**ScienceQtech Employee Performance Mapping.**

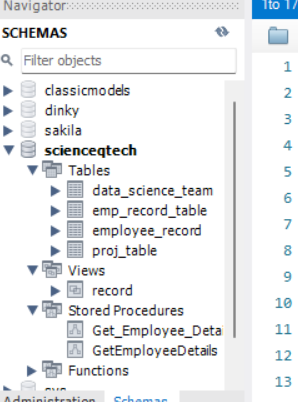
PROJECT REPORT SUBMITTED BY :DINKY

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.

**Output:**

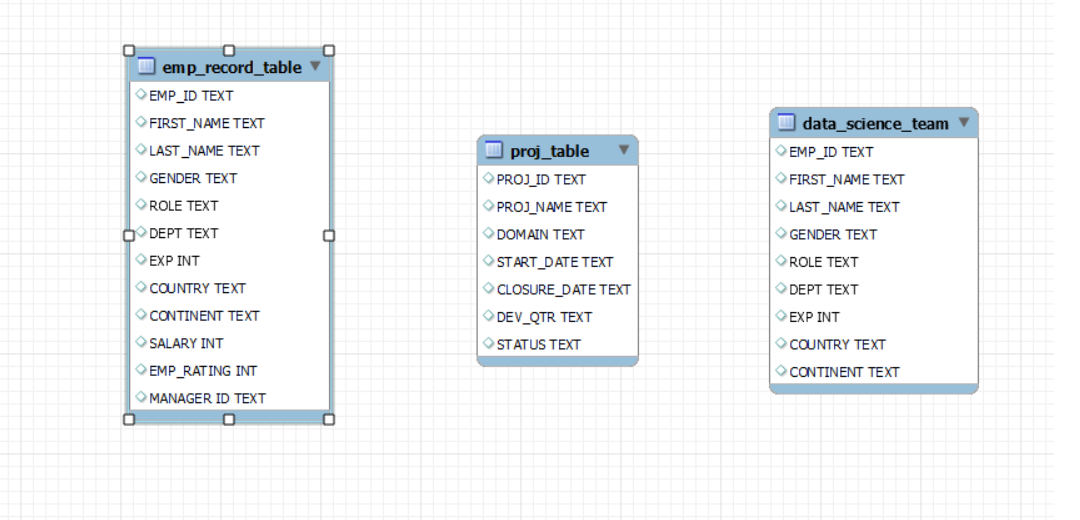
**I create a database named as scienceqtech**

CREATE DATABASE scienceqtech;



2.Create an ER diagram for the given **project** and the **employee** databases.

**Output:**

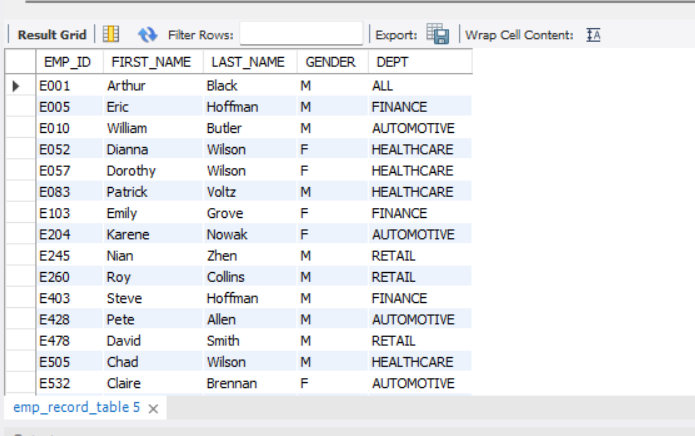
****

1. 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 code:**

SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,GENDER,DEPT FROM scienceqtech.emp\_record\_table;

