SQL

(Structured Query Language)

Syntax :

Database creation:

create database db_name ;

Table creation : For used for the create the table.

create table tbl_name (field_1 datatype, field_2 datatype, ..., field_n datatype) ;

Insert values : To insert the values into the table

inset into tbl_name(field_1 , field_2 , ..., field_n) values (value_1, value_2, ... value_n);

Update records : To update the values into the table

update tbl_name set new_values where condition;

Delete records : To delete the values into the table

delete from tbl_name where pk conditon;

Drop records : To drop the values into the table

drop table tbl_name where condition;

Select query :

Select all data :

```
select * from tbl_name ;
```

Select perticular field from the tbl :

```
select field_name form tbl_name;
```

Where clause :

```
select * from tbl_name whare condition;
```

Select combination :

```
select * from tbl_name where field_name in(data);

select * from tbl_name where field_name not in(data);

select * from tbl_name where condition1 and condition2;

select * from tbl_name where condition1 or condtion2;

select field_name from tbl_name as new_field_name;
```

Order by clause :

select */field_name from tbl_name order by ASC/DESC l ; (By default - asc)

Like pattern (Filteration) :

1) Start with - a%

2) End with - %a

3) Exact match - %a%

select */field_name from tbl_name where field_name like 'a% / %a / %a%';

Aggregate function :

There are five aggregate function -

These function can be perform only the numerical values.

max() - select max(field_name) from tbl_name;

min() - select max(field_name) from tbl_name;

avg() - select max(field_name) from tbl_name;

sum() - select max(field_name) from tbl_name;

count() select count(*) from tbl_name;

Normalization Form -

It is used for the reduce the unwanded data.

It is used for the reduce the data redandancy.

There are basic six type -

First normal form - 1NF (Into that the relation can't contain the multi-valued attribute and Every relation can be contine the automic value)

Secondnormal form - 2NF (The relation must be into the 1NF, Into that the raltion can split into the two tables and each table has its own primary key.)

Third normal form - 3NF

Byce and Codd normal form - BCNF

Fourth normal form - 4NF

Fifth normal form - 5 NF

Every table must be normlize upto 2NF.

If we want to get the date and time autometically hence use the this syntax,

table_name timestamptz not null default now()

Type of joins -

inner join : Select all records from both table wich is match to the particular conditon

left join : Select all records from the left table and select only condition matchig records from the both table and display the other table.

right join : Select all records from the right table and select only condition matching recorde from the both table and display on the other table.

cross join : Select all posible records from the both tables and display it into the another table but this table can't accept any condition.

Syntax for join - select */fields from tbl_name join type tbl_name on conditon;

ex , select name, sname from teacher inner join subject on teacher.id = subject.id;

PG SQL inbuild function (string)

lower() - this is used for the change the letter into the lower case. @ex - select lower(menu_name) from menu ; [O/P - shubham]

upper() - this is used for the change the letter into the upper case. @ex - select upper(menu_name) from menu ; [O/P - SHUBHAM]

trim() - this is used for the remove the space from the word. @ex - select trim(menu_name) from menu ; [O/P - shubham]

reverse() - this is used for the reverse the given string. @ex - select reverse(menu_name) from menu ; [O/P - mahbuhs]

length() - this is used for the count the length of the string. @ex - select length(menu_name) from menu ; [O/P - 7]

replace() - this is used for the replace the word/character from the replace. @ex - select replace(menu_name, 'am','u) from menu where mid = 2 ; [O/P - shubhu]

rigth() - this is used for the select the word from the rigth. @ex - select rigth(menu_name,2) from menu ; [O/P - am]

left() - this is used for the select the word from the left. @ex - select left(menu_name,3) from menu ; [O/P - shu]

PG SQL inbuild function (math)

abs() - this is used for the absulate

cbrt() - this is used for the cube root

ceil() - this is used for the get the round value

celling() - as the ceil function

floor() - floor get the round number but lower number

round() - this is used for the get the round number if the number is gratter than or eqaul to 5 then choose upper number elese lower number.

-- User defined function syntax

-- create or replace function function_name(parameters/arguments)

-- returns datatype as var_name

-- begin

```
--     return var_1 + var_2 ;
-- end;
-- $$ language plpgsql ;
```

@Example of the function

```
 create or replace function get_sum(
    a numeric,
    b numeric
   )
  returns numeric as $$
  begin
        return a + b ;
 end;

 $$ language plpgsql ;
```

Syntax for function calling - select function_name(parameters);

View in pgSQL

used for the if we want to any result again and again that time we create the result of that and select any time any where.

Syntax,

create view view_name as select field_name, field_mane from tble_name;

@Ex, create view cdata as select ename,ephone from staff;

created_at timestamptz not null default now()created_at timestamptz not null default now() - If we wand to add the current time and the date

Syntax for function,

There are two types of the fucntion :

Inbuild fucntion

Library fucntion

Uses : If we want perform the any task again and again that time we use the fucntion.

create or replace function function_name

returns argument_type_name (trigger,numeric) as var_name/$$

```
begin
    Quries;
     Return Values;
end;
$var_name$/$$ language plpgsql;
```

Viwes :

If we need access some data into the table again and again and do some word on it that time we use the viwes

Syntax,

create view view_name as Quereis; (@ex , create view cdata as select name,email,ph_no from emp;)

Procedure :

If we want like add some data and delete some data from the table at the same time that time we use the procedure.

Syntax,

```
create or replace procedure procudure_name (
Argument_01,
Argument_n
)
language plphsql as $$
begin
    Quereis;
commit;
end;
$$
```

Calling syntax,

```
call procudure_name;
```

NOTE : Into procedure, dosent perform any changes into the table till the procedure can commit.

TRIGGER :

This is the most imp topic into the SQL which is used for the provide the extra security to the table .

There are two types of the Trigger :

a. before

b. after

Perform which type of the trigger into the table depends upon the commands,

a. insert - after

b. update - before, after

c. delete - before

For trigger always create a fucntion for the perform any kind of operations.

Syntax,

```
create or replace fucntion function_name()
returns trigger as $$
begin
        quries;
      return new;
end;
$$ language plpgsql ;
```

```
create trigger trigger_name
after/before insert/delete/update on table_name
for each row
execute function funtion_name;
```