

Simplify your DAX with window functions

Greg Strzyminski

Power BI Next Step,
13.09.2024

KEARNEY

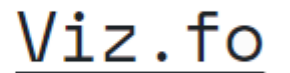


Big thank you to our great sponsors

Premium



Standard



Greg Strzyminski

KEARNEY

Senior BI Developer



@GregWorksPowerBI



/greg-strzyminski



/greg-strzyminski



After this session you will:

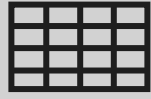
- Understand the concept and syntax of window functions.
- Be familiar with a number of common use-cases for window functions.
- Be equipped with the technical knowledge needed to implement the window functions in your own reports literally the next day.

Window functions in a nutshell

Let's see a 1-slide
summary of window
functions in DAX

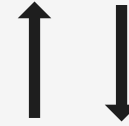
Window functions return a row or a set of rows from a sorted table.

We need 3 elements to make window functions work



Table

First, we need a table to make the window functions work. The table can be a one from the model, virtual, calculated etc. SUMMARIZE() and ALL() family functions come in handy here.



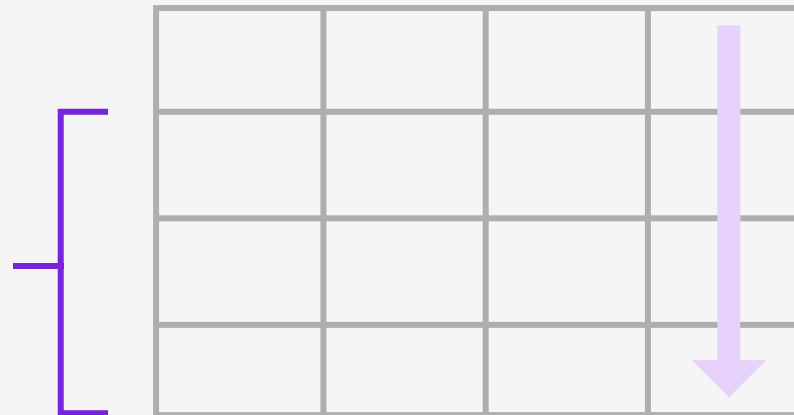
Sorting order

We need to know how to sort the table to make the row selection logical. The sorting can be defined by one or more columns existing in the table or measures that aren't part of the table.



Row(s) to return

Last, we must define which row(s) are to be selected. We can use absolute (e.g. first, second, last row) or relative (row after/before the current one) positions depending on the window function.



Use cases

Common scenarios to leverage window functions in DAX

Use-cases for window functions

Index

Use case 1:
We want to calculate sales of the **best country**

We are iterating any of these rows

Country	Sales
United States	\$4,756M
China	\$1,064M
Germany	\$663M
France	\$434M
United Kingdom	\$221M
Canada	\$176M
Japan	\$163M
Australia	\$79M
India	\$78M
Russia	\$71M
Italy	\$56M
Iran	\$52M
Turkmenistan	\$52M
Syria	\$45M
Pakistan	\$44M
South Korea	\$37M
Thailand	\$36M

Use-cases for window functions

Index

Use case 1:
We want to calculate sales of the **best country**

We are iterating any of these rows

Country	Sales	Sales of best country
United States	\$4,756M	\$4,756M
China	\$1,064M	\$4,756M
Germany	\$663M	\$4,756M
France	\$434M	\$4,756M
United Kingdom	\$221M	\$4,756M
Canada	\$176M	\$4,756M
Japan	\$163M	\$4,756M
Australia	\$79M	\$4,756M
India	\$78M	\$4,756M
Russia	\$71M	\$4,756M
Italy	\$56M	\$4,756M
Iran	\$52M	\$4,756M
Turkmenistan	\$52M	\$4,756M
Syria	\$45M	\$4,756M
Pakistan	\$44M	\$4,756M
South Korea	\$37M	\$4,756M
Thailand	\$36M	\$4,756M

Use-cases for window functions

Index

Use case 1:
We want to calculate sales of the **best country**

Use case 2:
We want to calculate sales of the **worst country**

Use case 3:
We want to calculate sales of the **3rd best country**

We are iterating any of these rows

Country	Sales
United States	\$4,756M
China	\$1,064M
Germany	\$663M
France	\$434M
United Kingdom	\$221M
Canada	\$176M
Japan	\$163M
Australia	\$79M
India	\$78M
Russia	\$71M
Italy	\$56M
Iran	\$52M
Turkmenistan	\$52M
Syria	\$45M
Pakistan	\$44M
South Korea	\$37M
Thailand	\$36M

