PDP ACADEMY

Funksiyalar bilan ishlash

Reja:

1. Functions(aggregate, math, String, Date)
2. SQL functions
3. PL/PgSql functions

PDP ACADEMY

Funksiyalar

PostgreSQLda **funksiyalar** , shuningdek, **protsedura** sifatida ham tanilgan, ma`lumotlar bazasida bitta funksiya odatda bir nechta so`rovlar talab qiladigan operatsiyalarni bajarishga imkon beradi. Funksiyalar ma`lumotlar bazasidan qayta foydalanishga imkon beradi, chunki boshqa ilovalar uchun o`rta darajadagi yoki takrorlanuvchi kod o`rniga to`g`ridan-to`g`ri saqlangan protseduralarni ishlatish qulay. Umuman olganda, funksiya tanlash, qo`shish o`chirish va yangilash kabi har qanday operatsiyani bajaradigan SQL ko`rsatmalari to`plamidir.

Funksiyalar siz tanlagan tilda yaratilishi mumkin, masalan SQL, PL/pgSQL, C, Python va boshqalar.

PDP ACADEMY

SQL tilida funksiyalar bilan ishlash

PostgreSQL da 4 xil turdagi funksiayalar mavjud:

- Ichki funksiyalar;
- SQL tilida yozilgan funksiyalar;
- Prosedurali dasturlash tili(Pl/pgsql)da yozilgan funksiyalar;
- C tilida yozilgan funksiayalar.

Ichki funksiyalar

- Aggregate Functions
- Math Functions
- String Functions
- Date Functions
- …..

Aggregate Functions

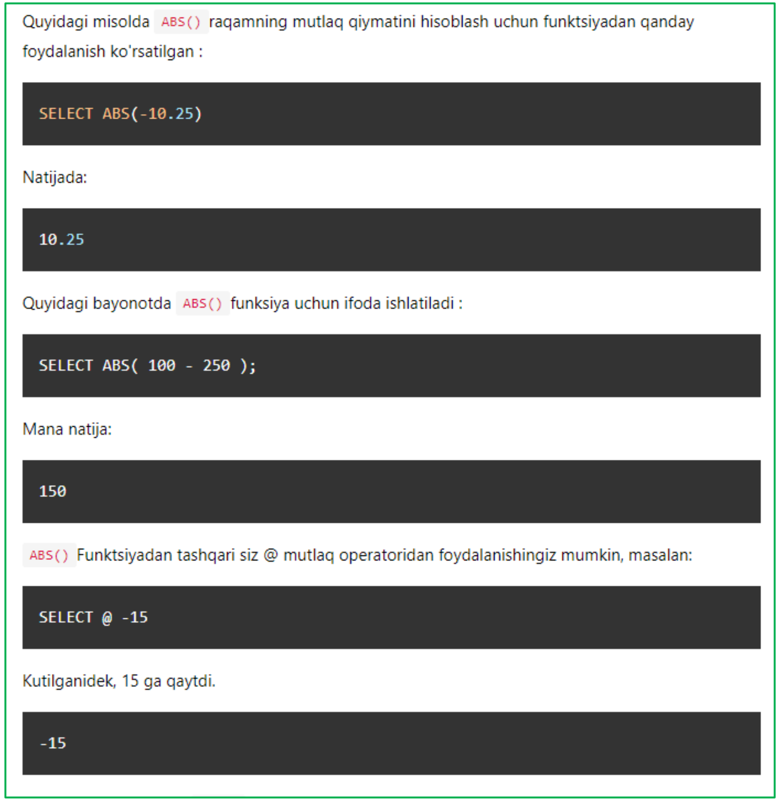
- AVG() - o`rtacha qiymatni qaytaradi.
- COUNT() – qiymatlar sonini qaytaradi.
- MAX() – maksimal qiymatni qaytaradi.
- MIN() – minimal qiymatni qaytaradi.
- SUM() - barcha yoki alohida qiymatlarning yig`indisini qaytaradi.

PDP ACADEMY

Math Functions

Function	Description	Example	Result
ABS	Calculate the absolute value of a number	ABS(-10)	10
FLOOR	Round a number down to the nearest integer, which is less than or equal to number	FLOOR(10.6)	10
MOD	Divide the first parameter by the second one and return the remainder	MOD(10,4)	2
PI	Return the value of PI	PI()	3.141592654
POWER	Raise a numeric value to the power of a second numeric value	POWER(5, 3)	125
ROUND	Round a number to the nearest integer or to a specified decimal places	ROUND(10.3)	10
SCALE	Return the number of decimal digits in the fractional part	SCALE(1.234)	3
SQRT	Return the square root of a numeric value	SQRT(3.0)	1.732050808
RANDOM	Return a random number that ranges from 0 to 1		0.968435665

