**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 employee;

**2.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

from emp\_record\_table;

**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 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

**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 emp\_record\_table

where Dept = 'Finance'

**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

e.EMP\_ID,

e.FIRST\_NAME,

COUNT(DISTINCT r.EMP\_ID) AS NumberOfReporters

FROM

emp\_record\_table e

JOIN

emp\_record\_table r ON e.EMP\_ID = r.MANAGER\_ID

GROUP BY

e.EMP\_ID, e.FIRST\_NAME

HAVING

COUNT(DISTINCT r.EMP\_ID) > 0

ORDER BY

e.EMP\_ID;

**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 = 'Finance'

Union

Select \* From emp\_record\_table

where Dept = 'Healthcare'

**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 e.EMP\_ID, e.FIRST\_NAME, e.LAST\_NAME, e.ROLE, e.Dept, e.EMP\_RATING, d.MAX\_EMP\_RATING

From emp\_record\_table e

Join(Select Dept, Max(Emp\_rating) AS MAX\_EMP\_RATING

From emp\_record\_table

Group By Dept)

d ON e.Dept = d.Dept;

**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;

**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,

RANK() OVER (ORDER BY EXP DESC) AS EXPERIENCE\_RANK

FROM emp\_record\_table;

**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 High\_Salary\_Employees AS

Select EMP\_ID, FIRST\_NAME, LAST\_NAME, COUNTRY, Salary

FROM emp\_record\_table

Where salary > 6000

SELECT \* FROM High\_Salary\_Employees;

**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, FIRST\_NAME, LAST\_NAME, EXP

FROM emp\_record\_table

WHERE EXP > 10;

**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 EmployeesWithHighExperience()

BEGIN

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, EXP

FROM emp\_record\_table

WHERE EXP > 3;

END //

CALL EmployeesWithHighExperience();

**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 CheckJobProfile(exp INT) RETURNS VARCHAR(50)

READS SQL DATA

DETERMINISTIC

BEGIN

DECLARE job\_profile VARCHAR(50);

IF exp <= 2 THEN

SET job\_profile = 'JUNIOR DATA SCIENTIST';

ELSEIF exp > 2 AND exp <= 5 THEN

SET job\_profile = 'ASSOCIATE DATA SCIENTIST';

ELSEIF exp > 5 AND exp <= 10 THEN

SET job\_profile = 'SENIOR DATA SCIENTIST';

ELSEIF exp > 10 AND exp <= 12 THEN

SET job\_profile = 'LEAD DATA SCIENTIST';

ELSEIF exp > 12 AND exp <= 16 THEN

SET job\_profile = 'MANAGER';

ELSE

SET job\_profile = 'Unknown';

END IF;

RETURN job\_profile;

END //

DELIMITER ;

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME,

CheckJobProfile(exp) AS ASSIGNED\_JOB\_PROFILE

FROM Data\_science\_team;

**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\_first\_name ON emp\_record\_table (FIRST\_NAME(50));

EXPLAIN SELECT \* FROM emp\_record\_table WHERE FIRST\_NAME = 'Eric';

**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, FIRST\_NAME, LAST\_NAME,

SALARY \* (EMP\_RATING \* 0.05) AS BONUS

FROM emp\_record\_table;

**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;