DESCRIPTION

Science Q-Tech Employee Performance Mapping.

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 is in need of generating 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.

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.

Dataset description:

emp\_record\_table: It contains the information of all the employees.

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: It contains information about the projects.

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: It contains information about all the employees in the 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

data\_science\_team



emp\_record\_table



proj\_table

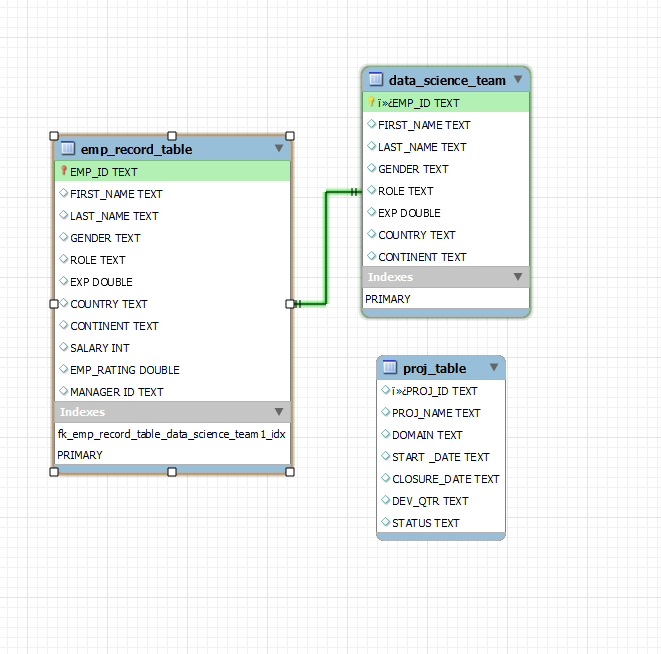


1. Create a database named employee, then import **data\_science\_team.csv** **proj\_table.csv** and **emp\_record\_table.csv** into the **employee**database from the given resources.

CREATE DATABASE employee;

USE employee;

1. Create an ER diagram for the given **employee**database.



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.

SELECT \* FROM emp\_record\_table;

ALTER TABLE employee.emp\_record\_table RENAME COLUMN ï»¿EMP\_ID TO EMP\_ID;

SELECT \* FROM emp\_record\_table;

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

ORDER BY DEPT,FIRST\_NAME,LAST\_NAME;

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

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING FROM employee.emp\_record\_table WHERE EMP\_RATING <2;

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING FROM employee.emp\_record\_table WHERE EMP\_RATING >4;

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING FROM employee.emp\_record\_table WHERE EMP\_RATING >=2 AND EMP\_RATING <=4 ORDER BY EMP\_RATING;

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.

SELECT concat(FIRST\_NAME, ' ',LAST\_NAME) AS NAME FROM employee.emp\_record\_table WHERE DEPT = 'FINANCE';

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

SELECT A.`MANAGER ID`,count(A.`MANAGER ID`)AS REPORTERS, concat(b.FIRST\_NAME, ' ',B.LAST\_NAME) AS NAME FROM employee.emp\_record\_table AS A JOIN employee.emp\_record\_table AS B

ON A.`MANAGER ID` = B.EMP\_ID

GROUP BY A.`MANAGER ID` ORDER BY A.`MANAGER ID` ;

1. 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 \* FROM employee.emp\_record\_table AS FIN\_DATA WHERE DEPT = 'FINANCE'

UNION All

SELECT \* FROM employee.emp\_record\_table AS HC\_DATA WHERE DEPT = 'HEALTHCARE'ORDER BY EMP\_ID;

#SELECT \* FROM employee.emp\_record\_table AS FIN\_DATA WHERE DEPT IN ( 'FINANCE','HEALTHCARE')

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.

SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,`ROLE`,DEPT,EMP\_RATING,MAX(EMP\_RATING) OVER (PARTITION BY DEPT) AS MAX\_RATING,CONCAT(EMP\_RATING , "/", MAX(EMP\_RATING) OVER (PARTITION BY DEPT) )AS EFF,REPEAT("\*", EMP\_RATING) AS STARS

FROM

employee.emp\_record\_table

ORDER BY EMP\_RATING DESC;

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.

SELECT   
`ROLE`, MIN(SALARY), MAX(SALARY)   
FROM employee.emp\_record\_table   
WHERE `ROLE`!= 'PRESIDENT'  
GROUP BY `ROLE`;

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

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EXP,

RANK() OVER(ORDER BY EXP DESC) AS 'Rank'

FROM employee.emp\_record\_table

ORDER BY 'Rank';

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.

CREATE VIEW V\_COUNTRY\_SAL AS

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EXP,COUNTRY,SALARY

FROM employee.emp\_record\_table

WHERE SALARY>6000;

SELECT \* FROM V\_COUNTRY\_SAL;

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

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, EXP, DEPT, EMP\_RATING FROM employee.emp\_record\_table WHERE EMP\_ID IN (SELECT EMP\_ID FROM employee.emp\_record\_table WHERE EXP >10);

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.

