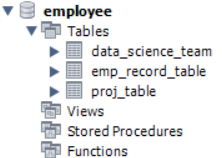
**Action 1: -**

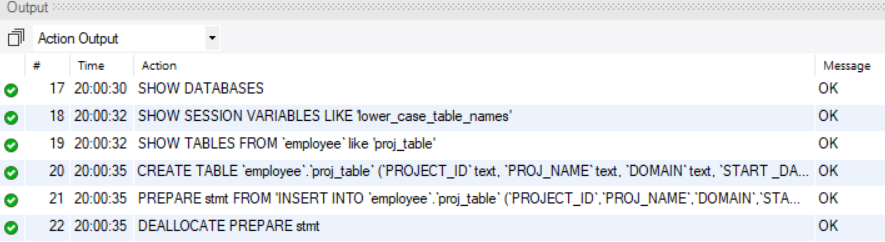
Create a database named employee, then import data\_science\_team.csv proj\_table.csv and emp\_record\_table.csv into the employee database from the given resources.

**Answer –**

**create database employee;**

****

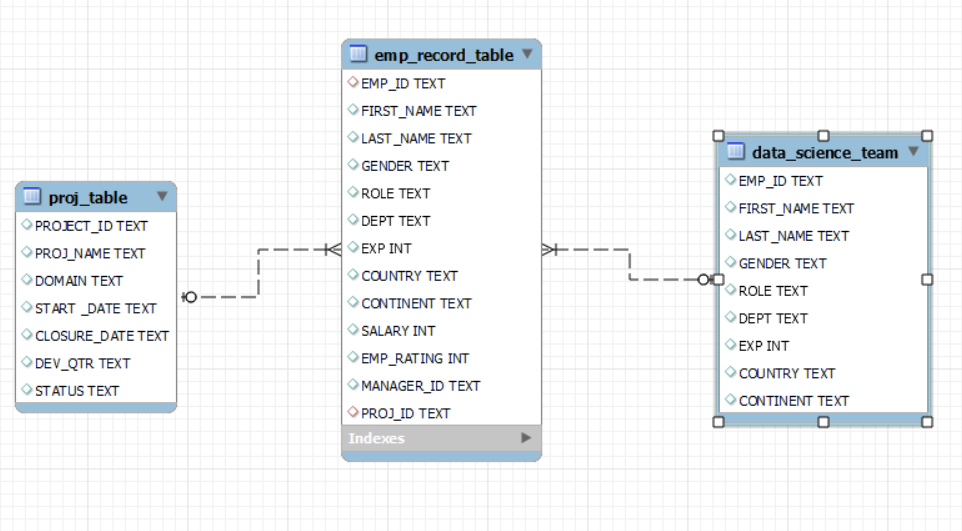




**Action 2: -**

Create an ER diagram for the given employee database.

**Answer –**

****

**Action 3: -**

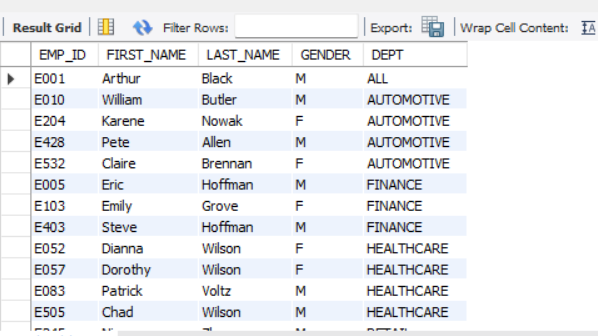
Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department.

