

Know How LOD Expressions work in Tableau



Upasana Priyadarshiny

Follow

Feb 20, 2019 · 9 min read



The aim of any BI tool is to give **Data Analysis** a better flow. If one, as a professional, faces the difficulty of using the tool while solving a problem, the state of flow is broken. One common cause of this problem is the need to work with data that has been aggregated to different **Level Of Detail in Tableau (LOD)**.

Tableau placing on top of the pile for the 6th time in a Row, on Gartner's Magic Quadrant, definitely says something about its demand in the market. This is probably the best time to .

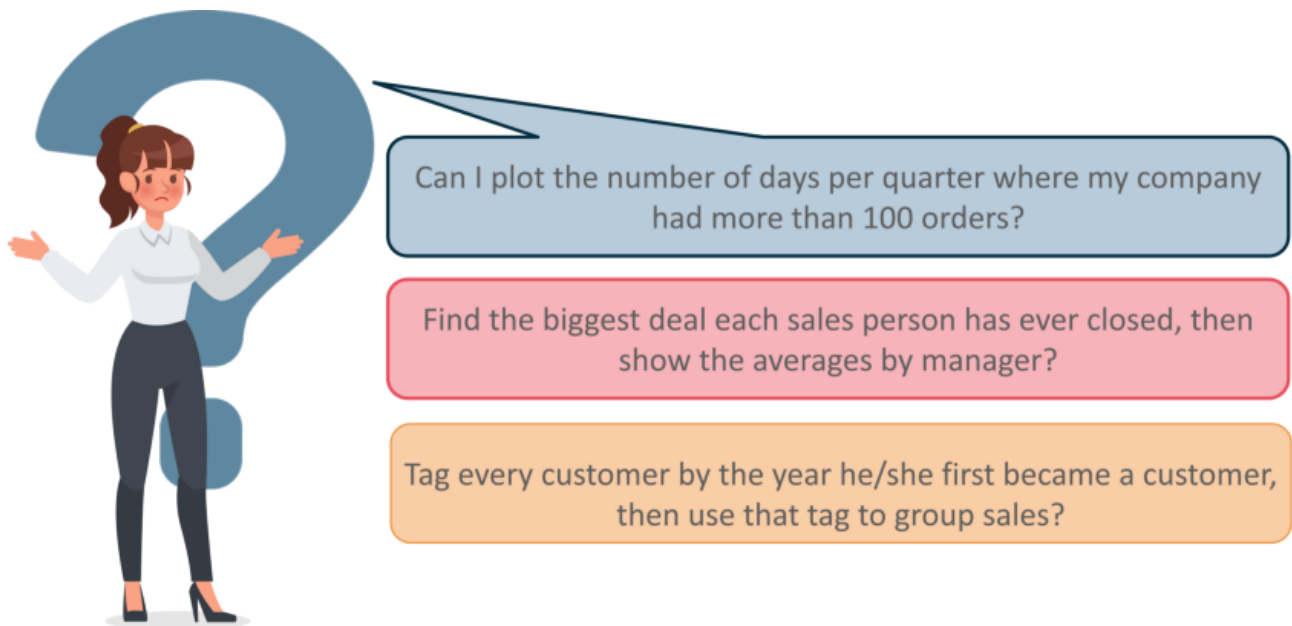
This blog will help you understand LOD expressions and discuss the following topics:

- Why do you need Level of Detail in Tableau?
- What is Level of Detail in Tableau?

- Row Level & View Level Expressions
- Types of LOD Expressions
- Aggregation and LOD Expressions
- Filters and LOD Expressions
- Creating LOD Expressions
- Data Sources supporting Level Of Detail in Tableau
- Table Calculations vs Level Of Detail in Tableau
- Limitations of Level Of Detail in Tableau

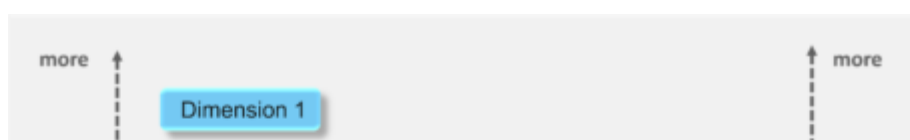
Why do you need LOD?

