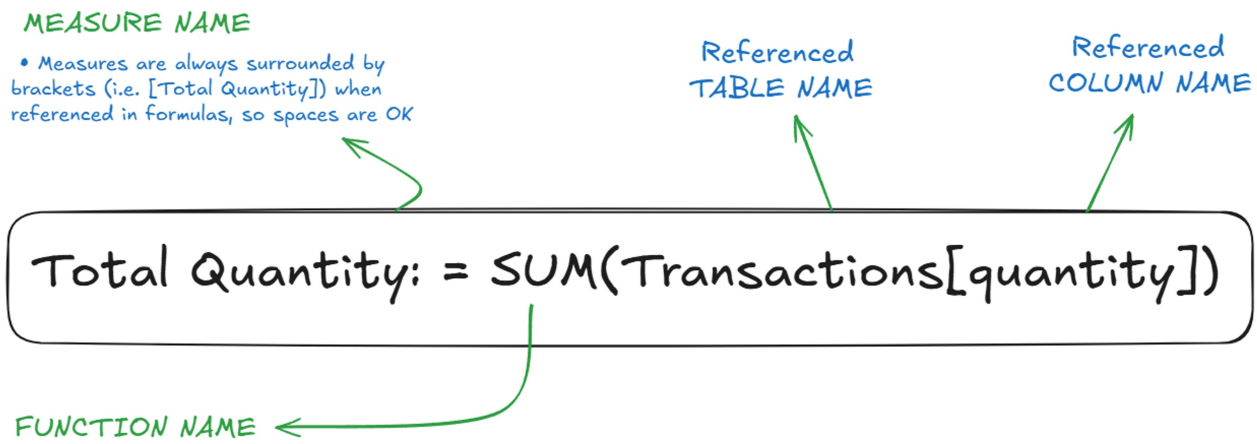
Power BI - DAX - p2

DAX SYNTAX



• 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

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" }

COMMON FUNCTION CATEGORIES

MATH & STATS Functions

Functions used for aggregation or iterative, row-level calculations

Common Examples:
- SUM
- AVERAGE
- MAX/MIN
- DIVIDE
- COUNT/COUNTA
- COUNTROWS
- DISTINCTCOUNT

Iterator Functions:
- SUMX
- AVERAGEX
- MAXX/MINX
- RANKX
- COUNTX

LOGICAL Functions

Functions that use conditional expressions (IF/THEN statements)

Common Examples:
- IF
- IFERROR
- AND
- OR
- NOT
- SWITCH
- TRUE
- FALSE

TEXT Functions

Functions used to manipulate text strings or value formats

Common Examples:
- CONCATENATE
- COMBINEVALUES
- FORMAT
- LEFT/MID/RIGHT
- UPPER/LOWER
- LEN
- SEARCH/FIND
- REPLACE
- SUBSTITUTE
- TRIM

FILTER Functions

Functions used to manipulate table and filter contexts

Common Examples:
- CALCULATE
- FILTER
- ALL
- ALLEXCEPT
- ALLSELECTED
- KEEPFILTERS
- REMOVEFILTERS
- SELECTEDVALUE

TABLE Functions

Functions that create or manipulate tables and output tables vs. scalar values

Common Examples:
- SUMMARIZE
- ADDCOLUMNS
- GENERATESERIES
- DISTINCT
- VALUES
- UNION
- INTERSECT
- TOPN

DATE & TIME Functions

Functions used to manipulate date & time values or handle time intelligence calculations