**Answer -**

select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT FROM

emp\_record\_table order by dept;

**Output**



**Action 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

**Answer –**

select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT,EMP\_RATING,

case

when EMP\_RATING < 2 then 'Less then 2'

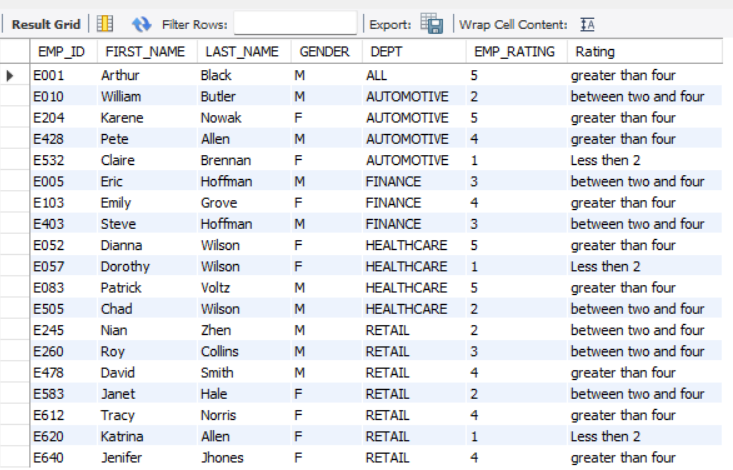
when EMP\_RATING < 4 then 'between two and four'

else 'greater than four '

end as Rating

from emp\_record\_table order by dept;

**Output**

****

**Action 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.

**Answer –**

select concat(first\_name," ",last\_name) as Name

from emp\_record\_table

where DEPT = 'FINANCE';

**Output**



**Action 6: -**

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

**Answer –**

select m.FIRST\_NAME as manager\_name, count(\*) No\_Emps\_reporting

from emp\_record\_table as e

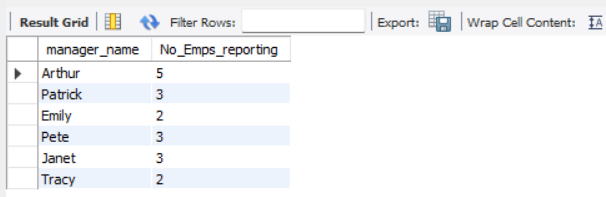
join emp\_record\_table as m

on m.emp\_id=e.MANAGER\_ID

group by manager\_name

having count(\*) > 0;

**OUTPUT**

****

**Action 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.

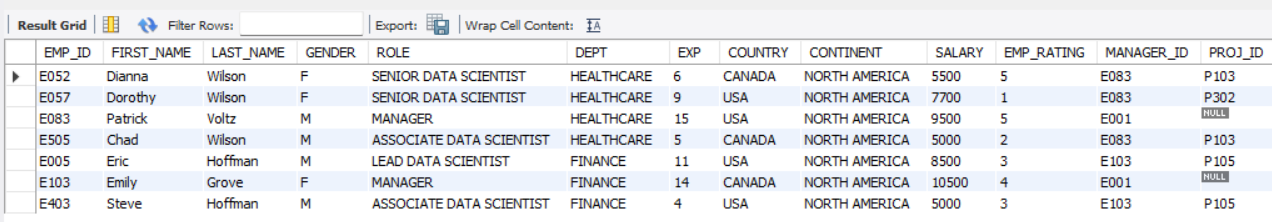
**Answer –**

select \* from emp\_record\_table where DEPT = 'Healthcare'

UNION

select \* from emp\_record\_table where DEPT = 'Finance';

