TABLEAU LOD NOTES

ADVANCED LEVEL

EDITION 2025

Developed by Swapnjeet S (Data Tutorials)

Why Use LOD Expressions in Tableau?

Level of Detail (LOD) expressions are a powerful feature in Tableau that allow analysts to perform calculations at different levels of granularity, independent of the visualization's current context. They provide flexibility and control over how data is aggregated and analyzed, enabling precise calculations that might not be possible with standard aggregations.

Reasons to Use LOD Expressions

1. Granular Control Over Aggregations
   - LOD expressions allow you to specify exactly how and at what level data should be aggregated, regardless of the dimensions in the view.
   - Example: Calculating the average sales per customer even when the visualization shows sales by region or product category.
2. Consistent Calculations Across Views
   - LOD ensures consistent results across different dashboards or sheets that use varying levels of detail.
   - Example: A fixed sales target for a region can be applied universally, even when analyzing data at a daily, monthly, or yearly level.
3. Combining Data at Different Levels
   - You can compare metrics calculated at different granularities, such as customer-level data with region-level data, in a single view.
   - Example: Calculating market share by comparing product category sales to total sales.
4. Overcoming Limitations of Default Aggregations
   - Tableau's default aggregations are tied to the dimensions in the view, which might not always align with your analytical needs. LOD expressions break free from this limitation.
5. Customizing Granularity for Business Logic

- LOD expressions enable calculations that align with specific business rules, such as calculating bonuses, sales targets, or commissions based on unique criteria.

6. Simplifying Complex Calculations
    - Without LOD, complex calculations often require restructuring data or creating additional data sources. LOD expressions simplify such tasks.

Advantages of LOD Expressions

1. Flexibility in Analysis
    - Perform calculations at any level of detail—higher, lower, or the same level as the current view.
    - Example: Comparing total sales across regions while viewing data at the product category level.
2. Improved Accuracy
    - Avoid errors caused by unintended aggregations or filters. LOD ensures that calculations are accurate and aligned with the intended granularity.
    - Example: Calculating a global average price per product, unaffected by filters in the view.
3. Dynamic Insights
    - LOD allows users to create dynamic metrics like rolling averages, running totals, and percentage contributions.
    - Example: Calculating a fixed baseline of yearly sales to analyze growth rates.
4. Enhanced Dashboard Performance
    - By consolidating complex calculations into LOD expressions, you can reduce the dependency on multiple calculated fields, improving dashboard performance.
5. Ease of Collaboration
    - LOD expressions make calculations reusable and easy to understand for other Tableau users, simplifying collaboration and maintenance.
6. Integration with Filters
    - LOD expressions can ignore, include, or fix dimensions, offering more flexibility when working with filtered datasets.
    - Example: Creating metrics that remain unaffected by filters, like total company-wide sales, regardless of filter selections.

Level of Detail (LOD) Expressions in Tableau

Level of Detail (LOD) expressions allow you to calculate aggregations at different granularities in Tableau, regardless of the visualization's level of detail. There are three main types of LOD expressions: **INCLUDE**, **EXCLUDE**, and **FIXED**. Each serves specific use cases in data analysis.

---

1. INCLUDE LOD

Purpose:

The INCLUDE LOD allows you to add dimensions to your aggregation, making it more granular than the default level of detail in the visualization.

Syntax:

{ INCLUDE [Dimension1], [Dimension2]: Aggregation([Measure]) }

Key Points:

- It works in conjunction with the dimensions already in the view.
- Useful for performing calculations at a finer granularity than the current visualization.

Example Use Case:

You want to calculate **average sales per product category and sub-category** while the visualization shows data at the Region level.

Steps:

1. Dataset fields: Region, Category, Sub-Category, Sales.
2. Visualization: Region on Rows and SUM(Sales) on Columns.
3. To find the **average sales per product sub-category**, use the following LOD:

{ INCLUDE [Sub-Category]: AVG([Sales]) }

4. Add this calculated field to the tooltip or another visualization layer.

Example 1: Average Sales per Product Sub-Category Across All Regions

Scenario:
You work for a retail company, and you want to analyze the **average sales per product sub-category** for each region. However, the current visualization shows data only at the **region** level.

Data Fields:

- Region

- Category
- Sub-Category
- Sales

LOD Expression:

{ INCLUDE [Sub-Category]: AVG([Sales]) }

Steps and Explanation:

1. **Visualization Setup:**
   Place Region on Rows and SUM(Sales) on Columns.
2. **Challenge:**
   At this level, Tableau aggregates sales by region. To add granularity and compute the average sales for sub-categories within each region, use INCLUDE.
3. **Calculated Field Result:**
   The LOD expression will compute average sales at the Sub-Category level within each Region, even though Sub-Category is not in the view.
4. **Outcome:**
   Add this calculated field to the tooltip or as a secondary layer in your visualization to show detailed insights.

