Case Study on ScienceQtech Employee Performance Mapping

Problem scenario:

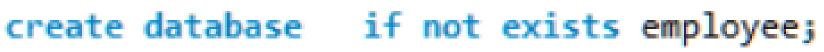
ScienceQtech is a startup that works in the Data Science field. ScienceQtech has worked on fraud detection, market basket, self-driving cars, supply chain, algorithmic early detection of lung cancer, customer sentiment, and the drug discovery field. With the annual appraisal cycle around the corner, 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.

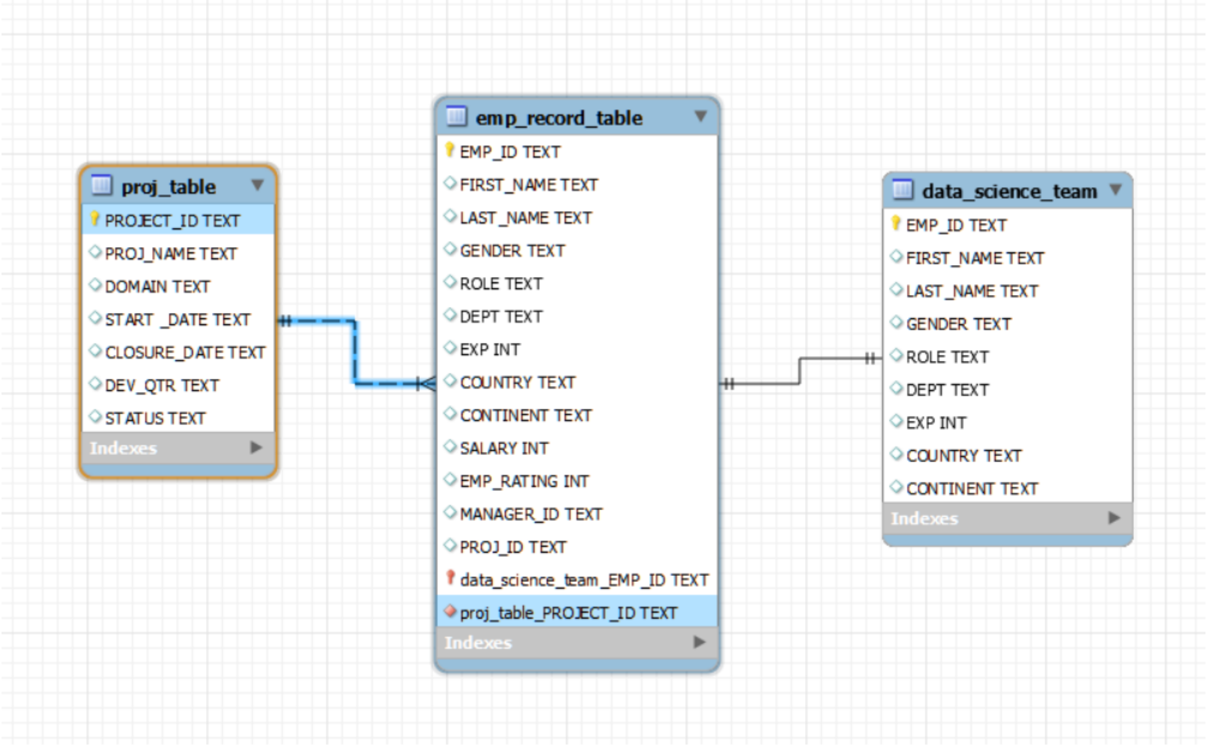
1.Create a database named *employee*, then import data_science_team.csv proj_table.csv and emp_record_table.csv into the employee database from the given resources.

QUERY:



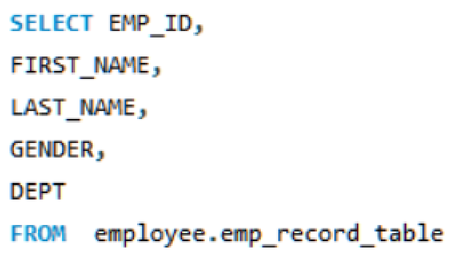
```
create database   if not exists employee;
```

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



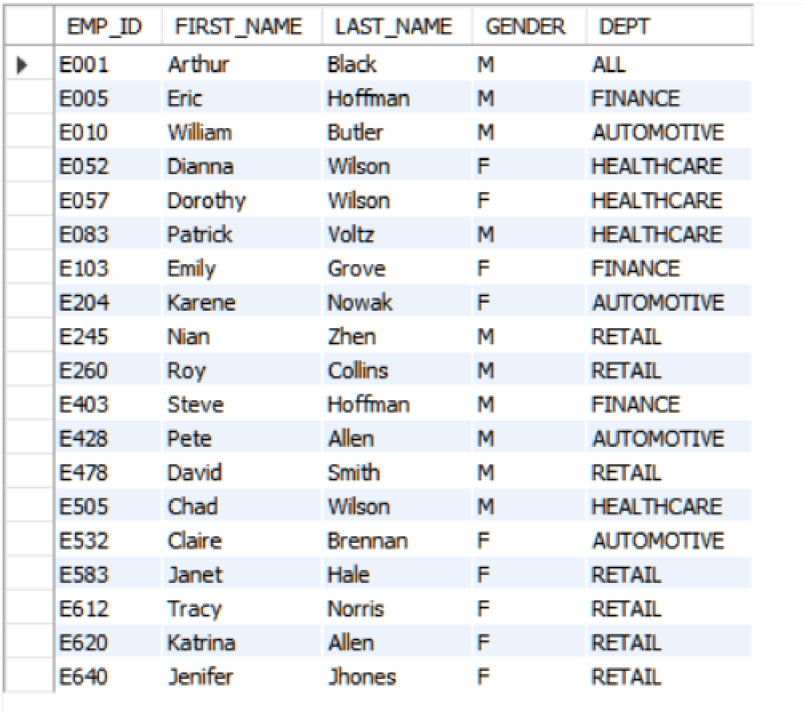
3. Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department

QUERY:



```
SELECT EMP_ID,
FIRST_NAME,
LAST_NAME,
GENDER,
DEPT
FROM  employee.emp_record_table
```

OUTPUT:



