

Power BI

Basic to Advanced

Part - 8

AGENDA

- Table Manipulation functions
- Text functions
- Time Intelligence functions
- New DAX functions

6) Table Manipulation functions

- These functions return a table or manipulate existing tables.

➤ DISTINCT (column)

Returns a one-column table that contains the distinct values from the specified column. In other words, duplicate values are removed and unique values are returned.

Syntax: DISTINCT(<column>)

E.g.: =

COUNTROWS(DISTINCT(InternetSales_USD[CustomerKey]))

- There is another version of the DISTINCT function, DISTINCT (table), that returns a table by removing duplicate rows from another table or expression.

➤ SELECTCOLUMNS

Returns a table with selected columns from the table and new columns specified by the DAX expressions.

Syntax: SELECTCOLUMNS(<Table>, [<Name>], <Expression>, <Name>], ...)

E.g.: = SELECTCOLUMNS(Customer, "Country, State",
[Country]&", "&[State])

➤ SUMMARIZE

Returns a summary table for the requested totals over a set of groups.

Syntax: SUMMARIZE (<table>, <groupBy_columnName>[,
<groupBy_columnName>]...[, <name>, <expression>]...)

E.g.: =SUMMARIZE(ResellerSales_USD
 , DateTime[CalendarYear]
 , ProductCategory[ProductCategoryName]
 , "Sales Amount (USD)",
SUM(ResellerSales_USD[SalesAmount_USD])
 , "Discount Amount (USD)",
SUM(ResellerSales_USD[DiscountAmount])
)

➤ TOPN

Returns the top N rows of the specified table.

Syntax: TOPN(<N_Value>, <Table>, <OrderBy_Expression>,
[<Order>[, <OrderBy_Expression>, [<Order>]]...])

```
E.g.: = SUMX(  
    TOPN(  
        10,  
        SUMMARIZE(  
            InternetSales,  
            InternetSales[ProductKey],  
            "TotalSales", SUM(InternetSales[SalesAmount])  
        ),  
        [TotalSales], DESC  
    ),  
    [TotalSales]  
)
```

There are other table manipulation functions such as GROUPBY, ROW, UNION, etc. To know more about them refer to the following link. [Table manipulation functions \(DAX\) - DAX | Microsoft Learn](#)

7) Text functions

- DAX includes a set of text functions based on the library of string functions in Excel, which have been modified to work with tables and columns in tabular models.

➤ CONCATENATE

Joins two text strings into one text string.

Syntax: CONCATENATE(<text1>, <text2>)

E.g.: CONCATENATE(Customer[LastName],
CONCATENATE(", ", Customer[FirstName]))

- The CONCATENATE function in DAX accepts only two arguments, whereas the Excel CONCATENATE function accepts up to 255 arguments. If you need to add more arguments, you can use the ampersand (&) operator.

➤ FIND

Returns the starting position of one text string within another text string. FIND is case-sensitive.

Syntax: FIND(<find_text>, <within_text>[, [<start_num>][,
<NotFoundValue>]])

E.g.: EVALUATE

CALCULATETABLE (

ADDCOLUMNS (

TOPN (10, SUMMARIZE('Reseller', [Reseller], [Business Type])),

"Position of Bike", FIND ("Bike", 'Reseller'[Reseller], 1, BLANK ())

),

'Reseller'[Business Type] IN { "Specialty Bike Shop", "Value Added Reseller", "Warehouse"}

)

➤ SEARCH

Returns the number of the character at which a specific character or text string is first found, reading left to right.

Search is case-insensitive and accent sensitive.

Syntax: SEARCH(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])

E.g.: EVALUATE

CALCULATETABLE (

ADDCOLUMNS (

TOPN (10, SUMMARIZE('Reseller', [Reseller], [Business Type])),

"Position of cycle", SEARCH ("cycle",
'Reseller'[Reseller], 1, BLANK ())

),

'Reseller'[Business Type] IN { "Specialty Bike Shop", "Value Added Reseller", "Warehouse"})

- The search function is accent-sensitive. Searching for "á" will find the first occurrence of 'á' but no occurrences of 'a', 'à', or the capitalized versions 'A', 'Á'. You can use the SEARCH function to determine the location of a character or text string within another text string, and then use the MID function to return the text or use the REPLACE function to change the text.

➤ FORMAT

Converts a value to text according to the specified format.

Syntax: FORMAT(<value>, <format_string>[, <locale_name>])

E.g.: = FORMAT(12345.67, "General Number")

= FORMAT(12345.67, "Currency")

= FORMAT(12345.67, "Fixed")

= FORMAT(12345.67, "Standard")

= FORMAT(12345.67, "Percent")

= FORMAT(12345.67, "Scientific")

- Using FORMAT changes a measure result to a text data type. If the measure result is originally of numeric data type, then with FORMAT, the measure can't be used on visuals where the values section requires a numeric data type, like with charts. In Power BI, you can alternatively use Dynamic format strings for measures specify a conditional format string that maintains the numeric data type of the measure.

