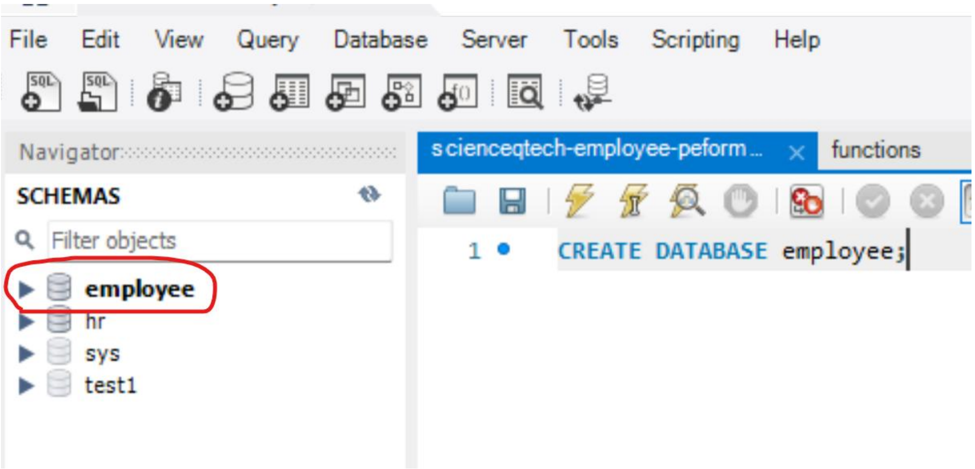
ScienceQtech DB Implementation Procedures

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.

Database Creation

>> Use 'CREATE DATABASE' statement to create 'employee' database.



```
CREATE DATABASE employee;
```

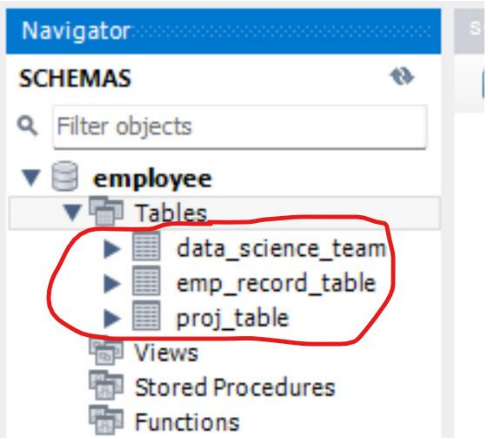
Importing Process

>> Right click 'Tables' under the employee database.

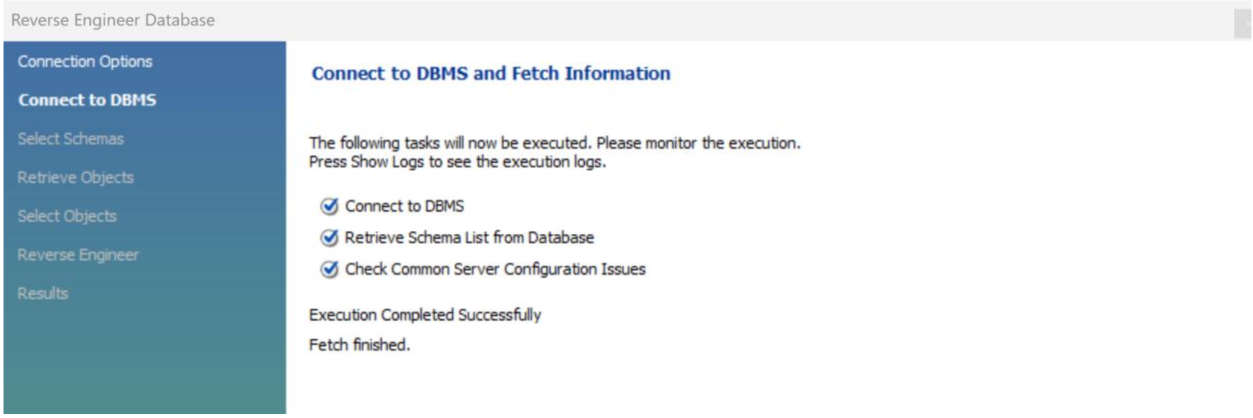
>> Click 'Table Data Import Wizard'

>> Select import file path(s)

>> Load the three datasets into the employee db.



2. **Create an ER diagram for the given employee database.**
   >> Go to ‘Database’ Tab
   >> Click on ‘Reverse Engineer
   >> Confirm instance configuration.
   >> Click ‘Next’
   >> Ensure that you able to connect to the DBMS.
   >> Click ‘Next’ again.



Connect to DBMS and Fetch Information

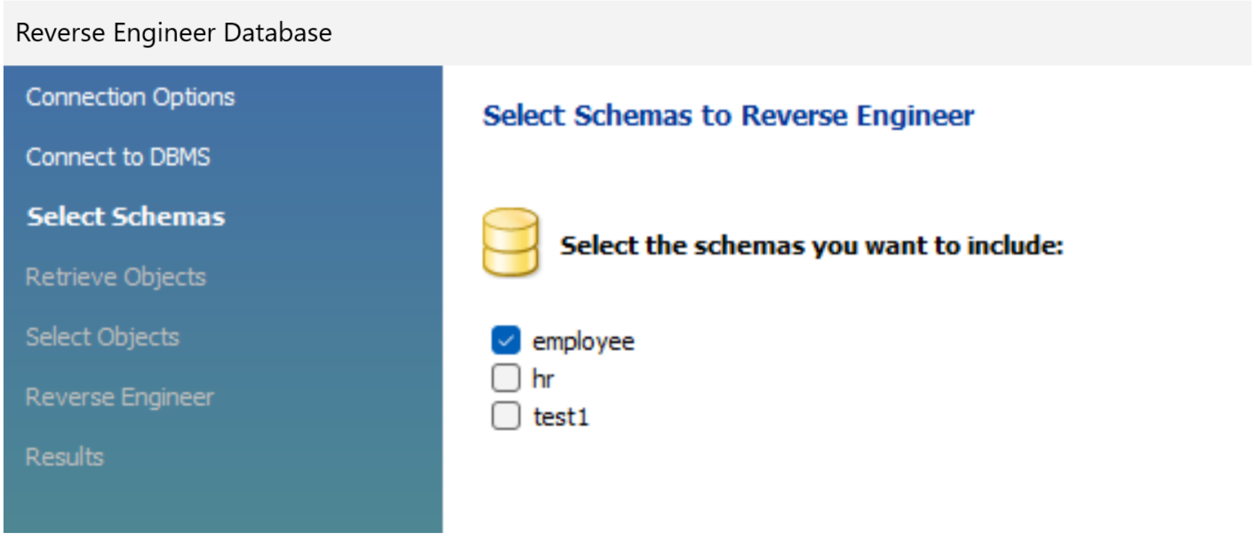
The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.

☑ Connect to DBMS
☑ Retrieve Schema List from Database
☑ Check Common Server Configuration Issues

Execution Completed Successfully

Fetch finished.

   >> Select ‘employee’ schema.
   >> Click ‘Next’



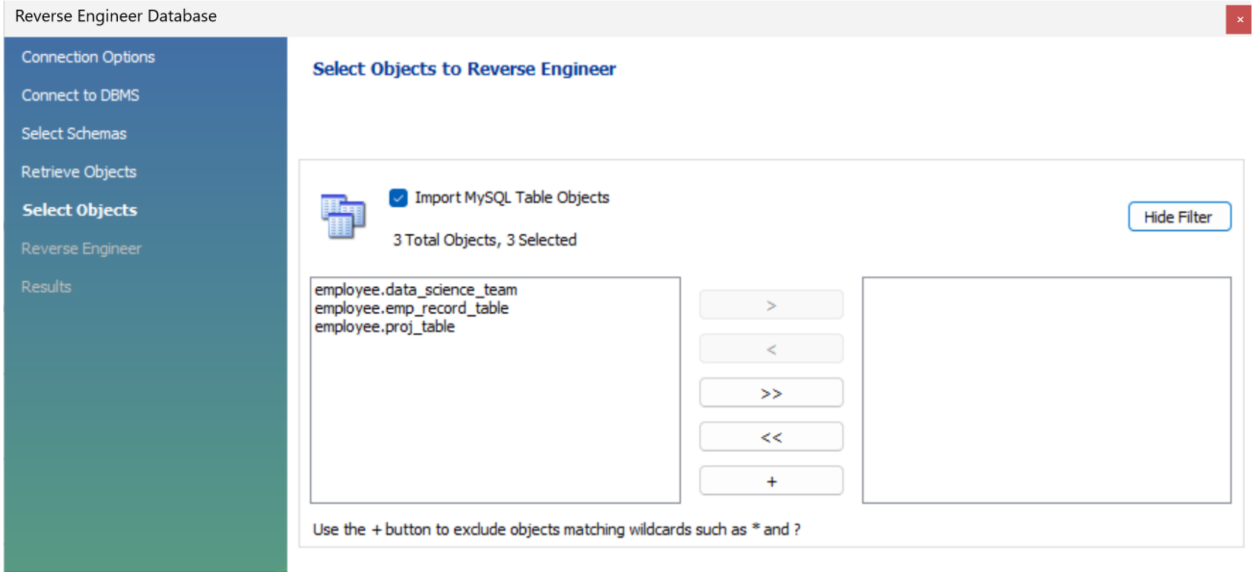
Reverse Engineer Database

Select Schemas to Reverse Engineer

Select the schemas you want to include:

☑ employee
□ hr
□ test1

>> Confirm that all three tables have been added to the model.
>> Click 'Execute'.



