/\* **ScienceQtech Employee Performance Mapping.\*/**

//To facilitate a better understanding, managers have provided ratings for each employee which will help the HR department to finalize the employee performance mapping. As a DBA, you should find the maximum salary of the employees and ensure that all jobs are meeting the organization's profile standard. You also need to calculate bonuses to find extra cost for expenses. This will raise the overall performance of the organization by ensuring that all required employees receive training.//

select \* from data\_science\_team;

select \* from emp\_record\_table;

select \* from proj\_table;

//\*\*\*\*1. Create a database named employee, then import data\_science\_team.csv proj\_table.csv

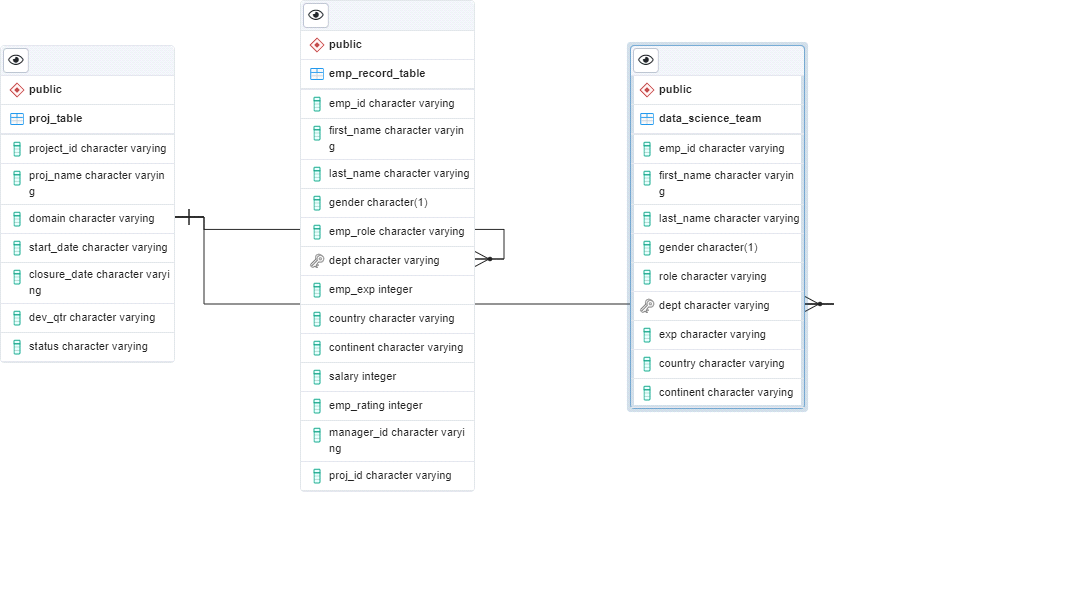
and emp\_record\_table.csv into the employee database from the given resources.\*\*\*\*//

copy emp\_record\_table from 'G:\SQL TRAINING\Project1\emp\_record\_table.csv' delimiter ',' csv header;

copy proj\_table from 'G:\SQL TRAINING\Project1\proj\_table.csv' delimiter ',' csv header;

copy data\_science\_team from 'G:\SQL TRAINING\Project1\data\_science\_team.csv' delimiter ',' csv header;

/\*2. Create an ER diagram for the given employee database.\*/



//\*\*\*\*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\_record\_table;

//\*\*\*\*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\_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;

//\*\*\*\*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(FIRST\_NAME,' ',LAST\_NAME)as Name from emp\_record\_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 \* from emp\_record\_table

//\*\*\*\*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 \* from emp\_record\_table where dept='HEALTHCARE'

UNION

select \* from emp\_record\_table where dept='FINANCE';

/\*\*\*\*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.\*\*\*\*//

select emp\_id, FIRST\_NAME, LAST\_NAME, emp\_role, dept,EMP\_RATING,

(select max(emp\_rating) from emp\_record\_table b where a.dept = b.dept )as maxRating

from emp\_record\_table a

order by a.dept

//\*\*\*\*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 emp\_role,min(salary) as minSalary, max(salary) as maxSalary

from emp\_record\_table

group by emp\_role

//\*\*\*\*10 Write a query to assign ranks to each employee based on their experience.

Take data from the employee record table.\*\*\*\*//

select emp\_id, emp\_exp, RANK () OVER ( ORDER BY emp\_exp desc) as rank\_number

from emp\_record\_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 view\_SalaryMoreThanSixThousand

as

select \* from emp\_record\_table where salary > 6000

select \* from view\_SalaryMoreThanSixThousand

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

Take data from the employee record table.\*\*\*\*//

select \* from emp\_record\_table where emp\_id in (select emp\_id from emp\_record\_table where emp\_exp >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.\*\*\*\*//

create or replace FUNCTION FN\_GetEmployeeByExp()

returns setof emp\_record\_table

AS

$func$

SELECT \*

FROM emp\_record\_table

where emp\_exp > 3;

$func$

LANGUAGE sql;

select \* from FN\_GetEmployeeByExp()

//\*\*\*\*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'.

\*\*\*\*//

create or replace function FN\_IsMatchesStandards()

returns table ( fn\_emp\_id varchar, fn\_role varchar,fn\_emp\_exp integer,fn\_Role\_As\_Per\_Standerds text, fn\_Is\_Role\_As\_Per\_Standerds boolean )

language plpgsql

as $$

begin

return query

select emp\_id, role, cast(exp as integer) as emp\_exp,

case

when(cast(exp as integer) <= 2) then('JUNIOR DATA SCIENTIST')

when(cast(exp as integer) between 3 and 5 ) then('ASSOCIATE DATA SCIENTIST')

when(cast(exp as integer) between 6 and 10 ) then('SENIOR DATA SCIENTIST')

when(cast(exp as integer) between 11 and 12 ) then('LEAD DATA SCIENTIST')

else('MANAGER')

end as Role\_As\_Per\_Standerds,

case when role = (case

when(cast(exp as integer) <= 2) then('JUNIOR DATA SCIENTIST')

when(cast(exp as integer) between 3 and 5 ) then('ASSOCIATE DATA SCIENTIST')

when(cast(exp as integer) between 6 and 10 ) then('SENIOR DATA SCIENTIST')

when(cast(exp as integer) between 11 and 12 ) then('LEAD DATA SCIENTIST')

else('MANAGER')

end)

then true

else false

end

as Is\_Role\_As\_Per\_Standerds

from Data\_science\_team

order by cast(exp as integer) desc;

end ;

$$

select \* from FN\_IsMatchesStandards();

/?\*\*\*\*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 IX\_FirstNAME1 ON emp\_record\_table USING btree (first\_name)

explain select \* from emp\_record\_table WHERE first\_name = 'Eric';

//\*\*\*\*16 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,salary,emp\_rating,((salary\*5/100)\*emp\_rating) as bonus FROM emp\_record\_table

//\*\*\*\*17 Write a query to calculate the average salary distribution based on the continent and country.

Take data from the employee record table.\*\*\*\*//

SELECT continent, avg(Salary) FROM emp\_record\_table group by continent;

SELECT country, avg(Salary) FROM emp\_record\_table group by country;

select continent,

(select avg(b.Salary) from emp\_record\_table b where a.continent = b.continent) as continentAvg,

country,

(select avg(c.Salary) from emp\_record\_table c where a.country = c.country) as countryAvg

FROM emp\_record\_table a

group by continent,country

Created by SATYAM BAVISKAR

Links: [www.linkedin.com/in/satyam-baviskar-b9891625a](http://www.linkedin.com/in/satyam-baviskar-b9891625a)

Links: <https://github.com/Satyam1391>