Common Examples:
- DATE
- DATEDIFF
- YEARFRAC
- YEAR/MONTH
- DAY/HOUR
- TODAY/NOW
- WEEKDAY
- WEEKNUM
- NETWORKDAYS

Time Intelligence:
- DATESYTD
- DATESMTD
- DATEADD
- DATESBETWEEN

RELATIONSHIP Functions

Functions used to manage &modify table relationships

Common Examples:
- RELATED
- RELATEDTABLE
- CROSSFILTER
- USERELATIONSHIP

BASIC MATH & STATS FUNCTIONS

SUM	Evaluates the sum of a column	=SUM(ColumnName)
AVERAGE	Returns the average (arithmetic mean) of all the numbers in a column	=AVERAGE(ColumnName)
MAX	Returns the largest value in a column or between two scalar expressions	=MAX(ColumnNameOrScalar1 , [Scalar2])
MIN	Returns the smallest value in a column or between two scalar expressions	=MIN(ColumnNameOrScalar1 , [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(ColumnName)
COUNTA	Counts the number of non-empty cells in a column (including Boolean values)	=COUNTA(ColumnName)
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])

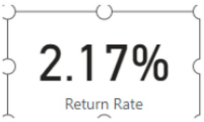
ASSIGNMENT: MATH & STATS

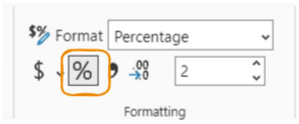
1.Create a measure named Total Customers, to calculate the number of distinct Adventure Works customers who made a transaction.

```
Total Customers =
    DISTINCTCOUNT(
        'Sales Data'[CustomerKey]
    )
```

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

```
Return Rate =
    DIVIDE(
    [Return Quanitity],
    [Quantity Sold],
    "No Sales")
```





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 multiples 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.

If Error

If(logical Expression , "True" , "False");

IFError(Value , "Result") :
IFError(Divide(num , den) , "Divide by 0 error")

=IFERROR(A11/B11,"Divide by Zero")

A	B	C	D
10	1	10	
20	2	10	
30	3	10	
40	4	10	
50	5	10	
60	6	10	
70	7	10	
80	8	10	
90	9	10	
100		Divide by Zero	
110	11	10	
120	12	10	

ASSIGNMENT: LOGICAL FUNCTIONS

1.Create a calculated column in the Customer Lookup table named Customer Priority:

- If the customer is a parent and has an annual income > $100,000, Customer Priority = Priority
- Otherwise, Customer Priority = Standard

```
Customer Priority =
    IF(
        'Customer Lookup'[Is Parent?] = "Yes" && 'Customer Lookup'[AnnualIncome] > 100000,
        "Priority",
        "Standard")
```

2.Create a calculated column in the Customer Lookup table named Income Level:

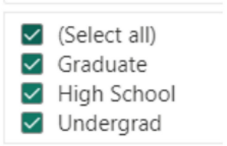
- If annual income is >= $150,000, Very High
- If annual income is >= $100,000, High
- If annual income is >= $50,000, Average
- Otherwise, Income Level = Low

```
Income Level =
    IF(
        'Customer Lookup'[AnnualIncome] > 150000 , "Very High",
        IF(
            'Customer Lookup'[AnnualIncome] > 100000 , "High",
            IF('Customer Lookup'[AnnualIncome]>50000 , "Average",
            "Low"
            )))
```

```
Income Level =
    SWITCH(
        TRUE(),
        'Customer Lookup'[AnnualIncome] > 150000 , "Very High",
        'Customer Lookup'[AnnualIncome] > 100000 , "High",
        'Customer Lookup'[AnnualIncome] > 50000 , "Average",
        "Low")
```

BONUS: Use a SWITCH function* to create another column named Education Category:

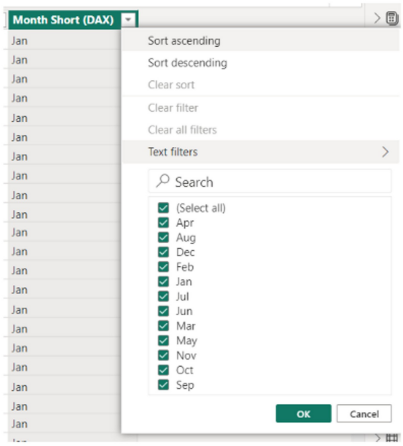
- If EducationLevel is High School or Partial High School, Education Category = High School
- If EducationLevel is Bachelors or Partial College, Education Category = Undergrad
- If EducationLevel is Graduate Degree, Education Category = Graduate




```
Education Category =
    SWITCH(
        'Customer Lookup'[EducationLevel],
        "High School" , "High School",
        "Partial High School" , "High School",
        "Bachelors" , "Undergrad",
        "Partial College" , "Undergrad",
        "Graduate Degree" , "Graduate")
```