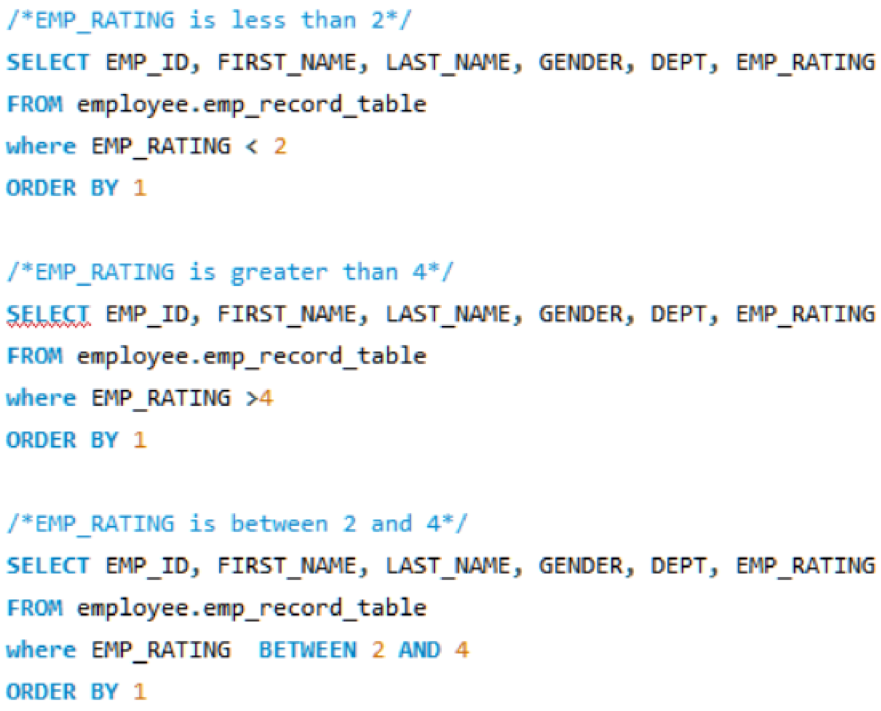
EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT
E001	Arthur	Black	M	ALL
E005	Eric	Hoffman	M	FINANCE
E010	William	Butler	M	AUTOMOTIVE
E052	Dianna	Wilson	F	HEALTHCARE
E057	Dorothy	Wilson	F	HEALTHCARE
E083	Patrick	Voltz	M	HEALTHCARE
E103	Emily	Grove	F	FINANCE
E204	Karene	Nowak	F	AUTOMOTIVE
E245	Nian	Zhen	M	RETAIL
E260	Roy	Collins	M	RETAIL
E403	Steve	Hoffman	M	FINANCE
E428	Pete	Allen	M	AUTOMOTIVE
E478	David	Smith	M	RETAIL
E505	Chad	Wilson	M	HEALTHCARE
E532	Claire	Brennan	F	AUTOMOTIVE
E583	Janet	Hale	F	RETAIL
E612	Tracy	Norris	F	RETAIL
E620	Katrina	Allen	F	RETAIL
E640	Jenifer	Jhones	F	RETAIL

This query helps in creating a list that combines employee details with their respective departments, providing a comprehensive view of the workforce distribution.

4. Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is:

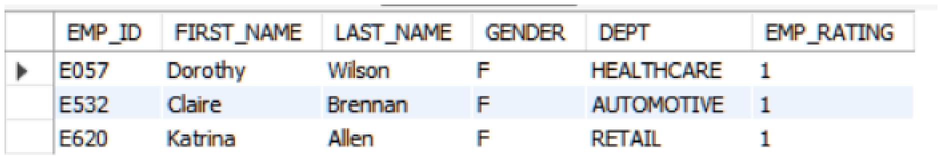
- **less than two**
- **greater than four**
- **between two and four**

QUERY:

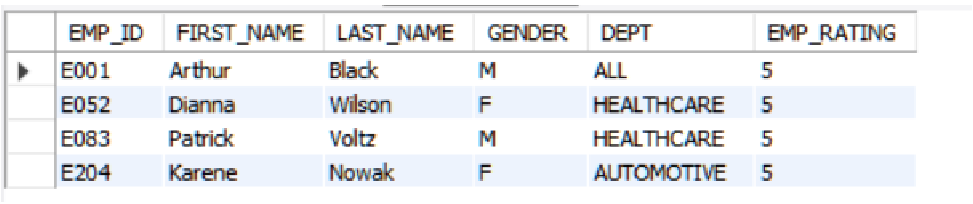


```
/*EMP_RATING is less than 2*/
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM employee.emp_record_table
where EMP_RATING < 2
ORDER BY 1

/*EMP_RATING is greater than 4*/
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM employee.emp_record_table
where EMP_RATING >4
ORDER BY 1

/*EMP_RATING is between 2 and 4*/
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM employee.emp_record_table
where EMP_RATING  BETWEEN 2 AND 4
ORDER BY 1
```

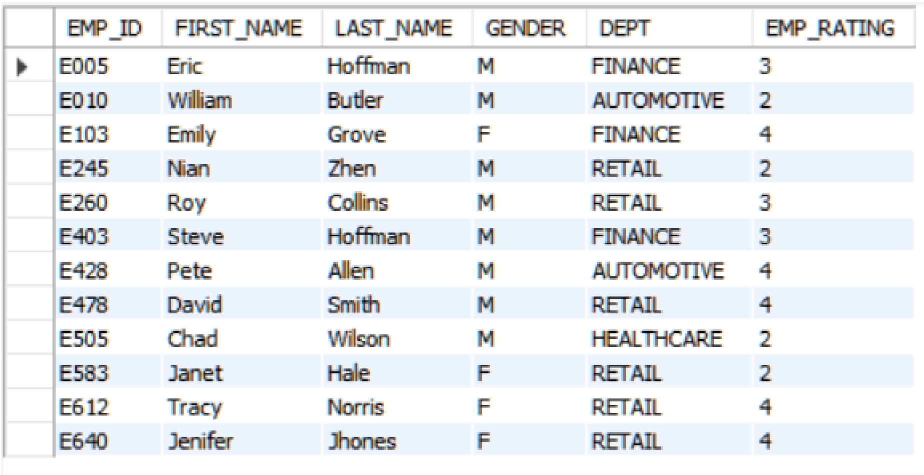
OUTPUT:



