**ScienceQtech Employee Performance Mapping.**

Q1) 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.

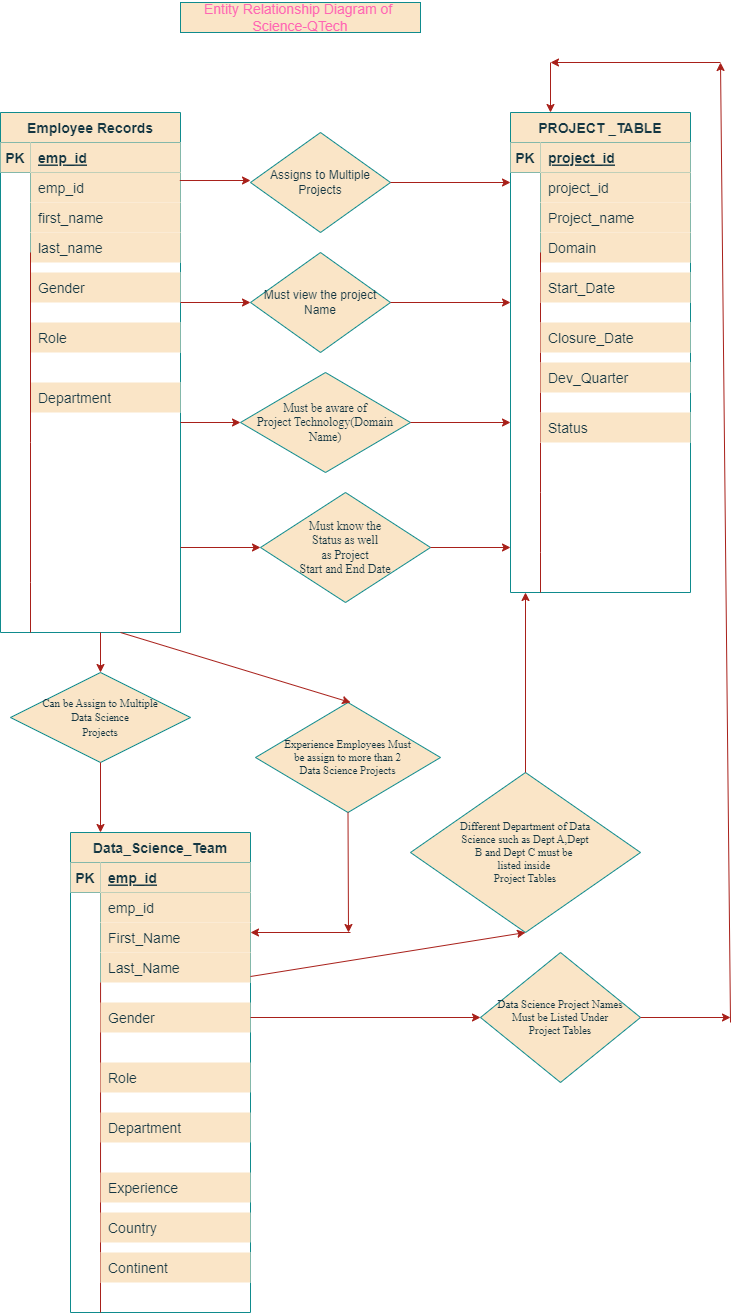
Ans) create database employee;

use employee;

show tables;

Q2) Create an ER diagram for the given **employee**database.

Ans)



Q3)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.

Ans) use employee;

show tables;

select \* from emp\_record\_table;

select emp\_id,first\_name,last\_name,gender,dept from emp\_record\_table order by dept;

Q4) 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

Ans) use employee;

select \* from emp\_record\_table;

select emp\_id,first\_name,last\_name,gender,dept,emp\_rating from emp\_record\_table where emp\_rating<2;

select emp\_id,first\_name,last\_name,gender,dept,emp\_rating from emp\_record\_table where emp\_rating >4;

select emp\_id,first\_name,last\_name,gender,dept,emp\_rating from emp\_record\_table where emp\_rating between 2 and 4;

Q5) 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.