Use-cases for window functions

Index

Use case 4:
We want to calculate sales of the country with the highest share of returns

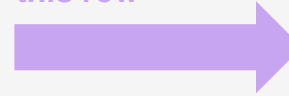
Country	Sales	Share of returns	Sales of country with most returns
United States	\$4,756M	260%	\$56M
China	\$1,064M	241%	\$56M
Germany	\$663M	291%	\$56M
France	\$434M	294%	\$56M
United Kingdom	\$221M	301%	\$56M
Canada	\$176M	272%	\$56M
Japan	\$163M	244%	\$56M
Australia	\$79M	236%	\$56M
India	\$78M	246%	\$56M
Russia	\$71M	302%	\$56M
Italy	\$56M	329%	\$56M
Iran	\$52M	259%	\$56M
Turkmenistan	\$52M	251%	\$56M
Syria	\$45M	219%	\$56M
Pakistan	\$44M	249%	\$56M
South Korea	\$37M	222%	\$56M
Thailand	\$36M	242%	\$56M
Bhutan	\$30M	206%	\$56M
Taiwan	\$26M	249%	\$56M
Armenia	\$26M	263%	\$56M

Use-cases for window functions

Offset

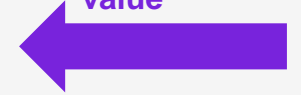
Use case 5:
We want to calculate sales of the **previous country** relative to the **current row**

We are iterating this row



Country	Sales
United States	\$4,756M
China	\$1,064M
Germany	\$663M
France	\$434M
United Kingdom	\$221M
Canada	\$176M
Japan	\$163M
Australia	\$79M
India	\$78M
Russia	\$71M
Italy	\$56M
Iran	\$52M
Turkmenistan	\$52M
Syria	\$45M
Pakistan	\$44M
South Korea	\$37M
Thailand	\$36M

We want this value



Use-cases for window functions

Offset

Use case 5:
We want to calculate sales of the **previous country** relative to the **current row**

We are iterating this row



Country	Sales	Sales of previous country
United States	\$4,756M	
China	\$1,064M	\$4,756M
Germany	\$663M	\$1,064M
France	\$434M	\$663M
United Kingdom	\$221M	\$434M
Canada	\$176M	\$221M
Japan	\$163M	\$176M
Australia	\$79M	\$163M
India	\$78M	\$79M
Russia	\$71M	\$78M
Italy	\$56M	\$71M
Iran	\$52M	\$56M
Turkmenistan	\$52M	\$52M
Syria	\$45M	\$52M
Pakistan	\$44M	\$45M
South Korea	\$37M	\$44M
Thailand	\$36M	\$37M

Use-cases for window functions

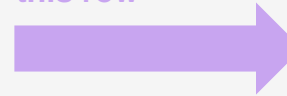
Offset

Use case 6:
We want to calculate sales of the **previous country** relative to the **current row**

Use case 7:
We want to calculate sales of the **next country** relative to the **current row**

Use case 8:
We want to calculate sales of the **country that's 3 places before** the **current row**

We are iterating this row



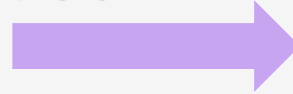
Country	Sales
United States	\$4,756M
China	\$1,064M
Germany	\$663M
France	\$434M
United Kingdom	\$221M
Canada	\$176M
Japan	\$163M
Australia	\$79M
India	\$78M
Russia	\$71M
Italy	\$56M
Iran	\$52M
Turkmenistan	\$52M
Syria	\$45M
Pakistan	\$44M
South Korea	\$37M
Thailand	\$36M

Use-cases for window functions

Window

Use case 7:
We want to calculate sales of **all countries** that are before the **current row** or are the current row

We are iterating this row



Country	Sales
United States	\$4,756M
China	\$1,064M
Germany	\$663M
France	\$434M
United Kingdom	\$221M
Canada	\$176M
Japan	\$163M
Australia	\$79M
India	\$78M
Russia	\$71M
Italy	\$56M
Iran	\$52M
Turkmenistan	\$52M
Syria	\$45M
Pakistan	\$44M
South Korea	\$37M
Thailand	\$36M

