**ScienceQtech Employee Performance Mapping.**

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.

STEP-1

CREATING DATABASE employee

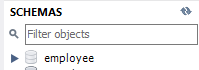
Solution-

Run query as-

CREATE DATABASE employee;



& refresh schema

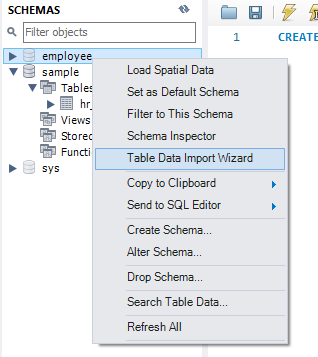


STEP-2

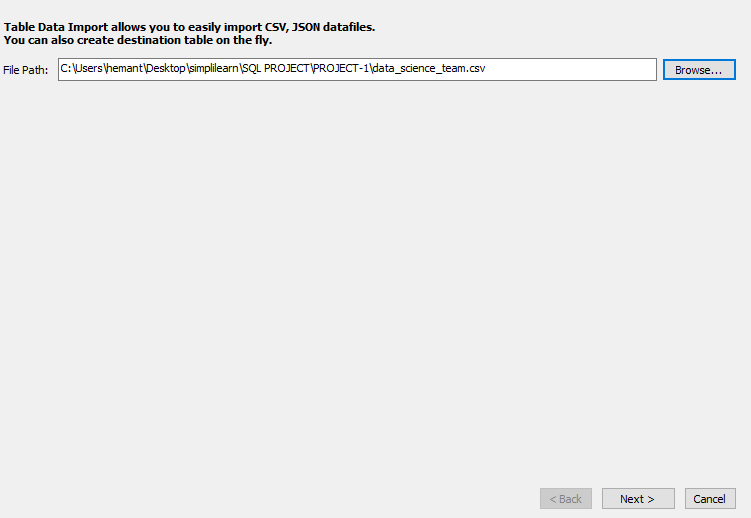
IMPORT CSV DATA IN TABLES IN employee database

Solution-

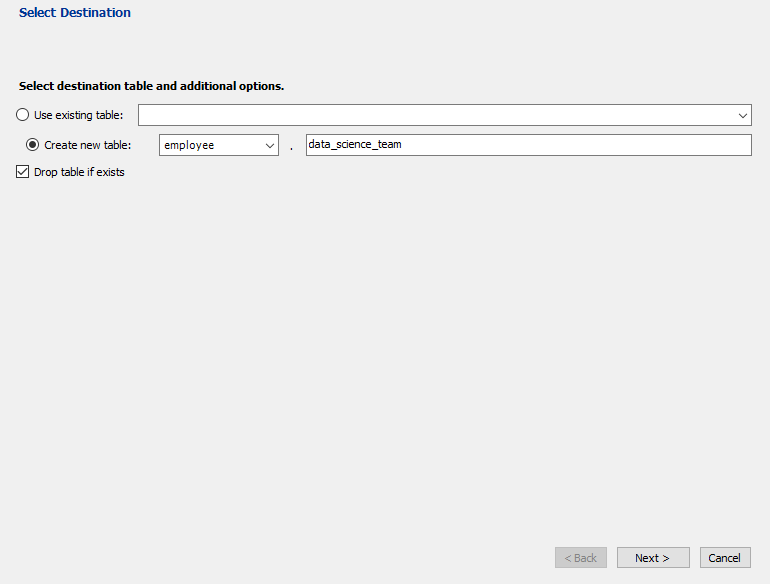
Right click on employee database & click on “Table Data Import Wizard”.



