

A decorative graphic consisting of several overlapping diamond and triangular shapes in teal, yellow, and green colors, located in the top right and bottom left corners of the slide.

“

# COMMON DAX FUNCTION

# DAX Meet

**MEASURE NAME**

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

Referenced  
**TABLE NAME**

Referenced  
**COLUMN NAME**


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)

**Note:** This is a "fully qualified" column, since it's preceded by the table name -- table names with spaces must be surrounded by **single quotes**:

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



**PRO TIP:**

For **column** references, use the fully qualified name (i.e. Table[Column])  
For **measure** references, just use the measure name (i.e. [Measure])

# DAX Operators

## Arithmetic operators

To perform basic mathematical operations such as addition, subtraction, or multiplication; combine numbers; and produce numeric results, use the following arithmetic operators.

Arithmetic operator	Meaning	Example
+ (plus sign)	Addition	3+3
– (minus sign)	Subtraction or sign	3–1–1
* (asterisk)	Multiplication	3*3
/ (forward slash)	Division	3/3
^ (caret)	Exponentiation	16^4

## Comparison operators

You can compare two values with the following operators. When two values are compared by using these operators, the result is a logical value, either TRUE or FALSE.

Comparison operator	Meaning	Example
=	Equal to	[Region] = "USA"
==	Strict equal to	[Region] == "USA"
>	Greater than	[Sales Date] > "Jan 2009"
<	Less than	[Sales Date] < "Jan 1 2009"
>=	Greater than or equal to	[Amount] >= 20000
<=	Less than or equal to	[Amount] <= 100
<>	Not equal to	[Region] <> "USA"

## Text concatenation operator

Use the ampersand (&) to join, or concatenate, two or more text strings to produce a single piece of text.

Text operator	Meaning	Example
& (ampersand)	Connects, or concatenates, two values to produce one continuous text value	[Region] & ", " & [City]

## Logical operators

Use logical operators (&&) and (||) to combine expressions to produce a single result.

Text operator	Meaning	Examples
&& (double ampersand)	Creates an AND condition between two expressions that each have a Boolean result. If both expressions return TRUE, the combination of the expressions also returns TRUE; otherwise the combination returns FALSE.	([Region] = "France") && ([BikeBuyer] = "yes"))
(double pipe symbol)	Creates an OR condition between two logical expressions. If either expression returns TRUE, the result is TRUE; only when both expressions are FALSE is the result FALSE.	((([Region] = "France")    ([BikeBuyer] = "yes"))
IN	Creates a logical OR condition between each row being compared to a table. Note: the table constructor syntax uses curly braces.	'Product'[Color] IN { "Red", "Blue", "Black" }

\*Head to [DAX operators - DAX | Microsoft Learn](#) for information about DAX syntax, operators, troubleshooting, etc.

# Common Function Categories

MATH & STATS Functions	LOGICAL Functions	TEXT Functions	FILTER Functions	DATE & TIME Functions
<i>Basic <b>aggregation</b> functions as well as “<b>iterators</b>” evaluated at the row-level</i>	<i>Functions for returning information about values in a given <b>conditional expression</b></i>	<i>Functions to manipulate <b>text strings</b> or <b>control formats</b> for dates, times or numbers</i>	<i><b>Lookup</b> functions based on related tables and <b>filtering</b> functions for dynamic calculations</i>	<i>Basic <b>date and time</b> functions as well as advanced <b>time intelligence</b> operations</i>
<b>Common Examples:</b> <ul style="list-style-type: none"><li>SUM</li><li>AVERAGE</li><li>MAX/MIN</li><li>DIVIDE</li><li>COUNT/COUNTA</li><li>COUNTROWS</li><li>DISTINCTCOUNT</li></ul> <b>Iterator Functions:</b> <ul style="list-style-type: none"><li>SUMX</li><li>AVERAGEX</li><li>MAXX/MINX</li><li>RANKX</li><li>COUNTX</li></ul>	<b>Common Examples:</b> <ul style="list-style-type: none"><li>IF</li><li>IFERROR</li><li>AND</li><li>OR</li><li>NOT</li><li>SWITCH</li><li>TRUE</li><li>FALSE</li></ul>	<b>Common Examples:</b> <ul style="list-style-type: none"><li>CONCATENATE</li><li>FORMAT</li><li>LEFT/MID/RIGHT</li><li>UPPER/LOWER</li><li>PROPER</li><li>LEN</li><li>SEARCH/FIND</li><li>REPLACE</li><li>REPT</li><li>SUBSTITUTE</li><li>TRIM</li><li>UNICHAR</li></ul>	<b>Common Examples:</b> <ul style="list-style-type: none"><li>CALCULATE</li><li>FILTER</li><li>ALL</li><li>ALLEXCEPT</li><li>RELATED</li><li>RELATEDTABLE</li><li>DISTINCT</li><li>VALUES</li><li>EARLIER/EARLIEST</li><li>HASONEVALUE</li><li>HASONEFILTER</li><li>ISFILTERED</li><li>USERRELATIONSHIP</li></ul>	<b>Common Examples:</b> <ul style="list-style-type: none"><li>DATEDIFF</li><li>YEARFRAC</li><li>YEAR/MONTH/DAY</li><li>HOUR/MINUTE/SECOND</li><li>TODAY/NOW</li><li>WEEKDAY/WEEKNUM</li></ul> <b>Time Intelligence Functions:</b> <ul style="list-style-type: none"><li>DATESYTD</li><li>DATESQTD</li><li>DATESMTD</li><li>DATEADD</li><li>DATESINPERIOD</li></ul>

