ScienceQtech Employee Performance Mapping.

Project 1

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 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 should also determine whether or not employees need a promotion, and 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
- PROJECT_ID – The project on which the employee is working or has worked on

Proj_table: It contains information about the projects.

- PROJ_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.

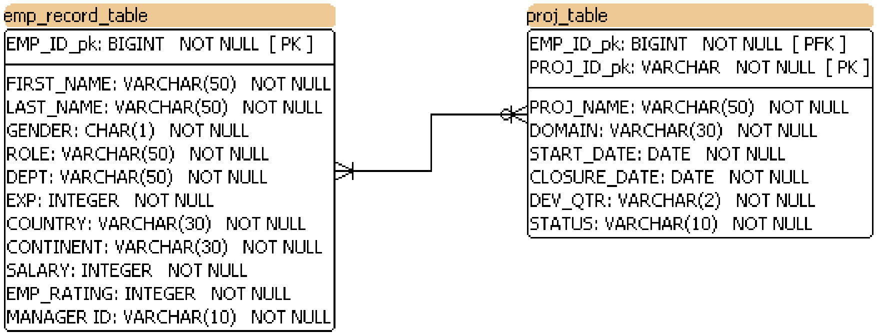
- EMP_ID – ID of the employee
- FIRST_NAME – First name of the employee
- LAST_NAME – Last name of the employee
- GENDER – Gender of the employee
- ROLE – Post of the employee
- DEPT – Field of the employee
- EXP – Years of experience the employee has
- COUNTRY – Country in which the employee is presently living
- CONTINENT – Continent in which the country is

The task to be performed:

☑ Create a database named *project* and *employee*, then import **data_science_team.csv** and **proj_table.csv** into the **project** database and **emp_record_table.csv** into the **employee** database from the given resources.

```
create database project_and_employee;
use project_and_employee;
set default_storage_engine = INNODB;
```

☑ Create an ER diagram for the given **project** and the **employee** databases.

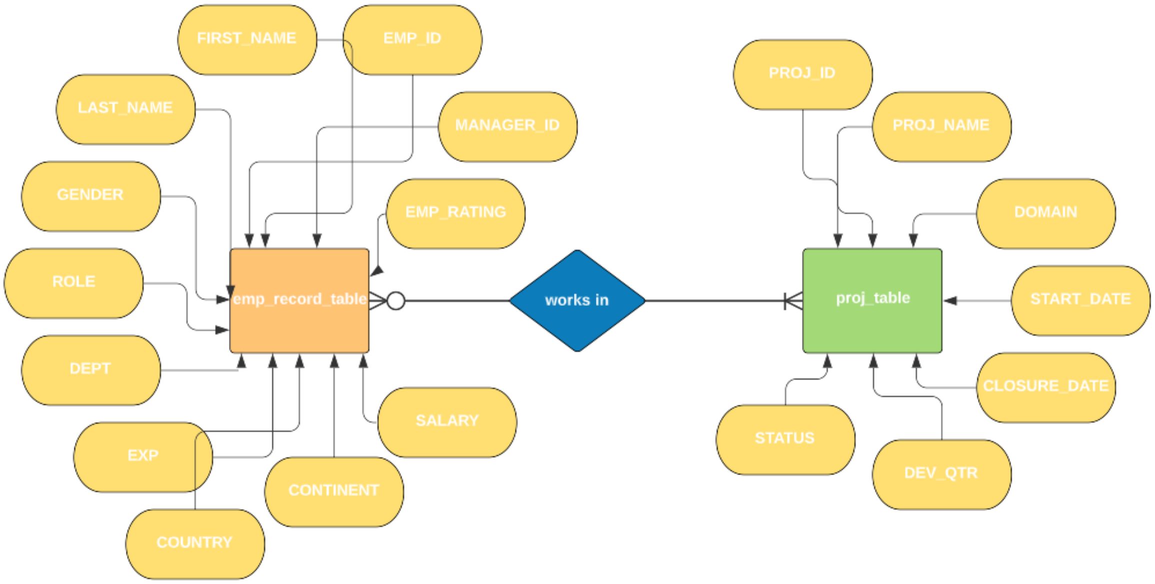


emp_record_table
EMP_ID_pk: BIGINT NOT NULL [PK]
FIRST_NAME: VARCHAR(50) NOT NULL
LAST_NAME: VARCHAR(50) NOT NULL
GENDER: CHAR(1) NOT NULL
ROLE: VARCHAR(50) NOT NULL
DEPT: VARCHAR(50) NOT NULL
EXP: INTEGER NOT NULL
COUNTRY: VARCHAR(30) NOT NULL
CONTINENT: VARCHAR(30) NOT NULL
SALARY: INTEGER NOT NULL
EMP_RATING: INTEGER NOT NULL
MANAGER ID: VARCHAR(10) NOT NULL

proj_table
EMP_ID_pk: BIGINT NOT NULL [PFK]
PROJ_ID_pk: VARCHAR NOT NULL [PK]
PROJ_NAME: VARCHAR(50) NOT NULL
DOMAIN: VARCHAR(30) NOT NULL
START_DATE: DATE NOT NULL
CLOSURE_DATE: DATE NOT NULL
DEV_QTR: VARCHAR(2) NOT NULL
STATUS: VARCHAR(10) NOT NULL