Math Functions examples



Quyidagi misolda `ABS()` raqamning mutlaq qiymatini hisoblash uchun funktsiyadan qanday foydalanish ko'rsatilgan :

```
SELECT ABS(-10.25)
```

Natijada:

```
10.25
```

Quyidagi bayonotda `ABS()` funksiya uchun ifoda ishlatiladi :

```
SELECT ABS( 100 - 250 );
```

Mana natija:

```
150
```

`ABS()` Funktsiyadan tashqari siz @ mutlaq operatoridan foydalanishingiz mumkin, masalan:

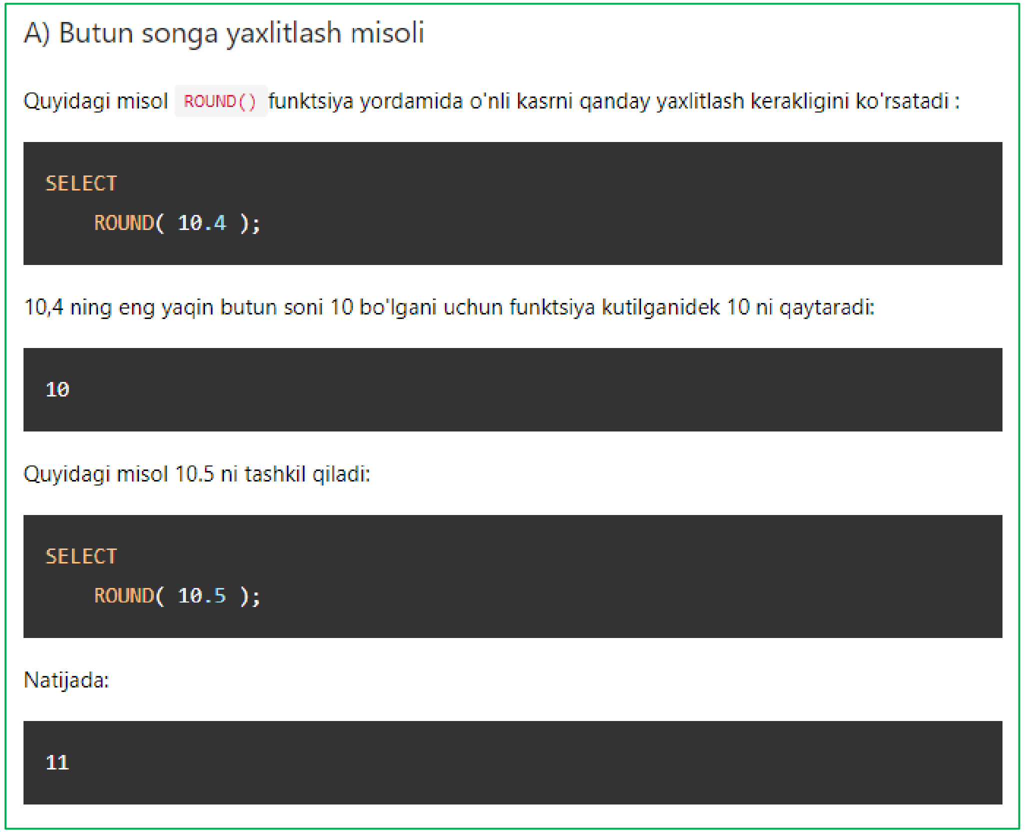
```
SELECT @ -15
```

Kutilganidek, 15 ga qaytdi.

```
-15
```

PDP ACADEMY

Math Functions examples



A) Butun songa yaxlitlash misoli

Quyidagi misol `ROUND()` funktsiya yordamida o'nli kasrni qanday yaxlitlash kerakligini ko'rsatadi :

```
SELECT
    ROUND( 10.4 );
```

10,4 ning eng yaqin butun soni 10 bo'lgani uchun funktsiya kutilganidek 10 ni qaytaradi:

```
10
```

