**-- Science Qtech Employee Performance Mapping --**

**create database employee;**

**use employee;**

**select \* from data\_science\_team;**

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

**select \* from proj\_table;**

**ALTER TABLE `employee`.`data\_science\_team`**

**MODIFY COLUMN `EMP\_ID` VARCHAR(10) NOT NULL,**

**ADD PRIMARY KEY (`EMP\_ID`);**

**ALTER TABLE `employee`.`emp\_record\_table`**

**MODIFY COLUMN `EMP\_ID` VARCHAR(10) NOT NULL,**

**ADD PRIMARY KEY (`EMP\_ID`);**

**ALTER TABLE `employee`.`proj\_table`**

**MODIFY COLUMN `PROJECT\_ID` VARCHAR(10) NOT NULL,**

**ADD PRIMARY KEY (`PROJECT\_ID`);**

**-- 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;**

**-- 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 OR EMP\_RATING > 4 OR (EMP\_RATING >= 2 AND EMP\_RATING <= 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 CONCAT(FIRST\_NAME, ' ', LAST\_NAME) AS NAME**

**FROM emp\_record\_table**

**WHERE DEPT = 'Finance';**

**-- Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President) --**

**SELECT e.EMP\_ID, e.FIRST\_NAME, e.LAST\_NAME, COUNT(r.EMP\_ID) AS num\_reporters**

**FROM emp\_record\_table e**

**LEFT JOIN emp\_record\_table r ON e.EMP\_ID = r.MANAGER\_ID**

**GROUP BY e.EMP\_ID, e.FIRST\_NAME, e.LAST\_NAME;**

**-- 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 EMP\_ID, FIRST\_NAME, LAST\_NAME, DEPT**

**FROM emp\_record\_table**

**WHERE DEPT = 'healthcare'**

**UNION**

**SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, DEPT**

**FROM emp\_record\_table**

**WHERE DEPT = 'finance';**

**-- 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 e.EMP\_ID, e.FIRST\_NAME, e.LAST\_NAME, e.ROLE, e.DEPT, e.EMP\_RATING, d.MAX\_EMP\_RATING**

**FROM emp\_record\_table e**

**JOIN (**

**SELECT DEPT, MAX(EMP\_RATING) AS MAX\_EMP\_RATING**

**FROM emp\_record\_table**

**GROUP BY DEPT**

**) d ON e.DEPT = d.DEPT;**

**-- 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, MIN(SALARY) AS MIN\_SALARY, MAX(SALARY) AS MAX\_SALARY**

**FROM emp\_record\_table**

**GROUP BY ROLE;**

**-- 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, EXP,**

**RANK() OVER (ORDER BY EXP DESC) AS experience\_rank**

**FROM emp\_record\_table;**

**-- 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 high\_salary\_employees\_view AS**

**SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, COUNTRY**

**FROM emp\_record\_table**

**WHERE SALARY > 6000;**

**select \* from high\_salary\_employees\_view;**

**-- 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, EXP**

**FROM emp\_record\_table**

**WHERE EXP > 10;**

**-- 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 GetExperiencedEmployees()**

**BEGIN**

**SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME**

**FROM emp\_record\_table**

**WHERE EXP > 3;**

**END;**

**//**

**DELIMITER ;**

**-- 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 --**

**-- TheFor 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'.**

**SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, EXP,**

**CASE**

**WHEN EXP <= 2 THEN 'JUNIOR DATA SCIENTIST'**

**WHEN EXP <= 5 THEN 'ASSOCIATE DATA SCIENTIST'**

**WHEN EXP <= 10 THEN 'SENIOR DATA SCIENTIST'**

**WHEN EXP <= 12 THEN 'LEAD DATA SCIENTIST'**

**ELSE 'MANAGER'**

**END AS EMP\_RANK**

**FROM emp\_record\_table**

**LIMIT 0, 1000;**

**-- 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\_firstname ON emp\_record\_table (FIRST\_NAME(50));**

**SELECT \* FROM emp\_record\_table WHERE FIRST\_NAME = 'Eric';**

**-- 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, FIRST\_NAME, LAST\_NAME, SALARY, EMP\_RATING,**

**0.05 \* SALARY \* EMP\_RATING AS BONUS**

**FROM emp\_record\_table;**

**-- 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, AVG(SALARY) AS AVG\_SALARY**

**FROM emp\_record\_table**

**GROUP BY CONTINENT, COUNTRY;**