

Оконные функции SQL

Реляционная модель

Преподаватель: канд. тех. наук, доц.
Озерова Г.П.

Оконные функции

Оконные функции являются частью запроса SQL.

Они позволяют получить некоторую дополнительную информацию о выборке данных .

С помощью оконных функций можно реализовать вычисления для набора строк, некоторым образом связанных с текущей строкой.

При этом использование оконной функции не группирует несколько строк в одну, а сохраняет все строки запроса.

Описание окна

Синтаксис запроса с оконными функциями:

```
SELECT . . . ,  
    оконная_функция (выражение) OVER имя_окна,  
    . . .  
FROM таблица  
WINDOW имя_окна  
AS (  
    PARTITION BY столбец_1, столбец_2, . . . – это окно  
    ORDER BY . . . – сортировка  
    ROWS BETWEEN – границы окна  
    . . .  
)
```

Операции с таблицей целиком

Чтобы выполнить некоторые действия над всеми записями таблицы используется следующее описание окна:

```
WINDOW имя_окна  
AS (  
    ORDER BY ...  
)
```

Оконные функции ранжирования

Окно с разделом сортировки, позволяет:

- пронумеровать строки в таблице;
- выполнить ранжирование строк;
- и т. д.

Для этого в окне (в нашем случае во всей таблице) осуществляется сортировка записей по указанным при описании окна в разделе **ORDER BY** столбцам.

После чего к отсортированным строкам окна в разделе **SELECT** можно применить оконные функции.

Оконные функции ранжирования

ROW_NUMBER()

Возвращает номер текущей строки в окне. Номера строк начинаются с 1 и заканчиваются количеством строк в окне.

RANK()

Возвращает ранг строки, то есть при одинаковом значении строкам присваивается один и тот же номер, с пропуском номеров. Диапазон возвращаемых значений от 1 до количества строк в окне.

CUME_DIST()

Возвращает совокупное распределение значения в группе значений (окне), то есть процент значений, меньший или равный значению в текущей строке. Вычисляется как количество строк, предшествующих или равных текущей строке в окне, деленное на общее количество строк в окне. Диапазон возвращаемых значений от 0 до 1.

PROCENT_RANK()

Возвращает процент значений окна меньше значения в текущей строке, исключая максимальное значение.

Возвращаемые значения находятся в диапазоне от 0 до 1 и представляют относительный ранг строки, вычисленный по формуле:

$$(\text{rank} - 1) / (\text{rows} - 1)$$

где **rank**- ранг строки,

rows - количество строк окна.

Оконные функции ранжирования

Пример. Дана таблица **test**. Применим оконные функции нумерации, ранжирования, распределения и процентного ранжирования, построив окно по всей таблице, отсортировав данные по столбцу **val**.

val
22
12
16
5
5
12
5
23
7
22

Оконные функции ранжирования

Пример. Дана таблица **test**. Применим оконные функции нумерации, ранжирования, распределения и процентного ранжирования, построив окно по всей таблице, отсортировав данные по столбцу **val**.

```
SELECT val,  
       ROW_NUMBER() OVER win_test AS Номер,  
       RANK() OVER win_test AS Ранг,  
       CUME_DIST() OVER win_test AS Распределение,  
       PERCENT_RANK() OVER win_test AS Процентный_ранг  
FROM test  
WINDOW win_test  
AS (  
    ORDER BY val  
);
```

Оконные функции ранжирования

Пример. Дана таблица **test**. Применим оконные функции **нумерации**, ранжирования, распределения и процентного ранжирования, построив окно по всей таблице, отсортировав данные по столбцу **val**.

val	Номер	Ранг	Распределение	Процентный ранг
5	1			
5	2			
5	3			
7	4			
12	5			
12	6			
16	7			
22	8			
22	9			
23	10			

Оконные функции ранжирования

Пример. Дана таблица **test**. Применим оконные функции нумерации, **ранжирования**, распределения и процентного ранжирования, построив окно по всей таблице, отсортировав данные по столбцу **val**.

val	Номер	Ранг	Распределение	Процентный ранг
5	1	1		
5	2	1		
5	3	1		
7	4	4		
12	5	5		
12	6	5		
16	7	7		
22	8	8		
22	9	8		
23	10	10		

Оконные функции ранжирования

Пример. Дана таблица **test**. Применим оконные функции нумерации, ранжирования, **распределения** и процентного ранжирования, построив окно по всей таблице, отсортировав данные по столбцу **val**.

val	Номер	Ранг	Распределение	Процентный ранг
5	1	1		
5	2	1		
5	3	1		
7	4	4		
12	5	5		
12	6	5		
16	7	7		
22	8	8		
22	9	8		
23	10	10		

Оконные функции ранжирования

Пример. Дана таблица **test**. Применим оконные функции нумерации, ранжирования, **распределения** и процентного ранжирования, построив окно по всей таблице, отсортировав данные по столбцу **val**.

val	Номер	Ранг	Распределение	Процентный ранг
5	1	1		
5	2	1		
5	3	1		
7	4	4		
12	5	5		
12	6	5		
16	7	7		
22	8	8		
22	9	8		
23	10	10		

Оконные функции ранжирования

Пример. Дана таблица **test**. Применим оконные функции нумерации, ранжирования, **распределения** и процентного ранжирования, построив окно по всей таблице, отсортировав данные по столбцу **val**.

val	Номер	Ранг	Распределение	Процентный ранг
5	1	1	3/10 = 0.3	
5	2	1		
5	3	1		
7	4	4		
12	5	5		
12	6	5		
16	7	7		
22	8	8		
22	9	8		
23	10	10		

Оконные функции ранжирования

Пример. Дана таблица **test**. Применим оконные функции нумерации, ранжирования, **распределения** и процентного ранжирования, построив окно по всей таблице, отсортировав данные по столбцу **val**.

val	Номер	Ранг	Распределение	Процентный ранг
5	1	1	0.3	
5	2	1	0.3	
5	3	1	0.3	
7	4	4	0.4	
12	5	5	0.6	
12	6	5	0.6	
16	7	7	0.7	
22	8	8	0.9	
22	9	8	0.9	
23	10	10	1.0	

Оконные функции ранжирования

Пример. Дана таблица **test**. Применим оконные функции нумерации, ранжирования, распределения и **процентного ранжирования**, построив окно по всей таблице, отсортировав данные по столбцу **val**.

val	Номер	Ранг	Распределение	Процентный ранг
5	1	1	0.3	0.0
5	2	1	0.3	0.0
5	3	1	0.3	0.0
7	4	4	0.4	
12	5	5	0.6	
12	6	5	0.6	
16	7	7	0.7	
22	8	8	0.9	
22	9	8	0.9	
23	10	10	1.0	

Оконные функции ранжирования

Пример. Дана таблица **test**. Применим оконные функции нумерации, ранжирования, распределения и **процентного ранжирования**, построив окно по всей таблице, отсортировав данные по столбцу **val**.

val	Номер	Ранг	Распределение	Процентный ранг
5	1	1	0.3	0.0
5	2	1	0.3	0.0
5	3	1	0.3	0.0
7	4	4	0.4	
12	5	5	0.6	
12	6	5	0.6	
16	7	7	0.7	
22	8	8	0.9	
22	9	8	0.9	
23	10	10	1.0	

Оконные функции ранжирования

Пример. Дана таблица **test**. Применим оконные функции нумерации, ранжирования, распределения и **процентного ранжирования**, построив окно по всей таблице, отсортировав данные по столбцу **val**.

val	Номер	Ранг	Распределение	Процентный ранг
5	1	1	0.3	0.0
5	2	1	0.3	0.0
5	3	1	0.3	0.0
7	4	4	0.4	
12	5	5	0.6	
12	6	5	0.6	
16	7	7	0.7	
22	8	8	0.9	
22	9	8	0.9	
23	10	10	1.0	

Оконные функции ранжирования

Пример. Дана таблица **test**. Применим оконные функции нумерации, ранжирования, распределения и **процентного ранжирования**, построив окно по всей таблице, отсортировав данные по столбцу **val**.

val	Номер	Ранг	Распределение	Процентный ранг
5	1	1	0.3	0.0
5	2	1	0.3	0.0
5	3	1	0.3	0.0
7	4	4	0.4	$(4-1)/(10-1)=0.33$
12	5	5	0.6	
12	6	5	0.6	
16	7	7	0.7	
22	8	8	0.9	
22	9	8	0.9	
23	10	10	1.0	

Оконные функции ранжирования

Пример. Дана таблица **test**. Применим оконные функции нумерации, ранжирования, распределения и **процентного ранжирования**, построив окно по всей таблице, отсортировав данные по столбцу **val**.

val	Номер	Ранг	Распределение	Процентный ранг
5	1	1	0.3	0.0
5	2	1	0.3	0.0
5	3	1	0.3	0.0
7	4	4	0.4	0.333333
12	5	5	0.6	0.444444
12	6	5	0.6	0.444444
16	7	7	0.7	0.666666
22	8	8	0.9	0.777777
22	9	8	0.9	0.777777
23	10	10	1.0	1.0

Оконные функции (первое, последнее...)

