DAX Logical Functions

Microsoft Power BI DAX provides various **Logical Functions** as shown below

- IF
- TRUE
- FALSE
- NOT
- OR
- IN
- AND
- IFERROR
- SWITCH

Now we will see how to use Power BI DAX Logical Functions with examples.

IF Function

The DAX **IF** function is used to check the given expression is True or False. The basic syntax of the Power BI DAX If Function is as shown below

Syntax

IF (LogicalTest, ResultIfTrue, [ResultIfFalse])

IF (Expression, True_Info, False_Info)

As you can see from the above syntax, this function accepts three arguments. First argument is the Boolean expression (which return true or false). If the expression results TRUE then second argument will return otherwise, third argument will return.

Syntax in Square Brackets is optional. If you are not providing third argument, then if the condition is not satisfied it will return NULL values.

Examples

If Example = IF(Orders[Quantity]>1,"Multiple Products", "Single Product")

Nested IF Function

In Power BI DAX, you can use the Nested If concept, One If statement inside another. Below statement will check whether the Sales amount of each column is less than 100 or not. If true then column will return “Low Sales” otherwise, it will enter into Nested If.

Examples

Nested If Example = IF(Orders[Sales]<100,"Low Sales", IF(Orders[Sales]>300, "High Sales", "Avg Sales"))

DAX TRUE & False Function

The DAX **TRUE** function will return a logical value **True**. The DAX **FALSE** function will return logical **False**.

True and False Example = IF(Orders[Sales]>100, TRUE (), FALSE ())

DAX NOT Function

The DAX **NOT** function will convert True to False, and False to True. I mean, it returns opposite result. The syntax of the Power BI DAX NOT Function is as shown below

```
NOT(Condition)
NOT(<logical>)
```

Example

NOT example = NOT(IF(Orders[Sales]>100, TRUE (), FALSE ()))

DAX OR Function

The DAX OR function is like either or statement in English, which is used to check multiple expressions. The syntax of the Power BI DAX OR Function is as shown below.

```
OR (Condition 1, Condition 2)
OR (Logical1, Logical2)
```

Example

```
Weekday = WEEKDAY (Orders [Order Date])
Weekend = IF(OR(Orders[Weekday]=1, Orders[Weekday]=7),"Weekend", "Weekday")
Weekend = IF(Orders[Weekday]=1 || Orders[Weekday]=7,"Weekend", "Weekday")
```

DAX IN Function

IN Example = IF(Orders[Weekday] **IN** {2,3,4,5,6}," Weekday ", Weekend"")

DAX AND Function

The DAX AND function is used to check multiple expressions. The syntax of the Power BI DAX AND Function is as shown below

```
AND (Condition 1, Condition 2)
AND (Logical1, Logical2)
```

As you can see from the above syntax, DAX **AND** function accepts two arguments. If both the conditions are True then it will return True otherwise, it return False.

Examples

And Example = IF(AND(Orders[Category]="Office Supplies", Orders[Sub-Category] = "Storage"), TRUE (), FALSE ())

And example = IF(Orders[Category]="Office Supplies" && Orders[Sub-Category] = "Storage", TRUE(), FALSE())

DAX IFERROR Function

The DAX IFERROR function is very useful to handle the arithmetic overflow, or any other errors. It simply performs calculation and return the result, if there is an error then it will return the value inside the second argument. The syntax of the DAX IFERROR Function is as shown below.

IFERROR (Calculation, Value_If_Error_Occurs)

Below statement will return 100 if error occurs. Indeed, all the records will throw error because we are dividing them 0. If there is no error, it will give calculation or expression value.

Error Example = IFERROR(Orders[Sales]/0, 100)

DAX SWITCH Function

The DAX SWITCH function helps you to return multiple options. For example, **IF** statement will return either True or false. However, you can use this switch case to multiple results. The syntax of the DAX SWITCH Function is as shown below.

SWITCH (Expression, Option 1, Result 1, Option 2, Result 2, ..., Else Result)

If Month of Order Date is 1 then below statement will return January, 2 means February, 3 means March, 4 means April, 5 means May, 12 means December otherwise, Unknown.

Switch Example = SWITCH (MONTH (Orders [Order Date]), 1, "January", 2, "February", 3, "March", 4, "April", 5, "May", 12, "December", "Unknown")

Switch Example = SWITCH (TRUE (), Orders [Sales]<100, "Low Sales", Orders[Sales]>300,"High Sales", "Medium Sales")

SWITCH () always test for equivalence. SWITCH () is still testing for equivalence! By providing the first argument as TRUE (), now each subsequent "test" is going to be checking for TRUE (). And since each of our inequality tests results in either TRUE () or FALSE () as a value, the test case that evaluate to TRUE () is the one that gets matched, and therefore the one that gets used. For instance, if [Measure] <1 evaluates to TRUE (), then expr1 gets returned.