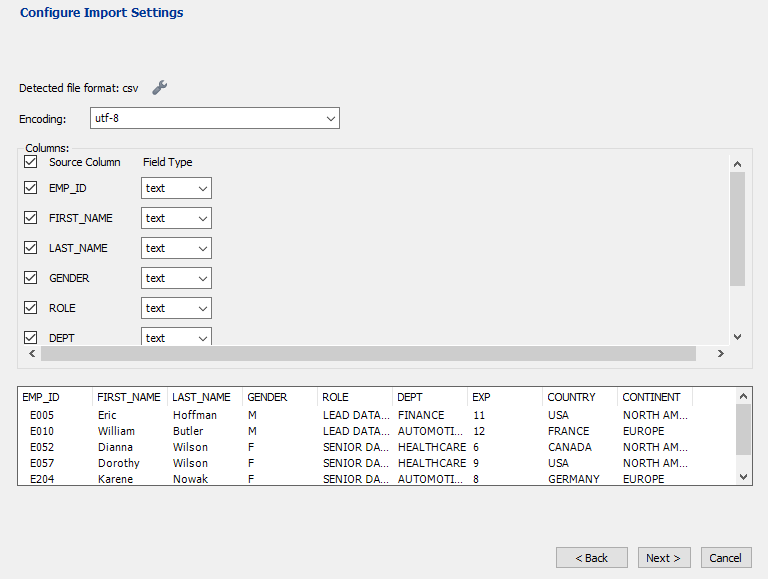
Browser csv file & click Next-



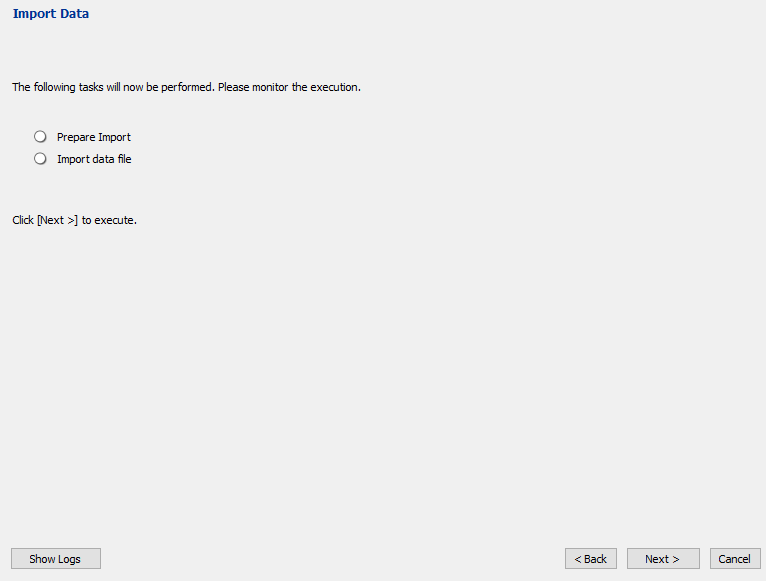
Creat new table-



Configure import data-

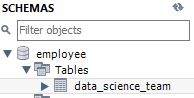


Import Data-

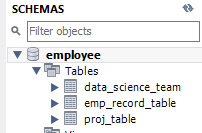


Then Finish…

Then refresh schema we will get table “data\_science\_team under employee database-



Similarly,other table are created and result as below-



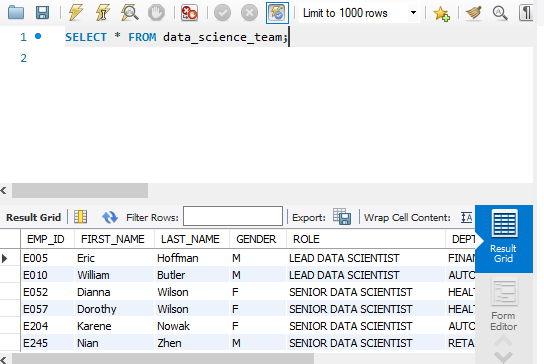
Then for selecting database execute query-

USE employee;



Then to show table execute query-

SELECT \* FROM data\_science\_team;

