Employee Performance Mapping

-- ---------- action 1 --

Create a database named employee,

```
CREATE DATABASE employee;
```

then import data_science_team.csv proj_table.csv and emp_record_table.csv into the employee database from the given resources.

```
use employee;

show databases;

show tables;
```

-- ---------- action 2 --

Create an ER diagram for the given employee database.

View in screen shot file

-- ---------- action 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 *
from employee.emp_record_table;

select emp_id, first_name, last_name, gender, dept
from employee.emp_record_table;

# Select → Mention the column (s) to be fetched
# From → Mention the table(s) from where the data needs to be fetched
# Where → Filter the data
```

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

---------- action 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
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;
```

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

---------- action 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 employee.emp_record_table
where dept = 'finance';
```

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

---------- action 6 --

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

प्रश्न के अनुसार, जिन कर्मचारियों के अधीन कुछ रिपोर्टर (अधीनस्थ कर्मचारी) हैं, उन्हें सूचीबद्ध करना है |इसीलिए मुजे inner join का use करना होगा

```
select
m.EMP_ID,
concat(m.first_name,'  ',m.last_name) AS MANAGER_NAME,
m.ROLE,
m.DEPT,
COUNT(r.EMP_ID) AS number_of_reporters  ## रिपोर्टरों की अनुमानित संख्या दर्शाई जानी चाहिए।
from emp_record_table as m
join emp_record_table as r
on m.emp_id=r.manager_id
GROUP BY     ## error code मे group by करनेको कहगाया है | क्योकि आगे हमने count का उपयोग किया है इसीलिए
m.EMP_ID, m.FIRST_NAME, m.LAST_NAME, m.ROLE, m.DEPT
ORDER BY                    ## अध्यक्ष  को पहेले recode मे दिखाय देना चाहिए
number_of_reporters DESC;


-- ------------------------------------------------------------------------------------------------------------
---------- action 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 in ('healthcare', 'finance');

SELECT * FROM emp_record_table WHERE DEPT = 'HEALTHCARE'
UNION
SELECT * FROM emp_record_table WHERE DEPT = 'FINANCE';


-- ------------------------------------------------------------------------------------------------------------
---------- acton 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.
```

प्रत्येक कर्मचारी का विवरण दिखाया जाना है (EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPARTMENT, and EMP_RATING)
इसके अलावा, DEPARTMENT से संबद्ध प्रत्येक कर्मचारी के लिए, उच्चतम EMP_RATING वाला कर्मचारी भी दिखाना है।
लेकिन प्रत्येक कर्मचारी के विवरण के साथ grouped by dept सीधे SQL में संभव नहीं है।
इसलिए, GROUP BY के बजाय, WINDOW FUNCTION का उपयोग करना सही होगा , अर्थात MAX(EMP_RATING) OVER (PARTITION BY DEPT), ताकि प्रत्येक पंक्ति में उस विभाग की उच्चतम रेटिंग दिखाई जा सके।

```
SELECT
        EMP_ID,
        concat(first_name,'  ',last_name) AS NAME,
  ROLE,
  DEPT,
  EMP_RATING,
  MAX(EMP_RATING) OVER (PARTITION BY DEPT) AS MAX_DEPT_RATING
FROM emp_record_table;
```

-- ---------- action 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
 ROLE,
 MIN(SALARY) AS MIN_SALARY,
 MAX(SALARY) AS MAX_SALARY
FROM emp_record_table
GROUP BY ROLE;
```

-- ---------- action 10 --

Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.
use WINDOW FUNCTION -> ranking functions -> rank()

```
select
        EMP_ID,
        concat(first_name,'  ',last_name) as NAME,
   exp AS EXPERIENCE,
   rank() over(order by exp desc) as EXPERIENCE_RANK
from emp_record_table;


-- ------------------------------------------------------------------------------------------------------------------
--------------------------- action 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 HighSalaryEmployees as
select *
from emp_record_table
where salary > 6000;

SELECT * FROM HighSalaryEmployees;


-- ------------------------------------------------------------------------------------------------------------------
------------------------------- action 12 --
```

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

```
select *
from emp_record_table
where exp > 10;

select *
from emp_record_table
where emp_id in (
   select emp_id
   from emp_record_table
        where exp > 10
);
```

-- ---------- action 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.

stored procedure में BEGIN...END structure चाहिये, BEGIN...END के बीच ';' delimiter का उपयोग होता है इसीलिए हमे नया delimiter '//' set करना होगा |

```
delimiter //
CREATE PROCEDURE Get_Experience_Employees()
BEGIN
  SELECT *
  FROM emp_record_table
  WHERE EXP > 3;
end //
delimiter ;   # You need to put a space in the code.

CALL Get_Experience_Employees();
```

-- ---------- action 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.

```
-- 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 //        # statements का अंत // पर होगा, default ; से नहीं।
```

SQL stored functions के लिए RETURN ज़रूरी है।

```
CREATE FUNCTION GetStandardJobProfile(exp INT) RETURNS VARCHAR(50)
DETERMINISTIC
BEGIN
 DECLARE profile VARCHAR(50);  #  job profile value store होगी
```

सारी coding python coding जेसी होगी

```
 IF exp <= 2 THEN
  SET profile = 'JUNIOR DATA SCIENTIST';
 ELSEIF exp > 2 AND exp <= 5 THEN
  SET profile = 'ASSOCIATE DATA SCIENTIST';
 ELSEIF exp > 5 AND exp <= 10 THEN
  SET profile = 'SENIOR DATA SCIENTIST';
 ELSEIF exp > 10 AND exp <= 12 THEN
  SET profile = 'LEAD DATA SCIENTIST';
 ELSEIF exp > 12 AND exp <= 16 THEN
  SET profile = 'MANAGER';
 ELSE
  SET profile = 'UNKNOWN';
 END IF;

 RETURN profile;
END //

DELIMITER ;       # पहला डिलिमिटर ; पर आने के लिए

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

SELECT
 e.emp_id,
 e.exp,
 p.PROJ_NAME,
 GetStandardJobProfile(e.exp) AS expected_profile,
 CASE
  WHEN p.PROJ_NAME = GetStandardJobProfile(e.exp) THEN 'MATCH'
  ELSE 'MISMATCH'                    #  then...else statement
 END AS result
FROM
 emp_record_table e
JOIN
 proj_table p
ON e.PROJ_ID = p.PROJECT_ID;


-- ------------------------------------------------------------------------------------------------------------
--------------------------------- action 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.

जब आप किसी कॉलम पर इंडेक्स बनाते हैं, तो यह तेज़ी से सर्च करने के लिए पॉइंटर की तरह काम करता है, इसलिए क्वेरीज़ तेज़ी से चलती हैं।

EXPLAIN का इस्तेमाल करके आप देख सकते हैं कि क्वेरी में इंडेक्स इस्तेमाल किए गए हैं या नहीं।

```
EXPLAIN SELECT * FROM emp_record_table WHERE FIRST_NAME = 'Eric';

CREATE INDEX idx_first_name ON emp_record_table(FIRST_NAME(50));
```

-- ---------- action 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
 EMP_ID,
 concat(first_name,'  ',last_name) AS NAME,
 SALARY,
 EMP_RATING,
 (SALARY * 0.05 * EMP_RATING) AS BONUS
FROM emp_record_table;
```

-- ---------- action 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,
 AVG(SALARY) AS Average_Salary
FROM emp_record_table
GROUP BY CONTINENT, COUNTRY
ORDER BY CONTINENT, COUNTRY;
```