows the following SQL query:

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

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
E403	Steve	Hoffman	M	FINANCE	3
E426	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

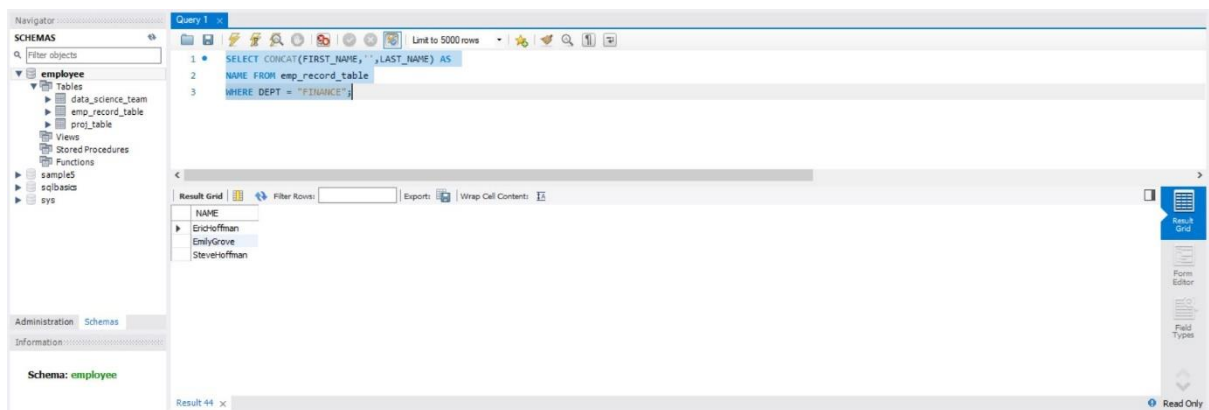
5. Write a query to concatenate the FIRST_NAME and the LAST_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME.

5A.

```
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).

6A.

```
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;
```


Query 1

```

1 SELECT m.EMP_ID,m.FIRST_NAME,m.LAST_NAME,m.ROLE,
2 m.EXP,COUNT(e.EMP_ID) as "EMP_COUNT"
3 FROM emp_record_table m
4 INNER JOIN emp_record_table e
5 ON m.EMP_ID = e.MANAGER_ID
6 GROUP BY m.EMP_ID
7 ORDER BY m.EMP_ID

```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	EXP	EMP_COUNT
E001	Arthur	Black	PRESIDENT	20	5
E083	Patrick	Voltz	MANAGER	15	3
E103	Emily	Grove	MANAGER	14	2
E428	Pete	Allen	MANAGER	14	3
E583	Janet	Halle	MANAGER	14	3
E612	Tracy	Norris	MANAGER	13	2

Result 45 x

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.

7A.

```
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;
```

Query 1

```

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

```

Result Grid

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

Result 46 x

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.

8A.

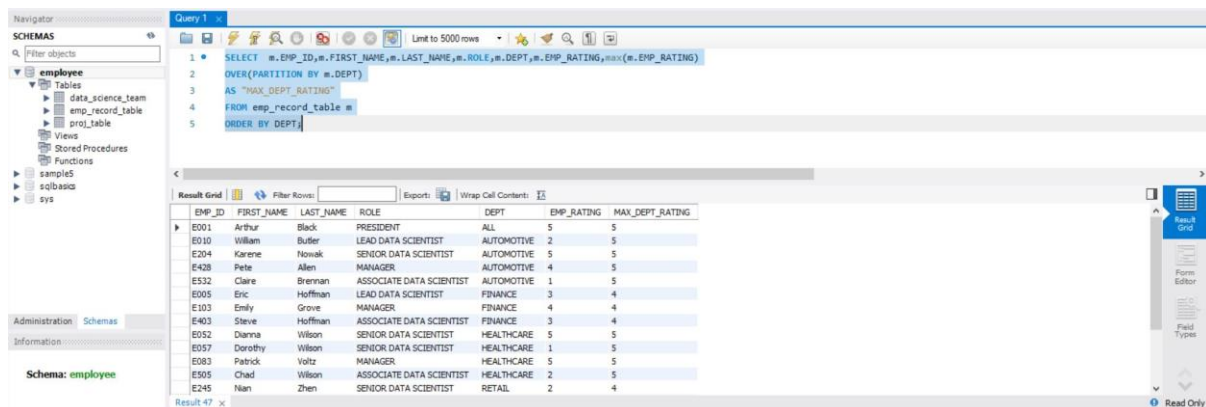
```
SELECT
m.EMP_ID,m.FIRST_NAME,m.LAST_NAME,m.ROLE,m.DEPT,m.EMP_RATING,max(m.EMP_RATING)