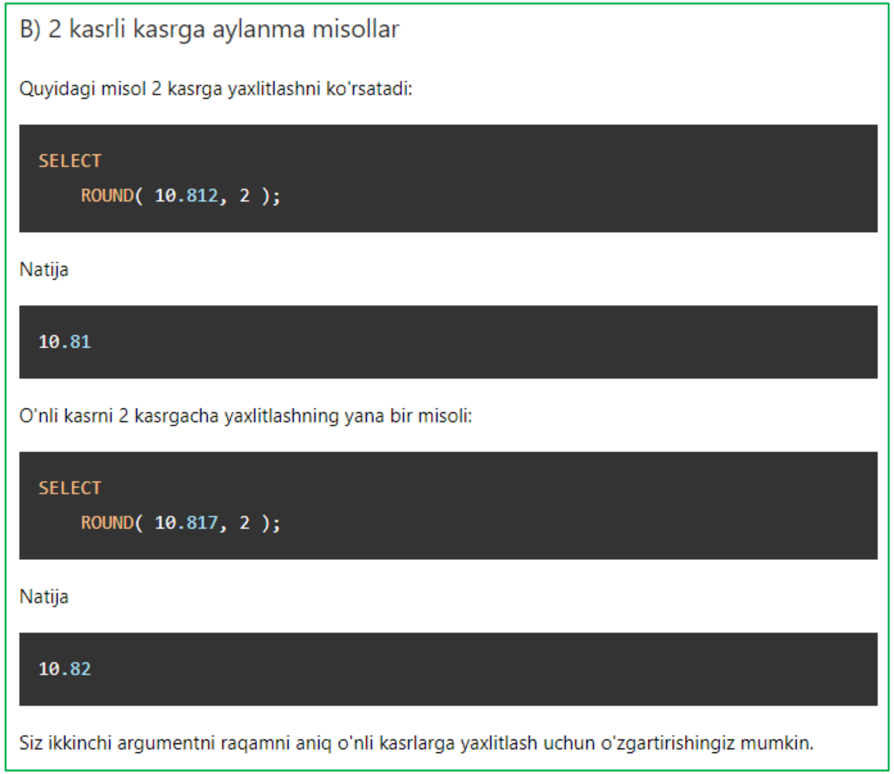
Quyidagi misol 10.5 ni tashkil qiladi:

```
SELECT
    ROUND( 10.5 );
```

Natijada:

```
11
```

Math Functions examples



B) 2 kasrli kasrga aylanma misollar

Quyidagi misol 2 kasrga yaxlitlashni ko'rsatadi:

```
SELECT
    ROUND( 10.812, 2 );
```

Natija

```
10.81
```

O'nli kasrni 2 kasrgacha yaxlitlashning yana bir misoli:

```
SELECT
    ROUND( 10.817, 2 );
```

Natija

```
10.82
```

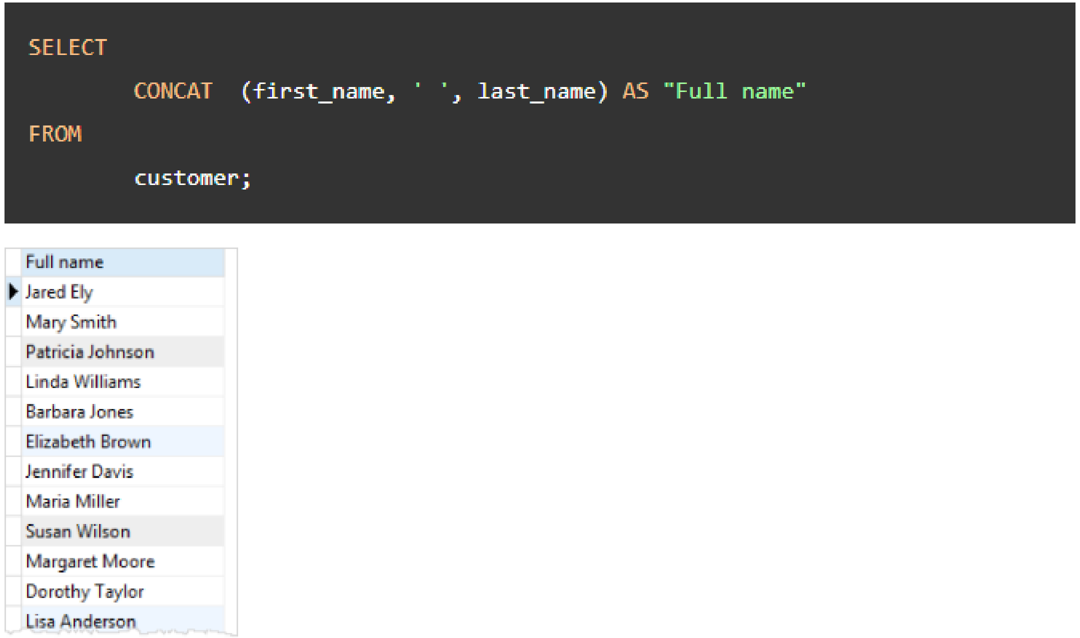
Siz ikkinchi argumentni raqamni aniq o'nli kasrlarga yaxlitlash uchun o'zgartirishingiz mumkin.