Reverse Engineer Database

Select Objects to Reverse Engineer

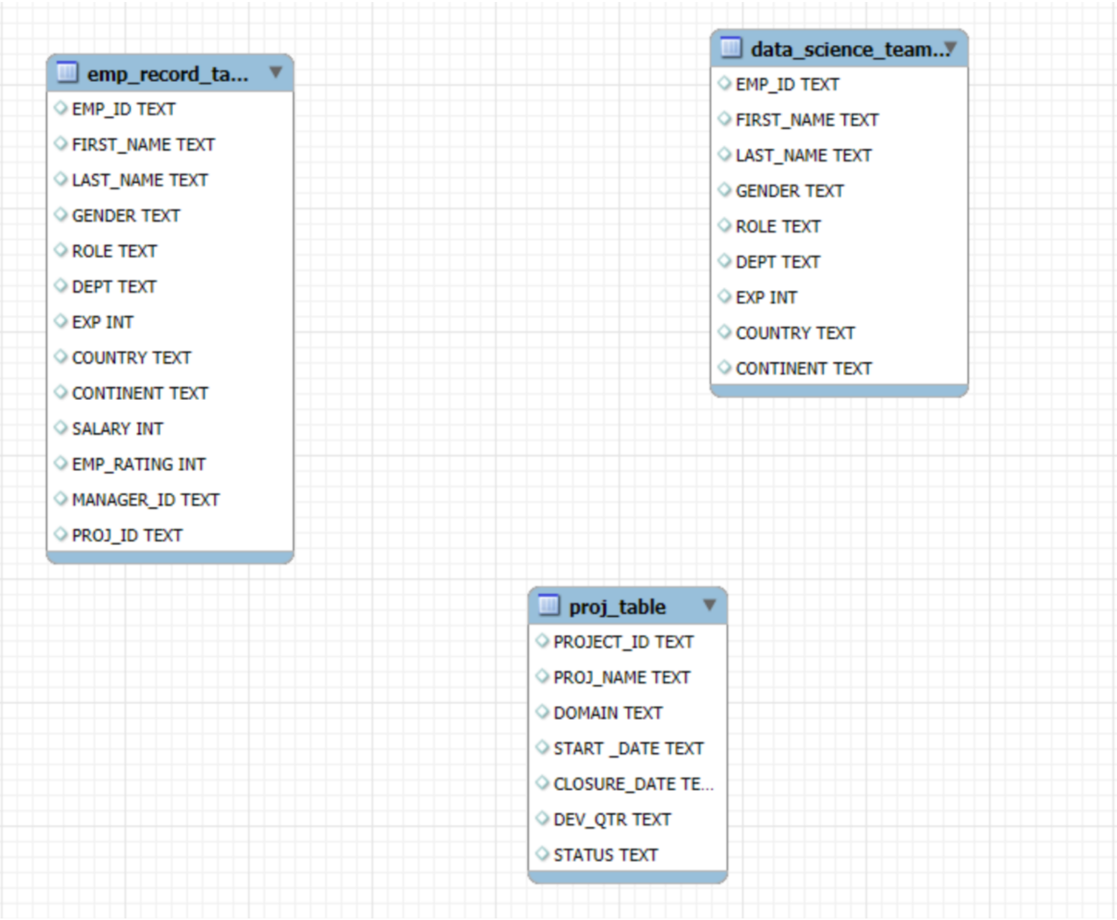
☑ Import MySQL Table Objects

3 Total Objects, 3 Selected

employee.data_science_team
employee.emp_record_table
employee.proj_table

Use the + button to exclude objects matching wildcards such as * and ?

>> ER diagram has been successfully created.



emp_record_ta...

EMP_ID TEXT
FIRST_NAME TEXT
LAST_NAME TEXT
GENDER TEXT
ROLE TEXT
DEPT TEXT
EXP INT
COUNTRY TEXT
CONTINENT TEXT
SALARY INT
EMP_RATING INT
MANAGER_ID TEXT
PROJ_ID TEXT

data_science_team...

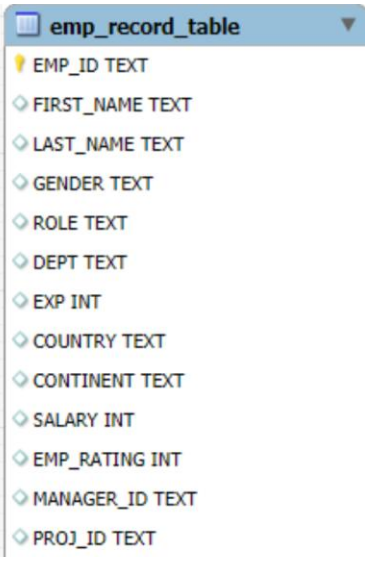
EMP_ID TEXT
FIRST_NAME TEXT
LAST_NAME TEXT
GENDER TEXT
ROLE TEXT
DEPT TEXT
EXP INT
COUNTRY TEXT
CONTINENT TEXT

proj_table

PROJECT_ID TEXT
PROJ_NAME TEXT
DOMAIN TEXT
START _DATE TEXT
CLOSURE_DATE TE...
DEV_QTR TEXT
STATUS TEXT

>> Next, assign your primary keys for the three tables

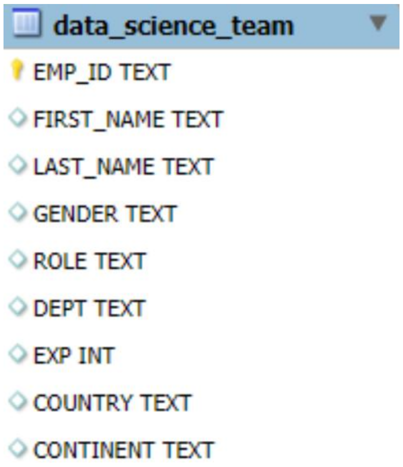
>> The employee record table will use the ‘emp_id’ attribute as the primary key.



emp_record_table

EMP_ID TEXT
FIRST_NAME TEXT
LAST_NAME TEXT
GENDER TEXT
ROLE TEXT
DEPT TEXT
EXP INT
COUNTRY TEXT
CONTINENT TEXT
SALARY INT
EMP_RATING INT
MANAGER_ID TEXT
PROJ_ID TEXT

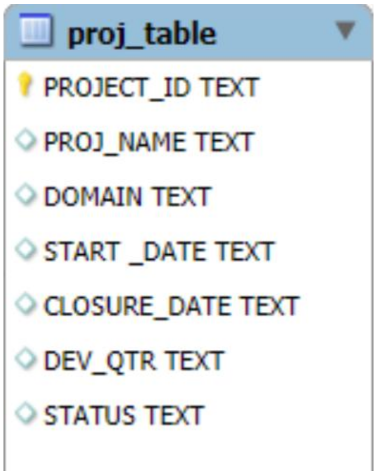
>> the ‘data_science_team’ table will use the ‘emp_id’ attribute as its foreign key.



data_science_team

EMP_ID TEXT
FIRST_NAME TEXT
LAST_NAME TEXT
GENDER TEXT
ROLE TEXT
DEPT TEXT
EXP INT
COUNTRY TEXT
CONTINENT TEXT

>> Lastly, the project table will use ‘project_id’ as its primary key.

