



---

# On-Demand Training: Analysis with Cubes and MDX Transcript

Welcome to this video on MDX. You can download the Exercises workbook or follow along in your own copy of Tableau.

## MDX

OLAP or “cube” data sources use the MDX language rather than SQL to define calculations. In Tableau Desktop, you do not write an MDX query to bring in the original data set; Tableau handles bringing in all predefined dimensions and measures for you. You can however define new measures and dimensions by creating what is called a “Calculated Member”.

In this video we will not be teaching MDX, but rather showing how to use it in Tableau. There are many online resources that provide knowledge about the actual MDX language.

One of the many uses of calculated members is to define new groupings that were not created by the cube’s architect.

## Connecting

For the following examples, we will be using a local cube version of the Superstore data used in the Connecting to and Using Cubes video.

- Select the “Local cube file” option, then press “Browse” and find the .cub file, then press OK.
- Next you will see a screen with two steps.
  - For Step 1: Select a Database, we choose “SuperCube”, and for
  - Step 2: Select a Cube, we also choose SuperCube.
- Now click Sheet 1 in the bottom tab pane.

## Creating a Calculated Member for Grouping

For this example, we’re going to group some of the members of the Sub-Category “Office Supplies” together into a “Small Items” group, much as you may have seen in other Tableau trainings.

- Open the Products dimension, then open the Product Groups hierarchy.
- Bring out the Product Category and the Product Sub-Category onto the Columns Shelf,
- and Sales onto the Rows shelf, then Sort on the Sales Axis.
- We can see the Sub-Categories we’d like to group together into Small Items, but if we select them and hover, there is no Grouping icon.
- We’ll have to write a Calculated Member to define them grouped together.
  - Right click in the Measures pane and choose “Calculated Members...”.
  - Press the New button under the first pane on the right. In the Name box, type “Small Items”.
  - Next we will define where the member belongs in the hierarchy.
  - Rather than creating a new level of hierarchy like grouping using a relational data source, we are creating a new member within the existing hierarchy level that will be composed of other existing

members.

- We want this “Small Items” member to be part of Office Supplies in the Product Group hierarchy.
- Now select the “[Products].[Product Group]” Hierarchy. Lastly, for Parent, choose “Selected Member” then choose “Office Supplies”.
- Click OK.
- Now we need to write out the MDX to group these members together. We are pasting in this calculation to save time, but you can see the basic MDX structure for putting Dimension members together.
  - `[Product].[Product Groups].[Office Supplies].[Envelopes] + [Product].[Product Groups].[Office Supplies].[Pens & Art Supplies] + [Product].[Product Groups].[Office Supplies].[Scissors, Rulers and Trimmers] + [Product].[Product Groups].[Office Supplies].[Labels] + [Product].[Product Groups].[Office Supplies].[Rubber Bands]`
- Members of the hierarchy are referenced using the “bracket and dot” notation, and the plus sign says to bring them together.
- If you are following along, try and type in the first two, and you can complete the others later by looking at the names in the viz.
- Press the “Check Formula” button to make sure you haven’t made any errors. If the calculation is valid, press ok.

You’ll now see there is a new bar in the viz for “Small Items”, alongside the sub-categories that make it up. This is very different from a relational data source, but the cube also allows us to filter out those sub-categories without affecting the Small Items sub-category. Let’s select them and then press “Exclude”.

You’ll also notice on the Product Sub-Category hierarchy there is a little equals sign to indicate we’ve created a calculated member in that hierarchy.

## Cohort Calculations

When working with a relational data source in Tableau, it is common to define a calculated field to only return a measure when a dimension has a certain value.

- For example, we might make a “West Sales” measure with the calculation
  - “ IF [Region] = ‘West’ then [Sales] END “.
- These are referred to as Cohort Calculations.

If you try and create a Calculated Field using a cube data source, you’ll notice that only measures are available.

- For any logic based on dimensions, we must use MDX to create a calculated member.
- We will create a new calculated member using a right-click just as before, naming it “West Sales”. This time, we want to place it in the [Measures] hierarchy.

- Again we will paste in the calculation.
  - ([Measures].[Sales] , [Customer].[Customer Geography].[Region].[West] )
    - This is the syntax to return a specific part of the larger Sales Measure.
    - We're asking for the [Sales] member of the Measures hierarchy, then placing a comma, then putting in the place in the hierarchy we want to limit the sales to, in this case the West region within the customer geography hierarchy.
- Unlike Tableau calculations, there are no hints or function names available for the MDX calculations. You will have to learn MDX and know what your desired calculations and member names are on your own. However, all of the Calculated Members can be edited from this single menu.
- Cubes do not behave exactly like relational data sources; rather than showing Null for all of the regions other than West, a cohort defined this way will still show values for all regions in the hierarchy, but the measure value will be identical for all of them (it will be the value for West only). Keep this in mind as you work with MDX and calculated members to get the results you want in Tableau.

There are many other examples of uses of MDX and Calculated Members in the Tableau Knowledge Base.

## Conclusion

Thank you for watching this training video on Cubes and MDX. We invite you to continue with the On-Demand Training videos to learn more about using Tableau.