String Functions

Function	Description	Example	Result
ASCII	Return the ASCII code value of a character or Unicode code point of a UTF8 character	ASCII(‘A’)	65
CHR	Convert an ASCII code to a character or a Unicode code point to a UTF8 character	CHR(65)	‘A’
CONCAT	Concatenate two or more strings into one	CONCAT(‘A’,’B’,’C’)	‘ABC’
FORMAT	Format arguments based on a format string	FORMAT(‘Hello %s’,’PostgreSQL’)	‘Hello PostgreSQL’
INITCAP	Convert words in a string to title case	INITCAP(‘hI tHERE’)	Hi There
RIGHT	Return last n characters in the string. When n is negative, return all but first |n| characters.	RIGHT(‘ABC’, 2)	‘BC’
RPAD	Pad on the right of a string with a character to a certain length	RPAD(‘ABC’, 6, ‘xo’)	‘ABCxox’

PDP ACADEMY

String Functions example



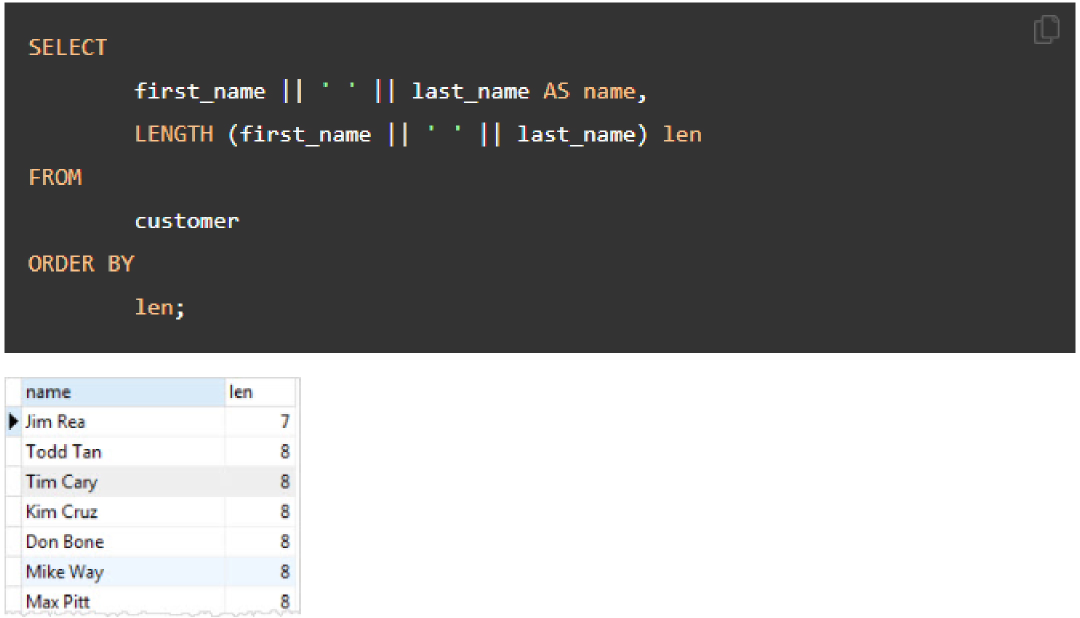
```
SELECT
        CONCAT  (first_name, ' ', last_name) AS "Full name"
FROM
        customer;
```

Full name
Jared Ely
Mary Smith
Patricia Johnson
Linda Williams
Barbara Jones
Elizabeth Brown
Jennifer Davis
Maria Miller
Susan Wilson
Margaret Moore
Dorothy Taylor
Lisa Anderson

String Functions

Function	Description	Example	Result
LEFT	Return the first n character in a string	LEFT(‘ABC’,1)	‘A’
LENGTH	Return the number of characters in a string	LENGTH(‘ABC’)	3
LOWER	Convert a string to lowercase	LOWER(‘hI tHERE’)	‘hi there’
LPAD	Pad on the left a a string with a character to a certain length	LPAD(‘123’, 5, ’00’)	‘00123’
LTRIM	Remove the longest string that contains specified characters from the left of the input string	LTRIM(‘00123’, ‘0’)	‘123’
POSITION	Return the location of a substring in a string	POSITION(‘B’ in ‘A B C’)	3
RTRIM	Remove the longest string that contains specified characters from the right of the input string	RTRIM(‘abcxxzx’, ‘xyz’)	‘abc’
SUBSTRING	Extract a substring from a string	SUBSTRING(‘ABC’,1,1)	A’
TRIM	Remove the longest string that contains specified characters from the left, right or both of the input string	TRIM(‘ ABC ‘)	‘ABC’

String Functions example



```
SELECT
        first_name || ' ' || last_name AS name,
        LENGTH (first_name || ' ' || last_name) len
FROM
        customer
ORDER BY
        len;
```

name	len
Jim Rea	7
Todd Tan	8
Tim Cary	8
Kim Cruz	8
Don Bone	8
Mike Way	8
Max Pitt	8