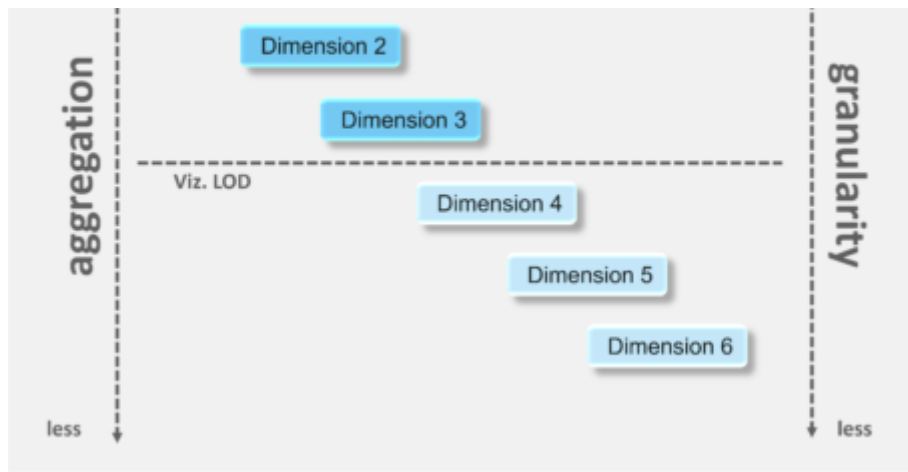
There often are questions that one is bound to come across when analysing data. These questions are often simple to ask, but hard to answer. They often sound something like:



To address these types of questions, a new syntax was introduced called **Level of Detail** in Tableau 9.0. This new syntax both simplified and extended Tableau's calculation language by making it possible to address these questions directly.

What is LOD?





LOD Expressions represent an elegant and powerful way to answer questions involving multiple levels of granularity in a single visualization.

Level of Detail in Tableau or LOD expressions allow you to compute values at the data source level and the visualization level. However, LOD expressions give you even more control on the level of granularity you want to compute. They can be performed at a *more granular* level (INCLUDE Calculation), a *less granular* level (EXCLUDE Calculation), or an *entirely independent level* (FIXED Calculation).

Row Level & View Level Expressions

Row Level

In Tableau, expressions referencing **unaggregated** data source columns are computed for each row in the underlying table. In this case, the dimensionality of the expression is *row level*. An example of a row-level expression is:

```
[Sales] / [Profit]
```

This calculation will be evaluated in each row of the database. The Sales value in each row will be divided by the Profit value in that row, producing a new column with the result of the multiplication (a profit ratio).

If you create a calculation with this definition, save it with the name [ProfitRatio], and then drag it from the Data pane to a shelf, Tableau shall typically aggregate the calculated field for the view:

```
SUM[ProfitRatio]
```

View Level

By contrast, expressions referencing **aggregated** data source columns are computed at the dimensionality defined by the dimensions in the view. In this case, the dimensionality of the expression is view level. An example of a view-level expression is:

```
SUM(Sales) / SUM(Profit)
```

If you drag this calculation to a shelf (or type it directly on a shelf as an ad-hoc calculation), Tableau encloses it in an **AGG function**:

```
AGG(SUM(Sales) / SUM(Profit))
```

This is what is known as an **Aggregate Calculation**.

Aggregation and LOD Expressions

LOD Expression is Coarser Than View Level of Detail

An expression has a coarser level of detail than the view when it references a **subset of the dimensions** in the view.

For example, for a view that contained the dimensions [Category] and [Segment], you could create a level of detail in Tableau that uses only one of these dimensions:

```
{FIXED [Segment] : SUM([Sales])}
```

In this case, the expression has a coarser level of detail than the view. It bases its values on one dimension ([Segment]), whereas the view is basing its view on two dimensions ([Segment] and [Category]).

