Notes:

Data size: Tabular is limited on the data size up to 1 Terabyte. Multi-dimensional can see up to 24 terabytes’.

Hardware Resources: Tabular requires more memory to run. Multi-dimensional uses Disk IO (input/output) in getting its data information (Tabular does not).

Query Languages: Tabular can us MDX and DAX (Released in 2010 and easier to learn than MDX). Multi-dimensional can only use MDX. Certain programs like “PowerView “only works with the tabular model.

Query Performance: Tabular is (easy) fast. Multi-dimensional (hard- difficult) fast. With Multi-dimensional you must worry about query performance with attributes, how things are set-up with relationships and how the database is set up.

Model Complexity: Complicated models can run into performance problems when implementing the model into tabular (tabular does not do well with many-to-many relationships).

Security: Both Tabular and Multi-dimensional handle Row Level security. Multi-dimensional also handles Cell Level Security, but can have all kinds of bottle necks, is very complicated to put in place and not used often to begin with.

Typical Development Effort: Model Complexity Medium-High for Multi-dimensional and Medium-Low for Tabular Models. You can import a Power Pivot Project Model into a Tabular Model, which allows you to do rapid prototyping (speeding up the time to solution).

Licensing: Multi-dimensional supports all licensing and Tabular only supports BI and Enterprise Licensing.

Tabular Limitations: Many-To-Many, Actions \*possible but not in the GUI, Translations, Writeback, Data Mining, Aggregations (not necessary).

<https://bidshelper.codeplex.com/>

20 Minute Tabular (Demo)

1. Created views [v Dim Customer], [v Dim Date], [v Dim Product], [v Dim Product Category], and [v Dim Product Subcategory] of the tables in the AdventureWorksDW2012 database with drop capabilities.
2. Created a new project called “20MinuteTabular” with a specific location using Analysis Services Tabular Project as the Project type.
3. Used the “import from data source” button icon and chose the “Microsoft SQL Server” as my connection type. Put in the Friendly Name, SQL Server instance (. “localhost”) for the connection, and pointed to the database name. Then I chose from the tables that already exist, selected the views that were needed, and picked finish to finish the wizard. Note: When you select views, you need to set up the relationships once this process if finished.
4. Setting up the relationships are done by first selecting the little “Diagram” button at the far-right bottom corner. You then position the tables in such a way the relationships are to be set up with the [Fact Internet Sales] table in the very middle. Dragged from the [Fact Internet Sales] the [Customer Key] to the Customer table onto the other [Customer Key] in the Customer table. Note: When you drag to create relationships, you drag away from the Fact table and/or from Parent to Child of existing tables.
5. Right click on the Customer table and choose “Go To”. Shows only the Customer columns with the data. Next, right click on the [First Name] column and chose “Insert Column” (creates a new column on the left of the [First Name] column).
6. Next, place a formula on the column using DAX =[LastName]&","&[FirstName] and select the calculated column header. The last name with the first name appears in all the rows. Rename the calculated column to “[Customer Name]”, then hide the [First Name], [Last Name] columns by right clicking on the columns and selecting “Hide from Client Tools”. Hiding from Client Tools will hide the column and not show up in the field explorer.
7. Choose the “Diagram” button in the far-right corner once again and you see the fields that were hidden. Note: You can hide other columns by selecting one or multiple columns and right clicking
8. Right click on the DimDate Column, select “Go To” and then change the [Full Date Alternate Key] name to “Date”.
9. While the column is highlighted, in the properties section, change the “Date Format” from general to “Short Date”. Important: Mark a column in one of the tables as the Date Table “Date” (highlighted), essentially gives access to calculated date functions.
10. Click on the “Analyze in Excel” icon in the upper left corner and then choose “current user”. Creates an Excel sheet with a Pivot Report View of all the data that is seen.
11. Go back to Visual Studio. In “Diagram” (lower right corner), select [Date], [Month Name], [Calendar Quarter], [Calendar Year], then right click and choose on the context menu “Create Hierarchy”. Give the Hierarchy the name “Calendar YQMD” and arrange the columns as needed in the hierarchy.
12. Go back to the Excel Pivot Table that you made earlier and refresh by picking on the Pivot Table and choosing “Refresh”. Pick your hierarchy and you notice that the months are out of order (not alphabetically in order).
13. In Visual Studio, right click on the Date Table and pick “Go To” once again. Select [Month Name] and then pick the “Sort by column” button in the tool bar. Sort by “[Month Number Of Year]”. Then switch back over to the pivot table and refresh the pivot table by right clicking on the pivot table and choosing “refresh”.
14. Next we will make a **product hierarchy**. In-order to make a product hierarchy, we need to create columns to migrate our category and subcategory columns that are needed by referencing these columns directly in the “Grid. While in “Diagram”, right click and pick “Go To” and then select a cell in the very last column at the top that is empty.
15. Place the code in the formula box (ProductSubcategory info): **=RELATED(ProductSubcategory[EnglishProductSubcategoryName])** to pull in the information for the calculated column. Note, that information is migrated into the column.
16. Pick the next column over that is empty and place the formula in the box for the next calculated column **=RELATED(ProductCategory[EnglishProductCategoryName])** to pull the information in for the Product Category Name.
17. Click on “Diagram” button and create your hierarchy as “Category of Product” by selecting EnglishProductSubcategory, EnglishProductCategory and EnglishProductName. Hide the ProductSubcategory and ProductCategory tables.
18. **Creating measures** in the [Internet Sales] table. Right click on the [Internet Sales] table and pick “Go To”, pick the [Sales Amount] measure column, right click the column and pick “**SUM**” from the tool bar (Sigma Symbol). Note, that a calculated column is created.
19. Pick the [Sales Order Number] column and then select the drop-down arrow next to the Sigma Sign and select “**Distinct Count**”. Note, that it created a calculated count.
20. Go back to the Excel sheet pivot table and refresh once again. Note, that **the order number format** needs to be changed from text to “**whole number**”. Click on the newly created calculated measure and in the properties window pick the format box and change it to “**Whole Number**”. Set the “**Thousand Separator**” as **True** to add in the comma.
21. Create a calculated measure for the Year To Date on Internet Sales Amount in a blank cell.

**Internet Sales Amount YTD:=TOTALYTD([Internet Sales Amount],'Date'[Date],All('Date'))**

1. Create a calculated measure for “Internet Sales Amount PY” prior year. Change the format in the properties window to “**Currency**”.

**Internet Sales Amount PY:=CALCULATE([Internet Sales Amount] ,SAMEPERIODLASTYEAR('Date'[Date]))**

1. Create another calculated measure in a cell for Prior Year To Date.

**Internet Sales Amount PYTD:=CALCULATE([Internet Sales Amount YTD],SAMEPERIODLASTYEAR('Date'[Date]))**

1. Select on the toolbar “**Model**”, then “**Roles**” to add a role to give user’s access to your data model.
2. Set-up Deployment on the Tabular Model to SQL Server Analysis Services.