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



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

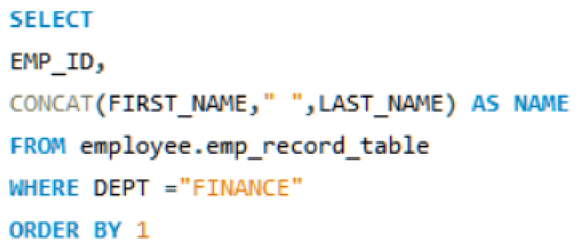


EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT	EMP_RATING
E005	Eric	Hoffman	M	FINANCE	3
E010	William	Butler	M	AUTOMOTIVE	2
E103	Emily	Grove	F	FINANCE	4
E245	Nian	Zhen	M	RETAIL	2
E260	Roy	Collins	M	RETAIL	3
E403	Steve	Hoffman	M	FINANCE	3
E428	Pete	Allen	M	AUTOMOTIVE	4
E478	David	Smith	M	RETAIL	4
E505	Chad	Wilson	M	HEALTHCARE	2
E583	Janet	Hale	F	RETAIL	2
E612	Tracy	Norris	F	RETAIL	4
E640	Jenifer	Jhones	F	RETAIL	4

This query categorizes employees based on their EMP_RATING, dividing them into three groups: less than two, greater than four, and between two and four. It aids in understanding the distribution of employee ratings and allows for targeted analysis of performance outliers.

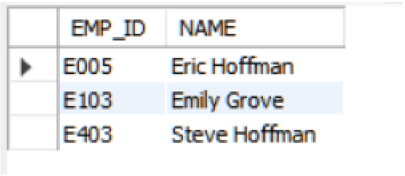
5.Write a query to concatenate the FIRST_NAME and the LAST_NAME of employees in the *Finance* department from the employee table and then give the resultant column alias as NAME.

QUERY:



```
SELECT
EMP_ID,
CONCAT(FIRST_NAME," ",LAST_NAME) AS NAME
FROM employee.emp_record_table
WHERE DEPT ="FINANCE"
ORDER BY 1
```

OUTPUT:

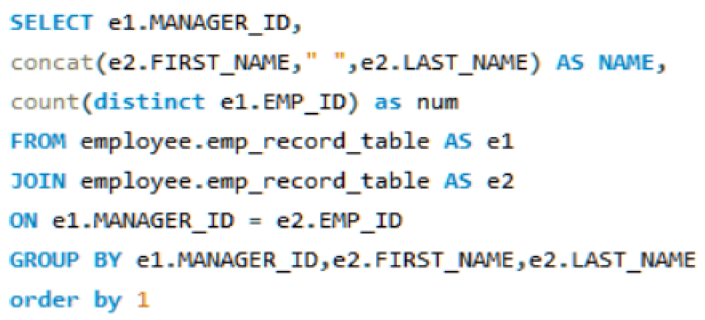


	EMP_ID	NAME
▶	E005	Eric Hoffman
	E103	Emily Grove
	E403	Steve Hoffman

The query combines the first and last names of employees in the Finance department, creating a unified column named "NAME."
It simplifies data presentation and can be useful for creating reports or user-friendly displays of employee names.

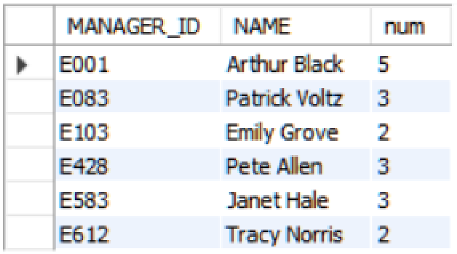
6.Write a query to list only those who have someone reporting to them. Also, show the number of reporters (including the President).

QUERY:



```
SELECT e1.MANAGER_ID,
concat(e2.FIRST_NAME," ",e2.LAST_NAME) AS NAME,
count(distinct e1.EMP_ID) as num
FROM employee.emp_record_table AS e1
JOIN employee.emp_record_table AS e2
ON e1.MANAGER_ID = e2.EMP_ID
GROUP BY e1.MANAGER_ID,e2.FIRST_NAME,e2.LAST_NAME
order by 1
```

OUTPUT:



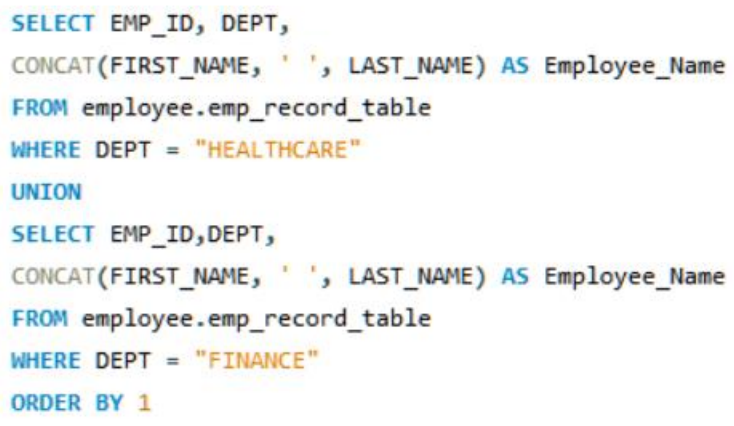
MANAGER_ID	NAME	num
E001	Arthur Black	5
E083	Patrick Voltz	3
E103	Emily Grove	2
E428	Pete Allen	3
E583	Janet Hale	3
E612	Tracy Norris	2

The query identifies employees(i.e Manager) who have at least one person reporting to them.

It provides insights into the organizational structure and hierarchy, highlighting employees with supervisory roles.

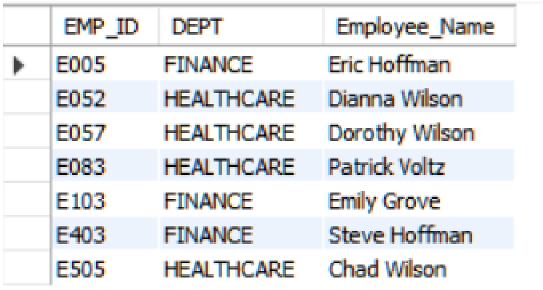
7. Write a query to list down all the employees from the healthcare and finance departments using union. Take data from the employee record table.

