Best Practices for Creating Calculations in Tableau

*Version: 2019.4 Applies to: Tableau Desktop, Tableau Online, Tableau Public, Tableau Server*

This article describes several tips and guidelines for creating efficient calculations in Tableau. These guidelines are designed to help you optimise your workbook performance. For more information about all the ways you can improve workbook performance, see the [Optimise Workbook Performance](https://help.tableau.com/current/pro/desktop/en-gb/performance_tips.htm) series.

General Rule: Avoid using a calculated field multiple times in another calculation

Referencing the same calculated field multiple times within another calculation will result in performance issues. If you use a calculated field within a calculation (also known as creating a Nested Calculation), try to reference it only once in the calculation.

Note that referencing a field (terminal field) multiple times in a calculation shouldn't degrade performance.

Example

Let's say you create a calculated field that uses a complicated multiple line calculation to find mentions, or Twitter handles, in tweets. The calculated field is titled, Twitter Handle. Each handle that is returned starts with the '@' sign (for example: @user).

For your analysis, you want to remove the '@' symbol.

To do so, you can use the following calculation to remove the first character from the string:

RIGHT([Twitter Handle], LEN([Twitter Handle]) -1)

This calculation is quite simple. However, since it references the Twitter Handle calculation twice, it performs that calculation twice for each record in your data source: once for the RIGHT function and again for the LEN function.

In order to avoid calculating the same calculation more than once, you can rewrite the calculation to one that uses the Twitter Handle calculation only once. In this example, you can use MID to accomplish the same goal:

MID([Twitter Handle], 2)

Tip 1: Convert multiple equality comparisons to a CASE expression or a group

Let's say you have the following calculation, which uses the calculated field, Person (calc), multiple times and employs a series of OR functions. This calculation, though a simple logical expression, will cause query performance issues because it performs the Person (calc) calculation at least ten times.

IF [Person (calc)] = 'Henry Wilson'  
OR [Person (calc)] = 'Jane Johnson'  
OR [Person (calc)] = 'Michelle Kim'  
OR [Person (calc)] = 'Fred Suzuki'  
OR [Person (calc)] = 'Alan Wang'  
THEN 'Lead'  
ELSEIF [Person (calc)] = 'Susan Nguyen'  
OR [Person (calc)] = 'Laura Rodriguez'  
OR [Person (calc)] = 'Ashley Garcia'  
OR [Person (calc)] = 'Andrew Smith'  
OR [Person (calc)] = 'Adam Davis'  
THEN 'IC'  
END

Instead of using an equality comparison, try the following solutions.

Solution 1

Use a CASE expression. For example:

CASE [Person (calc)]  
WHEN 'Henry Wilson' THEN 'Lead'  
WHEN 'Jane Johnson' THEN 'Lead'  
WHEN 'Michelle Kim' THEN 'Lead'  
WHEN 'Fred Suzuki' THEN 'Lead'  
WHEN 'Alan Wang' THEN 'Lead'  
  
WHEN 'Susan Nguyen' THEN 'IC'  
WHEN 'Laura Rodriguez' THEN 'IC'  
WHEN 'Ashley Garcia' THEN 'IC'  
WHEN 'Andrew Smith' THEN 'IC'  
WHEN 'Adam Davis' THEN 'IC'  
END

In this example, the calculated field, Person (calc), is only referenced once. Therefore, it is only performed once. CASE expressions are also further optimised in the query pipeline, so you gain an additional performance benefit.

Solution 2

Create a group instead of a calculated field. For more information, see [Group Your Data](https://help.tableau.com/current/pro/desktop/en-gb/sortgroup_groups_creating.htm).

Tip 2: Convert multiple string calculations into a single REGEXP expression

**Note**: REGEXP calculations are available only when using Tableau data extracts or when connected to Text File, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), and Oracle data sources. For more information, see [Additional Functions](https://help.tableau.com/current/pro/desktop/en-gb/functions_functions_additional.htm).

Example 1: CONTAINS

Let's say you have the following calculation, which uses the calculated field, Category (calc), multiple times. This calculation, though also a simple logical expression, will cause query performance issues because it performs the Category (calc) calculation multiple times.

IF CONTAINS([Segment (calc)],'UNKNOWN')  
OR CONTAINS([Segment (calc)],'LEADER')  
OR CONTAINS([Segment (calc)],'ADVERTISING')  
OR CONTAINS([Segment (calc)],'CLOSED')  
OR CONTAINS([Segment (calc)],'COMPETITOR')  
OR CONTAINS([Segment (calc)],'REPEAT')  
THEN 'UNKNOWN'  
ELSE [Segment (calc)] END

You can use a REGEXP expression to get the same results without as much repetition.

Solution

IF REGEXP\_MATCH([Segment (calc)], 'UNKNOWN|LEADER|ADVERTISING|CLOSED|COMPETITOR|REPEAT') THEN 'UNKNOWN'  
ELSE [Segment (calc)] END

With string calculations that use a similar pattern, you can use the same REGEXP expression.

Example 2: STARTSWITH

IF STARTSWITH([Segment (calc)],'UNKNOWN')  
OR STARTSWITH([Segment (calc)],'LEADER')  
OR STARTSWITH([Segment (calc)],'ADVERTISING')  
OR STARTSWITH([Segment (calc)],'CLOSED')  
OR STARTSWITH([Segment (calc)],'COMPETITOR')  
OR STARTSWITH([Segment (calc)],'REPEAT')  
THEN 'UNKNOWN'

Solution

IF REGEXP\_MATCH([Segment (calc)], '^(UNKNOWN|LEADER|ADVERTISING|CLOSED|COMPETITOR|REPEAT)') THEN 'UNKNOWN'  
ELSE [Segment (calc)] END

Note that the '^' symbol is used in this solution.