DELIMITER //

CREATE PROCEDURE EMP\_DETAILS()

BEGIN

SELECT \* FROM employee.emp\_record\_table WHERE EXP>3;

END //

DELIMITER //;enecccbettbglghiuukeegdcbhvdrrenbjfgcnrlngfi

CALL EMP\_DETAILS();

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'.

DELIMITER $$ ;

ALTER FUNCTION Role\_VERIFY (

EXP double

)

RETURNS VARCHAR(200)

DETERMINISTIC

BEGIN

DECLARE Role\_Validation VARCHAR(200);

IF EXP<=2 THEN

SET Role\_Validation = 'JUNIOR DATA SCIENTIST';

ELSEIF (EXP>=2 AND

EXP<=5) THEN

SET Role\_Validation = 'ASSOCIATE DATA SCIENTIST';

ELSEIF (EXP>=5 AND

EXP<=10) THEN

SET Role\_Validation = 'SENIOR DATA SCIENTIST';

ELSEIF (EXP>=10 AND

EXP<=12) THEN

SET Role\_Validation = 'LEAD DATA SCIENTIST';

ELSEIF (EXP>=12 AND

EXP<=16) THEN

SET Role\_Validation = 'MANAGER';

END IF;

-- return the customer occupation

RETURN (Role\_Validation);

END $$

DELIMITER $$;

SELECT \*,

CASE ROLE WHEN Role\_VERIFY(EXP)

THEN 'VALID' ELSE 'INVALID'

END AS VALIDATION

FROM employee.data\_science\_team;

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.

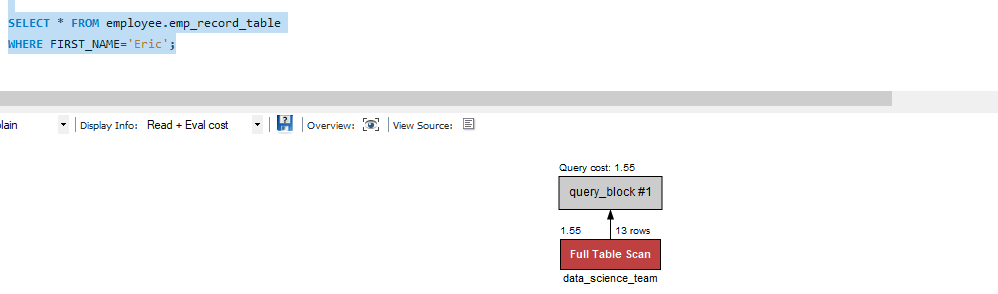
ALTER TABLE employee.emp\_record\_table DROP INDEX idx\_first\_name;

CREATE INDEX idx\_first\_name

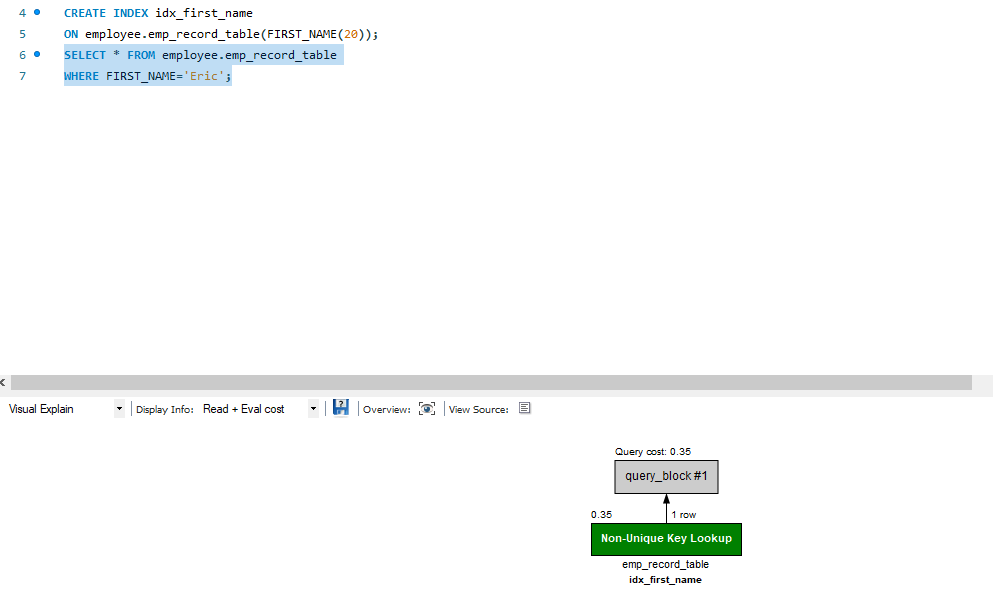
ON employee.emp\_record\_table(FIRST\_NAME(20));

SELECT \* FROM employee.emp\_record\_table

WHERE FIRST\_NAME='Eric';



Query cost before creating the index taking FIRST\_NAME is 1.55

.

Query cost decreased by 1.2 to get the final query cost as 0.35

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).

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, EXP, DEPT, EMP\_RATING, 0.05\*SALARY\*EMP\_RATING AS BONUS FROM employee.emp\_record\_table ORDER BY BONUS DESC;

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

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, EXP, DEPT, EMP\_RATING, AVG(SALARY) AS AVG\_SAL FROM employee.emp\_record\_table GROUP BY CONTINENT,COUNTRY;