Example 2: Employee Performance in a Team (Including Employee Dimension)

Scenario:
Your organization tracks **total sales performance at the team level**, but you also want to see the performance of individual employees within each team.

Data Fields:

- Team
- Employee Name
- Sales

LOD Expression:

{ INCLUDE [Employee Name]: SUM([Sales]) }

Steps and Explanation:

1. **Visualization Setup:**
   Place Team on Rows and SUM(Sales) on Columns.

2. **Challenge:**
   The view aggregates sales at the team level. To include employee-level granularity, use the INCLUDE LOD.
3. **Calculated Field Result:**
   This field calculates the total sales per employee within their respective teams, even though the view shows data at the team level.
4. **Outcome:**
   Add this field to the tooltip or use it in another chart to compare team-level performance with individual contributions.

Example 3: Monthly Sales Per Store Across a City

Scenario:
You manage multiple stores in a city, and the current report shows **total monthly sales for the city**. You need to compute the **monthly average sales per store** to understand store-level performance.

Data Fields:

- City
- Store
- Month
- Sales

LOD Expression:

{ INCLUDE [Store]: AVG([Sales]) }

Steps and Explanation:

1. **Visualization Setup:**
   Place City and Month on Rows, and SUM(Sales) on Columns.
2. **Challenge:**
   The data aggregates monthly sales at the city level. Adding granularity for individual stores requires an INCLUDE LOD expression.
3. **Calculated Field Result:**
   The field calculates average monthly sales for each store within the city, irrespective of the dimensions displayed in the view.
4. **Outcome:**
   Use this field to create an additional layer in the visualization or compare city-level and store-level performance.

Example 4: Average Order Size by Product and Customer

Scenario:
A company wants to analyze the **average order size** by product category for each customer, while the visualization focuses on customer-level sales only.

Data Fields:

- Customer Name
- Product Category
- Order Size

LOD Expression:

{ INCLUDE [Product Category]: AVG([Order Size]) }

Steps and Explanation:

1. **Visualization Setup:**
   Place Customer Name on Rows and SUM(Order Size) on Columns.
2. **Challenge:**
   At this level, you lose visibility into how each product category contributes to the order size for a customer.
3. **Calculated Field Result:**
   The field includes the Product Category dimension to calculate the average order size for each product category for a customer.
4. **Outcome:**
   Add this field as a layer in your visualization to drill down into product-level insights within each customer's data.

Example 5: Attendance Rate per Class Across Multiple Schools

Scenario:
A school district tracks **attendance rates per school**, but you want to analyze the **attendance rate for individual classes** while still displaying data at the school level.

Data Fields:

- School Name
- Class Name
- Attendance Percentage

LOD Expression:

{ INCLUDE [Class Name]: AVG([Attendance Percentage]) }

Steps and Explanation:

1. **Visualization Setup:**
   Place School Name on Rows and AVG(Attendance Percentage) on Columns.
2. **Challenge:**
   Aggregation at the school level hides individual class-level attendance data.
3. **Calculated Field Result:**
   This field calculates the average attendance rate for each class within the school.
4. **Outcome:**
   Display this calculated field as a detailed view for school administrators to understand which classes have low attendance rates.

Key Points for INCLUDE LOD in Real-Time Scenarios

1. **Adds Granularity:**
   INCLUDE is perfect for scenarios where additional dimensions are required for finer calculations without altering the overall visualization.
2. **Dynamic Behavior:**
   The calculated value adapts based on the existing dimensions in the view.
3. **Common Use Cases:**
   - Per-unit calculations (e.g., sales per product or per employee).
   - Analyzing sub-level details in aggregated views (e.g., store-level data in city-level views).
   - Combining high-level and detailed-level metrics in a single report.

Use these real-life scenarios to master the practical application of INCLUDE LOD expressions and their potential to solve complex analytical challenges.

2. EXCLUDE LOD

Purpose:

The EXCLUDE LOD removes specific dimensions from the aggregation, making it less granular than the visualization's current level of detail.

Syntax:

{ EXCLUDE [Dimension1], [Dimension2]: Aggregation([Measure]) }

Key Points:

- It excludes certain dimensions while performing calculations.
- Useful for generating totals or averages without specific granularities.

Example Use Case:

You want to calculate **total sales for all regions** while showing sales per state in the view.

