

Power BI - DAX - p3 - Lecture 17

DAX SYNTAX

MEASURE NAME

- Measures are always surrounded by brackets (i.e. [Total Quantity]) when referenced in formulas, so spaces are OK

Total Quantity: = SUM(Transactions[quantity])

FUNCTION NAME

- Calculated columns don't always use functions, but measures do:
 - In a Calculated Column, =Transactions[quantity] returns the value from the quantity column in each row (since it evaluates one row at a time).
 - In a Measure, = Transactions[quantity] will return an error since Power BI doesn't know how to translate that as a single value - you need some sort of aggregation

Referenced TABLE NAME

Referenced COLUMN NAME

This is a "fully qualified" column, since it's preceded by the table name.

NOTE: Table names with spaces must be surrounded by single quotes:

- Without a space: Transactions[quantity]
- With a space: 'Transactions Table'[quantity]

PRO TIP:

Column references use fully qualified names (i.e. 'Table'[Column])

Measure references just use the measure name (i.e. [Measure]) and can be called by typing an open square bracket "["

DAX OPERATORS



Arithmetic Operator	Meaning	Example
+	Addition	2 + 7
-	Subtraction	5 - 3
*	Multiplication	2 * 6
/	Division	4 / 2
^	Exponent	2 ^ 5

Comparison Operator	Meaning	Example
=	Equal to	[City] = "Boston"
>	Greater than	[Quantity] > 10
<	Less than	[Quantity] < 10
>=	Greater than or equal to	[Unit Price] >= 2.5
<=	Less than or equal to	[Unit Price] <= 2.5
<>	Not equal to	[Country] <> "Mexico"

"Important"



Text/Logical Operator	Meaning	Example
&	Concatenates two values to produce one text string	[City] & " " & [State]
&&	Create an AND condition between two logical expressions	([State] = "MA") && ([Quantity] > 10)
(double pipe)	Create an OR condition between two logical expressions	([State] = "MA") ([State] = "CT")
IN	Creates a logical OR condition based on a given list (using curly brackets)	'Store Lookup'[State] IN { "MA", "CT", "NY" }

It replaces multiple OR Logic.

COMMON FUNCTION CATEGORIES

MATH & STATS Functions	LOGICAL Functions	TEXT Functions	FILTER Functions	TABLE Functions	DATE & TIME Functions	RELATIONSHIP Functions
<p>Functions used for aggregation or iterative, row-level calculations</p> <p>Common Examples:</p> <ul style="list-style-type: none"> • SUM • AVERAGE • MAX/MIN • DIVIDE • COUNT/COUNTA • COUNTROWS • DISTINCTCOUNT <p>Iterator Functions:</p> <ul style="list-style-type: none"> • SUMX • AVERAGEX • MAXX/MINX • RANKX • COUNTX 	<p>Functions that use conditional expressions (IF/THEN statements)</p> <p>Common Examples:</p> <ul style="list-style-type: none"> • IF • IFERROR • AND • OR • NOT • SWITCH • TRUE • FALSE 	<p>Functions used to manipulate text strings or value formats</p> <p>Common Examples:</p> <ul style="list-style-type: none"> • CONCATENATE • COMBINEVALUES • FORMAT • LEFT/MID/RIGHT • UPPER/LOWER • LEN • SEARCH/FIND • REPLACE • SUBSTITUTE • TRIM 	<p>Functions used to manipulate table and filter contexts</p> <p>Common Examples:</p> <ul style="list-style-type: none"> • CALCULATE • FILTER • ALL • ALLEXCEPT • ALLSELECTED • KEEPFILTERS • REMOVEFILTERS • SELECTEDVALUE 	<p>Functions that create or manipulate tables and output tables vs. scalar values</p> <p>Common Examples:</p> <ul style="list-style-type: none"> • SUMMARIZE • ADDCOLUMNS • GENERATESERIES • DISTINCT • VALUES • UNION • INTERSECT • TOPN 	<p>Functions used to manipulate date & time values or handle time intelligence calculations</p> <p>Common Examples:</p> <ul style="list-style-type: none"> • DATE • DATEDIFF • YEARFRAC • YEAR/MONTH • DAY/HOUR • TODAY/NOW • WEEKDAY • WEEKNUM • NETWORKDAYS <p>Time Intelligence:</p> <ul style="list-style-type: none"> • DATESYTD • DATESMTD • DATEADD • DATESBETWEEN 	<p>Functions used to manage & modify table relationships</p> <p>Common Examples:</p> <ul style="list-style-type: none"> • RELATED • RELATEDTABLE • CROSSFILTER • USERELATIONSHIP