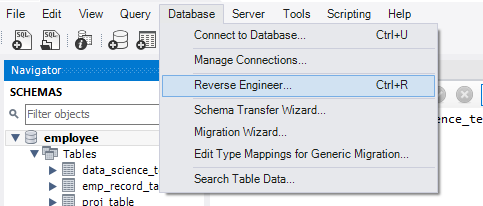


1. Create an ER diagram for the given **employee**database.

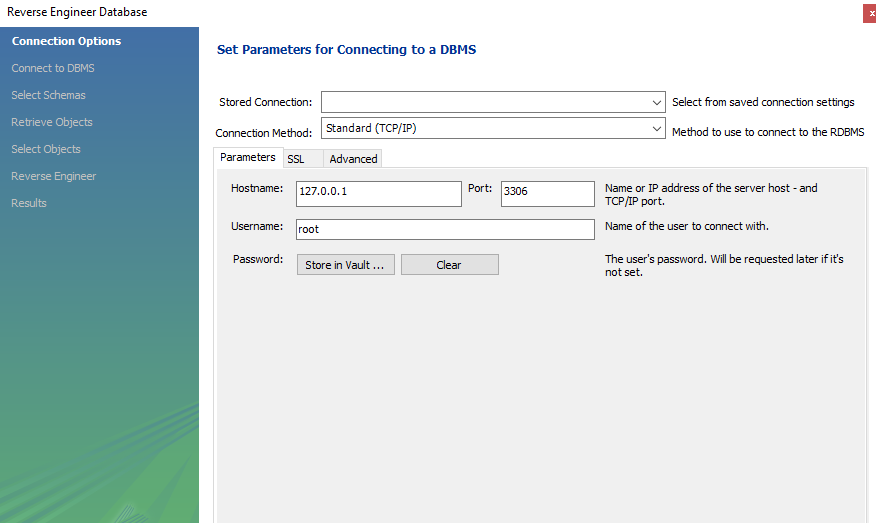
ER model is Entity-relationship model which compsosed of entity types & specific relationship that can exist between entities.

To creat ER model in mysql-

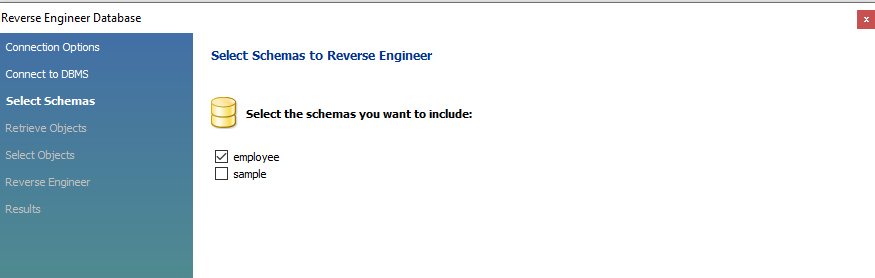
Go to database tb & click on reverse engineering-



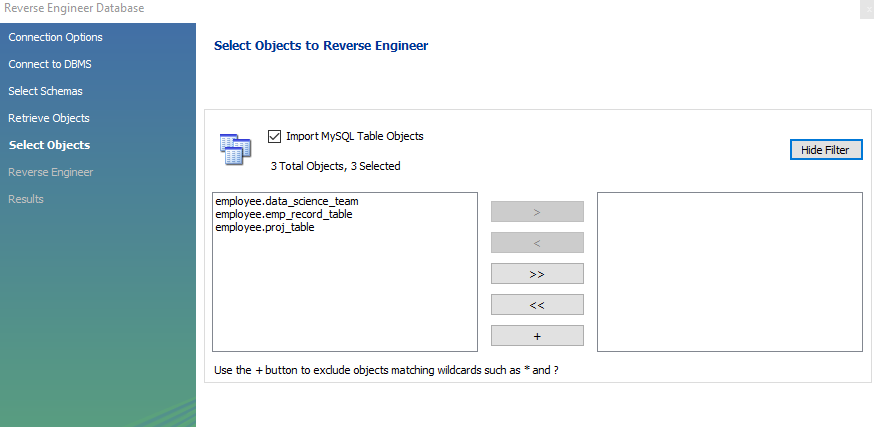
Set Parameters for connecting to a DBMS-