**OUTPUT**

****

**Action 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.

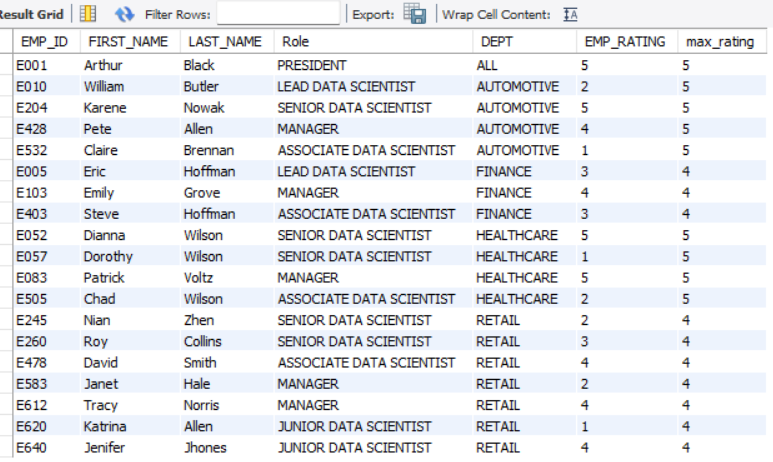
**Answer –**

select EMP\_ID,FIRST\_NAME,LAST\_NAME,Role,DEPT, EMP\_RATING,

max(emp\_rating) over(partition by DEPT) max\_rating

from emp\_record\_table;

**OUTPUT**

****

**Action 9: -**

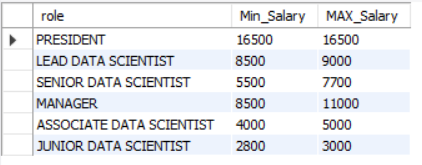
Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.

**Answer -**

select role,min(Salary) Min\_Salary ,Max(Salary) MAX\_Salary from emp\_record\_table

group by ROLE;

**OUTPUT**

****

**Action 10: -**

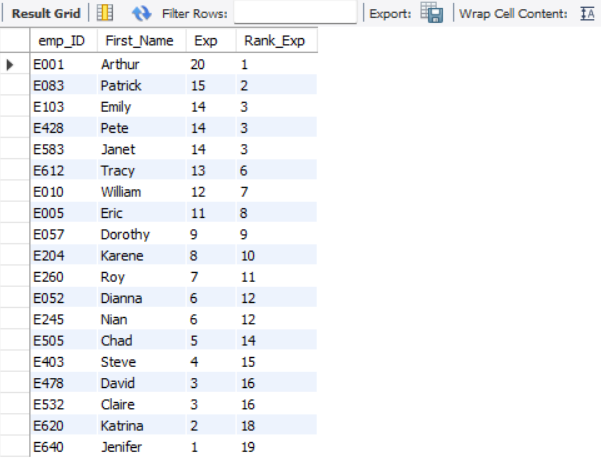
Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.

**Answer –**

select emp\_ID,First\_Name,Exp,Rank() over(order by Exp desc) Rank\_Exp

from emp\_record\_table;

**OUTPUT**

****

**Action 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.

**Answer –**

create VIEW VempSalAbove6k

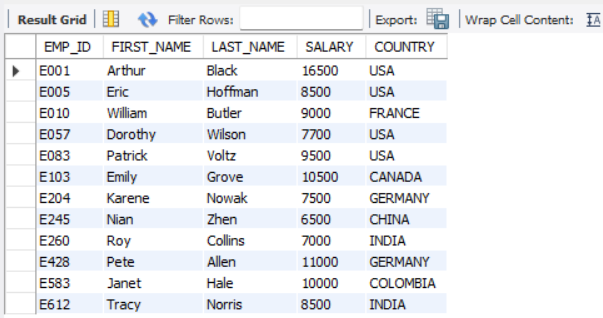
AS

select EMP\_ID,FIRST\_NAME,LAST\_NAME,SALARY,COUNTRY from emp\_record\_table

where SALARY > 6000;

select \* from VempSalAbove6k;

**OUTPUT**

****

**Action 12: -**

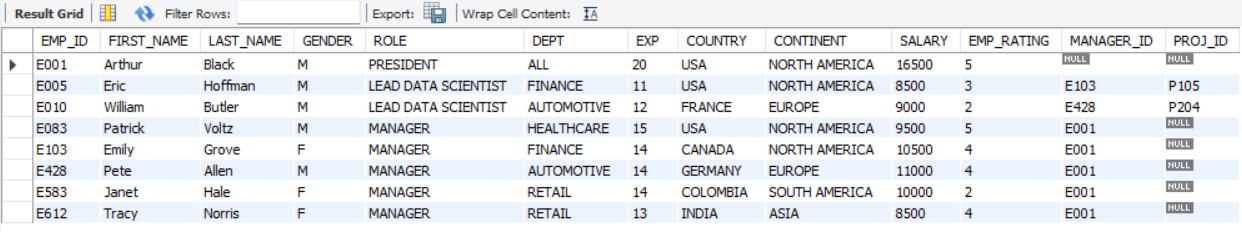
Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

