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 TABLE Proj\_table(

PROJECT\_ID varchar(50),

PROJ\_Name varchar(50),

DOMAIN varchar(50),

START\_DATE date,

CLOSURE\_DATE date,

DEV\_QTR varchar(50),

STATUS varchar(50));

alter table proj\_table

add primary key(PROJECT\_ID);

CREATE TABLE emp\_record\_table(

EMP\_ID varchar(50),

FIRST\_NAME varchar(50),

LAST\_NAME varchar(50),

GENDER varchar(50),

ROLE varchar(50),

DEPT varchar(50),

EXP varchar(50),

COUNTRY varchar(50),

CONTINENT varchar(50),

SALARY int,

EMP\_RATING int,

MANAGER\_ID varchar(50),

PROJ\_ID varchar(50)

);

create table Data\_science\_team(

EMP\_ID varchar(50),

FIRST\_NAME varchar(50),

LAST\_NAME varchar(50),

GENDER varchar(50),

ROLE varchar(50),

DEPT varchar(50),

EXP int,

COUNTRY varchar(50),

CONTINENT varchar(50)

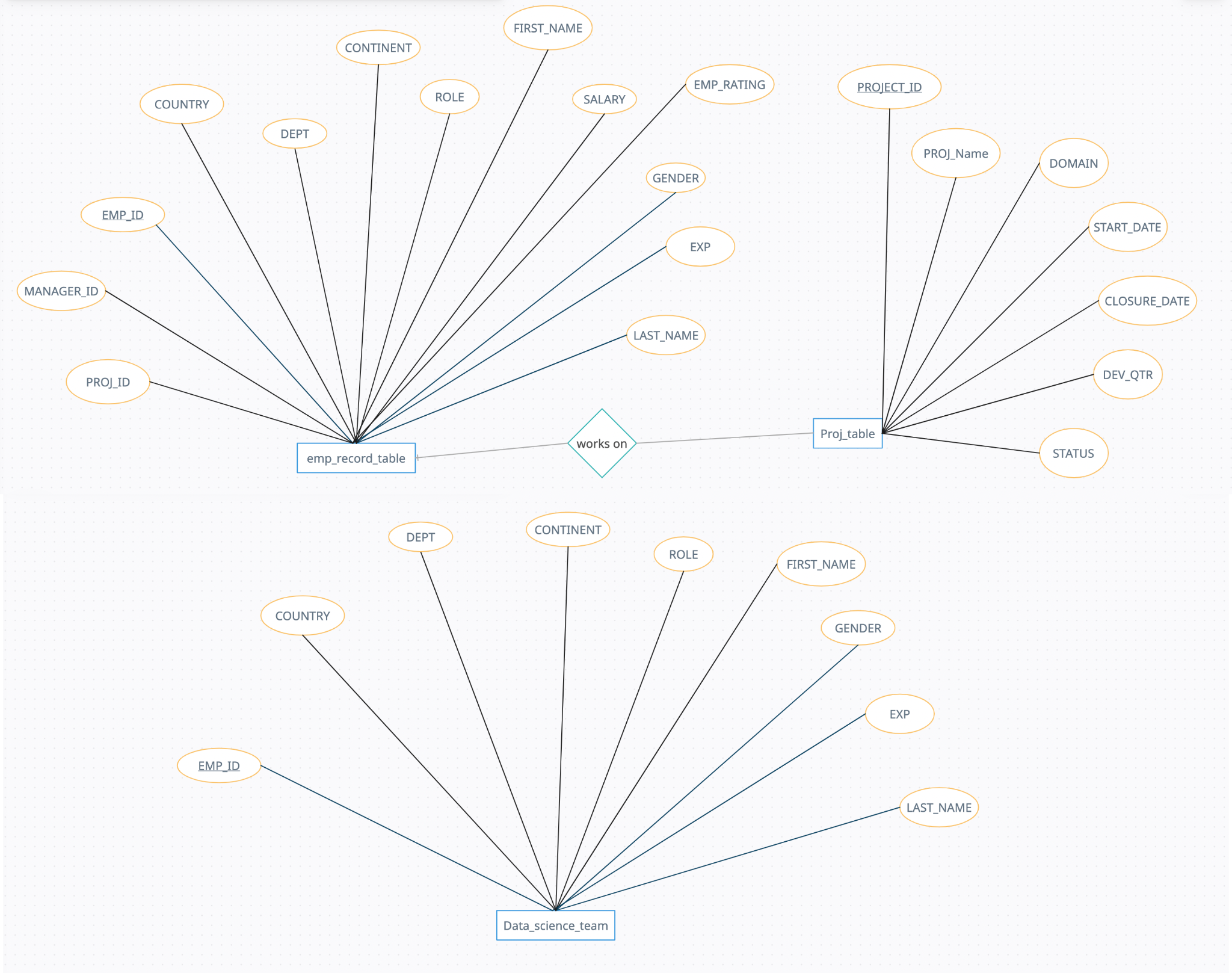
);

alter table emp\_record\_table

add foreign key (PROJ\_ID) references proj\_table(PROJECT\_ID);

