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.

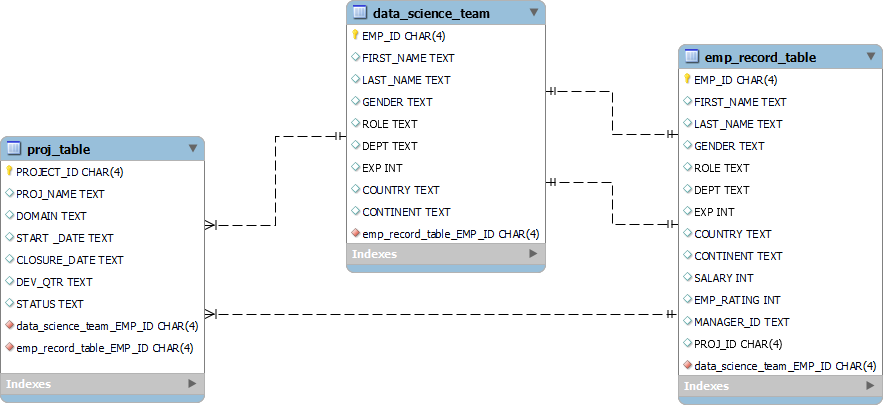
# create database employee; use employee;

* 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;