Lessons Learned

Admittedly, the best start right off the bat, would have been to create views based on the tables and manipulate the tables contents ([Date Keys], [Full Name Columns] and other transformations) through the views. Near the end I realized that creating views made things so much easier to manipulate. After making the multidimensional modeling techniques, it came to my attention that it makes a whole lot of difference to create the views first.

Consequently, there is a huge difference between multidimensional modeling and tabular modeling on how easy it is to do certain things. Take for instance creating hierarchies, tabular modeling is quick and easy with very little fuss. Furthermore, you just right click on the table in diagram mode and pick “Create Hierarchy”. In multidimensional modeling, you must go through configurations after dragging and dropping to create the hierarchy while in the dimension and if you configure the hierarchy incorrectly, the system lets you know right away. You will also notice if you don’t process the cube correctly when creating a hierarchy. The relationships tab can be a pain at time in multidimensional modeling, especially if you plan on creating many-many-relationship and reference relationship configurations. Parent-child relationships can be made in multidimensional modeling, but in tabular, you need to create a relationship using the “PATH” function. Role-playing relationships are easy in both multidimensional and tabular models, which consists of importing the table in more than once and then dragging and dropping the relationship from the fact table to the role-playing table’s primary key. Following this further, tabular doesn’t seem to have near the problems while creating relationships.

Certainly, how you define relationships in multidimensional modeling Verses tabular has a profound effect on the capabilities of getting information from tables. In tabular (there are two ways you can define relationships) you can just drag and drop from one key to the other or you can create a relationship using the wizard in the menu using “Table” then “Create Relationship”. Although, in addition to multidimensional modeling, you have the option of creating many-to-many relationships, which you can’t do in tabular, speed and less complexities have their advantages on the tabular side. Moreover, derived columns are much faster and easier to create using DAX, though, you must be in “Grid” view. One of the realizations to creating relationships in multidimensional modeling are that you must have an idea of what you’re doing, otherwise you will get errors during processing. With tabular modeling, you don’t get into these relationship traps as often.

Without a doubt, DAX is becoming a big part of the Business Intelligence communities on the internet and there are so many sources to choose from, but the hardest part is manipulating the DAX code based on what the different websites show with writing code to get results, but I found DAX easier to handle and more sensible when it came to being user friendly. It takes a lot of practice when it comes to understanding the syntax structure of writing DAX code, but this also goes the same way with writing MDX. Both languages are very different from one another but hold logical sequences based on the results you intend to achieve. Once you get deep into DAX, you begin losing touch with MDX, because DAX is so quick and easy to use. Indeed, it is no wonder why so many developers have chosen to use DAX in their projects, but every language has its place that goes without saying with capabilities and draw backs. Finally, as time goes languages seem to be getting better at understanding a humanly response when it comes to programming.