RDBMS ASSIGNMENT

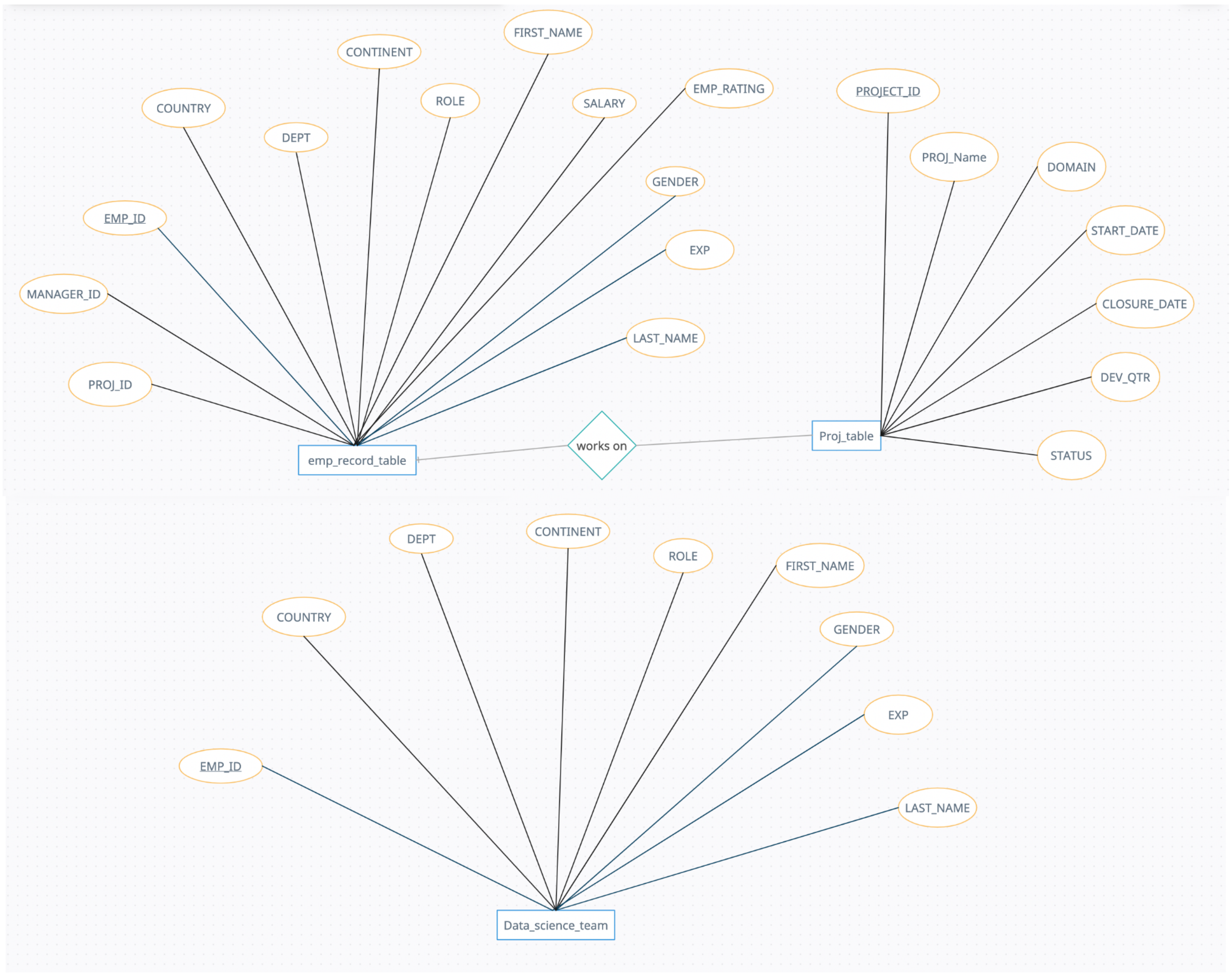
QUESTION 1 :-

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 RDBMS_ASSIGNMENT;
USE RDBMS_ASSIGNMENT;
```

After this, I imported data_science_team.csv, proj_table.csv and emp_record_table.csv into the **RDBMS_ASSIGNMENT** database from the given resources using MySQLWorkbench.

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


FIRST_NAME
CONTINENT
EMP_RATING
PROJECT_ID
ROLE
SALARY
COUNTRY
DEPT
PROJ_Name
DOMAIN
GENDER
EMP_ID
START_DATE
EXP
MANAGER_ID
CLOSURE_DATE
LAST_NAME
PROJ_ID
DEV_QTR
works on
Proj_table
STATUS
emp_record_table
CONTINENT
DEPT
ROLE
FIRST_NAME
COUNTRY
GENDER
EXP
EMP_ID
LAST_NAME
Data_science_team

QUESTION 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.

QUERY:

```
SELECT EMP_ID, FIRST_NAME,LAST_NAME,GENDER,DEPT
 FROM emp_record_table;
```

QUESTION 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**

QUERY:

-
  ```
  SELECT EMP_ID, FIRST_NAME,LAST_NAME,GENDER,DEPT ,EMP_RATING
  FROM emp_record_table
  WHERE EMP_RATING<2;
  ```

-
  ```
  SELECT EMP_ID, FIRST_NAME,LAST_NAME,GENDER,DEPT ,EMP_RATING
  FROM emp_record_table
  WHERE EMP_RATING>4;
  ```

-
  ```
  SELECT EMP_ID, FIRST_NAME,LAST_NAME,GENDER,DEPT ,EMP_RATING
  FROM emp_record_table
  WHERE EMP_RATING BETWEEN 2 AND 4;
  ```

QUESTION 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.

QUERY:

