



Multi-Dimensional Expression MDX Basics

IT 4713 BI

Jack G. Zheng

Spring 2020

<http://zheng.kennesaw.edu/teaching/it4713/>

Overview



- Basic multidimensional query and MDX concepts
- Basic MDX query select statement elements
- Advanced
 - Named sets
 - Calculated members
- All examples in this lecture notes can be used with the AdventureWorks OLAP database (2017). Please execute the queries to see their effects.

MDX (Multi-Dimensional eXpressions)



- MDX is an industry standard query and calculation language used to retrieve data from OLAP databases.
- MDX is based on the XML for Analysis (XMLA) specification, with specific extensions for SQL Server Analysis Services.
- Where is it used in Microsoft BI?
 - In SQL Server Analysis Services, MDX is the query language used to retrieve data, but also supports data definition and data manipulation.
 - Some tools and applications, such as Excel, use MDX constructs internally to query an Analysis Services data source.
 - MDX editors are built into SQL Server Management Studio, Reporting Services, and SQL Server Data Tools. We can use the MDX editors to create ad hoc queries or reusable script if the data operation is repeatable.
 - We can also use MDX programmatically, by enclosing MDX statement in an XMLA Execute request.
- MDX queries and expressions in SQL Server Analysis Services are used to do the following:
 - Return data to a client application from a SQL Server Analysis Services cube.
 - Format query results.
 - Perform cube design tasks, including the definition of calculated members, named sets, scoped assignments, and key performance indicators (KPIs).
 - Perform administrative tasks, including dimension and cell security.
- <https://docs.microsoft.com/en-us/sql/analysis-services/multidimensional-models/mdx/querying-multidimensional-data-with-mdx?view=sql-server-2017>

MDX Example (AdventureWorks)



- MDX is similar in many ways to the SQL syntax that is typically used with relational databases.
- However, MDX is not an extension of the SQL language and is different from SQL in many ways.
- MDX utilizes expressions composed of identifiers, values, statements, functions, and operators that Analysis Services can evaluate to retrieve an object (for example a set or a member), or a scalar value (for example, a string or a number).
- Query example:

```
SELECT { [Measures].[Sales Amount], [Measures].[Tax Amount] }  
      ON COLUMNS,  
      { [Date].[Fiscal].[Fiscal Year].&[2012], [Date].[Fiscal].[Fiscal Year].&[2013] } ON  
ROWS  
FROM [Adventure Works]  
WHERE ( [Sales Territory].[Southwest] )
```

MDX Key Concepts



- MDX is based on the same concepts as multidimensional analysis and OLAP, including
 - Measure, dimension, dimension attribute, member, hierarchy
- Read more
 - <https://docs.microsoft.com/en-us/analysis-services/multidimensional-models/mdx/key-concepts-in-mdx-analysis-services?view=sql-server-2017>

MDX Language Elements



- Reference

- <https://docs.microsoft.com/en-us/sql/mdx/mdx-syntax-elements-mdx?view=sql-server-2017>

<u>Identifiers</u>	Identifiers are the names of objects such as cubes, dimensions, members, and measures.
Data Types	Define the types of data that are contained by cells, member properties, and cell properties.
<u>Expressions (MDX)</u>	Expressions are units of syntax that Analysis Services can resolve to single (scalar) values or objects. Expressions include functions that return a single value, a set expression, and so on.
<u>Operators</u>	Operators are syntax elements that work with one or more simple MDX expressions to make more complex MDX expressions.
<u>Functions</u>	Functions are syntax elements that take zero, one, or more input values, and return a scalar value or an object. Examples include the Sum function for adding several values, the Members function for returning a set of members from a dimension or level, and so on.
<u>Reserved Keywords</u>	Reserved keywords are words that are reserved for the use of MDX and should not be used for object names used in MDX statements.
<u>Members, Tuples, and Sets</u>	Members, tuples and sets are core concepts of multidimensional data that you must understand before you create an MDX query.