FIRST_VALUE(столбец)

Возвращает значение столбца, расположенного в первой строке(записи) окна .

LAST_VALUE(столбец)

Возвращает значение указанного столбца, расположенного в последней строке части окна, ограниченной началом окна и последней записью, в которой значение столбца, по которому осуществляется сортировка, равно текущему значению этого столбца.

NTN_VALUE(столбец, номер)

Возвращает значение столбца, расположенного в строке с указанным номером, той части окна которая ограничена началом окна и текущей записью. Если в этой части окна записи с указанным номером нет, возвращается **NULL**.

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

shape	val
шар	22
куб	12
куб	16
шар	5
шар	5
конус	12
куб	5
куб	23
куб	7
конус	22

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

```
SELECT val, shape,  
    FIRST_VALUE(shape) OVER win_test AS Первое_shape,  
    LAST_VALUE(val) OVER win_test AS Последнее_val,  
    LAST_VALUE(shape) OVER win_test AS Последнее_shape,  
    NTH_VALUE(val, 5) OVER win_test AS Пятое_shape  
FROM test  
WINDOW win_test  
AS(  
    ORDER BY val  
);
```

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5				
шар	5				
куб	5				
куб	7				
куб	12				
конус	12				
куб	16				
шар	22				
конус	22				
куб	23				

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5				
шар	5				
куб	5				
куб	7				
куб	12				
конус	12				
куб	16				
шар	22				
конус	22				
куб	23				

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем **первое значение столбца shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5	шар			
шар	5	шар			
куб	5	шар			
куб	7	шар			
куб	12	шар			
конус	12	шар			
куб	16	шар			
шар	22	шар			
конус	22	шар			
куб	23	шар			

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5	шар			
шар	5	шар			
куб	5	шар			
куб	7	шар			
куб	12	шар			
конус	12	шар			
куб	16	шар			
шар	22	шар			
конус	22	шар			
куб	23	шар			

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5	шар	5		
шар	5	шар			
куб	5	шар			
куб	7	шар			
куб	12	шар			
конус	12	шар			
куб	16	шар			
шар	22	шар			
конус	22	шар			
куб	23	шар			

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5	шар	5		
шар	5	шар	5		
куб	5	шар	5		
куб	7	шар			
куб	12	шар			
конус	12	шар			
куб	16	шар			
шар	22	шар			
конус	22	шар			
куб	23	шар			

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5	шар	5		
шар	5	шар	5		
куб	5	шар	5		
куб	7	шар	7		
куб	12	шар			
конус	12	шар			
куб	16	шар			
шар	22	шар			
конус	22	шар			
куб	23	шар			

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5	шар	5		
шар	5	шар	5		
куб	5	шар	5		
куб	7	шар	7		
куб	12	шар	12		
конус	12	шар	12		
куб	16	шар	16		
шар	22	шар	22		
конус	22	шар	22		
куб	23	шар	23		

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5	шар	5		
шар	5	шар	5		
куб	5	шар	5		
куб	7	шар	7		
куб	12	шар	12		
конус	12	шар	12		
куб	16	шар	16		
шар	22	шар	22		
конус	22	шар	22		
куб	23	шар	23		

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5	шар	5		
шар	5	шар	5		
куб	5	шар	5		
куб	7	шар	7		
куб	12	шар	12		
конус	12	шар	12		
куб	16	шар	16		
шар	22	шар	22		
конус	22	шар	22		
куб	23	шар	23		

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5	шар	5	куб	
шар	5	шар	5	куб	
куб	5	шар	5	куб	
куб	7	шар	7		
куб	12	шар	12		
конус	12	шар	12		
куб	16	шар	16		
шар	22	шар	22		
конус	22	шар	22		
куб	23	шар	23		

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5	шар	5	куб	
шар	5	шар	5	куб	
куб	5	шар	5	куб	
куб	7	шар	7	куб	
куб	12	шар	12		
конус	12	шар	12		
куб	16	шар	16		
шар	22	шар	22		
конус	22	шар	22		
куб	23	шар	23		

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5	шар	5	куб	
шар	5	шар	5	куб	
куб	5	шар	5	куб	
куб	7	шар	7	куб	
куб	12	шар	12		
конус	12	шар	12		
куб	16	шар	16		
шар	22	шар	22		
конус	22	шар	22		
куб	23	шар	23		

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5	шар	5	куб	
шар	5	шар	5	куб	
куб	5	шар	5	куб	
куб	7	шар	7	куб	
куб	12	шар	12	Конус	
конус	12	шар	12	конус	
куб	16	шар	16		
шар	22	шар	22		
конус	22	шар	22		
куб	23	шар	23		

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, 5-ое значение столбца **val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5	шар	5	куб	
шар	5	шар	5	куб	
куб	5	шар	5	куб	
куб	7	шар	7	куб	
куб	12	шар	12	конус	
конус	12	шар	12	конус	
куб	16	шар	16	куб	
шар	22	шар	22	конус	
конус	22	шар	22	конус	
куб	23	шар	23	куб	

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, **5-ое значение столбца val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5	шар	5	куб	Null
шар	5	шар	5	куб	Null
куб	5	шар	5	куб	Null
куб	7	шар	7	куб	Null
куб	12	шар	12	конус	
конус	12	шар	12	конус	
куб	16	шар	16	куб	
шар	22	шар	22	конус	
конус	22	шар	22	конус	
куб	23	шар	23	куб	

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, **5-ое значение столбца val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5	шар	5	куб	Null
шар	5	шар	5	куб	Null
куб	5	шар	5	куб	Null
куб	7	шар	7	куб	Null
куб	12	шар	12	конус	12
конус	12	шар	12	конус	
куб	16	шар	16	куб	
шар	22	шар	22	конус	
конус	22	шар	22	конус	
куб	23	шар	23	куб	

Оконные функции (первое, последнее, ...)

Пример. Дана таблица **test**. Создадим окно, в котором отсортируем все числа в таблице по возрастанию, затем в **SELECT** найдем первое значение столбца **shape**, последнее значение столбцов **val** и **shape**, **5-ое значение столбца val**.

shape	val	Первое_shape	Последнее_val	Последнее_shape	Пятое_val
шар	5	шар	5	куб	Null
шар	5	шар	5	куб	Null
куб	5	шар	5	куб	Null
куб	7	шар	7	куб	Null
куб	12	шар	12	конус	12
конус	12	шар	12	конус	12
куб	16	шар	16	куб	12
шар	22	шар	22	конус	12
конус	22	шар	22	конус	12
куб	23	шар	23	куб	12

Оконные функции (следующее, предыдущее)

LAG(столбец)

Для каждой записи возвращает значение столбца, расположенного в предыдущей записи окна. Если предыдущей записи нет (строка первая в окне), то возвращается значение **NULL**.

LEAD(столбец)

Для каждой записи возвращает значение столбца, расположенного в следующей записи окна. Если следующей записи нет (строка последняя в окне), то возвращается значение **NULL**.

Оконные функции (следующее, предыдущее)

Пример. Создадим окно, в котором отсортируем все числа в таблице **test** по возрастанию. Затем в **SELECT** найдем для каждой записи значения столбцов **val** и **shape** из предыдущей и следующей записи.

```
SELECT val, shape,  
        LAG(val) OVER win_test AS Предыдущее_val,  
        LAG(shape) OVER win_test AS Предыдущее_shape,  
        LEAD(val) OVER win_test AS Следующее_val,  
        LEAD(shape) OVER win_test AS Следующее_shape  
FROM test  
WINDOW win_test  
AS(  
    ORDER BY val  
);
```

Оконные функции (следующее, предыдущее)

Пример. Создадим окно, в котором отсортируем все числа в таблице **test** по возрастанию. Затем в **SELECT** найдем для каждой записи значения столбцов **val** и **shape** из **предыдущей** и следующей записи.

shape	val	Предыдущее_val	Предыдущее_shape	Следующее_val	Следующее_shape
шар	5				
шар	5				
куб	5				
куб	7				
куб	12				
конус	12				
куб	16				
шар	22				
конус	22				
куб	23				

Оконные функции (следующее, предыдущее)

Пример. Создадим окно, в котором отсортируем все числа в таблице **test** по возрастанию. Затем в **SELECT** найдем для каждой записи значения столбцов **val** и **shape** из предыдущей и следующей записи.

shape	val	Предыдущее_val	Предыдущее_shape	Следующее_val	Следующее_shape
шар	5	NULL	NULL		
шар	5				
куб	5				
куб	7				
куб	12				
конус	12				
куб	16				
шар	22				
конус	22				
куб	23				

Оконные функции (следующее, предыдущее)

Пример. Создадим окно, в котором отсортируем все числа в таблице **test** по возрастанию. Затем в **SELECT** найдем для каждой записи значения столбцов **val** и **shape** из **предыдущей** и следующей записи.

