COURSE END PROJECT -SIMPLILEARN

SQL Training

SUBMITTED BY SREEJITH C SRINIVAS

Code:

create database ds;

use ds;

create table e\_rec(

emp\_id varchar(4) not null,

First\_name varchar(45) not null,

last\_name varchar(45) not null,

Gender varchar(1) not null,

role\_ varchar(45) not null,

dept varchar(45) not null,

exp int not null,

Country varchar(45) not null,

continent varchar(45) not null,

salary int not null,

emp\_rating int not null,

Manager\_id varchar(10),

Proj\_id varchar(10),

Primary key(emp\_id));

create table pro(

project\_id varchar(25) not null,

project\_name varchar(45) not null,

domain varchar(45) not null,

start\_date date not null,

closure\_date date not null,

dev\_qtr varchar(4) not null,

P\_status varchar(45) not null,

Primary key(project\_id));

create table team(

emp\_id varchar(4) not null,

first\_name varchar(45) not null,

last\_name varchar(45) not null,

Gender varchar(1) not null,

role\_ varchar(45) not null,

dept varchar(45) not null,

exp int not null,

country varchar(45) not null,

continent varchar(45) not null,

primary key(emp\_id));

Select emp\_id, first\_name, last\_name, gender, dept from ds.e\_rec;

Select emp\_id, first\_name, last\_name, gender, dept, emp\_rating from ds.e\_rec

where emp\_rating<2;

Select emp\_id, first\_name, last\_name, gender, dept, emp\_rating from ds.e\_rec

where emp\_rating<4;

Select emp\_id, first\_name, last\_name, gender, dept, emp\_rating from ds.e\_rec

where emp\_rating between 2 AND 4;

Select concat(first\_name, ' ', last\_name) AS NAME from ds.e\_rec;

Select emp\_id, first\_name, last\_name, role\_, dept from ds.e\_rec

where role\_ not in ("Junior Data Scientist", "Associate Data Scientist");

Select count(\*) from ds.e\_rec;

Select first\_name, last\_name, dept from ds.e\_rec

where dept = "Healthcare"

union

Select first\_name, last\_name, dept from ds.e\_rec

where dept = "Finance";

SET sql\_mode = '';

Select emp\_id, first\_name, last\_name, role\_, dept, emp\_rating, max(emp\_rating) as max\_ratings

from ds.e\_rec group by dept;

Select min(salary) as min\_salary, max(salary) as max\_salary from ds.e\_rec;

Select row\_number() over (order by exp desc) as emp\_ranks, emp\_id, first\_name, last\_name, dept from ds.e\_rec;

create view emp\_sal as

select first\_name, last\_name, country from ds.e\_rec where salary>6000;

select first\_name, last\_name from ds.e\_rec

where not exists (select 1 from ds.e\_rec where exp<0) and exp>=10;

delimiter //

create procedure Get\_emp\_details()

begin

select \* from ds.e\_rec where exp>3;

end//

Call Get\_emp\_details()

delimiter //

create function emp\_role(exp int)

returns varchar(50)

deterministic

begin

declare job\_role varchar(50);

if exp <=2 then

set job\_role= 'Junior Data Scientist';

elseif exp<=5 then

set job\_role = 'Associate Data Scientist';

elseif exp<=10 then

set job\_role = 'Senior Data Scientist';

elseif exp<=12 then

set job\_role = 'Lead Data Scientist';

else

set job\_role = 'Manager';

end if;

return (job\_role);

end//

select exp, emp\_role(exp) from ds.team;

create index first\_name\_indx on e\_rec(first\_name(20));

Select \* from e\_rec where first\_name = 'Eric';

Select emp\_id, first\_name, last\_name, Gender, role\_, salary, (salary + 0.05\*salary\*emp\_rating)

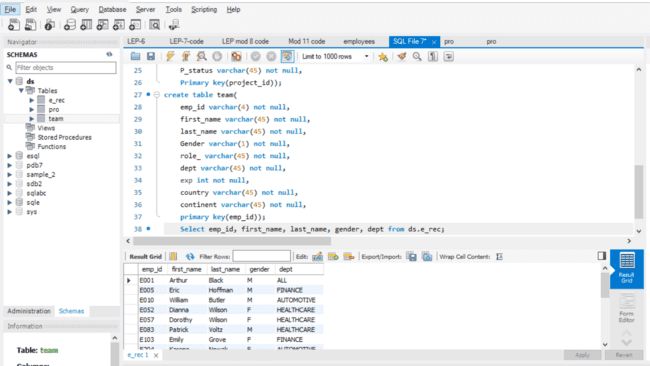
as updated\_salary from ds.e\_rec;

