/\*

SQL PROJECT 1:- By Divya Manupati

Science Qtech Employee Performance Mapping.

**DESCRIPTION:**

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.

**DATASET** **DESCRIPTION**:

**emp\_record\_table:** It contains the information of all the employees.

EMP\_ID – ID of the employee

FIRST\_NAME – First name of the employee

LAST\_NAME – Last name of the employee

GENDER – Gender of the employee

ROLE – Post of the employee

DEPT – Field of the employee

EXP – Years of experience the employee has

COUNTRY – Country in which the employee is presently living

CONTINENT – Continent in which the country is

SALARY – Salary of the employee

EMP\_RATING – Performance rating of the employee

MANAGER\_ID – The manager under which the employee is assigned

PROJ\_ID – The project on which the employee is working or has worked on

**Proj\_table:** It contains information about the projects.

PROJECT\_ID – ID for the project

PROJ\_Name – Name of the project

DOMAIN – Field of the project

START\_DATE – Day the project began

CLOSURE\_DATE – Day the project was or will be completed

DEV\_QTR – Quarter in which the project was scheduled

STATUS – Status of the project currently

**Data\_science\_team:** It contains information about all the employees in the Data Science team.

EMP\_ID – ID of the employee

FIRST\_NAME – First name of the employee

LAST\_NAME – Last name of the employee

GENDER – Gender of the employee

ROLE – Post of the employee

DEPT – Field of the employee

EXP – Years of experience the employee has

COUNTRY – Country in which the employee is presently living

CONTINENT – Continent in which the country is

\*/

**The** **task** **to** **be** **performed**:

/\* 1. Created 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;

use employee;

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

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema employee

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema employee

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `employee` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci ;

USE `employee` ;

-- -----------------------------------------------------

-- Table `employee`.`emp\_record\_table`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `employee`.`emp\_record\_table` (

`EMP\_ID` TEXT NOT NULL,

`FIRST\_NAME` TEXT NOT NULL,

`LAST\_NAME` TEXT NOT NULL,

`GENDER` TEXT NOT NULL,

`ROLE` TEXT NOT NULL,

`DEPT` TEXT NOT NULL,

`EXP` INT NOT NULL,

`COUNTRY` TEXT NOT NULL,

`CONTINENT` TEXT NOT NULL,

`SALARY` INT NOT NULL,

`EMP\_RATING` INT NOT NULL,

`MANAGER\_ID` TEXT NOT NULL,

`PROJ\_ID` TEXT NOT NULL,

`emp\_record\_tablecol` VARCHAR(45) NOT NULL,

`emp\_record\_tablecol1` VARCHAR(45) NOT NULL,

`data\_science\_team\_EMP\_ID` TEXT NOT NULL,

`data\_science\_team\_emp\_record\_table\_EMP\_ID` TEXT NOT NULL,

`data\_science\_team\_proj\_table\_PROJECT\_ID` TEXT NOT NULL,

`data\_science\_team\_proj\_table\_emp\_record\_table\_EMP\_ID` TEXT NOT NULL,

PRIMARY KEY (`EMP\_ID`),

INDEX `fk\_emp\_record\_table\_data\_science\_team1\_idx` (`data\_science\_team\_EMP\_ID` ASC, `data\_science\_team\_emp\_record\_table\_EMP\_ID` ASC, `data\_science\_team\_proj\_table\_PROJECT\_ID` ASC, `data\_science\_team\_proj\_table\_emp\_record\_table\_EMP\_ID` ASC) VISIBLE,

CONSTRAINT `fk\_emp\_record\_table\_data\_science\_team1`

FOREIGN KEY (`data\_science\_team\_EMP\_ID` , `data\_science\_team\_emp\_record\_table\_EMP\_ID` , `data\_science\_team\_proj\_table\_PROJECT\_ID` , `data\_science\_team\_proj\_table\_emp\_record\_table\_EMP\_ID`)

REFERENCES `employee`.`data\_science\_team` (`EMP\_ID` , `emp\_record\_table\_EMP\_ID` , `proj\_table\_PROJECT\_ID` , `proj\_table\_emp\_record\_table\_EMP\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `employee`.`proj\_table`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `employee`.`proj\_table` (