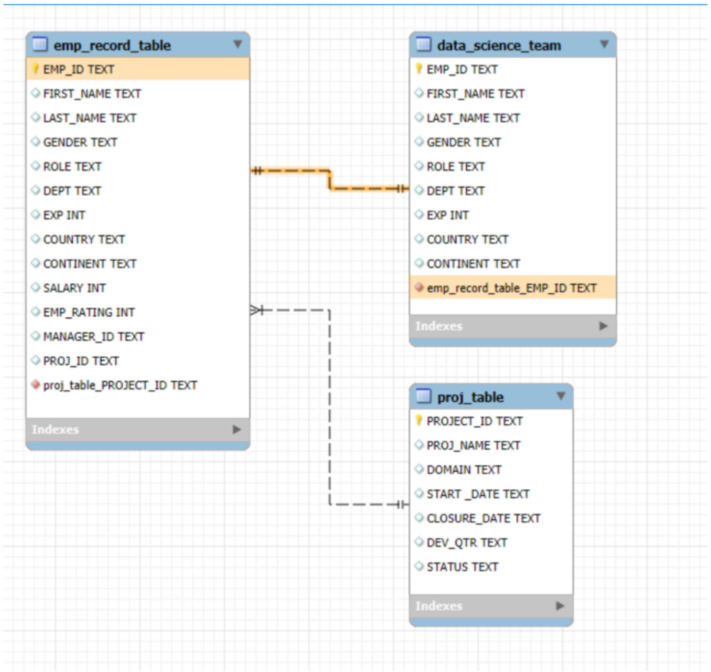


proj_table

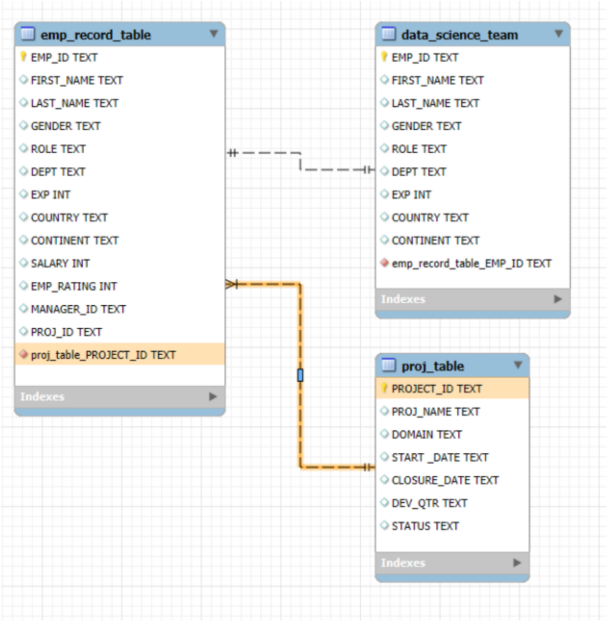
PROJECT_ID TEXT
PROJ_NAME TEXT
DOMAIN TEXT
START _DATE TEXT
CLOSURE_DATE TEXT
DEV_QTR TEXT
STATUS TEXT

>> Once the unique attributes have been declared, the next step is to establish cardinality between the three tables.

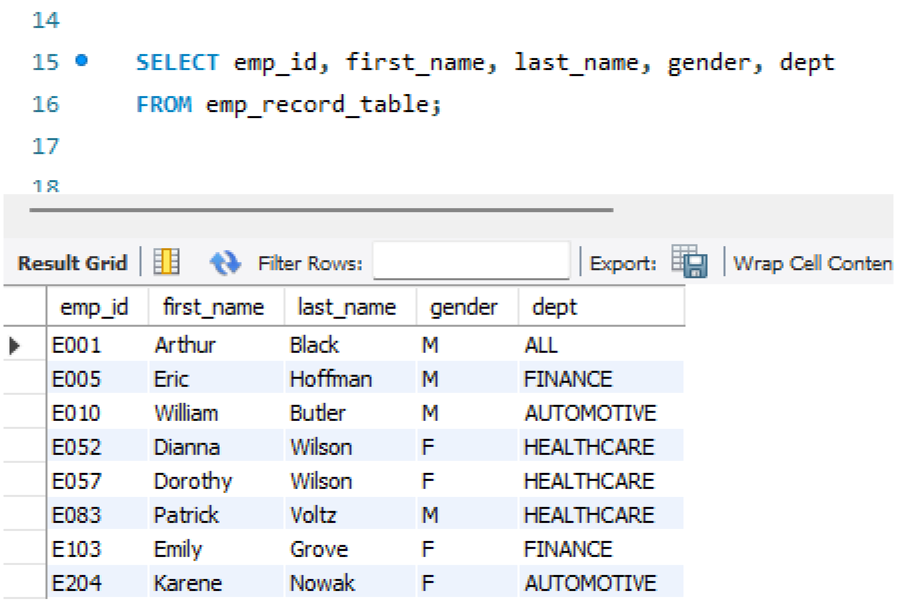
>> The employee record table shares a one-to-one relationship with the ‘data_science_team’ table via the ‘emp_id’ attribute.



>> The employee record table also shares a many to one relationship with the project table via the ‘project_id’ column.



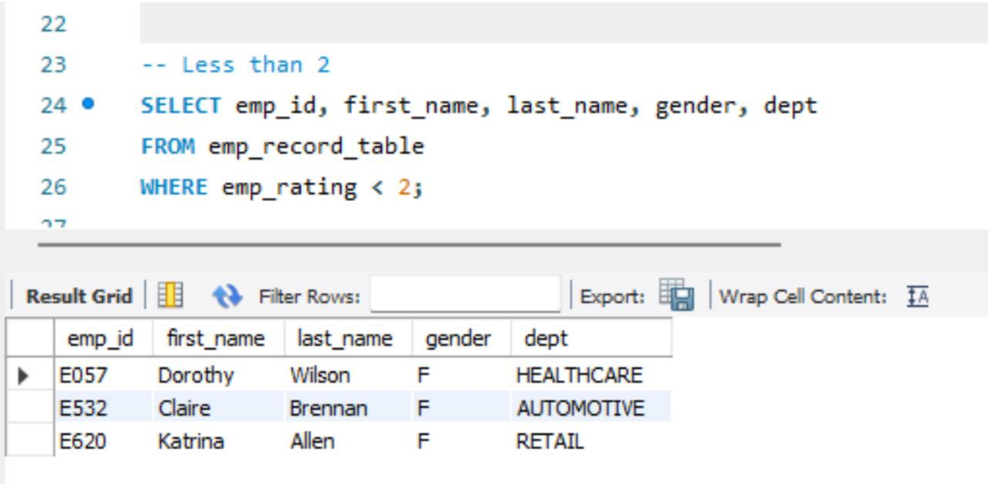
3. Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department.



```sql
SELECT emp_id, first_name, last_name, gender, dept
FROM emp_record_table;
```

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

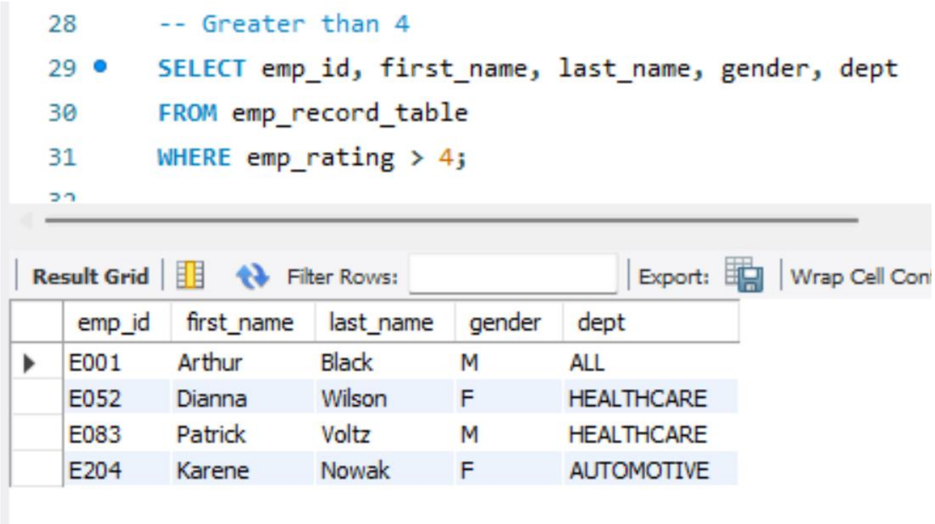
4. Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is:
    a. less than two



```sql
-- Less than 2
SELECT emp_id, first_name, last_name, gender, dept
FROM emp_record_table
WHERE emp_rating < 2;
```

emp_id	first_name	last_name	gender	dept
E057	Dorothy	Wilson	F	HEALTHCARE
E532	Claire	Brennan	F	AUTOMOTIVE
E620	Katrina	Allen	F	RETAIL

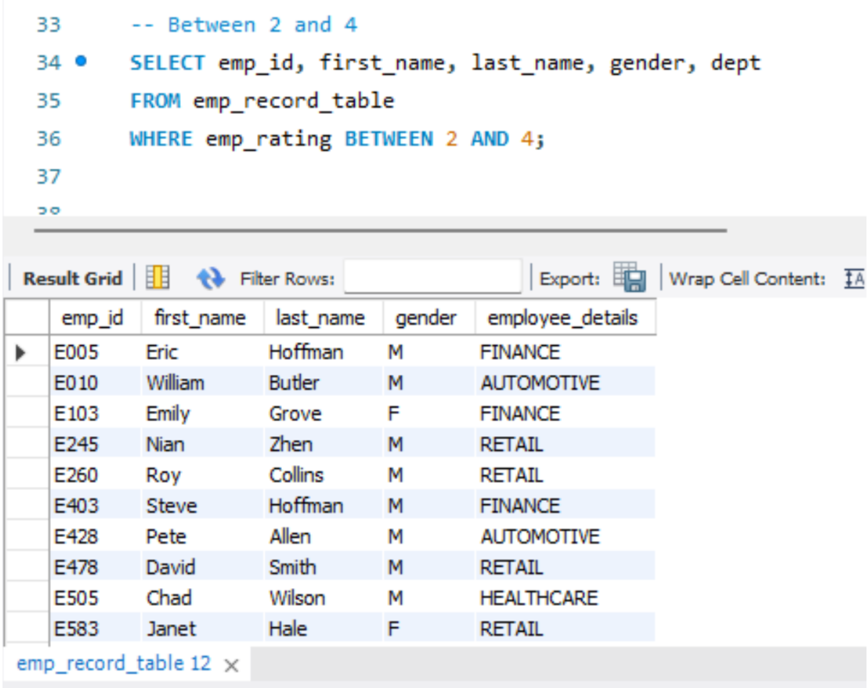
b. greater than four



```
-- Greater than 4
SELECT emp_id, first_name, last_name, gender, dept
FROM emp_record_table
WHERE emp_rating > 4;
```

emp_id	first_name	last_name	gender	dept
E001	Arthur	Black	M	ALL
E052	Dianna	Wilson	F	HEALTHCARE
E083	Patrick	Voltz	M	HEALTHCARE
E204	Karene	Nowak	F	AUTOMOTIVE

