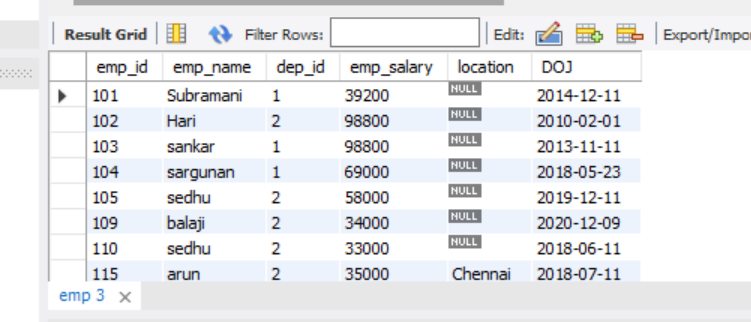
-- Basic Queries

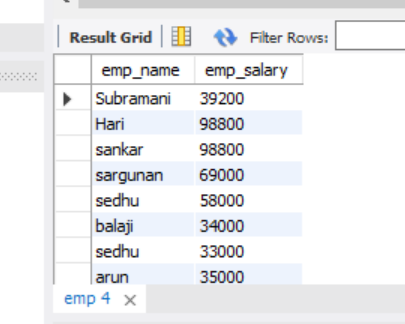
-- Write a query to display all rows and columns from the employees table.

select \* from emp;



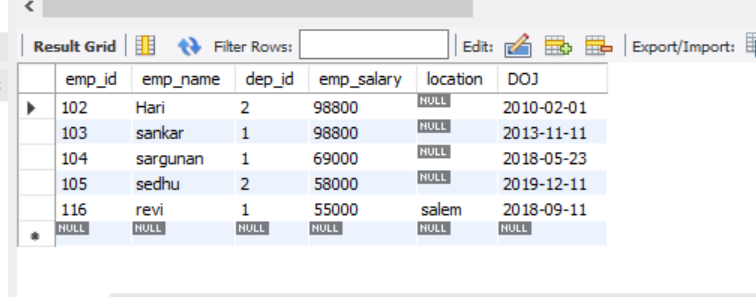
-- Retrieve only the name and salary of all employees from the employees table.

select emp\_name,emp\_salary from emp;



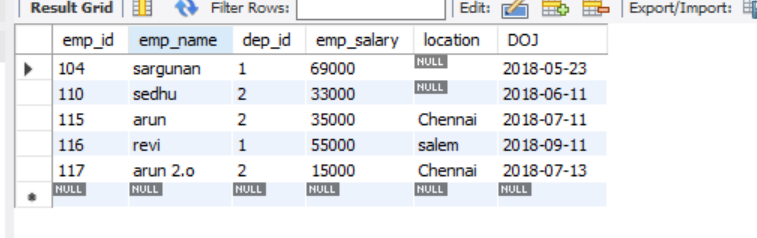
-- Write a query to find all employees whose salary is greater than 50,000.

select \* from emp where emp\_salary > 50000;



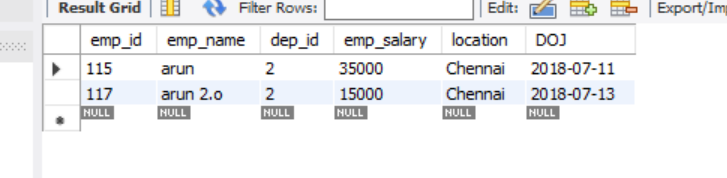
-- List all employees who joined the company in the year 2020.

select \* from emp where year(DOJ) = 2018;



-- Retrieve the details of employees whose names start with the letter 'A'.

