**PgSQL**

**DDL,DML,DCL,TCL,DQL**

1)\l

Displays the list of databases.

2)\dt

Displays the list of tables

3) create database data\_science;

4)\c data\_science;

This connects to the database;

5)Creating table

Create table students(

stud\_id int primary key,

name varchar(30),

dept varchar(40),

gender char,

age int);

6)Inserting values

insert into students values(1,'Anu','AI&DS','F',20);

//don’t use “ ” for adding varchar values.Use only ‘ ‘ single quotes.

**7)ALTER**

**Adding column:**

alter table students add column cgpa float;

**Changing table name:**

alter table students rename to stud\_details;

**Changing column name:**

alter table stud\_details

rename dept to department;

**8)To view table description**

/d table\_name;

**9)Update**

update stud\_details

set cgpa=9.7 where stud\_id=1;

**10)Delete**

insert into stud\_details values(4,'Danu','EEE','F',20,8.7);delete from stud\_details where stud\_id=4;

**11)Truncate**

Used to delete only the data in the table.

->truncate table stud\_details;

12)Delete columns

alter table stud\_details

drop column cgpa;

**13)Order by**

select\*from stud\_details order by age desc;

**14)Limit**

select\*from stud\_details limit 2;

**15)LIKE**

select \*from stud\_details where name like '%a'; //names ending with a.

select \*from stud\_details where name like '%l%';

**16)MIN AND MAX LENGTH NAME**

select name,length(name) from stud\_details order by length(name),name limit 1;

op: Min length name

|  |  |
| --- | --- |
| Name | Length(name) |
| Anu | 3 |

Max length name

select name,length(name) from stud\_details order by length(name)desc ,name limit 1;

**JOINS**

**1)INNER JOIN**

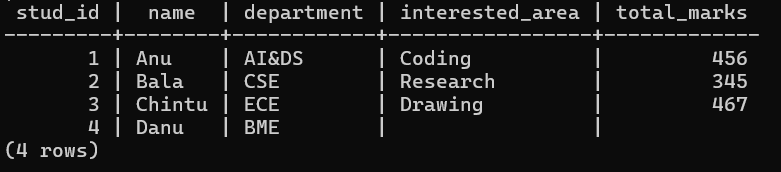
Displays only the matching column.

Select stud\_details.stud\_id,stud\_details.name,stud\_details.department,stud\_marks.interested\_area,stud\_marks.total\_marks from stud\_details Inner join stud\_marks on stud\_details.stud\_id=stud\_marks.stud\_id;

**2)LEFT JOIN**

Displays the matching columns and also the non matching columns in the left table.

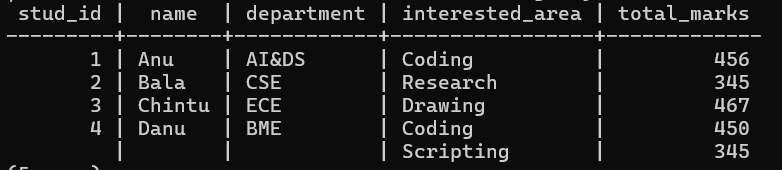
select stud\_details.stud\_id,stud\_details.name,stud\_details.department,stud\_marks.interested\_area,stud\_marks.total\_marks from stud\_details LEFT join stud\_marks on stud\_details.stud\_id=stud\_marks.stud\_id;



**3)RIGHT JOIN**

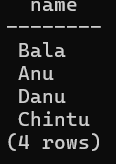
Displays the matching columns and also the non matching columns in the right table.

select stud\_details.stud\_id,stud\_details.name,stud\_details.department,stud\_marks.interested\_area,stud\_marks.total\_marks from stud\_details right join stud\_marks on stud\_details.stud\_id=stud\_marks.stud\_id;



->Query the Name of any student in **stud\_details** .Order your output by the last three characters of each name. If two or more students both have names ending in the same last three characters (i.e.: Bobby, Robby, etc.), secondary sort them by ascending ID.

select name from stud\_details order by right(name,2),stud\_id asc;



**WINDOW FUNCTIONS**

**RANGE,DENSE\_RANGE,ROW\_NUMBER**

select emp\_id,name,salary,dept,

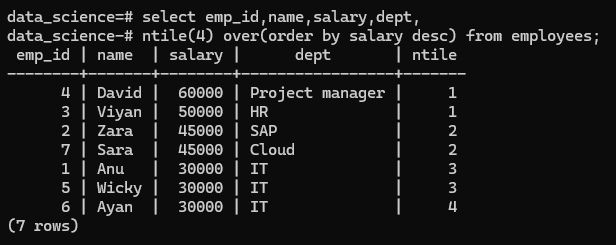
rank() over(order by salary desc) as rank,

dense\_rank() over(order by salary desc) as dense\_rank,

row\_number() over(order by salary desc) as row from employees;



