**Science\_Q\_tech 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.

create database employee;

use employee;

alter table emp\_record\_table

modify column emp\_id varchar(20);

alter table emp\_record\_table

add primary key(emp\_id);

alter table proj\_table

modify column project\_id varchar(20);

alter table proj\_table

add primary key(project\_id);

select \* from emp\_record\_table ;

update emp\_record\_table

set proj\_id=null

where proj\_id='NA';

alter table emp\_record\_table

modify column proj\_id varchar(20);

alter table emp\_record\_table

add foreign key(proj\_id) references proj\_table(project\_id);

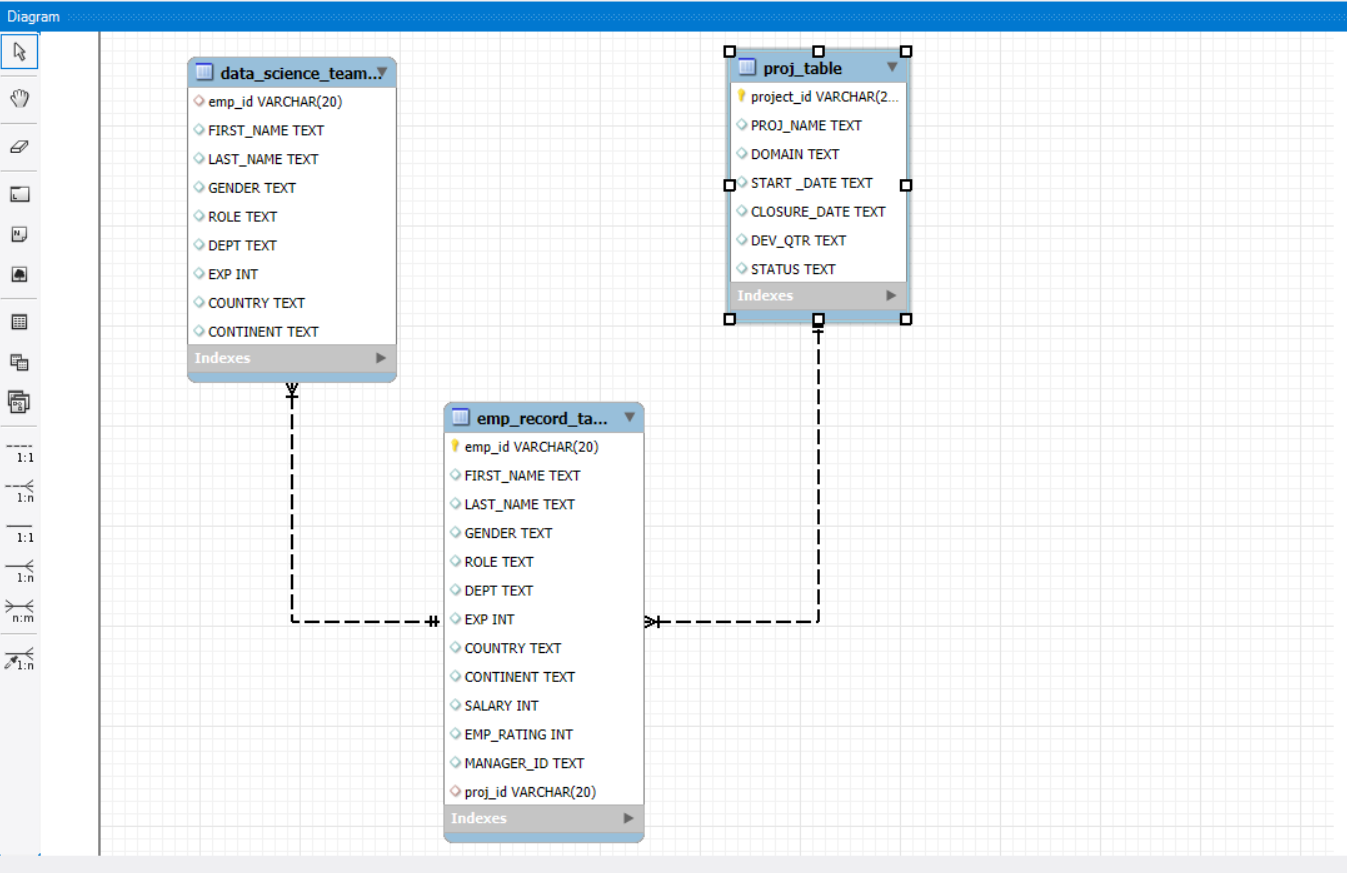
alter table data\_science\_team

modify column emp\_id varchar(20);