Select employee schema-

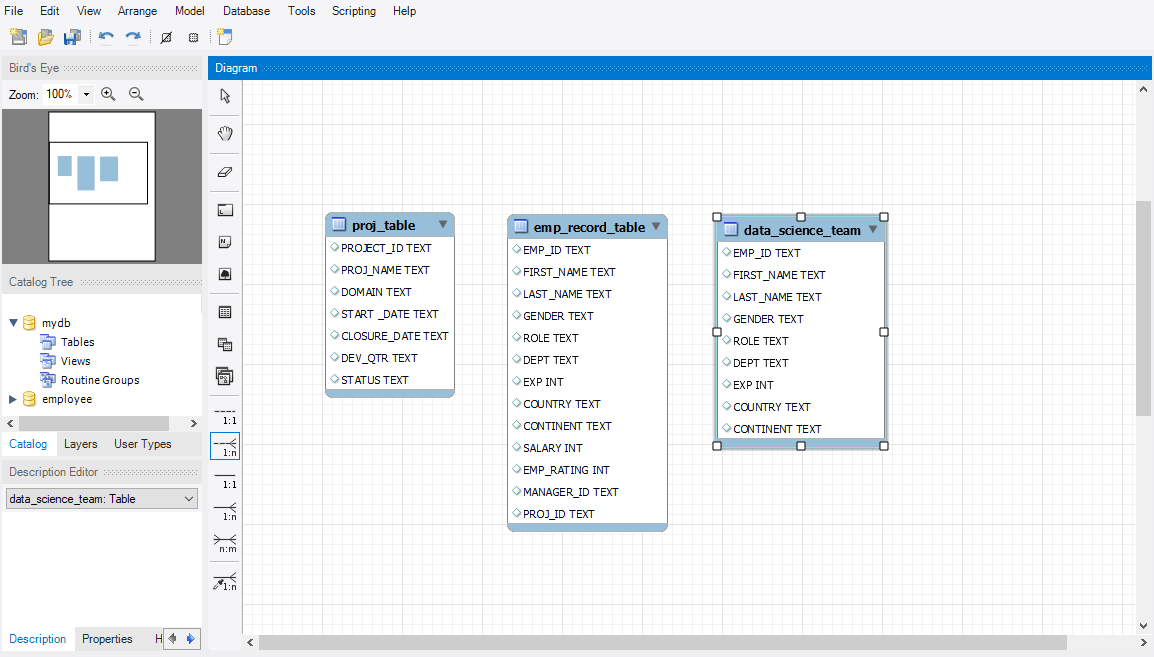


Select ovbject to reverse-



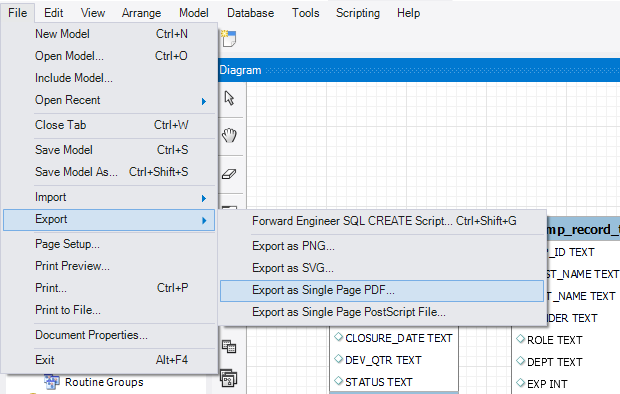
Then finish-

We will get ER diagram as below-



To export it-

Go to File tab>Export



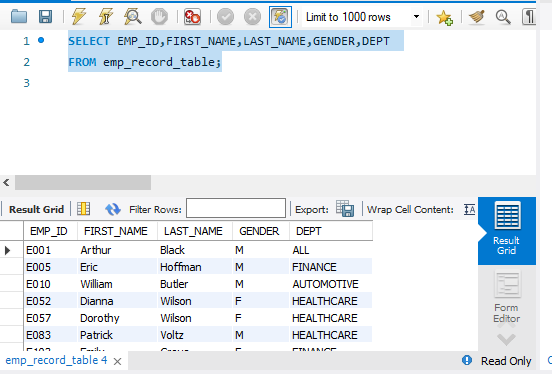
1. Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department.

Solution-

Run query as-

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

FROM emp\_record\_table;



1. 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

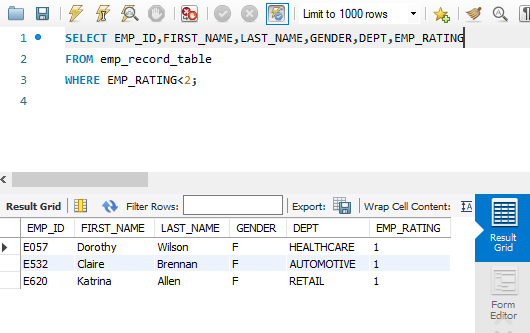
Solution-

For EMP\_RATING <2, run query-

SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,GENDER,DEPT,EMP\_RATING

FROM emp\_record\_table

WHERE EMP\_RATING<2;

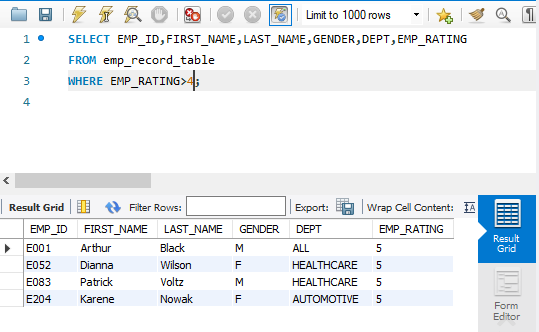


For EMP\_RATING >4, run query-

SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,GENDER,DEPT,EMP\_RATING

FROM emp\_record\_table

WHERE EMP\_RATING>4;