OVER(PARTITION BY m.DEPT)

AS "MAX_DEPT_RATING"

FROM emp_record_table m

ORDER BY DEPT;
```



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	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	Huffman	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
E305	Chad	Wilson	ASSOCIATE DATA SCIENTIST	HEALTHCARE	2	5
E245	Nan	Zhen	SENIOR DATA SCIENTIST	RETAIL	2	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.

9A.

```
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;
```

The screenshot shows a SQL query in the 'Query 1' window. The query is:

```

1 SELECT EMP_ID, FIRST_NAME, LAST_NAME, ROLE, MAX(SALARY), MIN(SALARY)
2 FROM emp_record_table
3 WHERE ROLE IN ('PRESIDENT', 'LEAD DATA SCIENTIST', 'SENIOR DATA SCIENTIST', 'MANAGER', 'ASSOCIATE DATA SCIENTIST', 'JUNIOR DATA SCIENTIST')
4 GROUP BY ROLE;

```

The 'Result Grid' shows the following data:

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	MAX(SALARY)	MIN(SALARY)
E001	Arthur	Black	PRESIDENT	16500	16500
E005	Eric	Hoffman	LEAD DATA SCIENTIST	9000	8500
E052	Dianna	Wilson	SENIOR DATA SCIENTIST	7700	5500
E083	Patrick	Voltz	MANAGER	11000	8500
E403	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	3000	4000
E620	Katrina	Allen	JUNIOR DATA SCIENTIST	3000	2800

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

10A.

```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP,
```

```
RANK() OVER (ORDER BY EXP) EXP_RANK
```

```
FROM emp_record_table;
```

The screenshot shows a SQL query in the 'Query 1' window. The query is:

```

1 SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP,
2 RANK() OVER (ORDER BY EXP) EXP_RANK
3 FROM emp_record_table;
4

```

The 'Result Grid' shows the following data:

EMP_ID	FIRST_NAME	LAST_NAME	EXP	EXP_RANK
E640	Jenifer	Jhones	1	1
E620	Katrina	Allen	2	2
E478	David	Smith	3	3
E532	Claire	Brennan	3	3
E403	Steve	Hoffman	4	5
E505	Chad	Wilson	5	6
E052	Dianna	Wilson	6	7
E245	Nan	Zhen	6	7
E260	Roy	Collins	7	9
E204	Karen	Novak	8	10
E057	Dorothy	Wilson	9	11
E005	Eric	Hoffman	11	12
E010	William	Butler	12	13

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.

11A.

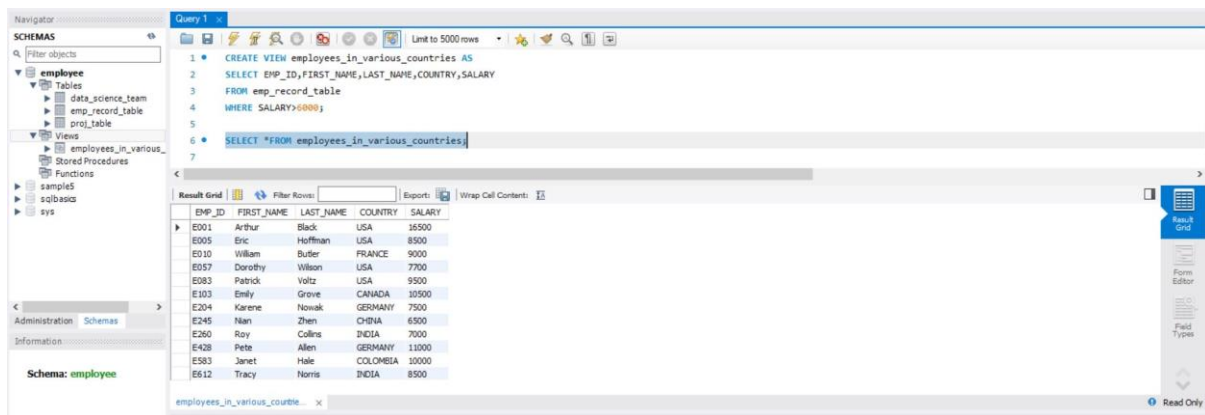
```
CREATE VIEW employees_in_various_countries AS
```