**Output:**

****

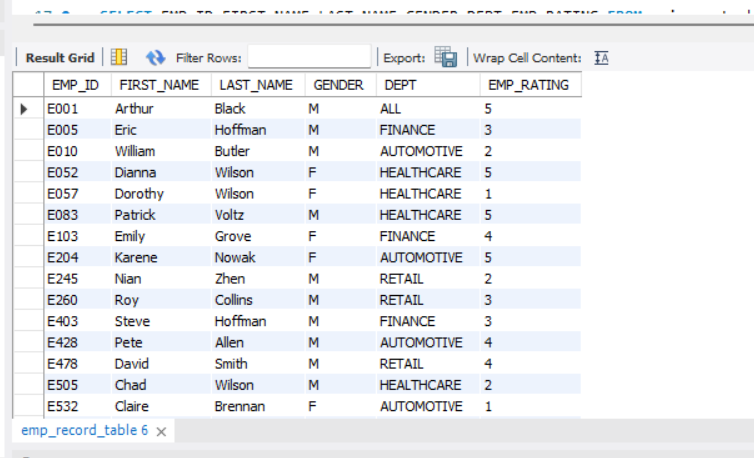
1. Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPARTMENT, and EMP\_RATING if the EMP\_RATING is:

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

**SQL code:**

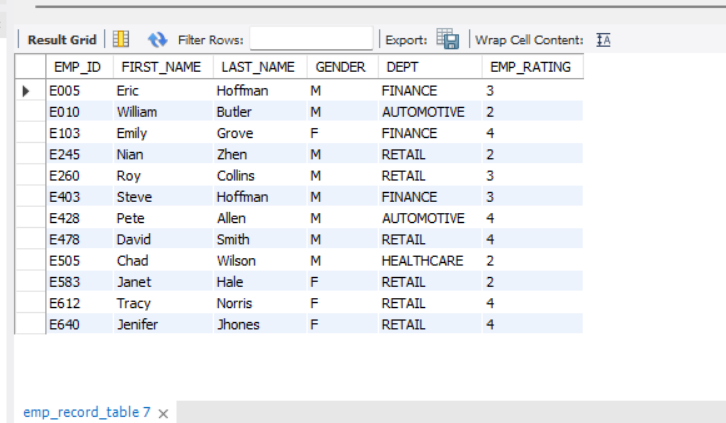
SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,GENDER,DEPT,EMP\_RATING FROM scienceqtech.emp\_record\_table;

**Output:**



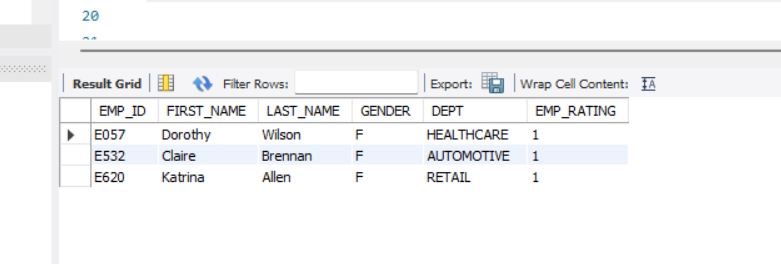
1)SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,GENDER,DEPT,EMP\_RATING FROM scienceqtech.emp\_record\_table where EMP\_RATING>4 ;

**Output:**



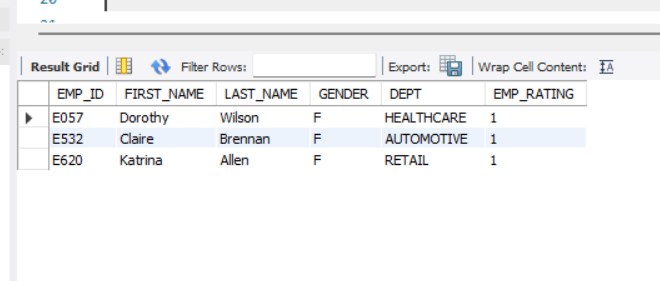
2)SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,GENDER,DEPT,EMP\_RATING FROM scienceqtech.emp\_record\_table where EMP\_RATING between 2 AND 4 ;

**Output:**



3)SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,GENDER,DEPT,EMP\_RATING FROM scienceqtech.emp\_record\_table where EMP\_RATING < 2 ;

**Output:**

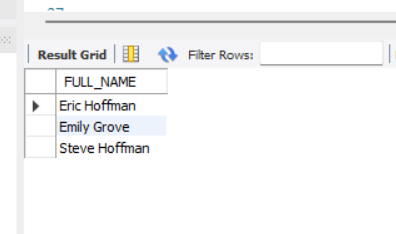


1. Write a query to concatenate the FIRST\_NAME and the LAST\_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME.

**SQL code:**

SELECT concat(FIRST\_NAME," " ,LAST\_NAME) AS FULL\_NAME FROM scienceqtech.emp\_record\_table where DEPT="FINANCE" ;