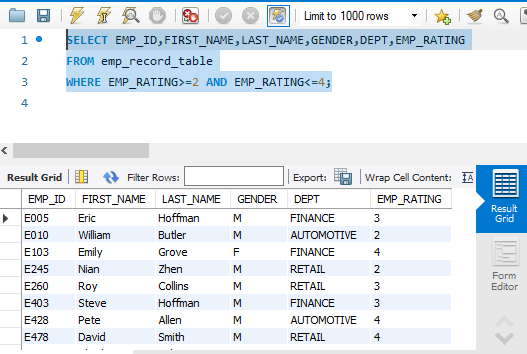


For EMP\_RATING between 2 & 4, run query-

SELECT EMP\_ID,FIRST\_NAME,LAST\_NAME,GENDER,DEPT,EMP\_RATING

FROM emp\_record\_table

WHERE EMP\_RATING>=2 AND EMP\_RATING<=4;



1. 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.

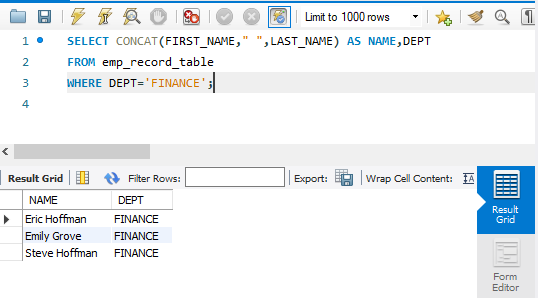
Solution-

Run query-

SELECT CONCAT(FIRST\_NAME," ",LAST\_NAME) AS NAME,DEPT

FROM emp\_record\_table

WHERE DEPT='FINANCE';



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

Solution-

Here we have to 1st assng name against each MANAGER\_ID,so we will link with EMP\_ID with self join along with find count of emplyee reporting to them so query is-

SELECT

m.EMP\_ID,m.FIRST\_NAME,m.LAST\_NAME,m.ROLE,

m.EXP,count(m.EMP\_ID) AS "EMP\_COUNT"

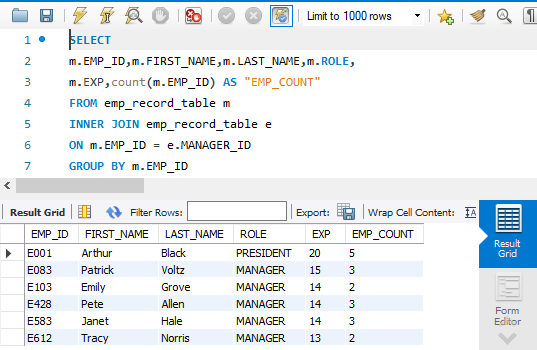
FROM emp\_record\_table m

INNER JOIN emp\_record\_table e

ON m.EMP\_ID = e.MANAGER\_ID

GROUP BY m.EMP\_ID

ORDER BY m.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.

Solution-

Run Query-

SELECT

m.EMP\_ID,m.FIRST\_NAME,m.LAST\_NAME,m.ROLE,

m.EXP,m.DEPT,m.EXP,m.COUNTRY,m.CONTINENT

FROM emp\_record\_table m

WHERE m.DEPT= 'HEALTHCARE'

UNION

SELECT

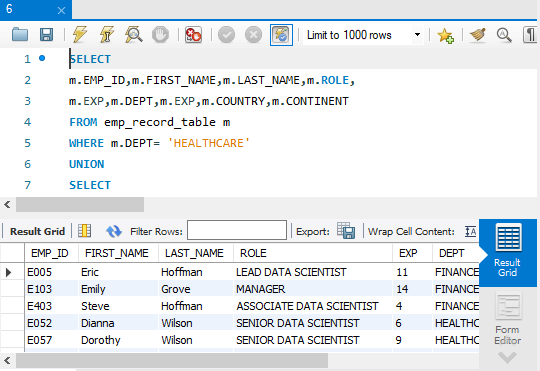
e.EMP\_ID,e.FIRST\_NAME,e.LAST\_NAME,e.ROLE,

e.EXP,e.DEPT,e.EXP,e.COUNTRY,e.CONTINENT

FROM emp\_record\_table e

WHERE e.DEPT= 'FINANCE'

ORDER BY DEPT,EMP\_ID;



1. 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.

