-- **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;

CREATE TABLE emp\_record\_table(

EMP\_ID varchar(50);,

FIRST\_NAME varchar(50)

LAST\_NAME varchar(50),

GENDER varchar(1),

ROLE varchar(50),

DEPT varchar(55),

EXP int,

COUNTRY varchar(55),

CONTINENT varchar(58),

SALARY int,

EMP\_RATING int,

MANAGER\_ID varchar(50),

PROJ\_ID varchar(50); references Proj\_table(PROJECT\_ID)

);

CREATE TABLE Proj\_table(

PROJECT\_ID varchar(50) primary key,

PROJ\_Name varchar(50),

DOMAIN varchar(50),

START\_DATE date,

CLOSURE\_DATE date,

DEV\_QTR varchar(50),

STATUS varchar(50);

);

CREATE TABLE Data\_science\_team

(

EMP\_ID varchar(20); primary key,

FIRST\_NAME varchar(250),

LAST\_NAME varchar(50),

GENDER varchar(5),

ROLE varchar(500),

DEPT varchar(30),

EXP int,

COUNTRY varchar(30),

CONTINENT varchar(30)

);

**-- 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) 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 OR REPLACE VIEW SAL\_VIEW AS

SELECT \* FROM EMP\_RECORD\_TABLE

WHERE SALARY>6000

SELECT \* FROM SAL\_VIEW

**-- QUESTION 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 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.

DELIMITER $$

CREATE PROCEDURE STORE()

BEGIN

SELECT \*

FROM EMP\_RECORD\_TABLE

WHERE EXP>3;

END $$

CALL STORE();

**-- 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'.

CREATE DEFINER=`root`@`localhost` FUNCTION `ROLE\_CHECK`( EMP\_ID\_TMP VARCHAR(20)) RETURNS tinyint(1)

DETERMINISTIC

BEGIN

DECLARE ROLE\_TMP VARCHAR(30);

DECLARE EXP\_TMP INT;

DECLARE FLAG BOOL;

SET FLAG=FALSE;

SELECT EXP INTO EXP\_TMP

FROM EMP\_RECORD\_TABLE

WHERE EMP\_ID=EMP\_ID\_TMP;

SELECT ROLE INTO ROLE\_TMP

FROM EMP\_RECORD\_TABLE

WHERE EMP\_ID=EMP\_ID\_TMP;

IF(EXP\_TMP<=2 AND ROLE\_TMP='JUNIOR DATA SCIENTIST') THEN

SET FLAG=TRUE;

END IF;

IF(EXP\_TMP>2 AND EXP\_TMP<=5 AND ROLE\_TMP='ASSOCIATE DATA SCIENTIST')

THEN

SET FLAG=TRUE;

END IF;

IF(EXP\_TMP>5 AND EXP\_TMP<=10 AND ROLE\_TMP='SENIOR DATA SCIENTIST') THEN

SET FLAG=TRUE;

END IF;

IF(EXP\_TMP>10 AND EXP\_TMP<=12 AND ROLE\_TMP='LEAD DATA SCIENTIST') THEN

SET FLAG=TRUE;

END IF;

IF(EXP\_TMP>12 AND EXP\_TMP<=16 AND ROLE\_TMP='MANAGER') THEN

SET FLAG=TRUE;

END IF;

RETURN FLAG;

END

$$

-- procedure to check all members with the help of role\_check function

DELIMITER $$

CREATE PROCEDURE ALL\_CHECK()

BEGIN

SELECT EMP\_ID, ROLE\_CHECK(EMP\_ID) FROM Data\_science\_team;

end

$$

call ALL\_CHECK()

**--Question 15**

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 IDX

ON EMP\_RECORD\_TABLE(FIRST\_NAME)

SELECT \*

FROM EMP\_RECORD\_TABLE

WHERE FIRST\_NAME='Eric'

**-- Question 16**

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, (.05\*SALARY\*EMP\_RATING)

BONUS, SALARY,EMP\_RATING

FROM EMP\_RECORD\_TABLE

**– Question 17**

17. Write a query to calculate the average salary distribution based on the continent and country.

Take data from the employee record table.

SELECT COUNTRY, AVG(SALARY)

FROM EMP\_RECORD\_TABLE

GROUP BY COUNTRY

SELECT CONTINENT,AVG(SALARY)

FROM EMP\_RECORD\_TABLE

GROUP BY CONTINENT