Select emp\_id, first\_name, last\_name, salary, country, continent, avg(salary) over(partition by country) as Country\_Avg\_Sal,

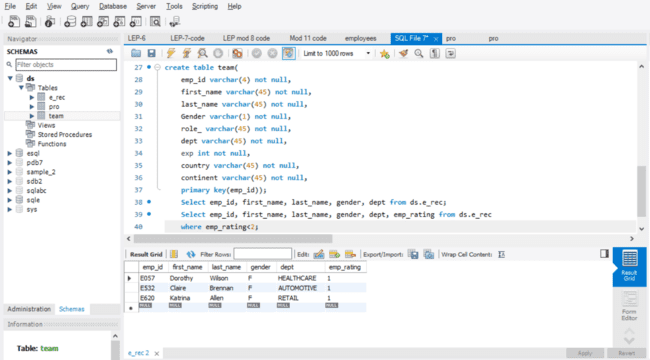
avg(salary) over(partition by continent) as Continent\_Avg\_Sal from ds.e\_rec;

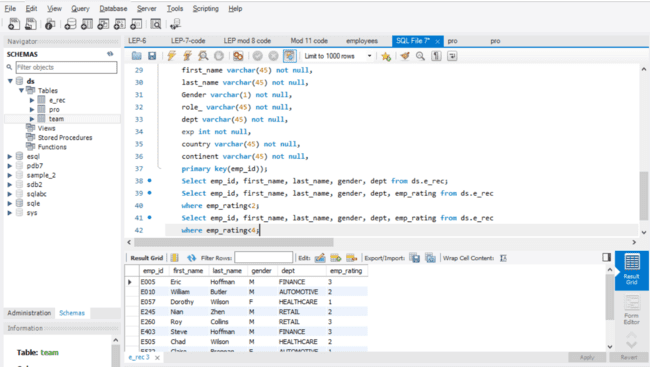
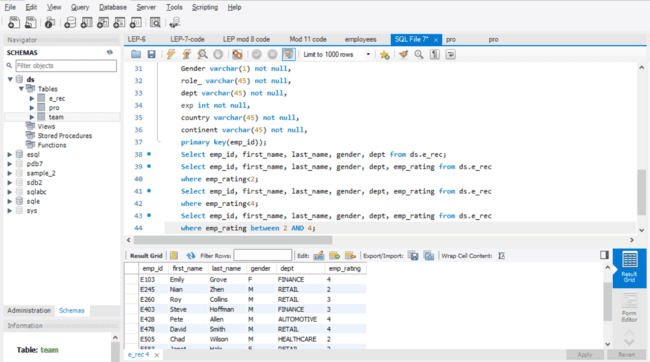
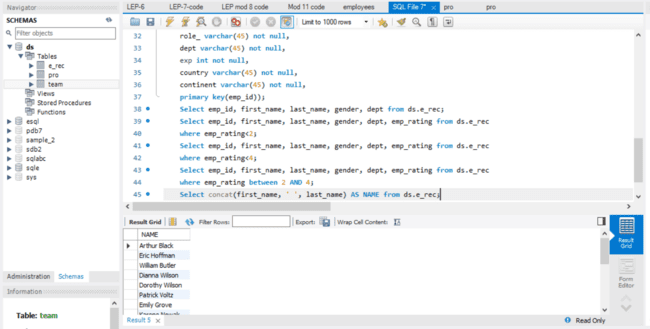
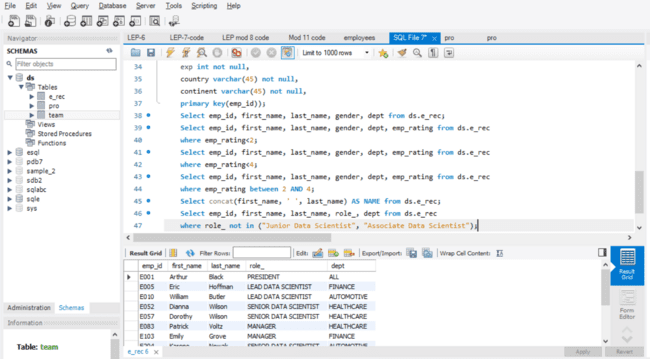
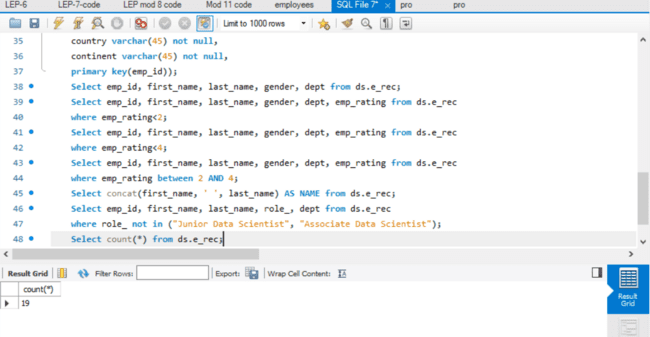
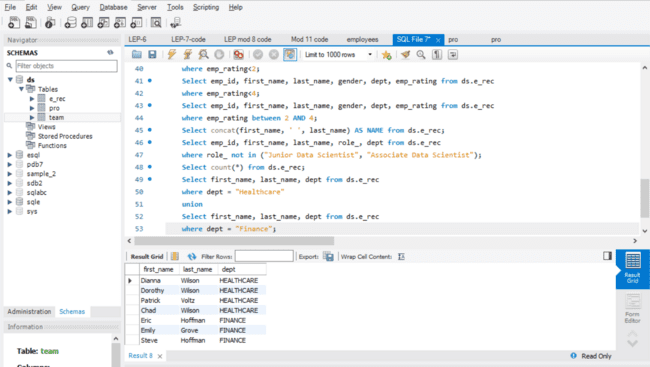
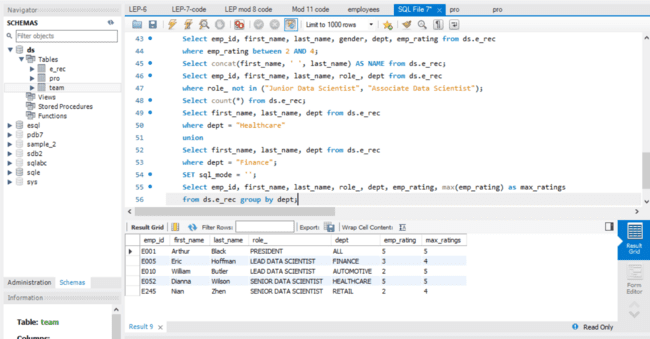
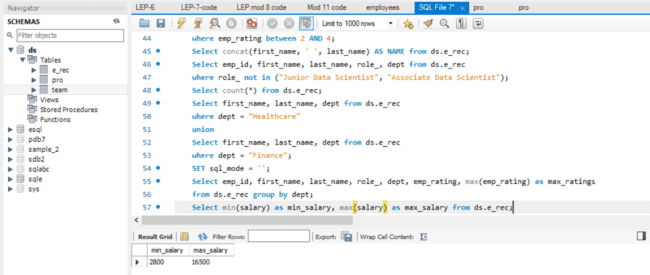
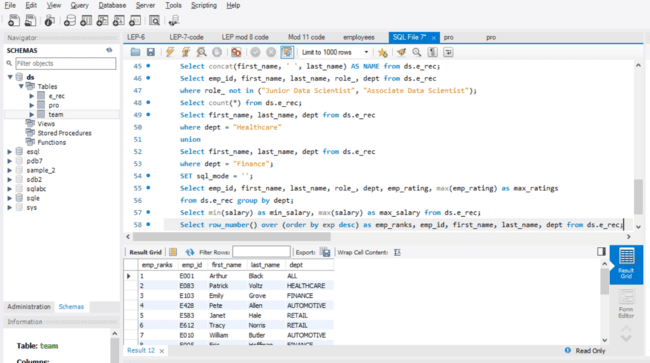