**Output:**



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

**SQL code:**

SELECT m.EMP\_ID, m.FIRST\_NAME, m.LAST\_NAME, m.ROLE,m.EXP, m.DEPT, COUNT(e.EMP\_ID) as "EMP\_COUNT"

FROM scienceqtech.emp\_record\_table m

INNER JOIN scienceqtech.emp\_record\_table e

ON m.EMP\_ID = e.MANAGER\_ID

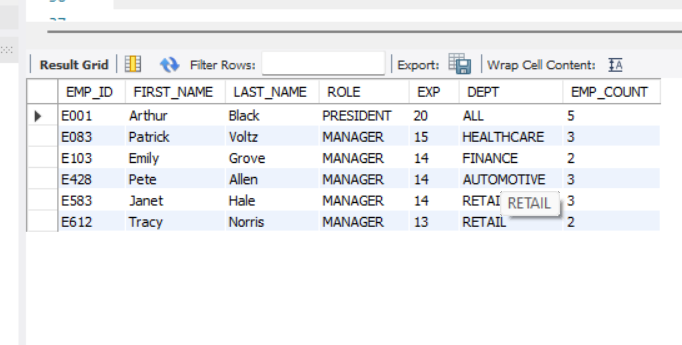
AND e.EMP\_ID != e.MANAGER\_ID

WHERE m.ROLE IN ("MANAGER", "PRESIDENT", "CEO")

GROUP BY m.EMP\_ID, m.FIRST\_NAME, m.LAST\_NAME, m.ROLE, m.EXP, m.DEPT

ORDER BY m.EMP\_ID;

**Output:**



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

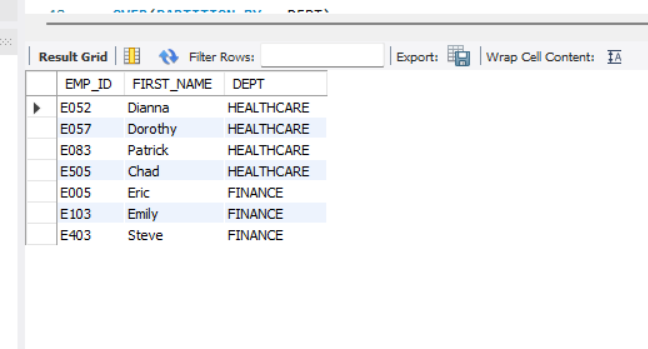
**SQL code:**

SELECT EMP\_ID,FIRST\_NAME,DEPT FROM emp\_record\_table WHERE DEPT="HEALTHCARE"

UNION

SELECT EMP\_ID,FIRST\_NAME,DEPT FROM emp\_record\_table WHERE DEPT="FINANCE";

**Output:**



1. 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.

**SQL code:**

SELECT m.EMP\_ID,m.FIRST\_NAME,m.LAST\_NAME,m.ROLE,m.DEPT,m.EMP\_RATING,MAX(m.EMP\_RATING)

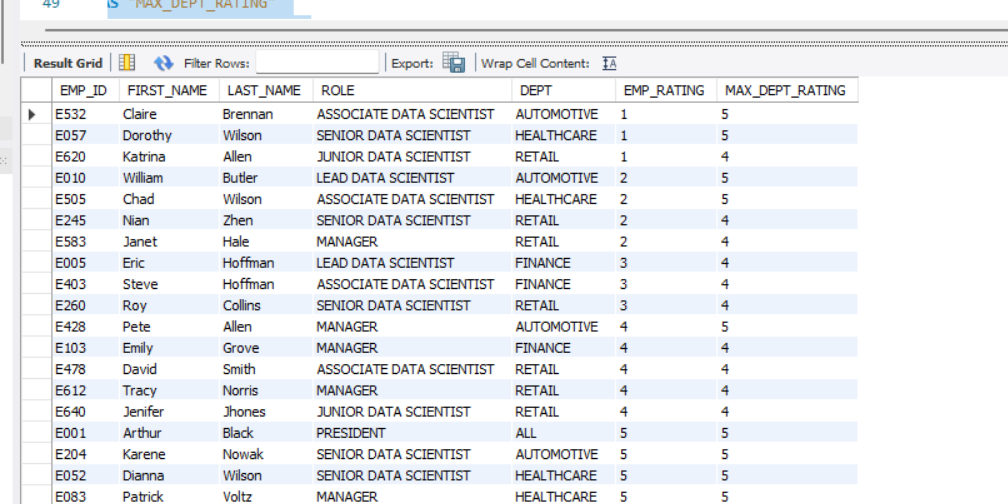
OVER(PARTITION BY m.DEPT)

