PDP ACADEMY

Opertorlar, PostgreSQL fundamentals

Reja:

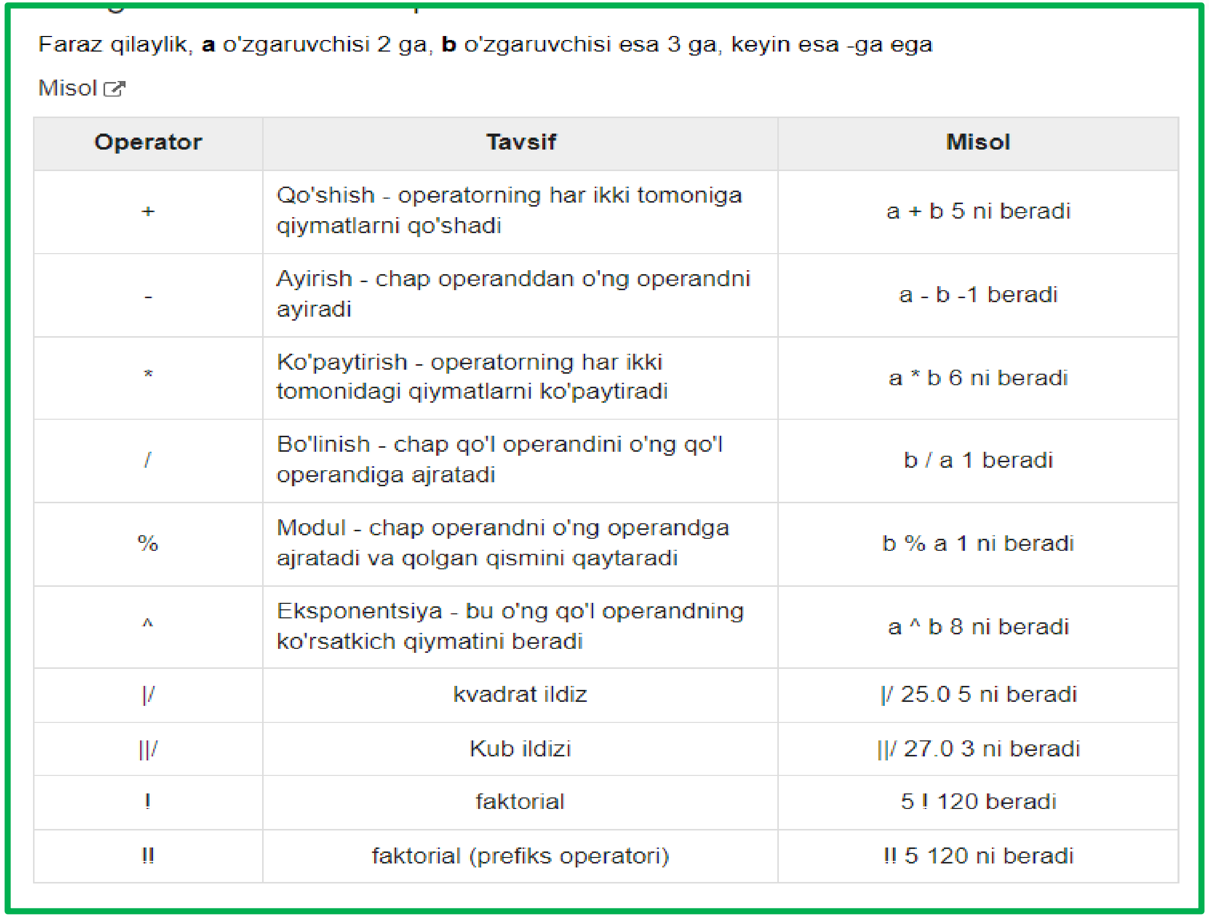
1. Operators(Arithmetic, Comparison, Logic)
2. PostgreSQL fundamentals (select distinct, order by, where, limit, fetch, in, between, like, is null, table and column aliases)

PostgreSQLda operator nima?

Operator - bu taqqoslash va arifmetik amallar kabi amal(lar)ni bajarish uchun asosan PostgreSQL bayonotining WHERE bandida ishlatiladigan zahiradagi so`z yoki belgi.
Operatorlar PostgreSQL bayonotida shartlarni belgilash va bayonotda bir nechta shartlar uchun birikma sifatida xizmat qilish uchun ishlatiladi.

- **Arifmetik operatorlar**
- **Taqqoslash operatorlari**
- **Mantiqiy operatorlar**
- **Bitli operatorlar**
- **V.h…**

