Project name: Science Qtech Employee Performance Mapping

Table of content

1. Introduction

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.

2. 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.

3. Sql code

```
1.create database employee;


3. SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
FROM emp_record_table;
SELECT * FROM emp_record_table;

4. SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM emp_record_table
WHERE EMP_RATING < 2;

 SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM emp_record_table
WHERE EMP_RATING >4;

SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM emp_record_table
WHERE EMP_RATING BETWEEN 2 AND 4;

5. SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS NAME
FROM emp_record_table
WHERE DEPT = 'Finance';
6. SELECT *
FROM emp_record_table
WHERE emp_id IN (
   SELECT DISTINCT manager_id
   FROM emp_record_table
```

```
   WHERE manager_id IS NOT NULL
);
```

7.
```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT
FROM emp_record_table
WHERE DEPT = "HEALTHCARE"
UNION
SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT
FROM emp_record_table
WHERE DEPT = "FINANCE"
```

8.
```
SELECT dept, emp_id, first_name, last_name, role, emp_rating, MAX(emp_rating) OVER (PARTITION BY dept) AS max_emp_rating
FROM emp_record_table
GROUP BY dept, emp_id, first_name, last_name, role, emp_rating;
```

9.
```
SELECT role, MIN(salary) AS min_salary, MAX(salary) AS max_salary
FROM emp_record_table
GROUP BY role;
```

10.
```
SELECT emp_id,FIRST_NAME,DEPT, exp, RANK() OVER (ORDER BY exp DESC)
FROM emp_record_table;
```

11.
```
CREATE VIEW employee_salary_view AS
SELECT *
FROM emp_record_table
WHERE salary > 6000
GROUP BY country;
```

12. SELECT *

FROM emp_record_table

WHERE exp > 10;

13. DELIMITER &&

CREATE PROCEDURE sp_employee_experience()

BEGIN

SELECT *

FROM emp_record_table

WHERE exp > 3;

END;

CALL sp_employee_experience()

14. DELIMITER &&

CREATE FUNCTION EMPLOYEE_ROLE_MATCH (EXP INT) RETURNS VARCHAR (50) DETERMINISTIC

BEGIN

DECLARE STANDARD_ROLE VARCHAR (50);

IF EXP < 2 THEN

SET STANDARD_ROLE ='JUNIOR DATA SCIENTIST';

ELSEIF (EXP >=2 and EXP <5) THEN

SET STANDARD_ROLE ='ASSOCIATE DATA SCIENTIST';

ELSEIF (EXP >=5 and EXP <10) THEN

SET STANDARD_ROLE ='SENIOR DATA SCIENTIST';

ELSEIF (EXP >=10 and EXP <12) THEN

SET STANDARD_ROLE ='LEAD DATA SCIENTIST';

ELSEIF (EXP >=12 and EXP <16) THEN

SET STANDARD_ROLE ='MANAGER';

```
    END IF;
    RETURN (STANDARD_ROLE);
END &&
```

15.
```
SELECT EMP.EMP_ID, DS.ROLE, EMP.EXP, EMPLOYEE_ROLE_MATCH(EMP.EXP)
FROM emp_record_table EMP
JOIN data_science_team DS ON EMP.EMP_ID = DS.EMP_ID;
```

16.
```
CREATE INDEX idx_employee_firstname ON emp_record_table (FIRST_NAME);
```

17.
```
SELECT EMP_ID, SALARY, emp_rating, 0.05 * SALARY * emp_rating AS bonus
FROM emp_record_table;
```

18.
```
SELECT
    continent,
    country,
    AVG(salary) AS avg_salary
FROM
    emp_record_table
GROUP BY
    continent,
    country;
```