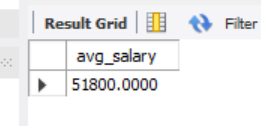
select \* from emp where emp\_name like 'a%';



-- Aggregate Functions

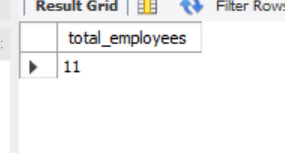
-- Write a query to calculate the average salary of all employees.

select avg(emp\_salary) as avg\_salary from emp;



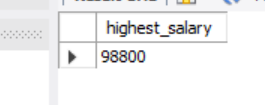
-- Find the total number of employees in the company.

select count(emp\_id) as total\_employees from emp;



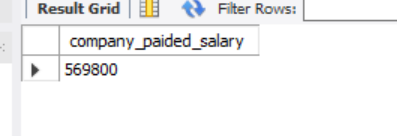
-- Write a query to find the highest salary in the employees table.

select emp\_salary as highest\_salary from emp order by emp\_salary desc limit 1;



-- Calculate the total salary paid by the company for all employees.

select sum(emp\_salary) as company\_paided\_salary from emp;



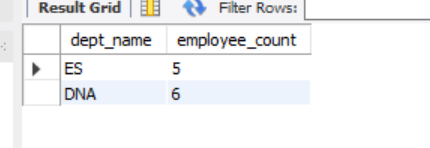
-- Find the count of employees in each department.

select department.dept\_name,count(emp.emp\_id) as employee\_count

from emp join department

on emp.dep\_id= department.department\_id

group by department.dept\_name;

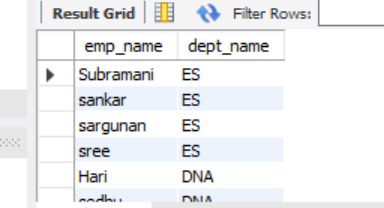


-- Joins

-- Write a query to retrieve employee names along with their department names (using employees and departments tables).

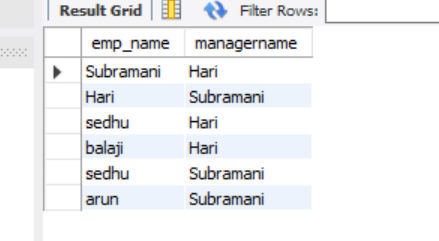
select emp.emp\_name,department.dept\_name from emp join department

on emp.dep\_id = department.department\_id;



-- List all employees who have a manager (self-join on employees table).

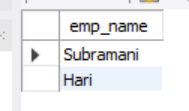
select e2.emp\_name,e1.emp\_name as managername from emp e1 join emp e2 on e1.emp\_id = e2.manager\_id;



-- Find the names of employees who are working on multiple projects (using employees and projects tables).

select emp.emp\_name from emp join emp\_projects on emp.emp\_id = emp\_projects.employee\_id

group by emp.emp\_name having count(emp.emp\_name) >1;

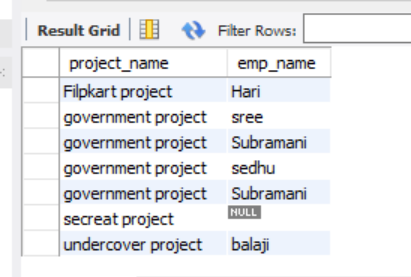


-- Write a query to display all projects and the employees assigned to them.

select project.project\_name,emp.emp\_name from project

left join emp\_projects on project.project\_id = emp\_projects.project\_id

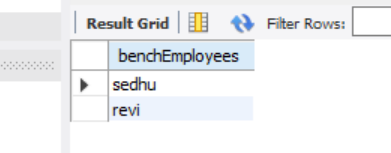
left join emp on emp.emp\_id = emp\_projects.employee\_id order by project.project\_name;



-- Retrieve the names of employees who do not belong to any department.

select emp\_name as benchEmployees from emp

where dep\_id is null;



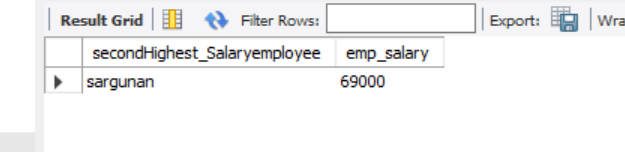
-- Subqueries

-- Write a query to find the employees with the second-highest salary.

