

SQL Оконные функции. Часть 2



СПИСОК ОКОННЫХ ФУНКЦИЙ:

Функции агрегации

- **avg()**
- **count()**
- **max()**
- **min()**
- **sum()**

Функции ранжирования

- **row_number()**
- **rank()**
- **dense_rank()**

Функции распределения

- **percent_rank()**
- **cume_dist()**

Аналитические функции

- **lead()**
- **lag()**
- **ntile()**
- **first_value()**
- **last_value()**
- **nth_value()**

ФУНКЦИИ АГРЕГАЦИИ

- **avg(expr)** – среднее значение в строках оконного фрейма
- **count(expr)** – количество значений в строках оконного фрейма
- **max(expr)** – максимальное значение в оконном фрейме
- **min(expr)** – минимальное значение в оконном фрейме
- **sum(expr)** – сумма значений в оконном фрейме

ORDER BY и оконный фрейм:

Функции агрегации не требуют ORDER BY. Они принимают определение оконного фрейма (ROWS, RANGE, GROUPS).

ФУНКЦИИ РАНЖИРОВАНИЯ

- **row_number()** – уникальный номер для каждой строки в партии (вне зависимости от связанности значений номера будут разными)
- **rank()** – ранжирование в партии, с пробелами и одинаковым ранжированием для связанных значений (после строк с одинаковым рангом будет номер строки)
- **dense_rank()** – ранжирование в партии, без пробелов с одинаковым ранжированием для связанных значений

city	price	row_number	rank	dense_rank
		over(order by price)		
Paris	7	1	1	1
Rome	7	2	1	1
London	8.5	3	3	2
Berlin	8.5	4	3	2
Moscow	9	5	5	3
Madrid	10	6	6	4
Oslo	10	7	6	4

ORDER BY и оконный фрейм: rank() и dense_rank() требуют ORDER BY, но row_number() не требует ORDER BY. Функции ранжирования не принимают определение оконного фрейма (ROWS, RANGE, GROUPS).

АНАЛИТИЧЕСКИЕ ФУНКЦИИ

- **lead(expr, offset, default)** – возвращает значение, следующее за текущим с указанным шагом. *offset* и *default* опциональные; по умолчанию *offset* = 1, *default* = NULL
- **lag(expr, offset, default)** – возвращает значение, предыдущее от текущего с указанным шагом. *offset* и *default* опциональные; по умолчанию *offset* = 1, *default* = NULL

lead(sold) OVER(ORDER BY month)

order by month	month	sold	
	1	500	300
	2	300	400
	3	400	100
	4	100	500
	5	500	NULL

lag(sold) OVER(ORDER BY month)

order by month	month	sold	
	1	500	NULL
	2	300	500
	3	400	300
	4	100	400
	5	500	100

lead(sold, 2, 0) OVER(ORDER BY month)

order by month	month	sold	
	1	500	400
	2	300	100
	3	400	500
	4	100	0
	5	500	0

lag(sold, 2, 0) OVER(ORDER BY month)

order by month	month	sold	
	1	500	0
	2	300	0
	3	400	500
	4	100	300
	5	500	400

- **ntile(n)** – делит строки в партии на максимально равные *n* группы и присваивает каждой строке свой номер группы.

ntile(3)

city	sold	
Rome	100	1
Paris	100	1
London	200	1
Moscow	200	2
Berlin	200	2
Madrid	300	2
Oslo	300	3
Dublin	300	3

ORDER BY и оконный фрейм:

ntile(), lead(), and lag() требуют ORDER BY. Они не принимают определение оконного фрейма (ROWS, RANGE, GROUPS).

ФУНКЦИИ РАСПРЕДЕЛЕНИЯ

- **percent_rank()** – номер ранжирования строки в процентиле – значение в интервале [0, 1]: (ранг - 1) / (общее количество строк - 1)
- **cume_dist()** – сводное распределение значения в группе значений, например номера строк со значениями меньше или равно значения текущей строки поделенной на общее количество строк; значение в интервале [0, 1]:

percent_rank() OVER(ORDER BY sold)

city	sold	percent_rank
Paris	100	0
Berlin	150	0.25
Rome	200	0.5
Moscow	200	0.5
London	300	1

← не включая эту строку
50% значений меньше, чем значение этой строки

cume_dist() OVER(ORDER BY sold)

city	sold	cume_dist
Paris	100	0.2
Berlin	150	0.4
Rome	200	0.8
Moscow	200	0.8
London	300	1

← 80% значений меньше или равны значению этой строки

ORDER BY и Оконные функции: Функции распределения требуют ORDER BY. Они не принимают определение оконного фрейма (ROWS, RANGE, GROUPS).

- **first_value(expr)** – значение первой строки в оконном фрейме
- **last_value(expr)** – значение последней строки в оконном фрейме

first_value(sold) OVER
(PARTITION BY city ORDER BY month)

city	month	sold	first_value
Paris	1	500	500
Paris	2	300	500
Paris	3	400	500
Rome	2	200	200
Rome	3	300	200
Rome	4	500	200

last_value(sold) OVER
(PARTITION BY city ORDER BY month
RANGE BETWEEN UNBOUNDED PRECEDING
AND UNBOUNDED FOLLOWING)

city	month	sold	last_value
Paris	1	500	400
Paris	2	300	400
Paris	3	400	400
Rome	2	200	500
Rome	3	300	500
Rome	4	500	500

Заметка: Обычно, с last_value() хочется использовать конструкцию RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING. С дефолтным оконным фреймом ORDER BY, RANGE UNBOUNDED PRECEDING, last_value() возвращает значение текущей строки

- **nth_value(expr, n)** – значение *n*-й строки внутри оконного фрейма; *n* должен быть числом

nth_value(sold, 2) OVER (PARTITION BY city

ORDER BY month RANGE BETWEEN UNBOUNDED
PRECEDING AND UNBOUNDED FOLLOWING)

city	month	sold	nth_value
Paris	1	500	300
Paris	2	300	300
Paris	3	400	300
Rome	2	200	300
Rome	3	300	300
Rome	4	500	300
Rome	5	300	300
London	1	100	NULL

ORDER BY и оконный фрейм:

first_value(), last_value(), and nth_value() не требуют ORDER BY. Они принимают определение оконного фрейма (ROWS, RANGE, GROUPS).