shape	val	Предыдущее_val	Предыдущее_shape	Следующее_val	Следующее_shape
шар	5	NULL	NULL		
шар	5				
куб	5				
куб	7				
куб	12				
конус	12				
куб	16				
шар	22				
конус	22				
куб	23				

Оконные функции (следующее, предыдущее)

Пример. Создадим окно, в котором отсортируем все числа в таблице **test** по возрастанию. Затем в **SELECT** найдем для каждой записи значения столбцов **val** и **shape** из **предыдущей** и следующей записи.

shape	val	Предыдущее_val	Предыдущее_shape	Следующее_val	Следующее_shape
шар	5	NULL	NULL		
шар	5				
куб	5				
куб	7				
куб	12				
конус	12				
куб	16				
шар	22				
конус	22				
куб	23				

Оконные функции (следующее, предыдущее)

Пример. Создадим окно, в котором отсортируем все числа в таблице **test** по возрастанию. Затем в **SELECT** найдем для каждой записи значения столбцов **val** и **shape** из **предыдущей** и следующей записи.

shape	val	Предыдущее_val	Предыдущее_shape	Следующее_val	Следующее_shape
шар	5	NULL	NULL		
шар	5	5	шар		
куб	5				
куб	7				
куб	12				
конус	12				
куб	16				
шар	22				
конус	22				
куб	23				

Оконные функции (следующее, предыдущее)

Пример. Создадим окно, в котором отсортируем все числа в таблице **test** по возрастанию. Затем в **SELECT** найдем для каждой записи значения столбцов **val** и **shape** из предыдущей и следующей записи.

shape	val	Предыдущее_val	Предыдущее_shape	Следующее_val	Следующее_shape
шар	5	NULL	NULL		
шар	5	5	шар		
куб	5	5	шар		
куб	7	5	куб		
куб	12	7	куб		
конус	12	12	куб		
куб	16	12	конус		
шар	22	16	куб		
конус	22	22	шар		
куб	23	22	конус		

Оконные функции (следующее, предыдущее)

Пример. Создадим окно, в котором отсортируем все числа в таблице **test** по возрастанию. Затем в **SELECT** найдем для каждой записи значения столбцов **val** и **shape** из предыдущей и следующей записи.

shape	val	Предыдущее_val	Предыдущее_shape	Следующее_val	Следующее_shape
шар	5	NULL	NULL		
шар	5	5	шар		
куб	5	5	шар		
куб	7	5	куб		
куб	12	7	куб		
конус	12	12	куб		
куб	16	12	конус		
шар	22	16	куб		
конус	22	22	шар		
куб	23	22	конус		

Оконные функции (следующее, предыдущее)

Пример. Создадим окно, в котором отсортируем все числа в таблице **test** по возрастанию. Затем в **SELECT** найдем для каждой записи значения столбцов **val** и **shape** из предыдущей и следующей записи.

shape	val	Предыдущее_val	Предыдущее_shape	Следующее_val	Следующее_shape
шар	5	NULL	NULL	5	шар
шар	5	5	шар		
куб	5	5	шар		
куб	7	5	куб		
куб	12	7	куб		
конус	12	12	куб		
куб	16	12	конус		
шар	22	16	куб		
конус	22	22	шар		
куб	23	22	конус		

Оконные функции (следующее, предыдущее)

Пример. Создадим окно, в котором отсортируем все числа в таблице **test** по возрастанию. Затем в **SELECT** найдем для каждой записи значения столбцов **val** и **shape** из предыдущей и следующей записи.

shape	val	Предыдущее_val	Предыдущее_shape	Следующее_val	Следующее_shape
шар	5	NULL	NULL	5	шар
шар	5	5	шар	5	куб
куб	5	5	шар	7	куб
куб	7	5	куб	12	куб
куб	12	7	куб	12	конус
конус	12	12	куб	16	Куб
куб	16	12	конус	22	шар
шар	22	16	куб	22	конус
конус	22	22	шар	23	куб
куб	23	22	конус		

Оконные функции (следующее, предыдущее)

Пример. Создадим окно, в котором отсортируем все числа в таблице **test** по возрастанию. Затем в **SELECT** найдем для каждой записи значения столбцов **val** и **shape** из предыдущей и следующей записи.

shape	val	Предыдущее_val	Предыдущее_shape	Следующее_val	Следующее_shape
шар	5	NULL	NULL	5	шар
шар	5	шар	5	5	куб
куб	5	шар	5	7	куб
куб	7	куб	5	12	куб
куб	12	куб	7	12	конус
конус	12	куб	12	16	Куб
куб	16	конус	12	22	шар
шар	22	куб	16	22	конус
конус	22	шар	22	23	куб
куб	23	конус	22	NULL	NULL

Оконные функции (следующее, предыдущее)

Пример. Создадим окно, в котором отсортируем все числа в таблице **test** по возрастанию. Затем в **SELECT** найдем для каждой записи значения столбцов **val** и **shape** из предыдущей и следующей записи.

```
SELECT val, shape,  
        IFNULL(LAG(val) OVER win_test , "-") AS Предыдущее_val,  
        IFNULL(LAG(shape) OVER win_test , "-") AS Предыдущее_shape,  
        IFNULL(LEAD(val) OVER win_test , "-") AS Следующее_val,  
        IFNULL(LEAD(shape) OVER win_test , "-") AS Следующее_shape  
FROM test  
WINDOW win_test  
AS(  
    ORDER BY val  
);
```

Оконные функции (следующее, предыдущее)

Пример. Создадим окно, в котором отсортируем все числа в таблице **test** по возрастанию. Затем в **SELECT** найдем для каждой записи значения столбцов **val** и **shape** из предыдущей и следующей записи.

shape	val	Предыдущее_val	Предыдущее_shape	Следующее_val	Следующее_shape
шар	5	-	-	5	шар
шар	5	шар	5	5	куб
куб	5	шар	5	7	куб
куб	7	куб	5	12	куб
куб	12	куб	7	12	конус
конус	12	куб	12	16	Куб
куб	16	конус	12	22	шар
шар	22	куб	16	22	конус
конус	22	шар	22	23	куб
куб	23	конус	22	-	-

Объявление окна, два раздела

Рассмотрим запрос, в котором объявление окна включает два раздела:

SELECT

...,

оконная_функция(выражение) **OVER** имя_окна,

...

FROM таблица

WINDOW имя_окна

AS(

PARTITION BY ...

ORDER BY ...

);

Объявление окна, два раздела

Такое оконное выражение позволяет выполнять некоторые действия над записями таблицы, ограниченными "**окном**". Окном считается совокупность записей, имеющих в столбцах, указанных после **PARTITION BY**, одинаковые значения.

shape	val
конус	12
конус	22
куб	16
куб	7
куб	12
куб	5
куб	23
шар	5
шар	5
шар	22

Объявление окна, два раздела

Такое оконное выражение позволяет выполнять некоторые действия над записями таблицы, ограниченными "**окном**". Окном считается совокупность записей, имеющих в столбцах, указанных после **PARTITION BY**, одинаковые значения.

shape	val
конус	12
конус	22
куб	16
куб	7
куб	12
куб	5
куб	23
шар	5
шар	5
шар	22

Объявление окна, два раздела

Такое оконное выражение позволяет выполнять некоторые действия над записями таблицы, ограниченными "**окном**". Окном считается совокупность записей, имеющих в столбцах, указанных после **PARTITION BY**, одинаковые значения.

shape	val
конус	12
конус	22
куб	16
куб	7
куб	12
куб	5
куб	23
шар	5
шар	5
шар	22

Объявление окна, два раздела

Такое оконное выражение позволяет выполнять некоторые действия над записями таблицы, ограниченными "**окном**". Окном считается совокупность записей, имеющих в столбцах, указанных после **PARTITION BY**, одинаковые значения.

shape	val
конус	12
конус	22
куб	16
куб	7
куб	12
куб	5
куб	23
шар	5
шар	5
шар	22

Объявление окна, два раздела

Для записей, принадлежащих окну, можно выполнить сортировку по значениям одного или нескольких столбцов в разделе **ORDER BY**.

shape	val
конус	12
конус	22
куб	5
куб	7
куб	12
куб	16
куб	23
шар	5
шар	5
шар	22



The diagram illustrates three overlapping windows defined by colored borders and arrows:

- Red window:** Covers the first two rows (cones). A red arrow points downwards from the top of this window.
- Green window:** Covers rows 3 through 7 (cubes). A green arrow points downwards from the top of this window.
- Blue window:** Covers the last three rows (spheres). A blue arrow points downwards from the top of this window.