`PROJECT\_ID` TEXT NOT NULL,

`PROJ\_NAME` TEXT NOT NULL,

`DOMAIN` TEXT NOT NULL,

`START \_DATE` TEXT NOT NULL,

`CLOSURE\_DATE` TEXT NOT NULL,

`DEV\_QTR` TEXT NOT NULL,

`STATUS` TEXT NOT NULL,

`proj\_tablecol` VARCHAR(45) NOT NULL,

`proj\_tablecol1` VARCHAR(45) NOT NULL,

`emp\_record\_table\_EMP\_ID` TEXT NOT NULL,

`emp\_record\_table\_EMP\_ID1` TEXT NOT NULL,

`emp\_record\_table\_EMP\_ID2` TEXT NOT NULL,

PRIMARY KEY (`PROJECT\_ID`, `emp\_record\_table\_EMP\_ID`),

INDEX `fk\_proj\_table\_emp\_record\_table\_idx` (`emp\_record\_table\_EMP\_ID` ASC) VISIBLE,

INDEX `fk\_proj\_table\_emp\_record\_table1\_idx` (`emp\_record\_table\_EMP\_ID1` ASC) VISIBLE,

INDEX `fk\_proj\_table\_emp\_record\_table2\_idx` (`emp\_record\_table\_EMP\_ID2` ASC) VISIBLE,

CONSTRAINT `fk\_proj\_table\_emp\_record\_table`

FOREIGN KEY (`emp\_record\_table\_EMP\_ID`)

REFERENCES `employee`.`emp\_record\_table` (`EMP\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_proj\_table\_emp\_record\_table1`

FOREIGN KEY (`emp\_record\_table\_EMP\_ID1`)

REFERENCES `employee`.`emp\_record\_table` (`EMP\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_proj\_table\_emp\_record\_table2`

FOREIGN KEY (`emp\_record\_table\_EMP\_ID2`)

REFERENCES `employee`.`emp\_record\_table` (`EMP\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `employee`.`data\_science\_team`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `employee`.`data\_science\_team` (

`EMP\_ID` TEXT NOT NULL,

`FIRST\_NAME` TEXT NOT NULL,

`LAST\_NAME` TEXT NOT NULL,

`GENDER` TEXT NOT NULL,

`ROLE` TEXT NOT NULL,

`DEPT` TEXT NOT NULL,

`EXP` INT NOT NULL,

`COUNTRY` TEXT NOT NULL,

`CONTINENT` TEXT NOT NULL,

`data\_science\_teamcol` VARCHAR(45) NOT NULL,

`emp\_record\_table\_EMP\_ID` TEXT NOT NULL,

`proj\_table\_PROJECT\_ID` TEXT NOT NULL,

`proj\_table\_emp\_record\_table\_EMP\_ID` TEXT NOT NULL,

`proj\_table\_PROJECT\_ID1` TEXT NOT NULL,

`proj\_table\_emp\_record\_table\_EMP\_ID1` TEXT NOT NULL,

PRIMARY KEY (`EMP\_ID`, `emp\_record\_table\_EMP\_ID`, `proj\_table\_PROJECT\_ID`, `proj\_table\_emp\_record\_table\_EMP\_ID`),

INDEX `fk\_data\_science\_team\_emp\_record\_table1\_idx` (`emp\_record\_table\_EMP\_ID` ASC) VISIBLE,

INDEX `fk\_data\_science\_team\_proj\_table1\_idx` (`proj\_table\_PROJECT\_ID` ASC, `proj\_table\_emp\_record\_table\_EMP\_ID` ASC) VISIBLE,

INDEX `fk\_data\_science\_team\_proj\_table2\_idx` (`proj\_table\_PROJECT\_ID1` ASC, `proj\_table\_emp\_record\_table\_EMP\_ID1` ASC) VISIBLE,

CONSTRAINT `fk\_data\_science\_team\_emp\_record\_table1`

FOREIGN KEY (`emp\_record\_table\_EMP\_ID`)

REFERENCES `employee`.`emp\_record\_table` (`EMP\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_data\_science\_team\_proj\_table1`

FOREIGN KEY (`proj\_table\_PROJECT\_ID` , `proj\_table\_emp\_record\_table\_EMP\_ID`)

REFERENCES `employee`.`proj\_table` (`PROJECT\_ID` , `emp\_record\_table\_EMP\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_data\_science\_team\_proj\_table2`

FOREIGN KEY (`proj\_table\_PROJECT\_ID1` , `proj\_table\_emp\_record\_table\_EMP\_ID1`)

REFERENCES `employee`.`proj\_table` (`PROJECT\_ID` , `emp\_record\_table\_EMP\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

