SQL Training

Course-End Project Problem Statement

ScienceQtech Employee Performance Mapping

Problem scenario:

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.

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.

Note: You must download the dataset from the course resource section in LMS and create a table to perform the above objective.

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

The task to be performed:

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.

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.

Assessment Started

```
# Creating Employee Database:
Create database Employee;

use Employee;

# Created Data_Science Table:
Create Table Data_Science (
EMP_ID varchar(255),
FIRST_NAME varchar(255),
LAST_NAME varchar(255),
GENDER varchar(255),
ROLE varchar(255),
DEPT varchar(255),
EXP int,
COUNTRY varchar(255),
CONTINENT varchar(255)
);

# Importing Dataset in Data_Science Table
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/data_science_team.csv"
INTO TABLE Data_Science
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

# Ctreated Projects Table:
Create Table Projects (
PROJECT_ID varchar(255),
PROJ_NAME varchar(255),
DOMAIN varchar(255),
START_DATE Date,
CLOSURE_DATE Date,
DEV_QTR varchar(255),
STATUS varchar(255)
);

# Importing Dataset in Projects Table
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Project Table.csv"
INTO TABLE Projects
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

```
# Created Emp_Record Table:
Create Table Emp_Record (
EMP_ID varchar(255),
FIRST_NAME varchar(255),
LAST_NAME varchar(255),
GENDER varchar(255),
ROLE varchar(255),
DEPT varchar(255),
EXP int,
COUNTRY varchar(255),
CONTINENT varchar(255),
SALARY int,
EMP_RATING int,
MANAGER_ID varchar(255),
PROJ_ID varchar(255)
);

# Importing Dataset in Emp_Record Table
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/emp_record_table.csv"
INTO TABLE Emp_Record
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

# Questions & Their Queries:

-- 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
   EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
FROM
   Emp_Record;
```

-- 2. **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,
   CASE
      WHEN EMP_RATING < 2 THEN 'Rating less than 2'
      WHEN EMP_RATING > 4 THEN 'Rating greater than 4'
      WHEN EMP_RATING BETWEEN 2 AND 4 THEN 'Rating between 2 and 4'
   END AS RATING_CATEGORY
FROM
   Emp_Record;
```

-- 3. **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
   Emp_Record
WHERE
   DEPT = 'FINANCE';
```

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

```
SELECT
   m.EMP_ID,
   m.FIRST_NAME,
   m.LAST_NAME,
   COUNT(e.EMP_ID) AS NUM_REPORTERS
FROM
   Emp_Record m
      JOIN
   Emp_Record e ON m.EMP_ID = e.MANAGER_ID
GROUP BY m.EMP_ID , m.FIRST_NAME , m.LAST_NAME
ORDER BY NUM_REPORTERS DESC;
```

-- 5. **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
   EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
FROM
   Emp_Record
WHERE
   DEPT = 'FINANCE'

UNION SELECT

   EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
FROM
   Emp_Record
WHERE
   DEPT = 'HEALTHCARE';
```

-- 6. **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_Dept_Rating
from Emp_Record
order by DEPT, EMP_RATING DESC;
```

-- 7. **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,
        DEPT,
   SALARY,
   Max(Salary) OVER(partition by DEPT) as Max_Salary,
   Min(Salary) OVER(partition by DEPT) as Min_Salary
from Emp_Record
order by DEPT;
```

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

```
SELECT
   FIRST_NAME,
   LAST_NAME,
   ROLE,
   DEPT,
   EXP,
   DENSE_RANK() OVER (ORDER BY EXP DESC) AS Exp_Rank
FROM Emp_Record
ORDER BY Exp_Rank;
```

-- 9. **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 Emp_Various_Countries AS
SELECT
   FIRST_NAME, LAST_NAME, GENDER, COUNTRY, SALARY
FROM
   Emp_Record
WHERE
   Salary > 6000;
```

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

```
SELECT
   FIRST_NAME, LAST_NAME, EXP
FROM
   Emp_Record
WHERE
   EMP_ID IN (SELECT
         EMP_ID
      FROM
         Emp_Record
      WHERE
         EXP > 10);
```

-- 11. **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_Exp_more_than_3_yrs()
BEGIN
Select
EMP_ID,
FIRST_NAME,
LAST_NAME,
EXP
from Emp_Record
Where EXP > 3;
END //
DELIMITER ;

CALL Emp_Exp_more_than_3_yrs ;
```

-- 12. 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 //
CREATE FUNCTION fn_get_standard_role(p_EXP int)
RETURNS varchar(50)
deterministic
BEGIN
        declare v_ROLE varchar(50);

   IF p_EXP <= 2 THEN
                SET v_ROLE = 'JUNIOR DATA SCIENTIST';
        ELSEIF p_EXP > 2 AND p_EXP <=5 THEN
                SET v_ROLE = 'ASSOCIATE DATA SCIENTIST';
        ELSEIF p_EXP > 5 AND p_EXP <= 10 THEN
                SET v_ROLE = 'SENIOR DATA SCIENTIST';
        ELSEIF p_EXP > 10 AND p_EXP <= 12 THEN
                SET v_ROLE = 'LEAD DATA SCIENTIST';
        ELSEIF p_EXP > 12 AND p_EXP <= 16 THEN
                SET v_ROLE = 'MANAGER';
        ELSE
                SET v_ROLE = 'OTHER'; -- Fallback
        END IF;

   RETURN v_ROLE;
END //

DELIMITER ;

SELECT
   EMP_ID,
   FIRST_NAME,
   LAST_NAME,
   EXP,
   ROLE AS Current_Role,
   fn_get_standard_role(EXP) AS Standard_Role,
   CASE
      WHEN ROLE = fn_get_standard_role(EXP)
      THEN 'MATCHED'
      ELSE 'MISMATCHED'
   END AS Validation_Status
FROM Emp_Record
WHERE DEPT = 'DATA SCIENCE';
```

-- 13. **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**

```
explain
Select EMP_ID, FIRST_NAME, LAST_NAME, DEPT, ROLE
FROM Emp_Record
Where FIRST_NAME = 'Eric';

CREATE INDEX idx_Firstname ON Emp_Record(FIRST_NAME);

explain
SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT, ROLE
from Emp_Record
Where FIRST_NAME = 'Eric';
```

-- 14. **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,
   ROLE,
   DEPT,
   EXP,
   SALARY,
   (0.05 * SALARY * EMP_RATING) AS BONUS
FROM
   Emp_Record;
```

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

```
SELECT
   CONTINENT,
   COUNTRY,
   AVG(SALARY) AS average_salary_distribution
FROM
   Emp_Record
GROUP BY
   CONTINENT, COUNTRY
ORDER BY
   CONTINENT, COUNTRY;
```