Оконные функции ранжирования

Пример. Создадим окно по каждому типу геометрических фигур. В каждом окне отсортируем все числа по возрастанию. Затем в разделе **SELECT** применим оконные функции ранжирования.

```
SELECT val,  
       ROW_NUMBER() OVER win_test AS Номер,  
       RANK() OVER win_test AS Ранг,  
       CUME_DIST() OVER win_test AS Распределение,  
       PERCENT_RANK() OVER win_test AS Процентный_ранг  
FROM test  
WINDOW win_test  
AS (  
    PARTITION BY shape  
    ORDER BY val  
);
```

Оконные функции ранжирования

Пример. Создадим окно по каждому типу геометрических фигур. В каждом окне отсортируем все числа по возрастанию. Затем в разделе **SELECT** применим оконные функции ранжирования.

shape	val	Номер	Ранг	Распределение	Процентный_ранг
конус	12				
конус	22				
куб	16				
куб	16				
куб	5				
куб	23				
куб	7				
шар	22				
шар	5				
шар	5				

Оконные функции ранжирования

Пример. Создадим окно по каждому типу геометрических фигур. В каждом окне **отсортируем все числа по возрастанию**. Затем в разделе **SELECT** применим оконные функции ранжирования.

shape	val	Номер	Ранг	Распределение	Процентный_ранг
конус	12	↓			
конус	22				
куб	5	↓			
куб	7				
куб	12				
куб	16				
куб	23				
шар	5	↓			
шар	5				
шар	22				

Оконные функции ранжирования

Пример. Создадим окно по каждому типу геометрических фигур. В каждом окне отсортируем все числа по возрастанию. Затем в разделе **SELECT** применим оконные функции ранжирования.

shape	val	Номер	Ранг	Распределение	Процентный_ранг
конус	12	1	1	0.5	0.0
конус	22	2	2	1.0	1.0
куб	5	1	1	0.2	0.0
куб	7	2	2	0.4	0.25
куб	12	3	3	0.6	0.5
куб	16	4	4	0.8	0.75
куб	23	5	5	1.0	1.0
шар	5				
шар	5				
шар	22				

Оконные функции ранжирования

Пример. Создадим окно по каждому типу геометрических фигур. В каждом окне отсортируем все числа по возрастанию. Затем в разделе **SELECT** применим оконные функции ранжирования.

shape	val	Номер	Ранг	Распределение	Процентный_ранг
конус	12	1	1	0.5	0.0
конус	22	2	2	1.0	1.0
куб	5	1	1	0.2	0.0
куб	7	2	2	0.4	0.25
куб	12	3	3	0.6	0.5
куб	16	4	4	0.8	0.75
куб	23	5	5	1.0	1.0
шар	5	1			
шар	5	2			
шар	22	3			

Оконные функции ранжирования

Пример. Создадим окно по каждому типу геометрических фигур. В каждом окне отсортируем все числа по возрастанию. Затем в разделе **SELECT** применим оконные функции ранжирования.

shape	val	Номер	Ранг	Распределение	Процентный_ранг
конус	12	1	1	0.5	0.0
конус	22	2	2	1.0	1.0
куб	5	1	1	0.2	0.0
куб	7	2	2	0.4	0.25
куб	12	3	3	0.6	0.5
куб	16	4	4	0.8	0.75
куб	23	5	5	1.0	1.0
шар	5	1	1		
шар	5	2	1		
шар	22	3	3		

Оконные функции ранжирования

Пример. Создадим окно по каждому типу геометрических фигур. В каждом окне отсортируем все числа по возрастанию. Затем в разделе **SELECT** применим оконные функции ранжирования.

shape	val	Номер	Ранг	Распределение	Процентный_ранг
конус	12	1	1	0.5	0.0
конус	22	2	2	1.0	1.0
куб	5	1	1	0.2	0.0
куб	7	2	2	0.4	0.25
куб	12	3	3	0.6	0.5
куб	16	4	4	0.8	0.75
куб	23	5	5	1.0	1.0
шар	5	1	1	$2 / 3 = 0.666$	
шар	5	2	1		
шар	22	3	3		

Оконные функции ранжирования

Пример. Создадим окно по каждому типу геометрических фигур. В каждом окне отсортируем все числа по возрастанию. Затем в разделе **SELECT** применим оконные функции ранжирования.

shape	val	Номер	Ранг	Распределение	Процентный_ранг
конус	12	1	1	0.5	0.0
конус	22	2	2	1.0	1.0
куб	5	1	1	0.2	0.0
куб	7	2	2	0.4	0.25
куб	12	3	3	0.6	0.5
куб	16	4	4	0.8	0.75
куб	23	5	5	1.0	1.0
шар	5	1	1	$2 / 3 = 0.666$	
шар	5	2	1	$2 / 3 = 0.666$	
шар	22	3	3		

Оконные функции ранжирования

Пример. Создадим окно по каждому типу геометрических фигур. В каждом окне отсортируем все числа по возрастанию. Затем в разделе **SELECT** применим оконные функции ранжирования.

shape	val	Номер	Ранг	Распределение	Процентный_ранг
конус	12	1	1	0.5	0.0
конус	22	2	2	1.0	1.0
куб	5	1	1	0.2	0.0
куб	7	2	2	0.4	0.25
куб	12	3	3	0.6	0.5
куб	16	4	4	0.8	0.75
куб	23	5	5	1.0	1.0
шар	5	1	1	$2 / 3 = 0.666$	
шар	5	2	1	$2 / 3 = 0.666$	
шар	22	3	3	$3 / 3 = 1.0$	

Оконные функции ранжирования

Пример. Создадим окно по каждому типу геометрических фигур. В каждом окне отсортируем все числа по возрастанию. Затем в разделе **SELECT** применим оконные функции ранжирования.

shape	val	Номер	Ранг	Распределение	Процентный_ранг
конус	12	1	1	0.5	0.0
конус	22	2	2	1.0	1.0
куб	5	1	1	0.2	0.0
куб	7	2	2	0.4	0.25
куб	12	3	3	0.6	0.5
куб	16	4	4	0.8	0.75
куб	23	5	5	1.0	1.0
шар	5	1	1	0.6666666666666666	$(1 - 1) / (3 - 1) = 0.0$
шар	5	2	1	0.6666666666666666	
шар	22	3	3	1.0	

Оконные функции ранжирования

Пример. Создадим окно по каждому типу геометрических фигур. В каждом окне отсортируем все числа по возрастанию. Затем в разделе **SELECT** применим оконные функции ранжирования.

shape	val	Номер	Ранг	Распределение	Процентный_ранг
конус	12	1	1	0.5	0.0
конус	22	2	2	1.0	1.0
куб	5	1	1	0.2	0.0
куб	7	2	2	0.4	0.25
куб	12	3	3	0.6	0.5
куб	16	4	4	0.8	0.75
куб	23	5	5	1.0	1.0
шар	5	1	1	0.66666666666666	$(1 - 1) / (3 - 1) = 0.0$
шар	5	2	1	0.66666666666666	$(1 - 1) / (3 - 1) = 0.0$
шар	22	3	3	1.0	

Оконные функции ранжирования

Пример. Создадим окно по каждому типу геометрических фигур. В каждом окне отсортируем все числа по возрастанию. Затем в разделе **SELECT** применим оконные функции ранжирования.

shape	val	Номер	Ранг	Распределение	Процентный_ранг
конус	12	1	1	0.5	0.0
конус	22	2	2	1.0	1.0
куб	5	1	1	0.2	0.0
куб	7	2	2	0.4	0.25
куб	12	3	3	0.6	0.5
куб	16	4	4	0.8	0.75
куб	23	5	5	1.0	1.0
шар	5	1	1	0.6666666666666666	$(1 - 1) / (3 - 1) = 0.0$
шар	5	2	1	0.6666666666666666	$(1 - 1) / (3 - 1) = 0.0$
шар	22	3	3	1.0	$(3 - 0) / (3 - 1) = 1.0$

Объявление окна, два раздела

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
    ORDER BY ...  
);
```

Объявление окна, два раздела

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
    ORDER BY ...  
);
```

FIRST_VALUE(столбец)

Возвращает значение столбца, расположенного в первой строке(записи) окна .

LAST_VALUE(столбец)

Возвращает значение указанного столбца, расположенного в текущей записи, если указана сортировка. В противном случае – последнюю запись окна

Объявление окна, два раздела

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
    ORDER BY ...  
);
```

NTH_VALUE(столбец, номер)

Возвращает значение столбца, расположенного в строке с указанным номером, той части окна которая ограничена началом окна и текущей записью. Если в этой части окна записи с указанным номером нет, возвращается **NULL**.

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, последнее и третье значение столбца **val** для каждого "окна".

