



Работа с СУБД **SQL**

Оконные функции



Проверить, идет ли запись

**Меня хорошо видно
&& слышно?**



Проверка


- настройка микрофона и аудио
- проверка работы чата





Оконные функции

Window functions / оконные функции



Применение аналитических функций на нескольких строках, в результате которого сохраняются все строки таблицы.

Window functions / оконные функции

Применение аналитических функций на нескольких строках, в результате которого сохраняются все строки таблицы.

- Агрегации вычисляются на строках, которые находятся в “плавающем” окне (window / frame).

Window functions / оконные функции

Применение аналитических функций на нескольких строках, в результате которого сохраняются все строки таблицы.

- Агрегации вычисляются на строках, которые находятся в “плавающем” окне (window / frame).
- Количество строк в результате = количество строк в изначальной таблице (в отличие от GROUP BY)

Window functions / оконные функции

Применение аналитических функций на нескольких строках, в результате которого сохраняются все строки таблицы.

- Агрегации вычисляются на строках, которые находятся в “плавающем” окне (window / frame).
- Количество строк в результате = количество строк в изначальной таблице (в отличие от GROUP BY)
- Рамки окон можно задавать в зависимости от значений в одном или нескольких столбцах

WINDOW / OVER

```
WINDOW имя_окна AS ( определение_окна ) [ , ... ]  
  [ имя_существующего_окна ]  
  [ PARTITION BY выражение [ , ... ] ]  
  [ ORDER BY выражение [ ASC | DESC | USING оператор ] [ NULLS { FIRST | LAST } ] [ , ... ] ]  
  [ предложение_рамки ]
```

рамка :

```
{ RANGE | ROWS | GROUPS } начало_рамки [ исключение_рамки ]  
{ RANGE | ROWS | GROUPS } BETWEEN начало_рамки AND конец_рамки [ исключение_рамки ]
```

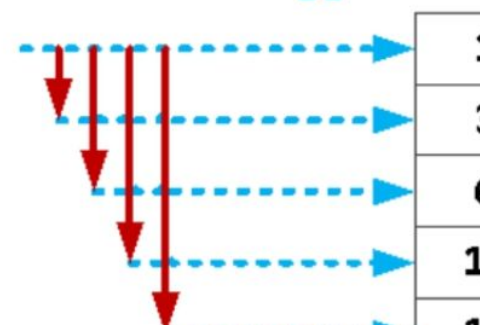
начало/конец	исключение
UNBOUNDED PRECEDING	EXCLUDE CURRENT ROW
смещение PRECEDING	EXCLUDE GROUP
CURRENT ROW	EXCLUDE TIES
смещение FOLLOWING	EXCLUDE NO OTHERS
UNBOUNDED FOLLOWING	

<https://postgrespro.ru/docs/postgresql/16/tutorial-window>

<https://postgrespro.ru/docs/postgresql/16/sql-select#SQL-WINDOW>

WINDOW / OVER

<i>i</i>	ORDER BY	agg	sum(<i>i</i>)
1			1
2			3
3			6
4			10
5			15
6			21
7			28
8			36
9			45
10			55



OVER(ORDER BY *i*)

==

OVER(ORDER BY *i*

ROWS BETWEEN

UNBOUNDED PRECEDING AND

CURRENT ROW)

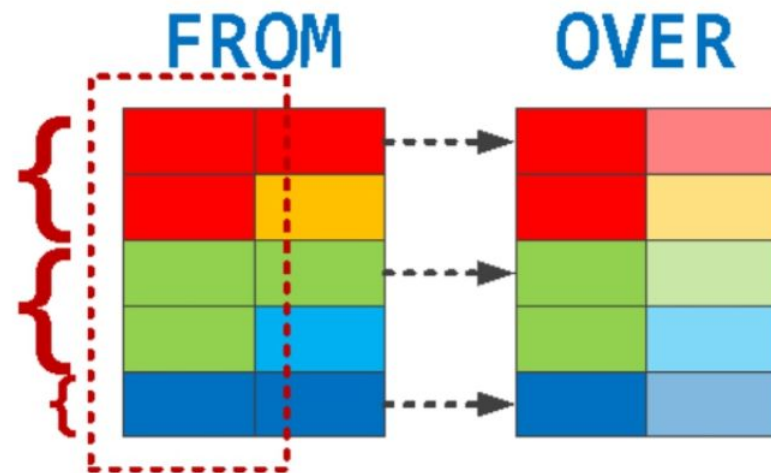
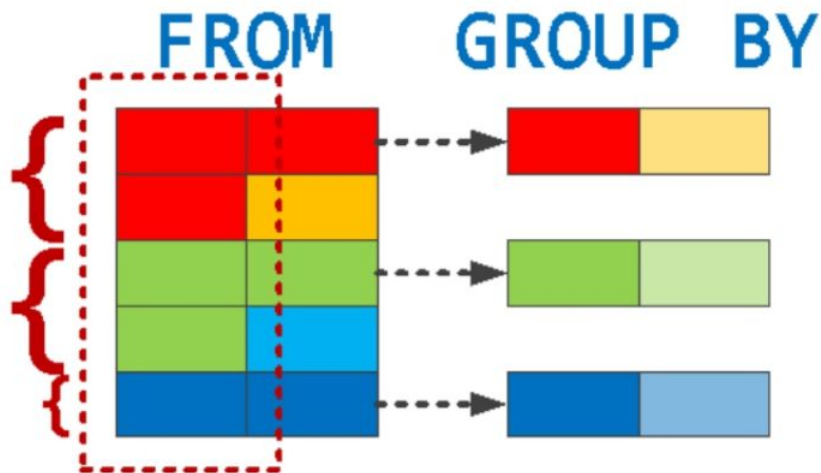
WINDOW / OVER

```
SELECT
  i
, sum(i) OVER() s
FROM
  generate_series(1, 10) i;
```

```
OVER(ORDER BY i)
==
OVER(ORDER BY i
  ROWS BETWEEN
    UNBOUNDED PRECEDING AND
    CURRENT ROW)
```

