

SQL Server solution to simple recursive function

Asked 8 years, 7 months ago Modified 8 years, 7 months ago Viewed 1k times



0



I am looking for a SQL Server solution for a simple recursive formula. In the example, X is my column of numbers and Y is the column I am trying to create with a SQL Query.

I have a list of numbers, denoted X, and wish to produce a special kind of running sum that is not allowed to go less than 0, denoted Y.



Base Case



$$Y_1 = \text{MAX}(X_1, 0)$$

Recursive Rule

$$Y_i = \text{MAX}(X_i + Y_{i-1}, 0)$$

EXAMPLE:

id	X(Input)	Y(Output)
1	15	15
2	-87	0
3	26	26
4	-87	0
5	4	4
6	-19	0
7	34	34
8	-4	30
9	40	70
10	-14	56

[sql](#) [sql-server](#) [recursion](#) [recursive-query](#)

Share Improve this question Follow

edited Mar 18, 2014 at 6:03



[marc_s](#)

714k

171

1315

1434

asked Mar 18, 2014 at 0:49



[user3431083](#)

404

5

19

Do you have a column that specifies the ordering of the rows? SQL tables are inherently unordered, so if this is your entire table, you cannot do what you want. – [Gordon Linoff](#) Mar 18, 2014 at 0:52

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

[Sign up](#)



1

Assuming you have an `id` column that specifies the ordering, I am pretty sure you have to do this with a recursive CTE. The problem is that the "set negative numbers to zero" complicates the situation.

Let me assume that the `id` identifies the ordering.

```
with t as (
    select t.*, row_number() over (order by id) as seqnum
    from table t
),
cte as (
    select X,
        (case when X < 0 then 0 else X end) as Y
    from t
    where id = 1
    union all
    select tnext.X,
        (case when tnext.X + cte.Y < 0 then 0 else tnext.X + cte.Y end) as Y
    from cte join
        t tnext
        on t.id + 1 = tnext.id
)
select *
from cte;
```

Share Improve this answer Follow

answered Mar 18, 2014 at 0:55



Gordon Linoff

1.2m 53 599 743

I successfully implemented this type of solution, but was wondering if it is flexible enough to handle the extension of the problem wherein I add a varchar column "product" and need to keep this same special running positive sum unique to each "product". When ordered by id, the products would occur multiple times and appear randomly in the list. – [user3431083](#) Mar 18, 2014 at 22:30

0

Using a cursor and a table variable to catch the calculated values might be good for performance.

```
declare @T table
(
    id int,
    X int,
    Y int
);

declare @id int;
declare @X int;
declare @Y int;

set @Y = 0;
```

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up



```
order by T.id;

open C;

fetch next from C into @id, @X;
while @@fetch_status = 0
begin
    set @Y = case when @X + @Y < 0 then 0 else @X + @Y end;
    insert into @T(id, X, Y) values (@id, @X, @Y);
    fetch next from C into @id, @X;
end

close C;
deallocate C;

select T.id, T.X, T.Y
from @T as T
order by T.id;
```

[SQL Fiddle](#)

Have a look at [Best approaches for running totals](#) by [Aaron Bertrand](#)

Share Improve this answer Follow

edited May 23, 2017 at 12:11



Community Bot
1 1

answered Mar 18, 2014 at 6:45



[Mikael Eriksson](#)

134k 22 202 274