Solution-

As here we have to show max rating department wise against each employee we have to use partition,so query is-

SELECT

m.EMP\_ID,m.FIRST\_NAME,m.LAST\_NAME,m.ROLE,m.DEPT,

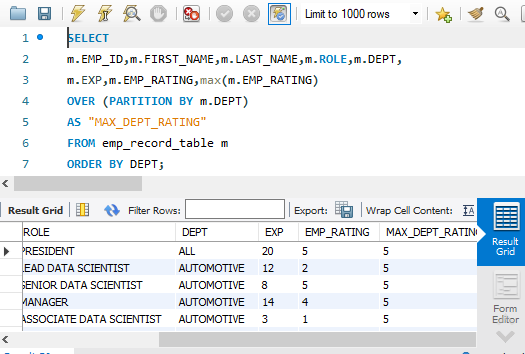
m.EXP,m.EMP\_RATING,max(m.EMP\_RATING)

OVER (PARTITION BY m.DEPT)

AS "MAX\_DEPT\_RATING"

FROM emp\_record\_table m

ORDER BY DEPT;



1. Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.

Solution-

Run query-

SELECT

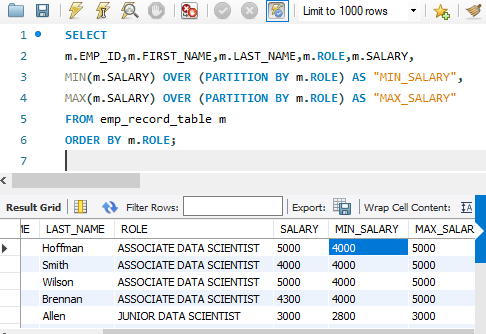
m.EMP\_ID,m.FIRST\_NAME,m.LAST\_NAME,m.ROLE,m.SALARY,

MIN(m.SALARY) OVER (PARTITION BY m.ROLE) AS "MIN\_SALARY",

MAX(m.SALARY) OVER (PARTITION BY m.ROLE) AS "MAX\_SALARY"

FROM emp\_record\_table m

ORDER BY m.ROLE;



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

Solution-

Run query-

SELECT

m.EMP\_ID,m.FIRST\_NAME,m.LAST\_NAME,m.DEPT,m.EMP\_RATING,

RANK() OVER (ORDER BY m.EMP\_RATING)

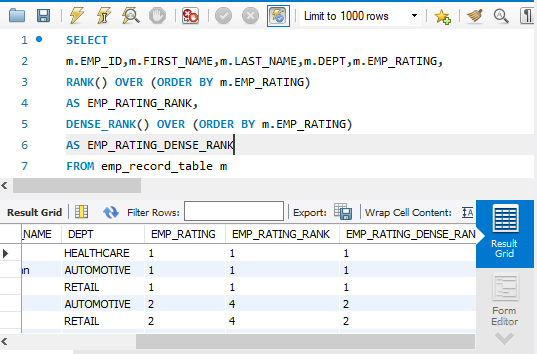
AS EMP\_RATING\_RANK,

DENSE\_RANK() OVER (ORDER BY m.EMP\_RATING)

AS EMP\_RATING\_DENSE\_RANK

FROM emp\_record\_table m

ORDER BY EMP\_RATING\_RANK;



1. 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.

Solution-

Run query to create view-

CREATE VIEW Employee\_View AS

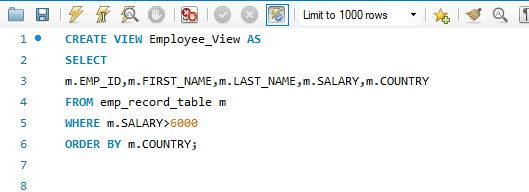
SELECT

m.EMP\_ID,m.FIRST\_NAME,m.LAST\_NAME,m.SALARY,m.COUNTRY

FROM emp\_record\_table m

WHERE m.SALARY>6000

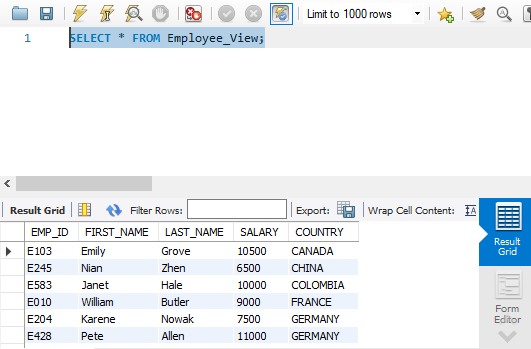
ORDER BY m.COUNTRY;