QUERY:



```
SELECT EMP_ID, DEPT,
CONCAT(FIRST_NAME, ' ', LAST_NAME) AS Employee_Name
FROM employee.emp_record_table
WHERE DEPT = "HEALTHCARE"
UNION
SELECT EMP_ID,DEPT,
CONCAT(FIRST_NAME, ' ', LAST_NAME) AS Employee_Name
FROM employee.emp_record_table
WHERE DEPT = "FINANCE"
ORDER BY 1
```

OUTPUT:



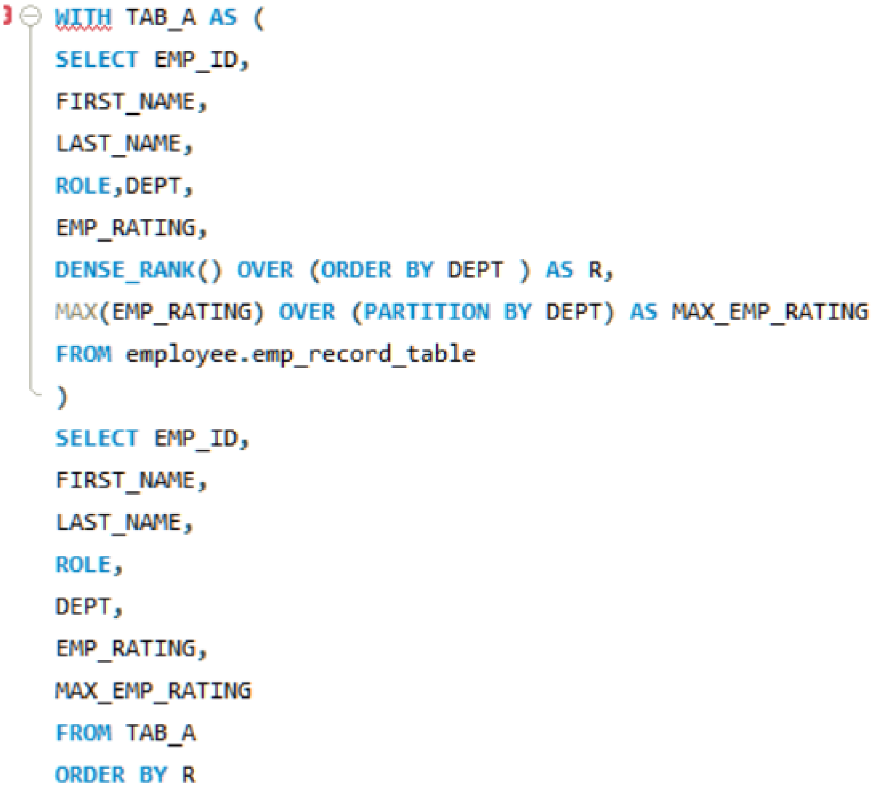
EMP_ID	DEPT	Employee_Name
E005	FINANCE	Eric Hoffman
E052	HEALTHCARE	Dianna Wilson
E057	HEALTHCARE	Dorothy Wilson
E083	HEALTHCARE	Patrick Voltz
E103	FINANCE	Emily Grove
E403	FINANCE	Steve Hoffman
E505	HEALTHCARE	Chad Wilson

This query combines data from two departments (Healthcare and Finance) using the UNION operator.

It facilitates the comparison of employees from different departments and can be helpful for cross-departmental analysis.

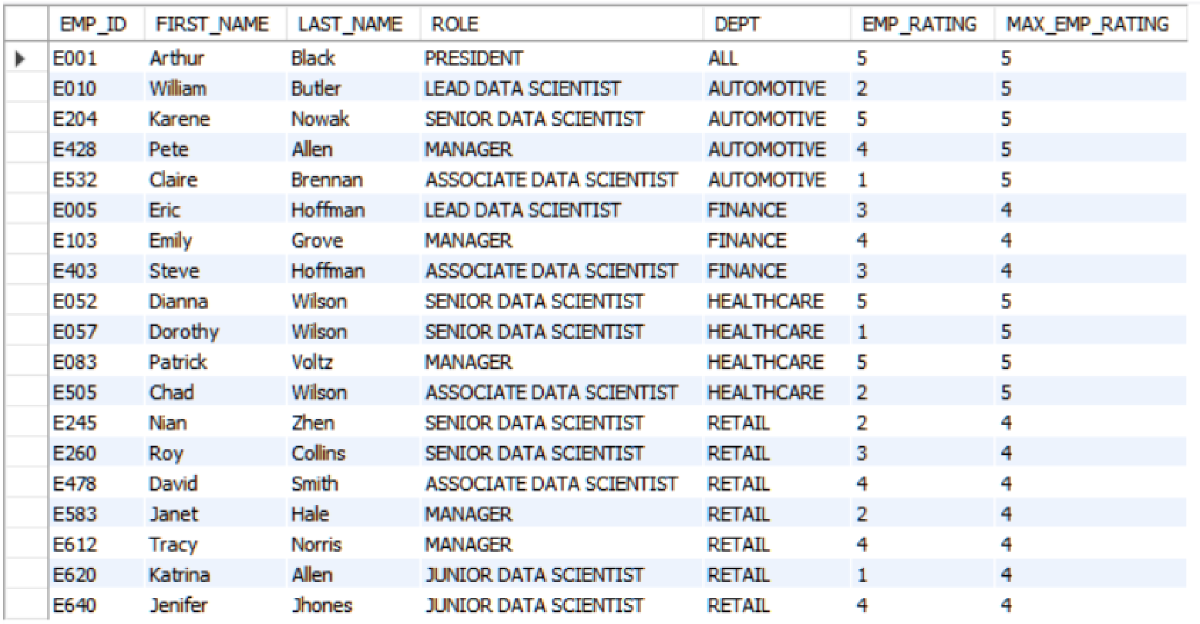
8. Write a query to list down employee details such as EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPARTMENT, and EMP_RATING grouped by dept. Also include the respective employee rating along with the max emp rating for the department.

QUERY:



```
WITH TAB_A AS (
SELECT EMP_ID,
FIRST_NAME,
LAST_NAME,
ROLE,DEPT,
EMP_RATING,
DENSE_RANK() OVER (ORDER BY DEPT ) AS R,
MAX(EMP_RATING) OVER (PARTITION BY DEPT) AS MAX_EMP_RATING
FROM employee.emp_record_table
)
SELECT EMP_ID,
FIRST_NAME,
LAST_NAME,
ROLE,
DEPT,
EMP_RATING,
MAX_EMP_RATING
FROM TAB_A
ORDER BY R
```

OUTPUT:



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

The query groups employee details by department, calculating the maximum EMP_RATING within each department.