ERD for Project and Employee DBs

Ivan Oliveira | January 9, 2022



☑ 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 project_and_employee.emp_record_table;
```

☑ 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, FIRST_NAME, LAST_NAME, GENDER, DEPT,
EMP_RATING
from  project_and_employee.emp_record_table
where EMP_RATING < 2;
```

- greater than four

```
select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT,
EMP_RATING
from  project_and_employee.emp_record_table

where EMP_RATING > 4;
```

- between two and four

```
select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT,
EMP_RATING
from project_and_employee.emp_record_table
where EMP_RATING between 2 and 4;
```

☑ 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
project_and_employee.emp_record_table

where DEPT = 'FINANCE';
```

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

```
select * from project_and_employee.emp_record_table

where EMP_ID in (

        select MANAGER_ID from
project_and_employee.emp_record_table

);


select count(*) as Reporters from
project_and_employee.emp_record_table

where EMP_ID <> MANAGER_ID in(

        select MANAGER_ID from
project_and_employee.emp_record_table

);
```

☑ Write a query to list down all the employees from the healthcare and finance departments domain using union. Take data from the employee record table.

```
select * from project_and_employee.emp_record_table

where DEPT = 'HEALTHCARE'

union

select * from project_and_employee.emp_record_table

where DEPT = 'FINANCE';
```

☑ 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, DEPT,
EMP_RATING, (select MAX(EMP_RATING)

from project_and_employee.emp_record_table)

as max_ratings

from project_and_employee.emp_record_table

order by DEPT;
```

☑ 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 FIRST_NAME, LAST_NAME, `ROLE`, SALARY AS
MAX_Salary

from project_and_employee.emp_record_table

where (ROLE, SALARY) in (

        SELECT `ROLE`, MAX(Salary) FROM
project_and_employee.emp_record_table GROUP BY ROLE

   );


Select FIRST_NAME, LAST_NAME, `ROLE`, SALARY AS
MIN_Salary

from project_and_employee.emp_record_table

where (ROLE, SALARY) in (

        SELECT `ROLE`, MIN(Salary) FROM
project_and_employee.emp_record_table GROUP BY ROLE

   );
```

☑ 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, ROLE, EXP,
ROW_NUMBER() over (ORDER BY EXP desc) EMP_RANK

from project_and_employee.emp_record_table;
```

☑ 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.

```
use project_and_employee;

create View VW_above_6000_salary as

select * from project_and_employee.emp_record_table

where SALARY >= '6000';
```

☑ 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, DEPT

from project_and_employee.emp_record_table

where EXP >= 10;
```

☑ 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.

```
create procedure EXP_ABOVE_3_YRS()

select FIRST_NAME, LAST_NAME, EXP, ROLE, SALARY

from project_and_employee.emp_record_table

where EXP>3;

call EXP_ABOVE_3_YRS;
```

☑ 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:

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'.

```
use project_and_employee;

delimiter $$

create procedure CHECK_STANDARD(in EMP_ID VARCHAR(4), OUT STANDARD VARCHAR(10))

BEGIN

    IF EXP<=2 and ROLE='JUNIOR DATA SCIENTIST'

        SET STANDARD='MATCHED'

    ELSE IF EXP>2 AND EXP<5 AND ROL='ASSOCIATE DATA SCIENTIST'

        SET STANDARD='MATCHED'

 ELSE IF EXP >=5 AND EXP <10 AND ROLE='SENIOR DATA SCIENTIST'

        SET STANDARD='MATCHED'

    ELSE IF EXP>=10 AND EXP<12 AND ROLE='LEAD DATA SCIENTIST'

        SET STANDARD='MATCHED'
        ELSE IF EXP>=12 AND EXP <=16 AND ROLE='MANAGER'

                SET STANDARD='MATCHED'

        ELSE

                SET STANDARD='MISMATCHED'

        END IF;

END $$
```

☑ 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.

```
create index EMP_ID on project_and_employee.emp_record_table(FIRST_NAME);

explain select EMP_ID, FIRST_NAME, LAST_NAME from project_and_employee.emp_record_table

where first_name = 'Eric';
```

☑ 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 *, (SALARY*0.05*EMP_RATING) as EMP_BONUS

from project_and_employee.emp_record_table;
```

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

```
select COUNTRY, avg(SALARY) AVG_SALARY_BY_COUNTRY
from project_and_employee.emp_record_table

group by COUNTRY

limit 100;


select CONTINENT, avg(SALARY)
AVG_SALARY_BY_CONTINENT from
project_and_employee.emp_record_table

group by CONTINENT

limit 100;
```