select emp\_name as secondHighest\_Salaryemployee,emp\_salary from

( select emp\_name,emp\_salary,dense\_rank() over(order by emp\_salary desc)as salary\_rank from emp) new\_ranktable

where salary\_rank = 2;



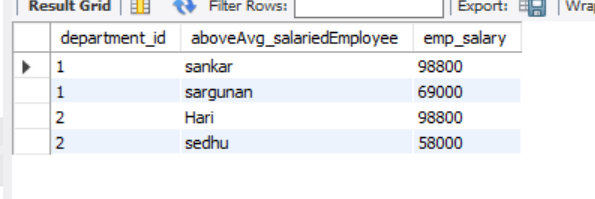
-- Retrieve the names of employees whose salary is above the department average salary.

select department.dept\_name,avg(emp.emp\_salary) from emp join department on emp.dep\_id = department.department\_id group by department.dept\_name;

select department.department\_id,emp.emp\_name as aboveAvg\_salariedEmployee,emp.emp\_salary from emp join department

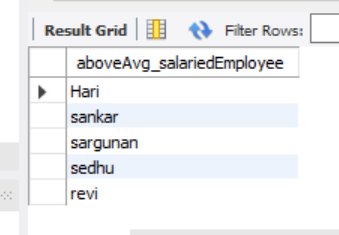
on emp.dep\_id = department.department\_id

where emp.emp\_salary > (select avg(e2.emp\_salary) from emp e2 where e2.dep\_id = emp.dep\_id );



-- Find employees who earn more than the average salary of the entire company.

select emp\_name as aboveAvg\_salariedEmployee from emp where emp\_salary > (select avg(emp\_salary) from emp);



-- Write a query to find the department with the highest number of employees.

select d1.dept\_name from department d1

left join emp e1 on d1.department\_id = e1.dep\_id

group by d1.dept\_name

having count(e1.emp\_id) =

(select max(employee\_count) from

(select count(e2.emp\_id) as employee\_count

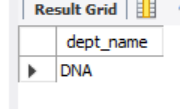
from emp e2

group by e2.dep\_id

)

count\_subquery

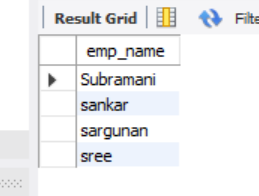
);



-- List all employees who work in a department located in 'New York'.

select emp.emp\_name from emp join department on emp.dep\_id = department.department\_id

where department.dept\_location = 'New york';



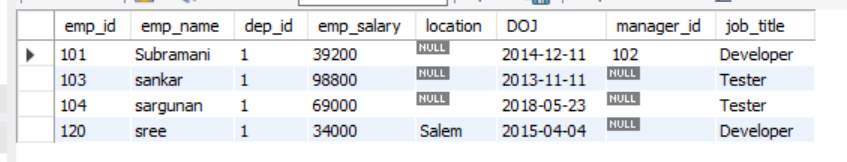
-- Set Operators

-- Write a query to find employees who work in either the 'HR' or 'Finance' department.

select e.\* from emp e join department d

on e.dep\_id = d.department\_id

where d.dept\_name = 'ES' or d.dept\_name = 'Finance';



-- Retrieve the names of employees who are working on both Project A and Project B.

select e.emp\_name from emp e

join emp\_projects e1 on e.emp\_id = e1.employee\_id

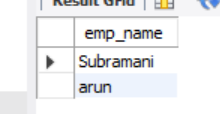
join project p1 on p1.project\_id = e1.project\_id

join emp\_projects e2 on e.emp\_id = e2.employee\_id

join project p2 on p2.project\_id = e2.project\_id

where p1.project\_name = 'Amazon project' and p2.project\_name = 'government project'

group by e.emp\_name;



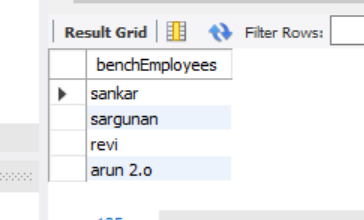
-- Find employees who are not assigned to any project

select emp\_name as benchEmployees from emp where emp\_name not in(

select e.emp\_name from emp e

join emp\_projects e1 on e.emp\_id = e1.employee\_id

join project p1 on p1.project\_id = e1.project\_id );



-- Write a query to get all unique job titles across all departments.