alter table data\_science\_team

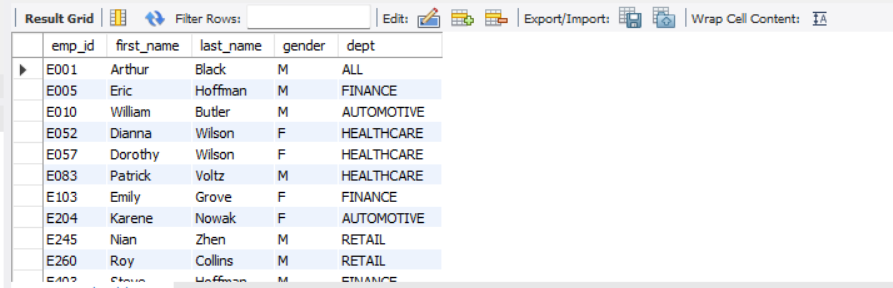
add foreign key(emp\_id) references emp\_record\_table(emp\_id);

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



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

SELECT emp\_id,first\_name,last\_name,gender,dept from emp\_record\_table;

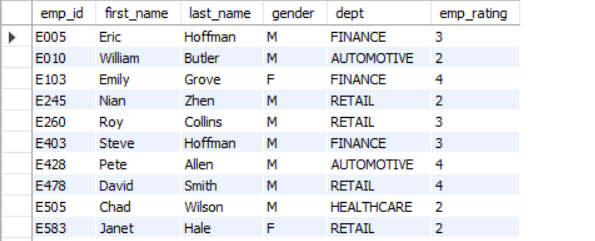


1. Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPARTMENT, and EMP\_RATING if the EMP\_RATING is:

* less than two
* greater than four
* between two and four

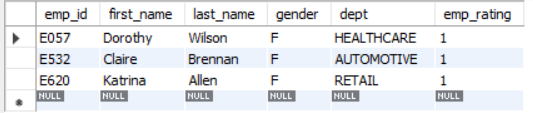
SELECT emp\_id,first\_name,last\_name,gender,dept,emp\_rating from emp\_record\_table

where emp\_rating between 2 and 4;



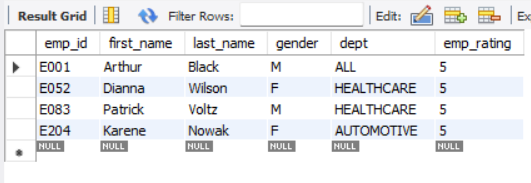
SELECT emp\_id,first\_name,last\_name,gender,dept,emp\_rating from emp\_record\_table

where emp\_rating <2;



SELECT emp\_id,first\_name,last\_name,gender,dept,emp\_rating from emp\_record\_table

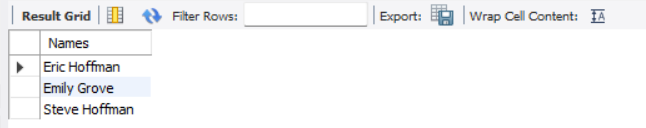
where emp\_rating >4;



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.

select concat(first\_name, ' ', last\_name) AS Names 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).

select concat(first\_name, ' ',last\_name) as ENAme, cnt from emp\_record\_table as t1

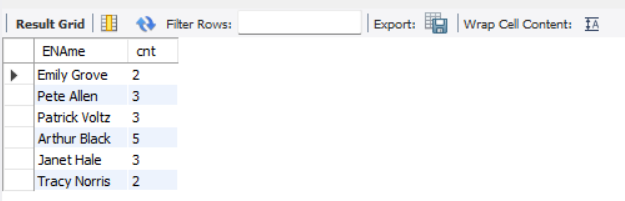
inner join

(select manager\_id, count(\*) as cnt from emp\_record\_table

group by manager\_id) as t2

on t1.emp\_id=t2.manager\_id

;

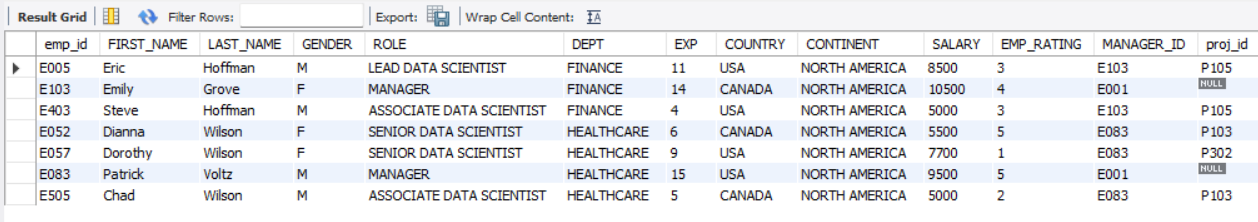


1. Write a query to list down all the employees from the healthcare and finance departments using union. Take data from the employee record table.