PostgreSQL arifmetik operatorlar

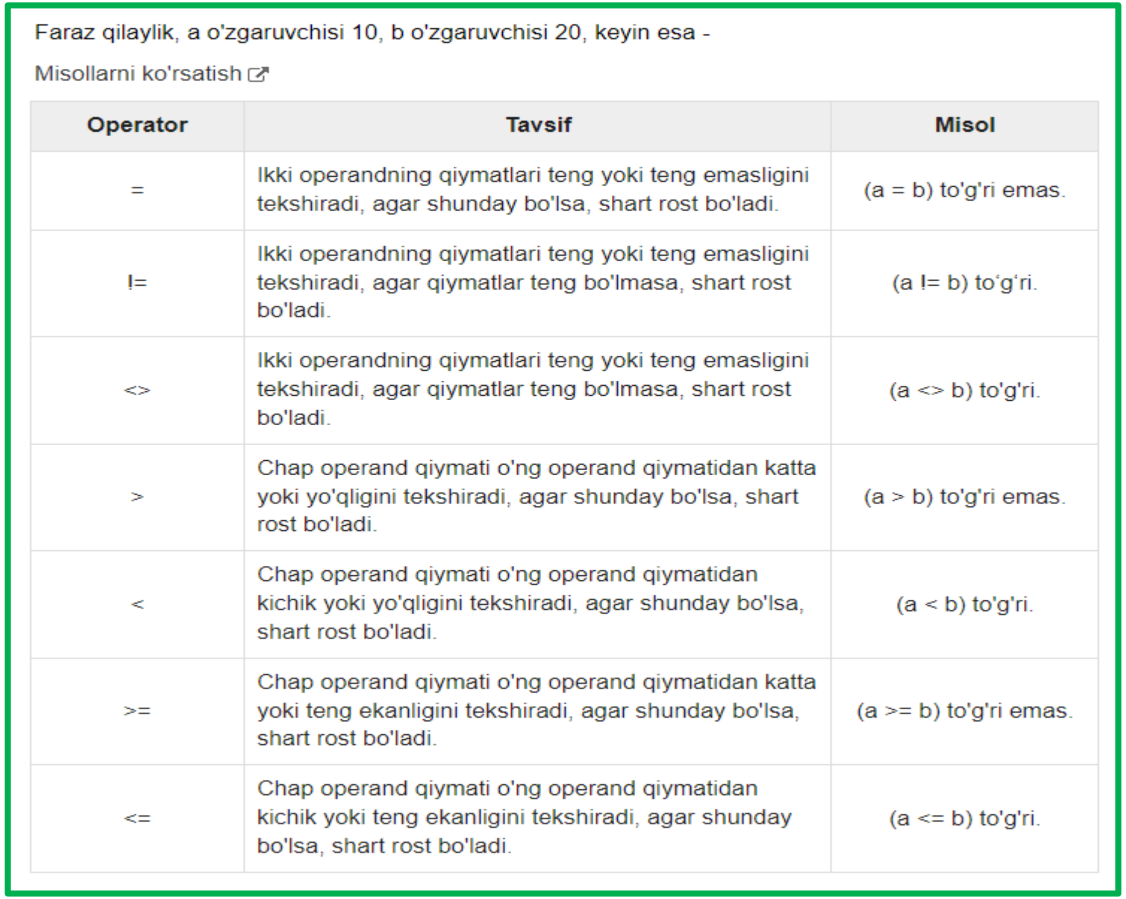


Faraz qilaylik, **a** o'zgaruvchisi 2 ga, **b** o'zgaruvchisi esa 3 ga, keyin esa -ga ega

Misol

Operator	Tavsif	Misol
+	Qo'shish - operatorning har ikki tomoniga qiymatlarni qo'shadi	a + b 5 ni beradi
-	Ayirish - chap operanddan o'ng operandni ayiradi	a - b -1 beradi
*	Ko'paytirish - operatorning har ikki tomonidagi qiymatlarni ko'paytiradi	a * b 6 ni beradi
/	Bo'linish - chap qo'l operandini o'ng qo'l operandiga ajratadi	b / a 1 beradi
%	Modul - chap operandni o'ng operandga ajratadi va qolgan qismini qaytaradi	b % a 1 ni beradi
^	Eksponentsiya - bu o'ng qo'l operandning ko'rsatkich qiymatini beradi	a ^ b 8 ni beradi
|/	kvadrat ildiz	|/ 25.0 5 ni beradi
||/	Kub ildizi	||/ 27.0 3 ni beradi
!	faktorial	5 ! 120 beradi
!!	faktorial (prefiks operatori)	!! 5 120 ni beradi

