



Postgre**SQL**

# Лекция 2

# Фильтрация записей

# Фильтрация исходных записей - WHERE

```
SELECT ...
```

```
...
```

```
WHERE <условие фильтрации строк>
```

- Содержит логическое условие
  - Записи, для которых условие возвращает **TRUE** - попадают в результирующую выборку
  - Записи, для которых условие возвращает **FALSE** или **UNKNOWN** - отфильтровываются
- В предложении WHERE не доступны псевдонимы столбцов
- Данные фильтруются на стороне сервера
  - Оптимизация за счет использования индексов
  - Снижение нагрузки на сеть и использование памяти на клиенте

# Операторы сравнения

Оператор	Описание	Пример
< > = <= >= <> !< !> !=	Операторы сравнения	Name != 'Vasia' или Name <> 'Vasia'
~ и !~	Проверка соответствия строки регулярному выражению POSIX с учетом регистра	'Thomas' ~ '.*thomas.*' → False
~* и !~*	Проверка соответствия строки регулярному выражению POSIX без учёта регистра	'Thomas' ~* '.*thomas' → True

<https://postgrespro.ru/docs/postgresql/14/functions-matching#FUNCTIONS-POSIX-REGEXP>

# Предикативные операторы

Оператор	Описание	Пример
BETWEEN / NOT BETWEEN	Проверка по диапазону	orderdate <b>BETWEEN</b> '2006-07-05' <b>AND</b> '2006-07-25'
IN / NOT IN	Проверка на основе списка	Price <b>IN</b> (50, 125, 253, 264)
LIKE / NOT LIKE	Сравнение строк по маске с учетом регистра	City <b>LIKE</b> 'London' → True City <b>LIKE</b> 'london' → False
ILIKE / NOT ILIKE	Сравнение строк по маске без учета регистра	City <b>iLIKE</b> 'London' → True City <b>iLIKE</b> 'london' → True
SIMILAR TO/ NOT SIMILAR TO	Сравнение строк по шаблону на основе регулярных выражений в стандарте SQL	'abc' <b>SIMILAR TO</b> '%(b d)%' → True
IS NULL / IS NOT NULL	Проверка на наличие /отсутствие NULL значений	region <b>IS NOT NULL</b>
IS DISTINCT FROM / IS NOT DISTINCT FROM	Проверка на неравенство/ равенство заданному значению. При этом NULL воспринимается как обычное значение	region <b>IS DISTINCT FROM</b> 'WA' (все регионы, в том числе NULL, кроме WA)

<https://postgrespro.ru/docs/postgresql/14/functions-comparison>

# Использование операторов сравнения

- Простая фильтрация

Операторы сравнения

=, >, <, >=, <=, <>, !=, !>, !<

```
SELECT productname, unitprice
FROM "Production"."Products" p
WHERE discontinued !=1::bit;
```

	productname	unitprice
1	Product HHYDP	18.0000
2	Product RECZE	19.0000
3	Product IMEHJ	10.0000
4	Product KSBRM	22.0000
5	Product VAIIV	25.0000
6	Product HMLNI	30.0000
7	Product WVJFP	40.0000

```
SELECT productname, unitprice
FROM "Production"."Products" p
WHERE unitprice >= 50::money;
```

	productname	unitprice
1	Product AOZBW	97.0000
2	Product CKEDC	62.5000
3	Product QHFFP	81.0000
4	Product VJXYN	123.7900
5	Product QDOMO	263.5000
6	Product APITJ	53.0000
7	Product UKXRI	55.0000

# Фильтрация с использованием логических операторов

- Приоритет логических операторов – **NOT, AND, OR**
- Если два оператора в выражении имеют один и тот же уровень приоритета, они вычисляются в порядке слева направо по мере их появления в выражении
- Чтобы изменить приоритет операторов в выражении, следует использовать скобки

```
SELECT *
FROM "Production"."Products" p
WHERE categoryid = 1 OR categoryid = 2 AND unitprice >= 40::money;
```

	productid	productname	supplierid	categoryid	unitprice	discontinued
1	1	Product HHYDP	1	1	18.0000	0
2	2	Product RECZE	1	1	19.0000	0
3	8	Product WVJFP	3	2	40.0000	0
4	24	Product QOGNU	10	1	4.5000	1