# Basic Date & Time Function

<b>DAY/MONTH/ YEAR()</b>	Returns the day of the month (1-31), month of the year (1-12), or year of a given date	= <b>DAY/MONTH/YEAR</b> (Date)
<b>HOUR/MINUTE/ SECOND()</b>	Returns the hour (0-23), minute (0-59), or second (0-59) of a given datetime value	= <b>HOUR/MINUTE/SECOND</b> (Datetime)
<b>TODAY/NOW()</b>	Returns the current date or exact time	= <b>TODAY/NOW</b> ()
<b>WEEKDAY/ WEEKNUM()</b>	Returns a weekday number from 1 (Sunday) to 7 (Saturday), or the week # of the year	= <b>WEEKDAY/WEEKNUM</b> (Date, [ReturnType])
<b>EOMONTH()</b>	Returns the date of the last day of the month, +/- a specified number of months	= <b>EOMONTH</b> (StartDate, Months)
<b>DATEDIFF()</b>	Returns the difference between two dates, based on a selected interval	= <b>DATEDIFF</b> (Date1, Date2, Interval)



# Basic Logical Functions (IF/AND/OR)

<b>IF()</b>	<i>Checks if a given condition is met, and returns one value if the condition is TRUE, and another if the condition is FALSE</i>	<b>=IF</b> (LogicalTest, ResultIfTrue, [ResultIfFalse])	
<b>IFERROR()</b>	<i>Evaluates an expression and returns a specified value if the expression returns an error, otherwise returns the expression itself</i>	<b>=IFERROR</b> (Value, ValueIfError)	
<b>AND()</b>	<i>Checks whether both arguments are TRUE, and returns TRUE if both arguments are TRUE, otherwise returns FALSE</i>	<b>=AND</b> (Logical1, Logical2)	<b>Note:</b> Use the <b>&amp;&amp;</b> and <b>  </b> operators if you want to include more than two conditions!
<b>OR()</b>	<i>Checks whether one of the arguments is TRUE to return TRUE, and returns FALSE if both arguments are FALSE</i>	<b>=OR</b> (Logical1, Logical2)	

# Text Function

<b>LEN()</b>	Returns the number of characters in a string	<b>=LEN</b> (Text) <i>Note: Use the &amp; operator as a shortcut, or to combine more than two strings!</i>
<b>CONCATENATE()</b>	Joins two text strings into one	<b>=CONCATENATE</b> (Text1, Text2)
<b>LEFT/MID/ RIGHT()</b>	Returns a number of characters from the start/middle/end of a text string	<b>=LEFT/RIGHT</b> (Text, [NumChars]) <b>=MID</b> (Text, StartPosition, NumChars)
<b>UPPER/LOWER/ PROPER()</b>	Converts letters in a string to upper/lower/proper case	<b>=UPPER/LOWER/PROPER</b> (Text)
<b>SUBSTITUTE()</b>	Replaces an instance of existing text with new text in a string	<b>=SUBSTITUTE</b> (Text, OldText, NewText, [InstanceNumber])
<b>SEARCH()</b>	Returns the position where a specified string or character is found, reading left to right	<b>=SEARCH</b> (FindText, WithinText, [StartPosition], [NotFoundValue])