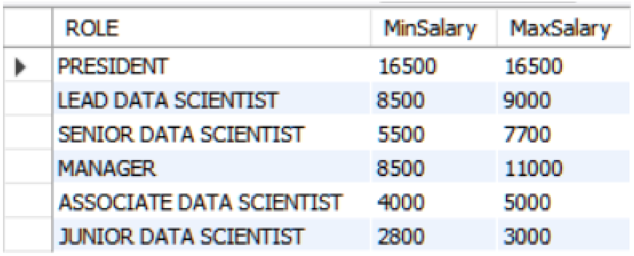
It provides an overview of employee ratings per department and identifies the highest rating in each department.

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.

QUERY:

```
SELECT
  ROLE,
  MIN(SALARY) AS MinSalary,
  MAX(SALARY) AS MaxSalary
FROM
  employee.emp_record_table
GROUP BY ROLE
```

OUTPUT:

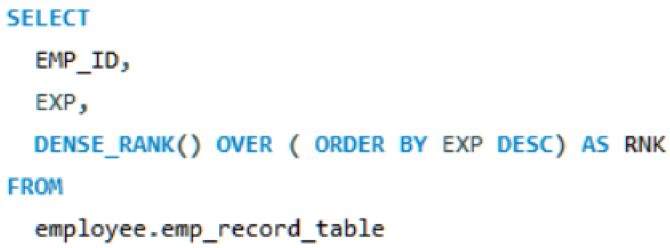


	ROLE	MinSalary	MaxSalary
▶	PRESIDENT	16500	16500
	LEAD DATA SCIENTIST	8500	9000
	SENIOR DATA SCIENTIST	5500	7700
	MANAGER	8500	11000
	ASSOCIATE DATA SCIENTIST	4000	5000
	JUNIOR DATA SCIENTIST	2800	3000

This query helps in understanding the salary range for different job roles, aiding in salary benchmarking and analysis.

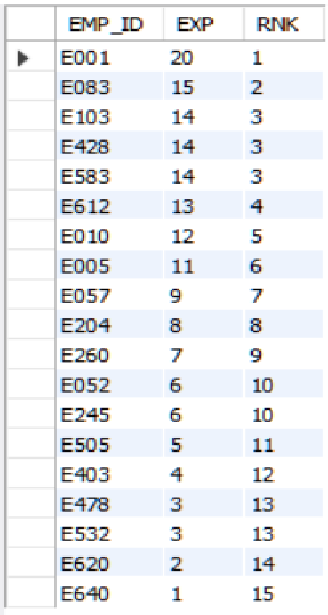
10. Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.

QUERY:



```
SELECT
  EMP_ID,
  EXP,
  DENSE_RANK() OVER ( ORDER BY EXP DESC) AS RNK
FROM
  employee.emp_record_table
```

OUTPUT:

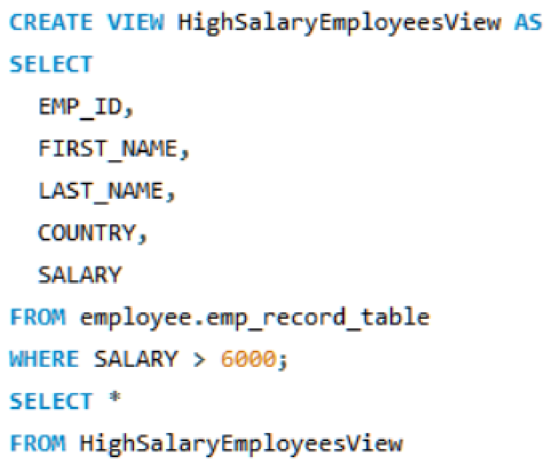


	EMP_ID	EXP	RNK
▶	E001	20	1
	E083	15	2
	E103	14	3
	E428	14	3
	E583	14	3
	E612	13	4
	E010	12	5
	E005	11	6
	E057	9	7
	E204	8	8
	E260	7	9
	E052	6	10
	E245	6	10
	E505	5	11
	E403	4	12
	E478	3	13
	E532	3	13
	E620	2	14
	E640	1	15

This query assists in visualizing the distribution of employee experience levels and categorizes them into specific roles.

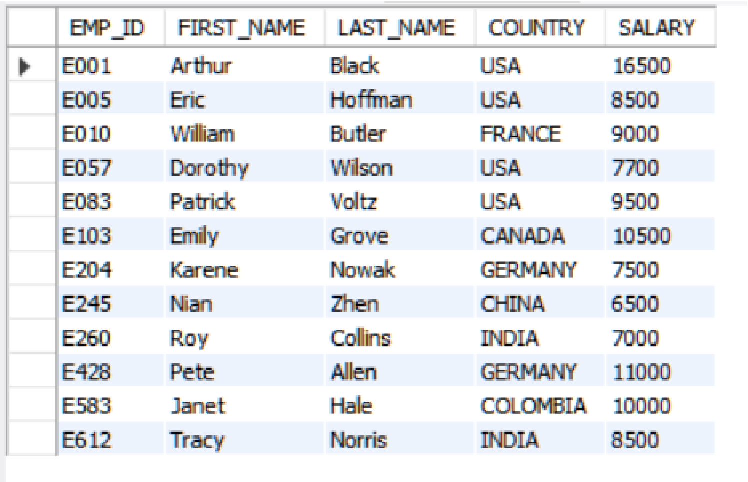
11.Write a query to create a view that displays employees in various countries whose salary is more than six thousand. Take data from the employee record table.

QUERY:



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

OUTPUT:



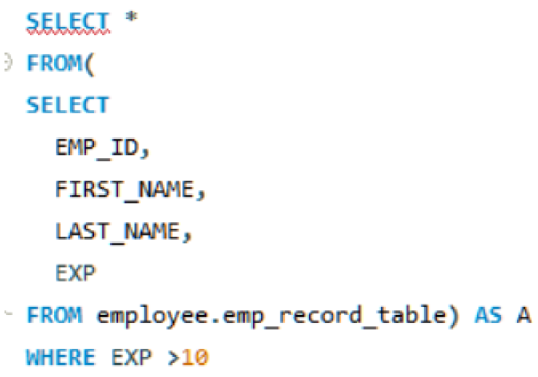
	EMP_ID	FIRST_NAME	LAST_NAME	COUNTRY	SALARY
▶	E001	Arthur	Black	USA	16500
	E005	Eric	Hoffman	USA	8500
	E010	William	Butler	FRANCE	9000
	E057	Dorothy	Wilson	USA	7700
	E083	Patrick	Voltz	USA	9500
	E103	Emily	Grove	CANADA	10500
	E204	Karene	Nowak	GERMANY	7500
	E245	Nian	Zhen	CHINA	6500
	E260	Roy	Collins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

The query creates a view that displays employees in various countries with salaries exceeding six thousand.