PostgreSQL solishtirish operatorlari

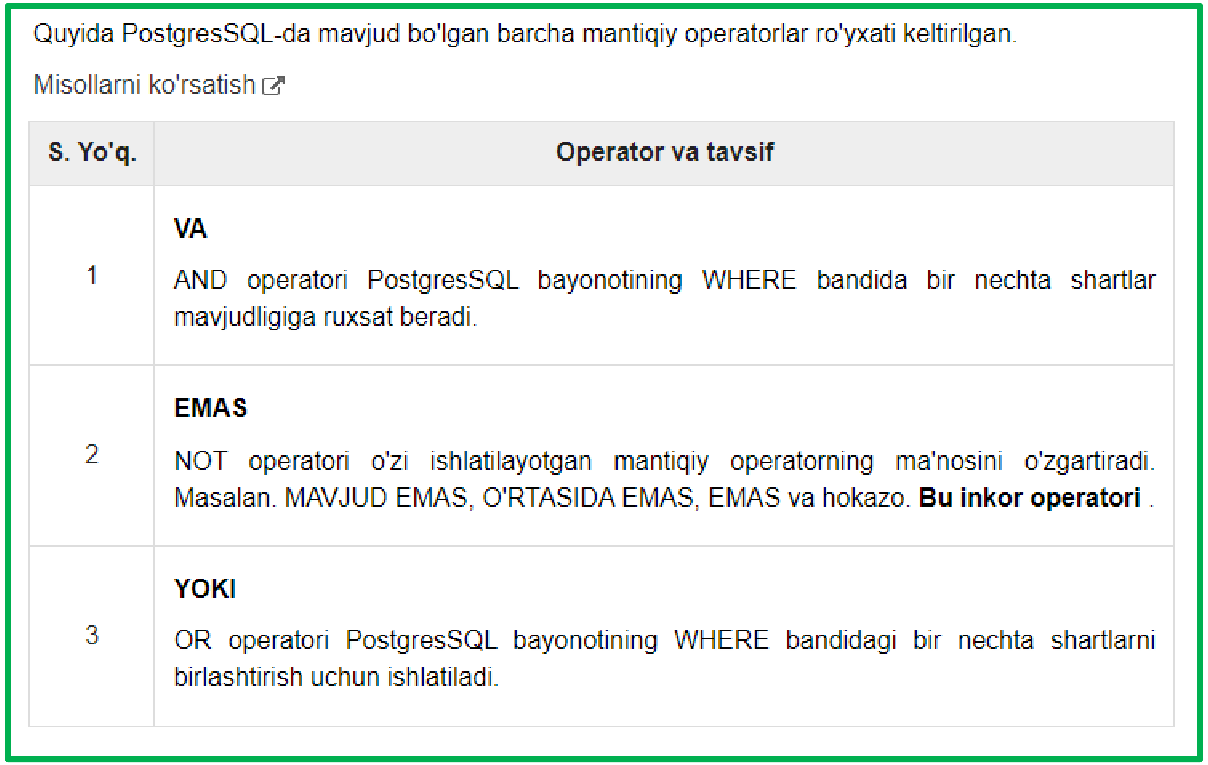


Faraz qilaylik, a o'zgaruvchisi 10, b o'zgaruvchisi 20, keyin esa -

Misollarni ko'rsatish

Operator	Tavsif	Misol
=	Ikki operandning qiymatlari teng yoki teng emasligini tekshiradi, agar shunday bo'lsa, shart rost bo'ladi.	(a = b) to'g'ri emas.
!=	Ikki operandning qiymatlari teng yoki teng emasligini tekshiradi, agar qiymatlar teng bo'lmasa, shart rost bo'ladi.	(a != b) to'g'ri.
<>	Ikki operandning qiymatlari teng yoki teng emasligini tekshiradi, agar qiymatlar teng bo'lmasa, shart rost bo'ladi.	(a <> b) to'g'ri.
>	Chap operand qiymati o'ng operand qiymatidan katta yoki yo'qligini tekshiradi, agar shunday bo'lsa, shart rost bo'ladi.	(a > b) to'g'ri emas.
<	Chap operand qiymati o'ng operand qiymatidan kichik yoki yo'qligini tekshiradi, agar shunday bo'lsa, shart rost bo'ladi.	(a < b) to'g'ri.
>=	Chap operand qiymati o'ng operand qiymatidan katta yoki teng ekanligini tekshiradi, agar shunday bo'lsa, shart rost bo'ladi.	(a >= b) to'g'ri emas.
<=	Chap operand qiymati o'ng operand qiymatidan kichik yoki teng ekanligini tekshiradi, agar shunday bo'lsa, shart rost bo'ladi.	(a <= b) to'g'ri.

PostgreSQL mantiqiy operatorlari



Quyida PostgresSQL-da mavjud bo'lgan barcha mantiqiy operatorlar ro'yxati keltirilgan.

Misollarni ko'rsatish

S. Yo'q.	Operator va tavsif
1	**VA** AND operatori PostgresSQL bayonotining WHERE bandida bir nechta shartlar mavjudligiga ruxsat beradi.
2	**EMAS** NOT operatori o'zi ishlatilayotgan mantiqiy operatorning ma'nosini o'zgartiradi. Masalan. MAVJUD EMAS, O'RTASIDA EMAS, EMAS va hokazo. **Bu inkor operatori** .
3	**YOKI** OR operatori PostgresSQL bayonotining WHERE bandidagi bir nechta shartlarni birlashtirish uchun ishlatiladi.

PDP ACADEMY

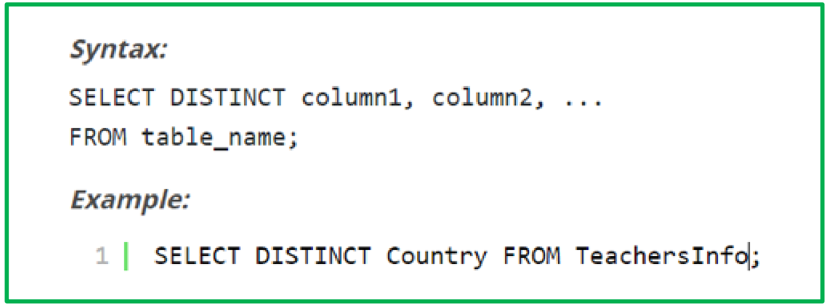
PostgreSQL fundamentals

Biz ma`lumotlarni tanlash, natijalar to`plamini saralash va qatorlarni filtrlash kabi asosiy so`rov usullaridan foydalangan holda bitta jadvaldan ma`lumotlarni so`rashni o`rganamiz. Keyin biz bir nechta jadvallarni birlashtirish, o`rnatilgan operatsiyalardan foydalanish va quyi so`rovni yaratish kabi ilg`or so`rovlarni o`rganamiz.