Steps:

1. Dataset fields: State, Region, Sales.
2. Visualization: State on Rows and SUM(Sales) on Columns.
3. To calculate total sales across all regions, use the following LOD:

{ EXCLUDE [State]: SUM([Sales]) }

4. Add this calculated field to the tooltip to display the total regional sales.

Example 1: City-Level Sales Without Store-Level Granularity

Scenario:
You manage multiple stores within cities and want to analyze **total sales per city** but exclude the effect of individual store-level granularity in your calculations.

Data Fields:

- City
- Store
- Sales

LOD Expression:

{ EXCLUDE [Store]: SUM([Sales]) }

Steps and Explanation:

1. **Visualization Setup:**
   Place City and Store on Rows, and SUM(Sales) on Columns.

2. **Challenge:**
   Including Store in the view creates store-level granularity, but you want to compute total city-level sales ignoring individual stores.
3. **Calculated Field Result:**
   The EXCLUDE LOD removes the Store dimension from the calculation and computes city-level sales.
4. **Outcome:**
   This calculation ensures that store-specific contributions are ignored, focusing only on city-wide metrics.

Example 2: Average Sales Per Region Excluding Year-Level Granularity

Scenario:
You analyze sales trends across multiple regions and years. However, you need to calculate **average sales per region** ignoring year-specific variations.

Data Fields:

- Region
- Year
- Sales

LOD Expression:

{ EXCLUDE [Year]: AVG([Sales]) }

Steps and Explanation:

1. **Visualization Setup:**
   Place Region and Year on Rows, and SUM(Sales) on Columns.
2. **Challenge:**
   Year-level data skews the calculation, but you need to focus on average sales per region.
3. **Calculated Field Result:**
   The EXCLUDE LOD removes Year as a dimension, aggregating average sales at the regional level.
4. **Outcome:**
   The resulting metric provides a consistent view of sales performance across regions, independent of year-specific data.

Example 3: Monthly Sales Excluding Week-Level Details

Scenario:
You track sales data weekly but need to compute **total monthly sales**, excluding the weekly granularity in your calculations.

Data Fields:

- Month
- Week
- Sales

LOD Expression:

{ EXCLUDE [Week]: SUM([Sales]) }

Steps and Explanation:

1. **Visualization Setup:**
   Place Month and Week on Rows, and SUM(Sales) on Columns.
2. **Challenge:**
   Week-level granularity makes it difficult to analyze total sales for the month.
3. **Calculated Field Result:**
   The EXCLUDE LOD removes the Week dimension and calculates total monthly sales.
4. **Outcome:**
   This approach ensures a clear and concise view of monthly performance without weekly breakdowns.

Example 4: Department-Level Attendance Rate Excluding Individual Employee Data

Scenario:
You manage employee attendance data and want to compute **attendance rates for departments**, excluding employee-level details.

Data Fields:

- Department
- Employee Name
- Attendance Rate

LOD Expression:

{ EXCLUDE [Employee Name]: AVG([Attendance Rate]) }

Steps and Explanation:

1. **Visualization Setup:**
   Place Department and Employee Name on Rows, and AVG(Attendance Rate) on Columns.
2. **Challenge:**
   The view focuses on individual employee attendance, but you want a higher-level departmental average.
3. **Calculated Field Result:**
   The EXCLUDE LOD removes Employee Name from the calculation and aggregates attendance rates at the department level.
4. **Outcome:**
   This provides a summary of departmental attendance rates, excluding individual employee variations.

Example 5: Total Revenue Excluding Product-Level Details

Scenario:
Your report displays total revenue by category but includes product-level data. You need to compute **total revenue by category**, ignoring product-level granularity.

Data Fields:

- Category
- Product
- Revenue

LOD Expression:

{ EXCLUDE [Product]: SUM([Revenue]) }

Steps and Explanation:

1. **Visualization Setup:**
   Place Category and Product on Rows, and SUM(Revenue) on Columns.
2. **Challenge:**
   The report becomes cluttered with product-level details, making it hard to focus on category performance.
3. **Calculated Field Result:**
   The EXCLUDE LOD removes the Product dimension and calculates category-level revenue.
4. **Outcome:**
   The clean category-level metric improves readability and enables high-level analysis.

Example 6: Total Sales Excluding Customer Segment

Scenario:
You analyze sales data across customer segments and want to compute **overall sales** without segment-specific details.

