Below task to be performed:

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.

   use employee;
   show tables;

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;

3. 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;

4. 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';

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

```
SELECT MANAGER_ID AS Employee_ID, COUNT(*) AS Number_of_Reporters
from emp_record_table
WHERE MANAGER_ID IS NOT NULL GROUP BY MANAGER_ID;
```

6. 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 concat(first_name , ' ',last_name ) as Name from emp_record_table
where dept='Finance'
union all
select concat(first_name , ' ' ,last_name ) as Name from emp_record_table
where dept='HealthCare';
```

7. 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 ,
max(emp_rating) over(order by dept) as Max_Rating
from emp_record_table;
```

8. 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;
```

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

```
select emp_id, concat(first_name , last_name) as name , exp ,  rank() over(order
by exp desc) as rank_exp from emp_record_table;
```

10. 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 country as
select emp_id, first_name , last_name , country , salary from emp_record_table
where salary > 6000;
```

```
select * from country;
```

11. 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>(select min(exp)+10 from emp_record_table) ;
    ```

12. 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.

    ```
    CREATE DEFINER=`root`@`localhost` PROCEDURE `emp_details`()
    BEGIN
            select * from emp_record_table where exp>3;
    END

    CALL `employee`.`emp_details`();
    ```

13. 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'.

```sql
select emp_id, first_name , last_name , exp, role,
case
    when exp<=2 then 'Junior scientist'
    when exp>2 and exp<=5 then 'associate data scientist'
    when exp>5 and exp<=10 then 'senior data scientist'
    when exp>10 and exp<=12 then 'lead data scientist'
    when exp>12 and exp<=16 then 'Manager'
    else 'other'
end as expected_role,
case
    when role=(case
                when exp<=2 then 'Junior scientist'
                when exp>2 and exp<=5 then 'associate data scientist'
                when exp>5 and exp<=10 then 'senior data scientist'
                when exp>10 and exp<=12 then 'lead data scientist'
                when exp>12 and exp<=16 then 'Manager'
                else 'other'
                end )
    then 'Match'
    else 'Mismatch'
end as status
from emp_record_table;
```

14. 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 i1 ON emp_record_table(first_name);
    SELECT * FROM employee.emp_record_table WHERE FIRST_NAME = 'Eric';

15. 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 * salary * emp_rating) as bonus from emp_record_table;

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

    select country , continent , avg(salary) as Avg_Salary from emp_record_table group by continent , country order by continent , country;