SQL Задачи и решения

Учебник. Сергей Моисеенко.

Произведение значений столбца

Почему среди агрегатных функций SQL нет произведения?

Такой вопрос часто задают в профессиональных социальных сетях. Речь идёт о произведении значений столбца таблицы при выполнении группировки. Функции типа **PRODUCT** нет в стандарте языка, и я не знаю СУБД, которая её бы имела. Хорошей же новостью является то, что такую функцию просто выразить через три другие, которые есть в арсенале практически всех серверов БД. Итак.

Пусть требуется перемножить значения столбца value следующей таблицы:

Выполнить Консоль

```
1.  SELECT value FROM (
2.  VALUES(2),(3),(4),(5)
3.  ) X(value);
```

value
2
3
4
5

Воспользуемся следующим свойством логарифмов: логарифм произведения равен сумме логарифмов, для нашего примера это означает

```
1.  ln(2*3*4*5) = ln(2) + ln(3) + ln(4) + ln(5)
```

Если теперь применить обратную к натуральному логарифму (**Ln**) функцию экспоненты (**exp**), то получим

```
1.  exp(ln(2*3*4*5)) = 2*3*4*5 = exp(ln(2) + ln(3) + ln(4) + ln(5))
```

Итак, произведение чисел мы можем заменить выражением, стоящим в равенстве справа. Осталось записать эту формулу на языке SQL, учитывая, что сами числа находятся в столбце value.

Выполнить Консоль

```
1.  SELECT exp(SUM(log(value))) product FROM (
2.  VALUES(2),(3),(4),(5)
3.  ) X(value);
```

product
120

Правильность результата легко проверить устным счетом, или в Excel :-).

Рассмотренное решение не является универсальным. Поскольку логарифм не определен для чисел <=0, то если в столбце появятся такие значения, например,

Выполнить Консоль

```
1.  SELECT exp(SUM(log(value))) product FROM (
2.  VALUES(2),(-3),(4),(-5)
3.  ) X(value);
```

будет получено сообщение об ошибке:

An invalid floating point operation occurred.

(Попытка выполнить недопустимую операцию с плавающей запятой.)

Для учета "недопустимых" значений доработаем наше решение в соответствии со следующим алгоритмом:

1. Если среди значений есть нули, то результатом будет 0.
2. Если число отрицательных значений нечетное, то домножаем произведение абсолютных значений столбца на -1.
3. Если число отрицательных значений четное, то результатом будет произведение абсолютных значений столбца.

Вот решение с комментариями, реализующее этот алгоритм:

Выполнить Консоль

```
 1. WITH T AS(SELECT * FROM (VALUES(-2),(-3),(4),(-5)) X(value)),
 2. P AS (
 3. SELECT SUM(CASE WHEN value<0 THEN 1 ELSE 0 END) neg, -- число отрицательных значений
 4. SUM(CASE WHEN value>0 THEN 1 ELSE 0 END) pos, -- число положительных значений
 5. COUNT(*) total -- общее число значений
 6. FROM T)
 7. SELECT CASE WHEN total <> pos+neg /* есть нули */ THEN 0 ELSE
 8. (CASE WHEN neg%2=1 THEN -1 ELSE +1 END) *exp(SUM(log(abs(value))))
 9. END product
10. FROM T,P
11. WHERE value <> 0
12. GROUP BY neg, pos, total;
```

product
-120

Обратите внимание на условие value <> 0 в последней строке запроса. Его присутствие связано с тем, что, хотя ветвь оператора CASE с вычислением выражения через логарифм не реализуется при наличии нулей среди значений столбца (возвращается 0), SQL Server всё равно вычисляет это выражение и возвращает ошибку.

Сообразительные уже спросили: "А как быть с NULL?"

Действительно, наше решение даёт в этом случае 0. Будем следовать общей логике поведения агрегатных функций - не учитывать NULL. Ниже приводится окончательное решение, которое имеет одно отличие по сравнению с предыдущим решением. Кто догадается какое?

Выполнить Консоль

```
 1. WITH T AS(SELECT * FROM (VALUES(-2),(-3),(4),(-5), (NULL) ) X(value)),
 2. P AS (
 3. SELECT SUM(CASE WHEN value<0 THEN 1 ELSE 0 END) neg, -- число отрицательных значений
 4. SUM(CASE WHEN value>0 THEN 1 ELSE 0 END) pos, -- число положительных значений
 5. COUNT(value) total -- общее число значений
 6. FROM T)
 7. SELECT CASE WHEN total <> pos+neg /* есть нули */ THEN 0 ELSE
 8. (CASE WHEN neg%2=1 THEN -1 ELSE +1 END) *exp(SUM(log(abs(value))))
 9. END
10. product  FROM T,P WHERE value <> 0 GROUP BY neg, pos, total;
```

агрегатные функции LOG EXP произведение

Предыдущая [Агрегатная функция от агрегатной функции]

[Использование в запросе нескольких источников записей] Следующая

Последние изменения:

- 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://pc-service.kz/ . Как и чем резать керамогранитную плитку