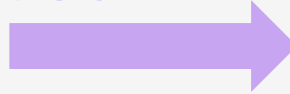
We want these values

Use-cases for window functions

Window

Use case 7:
We want to calculate sales of **all countries** that are before the **current row** or are the current row

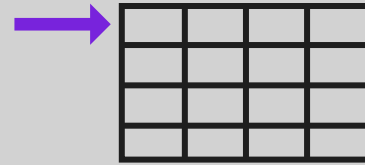
We are iterating this row



Country	Sales	Sales cumulative by country
United States	\$4,756M	\$4,756M
China	\$1,064M	\$5,820M
Germany	\$663M	\$6,483M
France	\$434M	\$6,917M
United Kingdom	\$221M	\$7,138M
Canada	\$176M	\$7,314M
Japan	\$163M	\$7,477M
Australia	\$79M	\$7,556M
India	\$78M	\$7,634M
Russia	\$71M	\$7,705M
Italy	\$56M	\$7,760M
Iran	\$52M	\$7,813M
Turkmenistan	\$52M	\$7,864M
Syria	\$45M	\$7,910M
Pakistan	\$44M	\$7,954M
South Korea	\$37M	\$7,991M
Thailand	\$36M	\$8,028M

Use-cases for window functions

Index

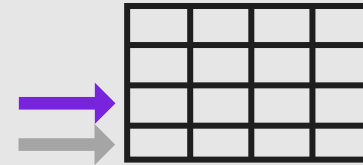


Get a set row from a table using its **absolute position**.

Use-cases:

- KPI value of best item
- KPI value of 2nd best item
- KPI value of worst item
- Newest KPI value
- Oldest KPI value
- Previous KPI value

Offset

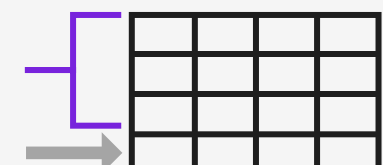


Get a set row from a table that is **relative** to the **current row**.

Use-cases:

- KPI value of previous item relative to the current item
- KPI value of next item relative to the current item
- KPI value of 3rd next item relative to the current item

Window



Get a set of **rows** from a table that are **relative** to the current row **or** using their **absolute** position.

Use-cases:

- KPI value of all previous items relative to the current item
- Cumulative calculations
- Pareto charts
- Rolling averages

Syntax of window functions

Window

Use case 7:

We want to calculate sales of **all countries** that are before the **current row** or are the current row


How do I even read this?

```
Sales cumulative by country =  
CALCULATE(  
    [Sales],  
    WINDOW(  
        1, ABS,  
        0, REL,  
        ALLSELECTED(Geography[RegionCountryName]),  
        ORDERBY(  
            [Sales], DESC  
        )  
    )  
)
```

Syntax of window functions

Choose rows to return in absolute or relative manner

We are iterating this row



Country	Sales	
United States	\$4,756M	1
China	\$1,064M	2
Germany	\$663M	3
France	\$434M	...
United Kingdom	\$221M	...
Canada	\$176M	-3
Japan	\$163M	-2
Australia	\$79M	-1
India	\$78M	0
Russia	\$71M	+1
Italy	\$56M	+2
Iran	\$52M	+3
Turkmenistan	\$52M	...
Syria	\$45M	...
Pakistan	\$44M	...
South Korea	\$37M	...
Thailand	\$36M	...
Bhutan	\$30M	-2
Taiwan	\$26M	-1

Country with highest sales =

```
INDEX(  
  1,  
  ALLSELECTED(Geography[RegionCountryName]),  
  ORDERBY(  
    [Sales], DESC  
  )  
)
```

Previous country by sales =

```
OFFSET(  
  -1,  
  ALLSELECTED(Geography[RegionCountryName]),  
  ORDERBY(  
    [Sales], DESC  
  )  
)
```

Syntax of window functions

Index

Use case 1:
We want to calculate sales of the **best country**

Country with highest sales =

```
INDEX(  
  1,  
  ALLSELECTED(Geography[RegionCountryName]),  
  ORDERBY(  
    [Sales], DESC  
  )  
)
```

Please give me the row which is the first one from a table with 1 column containing country name ordered by measure [Sales] in a descending order

Country	Sales
United States	\$4,756M
China	\$1,064M
Germany	\$663M
France	\$434M
United Kingdom	\$221M
Canada	\$176M