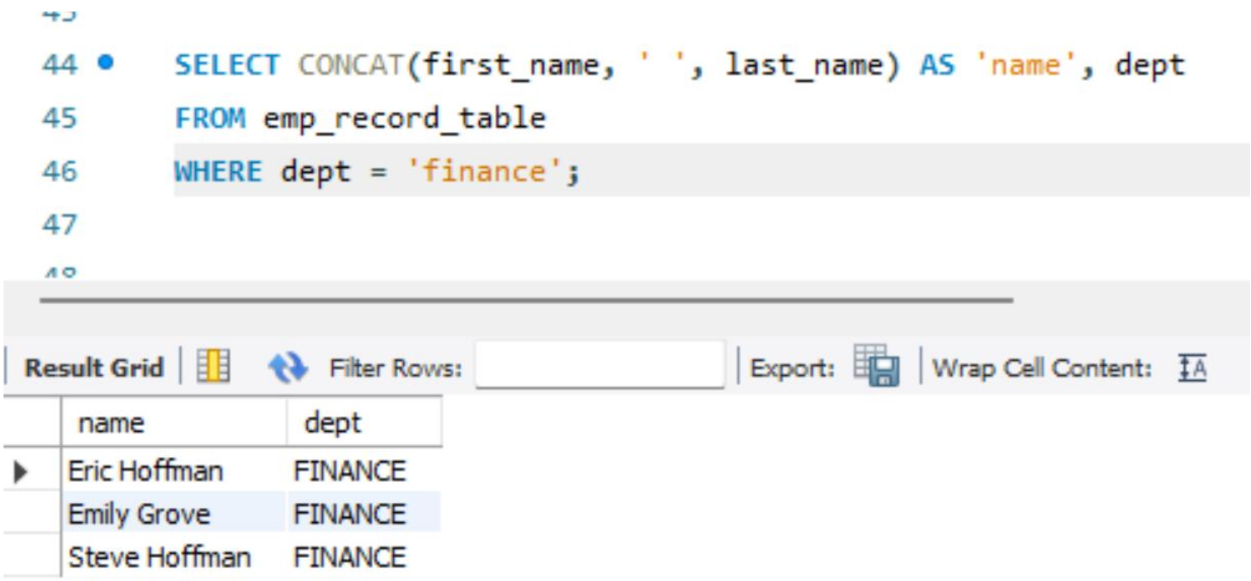
c. between two and four



```
-- Between 2 and 4
SELECT emp_id, first_name, last_name, gender, dept
FROM emp_record_table
WHERE emp_rating BETWEEN 2 AND 4;
```

emp_id	first_name	last_name	gender	employee_details
E005	Eric	Hoffman	M	FINANCE
E010	William	Butler	M	AUTOMOTIVE
E103	Emily	Grove	F	FINANCE
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
E583	Janet	Hale	F	RETAIL

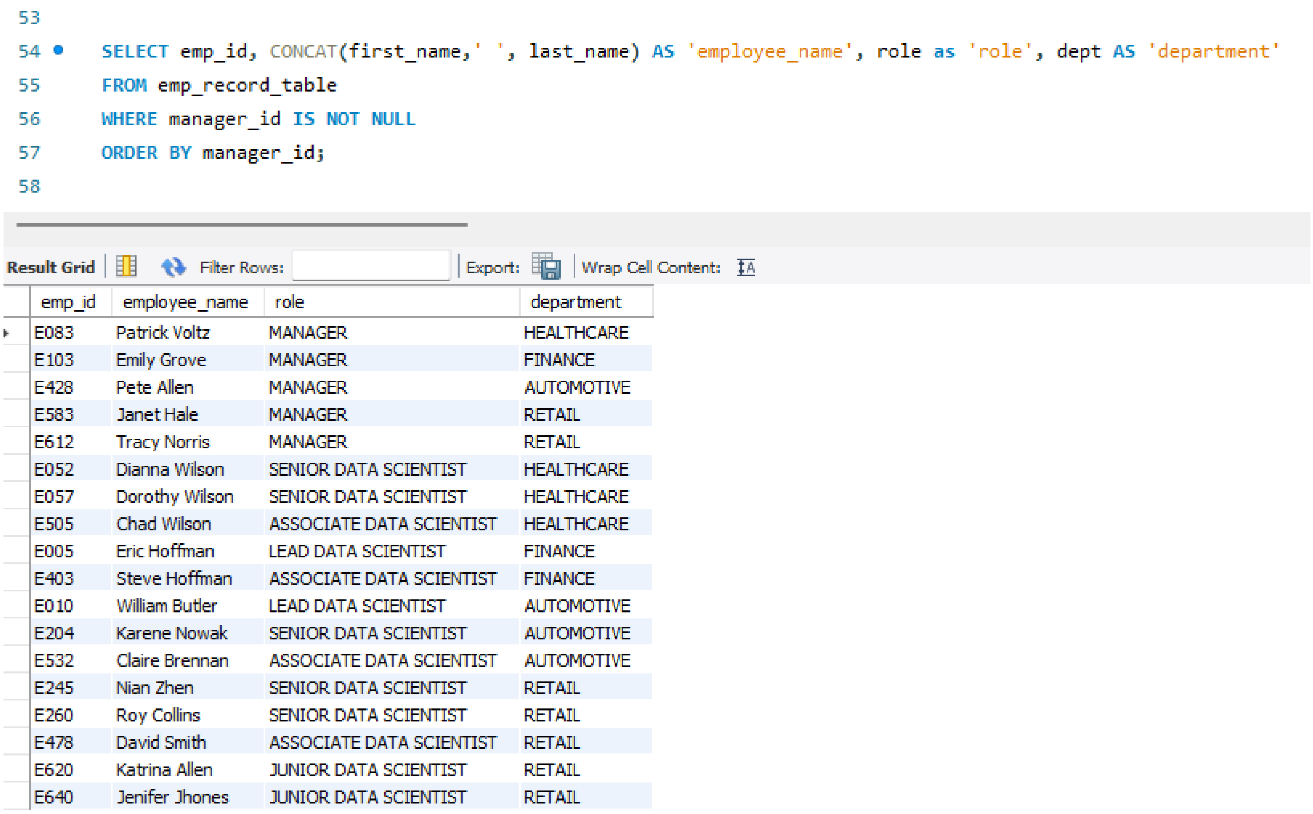
5. Write a query to concatenate the FIRST_NAME and the LAST_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME.



```
SELECT CONCAT(first_name, ' ', last_name) AS 'name', dept
FROM emp_record_table
WHERE dept = 'finance';
```

name	dept
Eric Hoffman	FINANCE
Emily Grove	FINANCE
Steve Hoffman	FINANCE

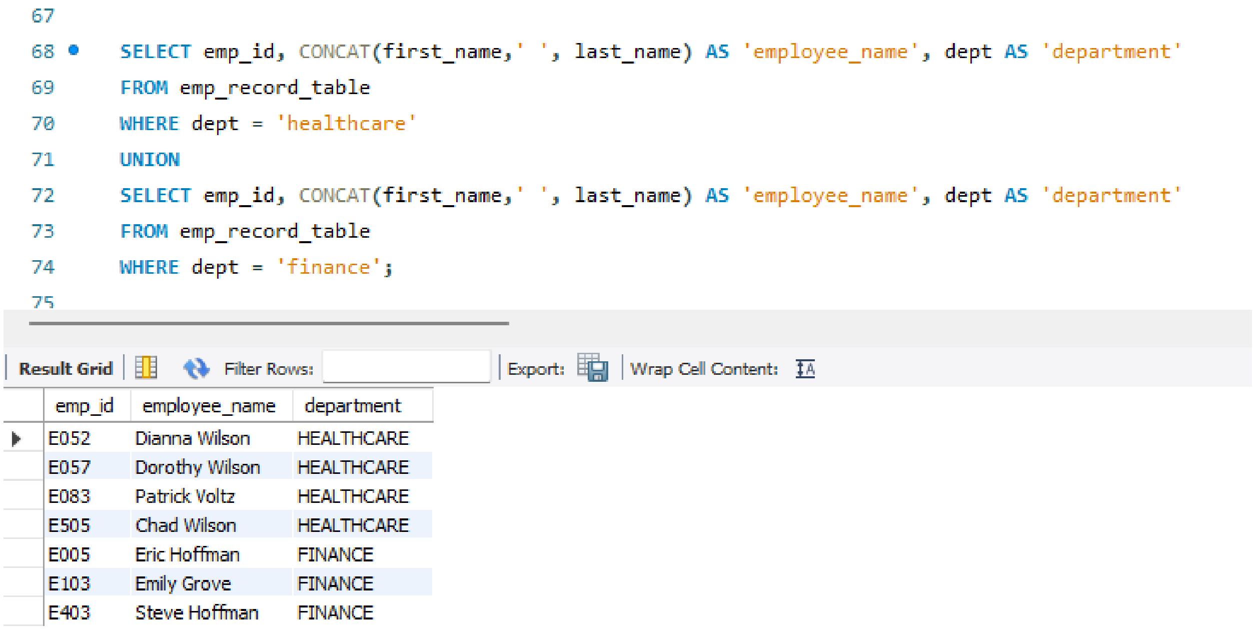
6. Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President).



```sql
SELECT emp_id, CONCAT(first_name,' ', last_name) AS 'employee_name', role as 'role', dept AS 'department'
FROM emp_record_table
WHERE manager_id IS NOT NULL
ORDER BY manager_id;
```

