Description

**Problem scenario:**

ScienceQtech is a startup that works in the Data Science field. ScienceQtech has worked on fraud detection, market basket, self-driving cars, supply chain, algorithmic early detection of lung cancer, customer sentiment, and the drug discovery field. With the annual appraisal cycle around the corner, the HR department has asked you (Junior Database Administrator) to generate reports on employee details, their performance, and on the project that the employees have undertaken, to analyze the employee database and extract specific data based on different requirements.

**Objective:**

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.

**SQL Queries**

-- 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.

**create database employee;**

-- 2. Create an ER diagram for the given employee database.

-- created ER Diagram

**alter table data\_science\_team**

**modify emp\_id varchar(10) not null primary key,**

**modify first\_name varchar(100) not null,**

**modify last\_name varchar(100) not null,**

**modify gender char(1) not null,**

**modify role varchar(100) not null,**

**modify dept varchar(100) not null,**

**modify exp int not null,**

**modify country varchar(100) not null,**

**modify continent varchar(100) not null;**

**alter table emp\_record\_table**

**modify emp\_id varchar(10) not null primary key,**

**modify first\_name varchar(100) not null,**

**modify last\_name varchar(100) not null,**

**modify gender char(1) not null,**

**modify role varchar(100) not null,**

**modify dept varchar(100) not null,**

**modify exp int not null,**

**modify country varchar(100) not null,**

**modify continent varchar(100) not null,**

**modify salary int not null,**

**modify emp\_rating int not null,**

**modify manager\_id varchar(10) null,**

**modify proj\_id varchar(10) null;**

**alter table proj\_table**

**modify proj\_id varchar(10) not null primary key,**

**modify proj\_name varchar(100) not null,**

**modify domain varchar(50) not null,**

**modify start\_date date not null,**

**modify closure\_date date not null,**

**modify dev\_qtr varchar(10) not null,**

**modify status varchar(20) not null;**

-- 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**

**ORDER BY emp\_id;**

-- 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

**SELECT**

**emp\_id, first\_name, last\_name, gender, dept, emp\_rating**

**FROM**

**emp\_record\_table**

**WHERE**

**emp\_rating < 2;**

-- ● greater than four

**SELECT**

**emp\_id, first\_name, last\_name, gender, dept, emp\_rating**

**FROM**

**emp\_record\_table**

**WHERE**

**emp\_rating > 4;**

-- ● between two and four

**SELECT**

**emp\_id, first\_name, last\_name, gender, dept, emp\_rating**

**FROM**

**emp\_record\_table**

**WHERE**

**emp\_rating > 2 and emp\_rating < 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**

**m.emp\_id, m.first\_name, m.last\_name, m.gender, m.role, COUNT(e.emp\_id) AS num\_reporters**

**FROM**

**emp\_record\_table m**

**JOIN**

**emp\_record\_table e ON m.emp\_id = e.manager\_id**

**GROUP BY**

**m.emp\_id, m.first\_name, m.last\_name, m.gender, m.role**

**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 \* 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, role, dept, emp\_rating,**

**max(emp\_rating) over (partition by dept) as max\_emp\_rating**

**from emp\_record\_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**

**emp\_id, FIRST\_NAME, LAST\_NAME, role,salary, MIN(salary) AS min\_salary, MAX(salary) AS max\_salary**

**FROM**

**emp\_record\_table**

**GROUP BY**

**emp\_id, FIRST\_NAME, LAST\_NAME, role, salary**

**ORDER BY 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, FIRST\_NAME, LAST\_NAME, role, dept, exp, rank() over (order by exp desc) as exp\_rank 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 v\_emp as**

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

**select \* from v\_emp;**

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

**select emp\_id, first\_name, last\_name, role, dept, exp from emp\_record\_table where emp\_id in (select emp\_id from emp\_record\_table where 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.

**USE `employee`;**

**DROP procedure IF EXISTS `emp\_detail\_exp`;**

**DELIMITER $$**

**USE `employee`$$**

**CREATE PROCEDURE `emp\_detail\_exp` ()**

**BEGIN**

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

**END$$**

**DELIMITER ;**

**call emp\_detail\_exp;**

-- 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.

**USE `employee`;**

**DROP function IF EXISTS `employee`.`Org\_set\_std`;**

**DELIMITER $$**

**USE `employee`$$**

**CREATE DEFINER=`root`@`localhost` FUNCTION `Org\_set\_std`(exp int) RETURNS varchar(50) CHARSET utf8mb4**

**DETERMINISTIC**

**BEGIN**

**declare role varchar(50);**

**if exp <= 2 then**

**set role ='Junior Data Scientist';**

**elseif exp> 2 and exp <= 5 then**

**set role ='Associate Data Scientist';**

**elseif exp> 5 and exp <= 10 then**

**set role ='Senior Data Scientist';**

**elseif exp> 10 and exp <= 12 then**

**set role ='Lead Data Scientist';**

**elseif exp> 12 and exp<= 16 then**

**set role ='Manager';**

**else**

**set role ='President';**

**end if;**

**RETURN role;**

**END$$**

**DELIMITER ;**

**select org\_set\_std(exp), exp from emp\_record\_table order by org\_set\_std(exp) asc;**

-- 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 indx\_first\_name on emp\_record\_table (first\_name);**

**SELECT \* FROM emp\_record\_table WHERE first\_name = 'Eric';**

**SHOW INDEX FROM emp\_record\_table;**

-- 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**

**salary,**

**emp\_rating,**

**ROUND((5 \* salary \* emp\_rating) / 100, 2) 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,**

**country,**

**ROUND(AVG(salary), 2) AS avg\_salary**

**FROM emp\_record\_table**

**GROUP BY continent, country;**