Expression



- An expression is a combination of identifiers, values, and operators that can be evaluated to get a result.
 - Reference: <https://docs.microsoft.com/en-us/sql/mdx/expressions-mdx?view=sql-server-2017>
- General member expression syntax

```
Dimension_Name .[Hierarchy_Name.]  
[[{Member_Name | &Member_Key}.]... ]  
{Member_Name | &Member_Key}
```

- Examples
 - [Measures].[Sales Amount]
 - [Date].[Calendar Year].&[2012]
 - [Date].[Calendar].[Calendar Quarter].&[2012]&[1]
 - [Employee].[Employees].&[112]
 - [Product].[Product Categories].[All Products]
 - [Date].[Calendar].[Calendar Year].&[2012]
 - [Date].[Calendar Year].[CY 2012]

Formatting Rules for Regular Identifiers



- The first character must be one of the following:
 - A letter as defined by the Unicode Standard 2.0. Besides letter characters from other languages, the Unicode definition of letters includes Latin characters from a through z and from A through Z.
 - The underscore (_).
- Subsequent characters can be:
 - Letters as defined in the Unicode Standard 2.0.
 - Decimal numbers from either Basic Latin or other national scripts.
 - The underscore (_).
- The identifier must not be an MDX reserved keyword. Reserved keywords are case-insensitive in MDX.
 - Referenec: <https://docs.microsoft.com/en-us/sql/mdx/reserved-keywords-mdx-syntax?view=sql-server-2017>
- Embedded spaces or special characters are not allowed.

Delimited Identifiers



- An identifier that does not comply with the formatting rules for regular identifiers must always be delimited by using brackets [].
- In the following hypothetical MDX statement, Sales Volume, Sales Cube, and select are delimited identifiers:
 - The [Sales Volume] and [Sales Cube] identifiers contain a space.
 - The [select] identifier is a reserved keyword.

```
SELECT Measures.[Sales Volume]  
FROM [Sales Cube]  
WHERE Product.[select]
```

Tuple



- A tuple uniquely identifies a slice of data from a cube. The tuple is formed by a combination of dimension members, as long as there are no two or more members that belong to the same hierarchy.
 - Reference: <https://docs.microsoft.com/en-us/sql/analysis-services/multidimensional-models/mdx/tuples?view=sql-server-2017>
 - Use ()

```
SELECT  
(  
    [Customer].[Country].&[United States],  
    [Date].[Calendar Year].&[2012]  
)  
ON COLUMNS,  
[Measures].[Internet Sales Amount] on ROWS  
FROM [Adventure Works];
```

Set



- A **set** consists of an ordered list of zero or more tuples. A set that does not contain any tuples is known as an empty set.
 - Use { }

```
SELECT [Measures].[Sales Amount] ON COLUMNS,  
       { [Date].[Fiscal].[Fiscal Year].&[2012], [Date].[Fiscal].[Fiscal Year].&[2013] }  
       ON ROWS  
FROM [Adventure Works]
```

- Use a range in a set “:”

```
SELECT [Measures].[Sales Amount] ON COLUMNS,  
       { [Date].[Fiscal].[Fiscal Year].&[2011]:[Date].[Fiscal].[Fiscal Year].&[2014] }  
       ON ROWS  
FROM [Adventure Works]
```

Set and Tuple



- The following two expressions produce the same results: in United States and in the year of 2012 and 2013.

```
(  
    [Customer].[Country].&[United States],  
    { [Date].[Calendar Year].&[2012], [Date].[Calendar Year].&[2013] }  
)
```

```
{  
    ( [Customer].[Country].&[United States], [Date].[Calendar Year].&[2012] ),  
    ( [Customer].[Country].&[United States], [Date].[Calendar Year].&[2013] )  
}
```