SELECT bilan birga quyidagi kalit so`zlardan foydalanish mumkin:

DISTINCT,
ORDER BY,
WHERE,
LIMIT,
FETCH,
IN,
BETWEEN,
LIKE,
IS NULL,
TABLE ALIASES,
COLUMN ALIASES

So`rov orqali qaytarilgan natijalar to`plamidan takroriy qatorlarni olib tashlash uchun PostgreSQLda **SELECT DISTINCT** dan foydalaniladi.



Syntax:

```
SELECT DISTINCT column1, column2, ...
FROM table_name;
```

Example:

```
SELECT DISTINCT Country FROM TeachersInfo;
```

PDP ACADEMY

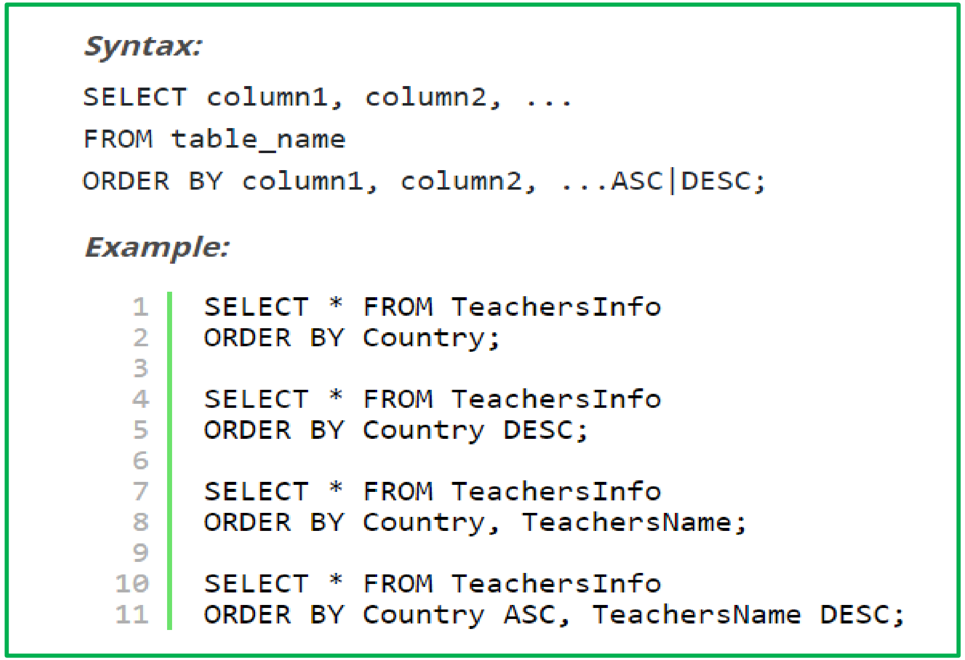
SELECT DISTINCT EXAMPLE

```
SELECT  DISTINCT column_1
FROM table_name;
```

```
SELECT DISTINCT
            bcolor
FROM
            t1
ORDER BY
            bcolor;
```

```
SELECT DISTINCT
            bcolor, fcolor
FROM  t1
ORDER BY
            bcolor,
            fcolor;
```

ORDER BY operatori kerakli natijalarni o`sish yoki kamayish tartibida saralash uchun ishlatiladi. Odatda natijalar o`sish tartibida saralanadi. Agar siz yozuvlarni kamayish tartibida saralashni istasangiz, ***DESC*** kalit ***so'zidan*** foydalanishingiz kerak.



Syntax:

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ...ASC|DESC;
```

Example:

```
SELECT * FROM TeachersInfo
ORDER BY Country;

SELECT * FROM TeachersInfo
ORDER BY Country DESC;

SELECT * FROM TeachersInfo
ORDER BY Country, TeachersName;

SELECT * FROM TeachersInfo
ORDER BY Country ASC, TeachersName DESC;
```

PDP ACADEMY

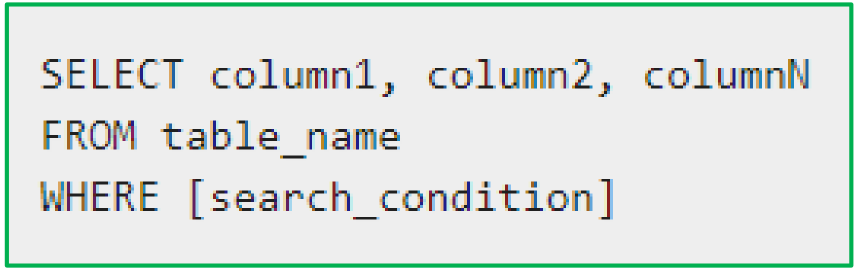
ORDER BY EXAMPLE