PDP ACADEMY

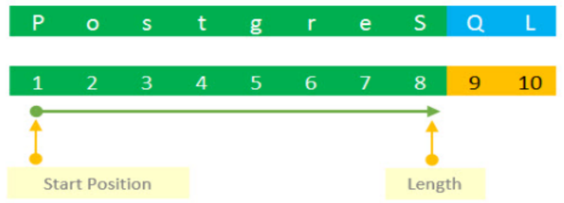
String Functions

Function	Description	Example	Result
REPEATE	Repeat string the specified number of times	REPEAT(‘*’, 5)	‘*****’
REPLACE	Replace all occurrences in a string of substring from with substring to	REPLACE(‘ABC’,’B’,’A’)	‘AAC’
REVERSE	Return reversed string.	REVERSE(‘ABC’)	‘CBA’
SUBSTRING	Extract a substring from a string	SUBSTRING(‘ ABC’,1,1)	A’
TRIM	Remove the longest string that contains specified characters from the left, right or both of the input string	TRIM(‘ ABC ‘)	‘ABC’
UPPER	Convert a string to uppercase	UPPER(‘hI tHERE’)	‘HI THERE’

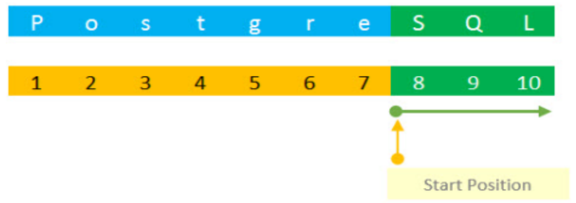
String Functions example

```sql
SELECT
	SUBSTRING ('PostgreSQL', 1, 8); -- PostgreS
SELECT
	SUBSTRING ('PostgreSQL', 8); -- SQL
```

Birinchi bayonotda biz uzunligi 8 bo'lgan va `PostgreSQL` satrning birinchi belgisidan boshlangan pastki qatorni chiqaramiz . `PostgreS` natijaga erishamiz . Quyidagi rasmga qarang:



Ikkinchi bayonotda biz 8-pozitsiyada boshlangan pastki qatorni chiqaramiz va biz uzunlik parametrini o'tkazib yuboramiz. Pastki satr 8 dan boshlanadigan satr bo'lib, u `SQL` .

