KNOWLEDGE BASE

Aggregate Error Messages In a Calculation Editor

Published: 11 Jan 2016
Last Modified Date: 24 Aug 2022

Issue

When creating a calculation, one of the following errors might occur:

- *"Cannot mix aggregate and non-aggregate arguments with this function." (Option 1,2,3 or 4 can be used).*
- *"All fields must be aggregate or constant when using table calculation functions or fields from multiple data sources." (Option 1 or 3 can be used).*
- *"Argument to sum (an aggregate function) is already an aggregation, and cannot be further aggregated." (Option 2, 3 or 4 can be used).*

Environment

Tableau Desktop

Resolution

Each option can result in different values (please reference the attached workbook in the right-hand pane and additional information section for specific examples).

Option 1 (Aggregate All Fields)

Wrap all fields in an aggregation.

Sample (not using LOD):
[Profit] / SUM ([Sales]) -> SUM ([Profit]) / SUM ([Sales])

Sample (using LOD):
RANK({fixed: SUM([Sales])}) --> RANK(SUM({fixed: SUM([Sales])}))

Option 2 (De-aggregate All Fields)

Remove aggregations from all of the fields.

Sample (not using LOD):
[Profit] / SUM ([Sales]) ->[Profit] / [Sales]

Sample (using LOD):
[Profit] / SUM ([Sales]) ->[Profit] / { fixed: SUM ([Sales]) }

Option 3 (De-aggregate All Fields Then Aggregate the Calculation.)

Move the aggregation so the calculation is aggregated.

For example, the calculation:
IF [Row ID] = 1
THEN SUM([Sales])
END

could become:
SUM(
IF [Row ID] = 1
THEN [Sales]
END)

PRINT THIS PAGE

Related Links

Trending Articles

Attachments

Aggregate Sample Workbook.twbx

Search Support

Option 4 (Use Level of Detail to De-aggregate a Field)

Use a Level of Detail (LOD) Expression to make an aggregation non-aggregate. All LOD expression return non-aggregated values.

[Sales]/SUM([Sales]) -> [Sales]/{FIXED: SUM([Sales])}

Cause

"Cannot mix aggregate and non-aggregate arguments with this function."
– All fields must be the same aggregation (aggregated or de-aggregated). Aggregations are computed at the level of detail in the view, which will return one value for several records. Non-aggregate fields are computed for every record in the underlying data, which will return one value per record.

"All fields must be aggregate or constant when using Table Calculations or fields from multiple data sources"
– When blending or using table calculations, all fields must be aggregated.

"Argument to sum (an aggregate function) is already an aggregation, and cannot be further aggregated.".
– An aggregated field cannot be aggregated again.

Additional Information

How to determine if a field is aggregated.

Imported fields from the underlying data and Level of Detail (LOD) expressions are always non-aggregated until they are wrapped in an aggregation, such as SUM(), MIN(), ATTR(). To see a full list of all aggregated functions, please look at the "Aggregate Functions" article in the related links section.

Calculated fields can be either aggregate or non-aggregate depending on how aggregations are used. If no aggregations are used, or if the outer most expression is a LOD expression, then the calculation will return non-aggregated results.

One trick to determine if a field is aggregated is to add the field into the view. If the field displays AGG("Field Name"), the field is already aggregated

Examples:

Below is an explanation of how each option works using the sample data set shown as a reference.

Sample Data Set

ROW ID	PROFIT	SALES
1	100	30
2	50	60
3	7	10

Option 1
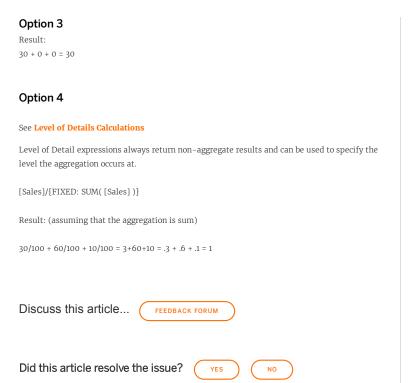
Formula:
SUM ([Profit]) / SUM ([Sales])

Result:
(100 + 50 + 7) / (30 + 60 + 10) = 157/100 = 1.57

Option 2
[Profit] / [Sales]

Result: (assuming that the aggregation in the view is SUM)

100/30 + 50/60 + 7/10 = 3.333 + .833 + .7 = 4.867

Option 3

Result:

30 + 0 + 0 = 30

Option 4

See **Level of Details Calculations**

Level of Detail expressions always return non-aggregate results and can be used to specify the level the aggregation occurs at.

[Sales]/{FIXED: SUM([Sales])}

Result: (assuming that the aggregation is sum)

30/100 + 60/100 + 10/100 = 3+60+10 = .3 + .6 + .1 = 1

Discuss this article...   **FEEDBACK FORUM**

Did this article resolve the issue?   **YES**   **NO**

All Support