```
SELECT val, shape,  
        FIRST_VALUE(val) OVER win_test AS Первое,  
        LAST_VALUE(val) OVER win_test AS Последнее,  
        NTH_VALUE(val, 3) OVER win_test AS Третье  
FROM test  
WINDOW win_test  
AS(  
    PARTITION BY shape  
    ORDER BY val  
);
```



Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, последнее и третье значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12			
конус	22			
куб	16			
куб	16			
куб	5			
куб	23			
куб	7			
шар	22			
шар	5			
шар	5			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором **отсортируем значения столбца val по возрастанию**. Затем в **SELECT** найдем первое, последнее и третье значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12			
конус	22			
куб	5			
куб	7			
куб	12			
куб	16			
куб	23			
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, последнее и третье значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5			
куб	7			
куб	12			
куб	16			
куб	23			
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем **первое**, последнее и третье значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5			
куб	7			
куб	12			
куб	16			
куб	23			
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем **первое**, последнее и третье значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5		
куб	7	5		
куб	12	5		
куб	16	5		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, **последнее** и третье значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5		
куб	7	5		
куб	12	5		
куб	16	5		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, **последнее** и третье значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5		
куб	7	5		
куб	12	5		
куб	16	5		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, **последнее** и третье значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5	5	
куб	7	5		
куб	12	5		
куб	16	5		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, **последнее** и третье значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5	5	
куб	7	5		
куб	12	5		
куб	16	5		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, **последнее** и третье значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5	5	
куб	7	5	7	
куб	12	5		
куб	16	5		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, **последнее** и третье значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5	5	
куб	7	5	7	
куб	12	5	12	
куб	16	5		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, **последнее** и третье значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5	5	
куб	7	5	7	
куб	12	5	12	
куб	16	5	16	
куб	23	5	23	
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, последнее и **третье** значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5	5	
куб	7	5	7	
куб	12	5	12	
куб	16	5	16	
куб	23	5	23	
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, последнее и **третье** значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5	5	NULL
куб	7	5	7	
куб	12	5	12	
куб	16	5	16	
куб	23	5	23	
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, последнее и **третье** значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5	5	NULL
куб	7	5	7	NULL
куб	12	5	12	
куб	16	5	16	
куб	23	5	23	
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, последнее и **третье** значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5	5	NULL
куб	7	5	7	NULL
куб	12	5	12	
куб	16	5	16	
куб	23	5	23	
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, последнее и **третье** значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5	5	NULL
куб	7	5	7	NULL
куб	12	5	12	12
куб	16	5	16	
куб	23	5	23	
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, последнее и **третье** значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5	5	NULL
куб	7	5	7	NULL
куб	12	5	12	12
куб	16	5	16	
куб	23	5	23	
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, последнее и **третье** значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5	5	NULL
куб	7	5	7	NULL
куб	12	5	12	12
куб	16	5	16	12
куб	23	5	23	
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, последнее и **третье** значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5	5	NULL
куб	7	5	7	NULL
куб	12	5	12	12
куб	16	5	16	12
куб	23	5	23	12
шар	5			
шар	5			
шар	22			

Оконные функции (первое, последнее, ...)

Пример. В таблице **test** создадим окно по типу фигуры, в котором отсортируем значения столбца **val** по возрастанию. Затем в **SELECT** найдем первое, последнее и **третье** значение столбца **val** для каждого "окна".

shape	val	Первое	Последнее	Третье
конус	12	12	12	NULL
конус	22	12	22	NULL
куб	5	5	5	NULL
куб	7	5	7	NULL
куб	12	5	12	12
куб	16	5	16	12
куб	23	5	23	12
шар	5	5	5	NULL
шар	5	5	5	NULL
шар	22	5	22	22

Объявление окна, два раздела

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
    ORDER BY ...  
);
```

Объявление окна, два раздела

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
    ORDER BY ...  
);
```

LAG(столбец)

Для каждой записи возвращает значение столбца, расположенного в предыдущей записи окна. Если предыдущей записи нет (строка первая в окне), то возвращается значение **NULL**.

Объявление окна, два раздела

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
    ORDER BY ...  
);
```

LAG(столбец)

Для каждой записи возвращает значение столбца, расположенного в предыдущей записи окна. Если предыдущей записи нет (строка первая в окне), то возвращается значение **NULL**.

LEAD(столбец)

Для каждой записи возвращает значение столбца, расположенного в следующей записи окна. Если следующей записи нет (строка последняя в окне), то возвращается значение **NULL**.

Оконные функции (следующее, предыдущее)

Пример. Создадим "окно" по типу геометрической фигуры. В этом окне отсортируем все числа по возрастанию, а затем в **SELECT** для каждой записи найдем значения столбца **val** из предыдущей и следующей записи.

```
SELECT val, shape,  
        LAG(val) OVER win_test AS Предыдущее,  
        LEAD(val) OVER win_test AS Следующее  
FROM test  
WINDOW win_test  
AS(  
    PARTITION BY shape  
    ORDER BY val  
);
```

Оконные функции (следующее, предыдущее)

Пример. Создадим "окно" по типу геометрической фигуры. В этом окне отсортируем все числа по возрастанию, а затем в **SELECT** для каждой записи найдем значения столбца **val** из предыдущей и следующей записи.

shape	val	Предыдущее	Следующее
конус	12		
конус	22		
куб	16		
куб	16		
куб	5		
куб	23		
куб	7		
шар	22		
шар	5		
шар	5		

Оконные функции (следующее, предыдущее)

Пример. Создадим "окно" по типу геометрической фигуры. В этом окне **отсортируем все числа по возрастанию**, а затем в **SELECT** для каждой записи найдем значения столбца **val** из предыдущей и следующей записи.

shape	val	Предыдущее	Следующее
конус	12	↓	
конус	22		
куб	5	↓	
куб	7		
куб	12		
куб	16		
куб	23		
шар	5	↓	
шар	5		
шар	22		

Оконные функции (следующее, предыдущее)

Пример. Создадим "окно" по типу геометрической фигуры. В этом окне отсортируем все числа по возрастанию, а затем в **SELECT** для каждой записи найдем значения столбца **val** из предыдущей и следующей записи.

shape	val	Предыдущее	Следующее
конус	12		
конус	22		
куб	5		
куб	7		
куб	12		
куб	16		
куб	23		
шар	5		
шар	5		
шар	22		

Оконные функции (следующее, предыдущее)

Пример. Создадим "окно" по типу геометрической фигуры. В этом окне отсортируем все числа по возрастанию, а затем в **SELECT** для каждой записи найдем значения столбца **val** из предыдущей и следующей записи.

shape	val	Предыдущее	Следующее
конус	12		
конус	22		
куб	5		
куб	7		
куб	12		
куб	16		
куб	23		
шар	5		
шар	5		
шар	22		

Оконные функции (следующее, предыдущее)

Пример. Создадим "окно" по типу геометрической фигуры. В этом окне отсортируем все числа по возрастанию, а затем в **SELECT** для каждой записи найдем значения столбца **val** из предыдущей и следующей записи.

shape	val	Предыдущее	Следующее
конус	12	NULL	22
конус	22		
куб	5		
куб	7		
куб	12		
куб	16		
куб	23		
шар	5		
шар	5		
шар	22		

Оконные функции (следующее, предыдущее)

Пример. Создадим "окно" по типу геометрической фигуры. В этом окне отсортируем все числа по возрастанию, а затем в **SELECT** для каждой записи найдем значения столбца **val** из предыдущей и следующей записи.

shape	val	Предыдущее	Следующее
конус	12	NULL	22
конус	22		
куб	5		
куб	7		
куб	12		
куб	16		
куб	23		
шар	5		
шар	5		
шар	22		

Оконные функции (следующее, предыдущее)

Пример. Создадим "окно" по типу геометрической фигуры. В этом окне отсортируем все числа по возрастанию, а затем в **SELECT** для каждой записи найдем значения столбца **val** из предыдущей и следующей записи.

shape	val	Предыдущее	Следующее
конус	12	NULL	22
конус	22	12	NULL
куб	5		
куб	7		
куб	12		
куб	16		
куб	23		
шар	5		
шар	5		
шар	22		

Оконные функции (следующее, предыдущее)

Пример. Создадим "окно" по типу геометрической фигуры. В этом окне отсортируем все числа по возрастанию, а затем в **SELECT** для каждой записи найдем значения столбца **val** из предыдущей и следующей записи.

shape	val	Предыдущее	Следующее
конус	12	NULL	22
конус	22	12	NULL
куб	5	NULL	7
куб	7	5	12
куб	12	7	16
куб	16	12	23
куб	23	16	NULL
шар	5	NULL	5
шар	5	5	22
шар	22	5	NULL

Объявление окна, один раздел

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
);
```

В оконных выражениях можно использовать групповые функции, которые будут вычислять значения, группируя данные по столбцу (столбцам), указанному после **PARTITION BY**.

При этом вычисленные значения будут доступны из КАЖДОЙ записи окна. И, не смотря на применение групповых функций, количество записей в окне НЕ ИЗМЕНИТСЯ.

Оконные групповые функции

SUM(столбец)

Возвращает сумму значений указанного в скобках столбца, относящихся к группе, построенной по столбцу после **PARTITION BY**. Значение доступно в каждой записи окна.

AVG(столбец)

Возвращает среднее значение, указанного в скобках столбца, записи которого относятся к группе, построенной по столбцу после **PARTITION BY**. Значение доступно в каждой записи окна.

COUNT(столбец)

Возвращает количество записей, относящейся к группе, построенной по столбцу после **PARTITION BY**. Значение доступно в каждой записи окна.

Оконные групповые функции

MIN(столбец)

Возвращает минимальное значение указанного в скобках столбца, относящихся к группе, построенной по столбцу после **PARTITION BY**. Значение доступно в каждой записи окна.

MAX(столбец)

Возвращает максимальное значение, указанного в скобках столбца, записи которого относятся к группе, построенной по столбцу после **PARTITION BY**. Значение доступно в каждой записи окна.

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры. Посчитаем количество записей в каждом окне, суммарный, средний, минимальный и максимальный вес каждой геометрической фигуры.