Ans) select concat(first\_name,last\_name) as Name from emp\_record\_table

where DEPT="Finance";

Q6)Write a query to list only those employees who have someone reporting to them.

Also, show the number of reporters (including the President)

Ans) select \* from emp\_record\_table;

select emp\_id,first\_name,last\_name,PROJ\_ID,MANAGER\_ID from emp\_record\_table

where PROJ\_ID is not Null and MANAGER\_ID is not Null;

select role,count(role) as Number\_of\_reporters from

emp\_record\_table

group by role;

Q7) Write a query to list down all the employees from the healthcare and finance departments using union.

Take data from the employee record table.

Ans) use employee;

select \* from emp\_record\_table;

select first\_name,last\_name from emp\_record\_table

where dept="HealthCare"

union

select first\_name,last\_name from emp\_record\_table

where dept="Finance";

Q8) 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.

Ans) use employee;

select emp\_id,first\_name,last\_name,

role,dept,emp\_rating from emp\_record\_table

order by dept;

select max(emp\_rating),dept

from emp\_record\_table

group by dept;

Q9) Write a query to calculate the minimum and the maximum salary of the employees in each role.

Take data from the employee record table.

Ans) use employee;

select min(salary) as Minimum\_Salary,max(salary) as Maximum\_Salary,role

from emp\_record\_table

group by role;

Q10) Write a query to assign ranks to each employee based on their experience.

Take data from the employee record table.

Ans) use employee;

select \* from emp\_record\_table;

drop procedure if exists empexp;

Delimiter $$

create procedure empexp(in empx bigint, out eranks varchar(100))

begin

declare myexp bigint default 1;

select exp into myexp from emp\_record\_table where exp=empx;

if myexp between 1 and 2 then

set eranks="Junior Data Scientist";

elseif myexp between 3 and 5 then

set eranks="Associate Data Scientist";

elseif myexp between 6 and 9 then

set eranks="Senior Data Scientist";

elseif myexp between 11 and 12 then

set eranks="Lead Data Scientist";

elseif myexp between 13 and 15 then

set eranks="Manager";

elseif myexp=20 then

set eranks="President";

else

set eranks="Not Applicable";

end if;

end $$

call empexp(13,@eranks);

select @eranks;

Q11) 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.

Ans) use employee;

select \* from emp\_record\_table;

create view emprecordsalary

as select first\_name,last\_name,salary from emp\_record\_table

where salary>6000;

select \* from emprecordsalary;

Q12) Write a nested query to find employees with experience of more than ten years.

Take data from the employee record table.

Ans) use employee;

select \* from emp\_record\_table;

select first\_name,last\_name,exp from emp\_record\_table

where exp=(select exp from emp\_record\_table where exp>10);

select first\_name,last\_name,exp from

emp\_record\_table where exp>10;

create view games

as select

first\_name,last\_name,exp from

emp\_record\_table where exp>10;

select \* from games;

Q13) 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.

Ans) use employee;

select \* from emp\_record\_table;

Delimiter $$

create procedure getemployeedetails()

begin

select \* from emp\_record\_table where exp >3;

end $$

call getemployeedetails();

Q14) 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'.

Ans) use employee;

select \* from emp\_record\_table;

Drop Function if exists Customer\_occupation;

DELIMITER $$

CREATE FUNCTION Customer\_Occupation(

exp int

)

RETURNS VARCHAR(100)

DETERMINISTIC

BEGIN

DECLARE customer\_occupation VARCHAR(100);

IF exp <= 2 THEN

SET customer\_occupation = 'Junior Data Scientist';

ELSEIF (exp <= 2 AND

exp >= 5) THEN

SET customer\_occupation = 'Associate Data Scientist';

ELSEIF exp >= 5 and exp<=10 THEN

SET customer\_occupation = 'Senior Data Scientist';

ELSEIF exp >= 10 and exp<=12 THEN

SET customer\_occupation = 'Lead Data Scientist';

ELSEIF exp >= 12 and exp<=16 THEN

SET customer\_occupation = 'Lead Data Manager';

END IF;

-- return the customer occupation

RETURN (customer\_occupation);

END$$

SHOW FUNCTION STATUS WHERE db = 'employee';

SELECT first\_name, last\_name, Customer\_Occupation(exp)

FROM emp\_record\_table

order by exp;

Q15-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.

