

ScienceQtech Employee Performance Mapping.

Course-end Project 1 By PAVAN LANDE

1. • Create a database named project and employee, then import data_science_team.csv and proj_table.csv into the project database and emp_record_table.csv into the employee database from the given resources.

```
create database if not exists employee;
```

```
Use employee;
```

```
create table if not exists emp_table (
```

```
Emp_id varchar(4) not null primary key,
```

```
First_name varchar(100) not null,
```

```
Last_name varchar(100) not null,
```

```
Gender varchar(1) not null,
```

```
role_emp varchar(100) not null,
```

```
Dept varchar (100) not null,
```

```
Exp_emp int not null check (Exp_emp>=0),
```

```
Country varchar(80) not null,
```

```
Continent varchar (50)not null,
```

```
salary int not null check (Salary>=0),
```

```
Emp_rating int not null check (Emp_rating>=0),
```

```
Manager_id varchar(100) not null,
```

```
constraint empid_check check( substr(Emp_id,1,1)='E'),
```

```
constraint manid_check check( substr(Manager_id,1,1)='E'),
```

```
constraint gender_check check(Gender in ('M','F','O'))
```

```
) engine = InnoDB;
```

```
select * from emp_table;
```

```
create database if not exists project;
```

```
use project;
```

```
create table if not exists DS_team (
```

```
Emp_id varchar(4) not null primary key,
```

```
First_name varchar(100) not null,
```

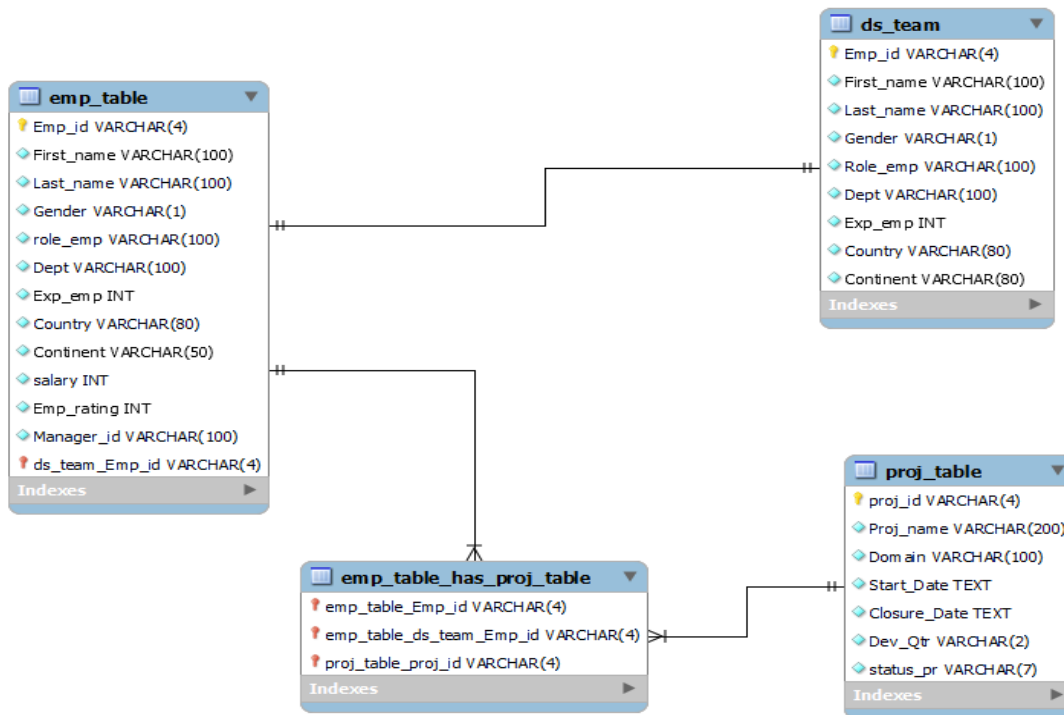
```
Last_name varchar(100) not null,  
Gender varchar (1) not null,  
Role_emp varchar(100) not null,  
Dept varchar (100) not null,  
Exp_emp int not null check(Exp_emp>=0),  
Country varchar(80) not null,  
Continent varchar(80) not null,  
constraint dsid_check check( substr(Emp_id,1,1)='E'),  
constraint dsgender_check check(Gender in ('M','F','O'))  
)engine= InnoDB;
```

```
Select * from DS_team;
```

```
create table if not exists proj_table (  
proj_id varchar(4) not null primary key,  
Proj_name varchar(200) not null,  
Domain varchar(100) not null,  
Start_Date text not null,  
Closure_Date text not null,  
Dev_Qtr varchar(2) not null,  
status_pr varchar(7) not null,  
constraint projid_check check( substr(Proj_id,1,1)='P'),  
constraint check_status check( Status_pr in ('YTS','WIP','DONE','DELAYED'))  
) engine=InnoDB;
```

```
SELECT * from proj_table;
```