```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, COUNTRY, SALARY
```

```
FROM emp_record_table
```

```
WHERE SALARY > 6000;
```

SELECT *FROM employees_in_various_countries;

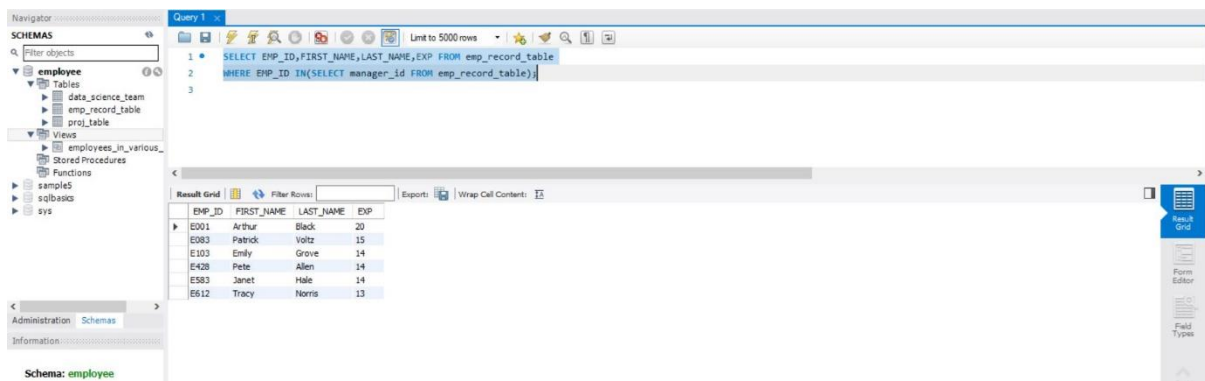


12. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

12A.

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.

13A.

DELIMITER &&

CREATE PROCEDURE get_experience_details()

BEGIN

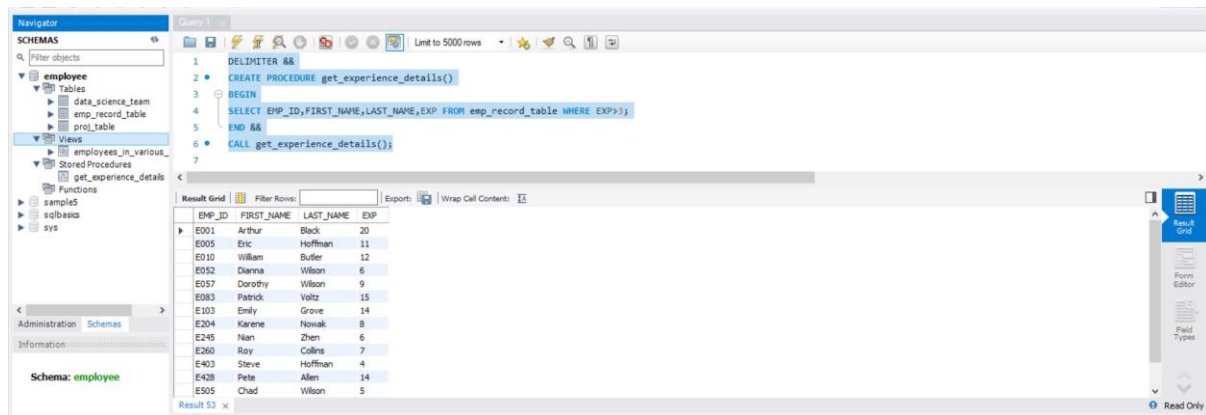
```

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'.