**Answer –**

select \* from emp\_record\_table where EMP\_ID in(

select Emp\_ID from emp\_record\_table where Exp > 10);

**OUTPUT**

****

**Action 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.

**Answer –**

USE `employee`;

DROP procedure IF EXISTS `employee`.`emp\_3plusExp`;

;

DELIMITER $$

USE `employee`$$

CREATE PROCEDURE `emp\_3plusExp`()

BEGIN

select \* from emp\_record\_table where Exp > 3;

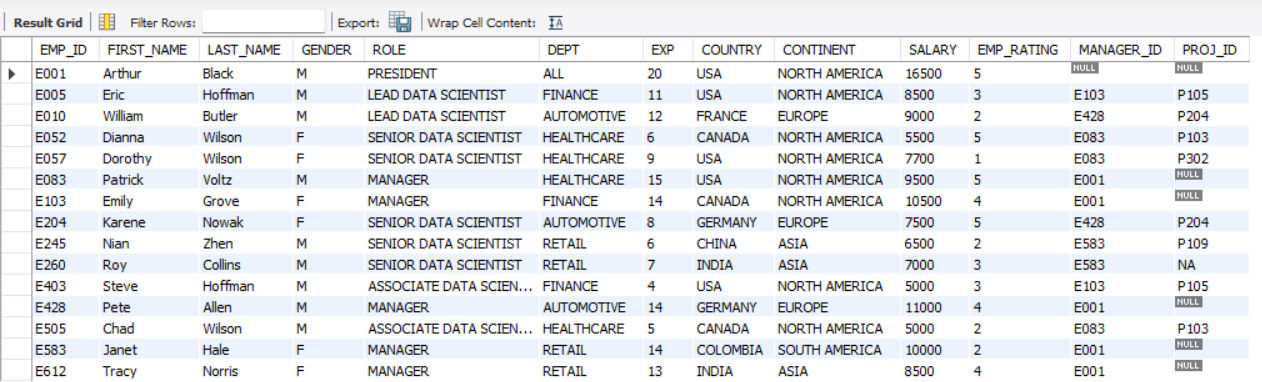
END$$

DELIMITER ;

;

call emp\_3plusExp;

**OUTPUT**

****

**Action 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'.

**Answer-**

CREATE DEFINER=`root`@`localhost` FUNCTION `Check\_Job\_Profile`(EXP INT, ROLE VARCHAR(50)) RETURNS varchar(50) CHARSET utf8mb4

DETERMINISTIC

BEGIN

declare Expected\_Role varchar(50);

IF EXP <= 2 THEN

SET expected\_role = 'JUNIOR DATA SCIENTIST';

ELSEIF EXP BETWEEN 3 AND 5 THEN

SET expected\_role = 'ASSOCIATE DATA SCIENTIST';

ELSEIF EXP BETWEEN 6 AND 10 THEN

SET expected\_role = 'SENIOR DATA SCIENTIST';

ELSEIF EXP BETWEEN 11 AND 12 THEN

SET expected\_role = 'LEAD DATA SCIENTIST';

ELSEIF EXP BETWEEN 13 AND 16 THEN

SET expected\_role = 'MANAGER';

ELSE

SET expected\_role = 'UNKNOWN ROLE'; ## In case an experience level doesn't match the standard

END IF;

IF expected\_role = ROLE THEN

RETURN 'MATCH';

ELSE

RETURN CONCAT('MISMATCH (Expected: ', expected\_role, ')');

END IF;

RETURN expected\_role;