BASIC MATH & STATS FUNCTIONS

SUM

Evaluates the sum of a column

=SUM(Column**Name**)

AVERAGE

Returns the average (arithmetic mean) of all the numbers in a column

=AVERAGE(Column**Name**)

MAX

Returns the largest value in a column or between two scalar expressions

=MAX(Column**Name**OrScalar1, [Scalar2])

MIN

Returns the smallest value in a column or between two scalar expressions

=MIN(Column**Name**OrScalar1, [Scalar2])

DIVIDE

Performs division and returns the alternate result (or blank) if DIV/0

=DIVIDE(Numerator, Denominator, [AlternateResult])

COUNTING FUNCTIONS

COUNT

Counts the number of non-empty cells in a column(excluding Boolean values)

=COUNT(Column**Name**)

COUNTA

Counts the number of non-empty cells in a column (including Boolean values)

=COUNTA(Column**Name**)

DISTINCT COUNT

Counts the number of distinct values in a column

=DISTINCTCOUNT(Column**Name**)

COUNTROWS

Counts the number of rows in the specified table, or a table defined by an expression

=COUNTROWS([Table])

ASSIGNMENT: MATH & STATS

1. Create a measure named Total Customers, to calculate the number of distinct Adventure Works customers who made a transaction.

2. Create a measure named Return Rate, defined as quantity returned divided by quantity sold.

Total Customer = DISTINCTCOUNT('Sales Data'[CustomerKey])

17416

Total Customer

Return Rate = Total Returns / Total Orders

Total Returns = SUM('Returns Data'[ReturnQuantity])

1828

Total Returns

Total Orders = SUM('Sales Data'[OrderQuantity])

84174

Total Orders

Return Rate

Measure Table

Structure

/ 1 Return Rate = DIVIDE([Total Returns],[Total Orders],"No Returns")

Format Percentage \$ % , 2

Formatting

Data category Un

Properties

2.17%

Return Rate

BASIC LOGICAL FUNCTIONS

IF

Checks if a given condition is met and returns one value if the condition is TRUE, and another if the condition is FALSE

=IF(LogicalTest, ResultIfTrue,
[ResultIfFalse])

IFERROR

Evaluates an expression and returns a specified value if it returns an error, otherwise returns the expression itself

=IFERROR(Value, ValueIfError)

SWITCH

Evaluates an expression against a list of values and returns one of multiple possible expressions

=SWITCH(Expression, Value1,
Result1, ..., [Else])

AND

Checks whether both arguments are TRUE to return TRUE, otherwise returns FALSE

=AND(Logical1, Logical2)

OR

Checks whether any argument is TRUE to return TRUE, otherwise returns FALSE

=OR(Logical1, Logical2)

Note: Use the && and || operators to include more than two conditions

SWITCH

SWITCH - Evaluates an expression against a list of values and returns one of multiple possible expressions

=SWITCH(Expression, Value1, Result1, ..., [Else])

Any DAX expression that returns a single scalar value, evaluated multiple times.

Examples:

- Calendar[Month ID]
- 'Product Lookup'[category]

List of values produced by the expression, each paired with a result to return for rows/cases that match.

Examples:

```
=SWITCH( Calendar[Month ID],  
1, "January",  
2, "February"
```

Value returned if the expression doesn't match any value argument

PRO TIP

SWITCH(TRUE) is a common DAX pattern to replace multiple nested IF statements

```
Month Number (DAX) =  
IF(  
    'Calendar Lookup'[Month Name] = "January" , "1",  
    IF(  
        'Calendar Lookup'[Month Name] = "February" , "2",  
        IF(  
            'Calendar Lookup'[Month Name] = "March" , "3",  
            IF('Calendar Lookup'[Month Name] = "April" , "4", "Other"  
        )))
```

Nested If , Can easily be handle with Switch Statement.

```
Month Number (Switch) = SWITCH('Calendar Lookup'[Month Name] ,  
    "January" , "1",  
    "February" , "2",  
    "March" , "3",  
    "April" , "4",  
    "May" , "5",  
    "June" , "6",  
    "July" , "7",  
    "August" , "8",  
    "September" , "9",  
    "October" , "10",  
    "November" , "11",  
    "December" , "12",  
    "Other")
```