emp_id	employee_name	role	department
E083	Patrick Voltz	MANAGER	HEALTHCARE
E103	Emily Grove	MANAGER	FINANCE
E428	Pete Allen	MANAGER	AUTOMOTIVE
E583	Janet Hale	MANAGER	RETAIL
E612	Tracy Norris	MANAGER	RETAIL
E052	Dianna Wilson	SENIOR DATA SCIENTIST	HEALTHCARE
E057	Dorothy Wilson	SENIOR DATA SCIENTIST	HEALTHCARE
E505	Chad Wilson	ASSOCIATE DATA SCIENTIST	HEALTHCARE
E005	Eric Hoffman	LEAD DATA SCIENTIST	FINANCE
E403	Steve Hoffman	ASSOCIATE DATA SCIENTIST	FINANCE
E010	William Butler	LEAD DATA SCIENTIST	AUTOMOTIVE
E204	Karene Nowak	SENIOR DATA SCIENTIST	AUTOMOTIVE
E532	Claire Brennan	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE
E245	Nian Zhen	SENIOR DATA SCIENTIST	RETAIL
E260	Roy Collins	SENIOR DATA SCIENTIST	RETAIL
E478	David Smith	ASSOCIATE DATA SCIENTIST	RETAIL
E620	Katrina Allen	JUNIOR DATA SCIENTIST	RETAIL
E640	Jenifer Jhones	JUNIOR DATA SCIENTIST	RETAIL

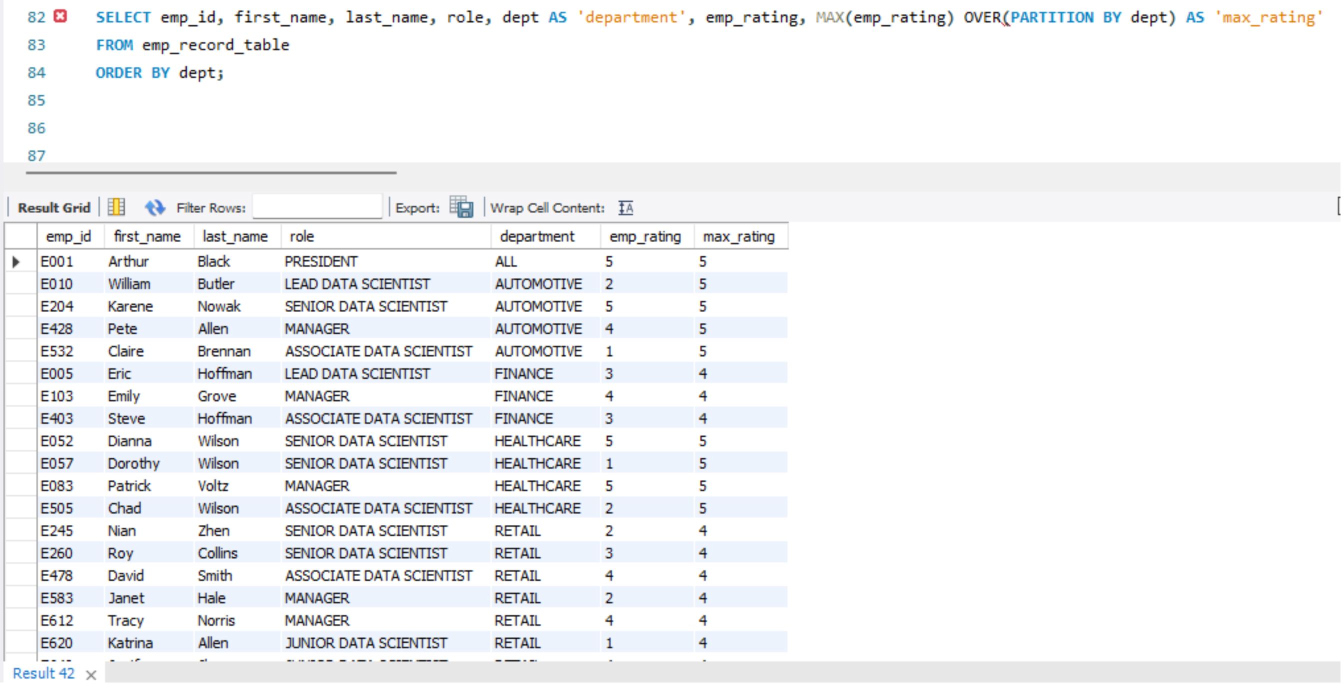
7. Write a query to list down all the employees from the healthcare and finance departments using union. Take data from the employee record table.



```sql
SELECT emp_id, CONCAT(first_name,' ', last_name) AS 'employee_name', dept AS 'department'
FROM emp_record_table
WHERE dept = 'healthcare'
UNION
SELECT emp_id, CONCAT(first_name,' ', last_name) AS 'employee_name', dept AS 'department'
FROM emp_record_table
WHERE dept = 'finance';
```

emp_id	employee_name	department
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

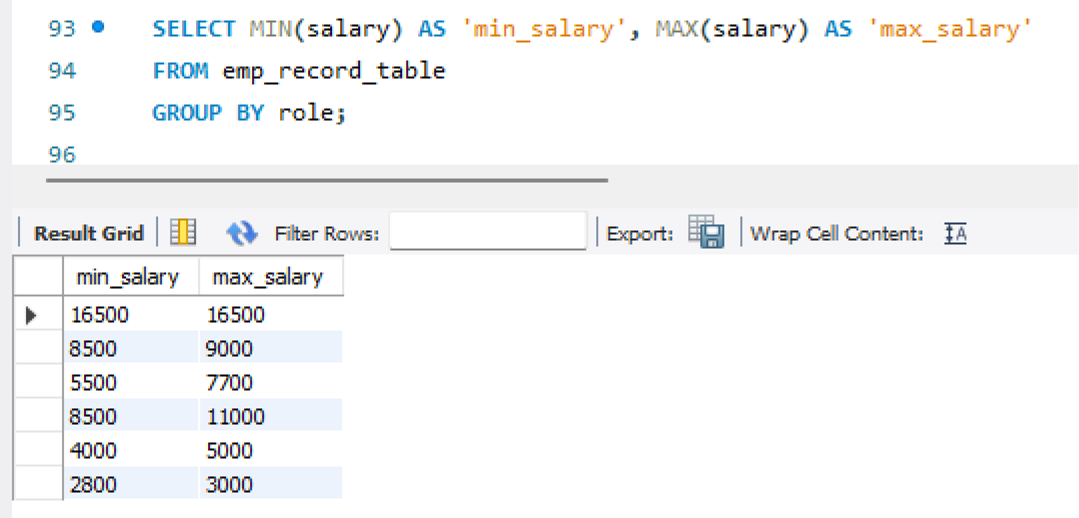
8. Write a query to list employee details such as EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPARTMENT, and EMP_RATING grouped by dept. Also include the respective employee rating along with the max emp rating for the department.



```sql
SELECT emp_id, first_name, last_name, role, dept AS 'department', emp_rating, MAX(emp_rating) OVER(PARTITION BY dept) AS 'max_rating'
FROM emp_record_table
ORDER BY dept;
```

emp_id	first_name	last_name	role	department	emp_rating	max_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

