by Ricka Johan

# ScienceQtech Employee Performance Mapping

**Background:**
ScienceQtech is a startup that works in the Data Science field for multiple industries such as fraud detection, market basket, supply chain, customer sentiment, and the drug discovery field.

**Objective:**
Find the maximum salary of the employees and ensure that all jobs are meeting the organization's profile standard, and calculate bonuses to find extra cost for expenses. This will raise the overall staff performance of the organization.

**DATASET DESCRIPTION:**

**emp_record_table:**

- 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:**

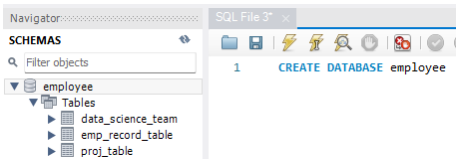
- PROJECT_ID – ID for the project
- PROJ_Name – Name of the project
- DOMAIN – Field of the project
- START_DATE – Day the project began
- CLOSURE_DATE – Day the project was or will be completed
- DEV_QTR – Quarter in which the project was scheduled
- STATUS – Status of the project currently

**Data_science_team:**

- 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

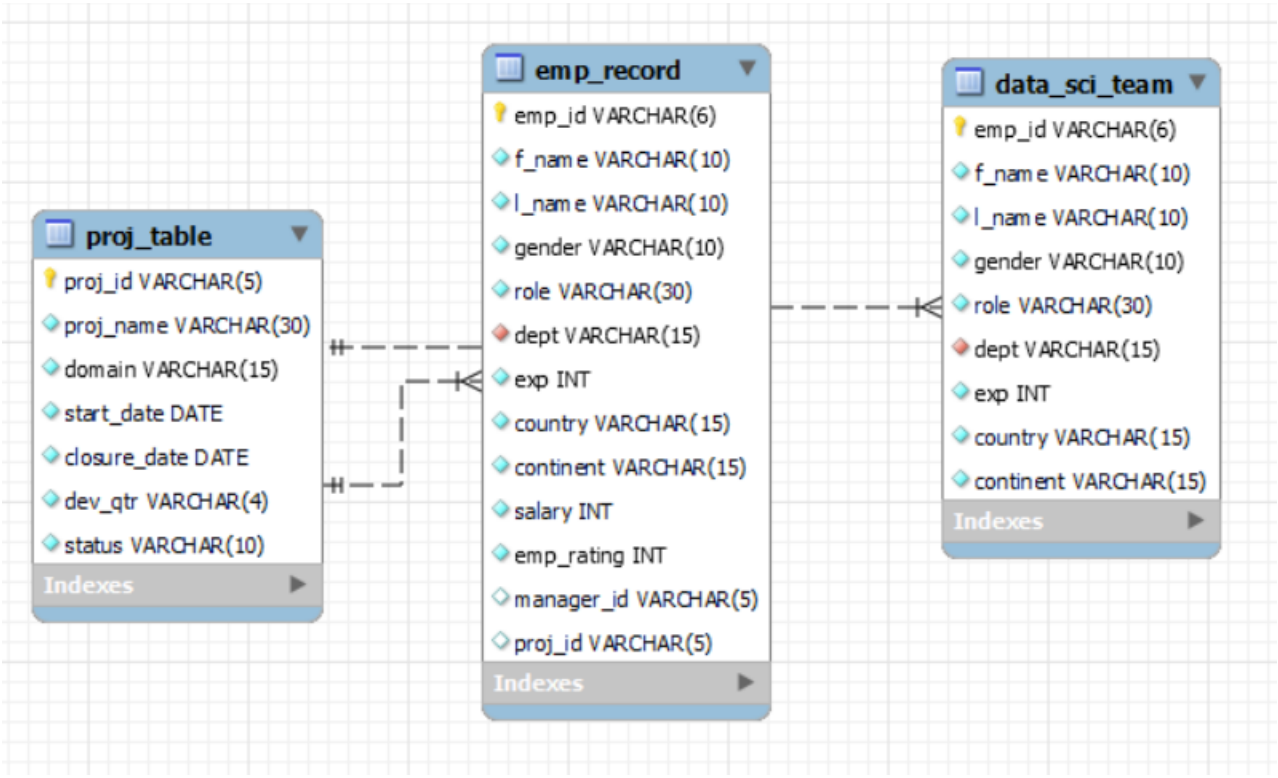
## SQL Tasks:

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.**

**ScienceQTech HR Entity Relation Diagram:**

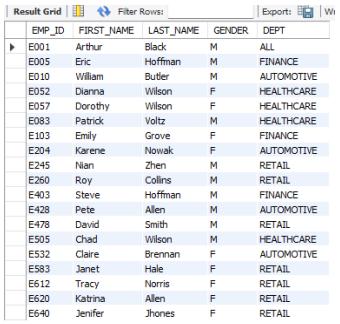


3. **Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department.**

SELECT emp_id, f_name, l_name, gender, dept

FROM emp_record;

**Output:**



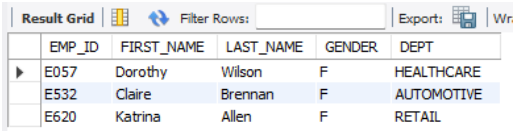
| EMP_ID | FIRST_NAME | LAST_NAME | GENDER | DEPT |
|---|---|---|---|---|
| E001 | Arthur | Black | M | ALL |
| E005 | Eric | Hoffman | M | FINANCE |
| E010 | William | Butler | M | AUTOMOTIVE |
| E052 | Dianna | Wilson | F | HEALTHCARE |
| E057 | Dorothy | Wilson | F | HEALTHCARE |
| E083 | Patrick | 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 |

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*

SELECT emp_id, f_name, l_name, gender, dept, emp_rating FROM emp_record
WHE RE emp_rating < 2;

**Output:**

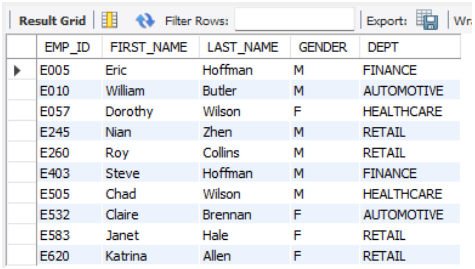


| EMP_ID | FIRST_NAME | LAST_NAME | GENDER | DEPT |
|---|---|---|---|---|
| E057 | Dorothy | Wilson | F | HEALTHCARE |
| E532 | Claire | Brennan | F | AUTOMOTIVE |
| E620 | Katrina | Allen | F | RETAIL |

- *greater than four*

SELECT emp_id, f_name, l_name, gender, dept, emp_rating FROM emp_record
WHER E emp_rating > 4;

**Output:**

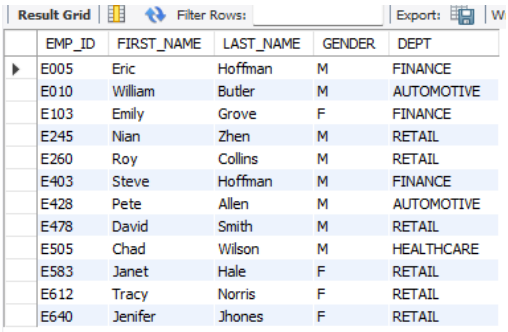


| EMP_ID | FIRST_NAME | LAST_NAME | GENDER | DEPT |
|---|---|---|---|---|
| E005 | Eric | Hoffman | M | FINANCE |
| E010 | William | Butler | M | AUTOMOTIVE |
| E057 | Dorothy | Wilson | F | HEALTHCARE |
| E245 | Nian | Zhen | M | RETAIL |
| E260 | Roy | Collins | M | RETAIL |
| E403 | Steve | Hoffman | M | FINANCE |
| E505 | Chad | Wilson | M | HEALTHCARE |
| E532 | Claire | Brennan | F | AUTOMOTIVE |
| E583 | Janet | Hale | F | RETAIL |
| E620 | Katrina | Allen | F | RETAIL |

- *between two and four*

SELECT emp_id, f_name, l_name, gender, dept, emp_rating FROM emp_record
WHERE emp_rati ng BETWEEN 2 AND 4;