# Фильтрация NULL значений

- NULL значения используются для маркировки отсутствующих значений (missing values)
- Для корректной обработки необходимо использовать предикаты:
  - IS NULL (ISNULL)** или **IS NOT NULL (NOTNULL)**
  - IS DISTINCT FROM (!=)** или **IS NOT DISTINCT FROM (=)**

```
SELECT companyname, region
FROM "Sales"."Customers" с
WHERE region IS NULL;
```

	companyname	region
1	Customer NRZBB	[NULL]
2	Customer MLTDN	[NULL]
3	Customer KBUDE	[NULL]
4	Customer HFBZG	[NULL]
5	Customer HGVLZ	[NULL]
6	Customer XHXJV	[NULL]

```
SELECT companyname, region
FROM "Sales"."Customers" с
WHERE region IS DISTINCT FROM 'WA';
```

Все  
регионы  
кроме WA

	companyname	region
25	Customer LCYBZ	OR
26	Customer NLTYP	MT
27	Customer YQQWWSP	
28	Customer SRQVM	SP
29	Customer NRZBB	[NULL]
30	Customer MLTDN	[NULL]
31	Customer KBUDE	[NULL]

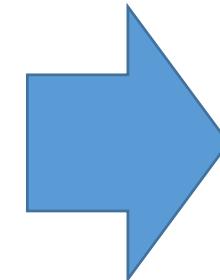
# Проверка на принадлежность диапазону

`test_expression [ NOT ] BETWEEN begin_expression AND end_expression`

- Границы диапазона включены
- Для задания исключающего диапазона используйте операторы "больше" (>) и "меньше" (<)
- Если любой параметр предиката **BETWEEN** или **NOT BETWEEN** имеет значение `NULL`, результат не определен (`UNKNOWN`)

```
SELECT productname, unitprice
FROM "Production"."Products" p
WHERE unitprice BETWEEN 30::money AND 38::money;
```

```
SELECT productname, unitprice
FROM "Production"."Products" p
WHERE unitprice >= 30::money
  AND unitprice <= 38::money;
```



	productname	unitprice
1	Product HMLNI	\$30.00
2	Product YHXGE	\$31.00
3	Product OSFNS	\$38.00
4	Product HLGZA	\$31.23
5	Product NUNAW	\$32.00
6	Product BKGEA	\$32.80
7	Product VKCMF	\$38.00
8	Product WHBYK	\$34.00

# Проверка на принадлежность множеству

`test_expression [ NOT ] IN ( subquery | expression [ ,...n ] )`

- Определяет, совпадает ли указанное значение с одним из значений, содержащихся во вложенном запросе или списке
- Использование значений NULL с предикатами **IN** или **NOT IN** может привести к непредвиденным результатам

```
SELECT productname, unitprice
FROM "Production"."Products" p
WHERE unitprice IN ( 22::money, 30::money, 32::money,
                      38::money);
```

```
SELECT productname, unitprice
FROM "Production"."Products" p
WHERE unitprice = 22::money OR unitprice = 30::money
      OR unitprice = 32::money OR unitprice = 38::money;
```



	productname	unitprice
1	Product KSBRM	\$22.00
2	Product HMLNI	\$30.00
3	Product OSFNS	\$38.00
4	Product NUNAW	\$32.00
5	Product VKCMF	\$38.00

# Проверка на соответствие SQL шаблону

match\_expression [ NOT ] **LIKE** pattern [ **ESCAPE** escape\_character ] – регистра-чувствителен  
match\_expression [ NOT ] **ILIKE** pattern [ **ESCAPE** escape\_character ] – регистра-нечувствителен