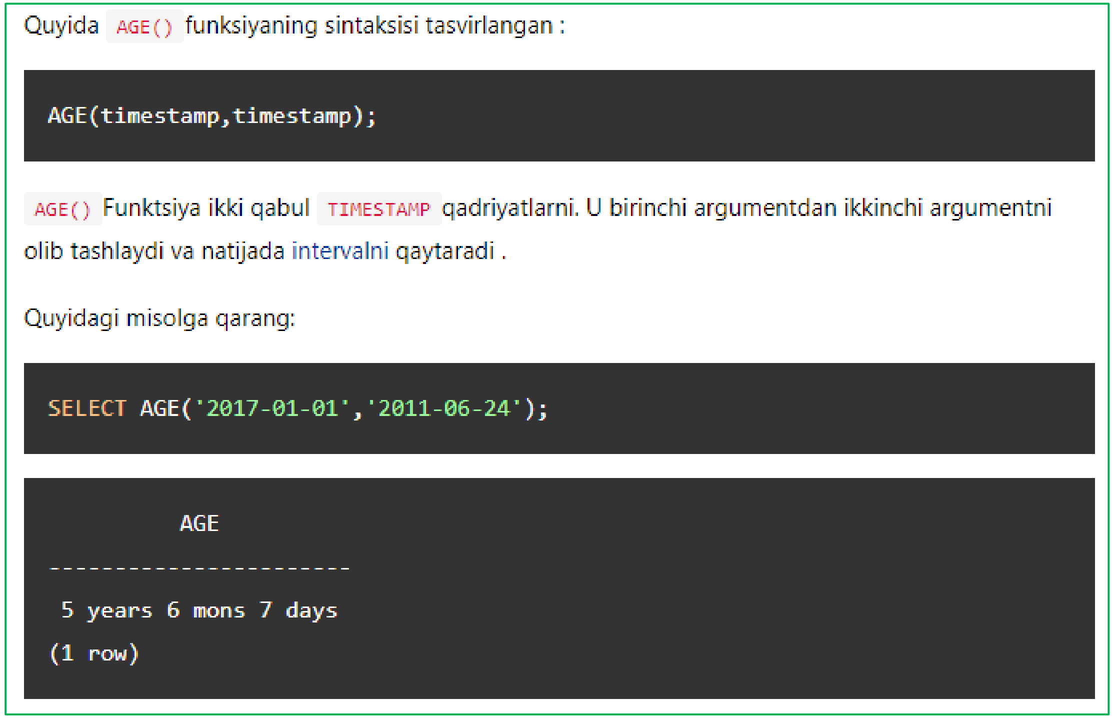


PDP ACADEMY

Date Functions

Function	Return Type	Description
AGE	INTERVAL	Calculate ages between two timestamps and returns a “symbolic” result which uses years and months
CURRENT_DATE	DATE	Return the current date
CURRENT_TIME	TIMESTAMPTZ	Return the current time
CURRENT_TIMESTAMP	TIMESTAMPTZ	Return the current date and time with time zone at which the current transaction starts
NOW	TIMESTAMPTZ	Return the date and time with time zone at which the current transaction start
TO_DATE	DATE	Convert a string to a date
TO_TIMESTAMP	TIMESTAMPTZ	Convert a string to a timestamp
……	…….	…….

Date Functions example



Quyida `AGE()` funksiyaning sintaksisi tasvirlangan :

```
AGE(timestamp,timestamp);
```

`AGE()` Funktsiya ikki qabul `TIMESTAMP` qadriyatlarni. U birinchi argumentdan ikkinchi argumentni olib tashlaydi va natijada intervalni qaytaradi .

Quyidagi misolga qarang:

```
SELECT AGE('2017-01-01','2011-06-24');
```

```
          AGE
-----------------------
 5 years 6 mons 7 days
(1 row)
```

SQL Function

- SQL funksiya – bu oxirgi so`rovning natijasini qaytaruvchi ketma-ket bajariladigan sql so`rovlar ketma-ketligidir.
- SQL funksiya body qismi ‘;’ bilan ajaratilgan sql so`rovlardan tashkil topadi. Oxirgi so`rovdan keyin ‘;’ qo`yish majburiy emas.
- Agarda funksiyaning qaytaradigan toifasi(return type) void bo`lmasa returnda oxirgi operator   SELECT, INSERT, UPDATE yoki DELETE  bo`lish kerak.
- SQL da yozilgan istalgan buyruqlar ketma-ketligini SQL funksiya sifatida belgilashimiz mumkin. Lekin oxirgi buyruq SELECT operatori bo`lishi yoki RETURN bilan boshqa operator bo`lishi kerak.

SQL Function yaratish

```
CREATE FUNCTION one() RETURNS integer AS $$
      SELECT 1 AS result;
$$ LANGUAGE SQL;

-- Yoki:

CREATE FUNCTION one() RETURNS integer AS '
      SELECT 1 AS result; '
LANGUAGE SQL;
```

Ma`lumot qaytarmaydigan funksiya

```
CREATE FUNCTION clean_emp() RETURNS void AS
' DELETE FROM emp
            WHERE salary < 0;
' LANGUAGE SQL;
```

Argumentlar bilan ishlash

```
CREATE FUNCTION add_em(x integer, y integer)
RETURNS integer AS $$
      SELECT x + y;
$$ LANGUAGE SQL;
```

```
CREATE FUNCTION add_em(integer, integer)
RETURNS integer AS $$
      SELECT $1 + $2;
$$ LANGUAGE SQL;
```

Default qiymatli argumentlar

```
CREATE FUNCTION foo(a int, b int DEFAULT 2, c int
DEFAULT 3) RETURNS int
LANGUAGE SQL
AS $$
       SELECT $1 + $2 + $3;
$$;
```

Chiqish parameterlari bilan ishlash

```
CREATE FUNCTION add_em(IN x int,IN y int, OUT sum int)AS
      'SELECT x + y'
LANGUAGE SQL;



CREATE FUNCTION sum_n_product(x int,y int,
OUT sum int, OUT product int) AS
      'SELECT x + y, x * y'
LANGUAGE SQL;
```

PDP ACADEMY

PDP ACADEMY

PL/PgSQL Function

Programming in PostgreSQL
with Pl/pgSQL

Procedural Language extension to postgreSQL

Introduction to PostgreSQL PL/pgSQL

PL/pgSQL – bu PostgreSQL MOBT uchun mo`ljallangan procedurali dasturlash tili.

PL/pgSQL yaxlit bir logika asosida server objectlarini yaratish orqali PostgreSQL ning imkoniyatlarini kengaytiradi.

PL/pgSQL asosan quyidagilar uchun mo`ljallangan:

•user-defined function(foydalanuvchi funksiya)lar, saqlanadigan proceduralar va triggerlarni yaratish;

•if, case, va loop kabi control structure lardan foydalanish orqali   standard SQL imkoniyatini oshirish.

