

## SQL project- Science Qtech employee performance mapping.

```
CREATE DATABASE employee;
```

```
USE employee;
```

```
-- 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;
```

```
/* 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  
      where EMP_RATING < 2;
```

```
select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record  
      where EMP_RATING > 4;
```

```
select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING from emp_record  
      where EMP_RATING BETWEEN 2 AND 4;
```

```
-- 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 FIRST_NAME, LAST_NAME, concat(FIRST_NAME, ' ', LAST_NAME) as NAME, DEPT from  
emp_record  
      where DEPT = "FINANCE";
```

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

```
select * from emp_record
      where EMP_ID in (select MANAGER_ID from emp_record)
      order by EMP_ID;
```

-- 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
      where DEPT = 'HEALTHCARE'
union
select * from emp_record
      where DEPT = 'FINANCE';
```

-- 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 distinct(ROLE), max(SALARY) as "MAX SAL", min(SALARY) as "MIN SAL" from emp_record
      group by ROLE;
```

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

```
select FIRST_NAME, LAST_NAME, EXP, rank() over (order by EXP desc) as 'RANK' from emp_record;
```

-- 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

```
create view view1 as select EMP_ID, FIRST_NAME, LAST_NAME, COUNTRY, SALARY from emp_record
      where salary > 6000;
```

-- Run view

```
select * from view1;
```

-- Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

```
select * from (select FIRST_NAME, LAST_NAME, EXP from emp_record
               where EXP >10) as tab;
```

-- 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 HIGHEXP
```

```
begin
```

```
    select FIRST_NAME, LAST_NAME, EXP from emp_record
           where EXP > 3;
```

```
end //
```

```
delimiter ;
```

```
call HIGHEXP;
```

-- 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 3 to 5 years assign 'ASSOCIATE DATA SCIENTIST',

For an employee with the experience of 6 to 9 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 13 to 16 years assign 'MANAGER'.\*/

-- Create query

```
delimiter $$
```

```
create function check_role(EXP integer)
```

```

returns VARCHAR(40)

deterministic

begin

declare chck VARCHAR(40);

if EXP <= 2 then set chck = "JUNIOR DATA SCIENTIST";

    elseif EXP between 3 and 5 then set chck = "ASSOCIATE DATA SCIENTIST";

        elseif EXP between 6 and 9 then set chck = "SENIOR DATA SCIENTIST";

            elseif EXP between 10 and 12 then set chck = "LEAD DATA SCIENTIST";

                elseif EXP >= 13 then set chck = "MANAGER";

                    end if; return (chck);

end $$

delimiter ;

-- Run query for all data science

select FIRST_NAME, LAST_NAME, ROLE, check_role(EXP) as EXPECTED_ROLE, EXP from
data_science_team;

-- Run query for outliers

select FIRST_NAME, LAST_NAME, ROLE, check_role(EXP) as EXPECTED_ROLE, EXP from
data_science_team

    where role != check_role(exp);

-- 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 FIRST_NAME, LAST_NAME, SALARY, EMP_RATING, (0.05 * SALARY * EMP_RATING) AS BONUS
from emp_record;

-- Write a query to calculate the average salary distribution based on the continent and country. Take
data from the employee record table.

-- Country

select COUNTRY, avg(SALARY) as AVG_SALARY from emp_record

    group by COUNTRY;

```

-- Continent

```
select CONTINENT, avg(SALARY) as AVG_SALARY from emp_record  
group by CONTINENT;
```