TEXT FUNCTIONS

LEN	Returns the number of characters in a string	=LEN(Text)
CONCATENATE	Joins two text strings into one	=CONCATENATE(Text1, Text2)
UPPER /LOWER	Converts a string to upper or lower case	=UPPER/LOWER (Text)
LEFT/ RIGHT/MID	Returns a number of characters from the start/middle/end of a text string	=LEFT/RIGHT(Text, [NumChars]) =MID(Text, StartPosition, NumChars)
SUBSTITUTE	Replaces an instance of existing text with new text in a string	=SUBSTITUTE(Text, OldText, NewText, [InstanceNumber])
SEARCH	Returns the position where a specified string or character is found, reading left to right	=SEARCH(FindText, WithinText, [StartPosition], [NotFoundValue]) 2 "-1"




```
Month Short (DAX) =
    LEFT(
        'Calendar Lookup'[Month Name] ,
        3)
```

```
Customer Full Name =
    'Customer Lookup'[Prefix] & " " &'Customer Lookup'[FirstName] & " " & 'Customer Lookup'[LastName]
```

ASSIGNMENT: TEXT FUNCTIONS

1. Update the Month Short column in the Calendar Lookup table to extract and capitalize the first 3 characters of the month name.

```
Month Short (DAX) =
    UPPER(
        LEFT(
            'Calendar Lookup'[Month Name] ,
            3))
```

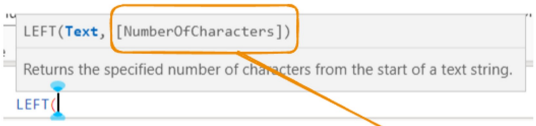
2. Create a new column in the Product Lookup table named SKU Category, to return any number of characters before the first hyphen in the ProductSKU column.

HL [SKU Category]

```
SKU Category =
    LEFT(
        'Product Lookup'[ProductSKU],
        SEARCH(
            "-",
            'Product Lookup'[ProductSKU]
            )
            -1
        )
```

3

No. of character



3 index for the first "-"

HLK - 2134 - A
HL - 2345 - B
KR - 1321 - C
KETY - 2232- D
K - 3212 - F

BASIC DATE & TIME FUNCTIONS

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

2.

```
Weekend =
    IF(
        'Calendar Lookup'[Week Of Day] IN {6,7},
        "Weekend",
        "Weekday")
```

1.

```
Week Of Day =
    WEEKDAY(
        'Calendar Lookup'[Date],
        2)
```

```
Create table instagram(
    username varchar(100),
    email varchar(200) unique,
    Post_Created TimeStamp DEFAULT Today() / NOW()
);
```

Default to NOW() or Today() - to get a new entry coming up when new baby born.

EOMonth(Now() - Dec 2024 , +11) - Last day of Nov 2025

```
Is Weekend? = IF('Calendar Lookup'[Week of Day] IN {6,7} , "Weekend" , "Weekday")
```

```
Week of Day = WEEKDAY('Calendar Lookup'[Date] , 2)
```

Is Weekend?	Week of Day
Weekday	3
Weekday	4
Weekday	5
Weekend	6
Weekend	7
Weekday	1
Weekday	2
Weekday	3
Weekday	4
Weekday	5
Weekend	6
Weekend	7
Weekday	1
Weekday	2
Weekday	3
Weekday	4
Weekday	5
Weekend	6