Scienceqtech Employee Performance Mapping Project Solution:

Description:

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 analyse the employee database and extract specific data based on different requirements.

Tasks to be performed:

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

- 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.
- 10. Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.
- 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.
- 12. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.
- 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.
- 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'.

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

Project Solution:

CREATE DATABASE employee;

USE EMPLOYEE;

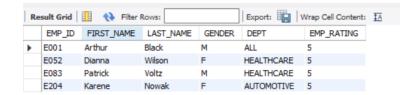
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
FROM emp_record_table;



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

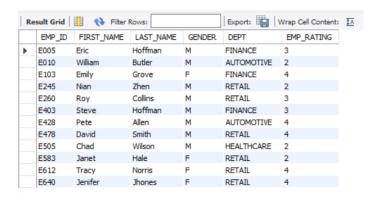


SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT,EMP_RATING
FROM emp_record_table
WHERE EMP_RATING > 4;

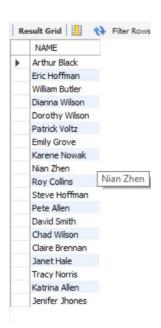


SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT,EMP_RATING FROM emp_record_table

WHERE EMP_RATING BETWEEN 2 AND 4;



SELECT concat(FIRST_NAME,' ',LAST_NAME) AS NAME FROM emp_record_table;



SELECT *

from emp_record_table WHERE (DEPT = 'healthcare')

UNION

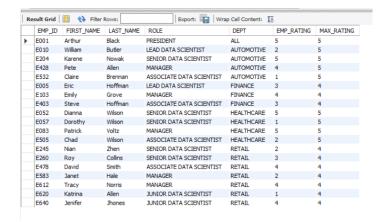
SELECT *

from emp_record_table

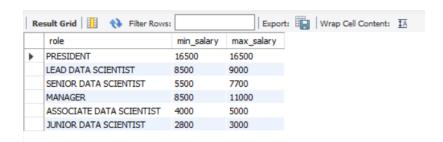
WHERE (DEPT = 'finance');



SELECT EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EMP_RATING,
MAX(EMP_RATING) OVER (PARTITION BY DEPT) AS MAX_RATING
FROM emp_record_table;



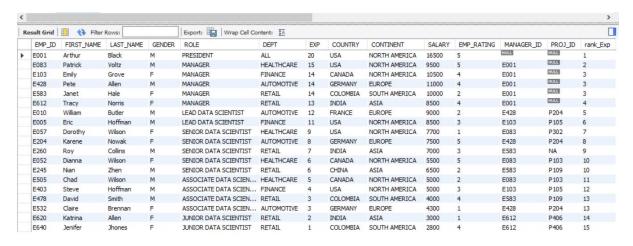
SELECT role, MIN(salary) AS min_salary, MAX(salary) AS max_salary FROM emp_record_table GROUP BY role;



SELECT *,

DENSE_RANK() OVER (ORDER BY EXP DESC) AS rank_Exp

FROM emp_record_table;



DROP VIEW empl_salary;

CREATE VIEW empl_salary AS SELECT EMP_ID, FIRST_NAME, LAST_NAME, SALARY, COUNTRY

FROM emp_record_table

WHERE SALARY > 6000;

SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP FROM emp_record_table

WHERE (SELECT EXP > 10);



DROP PROCEDURE emp_exp_Three;

DELIMITER &&

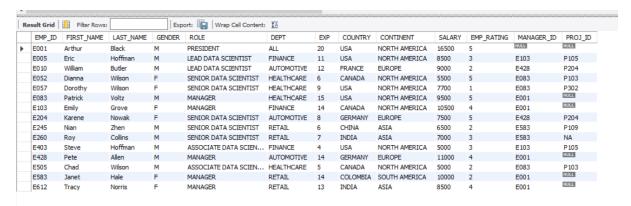
CREATE PROCEDURE emp_exp_Three ()

BEGIN

SELECT * FROM emp_record_table WHERE EXP > 3;

END &&

CALL emp_exp_Three ();



DROP FUNCTION emrol;

DELIMITER //

CREATE FUNCTION emrol(EXP INT)

RETURNS VARCHAR(40)

DETERMINISTIC

BEGIN

DECLARE emp_role VARCHAR(40);

IF EXP BETWEEN 12 AND 16 THEN

SET emp_role = 'MANAGER';

ELSEIF EXP BETWEEN 10 AND 12 THEN

SET emp_role = 'LEAD_DATA_SCIENTIST';

ELSEIF EXP BETWEEN 5 AND 10 THEN

SET emp_role ='SENIOR_DATA_SCIENTIST';

ELSEIF EXP BETWEEN 2 AND 5 THEN

SET emp_role ='ASSOCIATE_DATA_SCIENTIST';

ELSEIF EXP<= 2 THEN

SET emp_role ='JUNOR_DATA_SCIENTIST';

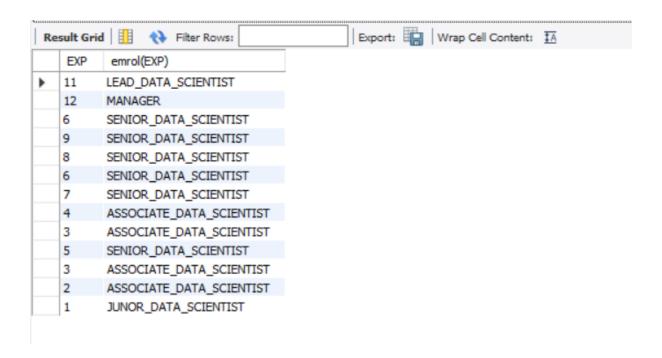
END IF;

RETURN (emp_role);

END //

SELECT EXP, emrol(EXP)

FROM data_science_team;



DROP INDEX indx ON emp_record_table;

CREATE INDEX indx ON emp_record_table (FIRST_NAME(40));

EXPLAIN SELECT EMP_ID, FIRST_NAME, LAST_NAME FROM emp_record_table WHERE

FIRST NAME = 'ERIC';

FROM emp_record_table;



SELECT EMP_ID, FIRST_NAME, LAST_NAME, SALARY, EMP_RATING,
(SALARY + SALARY*0.05* EMP_RATING) AS BONUS_SAL



SELECT EMP_ID,FIRST_NAME,LAST_NAME,COUNTRY,CONTINENT,

AVG(SALARY) OVER (PARTITION BY COUNTRY) AS avg_sal_cntry,

AVG(SALARY) OVER (PARTITION BY CONTINENT) AS avg_sal_cont,

COUNT(*) OVER (PARTITION BY COUNTRY) AS count_in_country,

COUNT(*) OVER (PARTITION BY CONTINENT) AS count_in_continent

FROM emp_record_table;

