# SQL PROJECT :-

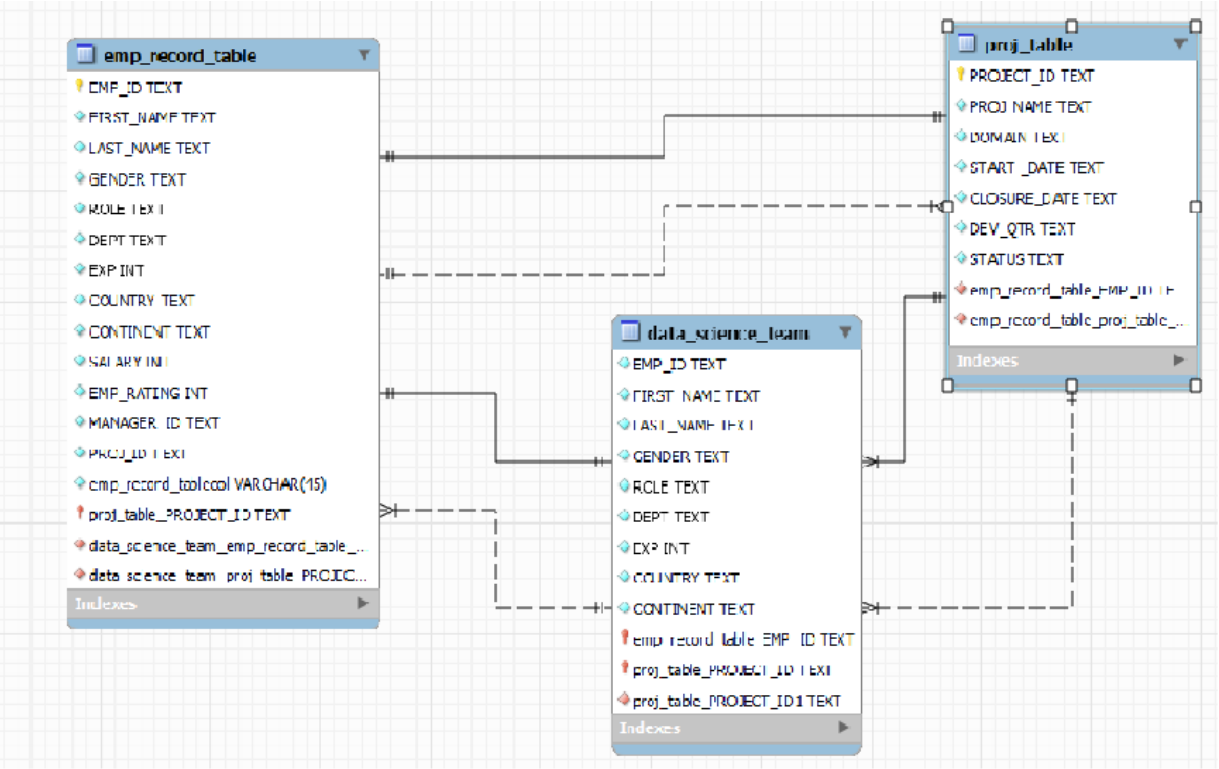
# Science Qtech Employee Performance Mapping.

- 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.

**Query :-** **create database employee;**

**use employee;**

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



- 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.

**Query :-  select emp_id, first_name, last_name, gender, dept from employee.emp_record_table;**

**select first_name, last_name, dept from 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,greater than four ,between two and four.

**Query :-  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 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.

**Query :-  select first_name, last_name, dept, concat(first_name,' ',Last_name) as Name from 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).

#### Method 1

**Query :-  SELECT emp_id, first_name, last_name, manager_id as reporters, role**

```
                    FROM emp_record_table  group by manager_id;
```

####Method 2

**Query :-**

```
SELECT DISTINCT emp_id, role FROM emp_record_table
  WHERE emp_id IN (SELECT manager_id FROM emp_record_table);
```

- Write a query to list down all the employees from the healthcare

and finance departments using union. Take data from the employee record table.

**Query :-**

```
select first_name, last_name, dept, emp_id from employee.emp_record_table where DEPT = "Healthcare"
union
select first_name, LAST_NAME, DEPT,EMP_ID from 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.

**Query :-**

```
select emp_id, first_name, Last_name, role, dept, emp_rating, max(emp_rating) from employee.emp_record_table group 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.

**Query :-**

```
select emp_id, first_name, last_name, role, min(salary) from employee.emp_record_table group by role;

select emp_id, first_name, last_name, role, max(salary) from employee.emp_record_table group by role;

select min(salary), max(salary), role from 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.

**Query :- select first_name, Last_name, emp_id, exp, rank() over (order by exp desc) as emp_rank from 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.

**Query :- create view emp_country as select emp_id, concat(first_name,' ', last_name), salary, country from employee.emp_record_table where salary > 6000;**

**select * from emp_country;**

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

**Query :- select exp, concat (first_name,' ', last_name) as emp_name from employee.emp_record_table where exp in (select exp from 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.

**Query :-**

```
delimiter //
create procedure exp_more_than_10 ()
 begin
select * from employee.emp_record_table where exp > 3;
 end //
  call exp_more_than_10 ();
```

- 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'.

**Query :-**

```
 delimiter //
 create procedure get_emp ( in eid varchar(10), out role varchar (50))
 begin
 declare gain int default 1;
 select exp into gain from employee.emp_record_table where emp_id = eid;
if gain <= 2 then set role = 'Junior Data Scientist';
elseif gain >2 and gain <= 5 then set role = 'Associate Data Scientist';
elseif gain >5 and gain <= 10 then set role = 'Senior Data Scientist';
elseif gain >10 and gain <=12 then set role = 'Lead Data Scientist';
elseif gain >12 and gain < 16 then set role = 'Manager';
else set role = 'President';
end if;
end //
 call get_emp ('E001', @role);
 select @role;
```

- 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.

**Query :-**

```
create index idx on emp_record_table(exp);

select * from 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).

**Query :-**

```
select emp_id, first_name, last_name, role, exp, salary, emp_rating, 0.05* salary* emp_rating

as bonus from emp_record_table order by bonus;
```

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

**Query :-**

```
select round(avg(salary)) as avg_sal, country, continent from emp_record_table group by country, CONTINENT;
```