

## Power BI - DAX - p2

Dimension Table		
CategoryName	Total Returns	Total Quantity
Accessories	1130	57809
Bikes	429	13929
Clothing	269	12436
<b>Total</b>	<b>1828</b>	<b>84174</b>

Fact Table

Accurate result.

Why Explicit Measure >> Implicit Measure.

## DAX OPERATORS

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

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

"Important"

Text/Logical Operator	Meaning	Example
&	Concatenates two values to produce one text string	[City] & " " & [State]
<b>&amp;&amp;</b>	Create an AND condition between two logical expressions	([State] = "MA") && ([Quantity] > 10)
<b>   (double pipe)</b>	Create an OR condition between two logical expressions	([State] = "MA")    ([State] = "CT")
<b>IN</b>	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	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"> <li>• SUM</li> <li>• AVERAGE</li> <li>• MAX/MIN</li> <li>• DIVIDE</li> <li>• COUNT/COUNTA</li> <li>• COUNTROWS</li> <li>• DISTINCTCOUNT</li> </ul> <p>Iterator Functions:</p> <ul style="list-style-type: none"> <li>• SUMX</li> <li>• AVERAGEX</li> <li>• MAXX/MINX</li> <li>• RANKX</li> <li>• COUNTX</li> </ul>	<p>Functions that use conditional expressions (IF/THEN statements)</p> <p>Common Examples:</p> <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>	<p>Functions used to manipulate text strings or value formats</p> <p>Common Examples:</p> <ul style="list-style-type: none"> <li>• CONCATENATE</li> <li>• COMBINEVALUES</li> <li>• FORMAT</li> <li>• LEFT/MID/RIGHT</li> <li>• UPPER/LOWER</li> <li>• LEN</li> <li>• SEARCH/FIND</li> <li>• REPLACE</li> <li>• SUBSTITUTE</li> <li>• TRIM</li> </ul>	<p>Functions used to manipulate table and filter contexts</p> <p>Common Examples:</p> <ul style="list-style-type: none"> <li>• CALCULATE</li> <li>• FILTER</li> <li>• ALL</li> <li>• ALLEXCEPT</li> <li>• ALLSELECTED</li> <li>• KEEPFILTERS</li> <li>• REMOVEFILTERS</li> <li>• SELECTEDVALUE</li> </ul>	<p>Functions that create or manipulate tables and output tables vs. scalar values</p> <p>Common Examples:</p> <ul style="list-style-type: none"> <li>• SUMMARIZE</li> <li>• ADDCOLUMNS</li> <li>• GENERATESERIES</li> <li>• DISTINCT</li> <li>• VALUES</li> <li>• UNION</li> <li>• INTERSECT</li> <li>• TOPN</li> </ul>	<p>Functions used to manipulate date &amp; time values or handle time intelligence calculations</p> <p>Common Examples:</p> <ul style="list-style-type: none"> <li>• DATE</li> <li>• DATEDIFF</li> <li>• YEARFRAC</li> <li>• YEAR/MONTH</li> <li>• DAY/HOUR</li> <li>• TODAY/NOW</li> <li>• WEEKDAY</li> <li>• WEEKNUM</li> <li>• NETWORKDAYS</li> </ul> <p>Time Intelligence:</p> <ul style="list-style-type: none"> <li>• DATESYTD</li> <li>• DATESMTD</li> <li>• DATEADD</li> <li>• DATESBETWEEN</li> </ul>	<p>Functions used to manage &amp; modify table relationships</p> <p>Common Examples:</p> <ul style="list-style-type: none"> <li>• RELATED</li> <li>• RELATEDTABLE</li> <li>• CROSSFILTER</li> <li>• USERELATIONSHIP</li> </ul>

## 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 Or Scalar 1, [Scalar 2])

**MIN**

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

=MIN(Column Name Or Scalar 1, [Scalar 2])

**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")
```

$$\text{Return Rate} = \text{Return Qty} / \text{Total Qty Sold}$$

Total Customer  
**17416**

Total Customer - Sales Record

Total Customer (C.T)  
**18018**

Total Customer - Customer Table

Customer available on Customer Table is higher than Sales record that means some customer have not done any transaction.

Return Rate = Return Qty / Total Qty Sold

DIVIDE(Numerator, Denominator, [AlternateResult])  
Safe Divide function with ability to handle divide by zero case.

1 Return Rate = DIVIDE([Return Qty], [Total Quantity])

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

Total Quantity = SUM('Sales Record 2020-2022'[OrderQuantity])

**Formatting:**  
\$% Format Percentage  
\$ % .00 2

CategoryName	Total Returns	Total Quantity	Return Rate
Accessories	1130	57809	1.95%
Bikes	429	13929	3.08%
Clothing	269	12436	2.16%
<b>Total</b>	<b>1828</b>	<b>84174</b>	<b>2.17%</b>

V Format your report page >

Build visual

```
if :
else if :
else if :
else if :
else :
```

Short Circuiting

AND LOGIC - F && T && F && T && F && T → False

OR LOGIC - T || T || F || T || F || F || F || T → True

False

True

## 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

IFERROR(expression , "Message")

IFERROR(DIVIDE(num , den) , "Division By Zero")

10/0 -- Infinite

## 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(AND('Customer Lookup'[Parent] = "Yes" , 'Customer Lookup'[AnnualIncome] >= 100000) , "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)

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])

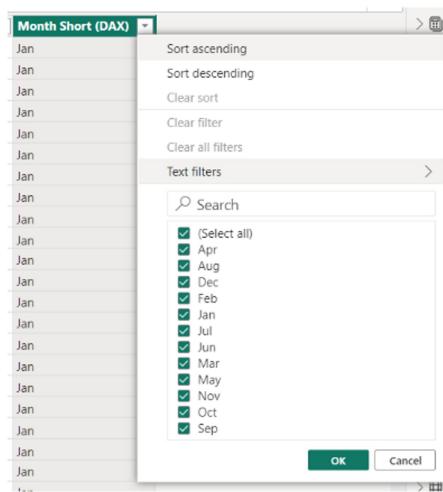
SEARCH

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]