```
# 1. Create Database
Create database employee;
use employee;

#Create structures of 3-tables and import data from CSV file
#Table no:1
Create table emp_record_table
(
EMP_ID varchar(10) PRIMARY KEY,
FIRST_NAME varchar(50) Not null,
LAST_NAME varchar(50) Not null,
GENDER varchar(1) Not null,
ROLE varchar(50) Not null,
DEPT varchar(50) Not null,
EXP INT(2) NOT NULL,
COUNTRY varchar(50) Not null,
CONTINENT varchar(50) Not null,
Salary Int(10) Not null,
EMP_RATING INT(1) NOT NULL,
MANAGER_ID VARCHAR(10),
PROJ_ID VARCHAR(10)
);
Select * from employee.emp_record_table;

#Table no:2
Create table data_science_team
(
EMP_ID varchar(10) Primary Key,
FIRST_NAME varchar(50) Not null,
LAST_NAME varchar(50) Not null,
GENDER varchar(1) Not null,
```

```sql
ROLE varchar(50) Not null,
DEPT varchar(50) Not null,
EXP INT(2) NOT NULL,
COUNTRY varchar(50) Not null,
CONTINENT varchar(50) Not null
);
Select * from employee.data_science_team;

#Table no:3
create table proj_table
(
PROJECT_ID VARCHAR(10) PRIMARY KEY,
PROJ_NAME VARCHAR(50) NOT NULL,
DOMAIN VARCHAR(50) NOT NULL,
START_DATE DATE NOT NULL,
CLOSURE_DATE DATE NOT NULL,
DEV_QTR VARCHAR(2) NOT NULL,
STATUS VARCHAR(10) NOT NULL
);
Select * from employee.proj_table;

# 2. Create ER Model

# 3. Fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, and DEPARTMENT from the employee record table
Select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT from employee.emp_record_table;

# 4. Fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING
Select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING  from employee.emp_record_table where EMP_RATING <2;
Select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING  from employee.emp_record_table where EMP_RATING >4;
Select EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING  from
employee.emp_record_table where EMP_RATING between 2 and 4;


# 5. Concatenate the FIRST_NAME and the LAST_NAME of employees in the Finance department

SELECT CONCAT(FIRST_NAME,' ', LAST_NAME) AS NAME FROM employee.emp_record_table WHERE
dept='Finance';


#6. List of employees who have someone reporting to them

#Select MANAGER_ID, COUNT(*) as number_of_Reporters from employee.emp_record_table group
by MANAGER_ID order by MANAGER_ID;

Select b.EMP_ID, b.FIRST_NAME, b.LAST_NAME, b.Role, COUNT(*) as Number_of_Reporters from
employee.emp_record_table b inner join employee.emp_record_table a on b.EMP_ID =
a.MANAGER_ID group by a.MANAGER_ID order by a.MANAGER_ID;


#7 List of all the employees from the healthcare and finance departments

 Select * from employee.emp_record_table where dept = 'HEALTHCARE'

 union

 Select * from employee.emp_record_table where dept = 'FINANCE';


 #8.List of employee details grouped by dept

 Select EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EMP_RATING from
employee.emp_record_table order by DEPT;

 #Select Dept, max(EMP_RATING) as Max_EMP_RATING from employee.emp_record_table group by
DEPT;

 #Select Dept, max(EMP_RATING) Over (partition by dept) as Max_EMP_RATING from
employee.emp_record_table;

 Select EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EMP_RATING, max(EMP_RATING) Over
(partition by dept) as Max_EMP_RATING from employee.emp_record_table order by DEPT;


 #9. Calculate the minimum and the maximum salary of the employees in each role

 Select ROLE, min(SALARY) as MIN_SALARY , max(SALARY) as MAX_SALARY from
employee.emp_record_table group by Role;


 #10. Assign ranks to each employee based on their experience
 Select *, rank() over (order by EXP) as EMP_RANK from employee.emp_record_table; # Rank in
Ascending order

 Select *, rank() over (order by EXP desc) as EMP_RANK from employee.emp_record_table; # Rank in
Descending order

 Select *, dense_rank() over (order by EXP) as EMP_RANK from employee.emp_record_table; #
Dense Rank in Ascending order

 Select *, dense_rank() over (order by EXP desc) as EMP_RANK from employee.emp_record_table; #
Dense Rank in Descending order
```

#11. Create a view that displays employees in various countries whose salary is more than six thousand

```
CREATE VIEW Employees AS

SELECT EMP_ID, FIRST_NAME, LAST_NAME, COUNTRY, SALARY FROM employee.emp_record_table
WHERE Salary > 6000;

Select * from employees;
```

#12. Find employees with experience of more than ten years

```
#select * from employee.emp_record_table where EXP > 10; #Simple Query

Select * from employee.emp_record_table where EMP_ID IN (select EMP_ID from
employee.emp_record_table where EXP > 10) ; #Nested Query
```

#13. Create a stored procedure to retrieve the details of the employees whose experience is more than three years

```
DELIMITER //

CREATE PROCEDURE Experience_Employee()

BEGIN

   select * from employee.emp_record_table where EXP > 3;

END //

DELIMITER ;

Call Experience_Employee();
```

#14. Create stored function to check whether the job profile assigned to each employee in the data science team matches the organization’s set standard

```
delimiter //
```

```
CREATE FUNCTION New_Role(Exp int)
RETURNS VARCHAR(50)
DETERMINISTIC
Begin
If Exp <= 2
Then RETURN 'JUNIOR DATA SCIENTIST';
ElseIf Exp between 3 and 5
Then RETURN 'ASSOCIATE DATA SCIENTIST';
ElseIf Exp between 6 and 10
Then RETURN 'SENIOR DATA SCIENTIST';
ElseIf Exp between 11 and 12
Then RETURN 'LEAD DATA SCIENTIST';
END If;
End //
Delimiter ;
```

```
Select EMP_ID, FIRST_NAME, LAST_NAME, ROLE, EXP, New_Role(Exp) as NEW_ROLE from employee.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 employee.emp_record_table (FIRST_NAME);
select * from employee.emp_record_table where FIRST_NAME = 'Eric';
```

#16. Calculate the bonus for all the employees, based on their ratings and salaries

```
Select EMP_ID, FIRST_NAME, LAST_NAME, EXP, SALARY, EMP_RATING, 0.05*SALARY*EMP_RATING as BONUS from employee.emp_record_table;
```

#17. Calculate the average salary distribution based on the continent and country

```
Select COUNTRY, Avg(SALARY) from employee.emp_record_table group by COUNTRY;
Select CONTINENT, Avg(SALARY) from employee.emp_record_table group by CONTINENT;
Select COUNTRY, CONTINENT, Avg(SALARY) from employee.emp_record_table group by COUNTRY,
CONTINENT;
```