




MS MDX DISTILLED

SSMS Environment

1. Basic vocabulary:
 - 1.1. **Measure** = Fact = Numbers = Data 
 - 1.2. **Dimension** = Attribute = Descriptive characteristic that categorizes the data (rectangular icon) 
 - 1.3. **Hierarchy** – Logical structure with levels to organize the data (pyramid bowling ball icon) 
2. Construct queries in pieces. Don't progress to the next step until the current step is correct:
 - 2.1. Start with `SELECT/FROM/WHERE`
 - 2.2. Create column output, stub out calculations and named sets (the column axis is required)
 - 2.3. Create row output (the row axis is *not* required)
 - 2.4. Implement calculations and named sets
 - 2.5. Add the rest: `NON EMPTY, FILTER, ORDER`, etc.
3. If a highly abstract, reusable query is not working, start off with a concrete example first and convert it later once it's working
4. If a highly complex query doesn't work and you don't know where the problem is, repeat step 1 (pull out the columns and test, pull out the rows, use comments to enable/disable parts of code, remove/disable the `NON EMPTY` if you suspect results are incorrect, etc.)
5. Verify your results
 - 5.1. Compare grand totals and specific slices with the OLTP and/or staging system
 - 5.2. Test calculations by also displaying original data used in calculations (e.g. percentages)
 - 5.3. Test abstract, reusable queries by plugging in concrete clauses at various levels
 - 5.4. Keep your verification queries. Build a test harness to run periodically (to check for changes in hierarchies, etc.)
6. If you have multiple connections, SSMS may get confused
7. There are usually multiple ways to write queries. If one approach doesn't work, try another
8. The order of the measures is important and drives the output (e.g. `Time Range, {Dollar Sales, Unit Sales}` vs. `{Dollar Sales, Unit Sales}, Time Range`)
9. If you print out columns and no rows, you get a grand total
10. Queries must use consecutive axes (e.g. `axis(0)` and `axis(1)` is valid; `axis(0)` and `axis(2)` is not)
11. Enclose multiple members in the *same* dimension in `{ }`
12. Enclose multiple members in *different* dimensions in `()` or use `*`
13. Do not combine `()` and `*` -- in certain cases this can create an invalid query
14. If `()` isn't working, try `*`, and vice versa
15. `Measures` itself is a dimension (therefore `{ }` and `()` rules apply)
16. Guideline: Measures are generally specified on the column axis (but there are exceptions)
17. Guideline: Dimensions are generally specified in the `WHERE` clause (but there are exceptions)
18. If you do *not* specify a measure, MDX will use the first one it finds (e.g. `Unit Sales`)
19. A specific hierarchy can be used in only one of the following in a query: rows, columns, or `WHERE` clause (e.g. `[Time].[YQMD]` cannot exist in both the columns axis AND the `WHERE` clause at the same time in a query)
20. If you do not specify a hierarchy, MDX implicitly uses the first hierarchy it encounters (e.g. `[Product].[Electronics]` implicitly refers to `[Product].[ByCategory].[Electronics]`). Note: hierarchy order is not alphabetical order.
21. If you do not specify a hierarchy and the member exists at multiple levels, MDX implicitly uses the first hierarchy it encounters (e.g. `[Product].[Electronics]` implicitly refers to `Electronics` at the Family level, not at the Subcategory level)
22. If a specific hierarchy doesn't work, try making it implicit, or use another hierarchy
23. If an implicit hierarchy doesn't work, try different specific hierarchies
24. If you run into ambiguous hierarchies on the job (e.g. in Waremart 2005 there is `Electronics` at the Family level, and two `Electronics` in the Subcategory level, keys `&[[115]]` and `&[[77]]`):

- 24.1. Try to get the database data and hierarchies cleaned up
- 24.2. If this is not possible, document the quirk for everyone (MDX developers to end users)
- 24.3. Try to configure the end user interface to be unambiguous (e.g. pull-downs in SSRS)
25. Constraints specified in the WHERE clause and FILTER clause *will not* show up in the results (e.g. WHERE [Time], [2005] will display results for 2005 but you will not see 2005 in the data)
26. Guideline: Use WHERE to slice to a particular attribute. Use FILTER to compare measures to specific values (aka WHERE clause is generally used on dimensions, FILTER generally on measures)
27. Separate multiple conditions in WHERE clause with commas, use AND keyword in FILTER
28. FILTER results before you ORDER them to minimize unnecessary processing time
29. Use NON EMPTY for sparse results to minimize unnecessary processing time
30. Use readable names (e.g. do not use axis(0) or axis(1) unless absolutely necessary)
31. .CurrentMember is often implied
32. Use .Members to get members for a dimension, hierarchy, or level. .Members returns [All]
33. Use .Children to return what .Members returns, minus the [All]
34. Keywords that are hierarchical in nature generally use a hierarchy (e.g. [Product].[Tools].Parent is meaningful; [Product][Rt Prod Brand].Parent is not)
35. .Children keyword can be used both with dimensions and hierarchies (it won't break, but it's usually more meaningful with hierarchies)
36. Keywords that operate on only one level can use either hierarchies or dimensions (e.g. [Time].[YQMD].PrevMember and [Time].[Month].PrevMember will work)
37. Check to see if both 0 and NULL conditions are covered (e.g. with IIF. Note: 0 and NULL are not necessarily treated the same way)
38. The second parameter of the DESCENDANTS function requires a hierarchy
39. SELF_AND_BEFORE and SELF_AND_AFTER have inclusive start/end points
40. To transfer data to Excel, right click on the upper left blank cell
41. The combination of : and .Children will not work
42. Use WITH MEMBER to rename column names and create calculations. Note: SSRS can override headings as well
43. Use WITH SET to create named sets
44. Use ^J to invoke Intellisense
45. Calculated members are put on the Measures dimension by default
46. Double click on a member name in the result set of a query to bring up properties
47. Use named sets to make code easier to read, and reuse in multiple queries
48. Warning: Name sets in SQL Server 2005 are static; they are optionally dynamic or static in 2008
49. Main menu: Tools>Options>Text Editor>All Languages>General to display line numbers
50. You can jump out of the WHERE clause scope (e.g. WHERE [Time].[2005] but you reference 2004 data for accounting purposes)
51. ParallelPeriod and PeriodsToDate must reference the level or higher than what the query specifies (e.g. you can't go back a quarter and select a month slice of data)
52. ParallelPeriod is used alone, sliced to a measure
53. PeriodsToDate is used in conjunction with aggregations (e.g. SUM)
54. Beware of using LastPeriod and LastChild with respect to future effective dating
55. Use HAVING, not FILTER, to do text searches
56. .Siblings = .Parent.Children
57. If you see an MDX error that isn't covered by one of the tips in this document (other than basic syntax errors), send a code sample and the error message tailored to Walmart as you may have found an addition.

BIDS Environment

1. Within the Calculations tab, ensure the first entry in the Script Organizer is a script command with the CALCULATE keyword, otherwise the data in your cube won't show.