/*

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';
```

```sql
/* 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);
```

```sql
/* 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";
```

```sql
/* 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);
```

```sql
/* 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 */
```