Ans) use employee;

desc emp\_record\_table;

alter table emp\_record\_table modify column first\_name varchar(100);

create index ixs on employee.emp\_record\_table(FIRST\_NAME);

show index from emp\_record\_table;

Q16)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).

Ans) use employee;

alter table emp\_record\_table add column bonus\_for\_employee bigint ;

alter table emp\_record\_table drop column bonus\_for\_employee;

/\* Syntax of Update While calculating bonus \*/

update emp\_record\_table set bonus\_for\_employee=0.05\*16500\*5

where emp\_id="E001";

update emp\_record\_table set bonus\_for\_employee=0.05\*8500\*3

where emp\_id="E005";

update emp\_record\_table set bonus\_for\_employee=0.05\*9000\*2

where emp\_id="E010";

update emp\_record\_table set bonus\_for\_employee=0.05\*5500\*5

where emp\_id="E052";

update emp\_record\_table set bonus\_for\_employee=0.05\*7700\*1

where emp\_id="E057";

update emp\_record\_table set bonus\_for\_employee=0.05\*9500\*5

where emp\_id="E083";

update emp\_record\_table set bonus\_for\_employee=0.05\*10500\*4

where emp\_id="E103";

update emp\_record\_table set bonus\_for\_employee=0.05\*7500\*5

where emp\_id="E204";

update emp\_record\_table set bonus\_for\_employee=0.05\*6500\*2

where emp\_id="E245";

update emp\_record\_table set bonus\_for\_employee=0.05\*7000\*3

where emp\_id="E260";

update emp\_record\_table set bonus\_for\_employee=0.05\*5000\*3

where emp\_id="E403";

update emp\_record\_table set bonus\_for\_employee=0.05\*11000\*4

where emp\_id="E428";

update emp\_record\_table set bonus\_for\_employee=0.05\*5000\*2

where emp\_id="E505";

update emp\_record\_table set bonus\_for\_employee=0.05\*4300\*1

where emp\_id="E532";

update emp\_record\_table set bonus\_for\_employee=0.05\*1000\*2

where emp\_id="E583";

update emp\_record\_table set bonus\_for\_employee=0.05\*8500\*4

where emp\_id="E612";

update emp\_record\_table set bonus\_for\_employee=0.05\*3000\*1

where emp\_id="E620";

update emp\_record\_table set bonus\_for\_employee=0.05\*2800\*4

where emp\_id="E640";

select \* from emp\_record\_table;

Q17) Write a query to calculate the average salary distribution

based on the continent and country. Take data from the employee record table.

Ans) use employee;

select \* from emp\_record\_table;

/\* Average Salary Distribution Query \*/

select avg(salary) from emp\_record\_table as AverageSalaryDistribution1

where continent="North America" and Country="USA";

select avg(salary) from emp\_record\_table as AverageSalaryDistribution2

where continent="North America" and Country="Canada";

select avg(salary) from emp\_record\_table as AverageSalaryDistribution3

where continent="Europe" and Country="Germany";

select avg(salary) from emp\_record\_table as AverageSalaryDistribution4

where continent="South America" and Country="Colombia";

select avg(salary) from emp\_record\_table as AverageSalaryDistribution5

where continent="Europe" and Country="France";

select avg(salary) from emp\_record\_table as AverageSalaryDistribution6

where continent="Asia" and Country="India";