- Символы шаблона

_	любой один символ	<b>WHERE LName LIKE '_етров'</b>
%	любая строка, содержащая ноль или более символов	<b>WHERE BookName LIKE '%компьютер%'</b> <b>WHERE BookName ILIKE '%Компьютер%'</b>
\	отключает (экранирует) специальное значение любого из этих метасимволов	<b>WHERE c1 LIKE '%10-15%\% off%</b>
<b>ESCAPE</b> 'символ'	символ, помещаемый перед символом-шаблоном, чтобы символ-шаблон (%, _, [, ]) рассматривался как обычный символ	<b>WHERE c1 LIKE '%10-15!% off%' ESCAPE '!' </b>

<https://postgrespro.ru/docs/postgresql/16/functions-matching#FUNCTIONS-LIKE>

# Фильтрация по шаблону

```
SELECT    custid
            , contactname
            , contacttitle
FROM "Sales"."Customers" c
WHERE contacttitle LIKE '%Manager%';
```

	custid	contactname	contacttitle
1	10	Bassols, Pilar Colome	Accounting Manager
2	32	Krishnan, Venky	Marketing Manager
3	34	Cohen, Shy	Accounting Manager
4	38	Lee, Frank	Marketing Manager
5	43	Deshpande, Anu	Marketing Manager
6	46	Dressler, Marlies	Accounting Manager
7	48	Szymczak, Radosław	Sales Manager
8	61	Florczyk, Krzysztof	Accounting Manager

```
SELECT    custid
            , contactname
            , contacttitle
FROM "Sales"."Customers" c
WHERE contactname LIKE 'C%'
        OR contactname LIKE 'L%';
```

	custid	contactname	contacttitle
1	31	Cheng, Yao-Qiang	Sales Associate
2	34	Cohen, Shy	Accounting Manager
3	35	Langohr, Kris	Sales Representative
4	37	Crăciun, Ovidiu V.	Sales Associate
5	38	Lee, Frank	Marketing Manager
6	47	Lupu, Cornel	Owner
7	88	Li, Yan	Sales Manager
8	25	Carlson, Jason	Marketing Manager

# Сопоставление с началом строки

Использование функции **starts\_with**

```
SELECT    custid
            , contactname
            , contacttitle
FROM "Sales"."Customers" c
WHERE starts_with(contactname, 'L');
```

	custid	contactname	contacttitle
1	35	Langohr, Kris	Sales Representative
2	38	Lee, Frank	Marketing Manager
3	47	Lupu, Cornel	Owner
4	88	Li, Yan	Sales Manager
5	41	Litton, Tim	Sales Manager
6	44	Louverdis, George	Sales Representative
7	87	Ludwig, Michael	Accounting Manager

Использование оператор проверки префикса **^@**

```
SELECT    custid
            , contactname
            , contacttitle
FROM "Sales"."Customers" c
WHERE contactname ^@ 'C'
        OR contactname ^@ 'L';
```

	custid	contactname	contacttitle
1	31	Cheng, Yao-Qiang	Sales Associate
2	34	Cohen, Shy	Accounting Manager
3	35	Langohr, Kris	Sales Representative
4	37	Crăciun, Ovidiu V.	Sales Associate
5	38	Lee, Frank	Marketing Manager
6	47	Lupu, Cornel	Owner
7	88	Li, Yan	Sales Manager
8	25	Carlson, Jason	Marketing Manager

# Проверка на соответствие регулярному выражению стандарта SQL

строка **SIMILAR TO** шаблон [ **ESCAPE** спецсимвол ]

строка **NOT SIMILAR TO** шаблон [ **ESCAPE** спецсимвол ]

- возвращает **true** или **false** в зависимости от того, соответствует ли данная строка шаблону или нет
  - шаблоны соответствуют определению регулярных выражений в стандарте SQL
  - регулярные выражения SQL представляют собой гибрид синтаксиса LIKE с синтаксисом обычных регулярных выражений POSIX
- Условие **SIMILAR TO** истинно, только если шаблон соответствует всей строке
  - это отличается от условий с регулярными выражениями, в которых шаблон может соответствовать любой части строки