➤ LEFT

Returns the specified number of characters from the start of a text string.

Syntax: LEFT(<text>, <num_chars>)

E.g.: =

CONCATENATE(LEFT('Reseller'[ResellerName],LEFT(GeographyKey,3))

- If the num_chars argument is a number that is larger than the number of characters available, the function returns the maximum characters available and does not raise an error.

➤ MID

Returns a string of characters from the middle of a text string, given a starting position and length.

Syntax: MID(<text>, <start_num>, <num_chars>)

E.g.: MID("abcde",2,3))

➤ REPLACE

Replaces part of a text string, based on the number of characters you specify, with a different text string.

Syntax: REPLACE(<old_text>, <start_num>, <num_chars>, <new_text>)

E.g.: = REPLACE('New Products'[Product Code],1,2,"OB")

➤ SUBSTITUTE

Replace existing text with new text in a text string.

Syntax: SUBSTITUTE(<text>, <old_text>, <new_text>, <instance_num>)

E.g.: = SUBSTITUTE([Product Code], "NW", "PA")

- Use the SUBSTITUTE function when you want to replace specific text in a text string; use the REPLACE function when you want to replace any text of variable length that occurs in a specific location in a text string.
- The SUBSTITUTE function is case-sensitive. If the case does not match between text and old_text, SUBSTITUTE will not replace the text.

There are other text functions such as TRIM, UPPER, LEN, etc. To know more about them refer to the following link.

[Text functions \(DAX\) - DAX | Microsoft Learn](#)

8) Time Intelligence functions

- DAX includes time-intelligence functions that enable you to manipulate data using periods, including days, months, quarters, and years, and then build and compare calculations over those periods. Before using any time-intelligence functions, make sure to mark one of the tables containing the date column as a Date Table.

➤ DATEADD

Returns a table that contains a column of dates, shifted either forward or backward in time by the specified number of intervals from the dates in the current context.

Syntax: DATEADD(<dates>,<number_of_intervals>,<interval>)

E.g.: = DATEADD(DateTime[DateKey],-1,year)

➤ DATESBETWEEN

Returns a table that contains a column of dates that begins with a specified start date and continues until a specified end date. This function is suited to pass as a filter to the CALCULATE function. Use it to filter an expression by a custom date range.

Syntax: DATESBETWEEN(<Dates>, <StartDate>, <EndDate>)

E.g.: Customers LTD =

```
CALCULATE(
    DISTINCTCOUNT(Sales[CustomerKey]),
    DATESBETWEEN(
        'Date'[Date],
        BLANK(),
        MAX('Date'[Date])
    )
)
```

➤ DATESINPERIOD

Returns a table that contains a column of dates that begins with a specified start date and continues for the specified number and type of date intervals. This function is suited to pass as a filter to the CALCULATE function. Use it to filter an expression by standard date intervals such as days, months, quarters, or years.

Syntax: DATESINPERIOD(<dates>, <start_date>,
<number_of_intervals>, <interval>)

E.g.: Revenue PY =

```
CALCULATE(  
    SUM(Sales[Sales Amount]),  
    DATESINPERIOD(  
        'Date'[Date],  
        MAX('Date'[Date]),  
        -1,  
        YEAR  
    )  
)
```

➤ SAMEPERIODLASTYEAR

Returns a table that contains a column of dates shifted one year back in time from the dates in the specified dates column, in the current context.

Syntax: SAMEPERIODLASTYEAR(<dates>)

E.g.: =

```
CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]),  
    SAMEPERIODLASTYEAR(DateTime[DateKey]))
```

There are other time intelligence functions such as DATESYTD, ENDOFYEAR, TOTALYTD, etc. To know more about them refer to the following link. [Time intelligence functions \(DAX\) - DAX | Microsoft Learn](#)

9) New DAX functions

DAX is continuously being improved with new functions and functionality to support new features. New functions and updates are included in service, application, and tool updates which in most cases are monthly.

While functions and functionality are being updated all the time, only those updates that have a visible and functional change exposed to users are described in the documentation. New functions and updates to existing functions within the past year are shown here.

Not all functions are supported in all versions of Power BI Desktop, Analysis Services, and Power Pivot in Excel. New and updated functions are typically first introduced in Power BI Desktop, and then later in Analysis Services, Power Pivot in Excel, and tools

To know more about them refer to the following link. [New DAX functions - DAX | Microsoft Learn](#)

Thank you!