```
SELECT
            column_1, column_2
FROM
            table_name
ORDER BY
            column_1 ASC,  column_2 DESC;
```

```
SELECT

            first_name,

            last_name

FROM

            customer

ORDER BY

            first_name ASC,

            last_name DESC;
```

```
SELECT

    first_name,

    last_name

FROM

    customer

ORDER BY

    last_name DESC;
```

PostgreSQLda **WHERE** bandi bitta jadvaldan ma`lumotlarni olish yoki bir nechta jadvallar bilan birlashishda shartni belgilash uchun ishlatiladi.
Agar berilgan shart bajarilsa, u jadvaldan aniq qiymatni qaytaradi. Siz WHERE bandidan foydalanib, natijalar to'plamiga kiritilishini istamagan qatorlarni filtrlashingiz mumkin.
WHERE bandi nafaqat SELECT iborasida, balki UPDATE, DELETE iboralarida va hokazolarda ham qo`llaniladi.



```
SELECT column1, column2, columnN
FROM table_name
WHERE [search_condition]
```

SELECT WHERE

```
SELECT  column_1, column_2 … column_n
FROM    table_name
WHERE   conditions;
```

Operator	Description
=	Equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<> or !=	Not equal
AND	Logical operator AND
OR	Logical operator OR

SELECT WHERE

```
SELECT last_name, first_name
FROM customer
WHERE first_name = 'Jamie';

SELECT last_name, first_name
FROM customer
WHERE first_name = 'Jamie' AND last_name = 'Rice';

SELECT customer_id,
amount,
payment_date
FROM payment
WHERE amount <= 1 OR amount >= 8;
```

PostgreSQL **LIMIT** SELECT bayonoti tomonidan qaytarilgan ma`lumotlar miqdorini cheklash uchun ishlatiladi.

```
SELECT column1, column2, columnN
FROM table_name
LIMIT [no of rows]
```

Quyida **LIMIT** ning **OFFSET** bilan birga ishlatilgandagi sintaksisi keltirilgan:

```
SELECT column1, column2, columnN
FROM table_name
LIMIT [no of rows] OFFSET [row num]
```

PDP ACADEMY

SELECT LIMIT

```
SELECT *
FROM table_name
LIMIT n;
```

```
SELECT *
FROM  table_name
LIMIT n OFFSET m;
```

```
SELECT
          film_id,
          title,
          release_year
FROM
          film
ORDER BY
          film_id
LIMIT 5;
```

```
SELECT
          film_id,
          title,
          release_year
FROM
          film
ORDER BY
          film_id
LIMIT 4 OFFSET 3;
```

PDP ACADEMY

SELECT FETCH

```
SELECT
    film_id,
    title
FROM
    film
ORDER BY
    title
FETCH FIRST ROW ONLY;
```

```
SELECT
    film_id,
    title
FROM   film
ORDER BY
title
OFFSET 5 ROWS
FETCH FIRST 5 ROW ONLY;
```

SELECT value IN

Qiymat ro`yxatdagi biron bir qiymatga mos kelishini tekshirish uchun **IN** operatoridan foydalaniladi.

```
value IN (value1,value2,...)
```

```
value IN (SELECT column_name FROM table_name);
```

PDP ACADEMY

SELECT value IN

```
SELECT
          rental_id,
          customer_id,
          return_date
FROM
          rental
WHERE
          customer_id = 1
OR        customer_id = 2
ORDER BY
          return_date DESC;
```

```
SELECT
          customer_id,
          rental_id,
          return_date
FROM
          rental
WHERE
          customer_id IN (1, 2)
ORDER BY
          return_date DESC;
```

SELECT value IN

```
SELECT
            customer_id,
            rental_id,
            return_date
FROM
            rental
WHERE
            customer_id <> 1
AND customer_id <> 2;
```

```
SELECT
            customer_id,
            rental_id,
            return_date
FROM
            rental
WHERE
            customer_id NOT IN (1, 2);
```

SELECT value BETWEEN

Biror qiymatni qiymatlar oralig`iga moslashtirish uchun BETWEEN operatoridan foydalaniladi .
Quyida BETWEEN operatorning sintaksisi tasvirlangan :

```
value BETWEEN low AND high;
```

```
value < low OR value > high
```

```
value >= low and value <= high
```

```
value NOT BETWEEN low AND high;
```

SELECT value BETWEEN

```
SELECT
        customer_id,
        payment_id,
        amount
FROM
        payment
WHERE
        amount BETWEEN 8 AND 9;
```

```
SELECT
        customer_id,
        payment_id,
        amount
FROM
        payment
WHERE
        amount NOT BETWEEN 8 AND 9;
```

SELECT value BETWEEN

```
SELECT
            customer_id,
            payment_id,
            amount,
            payment_date
FROM
            payment
WHERE
            payment_date BETWEEN '2007-02-07‘AND '2007-02-15';
```

SELECT LIKE

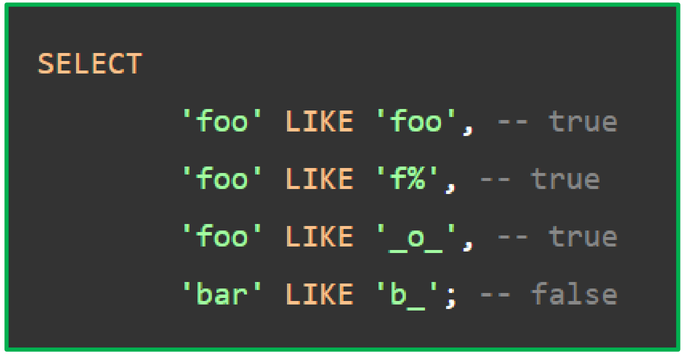
PostgreSQLda **LIKE** operatori turli xil mosliklar yordamida ma`lumotlarni so`rash uchun ishlatiladi .