The result is that using the level of detail expression in the view causes certain values to be replicated-that is, to **appear multiple times**.

LOD Expression is Finer Than View Level of Detail

An expression has a finer level of detail than the view when it references a **superset of the dimensions** in the view.

When you use such an expression in the view, Tableau will aggregate results up to the view level. For example, the following level of detail in Tableau references two dimensions:

```
{FIXED [Segment], [Category] : SUM([Sales])}
```

When this expression is used in a view that has only [Segment] as its level of detail, the values *must be aggregated*. Here's what you would see if you dragged that expression to a shelf:

```
AVG([FIXED [Segment]], [Category] : SUM([Sales]))
```

An **aggregation**(in this case, average) is automatically assigned by Tableau. You can change the aggregation as needed.

Adding an LOD Expression to the View

Whether a Level Of Detail in Tableau expression is aggregated or replicated in the view is determined by the **expression type** and **granularity**.

- INCLUDE expressions will have either the same level of detail as the view or a finer level of detail than the view. Therefore, values will never be replicated.
- FIXED expressions can have a finer level of detail than the view, a coarser level of detail, or the same level of detail. The need to aggregate the results of a FIXED level of detail depends on what dimensions are in the view.
- EXCLUDE expressions always cause replicated values to appear in the view. When calculations including EXCLUDE level of detail expressions are placed on a shelf, Tableau defaults to the **ATTR aggregation** as opposed to SUM or AVG, to indicate that the expression is not actually being aggregated and that changing the aggregation will have no effect on the view.

Level of detail expressions are always automatically wrapped in aggregate when they are added to a shelf in the view unless they're used as dimensions.

Filters and LOD Expressions



Table Calculation Filters

The image here depicts the order of execution of filters from top to bottom. The text on the right shows where the LOD expressions are evaluated in this sequence.

Extract Filters (in orange) are only relevant if you're creating a Tableau Extract from a data source. Table calculations filters (dark blue) are applied after calculations are executed and therefore hide marks without filtering out the underlying data used in the calculations.

FIXED calculations are applied before dimension filters, so unless you promote the fields on your Filter shelf to Improve View Performance with Context Filters, they will be ignored.

Types of LOD Expressions

INCLUDE Calculation

INCLUDE computes values using the specified dimensions in addition to whatever dimensions are in the view. This level of detail expression is most useful when including a dimension that isn't in the view.