AS "MAX\_DEPT\_RATING"

FROM emp\_record\_table m

order by EMP\_RATING;

**Output:**



1. 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 code:**

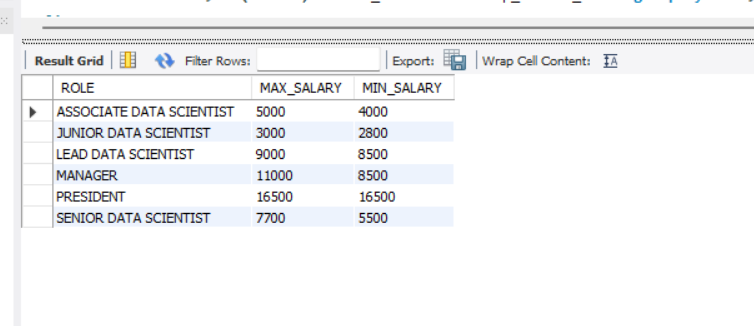
SELECT DISTINCT(ROLE), MAX(SALARY)

OVER (PARTITION BY ROLE) MAX\_SALARY, MIN(SALARY)

OVER (PARTITION BY ROLE) MIN\_SALARY

FROM scienceqtech.emp\_record\_table;

**Output:**

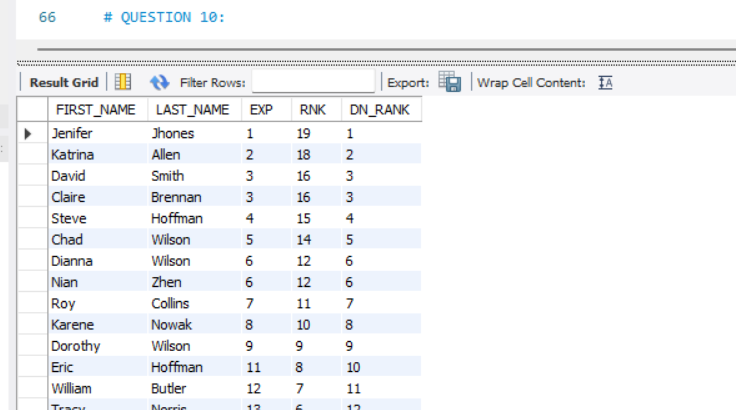


1. Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.

**SQL code:**

SELECT FIRST\_NAME,LAST\_NAME, EXP,RANK() over(order by EXP DESC) AS RNK ,DENSE\_RANK() OVER (ORDER BY EXP) AS DN\_RANK FROM emp\_record\_table ;

**Output:**



1. 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.

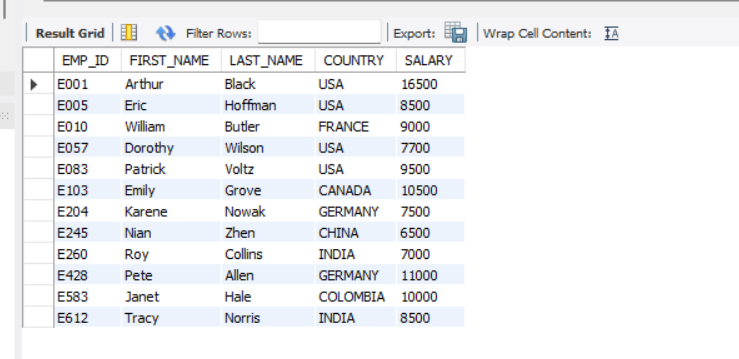
**SQL code:**

CREATE VIEW RECORD1 AS

SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,COUNTRY,SALARY FROM scienceqtech.emp\_record\_table WHERE SALARY>6000;

select \* FROM RECORD1;