```
value LIKE pattern
```

```
value NOT LIKE pattern
```

- Foiz belgisi (`%`) har qanday nol yoki undan ortiq belgilar ketma-ketligiga mos keladi.
- Pastki chiziq belgisi (`_`) har qanday bitta belgiga mos keladi.

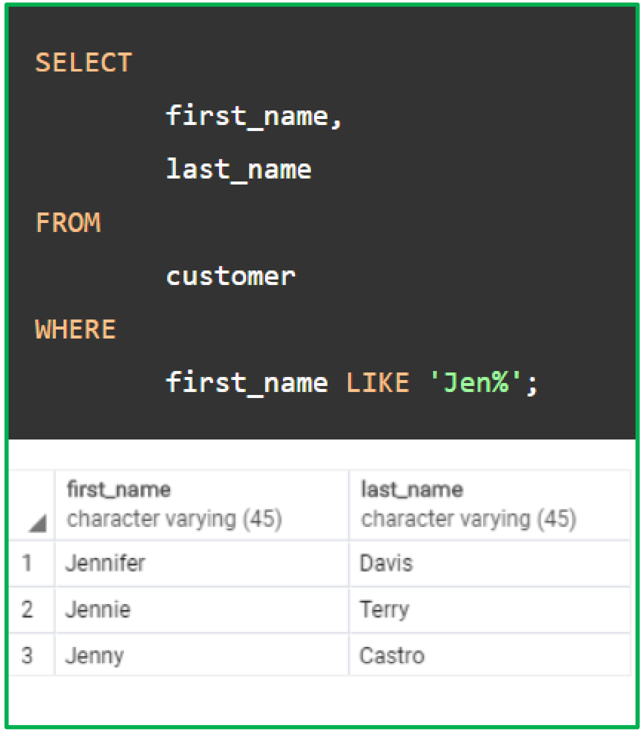
SELECT LIKE



```
SELECT
        'foo' LIKE 'foo', -- true
        'foo' LIKE 'f%', -- true
        'foo' LIKE '_o_', -- true
        'bar' LIKE 'b_'; -- false
```

```
SELECT
first_name,
        last_name
FROM
customer
WHERE
first_name LIKE '%er%'
```

SELECT LIKE



```
SELECT
        first_name,
        last_name
FROM
        customer
WHERE
        first_name LIKE 'Jen%';
```

	first_name character varying (45)	last_name character varying (45)
1	Jennifer	Davis
2	Jennie	Terry
3	Jenny	Castro

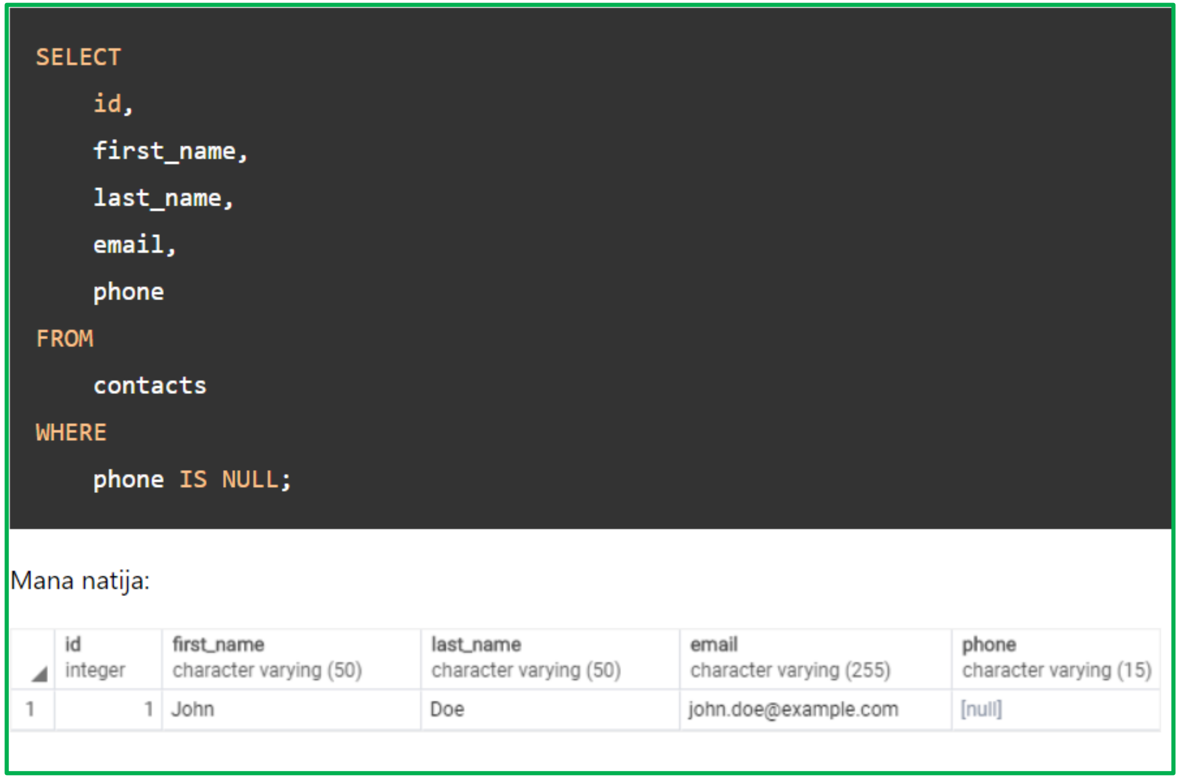
IS NULL

PostgreSQLda qiymat NULL yoki NULL EMAS ligini tekshirish uchun IS NULL operatoridan foydalaniladi.

