-- drop will basically drop the whole data as well as the table itself

drop table table_name;

-- trucate will only delete all the data inside the table

truncate table table_name;

-- delete

Primary key

Unique

Not null

Foreign key

Check

---------- DQL ----------

select

issueno- (ticket number) 23456

issue- my teams app is not working

support 23456

select * from 23456;

select eid, joiningdate from employee;

Operators:

- **Arithmetic**
- **Comparison**
- **Logical**
- **Bitwise**

Arithmetic :

+

-

*

/

%

^

|/

||/

!

5!= 120

Comparison :

=

>=

<=

<>

!=

>

<

Logical

AND

OR

NOT

Clause:

WHERE: specifying the conditions

(select, update, delete)

Fetch----------

PostgreSQL introduces the FETCH clause, which is used to recover various rows returned by a command.

```
SELECT id, name, age
FROM employee
ORDER BY name
FETCH FIRST ROW ONLY;
```

After executing the above command, we will get the below output, which displays only the first row from the table.

---Some of the most commonly used PostgreSQL conditions are as follows:

- AND Condition
- OR Condition
- AND & OR Condition
- NOT Condition
- LIKE Condition
- IN Condition
- NOT IN Condition
- BETWEEN Condition
- EXIST Condition

PostgreSQL View

The syntax of Create view command is as follows:

```
CREATE VIEW view-name AS
 SELECT column(s)
 FROM table(s)
 [WHERE condition(s)];
```

Create view demo_view as select ename from employee

----------JOIN----------

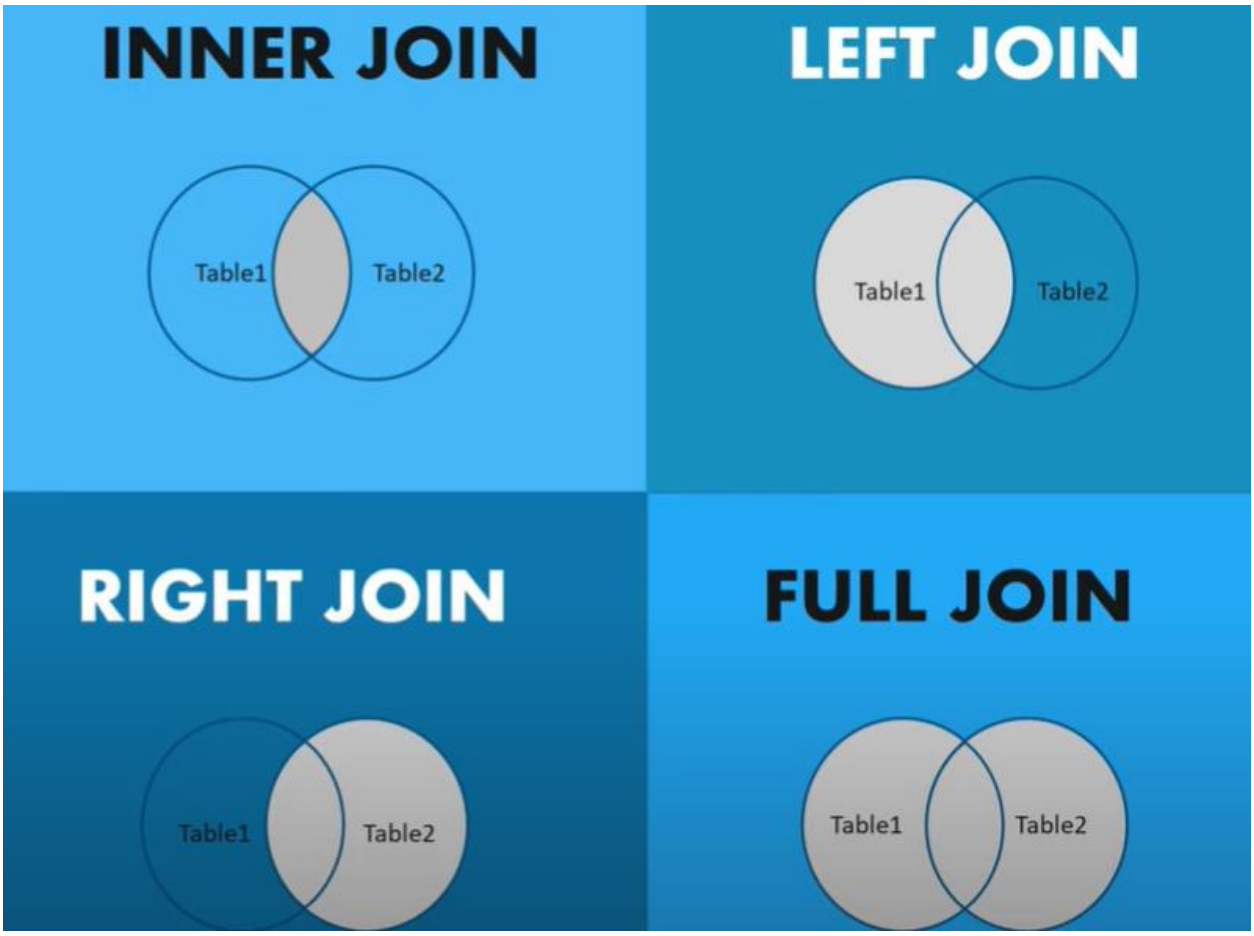
Join is used to combine records from two or more tables.

