```sql
-- 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;
use employee;
show tables;

show full tables where table_type = "BASE TABLE";


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


-- 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 FROM
employee.emp_record_table
ORDER BY DEPT;

/* 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 */

-- less than two
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING FROM
employee.emp_record_table WHERE EMP_RATING <2;

-- greater than four
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING FROM
employee.emp_record_table WHERE EMP_RATING >4;

-- between two and four */
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING FROM
employee.emp_record_table WHERE EMP_RATING >=2 AND EMP_RATING <=4 ORDER BY
EMP_RATING;

-- 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 count(EMP_ID) as REPORTERS from emp_record_table
where MANAGER_ID is not null
GROUP BY MANAGER_ID
order by MANAGER_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 employee.emp_record_table  WHERE DEPT = 'FINANCE'
UNION
SELECT * FROM employee.emp_record_table  WHERE DEPT = 'HEALTHCARE'
ORDER BY DEPT, EMP_ID;

-- 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, MAX(EMP_RATING)
OVER (PARTITION BY DEPT) AS MAX_RATING
FROM employee.emp_record_table
ORDER BY  EMP_RATING DESC;

-- 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_SAL_OF_ROLE, MAX(SALARY) as
MAX_SAL_OF_ROLE FROM employee.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, concat(FIRST_NAME, ' ', LAST_NAME) as FULL_NAME, DEPT,EXP,
      rank() OVER(order by EXP DESC) as EMP_EXP_RANK
from employee.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 or REPLACE VIEW EMP_COUNTRY_VIEW AS
SELECT EMP_ID, FIRST_NAME, LAST_NAME,COUNTRY,SALARY
FROM employee.emp_record_table
WHERE SALARY>6000
order by COUNTRY, EMP_ID;
select * from EMP_COUNTRY_VIEW;

-- 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(
      select * from emp_record_table
    WHERE EXP > 10
    order by exp
) as EXP_GREATER_THAN_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 EMP_DETAILS()
BEGIN
      SELECT * FROM employee.emp_record_table WHERE EXP>3 order by EXP;
END //
DELIMITER //;

CALL EMP_DETAILS();

/* 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 organizationâ€™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 check_role(exp int)
RETURNS VARCHAR(40)
DETERMINISTIC
BEGIN
      DECLARE chck VARCHAR(40);
    IF EXP <= 2 THEN
                 SET chck = "JUNIOR DATA SCIENTIST";
      elseif exp > 2 AND exp <= 5 THEN
                 SET chck = "ASSOCIATE DATA SCIENTIST";
      elseif exp > 5 AND exp <= 10 THEN
                 SET chck = "SENIOR DATA SCIENTIST";
      elseif exp > 10 AND exp <= 12 THEN
                 SET chck = "LEAD DATA SCIENTIST";
      elseif exp > 12 AND exp <= 16 THEN
                 SET chck = "MANAGER";
      end if;
    RETURN(chck);
END //
delimiter ;

-- checking Data Science Team
select  EMP_ID, FIRST_NAME, LAST_NAME, ROLE, check_role(exp)
from data_science_team WHERE ROLE != check_role(exp);
```

```
-- 15.Create an index to improve the cost and performance of the query to
find the employee whose FIRST_NAME is â€˜Ericâ€™ in the employee table
after checking the execution plan.

explain select * from employee.emp_record_table where FIRST_NAME = "Eric";
create index F_index on employee.emp_record_table(FIRST_NAME(10));
show indexes from employee.emp_record_table;

-- 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, concat(FIRST_NAME," ",LAST_NAME) as NAME, EMP_RATING,
SALARY,(SALARY*0.05)*EMP_RATING 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, avg(SALARY) from emp_record_table
group by CONTINENT
order by CONTINENT;
```