The above equals:

Return the country which has the highest sales

Syntax of window functions

Index

Use case 1:
We want to calculate sales of the **best country**

```
Sales of country with highest sales =  
CALCULATE(  
    [Sales],  
    INDEX(  
        1,  
        ALLSELECTED(Geography[RegionCountryName]),  
        ORDERBY(  
            [Sales], DESC  
        )  
    )  
)
```

Country	Sales	Sales of country with highest sales
United States	\$4,756M	\$4,756M
China	\$1,064M	\$4,756M
Germany	\$663M	\$4,756M
France	\$434M	\$4,756M
United Kingdom	\$221M	\$4,756M

The above equals:

Return the sales of the country which has the highest sales

Syntax of window functions

Offset

Use case 5:
We want to calculate sales of the **previous country** relative to the **current row**

Sales of previous country =

```
CALCULATE(  
    [Sales],  
    OFFSET( -----  
        -1, -----  
        ALLSELECTED(Geography[RegionCountryName]), -----  
        ORDERBY(  
            [Sales], DESC -----  
        )  
    )  
)
```

Please give me the row which is previous relative to the current one from a table with 1 column containing country name ordered by measure [Sales] in a descending order

The above equals:

Return the Sales of the country which is previous in terms of Sales to the current one

Country	Sales	Sales of previous country
United States	\$4,756M	
China	\$1,064M	\$4,756M
Germany	\$663M	\$1,064M
France	\$434M	\$663M
United Kingdom	\$221M	\$434M

Syntax of window functions

Window

Use case 7:

We want to calculate sales of **all countries** that are before the **current row** or are the current row

Sales cumulative by country =

```
CALCULATE(  
    [Sales],  
    WINDOW(  
        1, ABS,  
        0, REL,  
        ALLSELECTED(Geography[RegionCountryName]),  
        ORDERBY(  
            [Sales], DESC  
        )  
    )  
)
```

- > Please give me the set of rows which starts at the first row of the table and ends at the current row
- > from a table with 1 column containing country name
- > ordered by measure [Sales] in a descending order

The above equals:

Return the cumulative Sales of the current country

Country	Sales	Sales cumulative by country
United States	\$4,756M	\$4,756M
China	\$1,064M	\$5,820M
Germany	\$663M	\$6,483M

Syntax of window functions

Window

Use case X:
What would this **syntax** mean?

```
Mysterious window function 🤖 =  
CALCULATE(  
    [Sales],  
    WINDOW(  
        1, ABS,  
        1, ABS,  
        ALLSELECTED(Geography[RegionCountryName]),  
        ORDERBY(  
            [Sales], DESC  
        )  
    )  
)
```

Use case 1:
We want to calculate sales
of the **best country**

```
Sales of country with highest sales =  
CALCULATE(  
    [Sales],  
    INDEX(  
        1,  
        ALLSELECTED(Geography[RegionCountryName]),  
        ORDERBY(  
            [Sales], DESC  
        )  
    )  
)
```

Country	Sales	Sales of country with highest sales	Mysterious window function 🤖
United States	\$4,756M	\$4,756M	\$4,756M
China	\$1,064M	\$4,756M	\$4,756M
Germany	\$663M	\$4,756M	\$4,756M
France	\$434M	\$4,756M	\$4,756M
United Kingdom	\$221M	\$4,756M	\$4,756M
Canada	\$176M	\$4,756M	\$4,756M

Live demos

**Let's see some live
examples in Power BI**

Key takeaways

1. Window functions enable you to simplify and optimize your DAX code.
2. Typical use cases include references to previous/next or first/last item, running averages and cumulative sums.
3. Window functions are more optimized than the standard DAX code you would write for the same use-case
4. Syntax of window functions is not that overwhelming when you get a grasp of it.

Bonus Fun fact: window functions in DAX are a foundation for the Visual Calculations.

Q&A

Thank you

Grzegorz Strzymiński

Senior BI Developer

grzegorz.strzyminski@kearney.com

+48 784 032 770

Warsaw

Stay connected with Kearney



This document is exclusively intended for selected client employees. Distribution, quotations and duplications – even in the form of extracts – for third parties is only permitted upon prior written consent of Kearney.

Kearney used the text and charts compiled in this report in a presentation; they do not represent a complete documentation of the presentation.

KEARNEY



Raffle prizes from the sponsors



**Power BI
Next Step**

Feedback form

