SQL Project- 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.

  create database employee;

  use employee;

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

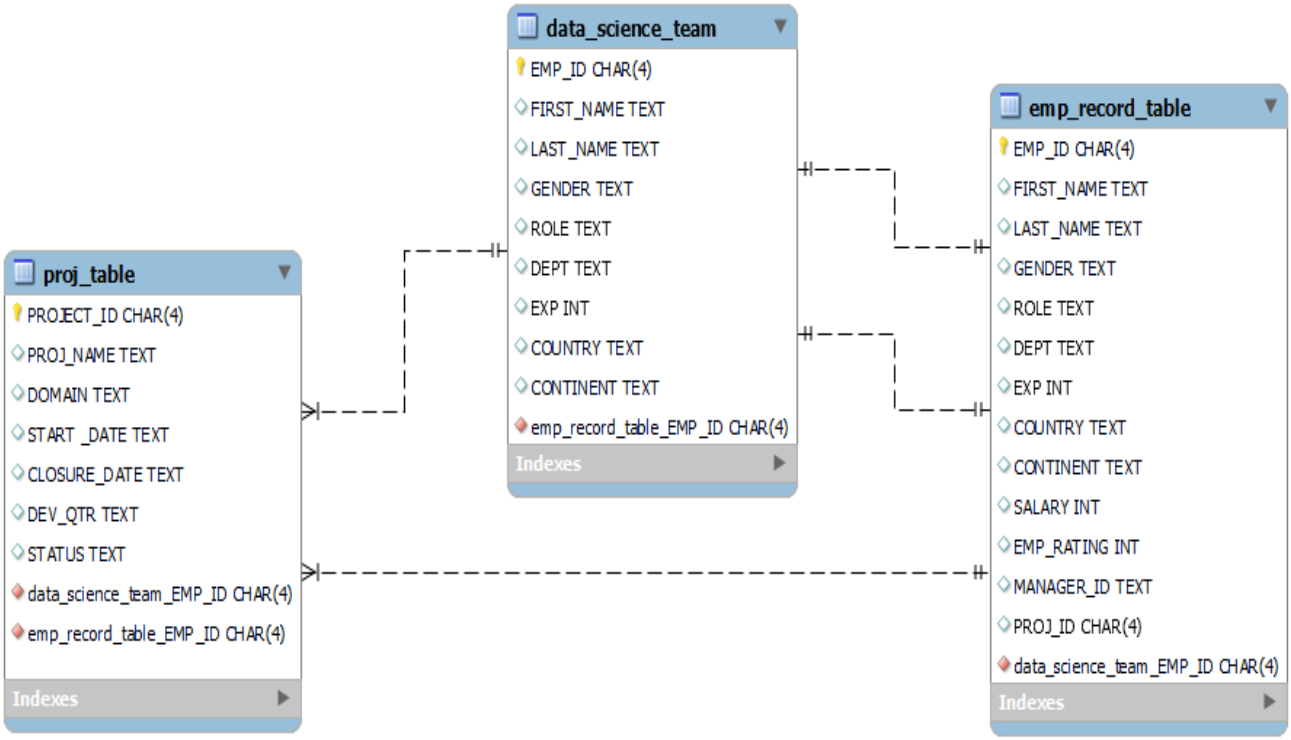
  alter table proj_table modify project_id char(4) primary key;

  alter table emp_record_table modify emp_id char(4) primary key;

  alter table emp_record_table modify proj_id char(4);

  alter table data_science_team modify emp_id char(4);

  describe data_science_team;



- 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;

- 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*/

```
delimiter $$
create procedure rating_case(in rating int)
begin
declare msg varchar(15);
set msg=' ';
if rating<2 then
select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
from emp_record_table where emp_rating <2;
   elseif rating between 2 and 4 then
select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
from emp_record_table where emp_rating between 2 and 4;
   elseif rating>4 then
select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
from emp_record_table where emp_rating>4;
   else
set msg='Invalid Rating';
```

```
select msg;
   end if;
end $$
call rating_case(1);
delimiter ;
```

- 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,DEPT from emp_record_table
where DEPT='FINANCE';
```

- Query to list only those employees who have someone reporting to

them. Also show the number of reporters (including the President).

```
select    t.first_name,t.last_name,e.manager_id,count(e.manager_id) as reporters
from emp_record_table e   join    emp_record_table t
on t.emp_id=e.manager_id group by e.manager_id;
```

- Query to list down all the employees from the healthcare & finance

departments using union.

Take data from the employee record table.

```
select * from emp_record_table where dept='HEALTHCARE'
union select * from emp_record_table where dept='FINANCE';
```

- 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

from emp_record_table;

- Query to calculate the minimum and the maximum salary of the employees

in each role. Take data from the employee record table.

select max(SALARY) as Max_Salary,min(SALARY) as Min_Salary,ROLE

from emp_record_table group by role;

- Query to assign ranks to each employee based on their experience.

Take data from the employee record table.

select *,rank() over(order by EXP desc) as RANKS from emp_record_table;

- Query to create a view that displays employees in various countries

whose salary is more than six thousand.

create view salary_distr as

select EMP_ID, FIRST_NAME, LAST_NAME,SALARY,COUNTRY

from emp_record_table where salary>6000 order by COUNTRY and SALARY desc;

select * from salary_distr;

- Nested query to find employees with experience of more than

ten years.

select * from emp_record_table where EXP IN (select EXP from emp_record_table where EXP>10)

order by EXP desc;

- query to create a stored procedure to retrievethe details

of the employees whose experience is more than three years.

DELIMITER $$

create procedure emp_exp()

begin

select * from emp_record_table where EXP>3

order by EXP desc;

end $$

DELIMITER ;

call emp_exp();

- 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 $$
create function employee.designation(experience int)
returns varchar(30)
deterministic
begin
declare assigned_profile varchar (30);
   set assigned_profile=' ';
   if experience<=2 then
set       assigned_profile='JUNIO          DATA SCIENTIST';
elseif(experience          >=2 and experience<=5)then
set       assigned_profile='ASSOCIATE    DATA SCIENTIST';
   elseif (experience>=5 and experience<=10)    then
set       assigned_profile='SENIOR DATA SCIENTIST';
   elseif           (experience>=10          and experience<=12)    then
set       assigned_profile='LEAD DATA SCIENTIST';
   elseif (experience>=12          and experience<=16)    then
set       assigned_profile='MANAGER';
   else
   set    assigned_profile='PRESIDENT';
   end if;
   return(assigned_profile);
 end$$
delimiter ;
```

select emp_id,first_name, employee.designation(exp) as assigned_profile,

role from employee.emp_record_table;

- 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 idxs on employee.emp_record_table(FIRST_NAME(20));

select * from employee.emp_record_table where FIRST_NAME='ERIC';

EXPLAIN select * from 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 emp_id,first_name,last_name,salary,emp_rating,round(0.05*salary*emp_rating) as BONUS

from employee.emp_record_table;

- Query to calculate the average salary distribution based on the

continent and country.

select continent,country,avg(salary) over(partition by country)

as avg_salary from emp_record_table group by country order by continent;