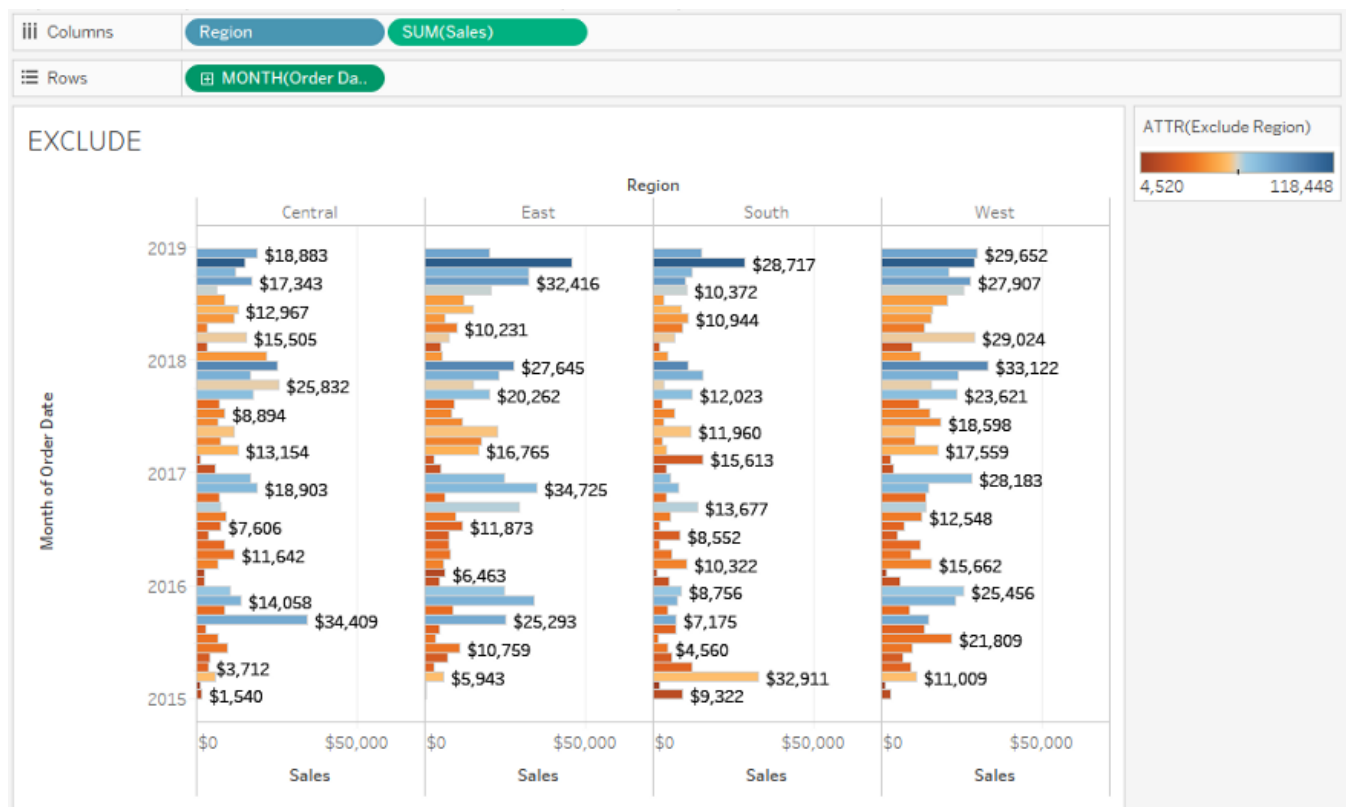
For Example: { INCLUDE [Customer Name] : SUM([Sales]) }



EXCLUDE Calculation

EXCLUDE explicitly remove dimensions from the expression-that is, they subtract dimensions from the view level of detail. This level of detail in Tableau, is most useful for eliminating a dimension in the view.

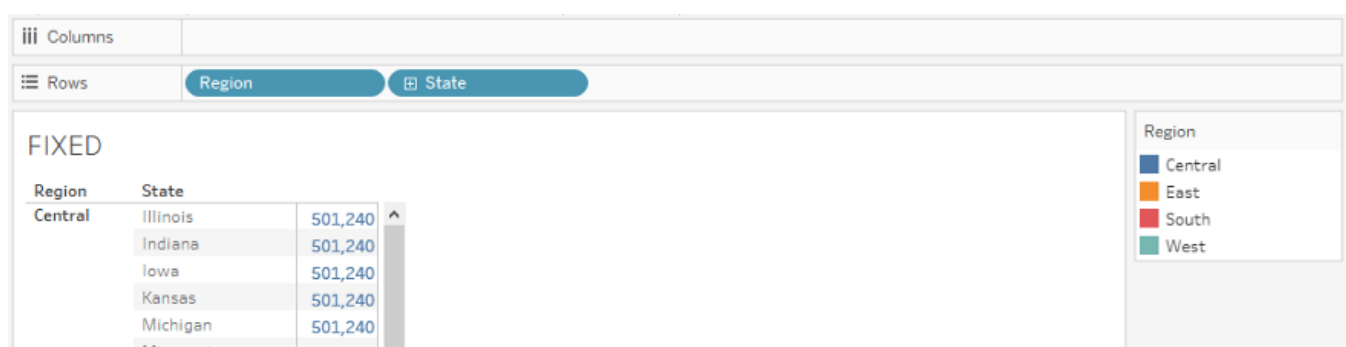
For Example: {EXCLUDE [Region]: SUM([Sales])}



FIXED Calculation

FIXED computes values using the specified dimensions without reference to the view level of detail-that is, without reference to any other dimensions in the view. This level of detail expression also ignores all the filters in the view other than context filters, data source filters, and extract filters.