END

**QUERY**

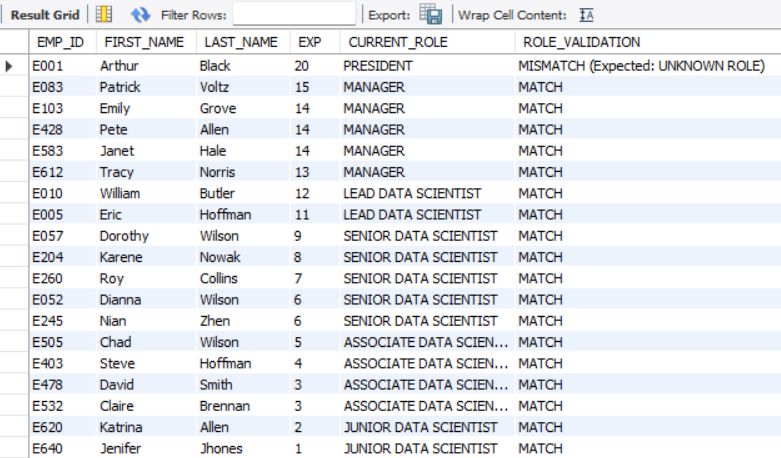
SELECT DS.EMP\_ID, DS.FIRST\_NAME, DS.LAST\_NAME, DS.EXP, DS.ROLE AS CURRENT\_ROLE,

Check\_Job\_Profile(DS.EXP, DS.ROLE) AS ROLE\_VALIDATION

FROM emp\_record\_table DS

order by EXP desc, CURRENT\_ROLE asc;

**OUTPUT**

****

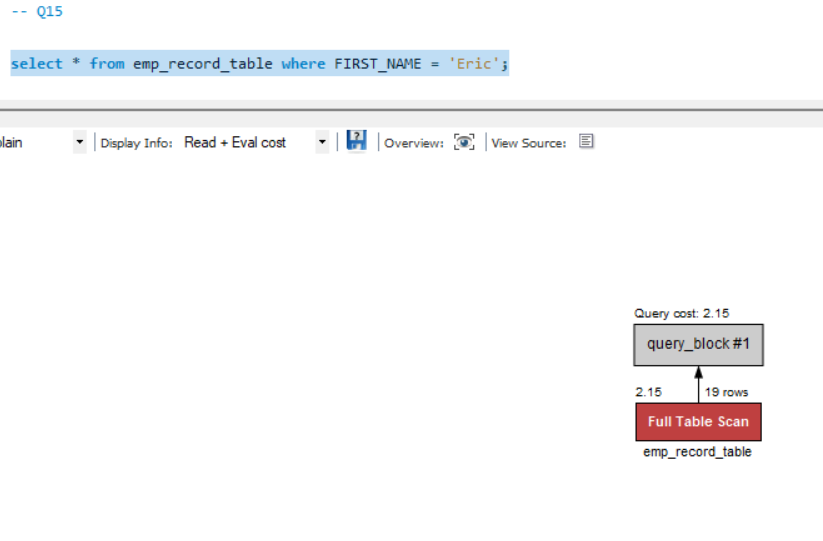
**Action 15: -**

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.