Data Fields:

- Customer Segment
- Region
- Sales

LOD Expression:

{ EXCLUDE [Customer Segment]: SUM([Sales]) }

Steps and Explanation:

1. **Visualization Setup:**
   Place Customer Segment and Region on Rows, and SUM(Sales) on Columns.
2. **Challenge:**
   The inclusion of customer segments adds unnecessary complexity.
3. **Calculated Field Result:**
   The EXCLUDE LOD removes Customer Segment and aggregates total sales at the regional level.
4. **Outcome:**
   This calculation provides a simplified view of sales performance by region, ignoring customer segmentation.

Key Points for EXCLUDE LOD in Real-Time Scenarios

1. **Removes Dimensions:**
   EXCLUDE is used to ignore specific dimensions in the calculation while retaining them in the visualization.
2. **Focuses on Higher-Level Insights:**
   It enables aggregation and metric computation at higher levels of granularity by excluding unnecessary details.
3. **Common Use Cases:**
   - Aggregating metrics (e.g., revenue by category ignoring product-level details).
   - Summarizing data at a higher level (e.g., attendance rates by department).
   - Simplifying views for executive reporting or presentations.

EXCLUDE LOD is a powerful tool for simplifying calculations, decluttering views, and focusing on higher-level insights in Tableau.

3. FIXED LOD

Purpose:

The FIXED LOD allows you to calculate aggregations at a fixed level of detail, irrespective of the dimensions present in the view.

Why FIXED LOD is Widely Used

1. **Precision in Aggregation**
   - FIXED LOD allows you to define exactly at which level data should be aggregated, providing control over calculations.
   - Example: Calculating total sales per region, even when analyzing data at the product level.
2. **Filter Independence**
   - FIXED LOD ignores filters applied in the view unless explicitly mentioned, making it ideal for creating metrics unaffected by user interactions.
   - Example: Calculating company-wide revenue, unaffected by a region filter.
3. **Reusability Across Views**
   - FIXED LOD calculations can be reused across different dashboards and sheets, ensuring consistency.
   - Example: A predefined customer segmentation metric can be used universally.
4. **Simplifies Complex Logic**
   - Without FIXED LOD, achieving similar calculations often requires restructuring data or creating additional data sources. FIXED simplifies these tasks.
   - Example: Calculating the average sales per customer across all transactions.
5. **Ideal for Baseline Metrics**
   - FIXED LOD is perfect for baseline metrics, such as total revenue, total customers, or predefined targets, which remain constant across analyses.
   - Example: Yearly sales targets for regions.

Advantages of FIXED LOD

1. **Control Over Granularity**
   - FIXED LOD allows you to bypass the dimensions in the visualization and focus on specific levels of granularity.
2. **Consistency Across Filters**

- By ignoring context filters, FIXED ensures calculations remain consistent regardless of filter changes in the view.

3. **Simplifies Complex Metrics**
   - Enables easy calculation of metrics like market share, unique counts, and baselines without complex SQL queries or data restructuring.
4. **Improves Performance**
   - FIXED calculations are processed once and reused, reducing the load on Tableau's query engine.

Syntax:

{ FIXED [Dimension1], [Dimension2]: Aggregation([Measure]) }

Key Points:

- The calculation is independent of the dimensions in the view.
- Useful for creating stable metrics (e.g., customer lifetime value, total sales per year).

Example Use Case:

You want to calculate **total sales per customer** regardless of the view's level of detail.

Steps:

1. Dataset fields: Customer Name, Order ID, Sales.
2. Visualization: Order ID on Rows and SUM(Sales) on Columns.
3. To calculate total sales per customer, use the following LOD:

{ FIXED [Customer Name]: SUM([Sales]) }

4. Add this calculated field to your visualization.

Best Practices for Using FIXED LOD

1. **Use When Filters Should Be Ignored**
   - FIXED LOD is ideal for metrics like total sales or unique customers, where filters like product or region shouldn't affect calculations.
2. **Combine with Context Filters if Needed**
   - If you need FIXED LOD to respect some filters, set those filters as context filters.
3. **Test and Validate**
   - Always cross-check FIXED LOD results against expectations to ensure accuracy, especially when working with complex datasets.

4. **Use with Other LODs**
   - Combine FIXED with INCLUDE or EXCLUDE for layered and advanced calculations.

Example 1: Total Sales by Region Regardless of the View

Scenario:
You need to calculate **total sales for each region**, irrespective of other dimensions like product, category, or year in the view.

Data Fields:

- Region
- Product
- Year
- Sales

LOD Expression:

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

Explanation:

1. This calculation fixes the aggregation to the Region level, ignoring any additional dimensions present in the view.
2. The Region acts as the anchor for this fixed calculation.
3. The result will always display the total sales for each region, regardless of filters or other dimensions in the visualization.