# Символы регулярного выражения стандарта SQL

-	любой один символ
%	любое количество любых символов
[...]	любой одиночный символ в диапазоне или наборе
[^...]	любой символ, кроме указанных в диапазоне или наборе
*	повторение предыдущего элемента 0 и более раз
+	повторение предыдущего элемента 1 и более раз
?	вхождение предыдущего элемента 0 или 1 раз
{m} {m,} {m,n}	повторение предыдущего элемента ровно {m раз}   {m или более раз}   {от m до n раз}
()	объединение нескольких элементов в одну логическую группу
	выбор (одного из двух вариантов)
\	отключает специальное значение любого из этих метасимволов

[https://docs.tantorlabs.ru/tdb/ru/17\\_7/be/functions-matching.html](https://docs.tantorlabs.ru/tdb/ru/17_7/be/functions-matching.html)

# Фильтрация с помощью SIMILAR TO

```
SELECT custid
      , contactname
      , contacttitle
FROM "Sales"."Customers" c
WHERE contactname SIMILAR to 'S[im]%' ;
```

	custid	contactname	contacttitle
1	33	Sigurdarson, Hallur	Owner
2	36	Smith, Denise	Sales Representative
3	89	Smith Jr., Ronaldo	Owner

```
SELECT custid
      , contactname
      , contacttitle
FROM "Sales"."Customers" c
WHERE contactname SIMILAR to '[^B-R]' ;
```

	custid	contactname	contacttitle
1	33	Sigurdarson, Hallur	Owner
2	36	Smith, Denise	Sales Representative
3	42	Steiner, Dominik	Marketing Assistant
4	45	Sunkammurali, Krishna	Owner
5	48	Szymczak, Radosław	Sales Manager
6	51	Taylor, Maurice	Marketing Assistant
7	75	Wojciechowska, Agnieszka	Sales Manager
8	78	Young, Robin	Marketing Assistant
9	82	Veninga, Tjeerd	Sales Associate

# Операторы регулярных выражений POSIX

Оператор	Описание	Пример
<code>~</code>	Проверка соответствия строки регулярному выражению с учётом регистра	<code>'thomas' ~ 't.*ma' → true</code>
<code>~*</code>	Проверка соответствия строки регулярному выражению без учёта регистра	<code>'thomas' ~* 'T.*ma' → true</code>
<code>!~</code>	Проверка несоответствия строки регулярному выражению с учётом регистра	<code>'thomas' !~ 't.*max' → true</code>
<code>!~*</code>	Проверка несоответствия строки регулярному выражению без учёта регистра	<code>'thomas' !~* 'T.*ma' → false</code>

**Шаблон может соответствовать любой части строки**

# Символы шаблона POSIX

.	любой один символ
[...]	любой одиночный символ в диапазоне или наборе
[^...]	любой символ, кроме указанных в диапазоне или наборе
*	повторение предыдущего элемента 0 и более раз
+	повторение предыдущего элемента 1 и более раз
?	вхождение предыдущего элемента 0 или 1 раз
{m}	повторение предыдущего элемента ровно m раз
{m,}	повторение предыдущего элемента m или более раз
{m,n}	повторение предыдущего элемента не менее чем m и не более чем n раз
()	объединение нескольких элементов в одну логическую группу
	выбор (одного из двух вариантов)
^   \A	привязывает шаблон к началу строки
\$   \Z	привязывает шаблон к концу строки
\m   \M	соответствует только в начале слова   только в конце слова

<https://postgrespro.ru/docs/postgresql/16/functions-matching#FUNCTIONS-POSIX-REGEXP>  
<https://regex101.com/>

# Фильтрация с использованием операторов регулярных выражений

```
SELECT region
FROM "Sales"."Customers" c
where region ~ '.*ra$';
```

	region
1	Lara
2	Táchira

```
SELECT companyname, contactname, city
FROM "Sales"."Customers" c
where contactname ~ '^[BKS].*[ek]$';
```

	companyname	contactname	city
1	Customer EEALV	Bassols, Pilar Colome	Tsawassen
2	Customer LVJSO	Smith, Denise	Elgin
3	Customer IAIJK	Steiner, Dominik	Vancouver
4	Customer DTDMN	Bueno, Janaina Burdan, Neville Madrid	