**Output:**

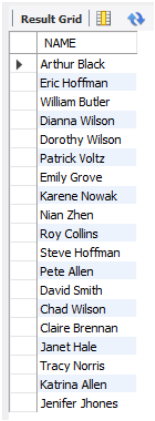


| EMP_ID | FIRST_NAME | LAST_NAME | GENDER | DEPT |
|---|---|---|---|---|
| E005 | Eric | Hoffman | M | FINANCE |
| E010 | William | Butler | M | AUTOMOTIVE |
| E103 | Emily | Grove | F | FINANCE |
| E245 | Nian | Zhen | M | RETAIL |
| E260 | Roy | Collins | M | RETAIL |
| E403 | Steve | Hoffman | M | FINANCE |
| E428 | Pete | Allen | M | AUTOMOTIVE |
| E478 | David | Smith | M | RETAIL |
| E505 | Chad | Wilson | M | HEALTHCARE |
| E583 | Janet | Hale | F | RETAIL |
| E612 | Tracy | Norris | F | RETAIL |
| E640 | Jenifer | Jhones | 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(f_name,' ',l_name) NAME FROM emp_record
WHERE dept = 'FINANCE';

**Output:**

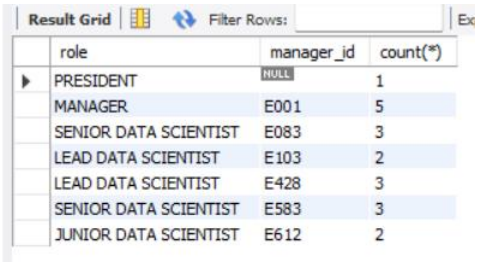


| NAME |
|---|
| Arthur Black |
| Eric Hoffman |
| William Butler |
| Dianna Wilson |
| Dorothy Wilson |
| Patrick Voltz |
| Emily Grove |
| Karene Nowak |
| Nian Zhen |
| Roy Collins |
| Steve Hoffman |
| Pete Allen |
| David Smith |
| Chad Wilson |
| Claire Brennan |
| Janet Hale |
| Tracy Norris |
| Katrina Allen |
| Jenifer Jhones |

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

```
SELECT role, manager_id, count(*)
FROM emp_record
GROUP BY manager_id
ORDER BY manager_id;
```

**Output:**

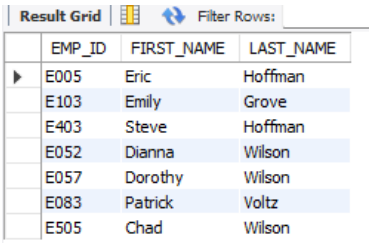


| role | manager_id | count(*) |
|---|---|---|
| PRESIDENT | NULL | 1 |
| MANAGER | E001 | 5 |
| SENIOR DATA SCIENTIST | E083 | 3 |
| LEAD DATA SCIENTIST | E103 | 2 |
| LEAD DATA SCIENTIST | E428 | 3 |
| SENIOR DATA SCIENTIST | E583 | 3 |
| JUNIOR DATA SCIENTIST | E612 | 2 |

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.**

```
SELECT f_name, l_name, dept FROM emp_record

WHERE dept = ‘HEALTHCARE’

UNION

SELECT f_name, l_name, dept FROM emp_record

WHERE dept = ‘FINANCE’;
```

**Output:**



| EMP_ID | FIRST_NAME | LAST_NAME |
|---|---|---|
| E005 | Eric | Hoffman |
| E103 | Emily | Grove |
| E403 | Steve | Hoffman |
| E052 | Dianna | Wilson |
| E057 | Dorothy | Wilson |
| E083 | Patrick | Voltz |
| E505 | Chad | Wilson |

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.**

```
SELECT emp_id, f_name, l_name, role, dept, emp_rating, emp_rating AS max_rating

FROM emp_record

WHERE (dept, emp_rating)

IN (SELECT dept, MAX(emp_rating) FROM emp_record GROUP By dept)
ORDER BY dept ASC;
```

**Output:**

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| DEPT | EMP_ID | FIRST_NAME | LAST_NAME | ROLE | EMP_RATING | MAX(EMP_RATING) |
|---|---|---|---|---|---|---|
| ALL | E001 | Arthur | Black | PRESIDENT | 5 | 5 |
| FINANCE | E005 | Eric | Hoffman | LEAD DATA SCIENTIST | 3 | 4 |
| AUTOMOTIVE | E010 | William | Butler | LEAD DATA SCIENTIST | 2 | 5 |
| HEALTHCARE | E052 | Dianna | Wilson | SENIOR DATA SCIENTIST | 5 | 5 |
| RETAIL | E245 | Nian | Zhen | SENIOR DATA SCIENTIST | 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.**

```
SELECT role, MIN(salary) AS minSalary, MAX(salary) AS maxSalary
FROM emp_record
GROUP BY role;
```