**– question 2**

**Create an ER diagram for the given employee database.**

****

**-- 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.**

SELECT EMP\_ID, FIRST\_NAME,LAST\_NAME,GENDER,DEPT DEPARTMENT

FROM EMP\_RECORD\_TABLE;

**-- QUESTION 4**

**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 DEPARTMENT

FROM emp\_record\_table

WHERE EMP\_RATING <2;

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT DEPARTMENT

FROM emp\_record\_table

WHERE EMP\_RATING >4;

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT DEPARTMENT

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.**

SELECT CONCAT(FIRST\_NAME,LAST\_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);.**

SELECT E1.EMP\_ID,COUNT(E1.EMP\_ID)

FROM EMP\_RECORD\_TABLE E1,EMP\_RECORD\_TABLE E2

WHERE E1.EMP\_ID=E2.MANAGER\_ID

GROUP BY E1.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.**

SELECT \*

FROM EMP\_RECORD\_TABLE

WHERE DEPT IN ('HEALTHCARE', 'FINANCE');

**-- QUESTION 8**

**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.**

SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,ROLE,DEPT,EMP\_RATING,

(SELECT MAX(EMP\_RATING) FROM EMP\_RECORD\_TABLE WHERE DEPT=E1.DEPT),

MAX\_RATING

FROM EMP\_RECORD\_TABLE E1

ORDER BY DEPT

**-- 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.**

SELECT DEPT,MAX(SALARY),MIN(SALARY)

FROM EMP\_RECORD\_TABLE

GROUP BY DEPT

**-- QUESTION 10**

**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

FROM EMP\_RECORD\_TABLE

ORDER BY EXP DESC

**– 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.**

CREATE VIEW rich\_emp AS

SELECT emp\_id,country,salary

FROM emp\_record\_table

WHERE salary>6000;

SELECT \* FROM rich\_emp;

**– question 12**

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

SELECT emp\_id,exp

FROM (SELECT \*

FROM emp\_record\_table

WHERE exp>10) e;

**– 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.**

DELIMITER $$

CREATE PROCEDURE procedure1()

BEGIN

SELECT \*

FROM emp\_record\_table

WHERE exp>3;

END $$ DELIMITER ;

call procedure1();

**– 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'.**

DELIMITER $$

CREATE FUNCTION ROLE\_CHECKER(

v\_empid int) RETURNS BOOLEAN DETERMINISTIC

BEGIN

declare p\_exp int;

declare p\_role varchar(30);

SELECT exp,role INTO p\_exp, p\_role

FROM data\_science\_team

WHERE emp\_id=v\_empid;

IF (p\_exp<=2 AND p\_role='JUNIOR DATA SCIENTIST') THEN

RETURN TRUE;

ELSE IF (p\_exp<5 AND p\_exp>=2 AND p\_role= 'ASSOCIATE DATA SCIENTIST') THEN

RETURN TRUE;

ELSE IF (p\_exp<10 AND p\_exp>=5 AND p\_role= 'SENIOR DATA SCIENTIST') THEN

RETURN TRUE;

ELSE IF (p\_exp<12 AND p\_exp>=10 AND p\_role= 'LEAD DATA SCIENTIST') THEN

RETURN TRUE;

ELSE IF (p\_exp<16 AND p\_exp>=12 AND p\_role= 'MANAGER' ) THEN

RETURN TRUE;

ELSE RETURN FALSE;

END IF;

END IF;

END IF;

END IF;

END IF;

END

$$ DELIMITER ;

SELECT ROLE\_CHECKER('E006')

select \* from data\_science\_team

**– 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.**

CREATE INDEX INDEX1

ON emp\_record\_table(first\_name);

SELECT \*

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).**

SELECT first\_name,salary,(salary\*emp\_rating\*0.05) 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.**

select \* from emp\_record\_table

SELECT first\_name,salary,country,continent,

AVG(salary) OVER(PARTITION BY continent) avg\_cont\_sal,salary,

AVG(salary) OVER(PARTITION BY country) avg\_country\_sal

FROM emp\_record\_table;