It offers a convenient way to access and analyze information about high-salary employees across different countries.

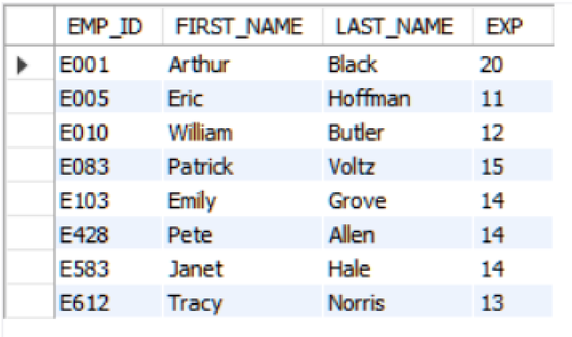
12. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

QUERY:



```
SELECT *
FROM(
SELECT
   EMP_ID,
   FIRST_NAME,
   LAST_NAME,
   EXP
FROM employee.emp_record_table) AS A
WHERE EXP >10
```

OUTPUT:

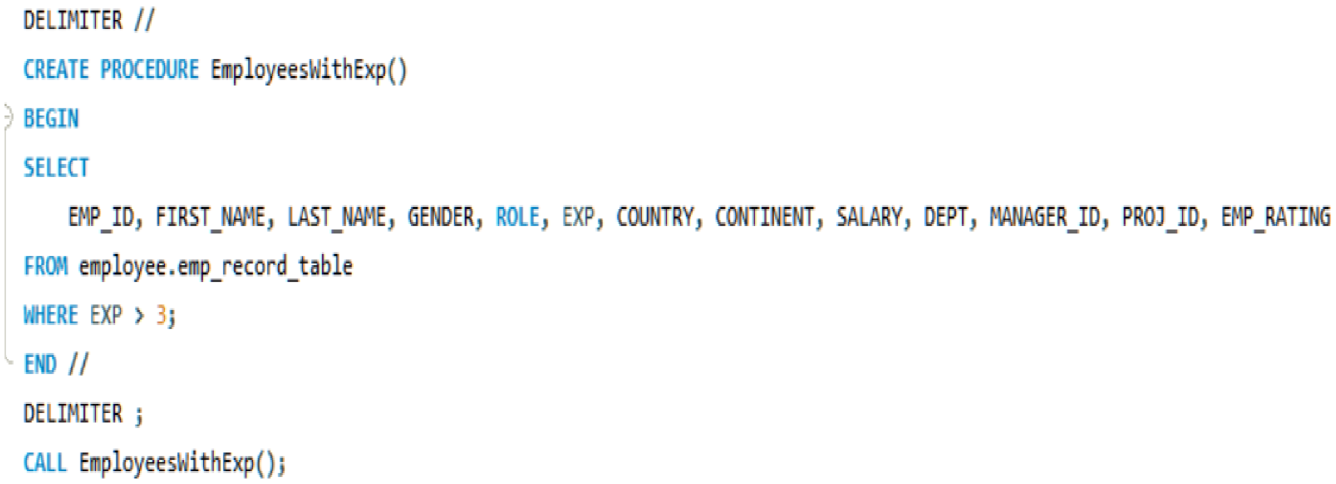


EMP_ID	FIRST_NAME	LAST_NAME	EXP
E001	Arthur	Black	20
E005	Eric	Hoffman	11
E010	William	Butler	12
E083	Patrick	Voltz	15
E103	Emily	Grove	14
E428	Pete	Allen	14
E583	Janet	Hale	14
E612	Tracy	Norris	13

This nested query identifies employees with more than ten years of experience.
It helps in isolating long-term employees for specific analysis or recognition.

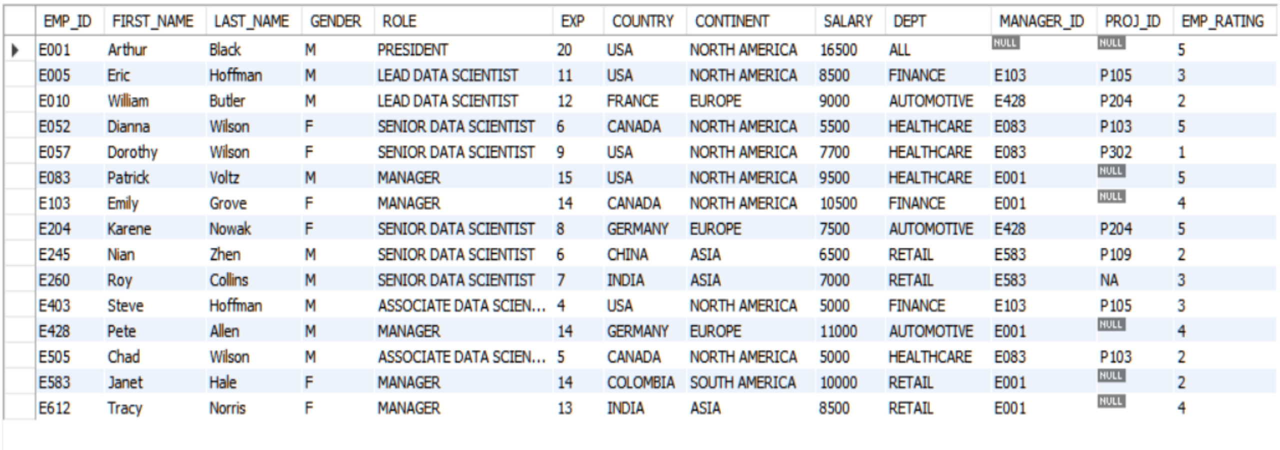
13. Write a query to create a stored procedure to retrieve the details of the employees whose experience is more than three years. Take data from the employee record table.

QUERY:



```
DELIMITER //
CREATE PROCEDURE EmployeesWithExp()
BEGIN
SELECT
    EMP_ID, FIRST_NAME, LAST_NAME, GENDER, ROLE, EXP, COUNTRY, CONTINENT, SALARY, DEPT, MANAGER_ID, PROJ_ID, EMP_RATING
FROM employee.emp_record_table
WHERE EXP > 3;
END //
DELIMITER ;
CALL EmployeesWithExp();
```

OUTPUT;



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

This stored procedure retrieves details of employees with more than three years of experience.