# Related

## RELATED()

Returns related values in each row of a table based on relationships with other tables

=RELATED(ColumnName)



The column that contains the values you want to retrieve

Examples:

- Product\_Lookup[ProductName]
- Territory\_Lookup[Country]



### HEY THIS IS IMPORTANT!

**RELATED** works almost *exactly* like a **VLOOKUP** function – it uses the relationship between tables (*defined by primary and foreign keys*) to pull values from one table into a new column of another

Since this function requires row context, it can only be used as a **calculated column** or as part of an **iterator function** that cycles through all rows in a table (*FILTER, SUMX, MAXX, etc*)

# Basic Math & Stats Functions

<b>SUM()</b>	<i>Evaluates the sum of a column</i>	= <b>SUM</b> (ColumnName)
<b>AVERAGE()</b>	<i>Returns the average (arithmetic mean) of all the numbers in a column</i>	= <b>AVERAGE</b> (ColumnName)
<b>MAX()</b>	<i>Returns the largest value in a column or between two scalar expressions</i>	= <b>MAX</b> (ColumnName) or = <b>MAX</b> (Scalar1, [Scalar2])
<b>MIN()</b>	<i>Returns the smallest value in a column or between two scalar expressions</i>	= <b>MIN</b> (ColumnName) or = <b>MIN</b> (Scalar1, [Scalar2])
<b>DIVIDE()</b>	<i>Performs division and returns the alternate result (or blank) if div/0</i>	= <b>DIVIDE</b> (Numerator, Denominator, [AlternateResult])

# Count, Counta, Distinctcount & Countrows

---

<b>COUNT()</b>	<i>Counts the number of cells in a column that contain numbers</i>	= <b>COUNT</b> (ColumnName)
<b>COUNTA()</b>	<i>Counts the number of non-empty cells in a column (numerical and non-numerical)</i>	= <b>COUNTA</b> (ColumnName)
<b>DISTINCTCOUNT()</b>	<i>Counts the number of distinct or unique values in a column</i>	= <b>DISTINCTCOUNT</b> (ColumnName)
<b>COUNTROWS()</b>	<i>Counts the number of rows in the specified table, or a table defined by an expression</i>	= <b>COUNTROWS</b> (Table)

# Calculate

## CALCULATE()

*Evaluates a given expression or formula under a set of defined filters*

**=CALCULATE**(Expression, [Filter1], [Filter2],...)

*Name of an existing measure, or a DAX formula for a valid measure*

*Examples:*

- [Total Orders]
- SUM>Returns\_Data[ReturnQuantity])

*List of simple Boolean (True/False) filter expressions  
(**note:** these require simple, fixed values; you cannot create filters based on measures)*

*Examples:*

- Territory\_Lookup[Country] = "USA"
- Calendar[Year] > 1998



### PRO TIP:

*CALCULATE works just like **SUMIF** or **COUNTIF** in Excel, except it can evaluate measures based on ANY sort of calculation (not just a sum, count, etc); it may help to think of it like "**CALCULATEIF**"*

# Calculate (Example)

```
1 BikesReturn = CALCULATE([TotalReturn],AW_Product_Category_Lookup[CategoryName]="Bikes")
```

Category Name	Total Return
Accessories	1,130
Bikes	429
Clothing	269

429
BikesReturn

Here we've defined a new measure named "**BikeReturns**", which evaluates the "**Total Returns**" measure when the *CategoryName* in the Products table equals "**Bikes**"

# Calculate Change the filter context

## CALCULATE

*Filters modified by CALCULATE*

[CategoryName] = "Bikes"

If the measure being evaluated contains a **CALCULATE** function, filter context is *overwritten* between **Step 1** & **Step 2**

## STEP 1

*Filter context is detected & applied*

Category Name	Total Return	BikesReturn
Accessories	1,130	429
Bikes	429	429
Clothing	269	429
Components		429

Products[CategoryName] = "Accessories"

Product Table
Accessories

Product Table
Bikes

## STEP 2

*Filters flow "downstream" to all related tables*

Product Table
Bikes

1 1

AW_Sales_Data
Bikes

## STEP 3

*Measure formula evaluates against the filtered table*

1 BikesReturn = CALCULATE([TotalReturn],AW\_Product\_Category\_Lookup[CategoryName]="Bikes")

Count of rows in the **AW\_Returns\_Data** table, filtered down to only rows where the product category is "Bikes" = 429

AW_Returns_Data
Bikes



# All

## ALL()

Returns all rows in a table, or all values in a column, ignoring any filters that have been applied

=ALL(**Table** or **ColumnName**, [ColumnName1], [ColumnName2],...)