**NTILE**

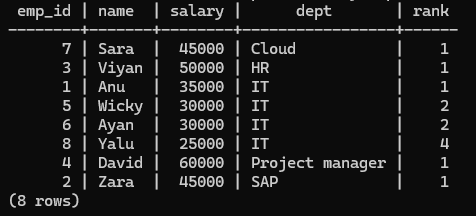
****

**PARTITION**

**\*Rank() with partition**

select emp\_id,name,salary,dept,

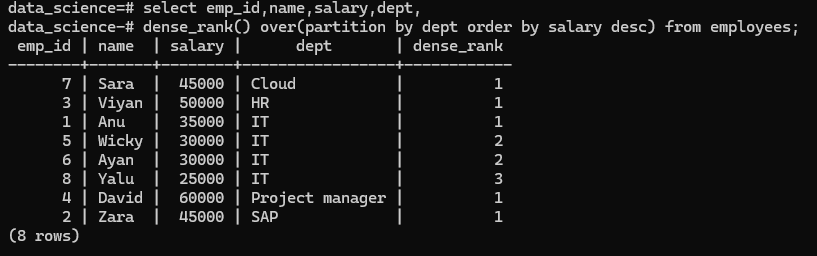
rank() over(partition by dept order by salary desc) from employees;



**\*Dense\_rank() with partition**

select emp\_id,name,salary,dept,

dense\_rank() over(partition by dept order by salary desc) from employees;



**SWAP IN LEETCODE(627)**

update salary

set sex =

case

when sex = 'm' then 'f'

when sex = 'f' then 'm'

end;

[**3475. DNA Pattern Recognition**](https://leetcode.com/problems/dna-pattern-recognition/)

Select sample\_id,dna\_sequence,species,

case

    when dna\_sequence like 'ATG%' then 1

    else 0

end as has\_start,

case

    when dna\_sequence like '%TAA' then 1

    when dna\_sequence like '%TAG' then 1

    when dna\_sequence like '%TGA' then 1

    else 0

end as has\_stop,

case

    when dna\_sequence like '%ATAT%' then 1

    else 0

end as has\_atat

case

    when dna\_sequence like '%GGG%' then 1

    when dna\_sequence like '%GGGG%' then 1

    else 0

end as has\_ggg

from Samples;

**21/03/2025**

**RANK,AVG,SUM**

**TABLE EMPLOYEES**

**Table creation:**

create table employees(id serial primary key,name varchar(100),department varchar(50),salary int);

**INSERT VALUES**

insert into employees values(1,'Anu','HR',50000),(2,'Balu','HR',70000),(3,'Chintu','IT',80000),(4,'Danvik','HR',89000),(5,'Sri','Finance',67000),(6,'Winzan','Finance',56900);

select\*from employees;

update employees

set salary=89000 where id=2;

**RANK,DENSERANK**

select id,name,department,salary,

rank() over(partition by department order by salary desc) as rank,

dense\_rank() over(partition by department order by salary desc) as dense\_rank,

row\_number() over(partition by department order by salary desc) as row\_number from employees;

**PARTITION BY SUM AND AVG SALARY**

select name,department,salary,sum(salary) over(partition by department ) as dept\_salary,avg(salary) over(partition by department) as avg\_dept\_salary from employees;

**AVG AND SUM SALARY (USING ROUND**)

select department,sum(salary),round(avg(salary),1) from employees group by department;

**LAG AND LEAD**

select

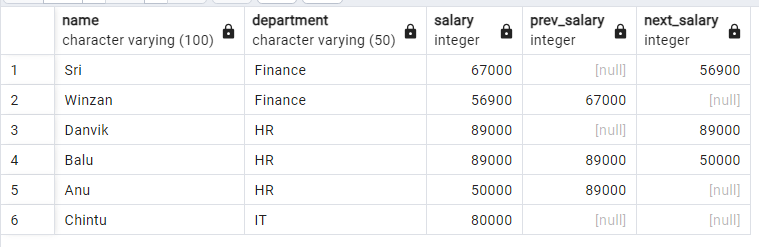
name,

department,

salary,

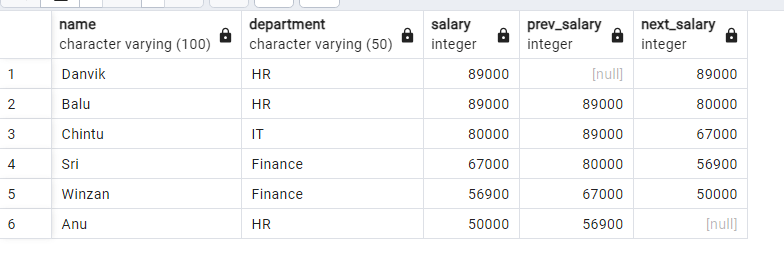
LAG(salary) over(partition by department order by salary desc) as prev\_salary,

lead(salary) over(partition by department order by salary desc) as next\_salary from employees;



select name,department,salary,LAG(salary) over(order by salary desc) as prev\_salary,

lead(salary) over(order by salary desc) as next\_salary from employees;

  
select e.name,e.department,e.salary from employees e

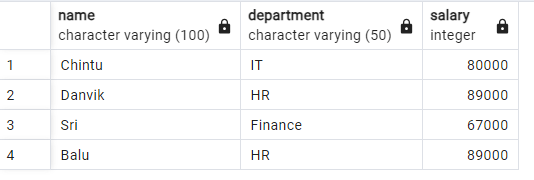
join

(

select department,max(salary) as max\_salary from employees group by department

)

sub on e.department =sub.department and e.salary=sub.max\_salary;



**LEETCODE**

**610. Triangle Judgement**select x,y,z,

case

when x+y>z and x+z>y and y+z>x then 'Yes'

else 'No'

end as triangle from Triangle;