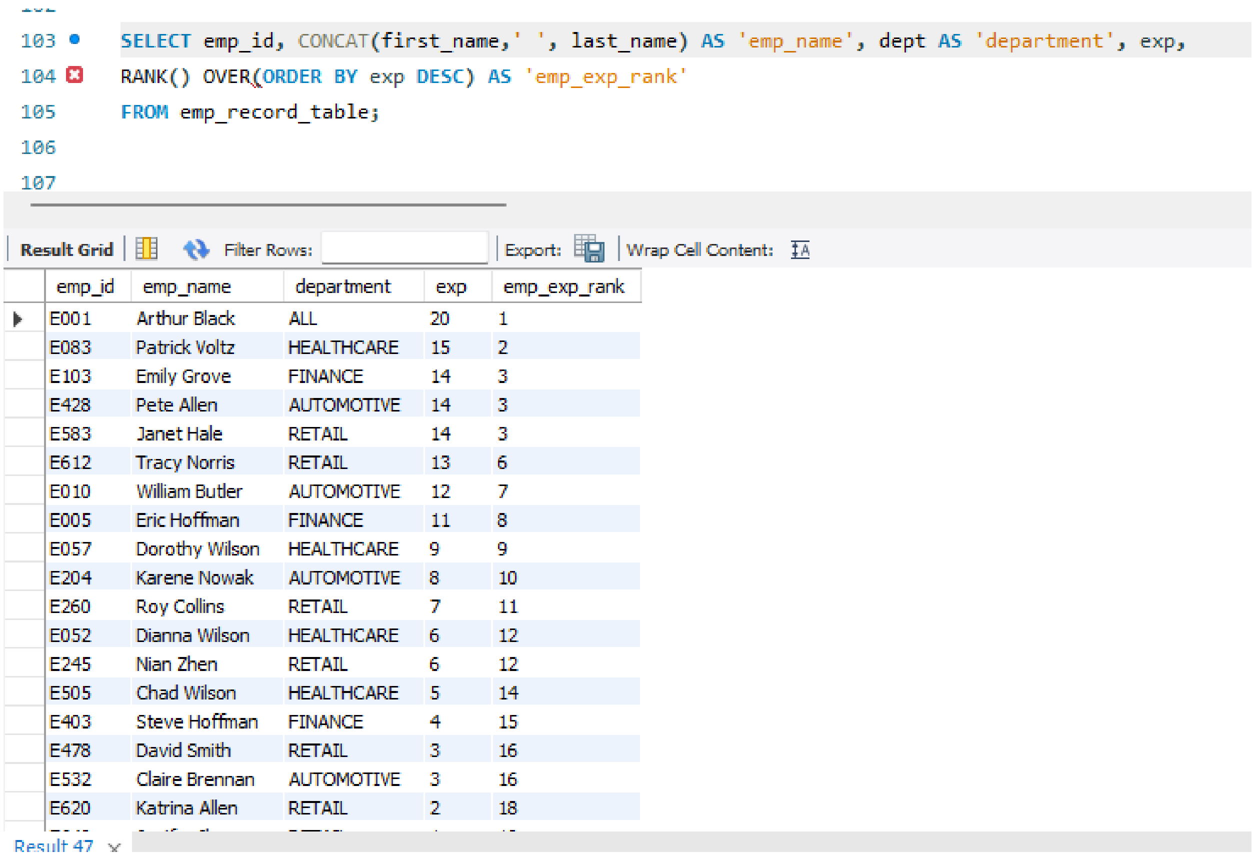
9. Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.



```sql
SELECT MIN(salary) AS 'min_salary', MAX(salary) AS 'max_salary'
FROM emp_record_table
GROUP BY role;
```

min_salary	max_salary
16500	16500
8500	9000
5500	7700
8500	11000
4000	5000
2800	3000

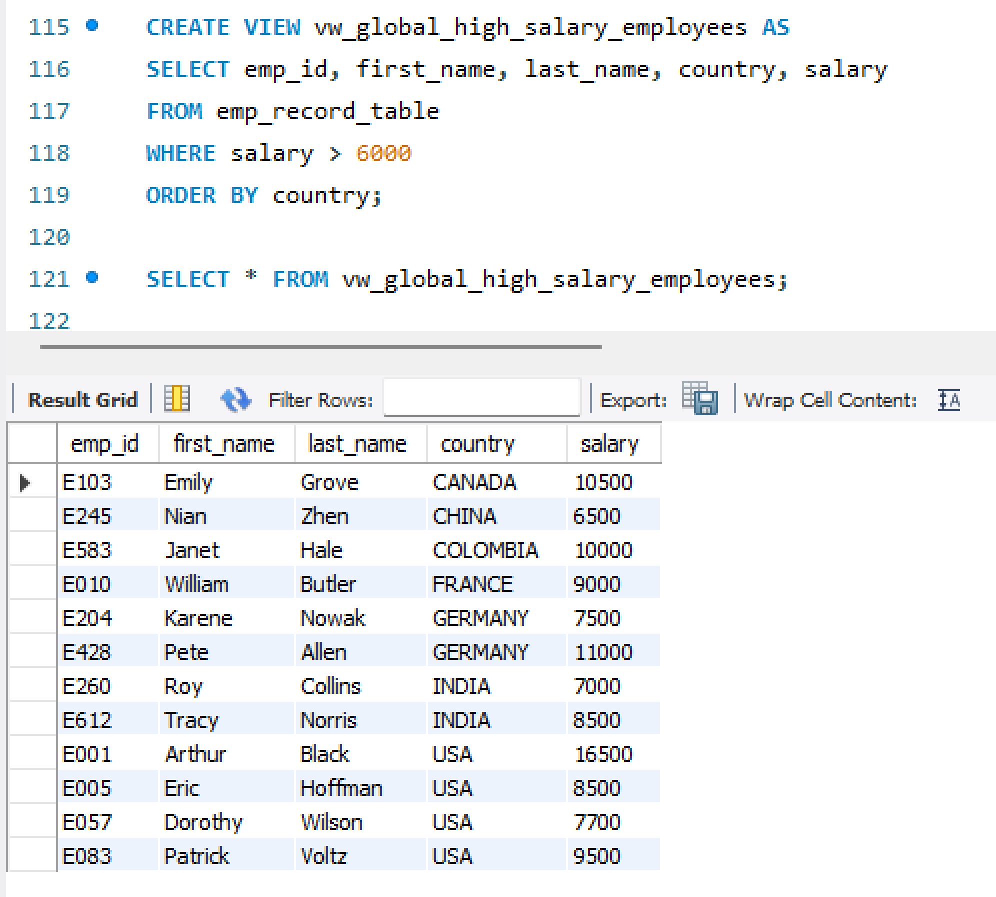
10. Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.



```sql
SELECT emp_id, CONCAT(first_name,' ', last_name) AS 'emp_name', dept AS 'department', exp,
RANK() OVER(ORDER BY exp DESC) AS 'emp_exp_rank'
FROM emp_record_table;
```

emp_id	emp_name	department	exp	emp_exp_rank
E001	Arthur Black	ALL	20	1
E083	Patrick Voltz	HEALTHCARE	15	2
E103	Emily Grove	FINANCE	14	3
E428	Pete Allen	AUTOMOTIVE	14	3
E583	Janet Hale	RETAIL	14	3
E612	Tracy Norris	RETAIL	13	6
E010	William Butler	AUTOMOTIVE	12	7
E005	Eric Hoffman	FINANCE	11	8
E057	Dorothy Wilson	HEALTHCARE	9	9
E204	Karene Nowak	AUTOMOTIVE	8	10
E260	Roy Collins	RETAIL	7	11
E052	Dianna Wilson	HEALTHCARE	6	12
E245	Nian Zhen	RETAIL	6	12
E505	Chad Wilson	HEALTHCARE	5	14
E403	Steve Hoffman	FINANCE	4	15
E478	David Smith	RETAIL	3	16
E532	Claire Brennan	AUTOMOTIVE	3	16
E620	Katrina Allen	RETAIL	2	18

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.

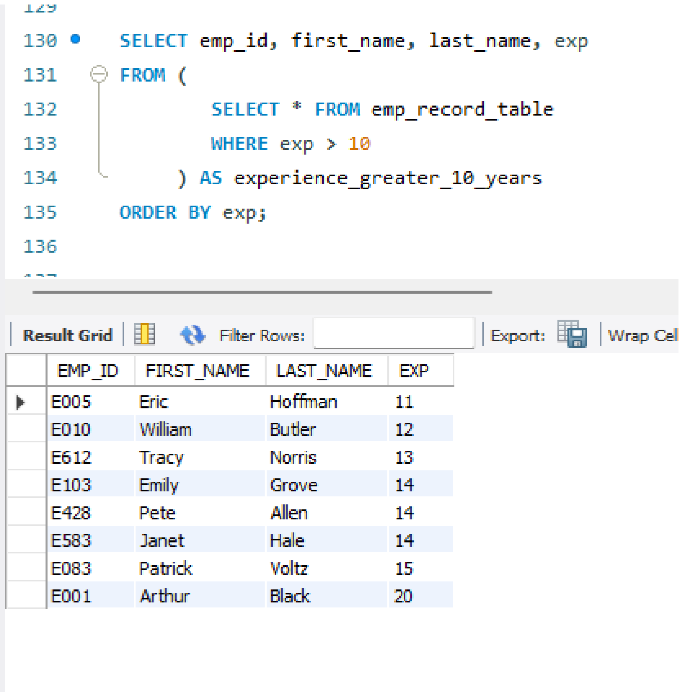


```sql
CREATE VIEW vw_global_high_salary_employees AS
SELECT emp_id, first_name, last_name, country, salary
FROM emp_record_table
WHERE salary > 6000
ORDER BY country;

SELECT * FROM vw_global_high_salary_employees;
```