**Output:**

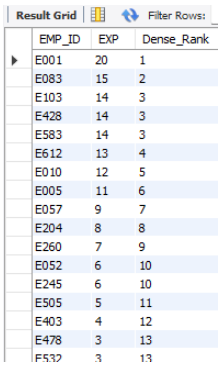
Result Grid | Filter Rows: | Export:

| MAX(SALARY) | MIN(SALARY) | ROLE |
|---|---|---|
| 16500 | 16500 | PRESIDENT |
| 9000 | 8500 | LEAD DATA SCIENTIST |
| 7700 | 5500 | SENIOR DATA SCIENTIST |
| 11000 | 8500 | MANAGER |
| 5000 | 4000 | ASSOCIATE DATA SCIENTIST |
| 3000 | 2800 | JUNIOR DATA SCIENTIST |

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

```
SELECT f_name, l_name, exp as experience,
DENSE_RANK() OVER (ORDER BY exp DESC) exp_rank
FROM emp_record;
```

**Output:**

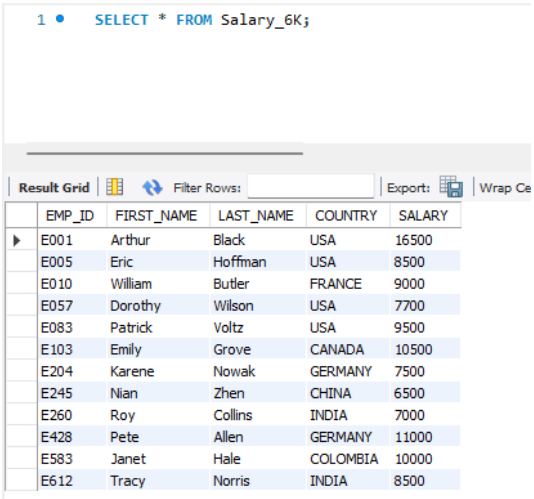


| EMP_ID | EXP | Dense_Rank |
|---|---|---|
| E001 | 20 | 1 |
| E083 | 15 | 2 |
| E103 | 14 | 3 |
| E428 | 14 | 3 |
| E583 | 14 | 3 |
| E612 | 13 | 4 |
| E010 | 12 | 5 |
| E005 | 11 | 6 |
| E057 | 9 | 7 |
| E204 | 8 | 8 |
| E260 | 7 | 9 |
| E052 | 6 | 10 |
| E245 | 6 | 10 |
| E505 | 5 | 11 |
| E403 |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

**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.**

```
CREATE VIEW 6K_salary AS
SELECT emp_id, f_name, l_name, country, salary
FROM emp_record
WHERE salary > 6000;

SELECT * FROM 6k_salary;
```

**Output:**



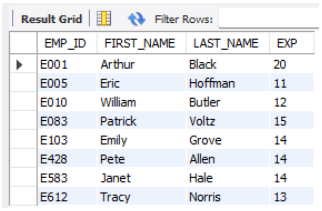
```
1 •  SELECT * FROM Salary_6K;
```

| 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 |

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

```
SELECT emp_id, f_name, l_name, exp
FROM emp_record
WHERE exp IN (
SELECT exp
FROM emp_record
WHERE exp > 10);
```

**Output:**

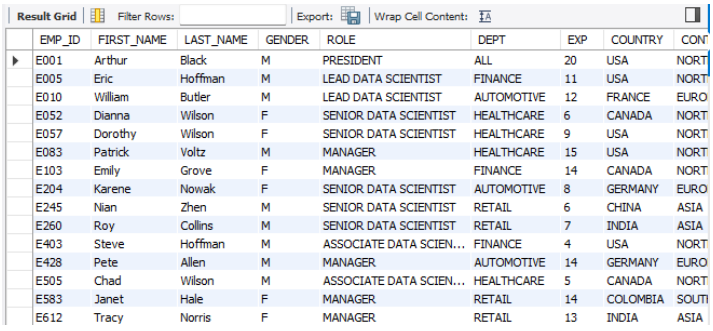


| EMP_ID | FIRST_NAME | LAST_NAME | EXP |
|---|---|---|---|
| E001 | Arthur | Black | 20 |
| E005 | Eric | Hoffman | 11 |
| E010 | William | Butler | 12 |
| E083 | Patrick | Voltz | 15 |
| E103 | Emily | Grove | 14 |
| E428 | Pete | Allen | 14 |
| E583 | Janet | Hale | 14 |
| E612 | Tracy | Norris |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

