

# SQL Задачи и решения

Учебник. Сергей Моисеенко.



◀ назад листать вперед ▶

Типичные проблемы

Коррелирующие подзапросы

## Накопительные итоги стр. 2

Расширение поддержки оконных функций в SQL Server 2012 позволяет решить задачу о накопительных итогах совсем просто.

Применительно к нашей задаче речь идет о следующих появившихся возможностях:

1. Использование сортировки в предложении OVER при применении агрегатных функций.
2. Спецификация диапазона, к значениям которого применяется агрегатная функция. При этом диапазон может быть как ограниченным, так и неограниченным, скажем, от текущей строки до конца или начала отсортированного набора.

Т.е. мы можем получить накопительный итог, упорядочив данные по дате и подсчитав сумму от текущей строки и (неограниченно) выше, причем сделать это с помощью одной функции!

Задачу о накопительных итогах для пункта 2, которая рассматривалась на предыдущей странице, теперь мы можем решить так:

→ Выполнить

Консоль

```
1. SELECT point, date, out,
2.     SUM(out) OVER (PARTITION BY point
3.                   ORDER BY date -- сортировка по дате
4.                   RANGE -- диапазон
5.                   UNBOUNDED -- неограниченный
6.                   PRECEDING -- от текущей строки и выше
7.                   )
8. FROM Outcome_o o
9. WHERE point = 2
10. ORDER BY point, date;
```

Для получения накопительных итогов по каждому пункту отдельно уберем из предыдущего запроса условие отбора по пункту:

→ Выполнить

Консоль

```
1. SELECT point, date, out,
2.     SUM(out) OVER (PARTITION BY point ORDER BY date RANGE UNBOUNDED PRECEDING)
3. FROM Outcome_o o
4. ORDER BY point, date;
```

Представленные здесь решения будут работать в PostgreSQL и Oracle. Что касается MySQL, то там поддержка оконных функций реализована в версии 8.0.

Если нам потребуется подсчитать накопительный итог с учетом не всех предшествующих строк, а, скажем, двух. В этом случае мы можем использовать следующий синтаксис:

→ Выполнить

Консоль

```
1. SELECT point,date, out,
2.     SUM(out) OVER (ORDER BY date,point ROWS BETWEEN 2 PRECEDING AND CURRENT ROW)
3. FROM Outcome_o o
4. ORDER BY date,point;
```

Суммирование происходит в окне, которое задается предложением

```
1. ROWS BETWEEN 2 PRECEDING AND CURRENT ROW
```

Здесь задается диапазон строк (rows) между (between) текущей строкой (current row) и двумя строками выше (2 preceding).

В этом примере рассматриваются все строки таблицы, упорядоченные по date, point (сортировка по point устраняет неоднозначность, поскольку несколько пунктов могут иметь отчетность в один и тот же день).

Рекомендуемые упражнения: **106**

Последние изменения:

Метод наименьших квадратов

Вставка строк в таблицу, содержащую автоинкрементируемое поле  
стр. 5

Сортировка по дням рождения  
стр. 2

Функция STRING\_AGG стр. 2

Функция TRANSLATE

Упражнение 151 (подсказки и решения)

Предикат LIKE

Приложение 2. Список задач  
стр. 2

Приложение 2. Список задач

Футбол



Тэги:

поиск по тэгам

ALL AND AUTO\_INCREMENT AVG  
battles CASE CAST CHAR  
CHARINDEX CHECK classes  
COALESCE CONSTRAINT Convert  
COUNT CROSS APPLY CTE  
DATEADD DATEDIFF DATENAME  
DATEPART DATETIME DDL  
DEFAULT DELETE DISTINCT DML  
EXCEPT EXISTS EXTRACT  
FOREIGN KEY FROM FULL JOIN  
GROUP BY Guadalcanal HAVING  
IDENTITY IN  
INFORMATION\_SCHEMA INNER  
JOIN insert INTERSECT IS NOT  
NULL IS NULL ISNULL laptop LEFT  
LEFT OUTER JOIN LEN maker

[Больше тэгов](#)

Учебник обновлялся  
месяц назад  
<https://exchangesumo.com/obmen/NEPRRUB/> . Штукатурка потолочной поверхности