MDX Query and Script



- Script
 - In SQL Server Analysis Services, a Multidimensional Expressions (MDX) script is made up of one or more MDX expressions or statements that populate a cube with calculations.
 - An MDX script defines the calculation process for a cube. An MDX script is also considered part of the cube itself. Therefore, changing an MDX script associated with a cube immediately changes the calculation process for the cube.
- Query (SELECT statement)
 - Multidimensional Expressions (MDX) enables you to query multidimensional objects, such as cubes, and return multidimensional cell sets that contain the cube's data.
 - Using SELECT

Basic SELECT



- MDX SELECT statement uses the following clauses:
 - A SELECT clause that determines the **query axes** of an MDX SELECT statement.
 - A FROM clause that determines which **cube** will be queried.
 - An **optional** WHERE clause that determines which members or tuples to use on the **slicer axis** to restrict the data returned.

```
SELECT <query axis clause>  
FROM <cube clause>  
WHERE <slicer axis clause>
```

- Example

```
SELECT [Measures].[Internet Sales Amount] ON COLUMNS,  
[Date].[Calendar].[Calendar Year] ON ROWS  
FROM [Adventure Works]  
WHERE [Sales Territory].[Europe]
```

Query Axes



- Query axes specify the edges of a cell set returned by an MDX SELECT statement. Specifying the edges of a cell set lets you restrict the returned data that is visible to the client.
- Each query axis defines a set, or one tuple (one tuple set)
- All query axes together defines a cube (or a matrix/pivot table visually)
- Reference
 - <https://docs.microsoft.com/en-us/sql/analysis-services/multidimensional-models/mdx/mdx-query-and-slicer-axes-specify-the-contents-of-a-query-axis?view=sql-server-2017>

Query Axis Examples



- A set in query axis

```
SELECT
    { [Measures].[Sales Amount],[Measures].[Tax Amount] } ON COLUMNS,
    { [Date].[Fiscal].[Fiscal Year].&[2012],[Date].[Fiscal].[Fiscal Year].&[2013] }
      ON ROWS
FROM [Adventure Works]
```

- Rows and columns can also be an integer

```
SELECT
    { [Measures].[Sales Amount],[Measures].[Tax Amount] } ON 0,
    { [Date].[Fiscal].[Fiscal Year].&[2012],[Date].[Fiscal].[Fiscal Year].&[2013] } ON 1
FROM [Adventure Works]
WHERE ( [Sales Territory].[Southwest] )
```

0 means columns;
1 means rows.

Query Axis Filtering



- NON EMPTY is used to remove any result contains null value

```
SELECT [Measures].[Internet Sales Amount] ON COLUMNS,  
NON EMPTY [Date].[Calendar].[Calendar Year] ON ROWS  
FROM [Adventure Works]  
WHERE [Sales Territory].[Europe]
```

- HAVING is used to filter contents based on criteria

```
SELECT {[Measures].[Internet Sales Amount]} ON COLUMNS,  
NON EMPTY  
{[Date].[Calendar].[Date].MEMBERS}  
HAVING [Measures].[Internet Sales Amount]>15000 ON ROWS  
FROM [Adventure Works]
```

Multiple Dimensions in One Axis



- Organize multiple dimensions to a tuple in one axis

```
SELECT
    ([Sales Territory].[Sales Territory].[Group], [Sales Channel].[Sales Channel].[Sales
Channel]) ON COLUMNS,
    ([Date].[Calendar Year].[Calendar Year], [Product].[Category].[Category] ) ON ROWS
FROM [Adventure Works]
WHERE [Measures].[Sales Amount]
```

Two dimensions stacked on the row and on the column.

- Pivoting (switch a dimension between rows and columns)
 - To organize how it is display in a table
 - We can also change their display order

```
SELECT
([Product].[Category].[Category] , [Sales Territory].[Sales Territory].[Group])
    ON COLUMNS,
([Sales Channel].[Sales Channel].[Sales Channel], [Date].[Calendar Year].[Calendar Year])
    ON ROWS
FROM [Adventure Works]
WHERE [Measures].[Sales Amount]
```

Compare to the query above:
which dimension is pivoted?

