# Syntax & Examples

In this new Post of The Series, I will go over the details for creating a correct Set Expression. Also, I will provide some useful examples for you to get your Hands On right away.

Follow these steps to build up your set expression and use the correct syntax:

1. First of all, you need to define what you want your expression to return. A good method to get it right would be to answer the following questions first:
   o What field will I use in my expression? (For example: 'Sales', containing the sales amount for every invoice.)
   o How will I aggregate the field? (It can be using Sum, Count, Avg, etc.)
   o What **explicit** selections do I need in my expression? Here you define a Field and its value(s). (For example: I want ONLY the 'South' Region. Another example would be to only include values associated with certain Year.)
   o Do I need to **exclude/ignore** some selections or values?
2. After you've answered the questions, you can go on to compse the expression. If you want to **Sum** the **Sales** amount, you would start with something like:
   o *Sum(Sales)*
3. Then, whe need to add the Set portion of the expression.:
   o This portion goes just after the first parenthesis, before the Field Name. The Set Expression will be enclosed in curly brackets: {*set expression*}.
   o After the first curly bracket we add either a dollar sign (which means the record set will be based on the current selections) or a number 1 (meaning we will use the full record set of all the records in the application). We will use the dollar sign to illustrate, since it is the most common, so you will now have *{$}*. Important to note is that the dollar sign can be ommited and the set expression will not be affected. It is good practice, however, to use it.
   o After the Dollar Sign, we define the fields that will play in our set expressions. All of these field-value definitions will be encolsed in less-than and grater-than symbols (< >).  The syntax is *FieldName = {FieldValue}*. If *FieldValue* is a literal or text, you should enclose it in single quotes. If you want to use a search string as the *FieldValue*, enclose it in double quotes. Here are some examples:
     ▪ *{$<Region = {'South'}>} w*ill result in a record set taking the current selections ($) where the Field Region has a value of 'South'.
     ▪ *{$<Year = {2010}>}* will give return a record set based on current selections ($) where the Year is 2010 EVEN if you select something else in the field Year.
     ▪ *{$<Year = {"20*"}>} w*ill give you the record set based on the current selections where the Year matches the search string "20*", meaning all years that begin with "20″.
     ▪ *{$<Year = {">=2007"}>}* Will return a record set where the Year is greater than or equal to 2007. Notice that here we are using a search string.

- ▪ ***{$<Region = {'South', 'North'}, Year = {2010}>}*** will give you a record set based on the current selections where *Region* is equal to *South* **or** *North* **and** *Year* is equal to *2010*.
4. Your final expression should look similar to:
   - o ***Sum({$<Region ={'South'}, Year = {2010}>} Sales)***
5. You can use variables instead of the hardcoded Field Value:
   - o If your variable is a number, use it as ***Field = {$(MyVariable)}***
   - o If your variable is text, use it as ***Field = {'$(MyTextVariable)'}***
   - o If your variable should be used as a search string, use ***Field = {"$(MySearchVariable)"}***
6. Also, you can create calculations to use them as Field Values:
   - o ***Year = {$(=Max(Year))}*** It is just as if you would use a variable (described above), just with an equal sign. NEVER forget the equal sign here.