The table or column that you want to clear filters on

**Examples:**

- Transactions
- Products[ProductCategory]

List of columns that you want to clear filters on (optional)

**Notes:**

- If your first parameter is a table, you can't specify additional columns
- All columns must include the table name, and come from the same table

**Examples:**

- Customer\_Lookup[CustomerCity], Customer\_Lookup[CustomerCountry]
- Products[ProductName]

# Filter

## **FILTER()**

*Returns a table that represents a subset of another table or expression*

**=FILTER**(Table, FilterExpression)

*Table to be filtered*

**Examples:**

- Territory\_Lookup
- Customer\_Lookup

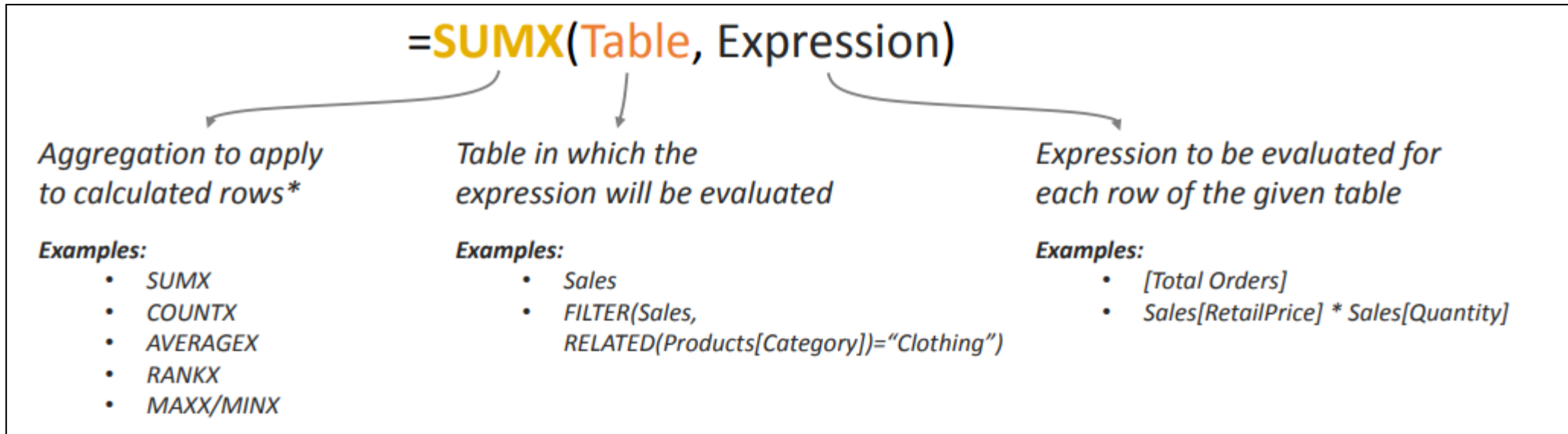
*A Boolean (True/False) filter expression to be evaluated for each row of the table*

**Examples:**

- Territory\_Lookup[Country] = "USA"
- Calendar[Year] = 1998
- Products[Price] > [Overall Avg Price]

# Iterator ("X") Functions

Iterator (or “X”) functions allow you to loop through the same calculation or expression on each row of a table, and then apply some sort of aggregation to the results (SUM, MAX, etc)



# Time Intelligence Formulas

Time Intelligence functions allow you to easily calculate common time comparisons:

Performance  
To-Date

=**CALCULATE**(Measure, **DATESYTD**(Calendar[Date]))

Use **DATESQTD** for Quarters or **DATESMTD** for Months

Previous  
Period

=**CALCULATE**(Measure, **DATEADD**(Calendar[Date], -1, **MONTH**))

Select an interval (**DAY**, **MONTH**, **QUARTER**, or **YEAR**) and the # of intervals to compare (i.e. previous month, rolling 10-day)

Running  
Total

=**CALCULATE**(Measure, **DATESINPERIOD**(Calendar[Date], **MAX**(Calendar[Date]), -10, **DAY**))