Common errors in CALCULATE Filter arguments and solutions

Article based on the following post by Marco Russo & Alberto Ferrari
(https://www.sqlbi.com/articles/solving-errors-in-calculate-filter-arguments/)

If you are a Power BI developer and have faced errors when saving DAX expressions that contain CALCULATE functions, that sometimes are not clear why they occur, this post will help. I'll explain the reasoning behind some of these errors and how they could be fixed, achieving the same intended result.

The expression contains columns from multiple tables, but only columns from a single table can be used in a True/False expression that is used as a table filter expression.

When you try to include references to columns from two or more different tables in the arguments section of the function you will get this error.

For example, trying to get the count of stores in the manager location will trigger this error, because an ALL function is used for each filter argument in the CALCULATE function, and the ALL function cannot be used in 2 tables at the same time:

```
1 Manager Location Store Count =
2 CALCULATE (
3     [Store Count],
4     KEEPFILTERS ( Manager[Location] = Store[Location] )
5 )
```

So, to solve this issue we must create a table using a FILTER and a CROSSJOIN:

```
1 Manager Location Store Count =
2 CALCULATE (
3     [Store Count],
4     KEEPFILTERS (
5         FILTER (
6             CROSSJOIN ( ALL (Manager[Location]), ALL (Store[Location]) ),
7             Customer[CountryRegion] = Store[CountryRegion]
8         )
9     )
10 )
```

A function 'XXXXXXXX' has been used in a True/False expression that is used as a table filter expression. This is not allowed.

It's simple, functions that will do a context transition cannot be used as arguments in a CALCULATE function because these types of expressions cannot be translated into a FILTER. There are several examples in which this could happen:

CALCULATE:

```
1 Top Sales =
2 CALCULATE (
3    [Sales],
4    Sales[Price] > CALCULATE ( MAX ( Sales[Price] ) * 0.9, REMOVEFILTERS() )
5 )
```

So, to fix this issue we can write an explicit filter as a variable that can later be used in the argument:

```
1  Top Sales =
2  VAR PriceLimit =
3     CALCULATE (
4        MAX ( Sales[Price] ) * 0.9,
5        REMOVEFILTERS()
6     )
7  VAR Val =
8     CALCULATE (
9        [Sales],
10       Sales[Price] > PriceLimit
11    )
12 RETURN Val
```

LASTDATE:

```
1 Latest Sales =
2 CALCULATE (
3    [Sales],
4    'Calendar'[Date] <= LASTDATE ( 'Calendar'[Date] )
5 )
```

Here the issue is similar, LASTDATE is a function that can only be used in a table context and will be different for each element evaluated in the expression, and it could not be converted into a FILTER, so the correct function to use would be MAX, and assigning it to a variable that will be later used as the conditional element for the evaluation inside the CALCULATE argument.

```
1 Latest Sales =
2 VAR _LatestDate = MAX ( 'Calendar'[Date] )
3 VAR Val =
4    CALCULATE (
5       [Sales],
6       'Calendar'[Date] <= _LatestDate
7    )
8 RETURN Val
```

Measure conditional with no column reference:

TAnother example related to the same issue is then we can to get a calculated measure based on the performance of another measure. For example, get the Sales that have a Margin greater than 50%:

```
1 Sales (Margin >50%) =
2 CALCULATE (
3    [Sales],
4    [Margin %] > 0.5
5 )
```

This case is a little different, because additionally it's not clear in the expression what do we want to filter, is the products that have a margin above 50%, or the regions, or the clients? So, we would need to specify that filter first in a variable, and then use that filter as the argument inside the CALCULATE function:

```
1
2 Sales (Products - Margin >50%) =
3 VAR _Filter = FILTER (
4    Products,
5    [Margin %] > 0.5
6 )
7 Return CALCULATE (
8    [Sales],
9    _Filter
10 )
```

Conclusions

When writing the filter arguments in a CALCULATE function, make sure you:

- Inside each argument you evaluate columns from one single table
- Add functions that perform context transition and measure reference calculations to a variable before the CALCULATE evaluation, and use the variable in the argument.
- Make sure measures reference in a FILTER have a table/column target to be filtered.