**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.**

```
DELIMITER //
CREATE PROCEDURE Employee3()
BEGIN
SELECT * FROM emp_record
WHERE exp > 3;
END //
DELIMITER ;

CALL Employee3;
```

**Output:**



| EMP_ID | FIRST_NAME | LAST_NAME | GENDER | ROLE | DEPT | EXP | COUNTRY | CON |
|---|---|---|---|---|---|---|---|---|
| E001 | Arthur | Black | M | PRESIDENT | ALL | 20 | USA | NORT |
| E005 | Eric | Hoffman | M | LEAD DATA SCIENTIST | FINANCE | 11 | USA | NORT |
| E010 | William | Butler | M | LEAD DATA SCIENTIST | AUTOMOTIVE | 12 | FRANCE | EURO |
| E052 | Dianna | Wilson | F | SENIOR DATA SCIENTIST | HEALTHCARE | 6 | CANADA | NORT |
| E057 | Dorothy | Wilson | F | SENIOR DATA SCIENTIST | HEALTHCARE | 9 | USA | NORT |
| E083 | Patrick | Voltz | M | MANAGER | HEALTHCARE | 15 | USA | NORT |
| E103 | Emily | Grove | F | MANAGER | FINANCE | 14 | CANADA | NORT |
| E204 | Karene | Nowak | F | SENIOR DATA SCIENTIST | AUTOMOTIVE | 8 | GERMANY | EURO |
| E245 | Nian | Zhen | M | SENIOR DATA SCIENTIST | RETAIL | 6 | CHINA | ASIA |
| E260 | Roy | Collins | M | SENIOR DATA SCIENTIST | RETAIL | 7 | INDIA | ASIA |
| E403 | Steve | Hoffman | M | ASSOCIATE DATA SCIEN... | FINANCE | 4 | USA | NORT |
| E428 | Pete | Allen | M | MANAGER | AUTOMOTIVE | 14 | GERMANY | EURO |
| E505 | Chad | Wilson | M | ASSOCIATE DATA SCIEN... | HEALTHCARE | 5 | CANADA | NORT |
| E583 | Janet | Hale | F | MANAGER | RETAIL | 14 | COLOMBIA | SOUT |
| E612 | Tracy | Norris | F | MANAGER | RETAIL | 13 | INDIA | ASIA |

**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',

```
DELIMITER //
CREATE PROCEDURE check_role()
BEGIN
        SELECT * FROM emp_record
        CASE
                WHEN exp <= 2 THEN SET role = ‘JUNIOR DATA SCIENTIST’;
                WHEN exp BETWEEN 3 AND 5 THEN SET role = ‘ASSOCIATE DATA SCIENTIST’;
                WHEN exp BETWEEN 6 AND 10 THEN SET role = ‘SENIOR DATA SCIENTIST’;
                WHEN exp BETWEEN 11 AND 12 THEN SET role = ‘LEAD DATA SCIENTIST’;
                WHEN exp BETWEEN 13 AND 16 THEN SET role = ‘MANAGER’;
ELSE SET role = ‘all good’;
END CASE;
END //
DELIMITER ;
```