CREATE FUNCTION

```
create [or replace] function function_name(param_list)
        returns return_type
        language plpgsql
        as
$$
declare
        -- variable declaration
begin
        -- logic
exception
        --exceptions
end;
$$
```

CREATE PROCEDURE

```
create [or replace] procedure procedure_name(parameter_list)
language plpgsql
as $$
declare
        -- variable declaration
begin
        -- stored procedure body
end; $$
```

VARIABLES & CONSTANTS

variable_name data_type [:= expression];

```
counter integer := 1;
first_name varchar(50) := 'John';
last_name varchar(50) := 'Doe’;
payment numeric(11,2) := 20.5;
```

VARIABLES & CONSTANTS

```
variable_name  table_name.column_name%type;
```

film_title film.title**%type**;

featured_title film_title**%type**;

VARIABLES & CONSTANTS

constant_name **constant** data_type := expression;

```
v constant numeric := 0.1;
net_price numeric := 20.5;
```

CONTROL STRUCTURES

If then

Case when

Loop

While Loop

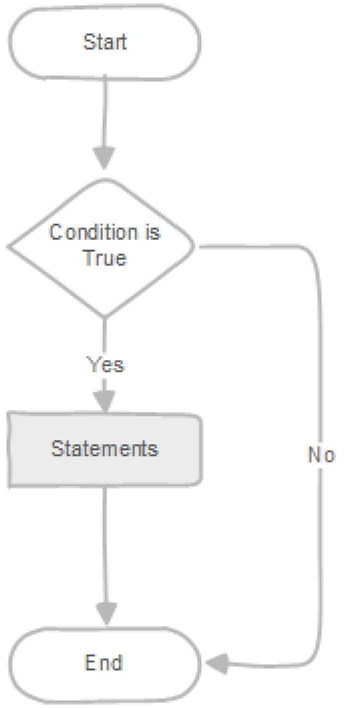
For Loop

Exit

Continue

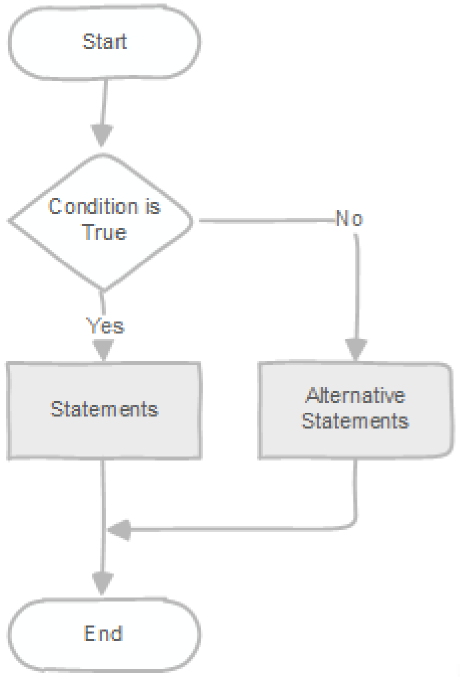
if-then statement

```
if condition then
   statements;
end if;
```



if-then-else statement

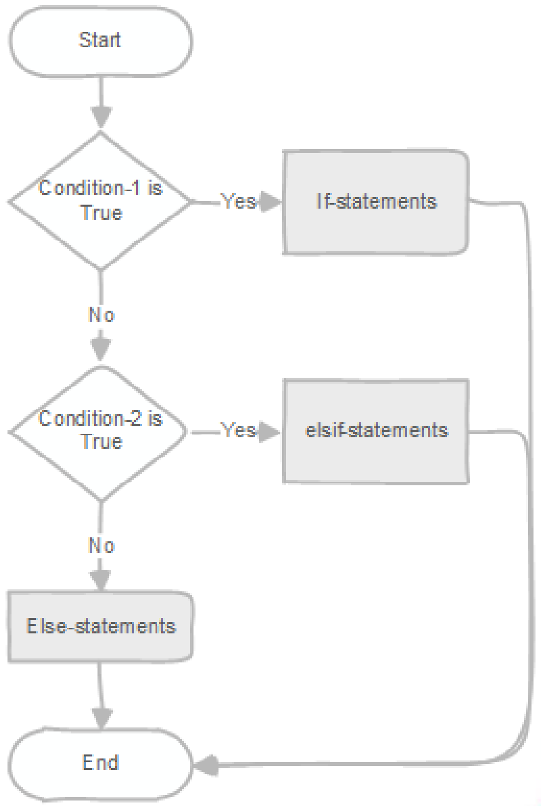
```
if condition then
   statements;
else
   alternative-statements;
end if;
```



PDP ACADEMY

if-then-elsif statement

```
if condition_1 then
   statement_1;
elsif condition_2 then
   statement_2
...
elsif condition_n then
   statement_n;
else
   else-statement;
end if;
```



Case statement

Simple case statement

```
case search-expression
   when expression_1 [, expression_2, ...] then
        when-statements [ ... ]
   [else
      else-statements ]
END case;
```