COLLATE = utf8mb4\_0900\_ai\_ci;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

/\* 3. Written a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, and DEPARTMENT

from the employee record table, and made a list of employees and details of their department.\*/

select emp\_id, first\_name, last\_name, gender, dept from employee.emp\_record\_table;

select first\_name, last\_name, dept from employee.emp\_record\_table;

/\* 4. Written 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

employee.emp\_record\_table where EMP\_RATING < 2;

select emp\_id, first\_name, last\_name, gender, dept, emp\_rating from

employee.emp\_record\_table where EMP\_RATING > 4;

select emp\_id, first\_name, last\_name, gender, dept, emp\_rating from

employee.emp\_record\_table where EMP\_RATING between 2 and 4;

/\* 5. Written a query to concatenate the FIRST\_NAME and the LAST\_NAME of employees in the

Finance department from the employee table and then given the resultant column alias as

NAME.\*/

select first\_name, last\_name, dept, concat(first\_name,' ',Last\_name) as Name from

employee.emp\_record\_table where dept = 'Finance';

/\* 6. Written a query to list only those employees who have someone reporting to them. Also, shown

the number of reporters (including the President).\*/

SELECT DISTINCT emp\_id, role FROM emp\_record\_table

WHERE emp\_id IN (SELECT manager\_id FROM emp\_record\_table);

/\* 7. Written a query to list down all the employees from the healthcare and finance departments

using union. Data taken from the employee record table.\*/

select first\_name, last\_name, dept, emp\_id from employee.emp\_record\_table where

DEPT = "Healthcare"

union

select first\_name, LAST\_NAME, DEPT,EMP\_ID from employee.emp\_record\_table where DEPT =

"Finance";

/\* 8. Written 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

FROM employee.emp\_record\_table

JOIN (

SELECT dept, MAX(emp\_rating) AS max\_emp\_rating

FROM employee.emp\_record\_table

GROUP BY dept

) AS max\_ratings

USING (dept);

/\* 9. Written a query to calculate the minimum and the maximum salary of the employees in each

role. Data taken from the employee record table.\*/

select min(salary), max(salary), role from employee.emp\_record\_table group by role;

/\* 10. Written a query to assign ranks to each employee based on their experience. Data taken from

the employee record table.\*/

select first\_name, Last\_name, emp\_id, exp, rank() over (order by exp desc) as emp\_rank

from employee.emp\_record\_table;

/\*11. Written a query to create a view that displays employees in various countries whose salary is

more than six thousand. Data taken from the employee record table.\*/

create view emp\_country as select emp\_id, concat(first\_name,' ', last\_name) as Name, salary,

country from employee.emp\_record\_table where salary > 6000;

select \* from emp\_country;

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

from the employee record table. \*/

select concat(first\_name,' ',last\_name)as 'Name', EXP

from employee.emp\_record\_table

where EXP>10;

/\*13. Written a query to create a stored procedure to retrieve the details of the employees whose

experience is more than three years. Data taken from the employee record table.\*/

delimiter //

create procedure Experiencedemployees ()

begin

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

end //

delimiter ;

call Experiencedemployees ();

/\*14. Written 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 $$

CREATE FUNCTION check\_job\_profile(exp integer)

RETURNS VARCHAR(40)

DETERMINISTIC

BEGIN

DECLARE chck VARCHAR(40);

if exp < 2 THEN SET chck = "JUNIOR DATA SCIENTIST";

elseif exp >=2 AND exp < 5 THEN SET chck = "ASSOCIATE DATA SCIENTIST";

elseif exp >=5 AND exp < 10 THEN SET chck = "SENIOR DATA SCIENTIST";

elseif exp >= 10 AND exp < 12 THEN SET chck = "LEAD DATA SCIENTIST";

elseif exp >= 12 THEN SET chck = "MANAGER";

end if; RETURN (chck);

END $$

delimiter ;

-- Checking Data Science Team

SELECT emp\_id, first\_name, last\_name, role, check\_job\_profile(exp)

FROM data\_science\_team WHERE ROLE != check\_job\_profile(exp);

/\* 15. Created 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 idx1\_firstname ON emp\_record\_table(FIRST\_NAME(10));

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

/\* 16. Written 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 employee.emp\_record\_table;

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

Data taken from the employee record table.\*/

SELECT continent, country, round(avg(salary)) AS average\_salary

FROM employee.emp\_record\_table

GROUP BY continent, country;

/\* Project end \*/