For Example: { FIXED [Region] : SUM([Sales]) }



	Missouri	501,240
	Nebraska	501,240
	North Dakota	501,240
	Oklahoma	501,240
	South Dakota	501,240
	Texas	501,240
	Wisconsin	501,240
East	Connecticut	678,781
	Delaware	678,781
	District of Columbia	678,781
	Maine	678,781
	Maryland	678,781
	Massachusetts	678,781
	New Hampshire	678,781
	New Jersey	678,781
	New York	678,781

Creating LOD Expressions

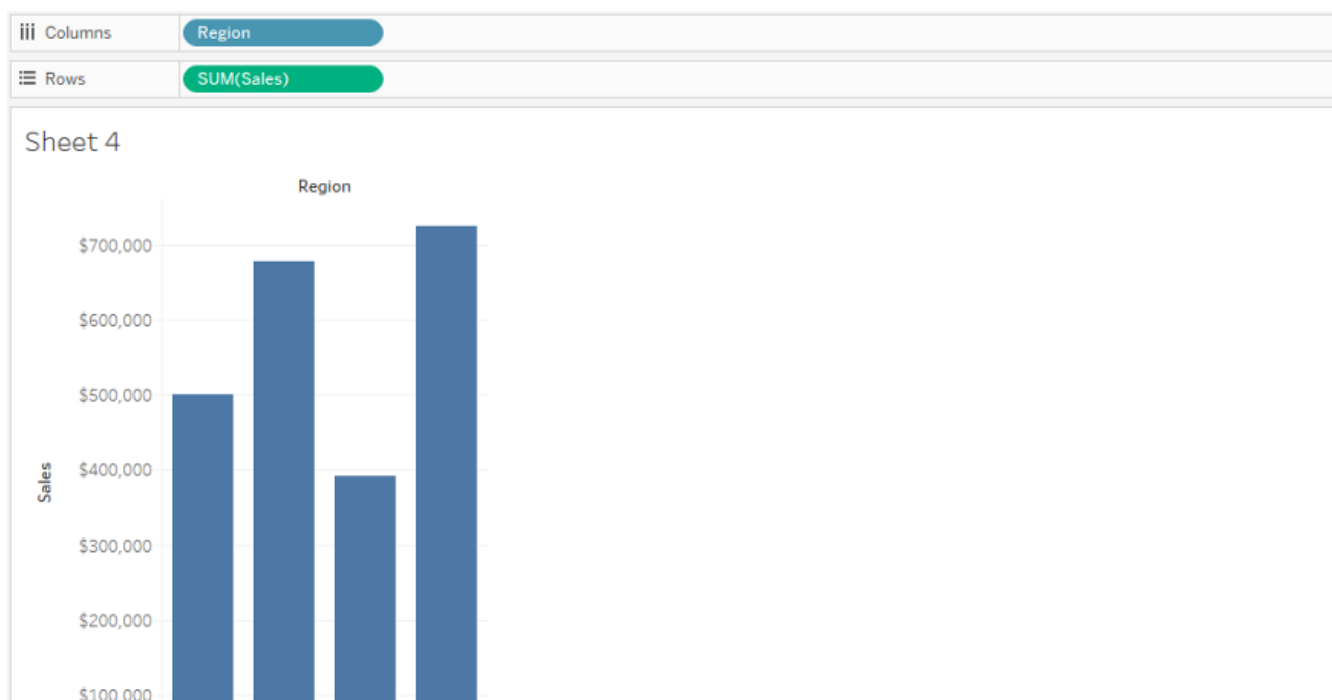
Syntax of an LOD Expression

A level of detail expression has the following structure:

```
{[FIXED | INCLUDE | EXCLUDE]<dimension declaration> : <aggregate expressions>}
```

Step 1: Set up the Visualization

1. Open Tableau Desktop and connect to the **Sample-Superstore** saved data source.
2. Navigate to a new worksheet.
3. From the **Data** pane, under Dimensions, drag Region to the Columns Shelf.
4. From the **Data** pane, under Measures, drag Sales to the Rows Shelf. A bar chart showing the sum of sales for each region appears.