1. Inner Join
2. Left Join
3. Right Join
4. Outer Join
5. Cross Join

Select eid , ename , dept from employee inner join employee2

On

Employee.common_col = Employee2.common_col;



	eid [PK] integer	ename character varying (20)	age integer	joining_date date	salary real
1	102	Shreya	23	2022-01-13	300000
2	101	Siddharth	22	2022-01-13	500000
3	103	Sayak	22	2022-01-13	500000
4	111	Vibhor	23	2022-01-13	500000
5	107	rishabh	23	2022-01-13	450000
6	108	rishabh	24	2022-01-13	500090
7	109	Tanay	23	2022-01-13	450000
8	110	Kaushik	24	2022-01-13	500090

	eid [PK] integer	dep text
1	101	Dev
2	102	Operati...
3	103	Support
4	120	QA
5	121	CICD
6	130	testing

	des_id [PK] integer	des_details character varying
1	101	Training
2	102	Project1
3	110	OperationProject

https://www.postgresqltutorial.com/postgresql-self-join/

--- UNION Operator:

If you want to combine the results of two or more select statement.

A ={1,2,3} b= {5,6}

A X B = 3 X 2 =6

- Function

Functions allows database resuse.

.. Stored Procedure.

```
void fun1()
{
Sop("fun1 running");
}
```

F2()

--- Syntax of function creation in PostgreSQL

```
create or replace function fun_name(argument(s))
Returns data_type
Language plpgsql
As $$
Declare decalaration
Begin
<function body>
Return var_name;
End;
$$
```

- Trigger

Insert, delete, update

For each row

Create trigger trigger_name {before/after} insert of column_name
On table_name
[
Trigger_Logic
]

__ Index=➔

Explain key word

Index on table:

Create index idex_name on table_name;

Index On Single Column:

Create index index_name on table_name (col_name);

Index on multicolumn:

Create index index_name on table_name (col_name1, col_name2.,….);

Unique Index:

Not only for performance but also for data integrity.

Create unique index index_name on table_name (col_name);

Total no of nodes= 2n+ (n-1)k (B Tree / B+ tree)

```
select * from employee;

----------VIEW----------

create view view_emp as select ename,age from employee;

select * from view_emp;

drop view view_emp;

----------JOIN----------

select * from employee1;



----Inner Join
select employee.ename, employee1.dep from employee inner join employee1
on employee.eid = employee1.eid;


--- Similarly Left Join & Right Join & Full Outer Join

--- cross Join: works like cross product
```

```
select ename, dep from employee cross join employee1;


---Aggregate Functions


--- USer defined functions


create or replace function total_emp()
returns integer as $$
declare total integer;
begin
select count(*) into total from employee;
return total;
end;
$$ language plpgsql;

-- once the function created , We can use it

select total_emp();


------ function to find out sum of 2 values
create function sum(a integer, b integer)
returns integer as $$
begin
return a + b;
end;
$$ language plpgsql;
```

-- use this function as:

select sum(20,30);

----------trigger concept----

create table employee2(id int primary key not null, name text, salary int not null);

create table audit(id int not null, name text not null, entry_date text);

-- two tables created.

-- now we will create a function so that as soon as any value inserted this trigger should run with help of function

```
create or replace function audit_log()
returns trigger as $$
begin
insert into audit(id , name, entry_date) values (new.id,new.name,current_timestamp);
return new;
end;
$$ language plpgsql;
```

--- finally we create our trigger

```
create trigger audit_trigger after insert on employee2
for each row execute procedure audit_log();
```

-- now we will check that our trigger is working or not

insert into employee2 values (130,'Anonymous',30000);

--- value is inserted , now check the audit table there should be logs added for one row.

select * from audit;

----------Index----------

create table test(first_name text not null, last_name text not null,company_name text not null, address text not null, city text not null);

select * from test;

select * from employee;

select * from employee where ename='Vibhor';

explain select * from employee where ename='Vibhor';

-- Seq Scan on employee (cost=0.00..20.00 rows=4 width=74) this was w/o using index.

-- now let's see with index

create index index_emp on employee("ename");

-- again checking with index on ename: Seq Scan on employee (cost=0.00..1.10 rows=1 width=74)

explain select * from employee where ename='Vibhor';

-- drop index

drop index_emp;

TLB hit

TLB miss

Cache

Memory Management:

Virtual Memory

Physical Memory ……..page Replacement Algorithm.Page Fault. TLB. TLA

1. Perform the update and delete in trigger
2. Create a