Then,

We can show created view with query-

SELECT \* FROM Employee\_View;



1. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

Solution-

As here we asked for nested query we have to generate sub query to calculate employee with more than 10 years of experience and match with parent query to get result, so query-

SELECT

m.EMP\_ID,m.FIRST\_NAME,m.LAST\_NAME,m.ROLE,m.EXP,m.DEPT

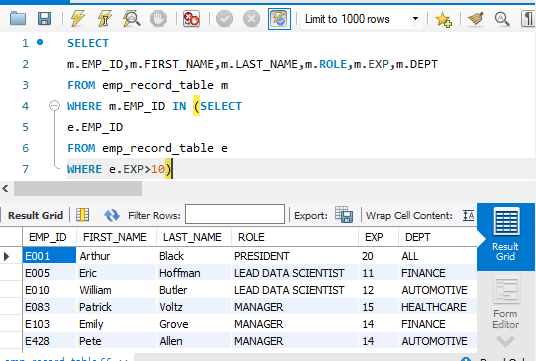
FROM emp\_record\_table m

WHERE m.EMP\_ID IN (SELECT

e.EMP\_ID

FROM emp\_record\_table e

WHERE e.EXP>10);



1. 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.

Solution-

Stored procedure is precompiled SQL command in database, so we generate precompiled command exp\_morethan\_3() and call it to get employee data with more than 3 years of experience, so query is-

DROP PROCEDURE IF EXISTS exp\_morethan\_3;

DELIMITER &&

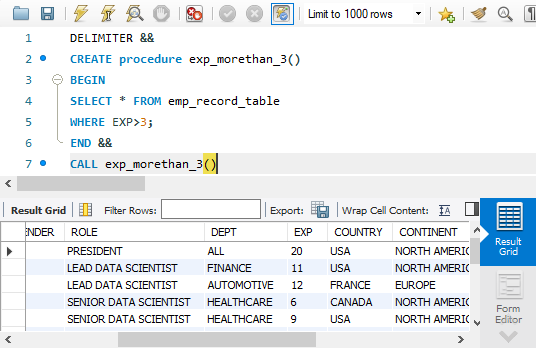
CREATE procedure exp\_morethan\_3()

BEGIN

SELECT \* FROM emp\_record\_table

WHERE EXP>3;

END &&

CALL exp\_morethan\_3()

1. 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'.

Solution-

For that we will use IF THEN ELSEIF ELSE stored function,so query-

DROP PROCEDURE IF EXISTS get\_req\_role;

DELIMITER $$

CREATE PROCEDURE get\_req\_role(

IN eid VARCHAR(4),

OUT req\_role VARCHAR(50))

BEGIN

DECLARE exper INT DEFAULT 1;

SELECT EXP INTO exper FROM emp\_record\_table

WHERE EMP\_ID= eid;

IF exper <=2 THEN

SET req\_role = 'JUNIOR DATA SCIENTIST';

ELSEIF exper <=5 THEN

SET req\_role = 'ASSOCIATE DATA SCIENTIST';

ELSEIF exper <=10 THEN

SET req\_role = 'SENIOR DATA SCIENTIST';

ELSEIF exper <=12 THEN

SET req\_role = 'LEAD DATA SCIENTIST';

ELSEIF exper <=16 THEN

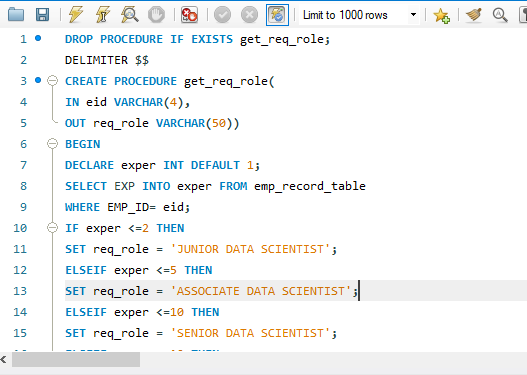
SET req\_role = 'MANAGER';