```
SELECT CONCAT(FIRST_NAME,' ',LAST_NAME) AS NAME FROM emp_record_table
WHERE DEPT='FINANCE';
```

QUESTION 6 :-

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

QUERY:

```
SELECT  M.EMP_ID , M.FIRST_NAME ,M.LAST_NAME , COUNT(*) AS CNT
FROM emp_record_table  E INNER JOIN emp_record_table M
ON E.MANAGER_ID=M.EMP_ID
GROUP BY M.EMP_ID;
```

QUESTION 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.

QUERY:

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

QUESTION 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.

QUERY:

```
SELECT EMP_ID,FIRST_NAME,LAST_NAME,ROLE,DEPT,EMP_RATING,
 MAX(EMP_RATING)  OVER(PARTITION BY DEPT) AS MAX_RATING_BY_DEPT
 FROM emp_record_table;
```

QUESTION 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.

QUERY:

```
SELECT DISTINCT ROLE, MAX(SALARY) OVER(PARTITION BY ROLE) AS
MAX_SALARY ,MIN(SALARY)    OVER(PARTITION BY ROLE)
AS MIN_SALARY  FROM emp_record_table;
```

QUESTION 10 :-

Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.

QUERY:

```
SELECT EMP_ID ,EXP, DENSE_RANK() OVER(ORDER BY EXP DESC) AS
RANK_BY_EXP
 FROM emp_record_table ;
```

QUESTION 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.

QUERY:

```
CREATE OR REPLACE VIEW SALAR_GREATER_THEN_SIX_THOUSAND AS
SELECT EMP_ID,COUNTRY,SALARY FROM emp_record_table
 WHERE SALARY>6000;

SELECT * FROM SALAR_GREATER_THEN_SIX_THOUSAND;
```

QUESTION 12 :-

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

QUERY:

```
SELECT *
FROM emp_record_table
WHERE EXP IN (SELECT EXP FROM emp_record_table WHERE EXP>10);
```

QUESTION 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.

QUERY:

```
DELIMITER $$
CREATE PROCEDURE Solve()
BEGIN
SELECT *
FROM emp_record_table
WHERE EXP>3;
END
$$ DELIMITER ;

CALL Solve();
```

QUESTION 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'.

QUERY:

```
 DELIMITER $$
 CREATE FUNCTION SCIENTIST_EXP2()
 RETURNS TINYINT(1)
 DETERMINISTIC
 BEGIN
 DECLARE RES INT DEFAULT 1;
 DECLARE RL VARCHAR(30);
 DECLARE SETEND INTEGER DEFAULT 0;
 DECLARE EP INT;
 DECLARE CURNAME CURSOR FOR
 SELECT ROLE,EXP FROM emp_record_table;
 DECLARE CONTINUE HANDLER FOR NOT FOUND
 SET SETEND=1;

OPEN CURNAME;
MY_LOOP:LOOP
FETCH CURNAME INTO RL,EP;
IF SETEND=1 THEN
LEAVE MY_LOOP;
END IF;

IF(RL='JUNIOR DATA SCIENTIST' AND EP>2) THEN
SET RES=0;
LEAVE MY_LOOP;
ELSEIF(RL='ASSOCIATE DATA SCIENTIST' AND (EP<=2 OR EP>5)) THEN
SET RES=0;
LEAVE MY_LOOP;
ELSEIF(RL='SENIOR DATA SCIENTIST' AND (EP<=5 OR EP>10)) THEN
SET RES=0;
LEAVE MY_LOOP;
ELSEIF(RL='LEAD DATA SCIENTIST' AND (EP<=10 OR EP>12)) THEN
SET RES=0;
LEAVE MY_LOOP;
```

```
 ELSEIF(RL='MANAGER' AND (EP<=12 OR EP>16)) THEN
 SET RES=0;
 LEAVE MY_LOOP;
 END IF;
 END LOOP MY_LOOP;
 CLOSE CURNAME;
 IF(RES=1) THEN
 RETURN TRUE;
 ELSE
 RETURN FALSE;
 END IF;
 END
 $$
 DELIMITER ;




 DELIMITER $$
 CREATE PROCEDURE Helper_Procedure()
 BEGIN
IF SCIENTIST_EXP2() THEN
SELECT 'THE PROFILE ASSIGNED TO EACH EMPLOYEE IN THE DATA SCIENCE
TEAM MATCHES THE ORGANIZATION SET STANDARD.' AS MESSAGE;
ELSE
SELECT 'THE PROFILE ASSIGNED TO EACH EMPLOYEE IN THE DATA SCIENCE
TEAM DOES NOT MATCH THE ORGANIZATION SET STANDARD.' AS MESSAGE;
END IF;
END $$
DELIMITER ;

CALL Helper_Procedure();
```

QUESTION 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.

QUERY:

```
CREATE INDEX Eric_Index
ON emp_record_table(FIRST_NAME);

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

QUESTION 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).

QUERY:

```
SELECT EMP_ID,SALARY,EMP_RATING, (SALARY*0.05*EMP_RATING)
AS BONUS FROM emp_record_table ;
```

QUESTION 17 :-

Write a query to calculate the average salary distribution based on the continent and country.Take data from the employee record table.

QUERY:

```
SELECT DISTINCT CONTINENT,COUNTRY ,
AVG(SALARY) OVER(PARTITION BY CONTINENT) AS
AVG_SALARY_BY_CONTINENT,
AVG(SALARY) OVER(PARTITION BY COUNTRY) AS
AVG_SALARY_BY_COUNTRY
FROM emp_record_table;
```