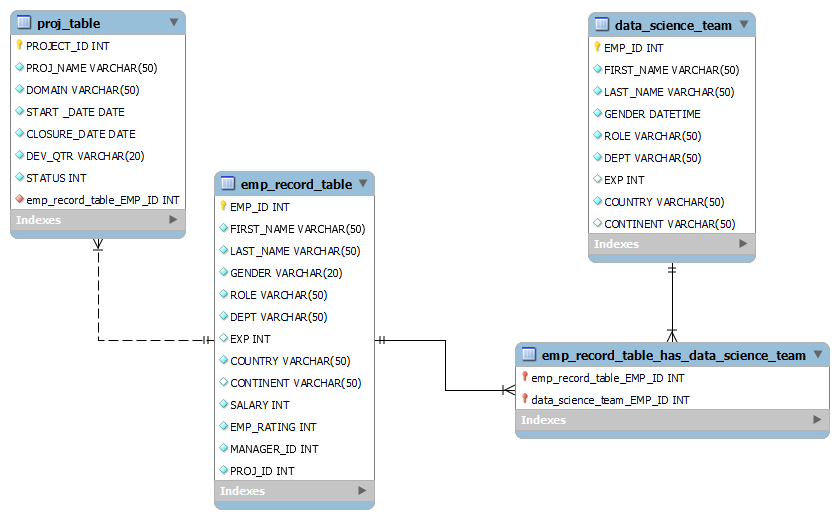
**ScienceQtech Employee Performance Mapping.**

**Task 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.

Step 1: **CREATE DATABASE Employee;**

Step 2: Add tables through ‘Table Data Import Wizard’’

**Task 2:** Create an ER diagram for the given employee database.



**Task 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.

**Select Emp\_ID, First\_name, last\_name, Gender, Dept**

**From emp\_record\_table;**



**Task 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

**Select** Emp\_ID, First\_name, last\_name, Gender, Dept, emp\_rating

**From** emp\_record\_table

**Where** emp\_rating<2;



* greater than four

**Select** Emp\_ID, First\_name, last\_name, Gender, Dept, emp\_rating

**From** emp\_record\_table

**Where** emp\_rating>4;



* between two and four

**Select** Emp\_ID, First\_name, last\_name, Gender, Dept, emp\_rating

**From** emp\_record\_table

**Where** emp\_rating between 2 and 4;



**Task 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.

**Select concat** (First\_name, ' ', last\_name) **as** Name

**From** emp\_record\_table

**Where** Dept='Finance'**;**



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

**Select** Emp\_ID, First\_name, last\_name, ROLE, Dept, Manager\_id

**From** emp\_record\_table

**Where** manager\_id **IN** ('E001','E083','E103','E428','E583','E612');



**Number of Reporters:**

**Select** Emp\_ID, First\_name, last\_name, ROLE, Dept

**From** emp\_record\_table

**Where** Emp\_ID **IN** ('E001','E083','E103','E428','E583','E612');



**Task 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.

**Select** Emp\_ID, First\_name, last\_name, ROLE, Dept

**From** emp\_record\_table

**Where** dept=**'Healthcare'**

**Union**

**Select** Emp\_ID, First\_name, last\_name, ROLE, Dept

**From** emp\_record\_table

**Where** dept=**'Finance'**;



**Task 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.

**Select** Emp\_ID, First\_name, last\_name, ROLE, Dept, emp\_rating

**From** emp\_record\_table

**Where** emp\_rating >3

**Group** **by** Dept;



**Task 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.

**Select** ROLE, Dept, **min(Salary)** **as** Min\_Salary, **max(salary)** **as** Max\_Salary

**From** emp\_record\_table

**Group** **by** ROLE;



**Task 10:** Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.

**Select** emp\_ID, **concat(first\_name, ' ', last\_name) as Name**, ROLE, Dept, Exp, **RANK()** **OVER** **(ORDER BY exp DESC)**

**From** emp\_record\_table;



**Task 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.

**Create View** high\_salary **As**

**Select** emp\_ID, concat(first\_name, ' ', last\_name) as Name, ROLE, Dept, Country, Salary

**From** emp\_record\_table

**Where** Salary > 6000;



**Task 12:** Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

Select Emp\_ID, concat(First\_name,' ',last\_name) as Name, ROLE, Dept, Exp

From emp\_record\_table

Where exp IN (Select exp From emp\_record\_table where exp > 10);



**Task 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.

**SQL Query:**

**DELIMITER &&**

**Create procedure** exp\_greater\_than\_3yrs()

**Begin**

**Select**\* **From** emp\_record\_table **Where** exp>3;

**End &&**

**For calling the result:**

**Call** exp\_greater\_than\_3yrs();



**Task 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 is:

* 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',

SQL Query:

**DELIMITER &&**

**Create** **procedure** check\_standard(

**emp\_exp** int,

**emp\_Role** varchar(20))

**Begin**

**Select** exp **INTO** emp\_exp;

**Select** Role **INTO** emp\_Role

**From** data\_science\_team;

**IF** emp\_exp<=2 **Then**

**Set** emp\_Role= 'JUNIOR DATA SCIENTIST';

**Elseif** emp\_exp Between 2 and 5 **Then**

**Set** emp\_Role= 'ASSOCIATE DATA SCIENTIST';

**Elseif** emp\_exp Between 5 And 10 **Then**

**Set** emp\_Role= 'SENIOR DATA SCIENTIST';

**Else**

**Set** emp\_Role= 'LEAD DATA SCIENTIST';

**End if;**

**End &&**

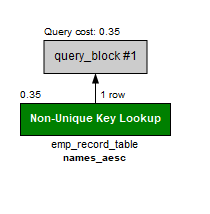
**Task 15:** Create an index to improve the cost and performance of the query to find the employee whose FIRST\_NAME is ‘Eric’ in the employee table after checking the execution plan.

**Create Index** names\_aesc **ON** emp\_record\_table(FIRST\_NAME(200));

**Select** \*

**From** emp\_record\_table

**where** first\_name = 'eric';

****

**Task 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).

**Select** Emp\_ID, concat(First\_name,' ',last\_name) as Name, ROLE, Salary, emp\_rating, **salary\* .05 \* emp\_rating as Bonus**

**From** emp\_record\_table;



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

**Select** Country, Continent, avg(salary)

**From** emp\_record\_table

**Group by** Country;

