

Power BI - Advanced DAX - p1 - Lecture 8

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**NameOrScalar1**, [Scalar2])

MIN

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

=MIN(Column**NameOrScalar1**, [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.

```
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 Quantity], [Quantity Sold], "No Sales")
```

Return Quantity = SUM('Returns Data'[ReturnQuantity])

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.

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 $\geq \$150,000$, Very High
- If annual income is $\geq \$100,000$, High
- If annual income is $\geq \$50,000$, 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

(Select all)
 Bachelors
 Graduate Degree
 High School
 Partial College
 Partial High School

(Select all)
 Graduate
 High School
 Undergrad

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)

Krishna@gmail.com

**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

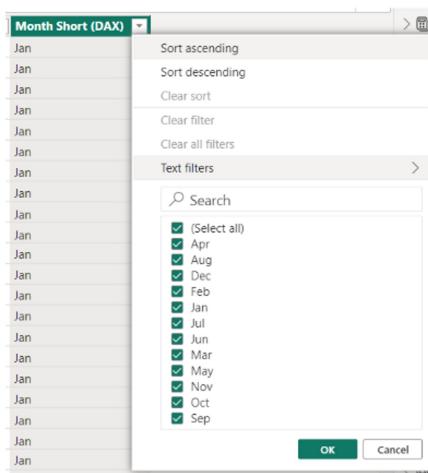
=SUBSTITUTE(Text, OldText, NewText, [InstanceNumber])

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.

SKU Category =

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

3

→ HL [SKU Category]

No. of character

KRI-13776-R

LEFT(Text, [NumberOfCharacters])
Returns the specified number of characters from the start of a text string.
LEFT

PWJK-37278-B

3 index for the first "-"

```
SEARCH(FindText, WithinText,
[StartPosition], [NotFoundValue])
```

Search("-",SKU) = 3 + 1

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)
```

Weekday[Interval] - 1 : Sunday [1] till Saturday [7]

Weekday[Interval] - 2 : Monday [1] till Sunday [7]

Weekday[Interval] - 3 : Monday [0] till Sunday [6]

ASSIGNMENT: DATE & TIME

1. Create a new column in the Customer Lookup table named Birth Year, to extract only the year from the BirthDate column

```
Birth Year =  
YEAR(  
    'Customer Lookup'[BirthDate]  
)
```

SKU Category 1 =

```
LEFT(  
    'Product Lookup'[ProductSKU],  
    SEARCH(  
        "-",  
        'Product Lookup'[ProductSKU],  
        SEARCH(  
            "-",  
            'Product Lookup'[ProductSKU]  
        ) + 1,  
        6  
)
```

SKU Category 1
BK-R93R-
BK-R68R-
BK-R68R-
BK-R68R-
BK-R68R-

Shifting the start position after 1st occurrence of "-"

```
Age = Year(Today()) - Birth Year  
= 2025 - 1990 = 35
```

```
Age = DATEDIFF(BirthDate , Today(), Year[Interval])
```

RELATED

RELATED() :-

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

=RELATED(Column Name)

The column from a related table containing the values you want to retrieve

Examples:

- 'Product Lookup'[Product Name]
- 'Territory Lookup'[Country]

HEY THIS IS IMPORTANT!

- RELATED works like a VLOOKUP function in Excel – 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.)

PRO TIP:

Instead of using RELATED to create extra columns (which increases file size), nest it within measures like FILTER or SUMX

Retail Price =

```
RELATED()  
    'Product Lookup'[ProductPrice])
```

Revenue =

```
'Sales Data'[Retail Price] * 'Sales Data'[OrderQuantity])
```