Collapse All Code: All

EARLIER Function (DAX)

Example See Also Send Feedback

Returns the current value of the specified column in an outer evaluation pass of the mentioned column.

EARLIER is useful for nested calculations where you want to use a certain value as an input and produce calculations based on that input. In Microsoft Excel, you can do such calculations only within the context of the current row; however, in PowerPivot you can store the value of the input and then make calculation using data from the entire table.

EARLIER is mostly used in the context of calculated columns.

Syntax

```
EARLIER(<column>, <number>)
```

Parameters

Term	Definition
column	A column or expression that resolves to a column.
num	(Optional) A positive number to the outer evaluation pass. The next evaluation level out is represented by 1; two levels out is represented by 2 and so on. When omitted default value is 1.

Property Value/Return Value

The current value of row, from **column**, at **number** of outer evaluation passes.

Exceptions

Description of errors

Remarks

EARLIER succeeds if there is a row context prior to the beginning of the table scan. Otherwise it returns an error.

The performance of **EARLIER** might be slow because it theoretically, it might have to perform a number of operations that is close to the total number of rows (in the column) times the same number (depending on the syntax of the expression). For example if you have 10 rows in the column, approximately a 100 operations could be required; if you have 100 rows then close to 10,000 operations might be performed.

Note

In practice, the VertiPaq engine performs optimizations to reduce the actual number of calculations, but you should be cautious when creating formulas that involve recursion.

Example

To illustrate the use of EARLIER, it is necessary to build a scenario that calculates a rank value and then uses that rank value in other calculations.

The following example is based on this simple table, **ProductSubcategory**, which shows the total sales for each ProductSubcategory.

The final table, including the ranking column is shown here.

ProductSubcategoryKey	EnglishProductSubcategoryName	TotalSubcategorySales	SubcategoryRanking
18	Bib-Shorts	$156,167.88	18
26	Bike Racks	$220,720.70	14
27	Bike Stands	$35,628.69	30
28	Bottles and Cages	$59,342.43	24
5	Bottom Brackets	$48,643.47	27
6	Brakes	$62,113.16	23
19	Caps	$47,934.54	28
7	Chains	$8,847.08	35
29	Cleaners	$16,882.62	32
8	Cranksets	$191,522.09	15
9	Derailleurs	$64,965.33	22
30	Fenders	$41,974.10	29
10	Forks	$74,727.66	21
20	Gloves	$228,353.58	12
4	Handlebars	$163,257.06	17
11	Headsets	$57,659.99	25
31	Helmets	$451,192.31	9
32	Hydration Packs	$96,893.78	20
21	Jerseys	$699,429.78	7
33	Lights		36
34	Locks	$15,059.47	33
1	Mountain Bikes	$34,305,864.29	2
12	Mountain Frames	$4,511,170.68	4

35	Panniers		36
13	Pedals	$140,422.20	19
36	Pumps	$12,695.18	34
2	Road Bikes	$40,551,696.34	1
14	Road Frames	$3,636,398.71	5
15	Saddles	$52,526.47	26
22	Shorts	$385,707.80	10
23	Socks	$28,337.85	31
24	Tights	$189,179.37	16
37	Tires and Tubes	$224,832.81	13
3	Touring Bikes	$13,334,864.18	3
16	Touring Frames	$1,545,344.02	6
25	Vests	$240,990.04	11
17	Wheels	$648,240.04	8

Creating a Rank Value

One way to obtain a rank value for a given value in a row is to count the number of rows, in the same table, that have a value larger (or smaller) than the one that is being compared. This technique returns a blank or zero value for the highest value in the table, whereas equal values will have the same rank value and next value (after the equal values) will have a non consecutive rank value. See the sample below.

A new calculated column, **SubCategorySalesRanking**, is created by using the following formula.

Copy Code

```
= COUNTROWS(FILTER(ProductSubcategory, EARLIER(ProductSubcategory[TotalSubcategorySales])<ProductSubcategory[TotalSubcategorySales]))+1
```

The following steps describe the method of calculation in more detail.

1. The **EARLIER** function gets the value of *TotalSubcategorySales* for the current row in the table. In this case, because the process is starting, it is the first row in the table
2. **EARLIER**([*TotalSubcategorySales*]) evaluates to $156,167.88, the current row in the outer loop.
3. The **FILTER** function now returns a table where all rows have a value of *TotalSubcategorySales* larger than $156,167.88 (which is the current value for **EARLIER**).
4. The **COUNTROWS** function counts the rows of the filtered table and assigns that value to the new calculated column in the current row plus 1. Adding 1 is needed to prevent the top ranked value from become a Blank.
5. The calculated column formula moves to the next row and repeats steps 1 to 4. These steps are repeated until the end of the table is reached.

The **EARLIER** function will always get the value of the column prior to the current table operation. If you need to get a value from the loop before that, set the second argument to 2.

See Also

Reference

EARLIEST Function (DAX)

Other Resources

Filter Functions (DAX)