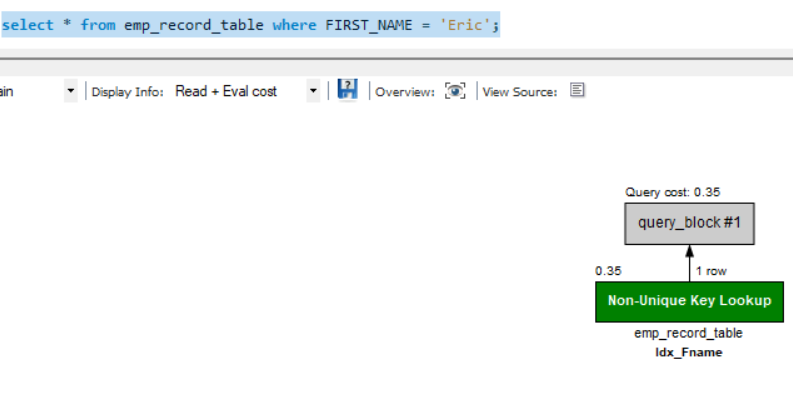
**Answer –**

**select \* from emp\_record\_table where FIRST\_NAME = 'Eric';**

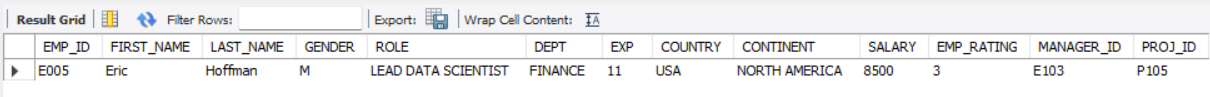
**BEFORE**

****

**AFTER**

****

**OUTPUT**

****

**Action 16: -**

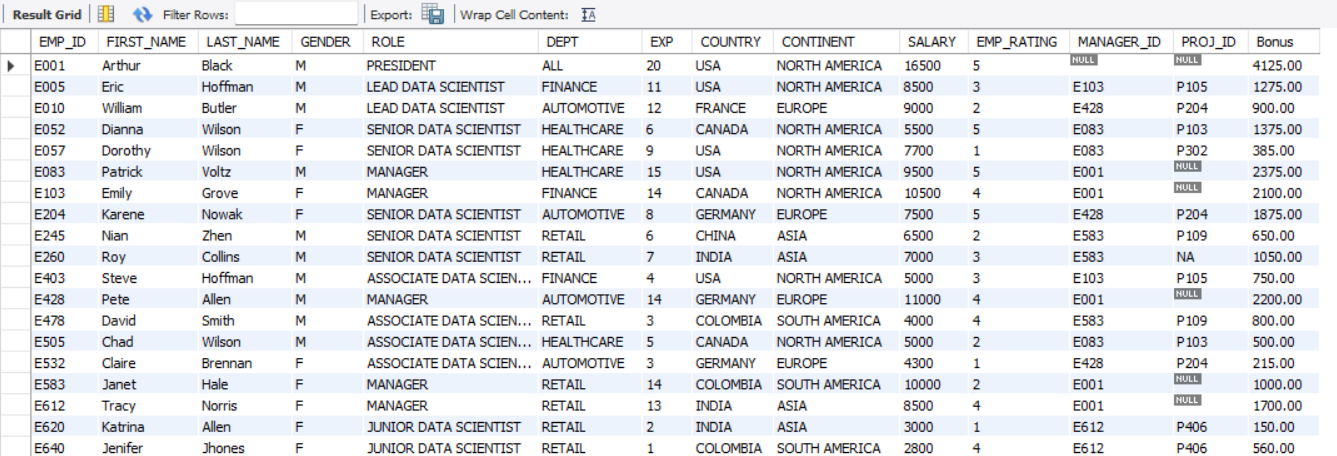
Write a query to calculate the bonus for all the employees, based on their ratings and salaries (Use the formula: 5% of salary \* employee rating).

**Answer –**

select \*,

SALARY \*.05 \* EMP\_RATING as Bonus from emp\_record\_table;

**OUTPUT**

****

**Action 17: -**

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

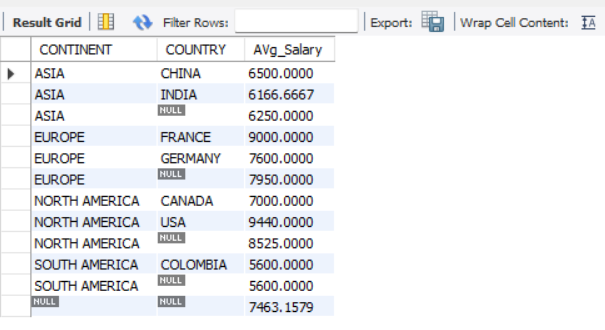
**Answer –**

select CONTINENT, COUNTRY, avg(Salary) as AVg\_Salary

from emp\_record\_table

group by CONTINENT, COUNTRY with rollup;

**OUTPUT**

****