```
SELECT shape, val,  
        COUNT(val) OVER win_test AS Количество,  
        SUM(val) OVER win_test AS Сумма,  
        AVG(val) OVER win_test AS Среднее,  
        MIN(val) OVER win_test AS Минимальное,  
        MAX(val) OVER win_test AS Максимальное  
FROM test  
WINDOW win_test  
AS(  
    PARTITION BY shape  
);
```

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры. Посчитаем количество записей в каждом окне, суммарный, средний, минимальный и максимальный вес каждой геометрической фигуры.

shape	val	Количество	Сумма	Среднее	Минимальное	Максимальное
конус	12					
конус	22					
куб	12					
куб	16					
куб	5					
куб	23					
куб	7					
шар	22					
шар	5					
шар	5					

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры. Посчитаем количество записей в каждом окне, суммарный, средний, минимальный и максимальный вес каждой геометрической фигуры.

shape	val	Количество	Сумма	Среднее	Минимальное	Максимальное
конус	12					
конус	22					
куб	12					
куб	16					
куб	5					
куб	23					
куб	7					
шар	22					
шар	5					
шар	5					

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры. Посчитаем **количество** записей в каждом окне, суммарный, средний, минимальный и максимальный вес каждой геометрической фигуры.

shape	val	Количество	Сумма	Среднее	Минимальное	Максимальное
конус	12	2				
конус	22					
куб	12					
куб	16					
куб	5					
куб	23					
куб	7					
шар	22					
шар	5					
шар	5					

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры. Посчитаем количество записей в каждом окне, **суммарный**, средний, минимальный и максимальный **вес** каждой геометрической фигуры.

shape	val	Количество	Сумма	Среднее	Минимальное	Максимальное
конус	12	2	12+22=34			
конус	22					
куб	12					
куб	16					
куб	5					
куб	23					
куб	7					
шар	22					
шар	5					
шар	5					

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры. Посчитаем количество записей в каждом окне, суммарный, **средний**, минимальный и максимальный **вес** каждой геометрической фигуры.

shape	val	Количество	Сумма	Среднее	Минимальное	Максимальное
конус	12	2	34	$34 / 2 = 17$		
конус	22					
куб	12					
куб	16					
куб	5					
куб	23					
куб	7					
шар	22					
шар	5					
шар	5					

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры. Посчитаем количество записей в каждом окне, суммарный, средний, **минимальный** и максимальный **вес** каждой геометрической фигуры.

shape	val	Количество	Сумма	Среднее	Минимальное	Максимальное
конус	12	2	34	17.0	12	
конус	22					
куб	12					
куб	16					
куб	5					
куб	23					
куб	7					
шар	22					
шар	5					
шар	5					

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры. Посчитаем количество записей в каждом окне, суммарный, средний, минимальный и **максимальный вес** каждой геометрической фигуры.

shape	val	Количество	Сумма	Среднее	Минимальное	Максимальное
конус	12	2	34	17.0	12	22
конус	22					
куб	12					
куб	16					
куб	5					
куб	23					
куб	7					
шар	22					
шар	5					
шар	5					

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры. Посчитаем количество записей в каждом окне, суммарный, средний, минимальный и максимальный вес каждой геометрической фигуры.

shape	val	Количество	Сумма	Среднее	Минимальное	Максимальное
конус	12	2	34	17.0	12	22
конус	22	2	34	17.0	12	22
куб	12					
куб	16					
куб	5					
куб	23					
куб	7					
шар	22					
шар	5					
шар	5					

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры. Посчитаем количество записей в каждом окне, суммарный, средний, минимальный и максимальный вес каждой геометрической фигуры.

shape	val	Количество	Сумма	Среднее	Минимальное	Максимальное
конус	12	2	34	17.0	12	22
конус	22	2	34	17.0	12	22
куб	12	5	63	12.6	5	23
куб	16	5	63	12.6	5	23
куб	5	5	63	12.6	5	23
куб	23	5	63	12.6	5	23
куб	7	5	63	12.6	5	23
шар	22	3	32	10.6667	5	22
шар	5	3	32	10.6667	5	22
шар	5	3	32	10.6667	5	22

Объявление окна, два раздела

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
    ORDER BY ...  
);
```

В оконных выражениях можно использовать групповые функции, которые будут вычислять значения, группируя данные по столбцу (столбцам), указанному после **PARTITION BY** с учетом отсортированных значений столбцов, указанных после **ORDER BY**.

Объявление окна, два раздела

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
    ORDER BY столбец_1  
);
```

В оконных выражениях можно использовать групповые функции, которые будут вычислять значения, группируя данные по столбцу (столбцам), указанному после **PARTITION BY** с учетом отсортированных значений столбцов, указанных после **ORDER BY**. Пусть указана сортировка по **столбцу_1**.

Оконные групповые функции

SUM(столбец)

Возвращает сумму значений указанного в скобках **столбца**, относящихся к группе, которая начинается с первой записи окна и включает записи, в которых значение **столбца_1** меньше или равно (если указана сортировка по возрастанию) значению **столбца_1** в текущей записи. Значение доступно в каждой записи окна.

Оконные групповые функции

AVG(столбец)

Возвращает среднее значений указанного в скобках **столбца**, относящихся к группе, которая начинается с первой записи окна и включает записи, в которых значение **столбца_1** меньше или равно (если указана сортировка по возрастанию) значению **столбца_1** в текущей записи. Значение доступно в каждой записи окна.

Оконные групповые функции

COUNT(столбец)

Возвращает количество записей, относящихся к группе, которая начинается с первой записи окна и включает записи, в которых значение **столбца_1** меньше или равно (если указана сортировка по возрастанию) значению **столбца_1** в текущей записи. Значение доступно в каждой записи окна.

Оконные групповые функции

MIN(столбец)

Возвращает минимальное значение указанного в скобках **столбца**, относящихся к группе, которая начинается с первой записи окна и включает записи, в которых значение **столбца_1** меньше или равно (если указана сортировка по возрастанию) значению **столбца_1** в текущей записи. Значение доступно в каждой записи окна.

Оконные групповые функции

MAX(столбец)

Возвращает максимальное значение указанного в скобках **столбца**, относящихся к группе, которая начинается с первой записи окна и включает записи, в которых значение **столбца_1** меньше или равно (если указана сортировка по возрастанию) значению **столбца_1** в текущей записи. Значение доступно в каждой записи окна.

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

SELECT

shape, val,

COUNT(val) **OVER** win_test **AS** Количество,

SUM(val) **OVER** win_test **AS** Сумма,

AVG(val) **OVER** win_test **AS** Среднее

FROM test

WINDOW win_test

AS(

PARTITION BY shape

ORDER BY val

);

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12			
конус	22			
куб	16			
куб	16			
куб	5			
куб	23			
куб	7			
шар	22			
шар	5			
шар	5			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем **сортировку по возрастанию веса в "окне"** и применим функции **SUM(), AVG(), COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12			
конус	22			
куб	5			
куб	7			
куб	16			
куб	16			
куб	23			
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12			
конус	22			
куб	5			
куб	7			
куб	16			
куб	16			
куб	23			
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12			
конус	22			
куб	5	1		
куб	7			
куб	16			
куб	16			
куб	23			
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12			
конус	22			
куб	5	1	5	
куб	7			
куб	16			
куб	16			
куб	23			
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12			
конус	22			
куб	5	1	5	5.0
куб	7			
куб	16			
куб	16			
куб	23			
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12			
конус	22			
куб	5	1	5	5.0
куб	7	2		
куб	16			
куб	16			
куб	23			
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12			
конус	22			
куб	5	1	5	5.0
куб	7	2	5 + 7 = 12	
куб	16			
куб	16			
куб	23			
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12			
конус	22			
куб	5	1	5	5.0
куб	7	2	12	$(5 + 7) / 2 = 6.0$
куб	16			
куб	16			
куб	23			
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12			
конус	22			
куб	5	1	5	5.0
куб	7	2	12	6.0
куб	16	4		
куб	16	4		
куб	23			
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12			
конус	22			
куб	5	1	5	5.0
куб	7	2	12	6.0
куб	16	4	5+7+16+16=44	
куб	16	4	44	
куб	23			
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12			
конус	22			
куб	5	1	5	5.0
куб	7	2	12	6.0
куб	16	4	44	44 / 4 = 11.0
куб	16	4	44	11.0
куб	23			
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12			
конус	22			
куб	5	1	5	5.0
куб	7	2	12	6.0
куб	16	4	44	44 / 4 = 11.0
куб	16	4	44	11.0
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12			
конус	22			
куб	5	1	5	5.0
куб	7	2	12	6.0
куб	16	4	44	44 / 4 = 11.0
куб	16	4	44	11.0
куб	23	5	44 + 23 = 63	
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12			
конус	22			
куб	5	1	5	5.0
куб	7	2	12	6.0
куб	16	4	44	11.0
куб	16	4	44	11.0
куб	23	5	67	67 / 5 = 13.4
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12			
конус	22			
куб	5	1	5	5.0
куб	7	2	12	6.0
куб	16	4	44	11.0
куб	16	4	44	11.0
куб	23	5	67	13.4
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне" и применим функции **SUM()**, **AVG()**, **COUNT()**.

shape	val	Количество	Сумма	Среднее
конус	12	1	12	12.0
конус	22	2	34	17.0
куб	5	1	5	5.0
куб	7	2	12	6.0
куб	16	4	44	11.0
куб	16	4	44	11.0
куб	23	5	67	13.4
шар	5	2	10	5.0
шар	5	2	10	5.0
шар	22	3	32	10.6667

Объявление окна, три раздела

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
    ORDER BY ...  
    ROWS спецификация  
);
```

Такое объявление окна позволяет разделить данные на "окна" по столбцам, указанным после **PARTITION BY**, отсортировать записи внутри окна по столбцам, указанным после **ORDER BY**, а затем использовать ГРУППОВЫЕ оконные функции для записей окна, ограниченных спецификацией.