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

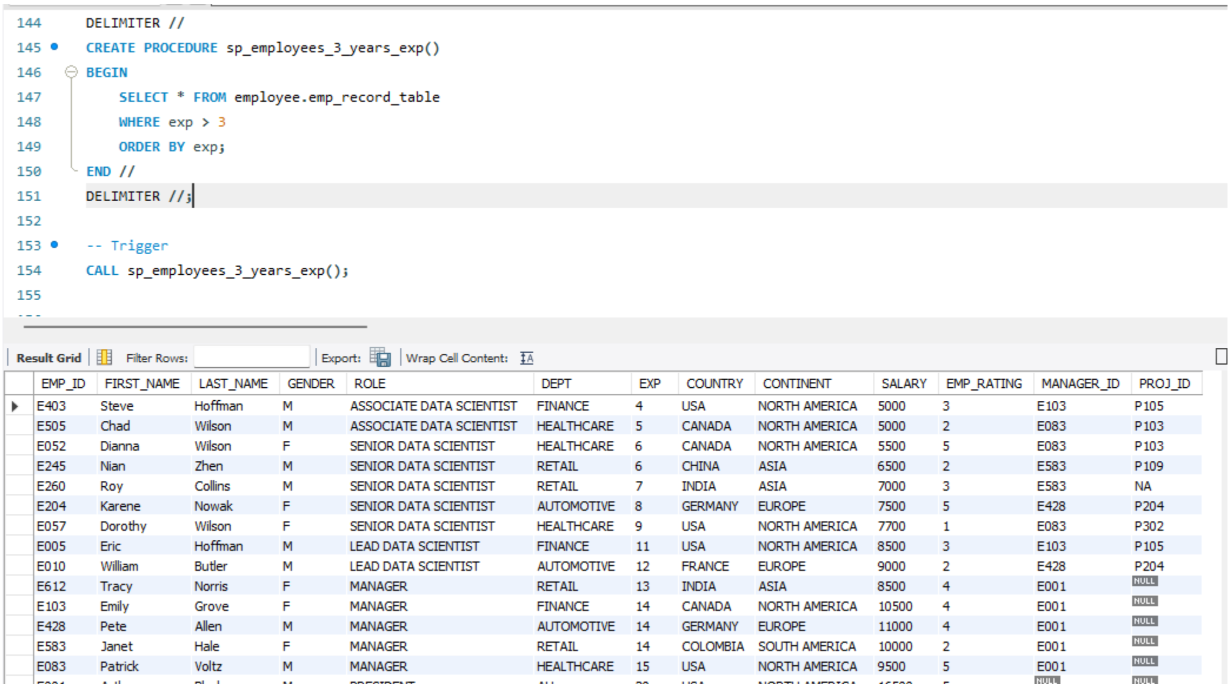
12. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.



```sql
SELECT emp_id, first_name, last_name, exp
FROM (
        SELECT * FROM emp_record_table
        WHERE exp > 10
     ) AS experience_greater_10_years
ORDER BY exp;
```

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

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.



```sql
DELIMITER //
CREATE PROCEDURE sp_employees_3_years_exp()
BEGIN
    SELECT * FROM employee.emp_record_table
    WHERE exp > 3
    ORDER BY exp;
END //
DELIMITER //;

-- Trigger
CALL sp_employees_3_years_exp();
```

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

13. 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:

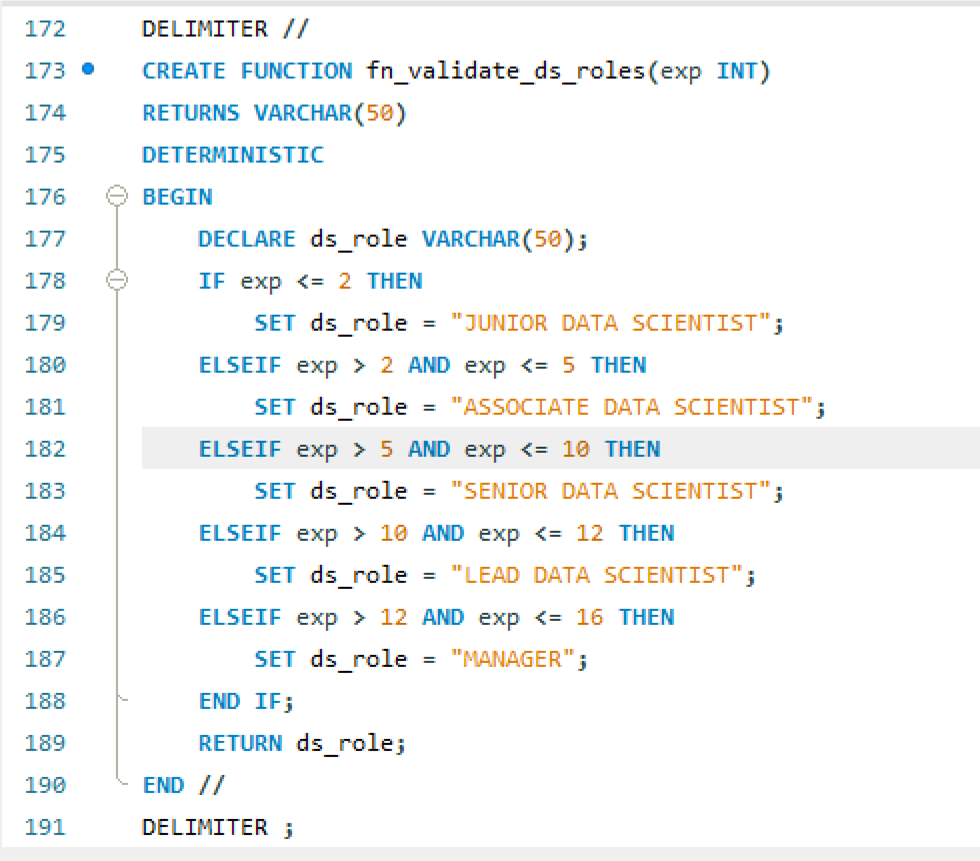
i. For an employee with experience less than or equal to 2 years assign 'JUNIOR DATA SCIENTIST',

ii. For an employee with the experience of 2 to 5 years assign 'ASSOCIATE DATA SCIENTIST',

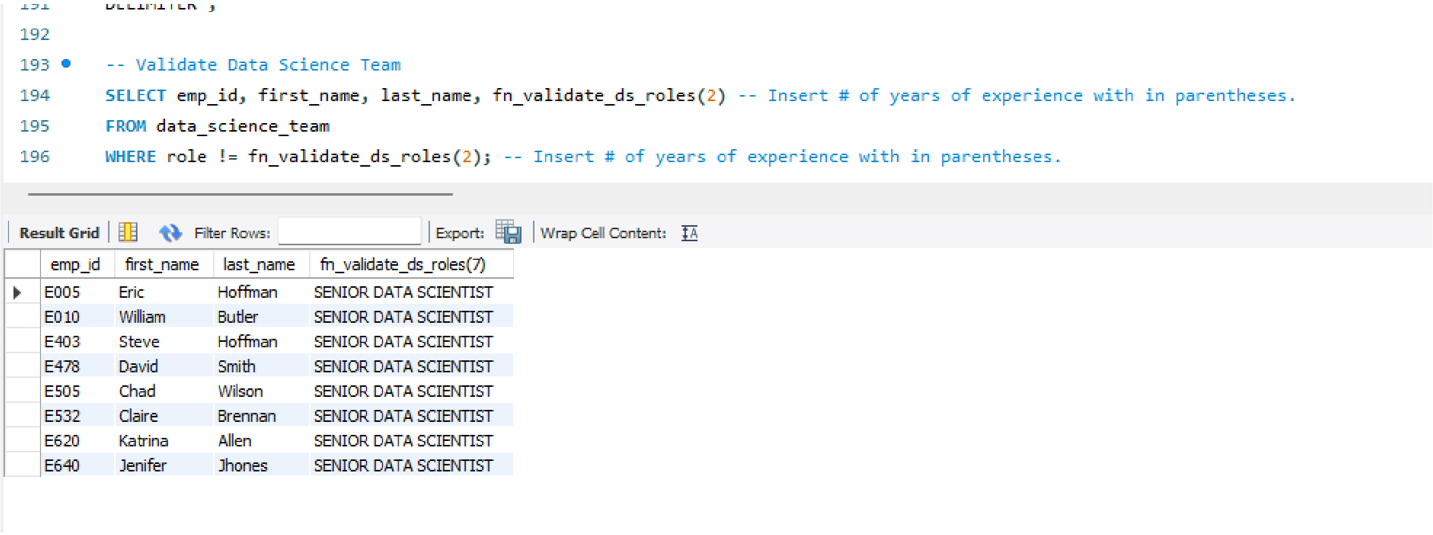
iii. For an employee with the experience of 5 to 10 years assign 'SENIOR DATA SCIENTIST',