**Output:**



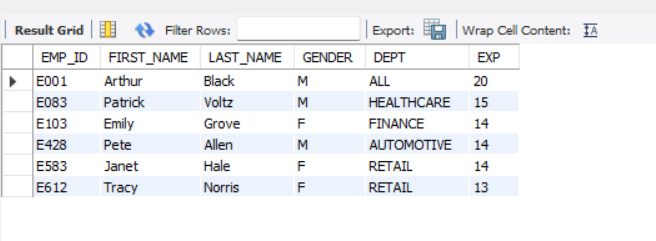
1. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

**SQL code:**

SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,GENDER,DEPT,EXP FROM emp\_record\_table

WHERE EMP\_ID IN (SELECT MANAGER\_ID FROM emp\_record\_table) ;

**Output:**



1. 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 code:**

DELIMITER //

CREATE PROCEDURE Get\_Employee\_Details1()

BEGIN

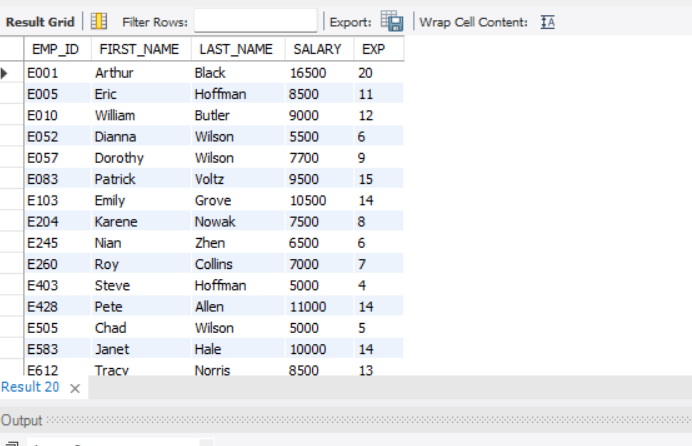
SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,SALARY,EXP FROM emp\_record\_table

WHERE EXP >3;

END //

CALL Get\_Employee\_Details1();

**Output:**



1. 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'.

**SQL code:**

DELIMITER //

CREATE FUNCTION JOB\_PROFILE

( EXP int)

RETURNS VARCHAR (50)

DETERMINISTIC

BEGIN

DECLARE JOB\_PROFILE VARCHAR (50);

IF EXP > 12 AND 16 THEN

SET JOB\_PROFILE="MANAGER";

ELSEIF EXP > 10 AND 12 THEN

SET JOB\_PROFILE="LEAD DATA SCIENTIST";

ELSEIF EXP > 5 AND 10 THEN

SET JOB\_PROFILE="SENIOR DATA SCIENTIST";

ELSEIF EXP > 2 AND 5 THEN

SET JOB\_PROFILE="ASSOCIATE DATA SCIENTIST";

ELSEIF EXP<=2 THEN

SET JOB\_PROFILE="JUNIOR DATA SCIENTIST";

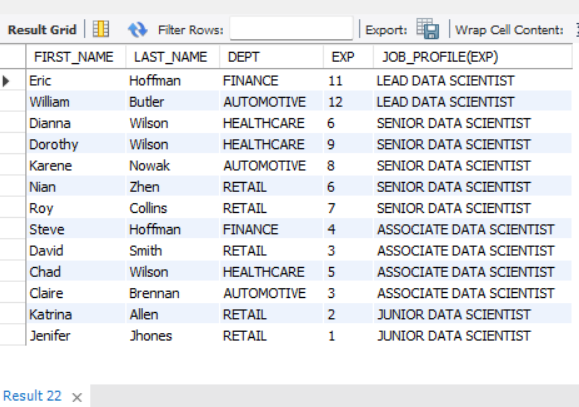
END IF;

RETURN (JOB\_PROFILE);

END //

SELECT FIRST\_NAME, LAST\_NAME, DEPT,EXP,JOB\_PROFILE(EXP) FROM data\_science\_team;