Спецификация ограничения строк

UNBOUNDED PRECEDING

Указывает на то, что оконные функции будут применяться для всех записей от начала окна до текущей записи до конца окна.

число PRECEDING

Указывает на то, что оконные функции будут применяться для текущей и предыдущих записей, количество которых меньше или равно указанному числу.

Спецификация ограничения строк

ВАЖНО!

Спецификация ограничения строк действует только для ГРУППОВЫХ оконных функций (**SUM()**, **COUNT()**, **AVG()**, **MIN()**, **MAX()**).

Остальные оконные функции (**ROW_NUMBER()**, **LAG()**, **LEAD()** и другие) работают СО ВСЕМИ СТРОКАМИ ОКНА, без учета указанных ограничений.

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение от начала окна до текущей записи и применим функции **SUM()**, **COUNT()**.

```
SELECT shape, val,  
       ROW_NUMBER() OVER win_test AS Nпп,  
       COUNT(val) OVER win_test AS Количество,  
       SUM(val) OVER win_test AS Сумма  
FROM test  
WINDOW win_test  
AS(  
    PARTITION BY shape  
    ORDER BY val  
    ROWS UNBOUNDED PRECEDING  
);
```

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение от начала окна до текущей записи и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	16			
куб	16			
куб	5			
куб	23			
куб	7			
шар	22			
шар	5			
шар	5			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение от начала окна до текущей записи и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5			
куб	7			
куб	16			
куб	16			
куб	23			
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение от начала окна до текущей записи и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5			
куб	7			
куб	16			
куб	16			
куб	23			
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение от начала окна до текущей записи и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1		
куб	7	2		
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение от начала окна до текущей записи и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	1	
куб	7	2		
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение от начала окна до текущей записи и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	1	
куб	7	2	2	
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение от начала окна до текущей записи и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	1	
куб	7	2	2	
куб	16	3	3	
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение от начала окна до текущей записи и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	1	
куб	7	2	2	
куб	16	3	3	
куб	16	4	4	
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение от начала окна до текущей записи и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	1	
куб	7	2	2	
куб	16	3	3	
куб	16	4	4	
куб	23	5	5	
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение от начала окна до текущей записи и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	1	5
куб	7	2	2	
куб	16	3	3	
куб	16	4	4	
куб	23	5	5	
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение от начала окна до текущей записи и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	1	5
куб	7	2	2	5 + 7 =12
куб	16	3	3	
куб	16	4	4	
куб	23	5	5	
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение от начала окна до текущей записи и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	1	5
куб	7	2	2	12
куб	16	3	3	5+7+16=28
куб	16	4	4	
куб	23	5	5	
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение от начала окна до текущей записи и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	1	5
куб	7	2	2	12
куб	16	3	3	28
куб	16	4	4	5+7+16+16=44
куб	23	5	5	
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение от начала окна до текущей записи и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	1	5
куб	7	2	2	12
куб	16	3	3	28
куб	16	4	4	44
куб	23	5	5	44 + 23 = 67
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение от начала окна до текущей записи и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12	1	1	12
конус	22	2	2	34
куб	5	1	1	5
куб	7	2	2	12
куб	16	3	3	28
куб	16	4	4	44
куб	23	5	5	67
шар	5	1	1	5
шар	5	2	2	10
шар	22	3	3	32

Объявление окна, три раздела

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
    ORDER BY ...  
    ROWS BETWEEN начало AND конец  
);
```

Такое объявление окна позволяет разделить данные на "окна" по столбцам, указанным после **PARTITION BY**, отсортировать записи внутри окна по столбцам, указанным после **ORDER BY**, а затем использовать ГРУППОВЫЕ оконные функции для записей окна, ограниченных строками, обозначенными спецификацией **начало**, и строками, обозначенными спецификацией **конец**, включая границы.

Объявление окна, три раздела

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
    ORDER BY ...  
    ROWS BETWEEN начало AND конец  
);
```

В качестве спецификаций **начало** и **конец** можно использовать :

- **CURRENT ROW** - текущая строка;
- **UNBOUNDED PRECEDING** - первая строка окна;
- **UNBOUNDED FOLLOWING** - последняя строка окна;
- **число PRECEDING** - строка, расположенная перед текущей на указанное **число** строк;
- **число FOLLOWING** - строка, расположенная после текущей на указанное **число** строк.

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

```
SELECT shape, val,  
        ROW_NUMBER() OVER win_test AS Nпп,  
        COUNT(val) OVER win_test AS Количество,  
        SUM(val) OVER win_test AS Сумма  
FROM test  
WINDOW win_test  
AS(  
    PARTITION BY shape  
    ORDER BY val  
    ROWS BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING  
);
```

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	16			
куб	16			
куб	5			
куб	23			
куб	7			
шар	22			
шар	5			
шар	5			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем **сортировку по возрастанию веса в "окне"**, укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5			
куб	7			
куб	16			
куб	16			
куб	23			
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5			
куб	7			
куб	16			
куб	16			
куб	23			
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1		
куб	7	2		
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	
куб	7	2		
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	5+7+16+16+23
куб	7	2		
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	67
куб	7	2	4	
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	67
куб	7	2	4	7+16+16+23
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	67
куб	7	2	4	62
куб	16	3	3	
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	67
куб	7	2	4	62
куб	16	3	3	16+16+23=55
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	67
куб	7	2	4	62
куб	16	3	3	55
куб	16	4	2	
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	67
куб	7	2	4	62
куб	16	3	3	55
куб	16	4	2	16+23=39
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	67
куб	7	2	4	62
куб	16	3	3	55
куб	16	4	2	39
куб	23	↕	1	
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	67
куб	7	2	4	62
куб	16	3	3	55
куб	16	4	2	39
куб	23	↕	1	23
шар	5			
шар	5			
шар	22			

Оконные групповые функции

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12	1	2	24
конус	22	2	1	22
куб	5	1	5	67
куб	7	2	4	62
куб	16	3	3	55
куб	16	4	2	39
куб	23	5	1	23
шар	5	1	3	32
шар	5	2	2	27
шар	22	3	1	22

Объявление окна, три раздела

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
    ORDER BY ...  
    RANGE BETWEEN начало AND конец  
);
```

Такое объявление окна позволяет разделить данные на "окна", отсортировать записи внутри окна по столбцам, указанным после **ORDER BY**, а затем использовать групповые оконные функции для записей окна, ограниченных строками, обозначенными спецификацией **начало**, и строками, обозначенными спецификацией **конец**, включая границы.

Объявление окна, три раздела

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
    ORDER BY столбец_1  
    RANGE BETWEEN начало AND конец  
);
```

В качестве спецификаций **начало** и **конец** можно использовать:

- **CURRENT ROW** - текущая строка или строки, имеющие одинаковые с текущей значения в **столбце_1**;
- **UNBOUNDED PRECEDING** – первая строка окна или строки, имеющие одинаковые с первой значения в **столбце_1**;
- **UNBOUNDED FOLLOWING** – последняя строка окна или строки, имеющие одинаковые с последней значения в **столбце_1**;

Объявление окна, три раздела

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
    ORDER BY столбец_1  
    RANGE BETWEEN начало AND конец  
);
```

В качестве спецификаций **начало** и **конец** можно использовать:

- **число PRECEDING** - строки со значениями, равными текущему значению строки в **столбце_1** минус **число**.
- **число FOLLOWING** - строки со значениями, равными текущему значению строки в **столбце_1** плюс **число**.

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