Example 2: Average Revenue Per Customer

Scenario:
You are analyzing customer spending behavior and want to calculate the **average revenue per customer** irrespective of product categories or transaction details in the view.

Data Fields:

- Customer ID
- Product Category
- Revenue

LOD Expression:

{ FIXED [Customer ID]: SUM([Revenue]) }

Steps and Explanation:

1. **Fixing to Customer Level:**
   This fixes the calculation to the Customer ID level, summing up revenue for each customer.
2. **Averaging Across Customers:**
   To get the average revenue, wrap this LOD calculation in another aggregation:

tableau

Copy code

AVG({ FIXED [Customer ID]: SUM([Revenue]) })

3. **Outcome:**
   You can now see the average revenue per customer, independent of other dimensions like product or transaction date.

Example 3: Profit Contribution by Product Category Across All Regions

Scenario:
You want to analyze the **profit contribution of each product category**, ignoring regional or sub-category-level granularity.

Data Fields:

- Region
- Product Category
- Sub-Category
- Profit

LOD Expression:

{ FIXED [Product Category]: SUM([Profit]) }

Explanation:

1. This fixes the calculation at the Product Category level.
2. Even if you include regions or sub-categories in your view, the profit calculation will remain fixed to the product category.

Example 4: Employee Count by Department

Scenario:
You have a dataset with employee details and need to calculate the **total number of employees in each department**, independent of their location or roles.

Data Fields:

- Department
- Location
- Role
- Employee ID

LOD Expression:

{ FIXED [Department]: COUNTD([Employee ID]) }

Explanation:

1. The calculation counts unique employees (COUNTD) and fixes it at the department level.
2. Even if you add Location or Role to your visualization, the employee count will still represent the department-level totals.

Example 5: Monthly Sales Target vs. Actual Sales

Scenario:
Your sales team has monthly targets, and you want to compare **monthly sales target** with the **actual sales**, fixing the target calculation at the month level.

Data Fields:

- Month
- Region
- Sales
- Target

LOD Expression:

{ FIXED [Month]: SUM([Target]) }

Steps and Explanation:

1. **Fixing Targets:**
   This fixes the Target calculation to the Month level, ensuring it's consistent across regions or other dimensions.
2. **Comparison:**
   Create another calculated field to compare the fixed target with actual sales:

tableau

Copy code

SUM([Sales]) - { FIXED [Month]: SUM([Target]) }

3. **Outcome:**
   The calculation provides the variance between actual sales and targets for each month.

Example 6: Customer Lifetime Value (CLV)

Scenario:
You are tasked with calculating **customer lifetime value (CLV)**, which is the total revenue generated by a customer over their lifetime, irrespective of transaction dates or product details.

Data Fields:

- Customer ID
- Transaction Date
- Revenue

LOD Expression:

{ FIXED [Customer ID]: SUM([Revenue]) }

Explanation:

1. This fixes the calculation to the Customer ID level.
2. The result will show the total revenue contributed by each customer, ignoring transaction-level granularity.

Example 7: Market Share by Product Category

Scenario:
You want to calculate the **market share of each product category** relative to total sales across all categories.

Data Fields:

- Product Category
- Sales

LOD Expression:

1. **Total Sales (Fixed):**

{ FIXED : SUM([Sales]) }

This calculates the total sales across all categories.

2. **Market Share Calculation:**

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

3. **Explanation:**
   The fixed total sales value is used as the denominator to compute the proportion of each category's sales.

Example 8: Yearly Sales Per Region

Scenario:
You need to analyze **yearly sales for each region**, regardless of product or sub-category details.

Data Fields:

- Year
- Region
- Sales

LOD Expression:

{ FIXED [Year], [Region]: SUM([Sales]) }

Explanation:

1. The calculation fixes both Year and Region as anchor dimensions.
2. The result shows yearly sales for each region, irrespective of other dimensions like product or sub-category.

Key Takeaways for FIXED LOD in Real-Time Scenarios

1. **Anchor Dimensions:**
   FIXED explicitly defines the dimensions at which calculations are performed.
2. **Independent of View Context:**
   FIXED ignores the filters and granularity in the view unless the filter is applied to the same dimensions explicitly.
3. **Common Use Cases:**
   - Aggregating metrics (e.g., customer lifetime value, total sales by region).
   - Comparing metrics (e.g., sales vs. targets).
   - Creating consistent baselines for further calculations (e.g., market share).

The FIXED LOD is one of the most versatile tools in Tableau, enabling analysts to calculate metrics with precision and flexibility.