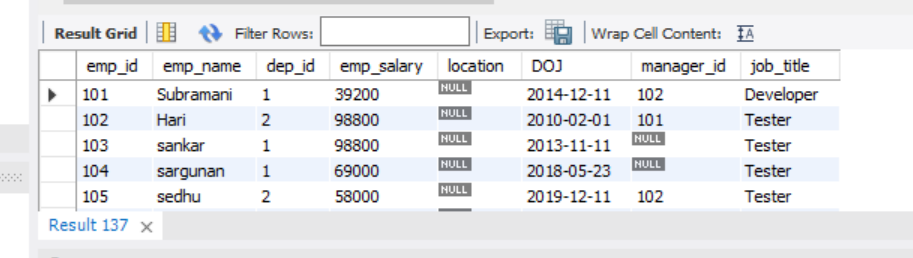
select distinct emp.job\_title from emp join department

on emp.dep\_id = department.department\_id;



-- Combine two tables (employees and former\_employees) and remove duplicates.

select \* from emp union select \* from formeremployee3



-- DML and DDL

-- Write a query to add a new employee to the employees table.

insert into emp values(121,'balaji',3,98000,'covai','2019-03-01','104','Developer');

-- Update the salary of all employees in the 'ES' department by 10%.

update emp join department on emp.dep\_id = department.department\_id

set emp.emp\_salary = emp.emp\_salary\*1.10

where department.dept\_name = 'ES';

-- Delete all employees who have not worked for more than 5 years.

delete from emp where DOJ > date\_sub( curdate(), interval 5 year );

-- Create a new table departments\_backup with the same structure as the departments table.

create table departments\_backup(

department\_id numeric primary key,

dept\_name varchar(50),

emp\_count numeric

);

-- Drop the temporary\_data table from the database

drop table departments\_backup;

-- Constraints

-- Add a primary key to the employees table.

-- Already added

-- Write a query to create a foreign key between employees and departments tables.

-- Already added

-- Add a unique constraint to the email column in the employees table.

alter table emp add column email varchar(50) unique;

-- Write a query to check all constraints applied on the employees table.

-- Remove the NOT NULL constraint from the phone\_number column in the employees table.

alter table emp add column phone\_number numeric(10) not null;

alter table emp modify phone\_number numeric(10) ;

PL/SQL Questions

Basic PL/SQL Programs

Write a PL/SQL block to calculate the factorial of a given number.

DELIMITER $$

create procedure CalculateFactorial(in n int)

begin

declare a INT DEFAULT 1;

WHILE n > 0 DO

SET a = n \* a;

SET n = n - 1;

END WHILE;

SELECT a;

END $$

DELIMITER ;

call CalculateFactorial(5)

-- Write a PL/SQL block to display the Fibonacci series up to n terms.

delimiter //

create procedure get\_fibonacci2(in input int)

begin

declare a int default 0;

declare b int default 1;

declare temp int;

declare i int default 1;

declare f varchar(255) default '';

set f =concat(f,a);

set f = concat(f,', ',b);

while i < input - 1 do

set temp = a+b;

set a=b;

set b = temp;

set f = concat(f,', ',b);

set i = i+1;

end while;

select f as fibinocci\_series;

end //

call get\_fibonacci2(10)

Create a PL/SQL block to reverse a given string.

Write a program to find whether a number is prime or not.

Write a program to calculate the sum of all digits of a given number.