

ОКОННЫЕ ФУНКЦИИ В POSTGRESQL

| ВВЕДЕНИЕ

На этом уроке мы

- Продолжим работу с SQL-запросами, содержащими соединения нескольких таблиц
- Освоим применение «оконных функций» и вычислений по партициям при помощи операторов WINDOW и PARTITION BY

| ОКОННАЯ ФУНКЦИЯ В SQL

Функция, которая работает с выделенным набором строк (окном, партицией) и выполняет вычисление для этого набора строк в отдельном столбце

| ПАРТИЦИИ (ОКНА ИЗ НАБОРА СТРОК)

это набор строк, указанный для оконной функции по одному из столбцов или группе столбцов таблицы. Партиции для каждой оконной функции в запросе могут быть разделены по различным колонкам таблицы

Партиции
оконной функции
(в данном примере
по полю **Имя**)



Имя	Предмет	Оценка
Петя	матем	3
Петя	рус	4
Петя	физ	5
Петя	история	4
Маша	матем	4
Маша	рус	3
Маша	физ	5
Маша	история	3

| В чем заключается главное отличие оконных функций от функций агрегации с группировкой?

При использовании агрегирующих функций предложение GROUP BY сокращает количество строк в запросе с помощью их группировки

Применение функции агрегации и команды GROUP BY

Имя	Предмет	Оценка
Петя	матем	3
Петя	рус	4
Петя	физ	5
Петя	история	4
Маша	матем	4
Маша	рус	3
Маша	физ	5
Маша	история	3

→

Имя	Средняя оценка
Петя	4
Маша	3,75

При использовании оконных функций количество строк в запросе не уменьшается по сравнению с исходной таблицей

Применение Оконной функции

Имя	Предмет	Оценка
Петя	матем	3
Петя	рус	4
Петя	физ	5
Петя	история	4
Маша	матем	4
Маша	рус	3
Маша	физ	5
Маша	история	3

→

Имя	Предмет	Оценка	Средняя оценка
Петя	матем	3	4
Петя	рус	4	4
Петя	физ	5	4
Петя	история	4	4
Маша	матем	4	3,75
Маша	рус	3	3,75
Маша	физ	5	3,75
Маша	история	3	3,75

Порядок расчета оконных функций в SQL запросе

FULL JOIN

list of columns,
window functions

FROM

table / combined tables /

WHERE

filtering clause

GROUP BY

list of columns

HAVING

aggregation filtering clause

ORDER BY

list of columns / window functions

- 01 Сначала выполняется команда выборки таблиц, их объединения и возможные подзапросы под командой **FROM**

- 02** Далее выполняются условия фильтрации **WHERE**, группировки GROUP BY и возможная фильтрация с **HAVING**
- 03** Применяется команда выборки столбцов **SELECT** и расчет оконных функций под выборкой
- 04** После этого идет условие сортировки **ORDER BY**, где тоже можно указать столбец расчета оконной функции для сортировки

Здесь важно уточнить, что партиции или окна оконных функций создаются после разделения таблицы на группы с помощью команды GROUP BY, если эта команда используется в запросе

| СИНТАКСИС ОКОННЫХ ФУНКЦИЙ

Имя оконной функции одного из классов →
 Необязательное выражение фильтрации →
 Ключевое слово определения оконной ф →
 Определениеパーティций по колонкам →
 Сортировка вычисления оконной функции →
 Указание фрейма для партиции →

```
o o o
FUNCTION NAME(column name)
[FILTER (WHERE filter_clause)]
OVER
(
PARTITION BY (column_names),
ORDER BY (column names),
[frame_clause]
)
```

Оконные функции можно прописывать как под командой SELECT, так и в отдельном ключевом слове WINDOW, где окну дается алиас (псевдоним), к которому можно обращаться в SELECT выборке

```
o o o
select name, subject, grade,
row_number() over (partition by name order by grade desc),
rank() over (partition by name order by grade desc),
dense_rank() over (partition by name order by grade desc)
from student_grades;
```

||

```
o o o
select name, subject, grade,
row_number() over name_grade,
rank() over name_grade,
dense_rank() over name_grade
from student_grades
window name_grade as (partition by name order by grade desc);
```

дубли
определения
окна

I КЛАССЫ ОКОННЫХ ФУНКЦИЙ

Множество оконных функций можно разделять на 3 класса

01

Агрегирующие (Aggregate)

02

Ранжирующие (Ranking)

03

Функции смещения (Value)

Window Functions

Aggregate

AVG()

COUNT()

MAX()

MIN()

SUM()

Ranking

DENSE_RANK()

NTILE()

RANK()

ROW_NUMBER()

CUME_DIST()

Value

FIRST_VALUE()

LAST_VALUE()

LAG()

LEAD()

NTH_VALUE()

01 Агрегирующие (Aggregate)

Можно применять любую из агрегирующих функций - SUM, AVG, COUNT, MIN, MAX

ооо

```
select name, subject, grade,
sum(grade) over (partition by name) as sum_grade,
avg(grade) over (partition by name) as avg_grade,
count(grade) over (partition by name) as count_grade,
min(grade) over (partition by name) as min_grade,
max(grade) over (partition by name) as max_grade
from student_grades;
```

name	subject	grade	sum_grade	avg_grade	count_grade	min_grade	max_grade
Маша	история	3	15	3,75	4	3	5
Маша	математика	4	15	3,75	4	3	5
Маша	русский	3	15	3,75	4	3	5
Маша	физика	5	15	3,75	4	3	5
Петя	математика	3	16	4	4	3	5
Петя	русский	4	16	4	4	3	5
Петя	физика	5	16	4	4	3	5
Петя	история	4	16	4	4	3	5

02 Ранжирующие (Ranking)

В ранжирующих функциях под ключевым словом OVER обязательным идет указание условия ORDER BY, по которому будет происходить сортировка ранжирования

ROW_NUMBER() - функция вычисляет последовательность ранг (порядковый номер) строк внутри партиции, НЕЗАВИСИМО от того, есть ли в строках повторяющиеся значения или нет

RANK() - функция вычисляет ранг каждой строки внутри партиции. Если есть повторяющиеся значения, функция возвращает одинаковый ранг для таких строчек, пропуская при этом следующий числовой ранг

DENSE_RANK() - то же самое что и RANK, только в случае одинаковых значений DENSE_RANK не пропускает следующий числовой ранг, а идет последовательно

ооо

```
select name, subject, grade,
row_number() over (partition by name order by grade desc),
rank() over (partition by name order by grade desc),
dense_rank() over (partition by name order by grade desc)
from student_grades;
```

abc name	abc subject	123 grade	123 row_number	123 rank	123 dense_rank
Маша	физика	5	1	1	1
Маша	математика	4	2	2	2
Маша	история	3	3	3	3
Маша	русский	3	4	3	3
Петя	физика	5	1	1	1
Петя	русский	4	2	2	2
Петя	история	4	3	2	2
Петя	математика	3	4	4	3

← Без пропуска значения
↑ Пропуск значения «3»

Про NULL в случае ранжирования:

Для SQL пустые NULL значения будут определяться одинаковым рангом

03 Функции смещения (Value)

Это функции, которые позволяют перемещаясь по выделенной партиции таблицы обращаться к предыдущему значению строки или крайним значениям строк в партиции

LAG() - функция, возвращает предыдущее значение столбца по порядку сортировки

LEAD() - функция, возвращающая следующее значение столбца по порядку сортировки

I ИТОГ

- ✓ Вы научились применять оконные функции и вычисления поパーティциям и разобрали примеры сочетания данных функций с рассмотренными ранее элементами SQL-запросов