```
value IS NULL
```

```
value IS NOT NULL
```

IS NULL



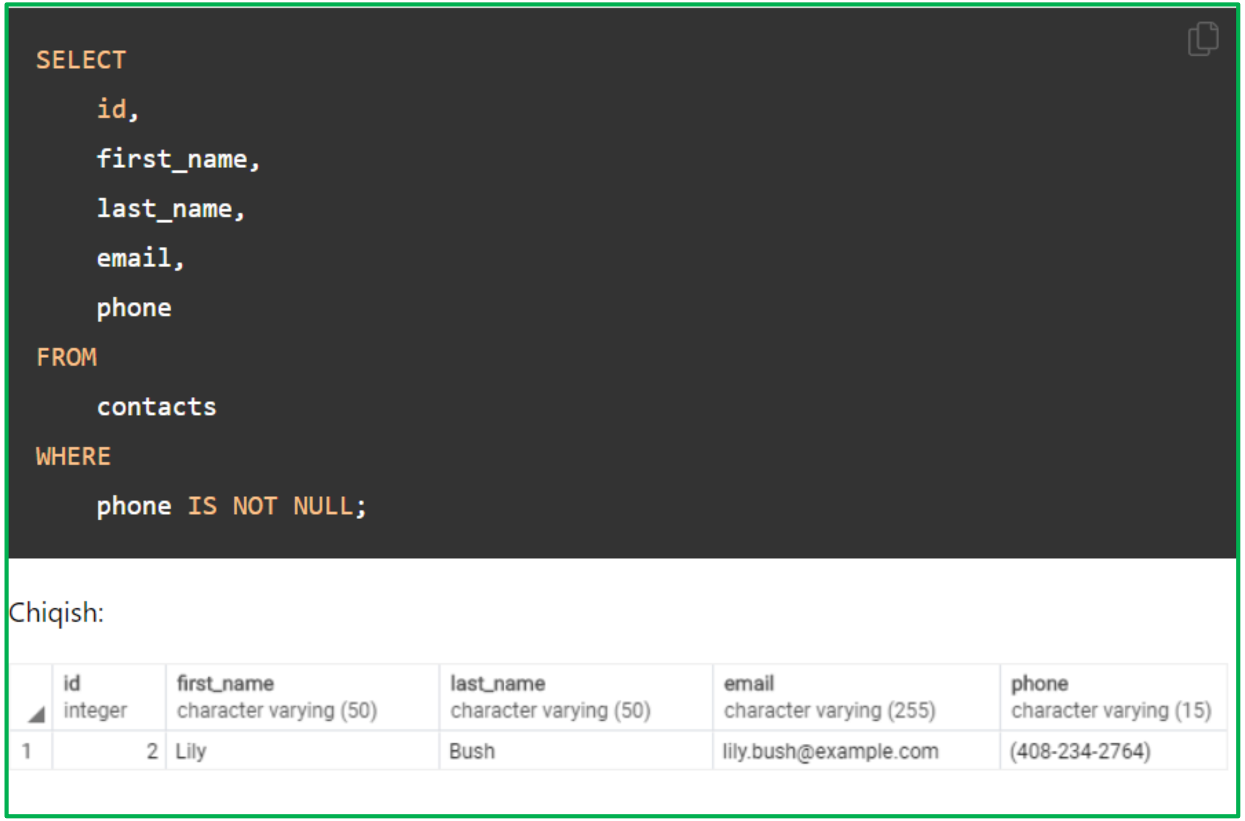
```
SELECT
    id,
    first_name,
    last_name,
    email,
    phone
FROM
    contacts
WHERE
    phone IS NULL;
```

Mana natija:

	id integer	first_name character varying (50)	last_name character varying (50)	email character varying (255)	phone character varying (15)
1	1	John	Doe	john.doe@example.com	[null]

PDP ACADEMY

IS NOT NULL