select \* from emp\_record\_table where dept='Finance'

union

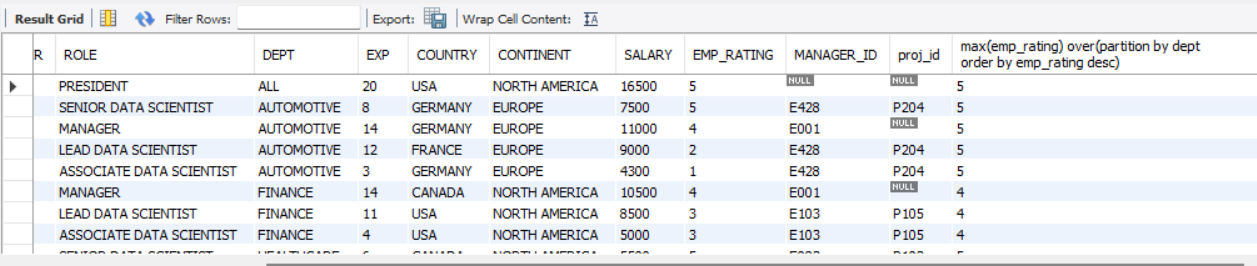
select \* from emp\_record\_table where dept='HealthCare';



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.

select \*, max(emp\_rating) over(partition by dept order by emp\_rating desc)

from emp\_record\_table;



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.

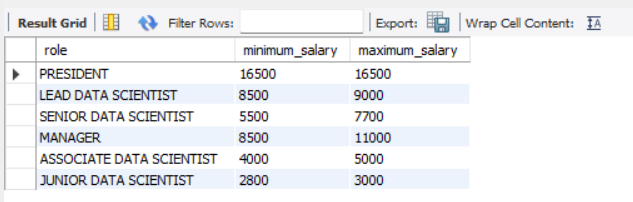
select role,

MIN(salary) as minimum\_salary,

MAX(salary) as maximum\_salary

from emp\_record\_table

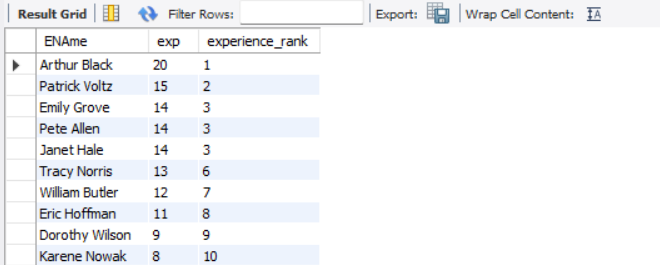
group by role;



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

select concat(first\_name, ' ',last\_name) as ENAme, exp, rank() over (order by exp desc) as experience\_rank

from emp\_record\_table;



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.

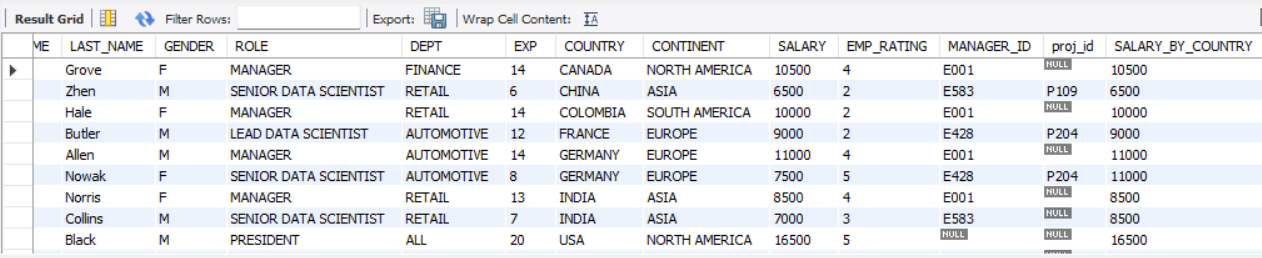
CREATE VIEW EMPLOYEES\_BY\_COUNTRY

AS

select salary, first\_name, country from emp\_record\_table

WHERE SALARY > 6000;

select \* from EMPLOYEES\_BY\_COUNTRY;

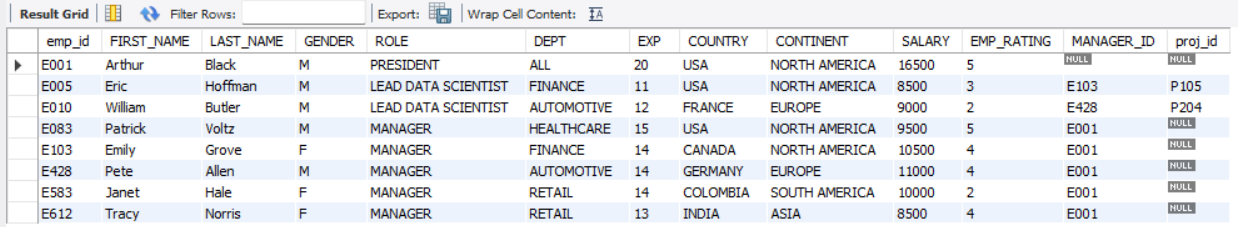


1. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

select \* from (

select \* from emp\_record\_table

where exp>10) as t;