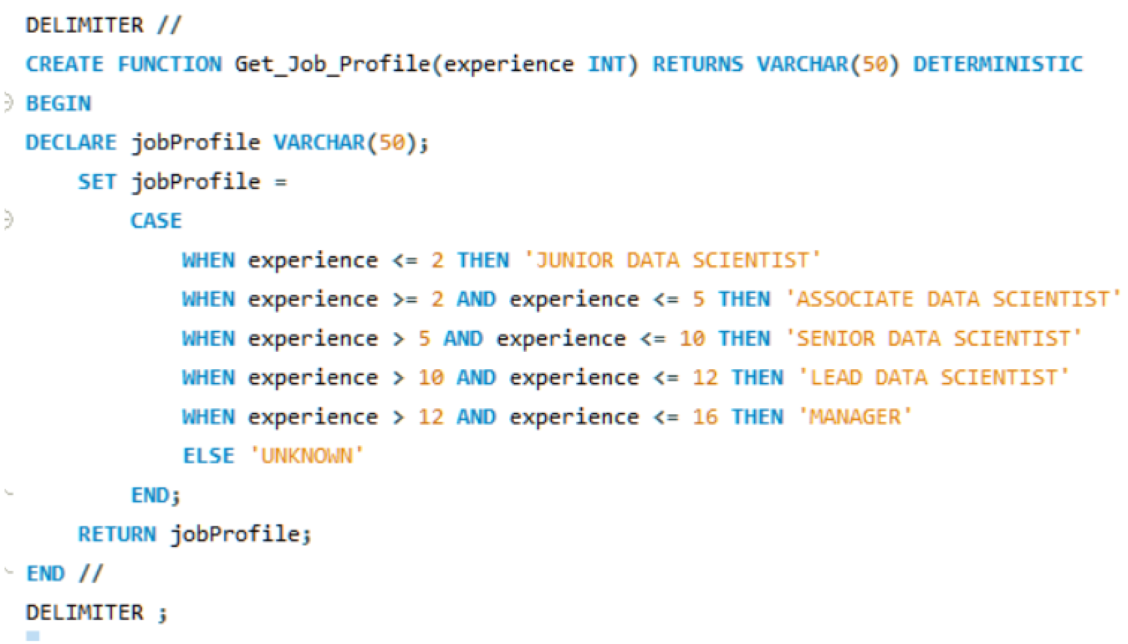
It provides a reusable and efficient way to obtain specific employee information.

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'.

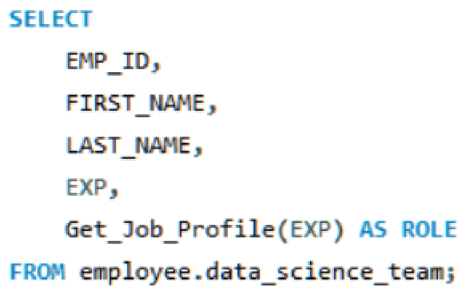
QUERY:

(i)Stored functions



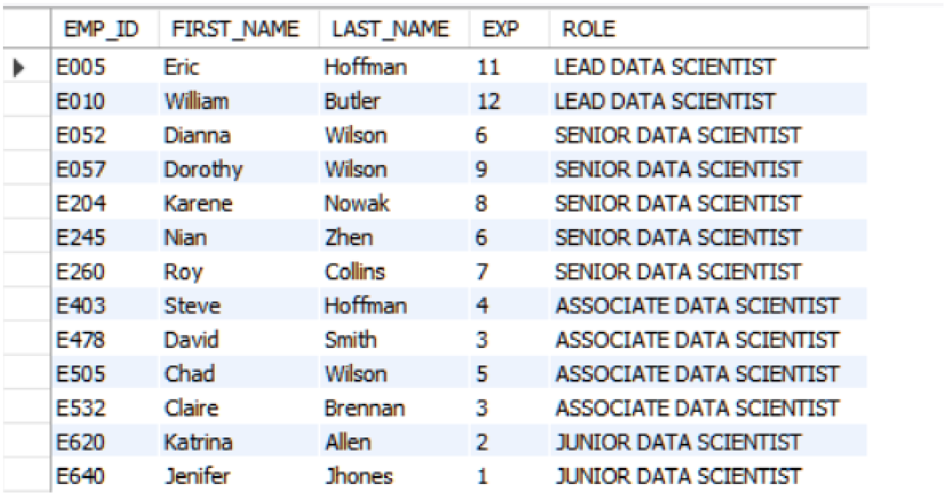
```
DELIMITER //
CREATE FUNCTION Get_Job_Profile(experience INT) RETURNS VARCHAR(50) DETERMINISTIC
BEGIN
DECLARE jobProfile VARCHAR(50);
    SET jobProfile =
        CASE
            WHEN experience <= 2 THEN 'JUNIOR DATA SCIENTIST'
            WHEN experience >= 2 AND experience <= 5 THEN 'ASSOCIATE DATA SCIENTIST'
            WHEN experience > 5 AND experience <= 10 THEN 'SENIOR DATA SCIENTIST'
            WHEN experience > 10 AND experience <= 12 THEN 'LEAD DATA SCIENTIST'
            WHEN experience > 12 AND experience <= 16 THEN 'MANAGER'
            ELSE 'UNKNOWN'
        END;
    RETURN jobProfile;
END //
DELIMITER ;
```

(ii)RETRIEVING DATA



```
SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    EXP,
    Get_Job_Profile(EXP) AS ROLE
FROM employee.data_science_team;
```

OUTPUT:



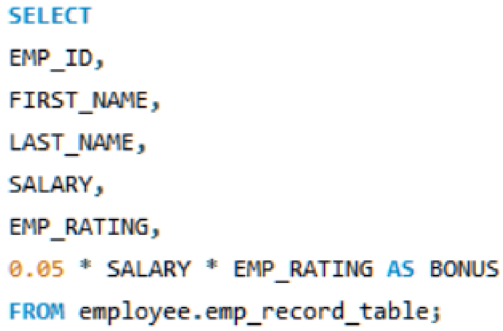
	EMP_ID	FIRST_NAME	LAST_NAME	EXP	ROLE
▶	E005	Eric	Hoffman	11	LEAD DATA SCIENTIST
	E010	William	Butler	12	LEAD DATA SCIENTIST
	E052	Dianna	Wilson	6	SENIOR DATA SCIENTIST
	E057	Dorothy	Wilson	9	SENIOR DATA SCIENTIST
	E204	Karene	Nowak	8	SENIOR DATA SCIENTIST
	E245	Nian	Zhen	6	SENIOR DATA SCIENTIST
	E260	Roy	Collins	7	SENIOR DATA SCIENTIST
	E403	Steve	Hoffman	4	ASSOCIATE DATA SCIENTIST
	E478	David	Smith	3	ASSOCIATE DATA SCIENTIST
	E505	Chad	Wilson	5	ASSOCIATE DATA SCIENTIST
	E532	Claire	Brennan	3	ASSOCIATE DATA SCIENTIST
	E620	Katrina	Allen	2	JUNIOR DATA SCIENTIST
	E640	Jenifer	Jhones	1	JUNIOR DATA SCIENTIST

The query utilizes stored functions to assign job profiles in the Data Science team based on experience.