```
SELECT
    id,
    first_name,
    last_name,
    email,
    phone
FROM
    contacts
WHERE
    phone IS NOT NULL;
```

Chiqish:

	id	first_name	last_name	email	phone
	integer	character varying (50)	character varying (50)	character varying (255)	character varying (15)
1	2	Lily	Bush	lily.bush@example.com	(408-234-2764)

TABLE ALIASES

PostgreSQLda jadvalga taxallus berish uchun **AS** operatori ishlatiladi. Jadval taxallusi so`rovni bajarish vaqtida jadvallarga vaqtincha yangi nom beradi.
Quyida jadval taxallusining sintaksisi ko'rsatilgan:

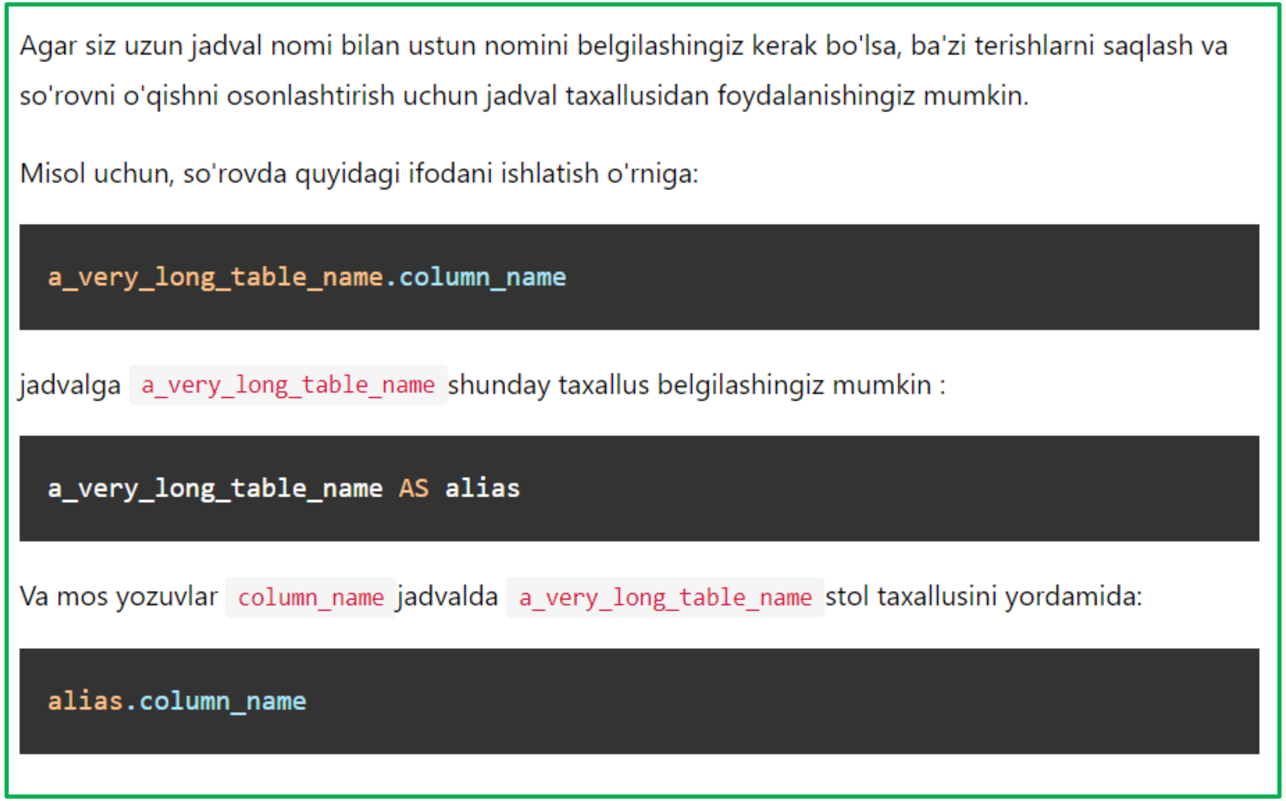
```
table_name AS alias_name;
```

TABLE ALIASES

```
SELECT    column_list
FROM   table_name AS alias_name;



SELECT t.column_name
FROM a_very_long_table_name t;
```

PDP ACADEMY

COLUMN ALIASES



Agar siz uzun jadval nomi bilan ustun nomini belgilashingiz kerak bo'lsa, ba'zi terishlarni saqlash va so'rovni o'qishni osonlashtirish uchun jadval taxallusidan foydalanishingiz mumkin.

Misol uchun, so'rovda quyidagi ifodani ishlatish o'rniga:

```
a_very_long_table_name.column_name
```

jadvalga `a_very_long_table_name` shunday taxallus belgilashingiz mumkin :

```
a_very_long_table_name AS alias
```

Va mos yozuvlar `column_name` jadvalda `a_very_long_table_name` stol taxallusini yordamida:

```
alias.column_name
```

COLUMN ALIASES

```
SELECT column_name AS alias_name
FROM table;


SELECT column_name alias_name
FROM table;


SELECT expression alias_name
FROM table;
```

PDP ACADEMY

COLUMN ALIASES

```
SELECT
    first_name || ' ' || last_name
FROM
    customer
ORDER BY
    first_name || ' ' || last_name;
```

```
SELECT
    first_name || ' ' || last_name AS full_name
FROM
    customer
ORDER BY
    full_name;
```

PDP ACADEMY

E`TIBORINGIZ UCHUN RAHMAT!