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.

delimiter //

create procedure sp\_high\_exp\_emps()

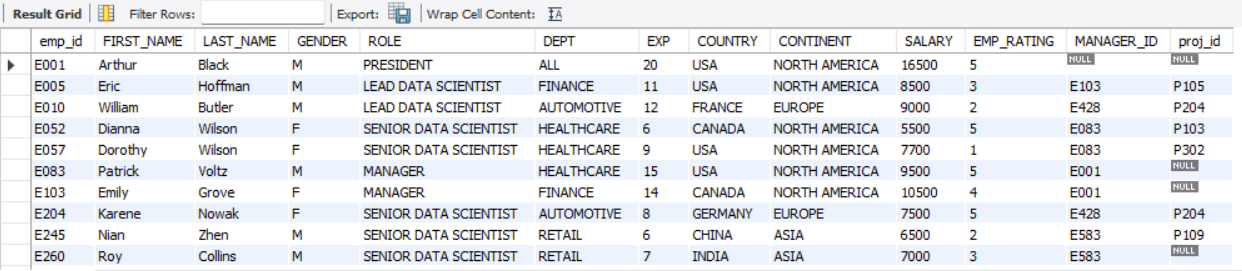
begin

select \* from emp\_record\_table where exp > 3;

end //

delimiter;

call sp\_high\_exp\_emps;



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.

select \* , case

when exp <=2 then 'JUNIOR DATA SCIENTIST'

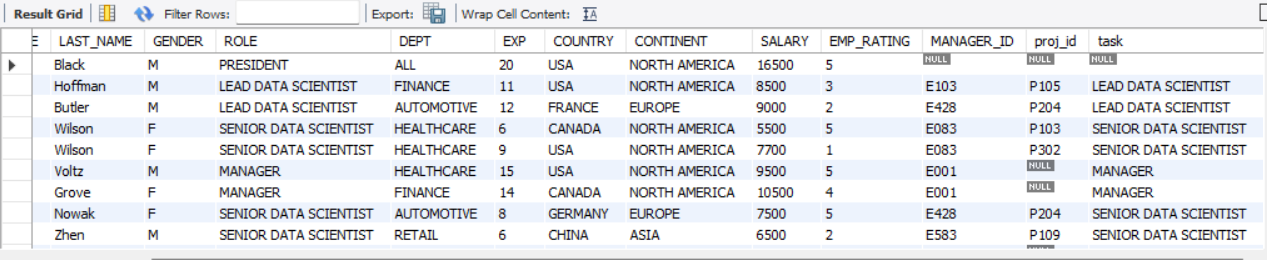
when exp <=5 then 'ASSOCIATE DATA SCIENTIST'

when exp <=10 then 'SENIOR DATA SCIENTIST'

when exp <=12 then 'LEAD DATA SCIENTIST'

when exp <=16 then 'MANAGER'

end as task from emp\_record\_table;



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.

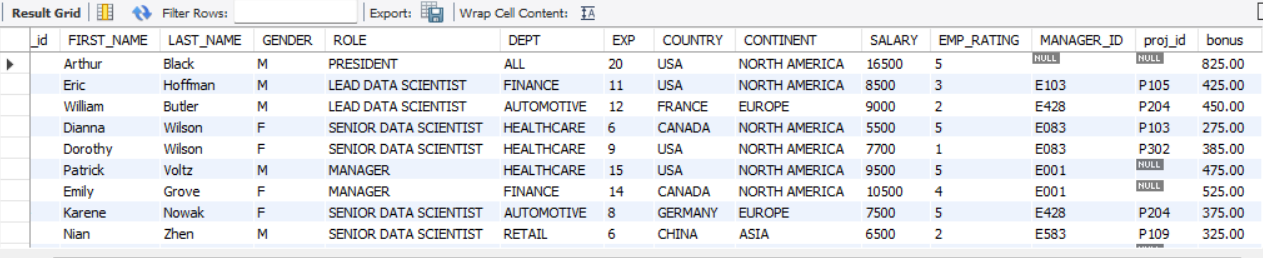
alter table emp\_record\_table

modify column FIRST\_NAME varchar(20);

create index ix\_fname on emp\_record\_table(first\_name);

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).

select \*, salary\*0.05 as bonus from emp\_record\_table;



1. 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 Avg from emp\_record\_table

group by continent ,country;