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

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



```sql
DELIMITER //
CREATE FUNCTION fn_validate_ds_roles(exp INT)
RETURNS VARCHAR(50)
DETERMINISTIC
BEGIN
    DECLARE ds_role VARCHAR(50);
    IF exp <= 2 THEN
        SET ds_role = "JUNIOR DATA SCIENTIST";
    ELSEIF exp > 2 AND exp <= 5 THEN
        SET ds_role = "ASSOCIATE DATA SCIENTIST";
    ELSEIF exp > 5 AND exp <= 10 THEN
        SET ds_role = "SENIOR DATA SCIENTIST";
    ELSEIF exp > 10 AND exp <= 12 THEN
        SET ds_role = "LEAD DATA SCIENTIST";
    ELSEIF exp > 12 AND exp <= 16 THEN
        SET ds_role = "MANAGER";
    END IF;
    RETURN ds_role;
END //
DELIMITER ;
```

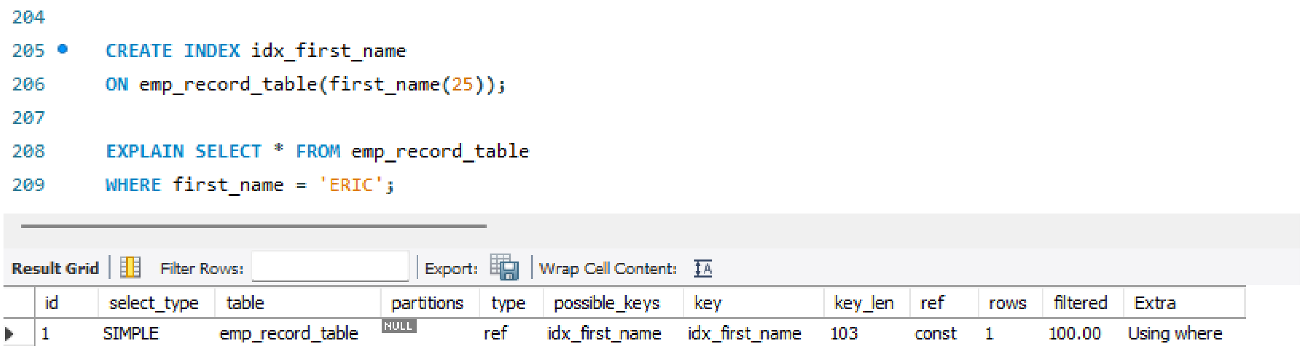


```sql
DELIMITER ;

-- Validate Data Science Team
SELECT emp_id, first_name, last_name, fn_validate_ds_roles(2) -- Insert # of years of experience with in parentheses.
FROM data_science_team
WHERE role != fn_validate_ds_roles(2); -- Insert # of years of experience with in parentheses.
```

emp_id	first_name	last_name	fn_validate_ds_roles(7)
E005	Eric	Hoffman	SENIOR DATA SCIENTIST
E010	William	Butler	SENIOR DATA SCIENTIST
E403	Steve	Hoffman	SENIOR DATA SCIENTIST
E478	David	Smith	SENIOR DATA SCIENTIST
E505	Chad	Wilson	SENIOR DATA SCIENTIST
E532	Claire	Brennan	SENIOR DATA SCIENTIST
E620	Katrina	Allen	SENIOR DATA SCIENTIST
E640	Jenifer	Jhones	SENIOR DATA SCIENTIST

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.

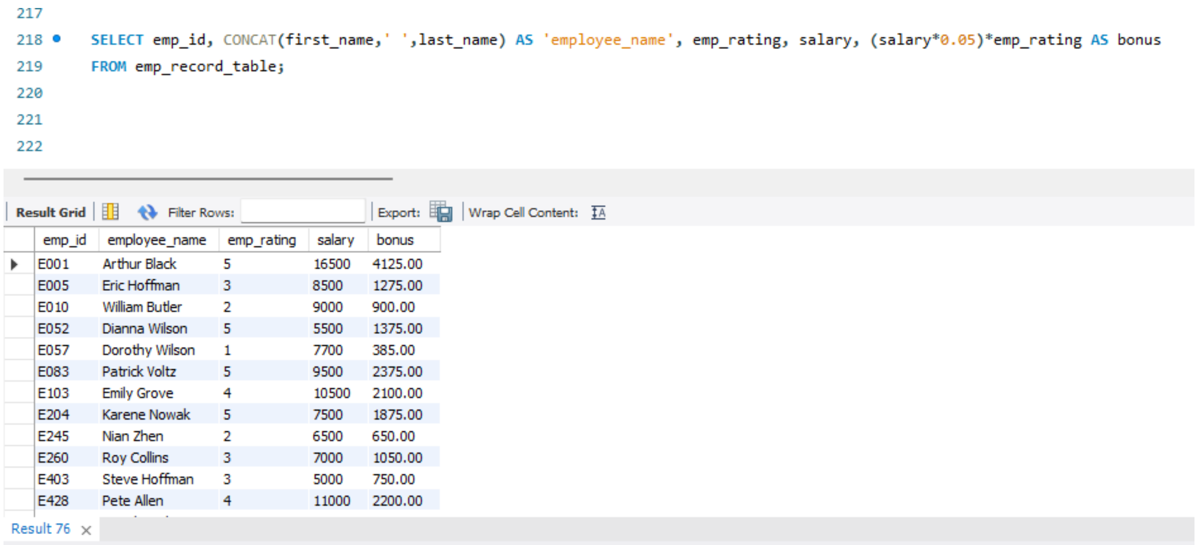


```
CREATE INDEX idx_first_name
ON emp_record_table(first_name(25));

EXPLAIN SELECT * FROM emp_record_table
WHERE first_name = 'ERIC';
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	emp_record_table	NULL	ref	idx_first_name	idx_first_name	103	const	1	100.00	Using where

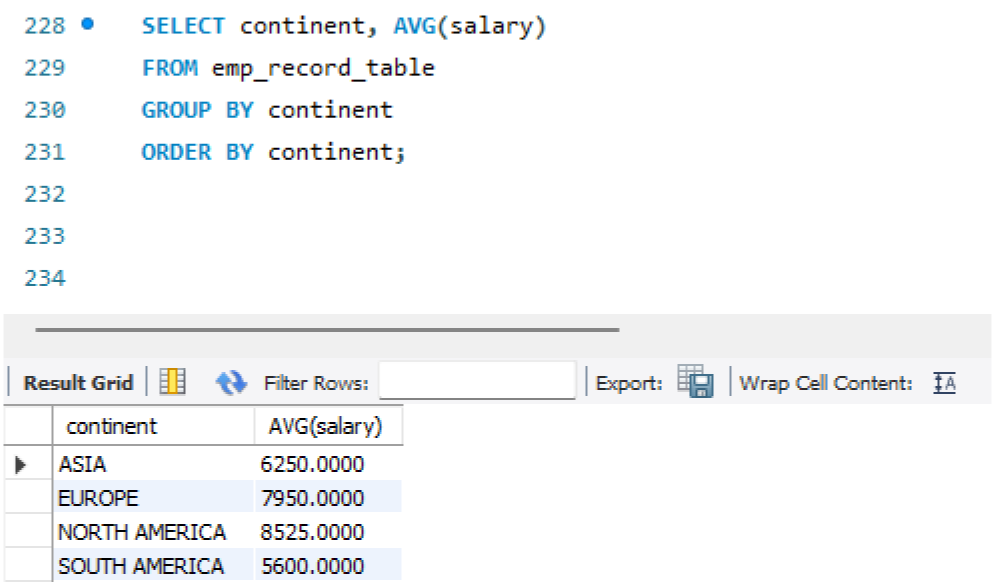
16. Write a query to calculate the bonus for all the employees, based on their ratings and salaries (Use the formula: 5% of salary * employee rating).



```
SELECT emp_id, CONCAT(first_name,' ',last_name) AS 'employee_name', emp_rating, salary, (salary*0.05)*emp_rating AS bonus
FROM emp_record_table;
```

emp_id	employee_name	emp_rating	salary	bonus
E001	Arthur Black	5	16500	4125.00
E005	Eric Hoffman	3	8500	1275.00
E010	William Butler	2	9000	900.00
E052	Dianna Wilson	5	5500	1375.00
E057	Dorothy Wilson	1	7700	385.00
E083	Patrick Voltz	5	9500	2375.00
E103	Emily Grove	4	10500	2100.00
E204	Karene Nowak	5	7500	1875.00
E245	Nian Zhen	2	6500	650.00
E260	Roy Collins	3	7000	1050.00
E403	Steve Hoffman	3	5000	750.00
E428	Pete Allen	4	11000	2200.00

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



```sql
SELECT continent, AVG(salary)
FROM emp_record_table
GROUP BY continent
ORDER BY continent;
```

continent	AVG(salary)
ASIA	6250.0000
EUROPE	7950.0000
NORTH AMERICA	8525.0000
SOUTH AMERICA	5600.0000