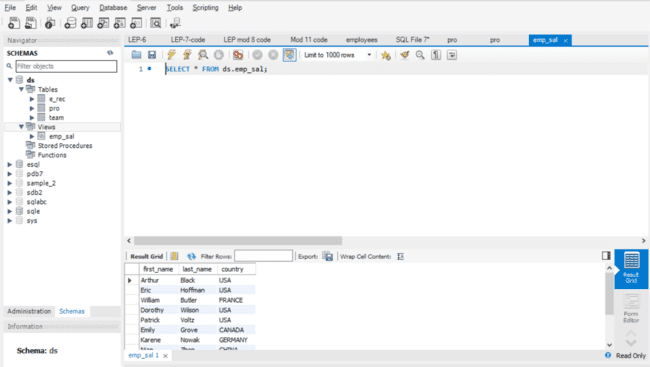
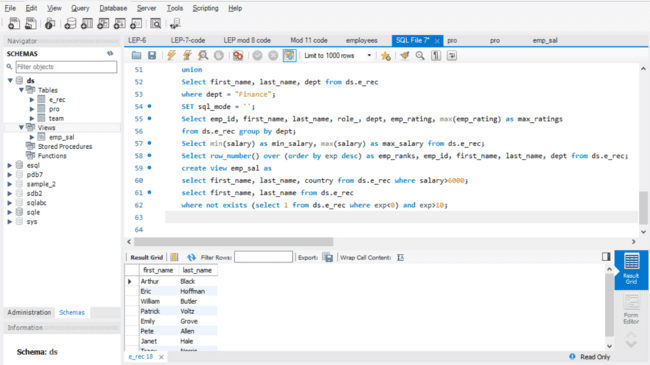
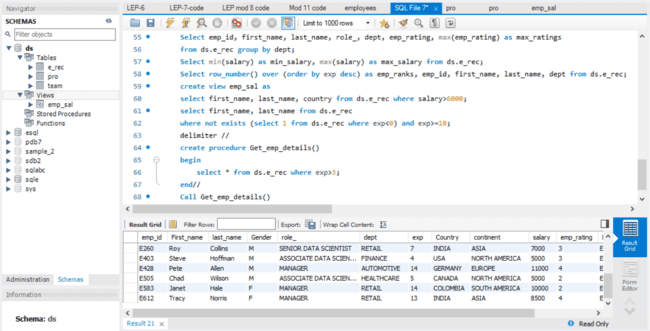
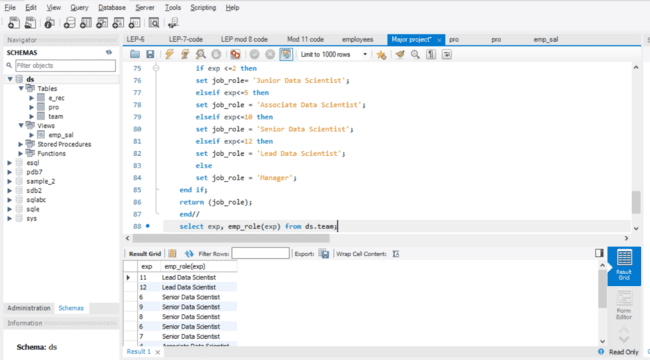
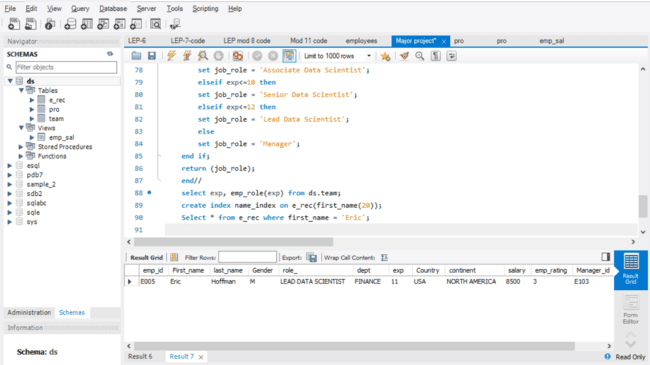
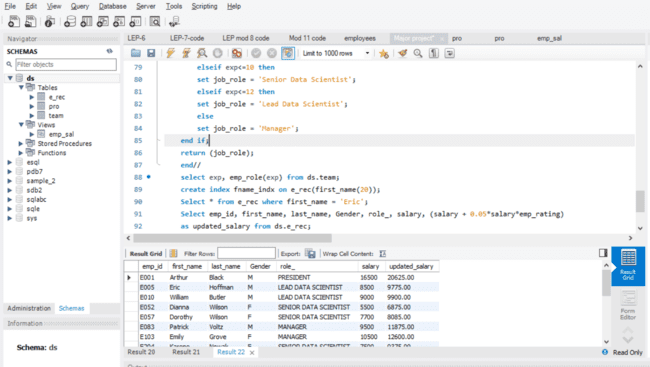
Outputs:

1. Retreving Data from table



1. Emp\_rating < 2:



1. Emp Rating < 4
2. Emp rating between 2 & 4
3. Concatenate names: 
4. Employees who have someone reporting to them: 
5. Count of reporters: 
6. Employees in healthcare and finance using UNION: 
7. Group employees by dept and determine max rating in each department: 
8. Calculate minimum and maximum salary: 
9. Ranking employees based on their experience: 
10. Employee with Salary>6000 from various countries using a view: 
11. Exp>10 years using nested query: 
12. Stored procedure for emp with exp>3: 
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.
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 organization’s set standard: 
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. 
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). 
17. Write a query to calculate the average salary distribution based on the continent and country. Take data from the employee record table. 