2. Create an ER diagram for the given project and the employee databases.



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_table
order by Dept;
  
```

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
- greater than four
- between two and four

```

select emp_id, first_name, last_name, gender, Dept, Emp_rating
from emp_table
where Emp_rating < 2;
  
```

```

select emp_id, first_name, last_name, gender, Dept, Emp_rating
from emp_table
where Emp_rating > 4;
  
```

```
select emp_id, first_name, last_name, gender, Dept, Emp_rating
from emp_table
where Emp_rating between 2 and 4;
```

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_ws(' ', First_name, Last_name) as Name from employee.emp_table where Dept="FINANCE";
```

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 m.Emp_id, m.First_name, m.Last_name, m.Role_emp, m.Dept, count(e.Emp_id) as Emp_recording from
emp_table m
inner join emp_table e
on m.emp_id = e.Manager_id
and e.emp_id != e.Manager_id
where m.Role_emp in("MANAGER", "PRESIDENT", "CEO")
group by m.Emp_id
order by m.Emp_id;
```

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 m.Emp_id, m.First_name, m.Last_name, m.Dept from emp_table m
where m.Dept in ("HEALTHCARE", "FINANCE")
union
select m.Emp_id, m.First_name, m.Last_name, m.Dept from emp_table m
where m.Dept in ("HEALTHCARE", "FINANCE")
order by Dept;
```

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.

First variant:

```
SELECT Dept, emp_id, First_name, Last_name, Role_emp, emp_rating,  
       MAX(EMP_RATING) OVER (PARTITION BY DEPT) AS MAX_RATING  
FROM emp_table  
ORDER BY DEPT, EMP_RATING DESC;
```

```
SELECT Dept, emp_id, First_name, Last_name, Role_emp, emp_rating,  
MAX(EMP_RATING) OVER (PARTITION BY DEPT) AS MAX_RATING  
from emp_table;
```

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_emp, MIN(SALARY) AS MIN_SALARY, MAX(SALARY) AS MAX_SALARY  
FROM emp_table  
GROUP BY role_emp;
```

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

```
select emp_id, first_name, last_name, role_emp, Dept, Exp_emp, rank() over (order by Exp_emp)  
emp_experience_rank,  
dense_rank() over (order by Exp_emp) emp_experience_denserank from emp_table;
```

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 Employee_view as  
select Emp_id, Country, Salary from emp_table where salary > 6000;
```

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

Take data from the employee record table.

```
select m.Emp_id,m.First_name,m.Last_name,m.Dept,m.Exp_emp from emp_table m
where exists(Select 1 from emp_table where Exp_emp > 10) and m. Exp_emp > 10;
```

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.

Delimiter &&

```
create procedure get_experience()
begin
select * from emp_table where Exp_emp > 3;
end &&
call get_experience();
```

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 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 \$\$

```
DROP FUNCTION IF EXISTS employee.get_job_profile;
CREATE FUNCTION employee.get_job_profile(Exp_emp INT)
RETURNS VARCHAR(2000) DETERMINISTIC
BEGIN
    DECLARE job_profile VARCHAR(2000);
    IF Exp_emp <= 2 THEN
        SET job_profile = 'JUNIOR DATA SCIENTIST';
    ELSEIF Exp_emp <= 5 THEN
```

```

    SET job_profile = 'ASSOCIATE DATA SCIENTIST';
ELSEIF Exp_emp <= 10 THEN
    SET job_profile = 'SENIOR DATA SCIENTIST';
ELSEIF Exp_emp <= 12 THEN
    SET job_profile = 'LEAD DATA SCIENTIST';
ELSEIF Exp_emp <= 16 THEN
    SET job_profile = 'MANAGER';
END IF;
RETURN job_profile;
END $$
DELIMITER ;

```

```

SELECT First_name, Last_name, Exp_emp, Role_emp, job_profile(Exp_emp) AS Employee_profile
FROM employee.emp_table order by Exp_emp;

```

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 idx_word on emp_table(First_name);
select * from emp_table where First_name = 'Eric';

```

16. Write a query to calculate the bonus for all the employees, alter based on their ratings and salaries (Use the formula: 5% of salary * employee rating).

```

select Emp_id, First_name, Last_name, Role_emp, Salary, Emp_rating, 0.05*Salary*Emp_rating as BONUS
from emp_table order by BONUS;

```

```

SELECT Emp_id, First_name, Last_name, Role_emp, SUM(0.05*Salary*Emp_rating) AS BONUS_TOTAL
FROM emp_table
GROUP BY Emp_id, First_name, Last_name, Role_emp;

```

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) over ( partition by country) AVG_Salary_Distr_Country,  
avg(Salary) over ( partition by Continent) AVG_Salary_Distr_Continent from emp_table;
```

```
/*-----END OF PROJECT-----*/
```