14A.

```
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 &&

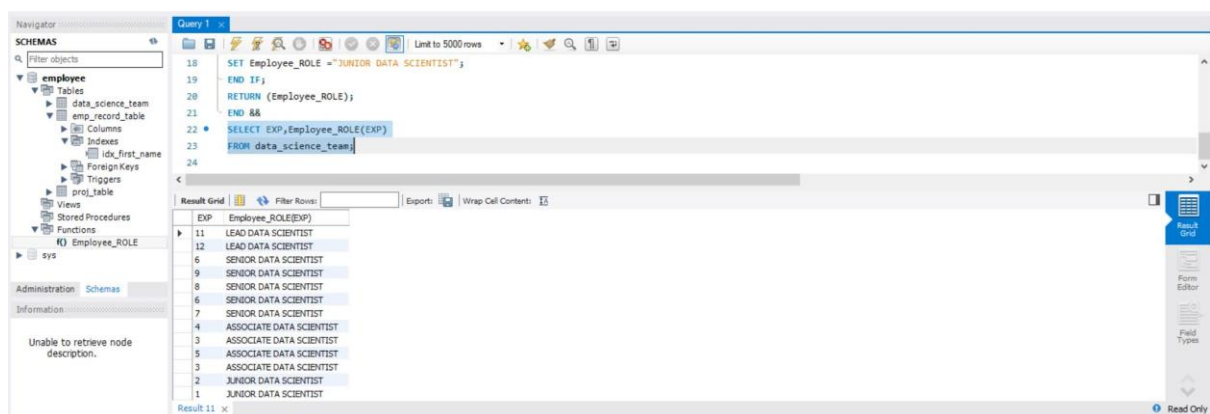
```

```

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.

15A.

```

CREATE INDEX idx_first_name

```


ON emp_record_table(FIRST_NAME(20));

SELECT * FROM emp_record_table

WHERE FIRST_NAME='Eric';

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

```
2  
3  
4  
5 CREATE INDEX idx_first_name  
6 ON emp_record_table(FIRST_NAME(20));  
7 SELECT * FROM emp_record_table  
8 WHERE FIRST_NAME='Eric';
```

The query is executed, and the Explain Plan is displayed. The plan shows a full table scan for the emp_record_table. The cost is 2.15, and 19 rows are returned. The execution plan is as follows:

```
graph TD
    query_block[query_block#1] --> full_table_scan[Full Table Scan]
    full_table_scan --> emp_record_table[emp_record_table]
```

The emp_record_table is accessed via a Full Table Scan. The cost is 2.15, and 19 rows are returned. The execution plan is as follows:

```
graph TD
    query_block[query_block#1] --> full_table_scan[Full Table Scan]
    full_table_scan --> emp_record_table[emp_record_table]
```

The emp_record_table is accessed via a Full Table Scan. The cost is 2.15, and 19 rows are returned. The execution plan is as follows:

```
graph TD
    query_block[query_block#1] --> full_table_scan[Full Table Scan]
    full_table_scan --> emp_record_table[emp_record_table]
```

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

```
2  
3  
4  
5 CREATE INDEX idx_first_name  
6 ON emp_record_table(FIRST_NAME(20));  
7 SELECT * FROM emp_record_table  
8 WHERE FIRST_NAME='Eric';
```

The query is executed, and the Explain Plan is displayed. The plan shows a non-unique key lookup for the emp_record_table. The cost is 0.35, and 1 row is returned. The execution plan is as follows:

```
graph TD
    query_block[query_block#1] --> non_unique_key_lookup[Non-Unique Key Lookup]
    non_unique_key_lookup --> emp_record_table[emp_record_table]
    emp_record_table --> idx_first_name[idx_first_name]
```

The emp_record_table is accessed via a Non-Unique Key Lookup. The cost is 0.35, and 1 row is returned. The execution plan is as follows:

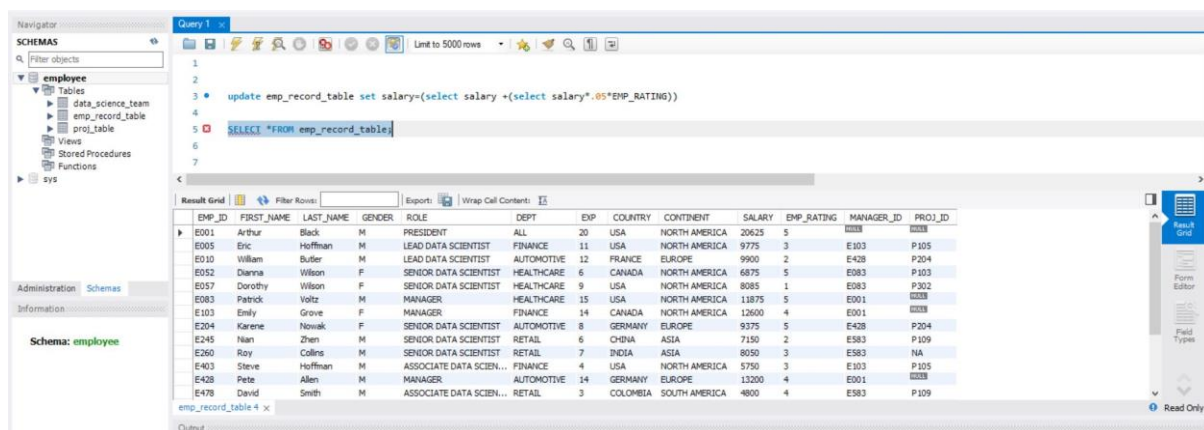
```
graph TD
    query_block[query_block#1] --> non_unique_key_lookup[Non-Unique Key Lookup]
    non_unique_key_lookup --> emp_record_table[emp_record_table]
    emp_record_table --> idx_first_name[idx_first_name]
```

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).

16A.

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

```
SELECT *FROM emp_record_table;
```



The screenshot shows a database management interface. On the left, a 'Navigator' pane displays a schema named 'employee' with tables like 'data_science_team', 'emp_record_table', and 'proj_table'. The main area is a 'Query Editor' with a 'Query 1' tab. It contains the following SQL code:

```
1
2
3 update emp_record_table set salary=(select salary +(select salary*.05*EMP_RATING))
4
5 SELECT *FROM emp_record_table;
6
7
```

Below the query editor is a 'Result Grid' showing the output of the query. It lists 15 employees with columns: EMP_ID, FIRST_NAME, LAST_NAME, GENDER, ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY, EMP_RATING, MANAGER_ID, and PROJ_ID. The data is as follows:

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	20625	5	E103	P105
E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	9775	3	E103	P105
E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9900	2	E428	P204
E052	Dianna	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	6875	5	E083	P103
E057	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA	8085	1	E083	P302
E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	11875	5	E001	E083
E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	12600	4	E001	E083
E204	Karen	Novak	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE	9375	5	E428	P204
E245	Nian	Zhen	M	SENIOR DATA SCIENTIST	RETAIL	6	CHINA	ASIA	7150	2	E583	P109
E260	Roy	Collins	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA	8050	3	E583	NA
E403	Steve	Hoffman	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA	5750	3	E103	P105
E428	Pete	Allen	M	MANAGER	AUTOMOTIVE	14	GERMANY	EUROPE	13200	4	E001	E083
E478	David	Smith	M	ASSOCIATE DATA SCIENTIST	RETAIL	3	COLUMBIA	SOUTH AMERICA	4800	4	E583	P109

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

17A.

```
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;
```


Navigator

Schemas

Filter objects

employee

- data_science_team
- emp_record_table
- proj_table
- Views
- employees_in_various
- Stored Procedures
- get_experience_details
- Functions
- samples
- sqlbasics
- sys

Administration Schemas

Information

No object selected

Query 1

Limit to 5000 rows

```
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
7
```

Result Grid

Filter Rows

Export

Wrap Cell Content

EMP_ID	FIRST_NAME	LAST_NAME	SALARY	COUNTRY	CONTINENT	AVG_salary_IN_COUNTRY	AVG_salary_IN_CONTINENT	COUNT_IN_COUNTRY	COUNT_IN_CONTINENT
E245	Nan	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	Kahna	Allen	3000	INDIA	ASIA	6166.6667	6250.0000	3	4
E010	William	Butler	9000	FRANCE	EUROPE	9000.0000	7950.0000	1	4
E204	Karene	Nowak	7500	GERMANY	EUROPE	7600.0000	7950.0000	3	4
E428	Pete	Allen	11000	GERMANY	EUROPE	7600.0000	7950.0000	3	4
E532	Claire	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
E005	Eric	Hoffman	8500	USA	NORTH AMERICA	9440.0000	8525.0000	5	8

Result 74 x

Read Only