i	s
integer	bigint
1	55
2	55
3	55
4	55
5	55
6	55
7	55
8	55
9	55
10	55

Оконные функции вместо группировки



Если при группировке исходные данные теряются (группы "схлопываются" до единственной строки результата), а правила сегментирования должны быть едины для всех агрегатов, то оконные функции и исходные данные не трогают, и правила могут быть свои для каждой

Оконные функции вместо группировки

```
SELECT DISTINCT ON(i / 5) -- сегмент
    i / 5
, sum(i) OVER(PARTITION BY i / 5) s
FROM
    generate_series(0, 9) i;
```

?column?	s
integer	bigint
0	10
1	35

```
SELECT
    i / 5 -- ключ группировки
, sum(i) s
FROM
    generate_series(0, 9) i
GROUP BY
    1;
```

Оконные функции: пример 1

Подсчитать промежуточную сумму покупок (running total)

```
SELECT date,  
       sales_amount ,  
       Sum(sales_amount)  
         OVER (  
           ORDER BY date) AS running_total  
FROM   sales;
```

ID	sales_amount	date
1	102	2012-10-23
2	47	2012-10-27
3	18	2012-11-05
4	20	2012-11-14
5	160	2012-12-03

Оконные функции: пример 1

Подсчитать промежуточную сумму покупок (running total)

```
SELECT date,  
       sales_amount ,  
       Sum(sales_amount)  
       OVER (  
         ORDER BY date) AS running_total  
FROM   sales;
```

ID	sales_amount	date
1	102	2012-10-23
2	47	2012-10-27
3	18	2012-11-05
4	20	2012-11-14
5	160	2012-12-03



sales_amount	date	running_total
102	2012-10-23	102
47	2012-10-27	149
18	2012-11-05	167
20	2012-11-14	187
160	2012-12-03	347

Оконные функции: пример 2

Подсчитать общую сумму покупок по каждому клиенту.

```
SELECT *,
        Sum(sales_amount)
          OVER (
            partition BY client) AS
sales_total
FROM   sales;
```

ID	client	sales_amount	date	sales_total
1	Петр	102	2012-10-23	262
2	Василий	47	2012-10-27	47
3	Мария	18	2012-11-05	38
4	Мария	20	2012-11-14	38
5	Петр	160	2012-12-03	262

Оконные функции

Другие примеры аналитических оконных функций:

- **ROW_NUMBER:** номер строки внутри окна;

Оконные функции

Другие примеры аналитических оконных функций:

- **ROW_NUMBER:** номер строки внутри окна;
- **RANK, DENSE_RANK:** номер строки в окне с отсортированными строками;

Оконные функции

Другие примеры аналитических оконных функций:

- **ROW_NUMBER:** номер строки внутри окна;
- **RANK, DENSE_RANK:** номер строки в окне с отсортированными строками;
- **LAG, LEAD:** Значение предыдущей или последующей строки в окне;

Оконные функции

Другие примеры аналитических оконных функций:

- **ROW_NUMBER:** номер строки внутри окна;
- **RANK, DENSE_RANK:** номер строки в окне с отсортированными строками;
- **LAG, LEAD:** Значение предыдущей или последующей строки в окне;
- **FIRST_VALUE, LAST_VALUE:** Первое или последнее значение в окне;

Оконные функции вместо рекурсии

```
WITH RECURSIVE sum(i, s) AS (  
    VALUES(1, 1)  
    UNION ALL  
    SELECT  
        i + 1  
        , s + (i + 1)  
    FROM  
        sum  
    WHERE  
        i < 10  
)  
TABLE sum;
```

```
SELECT  
    i  
    , sum(i) OVER(ORDER BY i) s -- "окно"  
FROM  
    generate_series(1, 10) i;
```

$$1 = 1 \cdot$$

$$3 = 1+2 \therefore$$

$$6 = 1 + 2 + 3 \therefore \therefore$$

$$10 = 1 + 2 + 3 + 4 \therefore \therefore \therefore$$

Совместное вычисление по разным "окнам"

```
SELECT
  i
, row_number() OVER w r
, sum(i) OVER(w
  ROWS BETWEEN 1 PRECEDING
  AND 1 FOLLOWING -- скользящее окно
  EXCLUDE CURRENT ROW
) s
FROM
  generate_series(0, 9) i
WINDOW -- тут все окна
  w AS (PARTITION BY i / 5 ORDER BY i);
```

i	r	s
integer	bigint	bigint
0	1	1 -- i / 5 == 0
1	2	2
2	3	4 -- 1(p) + 3(f)
3	4	6
4	5	3
5	1	6 -- i / 5 == 1
6	2	12
7	3	14
8	4	16
9	5	8 -- 8(p) + NULL(f)

ДЗ №5

<https://github.com/DWH-course-examples/SQL-postgres/blob/main/homework/task5.md>

СПАСИБО ЗА ВНИМАНИЕ!

#аис
#учисьваис