It automates the process of assigning roles, ensuring alignment with the organization's set standards

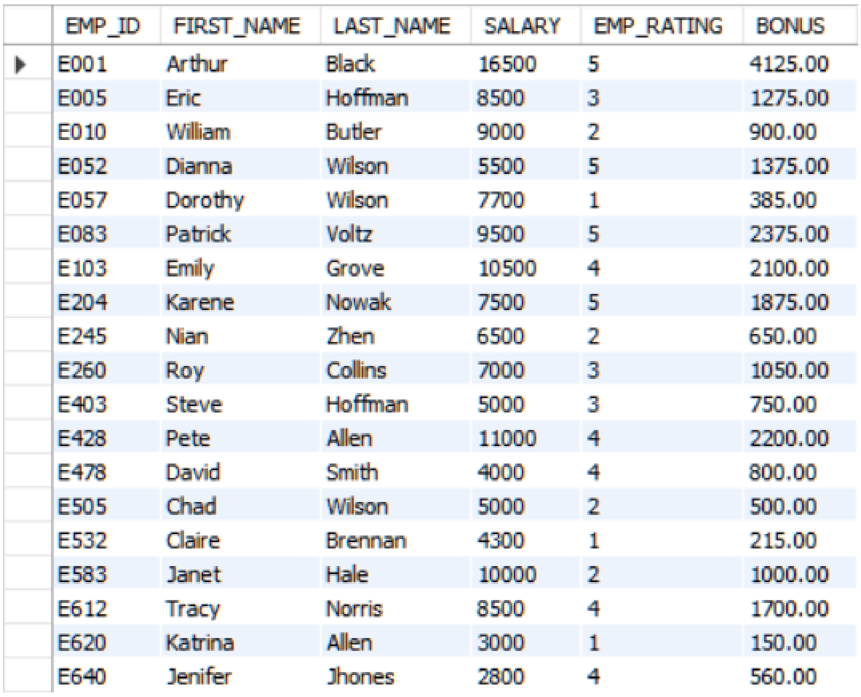
15. Write a query to calculate the bonus for all the employees, based on their ratings and salaries (Use the formula: 5% of salary * employee rating).

QUERY:



```
SELECT
EMP_ID,
FIRST_NAME,
LAST_NAME,
SALARY,
EMP_RATING,
0.05 * SALARY * EMP_RATING AS BONUS
FROM employee.emp_record_table;
```

OUTPUT:

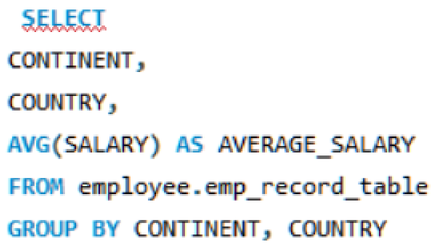


EMP_ID	FIRST_NAME	LAST_NAME	SALARY	EMP_RATING	BONUS
E001	Arthur	Black	16500	5	4125.00
E005	Eric	Hoffman	8500	3	1275.00
E010	William	Butler	9000	2	900.00
E052	Dianna	Wilson	5500	5	1375.00
E057	Dorothy	Wilson	7700	1	385.00
E083	Patrick	Voltz	9500	5	2375.00
E103	Emily	Grove	10500	4	2100.00
E204	Karene	Nowak	7500	5	1875.00
E245	Nian	Zhen	6500	2	650.00
E260	Roy	Collins	7000	3	1050.00
E403	Steve	Hoffman	5000	3	750.00
E428	Pete	Allen	11000	4	2200.00
E478	David	Smith	4000	4	800.00
E505	Chad	Wilson	5000	2	500.00
E532	Claire	Brennan	4300	1	215.00
E583	Janet	Hale	10000	2	1000.00
E612	Tracy	Norris	8500	4	1700.00
E620	Katrina	Allen	3000	1	150.00
E640	Jenifer	Jhones	2800	4	560.00

This query provides a method for quantifying performance-related bonuses, linking ratings and compensation.

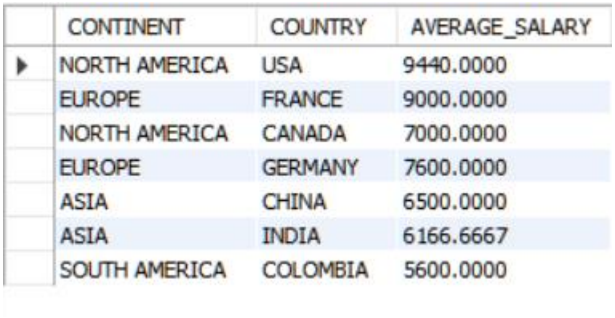
16. Write a query to calculate the average salary distribution based on the continent and country. Take data from the employee record table.

QUERY:



```
 SELECT
CONTINENT,
COUNTRY,
AVG(SALARY) AS AVERAGE_SALARY
FROM employee.emp_record_table
GROUP BY CONTINENT, COUNTRY
```

OUTPUT:



CONTINENT	COUNTRY	AVERAGE_SALARY
NORTH AMERICA	USA	9440.0000
EUROPE	FRANCE	9000.0000
NORTH AMERICA	CANADA	7000.0000
EUROPE	GERMANY	7600.0000
ASIA	CHINA	6500.0000
ASIA	INDIA	6166.6667
SOUTH AMERICA	COLOMBIA	5600.0000

This query offers insights into salary patterns across different geographical regions, aiding in compensation analysis and planning.

These queries collectively provide a comprehensive toolkit for analyzing and understanding various aspects of employee data, performance, and organizational structure.