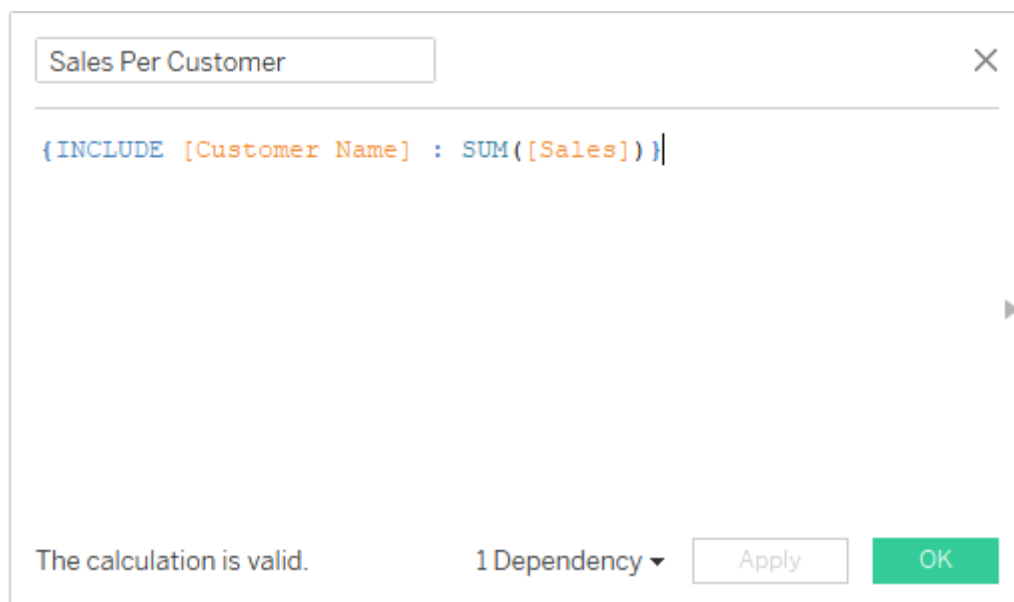


Step 2: Create the LOD expression

Instead of the sum of all sales per region, perhaps you want to also see the average sales per customer for each region. You can use an LOD expression to do this.

1. Select Analysis > Create Calculated Field.
2. In the Calculation editor that opens, do the following:
 - Name the calculation, Sales Per Customer.
 - Enter the following LOD expression:

```
{INCLUDE [Customer Name] : SUM([SALES]) }
```



3. When finished, click OK. The newly created LOD expression is added to the Data pane, under Measures.

Step 3: Use the LOD expression in the visualization

1. From the **Data** pane, under Measures, drag Sales Per Customer to the Rows shelf and place it to the left of SUM(Sales).
2. On the Rows shelf, right-click Sales Per Customer and select Measure (Sum) > Average. You can now see both the sum of all sales and the average sales per

customer for each region. For example, you can see that in the Central region, the sales totaled approximately **500,000 USD** with an average sale for each customer being approximately **800 USD**.



Data Sources supporting LOD Expressions

Data Source	Supported/Not Supported
Action Vectorwise	Not Supported.
Amazon EMR Hadoop Hive	Supported Hive 0.13 onwards.
Amazon Redshift	Supported.
Aster Database	Supported version 4.5 onwards.
Cloudera Hadoop	Supported Hive 0.13 onwards.
Cloudera Impala	Supported Impala 1.2.2 onwards.
Cubes (Multidimensional Data Sources)	Not Supported.
DataStax Enterprise	Not Supported.
EXASOL	Supported.
Firebird	Supported version 2.0 onwards.
Generic ODBC	Limited. Depends on Data Source.
Google Big Query	Supported for Standard SQL, not Legacy.
IBM DB2	Supported version 8.1 onwards.