**Output:**



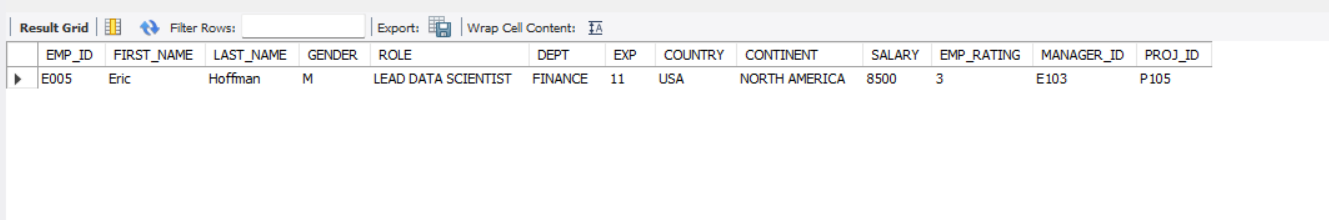
1. Create an index to improve the cost and performance of the query to find the employee whose FIRST\_NAME is ‘Eric’ in the employee table after checking the execution plan.

**SQL code:**

SELECT \* FROM scienceqtech.emp\_record\_table

WHERE FIRST\_NAME='Eric';

**Output:**

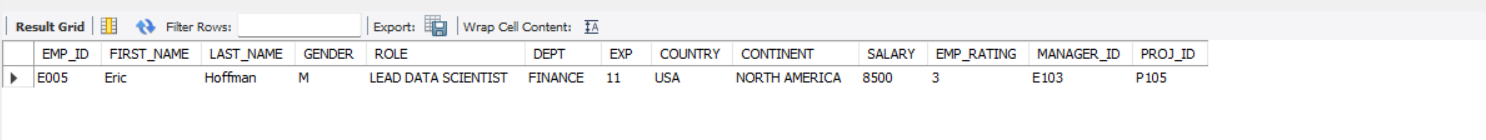


**SQL code:**

CREATE INDEX RAM ON scienceqtech.emp\_record\_table(FIRST\_NAME(40));

SELECT \* FROM emp\_record\_table WHERE FIRST\_NAME="ERIC";

**Output:**



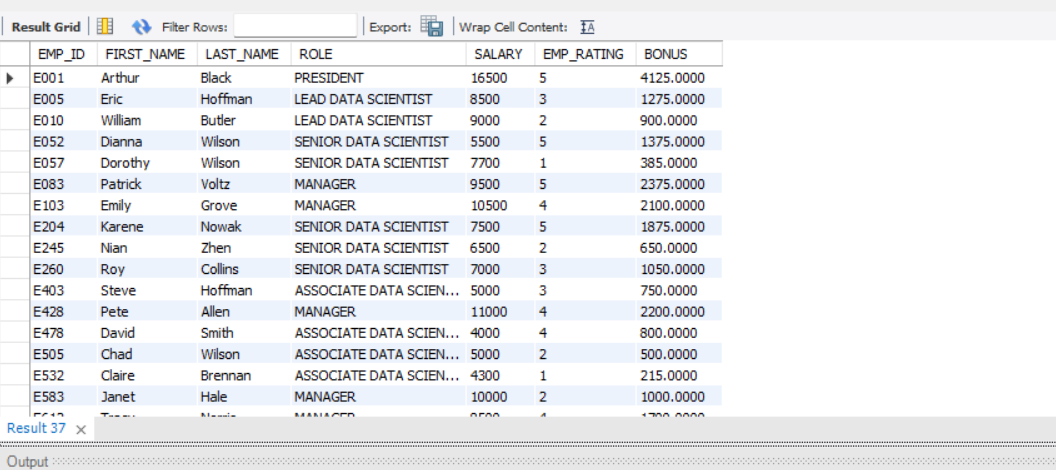
1. Write a query to calculate the bonus for all the employees, based on their ratings and salaries (Use the formula: 5% of salary \* employee rating).

**SQL code:**

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, SALARY, EMP\_RATING, (SALARY/20)\*EMP\_RATING AS BONUS

FROM scienceqtech.emp\_record\_table;

**Output:**



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

**SQL code:**

SELECT CONTINENT, COUNTRY, AVG(SALARY) as AVERAGE\_SALARY

FROM scienceqtech.emp\_record\_table

GROUP BY CONTINENT, COUNTRY WITH ROLLUP;

**Output:**