```
SELECT shape, val,  
       ROW_NUMBER() OVER win_test AS Nпп,  
       COUNT(val) OVER win_test AS Количество,  
       SUM(val) OVER win_test AS Сумма  
FROM test  
WINDOW win_test  
AS(  
    PARTITION BY shape  
    ORDER BY val  
    RANGE BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING  
);
```

Оконные функции, ограничение строк

Пример В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5			
куб	5			
куб	16			
куб	16			
куб	16			
шар	5			
шар	5			
шар	22			

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1		
куб	5	2		
куб	16	3		
куб	16	4		
куб	16	5		
шар	5			
шар	5			
шар	22			

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	
куб	5	2		
куб	16	3		
куб	16	4		
куб	16	5		
шар	5			
шар	5			
шар	22			

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	5+5+16+16+16
куб	5	2		
куб	16	3		
куб	16	4		
куб	16	5		
шар	5			
шар	5			
шар	22			

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	58
куб	5	2	5	
куб	16	3		
куб	16	4		
куб	16	5		
шар	5			
шар	5			
шар	22			

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	58
куб	5	2	5	58
куб	16	3		
куб	16	4		
куб	16	5		
шар	5			
шар	5			
шар	22			

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	58
куб	5	2	5	58
куб	16	3	3	
куб	16	4		
куб	16	5		
шар	5			
шар	5			
шар	22			

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	58
куб	5	2	5	58
куб	16	3	3	16+16+16=48
куб	16	4		
куб	16	5		
шар	5			
шар	5			
шар	22			

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	58
куб	5	2	5	58
куб	16	3	3	48
куб	16	4	3	
куб	16	5		
шар	5			
шар	5			
шар	22			

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	58
куб	5	2	5	58
куб	16	3	3	48
куб	16	4	3	48
куб	16	5		
шар	5			
шар	5			
шар	22			

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	5	58
куб	5	2	5	58
куб	16	3	3	48
куб	16	4	3	48
куб	16	5	3	48
шар	5			
шар	5			
шар	22			

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", укажем ограничение строк от текущей до конца окна и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12	1	2	24
конус	22	2	1	22
куб	5	1	5	58
куб	5	2	5	58
куб	16	3	3	48
куб	16	4	3	48
куб	16	5	3	48
шар	5	1	3	32
шар	5	2	3	32
шар	22	3	1	22

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

```
SELECT shape, val,  
        ROW_NUMBER() OVER win_test AS Nпп,  
        COUNT(val) OVER win_test AS Количество,  
        SUM(val) OVER win_test AS Сумма  
FROM test  
WINDOW win_test  
AS(  
    PARTITION BY shape  
    ORDER BY val  
    RANGE BETWEEN 10 PRECEDING AND 10 FOLLOWING  
);
```

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5			
куб	7			
куб	16			
куб	16			
куб	23			
шар	5			
шар	5			
шар	22			

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1		
куб	7	2		
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1		
куб	7	2		
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

от (5 - 10) до (5 + 10)
от -5 до 15

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1		
куб	7	2		
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

от (5 - 10) до (5 + 10)
от -5 до 15

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	2	5+7=12
куб	7	2		
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

от (5 - 10) до (5 + 10)
от -5 до 15

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	2	12
куб	7	2		
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

от (7 - 10) до (7 + 10)
от -3 до 17

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	2	12
куб	7	2		
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

от (7 - 10) до (7 + 10)
от -3 до 17

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	2	12
куб	7	2	4	5+7+16+16=44
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

от (7 - 10) до (7 + 10)
от -3 до 17

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	2	12
куб	7	2	4	44
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

от (16 - 10) до (16 + 10)
от 6 до 26

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	2	12
куб	7	2	4	44
куб	16	3		
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

от (16 - 10) до (16 + 10)
от 6 до 26

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	2	12
куб	7	2	4	44
куб	16	3	4	7+16+16+23=62
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

от (16 - 10) до (16 + 10)
от 6 до 26

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	2	12
куб	7	2	4	44
куб	16	3	4	62
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

от (16 - 10) до (16 + 10)
от 6 до 26

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	2	12
куб	7	2	4	44
куб	16	3	4	62
куб	16	4		
куб	23	5		
шар	5			
шар	5			
шар	22			

от (16 - 10) до (16 + 10)
от 6 до 26

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	2	12
куб	7	2	4	44
куб	16	3	4	62
куб	16	4	4	7+16+16+23=62
куб	23	5		
шар	5			
шар	5			
шар	22			

от (16 - 10) до (16 + 10)
от 6 до 26

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	2	12
куб	7	2	4	44
куб	16	3	4	62
куб	16	4	4	62
куб	23	5		
шар	5			
шар	5			
шар	22			

от (23 - 10) до (23 + 10)
от 13 до 33

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	2	12
куб	7	2	4	44
куб	16	3	4	62
куб	16	4	4	62
куб	23	5		
шар	5			
шар	5			
шар	22			

от (23 - 10) до (23 + 10)
от 13 до 33

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12			
конус	22			
куб	5	1	2	12
куб	7	2	4	44
куб	16	3	4	62
куб	16	4	4	62
куб	23	5	3	16+16+23=55
шар	5			
шар	5			
шар	22			

от (23 - 10) до (23 + 10)
от 13 до 33

Оконные функции, ограничение строк

Пример. В таблице **test** создадим "окно" по типу геометрической фигуры, укажем сортировку по возрастанию веса в "окне", выделим записи, вес фигур которых отличается в большую или меньшую сторону на 10 включительно и применим функции **SUM()**, **COUNT()**.

shape	val	Нпп	Количество	Сумма
конус	12	1	2	34
конус	22	2	2	34
куб	5	1	2	12
куб	7	2	4	44
куб	16	3	4	62
куб	16	4	4	62
куб	23	5	3	55
шар	5	1	2	10
шар	5	2	2	10
шар	22	3	1	22

Структура запроса

SELECT

... ,

ОКОННАЯ_ФУНКЦИЯ (выражение) **OVER** имя_окна ,

...

FROM ...

WHERE условие

WINDOW имя_окна

AS (

...

)

ORDER BY ...

Порядок выполнения

SELECT

... /
око~~н~~ная_фу~~нк~~ция (выраже~~ни~~е) **OVER** имя_о~~к~~на,
...

FROM ...

WHERE усло~~ви~~е

WINDOW имя_о~~к~~на

AS (

...

)

ORDER BY ...

Порядок выполнения

SELECT

... /

ОКОННАЯ_ФУНКЦИЯ (выражение) **OVER** имя_окна ,

...

FROM ...

WHERE условие

WINDOW имя_окна

AS (

...

)

ORDER BY ...



Порядок выполнения

SELECT

... ,
око~~н~~ная_фу~~нк~~ция (выраже~~ни~~е) **OVER** имя_о~~к~~на ,

...

FROM ...

WHERE усло~~ви~~е

WINDOW имя_о~~к~~на

AS (

...

)

ORDER BY ...

I

II

III

Порядок выполнения

SELECT

... /

ОКОННАЯ_ФУНКЦИЯ (выражение) **OVER** имя_окна ,

...

FROM ...

WHERE условие

WINDOW имя_окна

AS (

...

)

ORDER BY ...

IV

I

II

III

Порядок выполнения

SELECT

... /

ОКОННАЯ_ФУНКЦИЯ (выражение) **OVER** имя_окна ,

...

FROM ...

WHERE условие

WINDOW имя_окна

AS (

...

)

ORDER BY ...

IV

I

II

III

V

Структура запроса

Окна можно создавать (и/или) в

- основном запросе;
- вложенном запросе;
- табличном выражении.

Структура запроса, табличные выражения

```
WITH имя_табличного_выражения (...)  
AS (  
    SELECT  
        ... ,  
        оконная_функция (выражение) OVER имя_окна ,  
        ...  
    FROM  
        ...  
    WINDOW имя_окна  
    AS (  
        ...  
    )  
)  
SELECT ...  
FROM ... имя_табличного_выражения ...  
...
```

Данные в запросе могут быть разбиты на «окна». Описание окна имеет вид:

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
    ORDER BY ...  
    ограничение строк  
);
```

После указываются **PARTITION BY** один или несколько столбцов, «окном» считаются записи, которые имеют одинаковые значения в этих столбцах.

Данные в каждом окне можно отсортировать по одному или нескольким столбцам, для этого их нужно указать в разделе **ORDER BY**.

Данные в запросе могут быть разбиты на «окна». Описание окна имеет вид:

```
WINDOW имя_окна  
AS(  
    PARTITION BY ...  
    ORDER BY ...  
    ограничение строк  
);
```

Чтобы выполнять операции не над всеми записями окна, используется раздел «ограничение строк».

Для работы с данными внутри окна используются **ОКОННЫЕ ФУНКЦИИ**:

- функции ранжирования (**ROW_NUMBER()**, **RANK()**, **PERCENT_RANK()**);
- функции смещения (**LAG()**, **LEAD()**, **FIRST_VALUE()**, **LAST_VALUE()**);
- агрегатные функции (**SUM()**, **COUNT()**, **AVG()**, **MIN()**, **MAX()**).

Задание

«Интерактивный тренажер по SQL»:

- модуль 3, урок 3, 4

«Расширенные возможности SQL» (по желанию)

- модуль 3, уроки 1-3

Спасибо за внимание!