Example 3: ENDSWITH

IF ENDSWITH([Segment (calc)],'UNKNOWN')  
OR ENDSWITH([Segment (calc)],'LEADER')  
OR ENDSWITH([Segment (calc)],'ADVERTISING')  
OR ENDSWITH([Segment (calc)],'CLOSED')  
OR ENDSWITH([Segment (calc)],'COMPETITOR')  
OR ENDSWITH([Segment (calc)],'REPEAT')  
THEN 'UNKNOWN'  
ELSE [Segment (calc)] END

Solution

IF REGEXP\_MATCH([Segment (calc)], '(UNKNOWN|LEADER|ADVERTISING|CLOSED|COMPETITOR|REPEAT)$') THEN 'UNKNOWN'  
ELSE [Segment (calc)] END

Note that the '$' symbol is used in this solution.

Tip 3: Manipulate strings with REGEXP instead of LEFT, MID, RIGHT, FIND, LEN

Regular expressions can be a very powerful tool. When doing complex string manipulation, consider using regular expressions. In a lot of cases, using a regular expression will result in a shorter and more efficient calculation. For more information about regular expressions, see the [Become a regex regular and wrangle imperfect data](https://www.tableau.com/en-gb/about/blog/2015/6/become-regular-regular-expressions-39802) post on the Tableau blog.

Example 1

Let's say you have the following calculation, which removes protocols from URLs. For example: "https://www.tableau.com" becomes "www.tableau.com".

IF (STARTSWITH([Server], "http://")) THEN  
MID([Server], Len("http://") + 1)  
ELSEIF(STARTSWITH([Server], "https://")) THEN  
MID([Server], Len("https://") + 1)  
ELSEIF(STARTSWITH([Server], "tcp:")) THEN  
MID([Server], Len("tcp:") + 1)  
ELSEIF(STARTSWITH([Server], "\\")) THEN  
MID([Server], Len("\\") + 1)  
ELSE [Server]  
END

Solution

You can simplify the calculation and improve performance by using a REGEXP\_REPLACE function.

REGEXP\_REPLACE([Server], "^(http://|https://|tcp:|\\\\)", "")

Example 2

Let's say you have the following calculation, which returns the second part of an IPv4 address. For example: "172.16.0.1" becomes "16".

IF (FINDNTH([Server], ".", 2) > 0) THEN  
MID([Server],  
FIND([Server], ".") + 1,  
FINDNTH([Server], ".", 2) - FINDNTH([Server], ".", 1) - 1  
)  
END

Solution

You can simplify the calculation and improve performance by using a REGEXP\_EXTRACT function.

REGEXP\_EXTRACT([Server], "\.([^\.]\*)\.")

Tip 4: Do not use sets in calculations

If you are using sets in a calculation, consider replacing them with an alternative, but equivalent calculation.

Example

Let's say you have the following calculation, which uses the set, Top Customers (set).

IF ISNULL([Customer Name]) OR [Top customers (set)] THEN [Segment] ELSE [Customer Name] END

Solution 1

If the set is simple, you can create a calculated field that returns the same result as the set. For example:

CASE [Customer Name]  
WHEN 'Henry Wilson' THEN True  
WHEN 'Jane Johnson' THEN True  
WHEN 'Michelle Kim' THEN True  
WHEN 'Fred Suzuki' THEN True  
WHEN 'Alan Wang' THEN True  
ELSE False  
END

**Note**: Using the pattern WHEN TRUE … ELSE is recommended in this situation to avoid performance issues due to the use of sets. It is not a recommended pattern in most scenarios.

Solution 2

If the set is more complex, consider creating a group that maps all the elements in the set to a given value or attribute, such as 'IN', and then modify the calculation to check for that value/attribute. For example:

IF ISNULL([Customer Name]) OR [Top Customers(group)]='IN' THEN [Segment] ELSE [Customer Name] END

For more information, see [Group Your Data](https://help.tableau.com/current/pro/desktop/en-gb/sortgroup_groups_creating.htm) and [Create Sets](https://help.tableau.com/current/pro/desktop/en-gb/sortgroup_sets_create.htm).

Tip 5: Do not use sets to group your data

Sets are meant to make comparisons on subsets of data. Groups are meant to combine related members in a field. Converting sets to groups, such as with the following example, is not recommended:

IF [Americas Set] THEN "Americas"  
ELSEIF [Africa Set] THEN "Africa"  
ELSEIF [Asia Set] THEN "Asia"  
ELSEIF [Europe Set] THEN "Europe"  
ELSEIF [Oceania Set] THEN "Oceania"  
ELSE "Unknown"  
END

This is not recommended for the following reasons:

* **Sets are not always exclusive**. Some members can appear in multiple sets. For example, Russia could be placed both in the Europe set and the Asia set.
* **Sets cannot always be translated to groups**. If the sets are defined by exclusion, conditions, or limits, it might be difficult or even impossible to create an equivalent group.

Solution

Group your data using the Group feature. For more information, see [Group Your Data](https://help.tableau.com/current/pro/desktop/en-gb/sortgroup_groups_creating.htm).