MARKLOGIC	Supported version / .0 onwards.
SAP HANA	Supported.
SAP Sybase ASE	Supported.
SAP Sybase IQ	Supported version 15.1 onwards.
Spark SQL	Supported.
Splunk	Not Supported.
Tableau Data Extract	Supported.
Teradata	Supported.
Vertica	Supported version 6.1 onwards.
Microsoft Access	Not Supported.
Microsoft Jet-Based Connections	Not Supported.
Hortonworks Hadoop Hive	Supported Hive 0.13 onwards. On version 1.1 of HIVE, LOD expressions that produce cross-joins are not reliable.
IBM BigInsights	Supported.
Microsoft SQL Server	Supported SQL Server 2005 onwards.
MySQL	Supported.
IBM PDA (Netezza)	Supported version 7.0 onwards.
Oracle	Supported version 9i onwards.
Action Matrix (ParAccel)	Supported version 3.1 onwards.
Pivotal Greenplum	Supported version 3.1 onwards.
PostgreSQL	Supported version 7.0 onwards.
Progress OpenEdge	Supported.

Table Calculations vs LOD

LOD Expressions are not a new form of Table Calculations. Though they can replace many Table Calculations, their main aim is to open new possibilities. LOD Expressions and Table Calculations operate differently.

Table Calculations	LOD Expressions
Table Calculations are generated by Query Results .	LOD Expressions are generated as a part of the Query to the underlying Data Source . They are expressed as a nested select, so, depending upon DBMS performance.
Table Calculations can only produce results either equal to or less	LOD can produce results independent of

results either equal to or less granular than the said LOD.	said LOD.
Dimensions that control the operations of a table are separate from the calculation syntax.	Dimensions that control the operations of an LOD expression are embedded in the expression itself.
Table Calculations are used as aggregated measures .	LOD expressions can be used in other constructs.
Filters on Table Calculations act as a HIDE .	Filters on the LOD act as an EXCLUDE .

Limitations of LOD

The following are the constraints that apply for LOD expressions.

- LOD expressions that reference floating-point measures tend to behave in an unreliable fashion when used in a view that requires a comparison of the values in the expression.
- LOD is not shown on the Data Source page.
- When referencing a parameter in a dimensionality declaration, always use the parameter name, and not the parameter value.
- With data blending, the linking field from the primary data source must be in the view before you can use a Level Of Detail expression from the secondary data source.

In addition, some data sources have complexity limits. Tableau will not disable calculations for these databases, but query errors are a possibility if calculations become too complex. With this, we come to an end to this article on LOD In Tableau.

If you wish to check out more articles on the market's most trending technologies like Artificial Intelligence, DevOps, Ethical Hacking, then you can refer to [Edureka's official site](#).

Do look out for other articles and videos in the series which will help you understand various concepts of Tableau.

1. [Tableau Tutorial](#)

2. [What is Tableau?](#)

3. [Tableau Functions](#)

4. [Tableau Dashboard](#)
 5. [Tableau Charts](#)
 6. [Tableau Tips and Tricks](#)
 7. [Step By Step Guide To Learn Tableau Public](#)
 8. [Tableau Desktop vs Tableau Public vs Tableau Reader](#)
 9. [How to Create and Use Parameters in Tableau?](#)
 10. [What are Sets in Tableau And How To Create Them](#)
 11. [Data-blending](#)
 12. [Donut Charts in Tableau](#)
 13. [Top 50 Tableau Interview Questions You Must Prepare In 2020](#)
 14. [How And When To Use Different Tableau Charts](#)
-

Originally published at <https://www.edureka.co> on February 20, 2019.

[Data Science](#)[Tableau](#)[Lod In Tableau](#)[Level Of Detail](#)[Tableau Training](#)[About](#) [Write](#) [Help](#) [Legal](#)

Get the Medium app

