Оконные функции

Оконная агрегирующая функция создаёт в итоговой таблице поле с агрегацией всего поля или по группам другого поля (полей), в зависимости от параметров указанных в `OVER`.

Пример расчёта агрегирующих оконных функций по всему окну:

```
SELECT SUM(поле) OVER (), -- подсчёт суммы всего поля в отдельном "столбце"
       COUNT(поле) OVER(),-- подсчёт числа записей всего поля в отдельном "столбце"
       AVG(поле) OVER (),-- подсчёт среднего всего поля в отдельном "столбце"
       MIN(поле) OVER (),-- подсчёт минимального значения всего поля в отдельном "столбце"
       MAX(поле) OVER ()-- подсчёт максимального всего поля в отдельном "столбце"
FROM таблица_1
```

Пример расчёта агрегирующих оконных функций в зависимости от группы в окне:

```
SELECT SUM(поле) OVER (PARTITION BY поле_2),
-- подсчёт суммы в отдельном "столбце" с разбиением на группы в поле_2
       COUNT(поле) OVER(PARTITION BY поле_2),
-- подсчёт числа в отдельном "столбце" с разбиением на группы в поле_2
       AVG(поле) OVER (PARTITION BY поле_2),
-- подсчёт среднего в отдельном "столбце" с разбиением на группы в поле_2
       MIN(поле) OVER (PARTITION BY поле_2),
-- подсчёт минимального в отдельном "столбце" с разбиением на группы в поле_2
       MAX(поле) OVER (PARTITION BY поле_2)
-- подсчёт максимального в отдельном "столбце" с разбиением на группы в поле_2
FROM таблица_1
```

Оконные функции ранжирования — оконные функции, при помощи которых можно ранжировать записи, то есть нумеровать их по определённому правилу.
Параметр оконной функции `ORDER BY` (не путать с оператором сортировки) позволяет задать порядок назначения рангов. Может формировать порядок по возрастанию (указывается `ASC` или ничего не указывается) или по убыванию (указывается `DESC`).
В сочетании с `PARTITION BY` формирует упорядоченное ранжирование по группам.
`ROW_NUMBER()` возвращает номер строки в зависимости от параметров. Номера строк начинаются с 1 и увеличиваются на 1 для каждой следующей строки.
`RANK()` возвращает ранг строки в зависимости от параметров. Если несколько строк имеют одинаковые значения, они получают один и тот же ранг, а следующий ранг будет пропущен.
`DENSE_RANK()` возвращает ранг строки в зависимости от параметров. Если несколько строк имеют одинаковые значения, они получают один и тот же ранг, и следующий ранг не будет пропущен.

Примеры использования:

```
SELECT ROW_NUMBER() OVER (),
-- назначит номер для каждой строки таблицы
       RANK() OVER(PARTITION BY поле_2 ORDER BY поле_3 DESC),
-- назначит ранги в обратном порядке от поля_3 с разделением на группы в поле_2
-- если несколько строк поля_3 имеют одинаковые значения,
-- они получают один и тот же ранг, а следующий ранг будет пропущен.
       DENSE_RANK() OVER (ORDER BY поле_3),
-- назначит ранги в прямом порядке от поля_3.
-- если несколько строк в поле_3 имеют одинаковые значения,
-- они получают один и тот же ранг, и следующий ранг не будет пропущен.
FROM таблица_1
```

Любая агрегирующая функция преобразуется в кумулятивную функцию, если к ней добавляется параметр окна `ORDER BY` .
Кумулятивные значения — значения, подсчитывающиеся с накоплением.
Примеры кумулятивных расчётов:

```
SELECT SUM(revenue) OVER (ORDER BY order_data)
-- будет подсчитана кумулятивная сумма поля revenue в зависимости от
-- порядка дат в поле order_data, то есть на каждую следующую дату
-- будет рассчитана сумма поля revenue всех предыдущих дат.
       AVG(revenue) OVER (ORDER BY order_data)
-- будет подсчитано кумулятивное среднее поля revenue в зависимости от
-- порядка дат в поле order_data, то есть на каждую следующую дату
-- будет рассчитано среднее значение поля revenue всех предыдущих дат.
       MIN(revenue) OVER (ORDER BY order_data)
-- в новом поле будет рассчитано минимальное значение поля revenue,
-- в зависимости от порядка дат в поле order_data, то есть в новом поле будет
-- присутствовать минимальное значение только от предыдущих значений
      MAX(revenue) OVER (ORDER BY order_data)
-- аналогично с MIN
FROM таблица_1
```

Оконная функция `LAG()` позволяет получить значение поля предыдущей строки в рамках заданного окна и порядка формирования.

```
SELECT LAG(revenue) OVER (ORDER BY order_data)
-- в новом поле итоговой таблицы сформируются предыдущие значения
-- в зависимости от поля с датами order_data
FROM таблица_1
```

Оконная функция `LEAD()` позволяет получить значение поля следующей строки в рамках заданного окна и порядка формирования.

```
SELECT LEAD(revenue) OVER (ORDER BY order_data)
-- в новом поле итоговой таблицы сформируются последующие значения
-- в зависимости от поля с датами order_data
FROM таблица_1
```