ELSE

SET req\_role = 'INVALID ROLE';

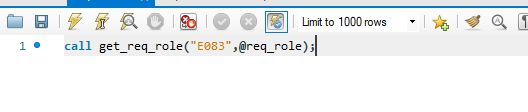
END IF;

END$$



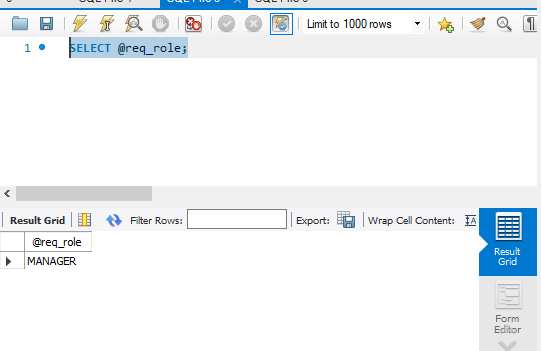
Then call function for getting required role for employee with EMP\_ID as “E083”,so query is-

call get\_req\_role("E083",@req\_role);



Then show req\_role, so query is-

SELECT @req\_role;



1. 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.

Solution-

Index is nothing but just like index page of book where we get page no. chapter wise so as we can shorten time of search.

So query is-

CREATE INDEX idx\_first\_name

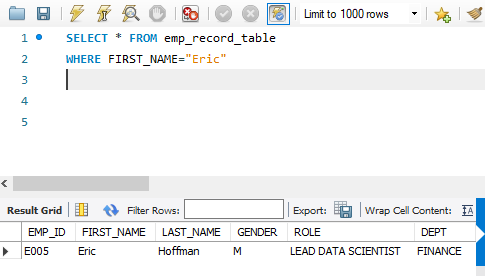
ON emp\_record\_table(FIRST\_NAME);



After that run query-

SELECT \* FROM emp\_record\_table

WHERE FIRST\_NAME="Eric";



1. 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).

Solution-

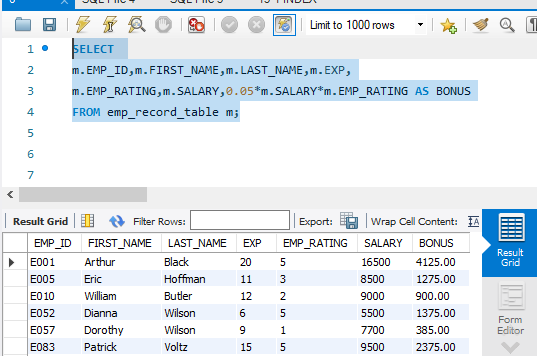
Run query-

SELECT

m.EMP\_ID,m.FIRST\_NAME,m.LAST\_NAME,m.EXP,

m.EMP\_RATING,m.SALARY,0.05\*m.SALARY\*m.EMP\_RATING AS BONUS

FROM emp\_record\_table m;



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

Solution-

Average Salary country wise,query is-

SELECT

m.EMP\_ID,m.FIRST\_NAME,m.LAST\_NAME,

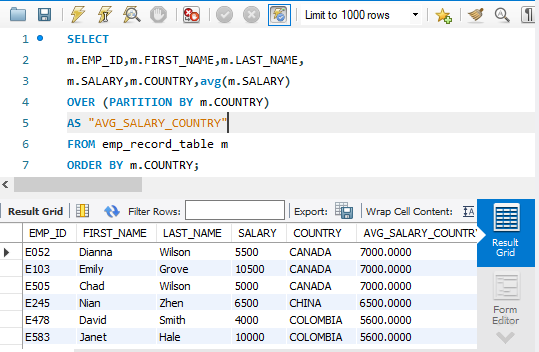
m.SALARY,m.COUNTRY,avg(m.SALARY)

OVER (PARTITION BY m.COUNTRY)

AS "AVG\_SALARY\_COUNTRY"

FROM emp\_record\_table m

ORDER BY m.COUNTRY;



Similarly,

Average salary continent wise,query is-

SELECT

m.EMP\_ID,m.FIRST\_NAME,m.LAST\_NAME,

m.SALARY,m.CONTINENT,avg(m.SALARY)

OVER (PARTITION BY m.CONTINENT)

AS "AVG\_SALARY\_CONTINENT"

FROM emp\_record\_table m

ORDER BY m.CONTINENT;