**Output:**

```
1 •  SELECT EMP_ID, data_science_team.ROLE, EXP, ExpStandard(Role, EXP) FROM data_science_team;
```

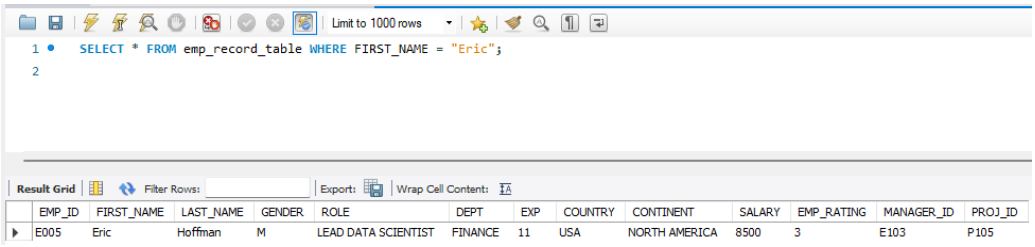
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| EMP_ID | ROLE | EXP | ExpStandard(Role, EXP) |
|---|---|---|---|
| E005 | LEAD DATA SCIENTIST | 11 | LEAD DATA SCIENTIST |
| E010 | LEAD DATA SCIENTIST | 12 | LEAD DATA SCIENTIST |
| E052 | SENIOR DATA SCIENTIST | 6 | SENIOR DATA SCIENTIST |
| E057 | SENIOR DATA SCIENTIST | 9 | SENIOR DATA SCIENTIST |
| E204 | SENIOR DATA SCIENTIST | 8 | SENIOR DATA SCIENTIST |
| E245 | SENIOR DATA SCIENTIST | 6 | SENIOR DATA SCIENTIST |
| E260 | SENIOR DATA SCIENTIST | 7 | SENIOR DATA SCIENTIST |
| E403 | ASSOCIATE DATA SCIENTIST | 4 | ASSOCIATE DATA SCIENTIST |
| E478 | ASSOCIATE DATA SCIENTIST | 3 | ASSOCIATE DATA SCIENTIST |
| E505 | ASSOCIATE DATA SCIENTIST | 5 | ASSOCIATE DATA SCIENTIST |
| E532 | ASSOCIATE DATA SCIENTIST | 3 | ASSOCIATE DATA SCIENTIST |
| E620 | JUNIOR DATA SCIENTIST | 2 | JUNIOR DATA SCIENTIST |
| E640 | JUNIOR DATA SCIENTIST | 1 | JUNIOR 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.**

ALTER TABLE emp_record ADD INDEX fname_index (f_name);
SELECT * FROM emp_record WHERE f_name = 'Eric';

**Output:**



```
SELECT * FROM emp_record_table WHERE FIRST_NAME = "Eric";
```

| EMP_ID | FIRST_NAME | LAST_NAME | GENDER | ROLE | DEPT | EXP | COUNTRY | CONTINENT | SALARY | EMP_RATING | MANAGER_ID | PROJ_ID |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E005 | Eric | Hoffman | M | LEAD DATA SCIENTIST | FINANCE | 11 | USA | NORTH AMERICA | 8500 | 3 | E103 | P105 |

**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 first_name, last_name, ((SALARY*0.05)*emp_rating) AS BONUS
FROM emp_record_table;

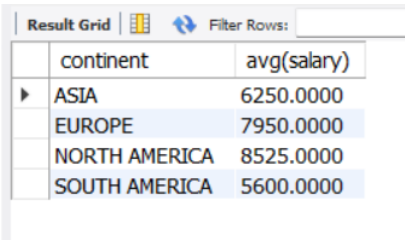
**Output:**

| f_name | l_name | salary | bonus |
|---|---|---|---|
| Arthur | Black | 16500 | 4125.00 |
| Eric | Hoffman | 8500 | 1275.00 |
| William | Butler | 9000 | 900.00 |
| Dianna | Wilson | 5500 | 1375.00 |
| Dorothy | Wilson | 7700 | 385.00 |
| Patrick | Voltz | 9500 | 2375.00 |
| Emily | Grove | 10500 | 2100.00 |
| Karene | Nowak | 7500 | 1875.00 |
| Nian | Zhen | 6500 | 650.00 |
| Roy | Collins | 7000 | 1050.00 |
| Steve | Hoffman | 5000 | 750.00 |
| Pete | Allen | 11000 | 2200.00 |
| David | Smith | 4000 | 800.00 |
| Chad | Wilson | 5000 | 500.00 |
| Claire | Brennan | 4300 | 215.00 |
| Janet | Hale | 10000 | 1000.00 |
| Tracy | Norris | 8500 | 1700.00 |
| Katrina | Allen | 3000 | 150.00 |
| Jenifer | Jhones | 2800 | 560.00 |

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

```
SELECT continent, AVG(salary)
FROM emp_record
GROUP BY continent
ORDER BY continent ASC;
```

**Output:**



Result Grid | Filter Rows:

| continent | avg(salary) |
|---|---|
| ASIA | 6250.0000 |
| EUROPE | 7950.0000 |
| NORTH AMERICA | 8525.0000 |
| SOUTH AMERICA | 5600.0000 |