Searched case statement

```
case
   when boolean-expression-1 then
      statements
   [when boolean-expression-2 then
      statements ... ]
   [ else statements ]
end case;
```

Simple Loop

```
loop
   statements;
end loop;
```

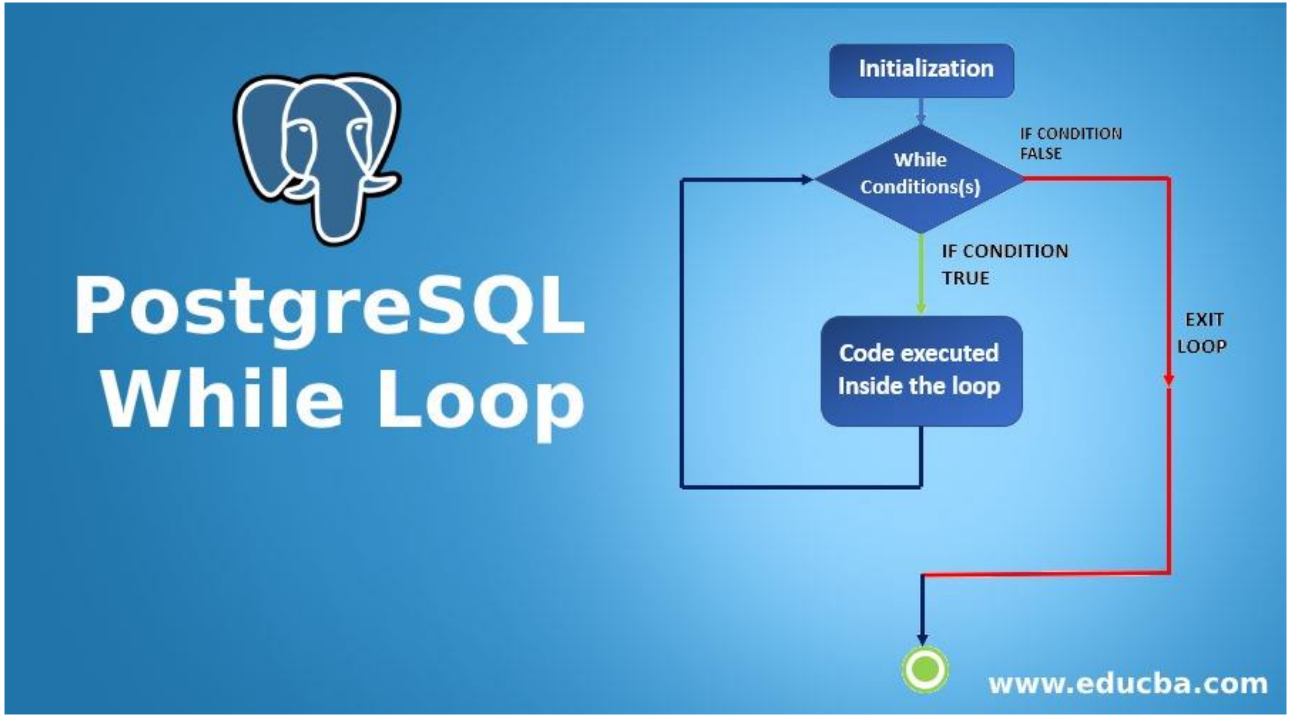
```
loop
   statements;
   if condition then
      exit;
   end if;
end loop;
```

```
loop
   statements;
   exit when condition
end loop;
```

While Loop

```
while condition loop
   statements;
end loop;
```

PDP ACADEMY


PostgreSQL While Loop
Initialization
While Conditions(s)
IF CONDITION FALSE
IF CONDITION TRUE
Code executed Inside the loop
EXIT LOOP
www.educba.com

PDP ACADEMY

For Loop

```
for loop_counter in [ reverse ] from.. to [ by step ] loop

       statements
end loop;
```

PDP ACADEMY

REPORTING MESSAGES & ERRORS

```
do $$
begin
       raise info 'information message %', now();
       raise warning 'warning message %', now();
       raise notice 'notice message %', now();
end $$;
```

REPORTING MESSAGES & ERRORS

```
do $$
begin
        statements;
exception
        [when condition [or condition...] then handle_exception;
        [when condition [or condition...] then 
handle_exception;]
        [when others then handle_other_exceptions; ]
end;
```

PDP ACADEMY

REPORTING MESSAGES & ERRORS

```
exception
        when no_data_found then
        raise exception 'data % not found’, data;
```

https://www.postgresql.org/docs/current/errcodes-appendix.html

```
exception
        when  sqlstate 'P0002' then
        raise exception 'data % not found’, data;
```

PDP ACADEMY

E`TIBORINGIZ UCHUN RAHMAT!