Slicer Axis



- The slicer axis (using WHERE) filters the data returned by the Multidimensional Expressions (MDX) SELECT statement, restricting the returned data so that only data intersecting with the specified members will be returned.
- It can be thought of as an invisible extra axis in a query.

```
SELECT {[Measures].[Internet Sales Amount]} ON COLUMNS,  
[Date].[Calendar Year].MEMBERS ON ROWS  
FROM [Adventure Works]  
WHERE([Customer].[Customer Geography].[Country].[United States])
```

- The slicer axis cannot use the same hierarchy/dimension that is already used as query axis.
- Reference
 - <https://docs.microsoft.com/en-us/sql/analysis-services/multidimensional-models/mdx/mdx-query-and-slicer-axes-specify-the-contents-of-a-slicer-axis?view=sql-server-2017>

Filtering Results in Slicer or Query Axis?



- Query axis is used to explicitly set and display filtered results – results can be seen as row and column labels.
- Slicer axis is used to implicitly set query restriction – it is not displayed in results.

```
SELECT
    [Date].[Fiscal].[Fiscal Year] ON 0,
    [Measures].[Sales Amount] ON 1
FROM [Adventure Works]
```

Vs.

```
SELECT
    [Date].[Fiscal].[Fiscal Year] ON 0
FROM [Adventure Works]
WHERE [Measures].[Sales Amount]
```

```
SELECT
    [Measures].[Sales Amount] ON 0,
    [Date].[Fiscal].[Fiscal Year].&[2012]
ON 1
FROM [Adventure Works]
```

Vs.

```
SELECT
    [Measures].[Sales Amount] ON 0
FROM [Adventure Works]
WHERE
    [Date].[Fiscal].[Fiscal Year].&[2012]
```

Named Set



- Named set is a set expression to which an alias has been assigned.
- A named set can incorporate any members or functions that can ordinarily be incorporated into a set.
- Because MDX treats the named set alias as a set expression, you can use that alias anywhere a set expression is accepted.
- You can define a named set to have one of the following contexts:
 - **Query-scoped** To create a named set that is defined as part of an MDX query, and therefore whose scope is limited to the query, you use the WITH keyword.
 - **Session-scoped** To create a named set whose scope is wider than the context of the query, that is, whose scope is the lifetime of the MDX session, you use the CREATE SET statement. A named set defined by using the CREATE SET statement is available to all MDX queries in that session.
- Reference
 - <https://docs.microsoft.com/en-us/sql/analysis-services/multidimensional-models/mdx/mdx-named-sets-building-named-sets?view=sql-server-2017>

Query-Scoped Named Set



Use the WITH clause at the beginning of the query to define query scoped named set.

```
WITH SET [EU] AS
{
    [Sales Territory].[Sales Territory Country].&[France],
    [Sales Territory].[Sales Territory Country].&[Germany]
}
SELECT [Measures].[Internet Sales Amount] ON 0,
EU ON 1
FROM [Adventure Works]
```

	Internet Sales Amount
France	\$2,644,017.71
Germany	\$2,894,312.34

Calculated Member



- A calculated member is a member that is resolved by calculating an MDX expression to return a value.
 - <https://docs.microsoft.com/en-us/sql/analysis-services/multidimensional-models/mdx/mdx-calculated-members-building-calculated-members?view=sql-server-2017>
- You can define a calculated member to have one of the following contexts:
 - **Query-scoped** To create a calculated member that is defined as part of an MDX query, and therefore whose scope is limited to the query, you use the WITH keyword. You can then use the calculated member within an MDX SELECT statement.
 - **Session-scoped** To create a calculated member whose scope is wider than the context of the query, that is, whose scope is the lifetime of the MDX session, you use the CREATE MEMBER statement. A calculated member defined by using the CREATE MEMBER statement is available to all MDX queries in that session.

Query-based Calculated Member



- Simple multiplication factor

Use the WITH clause at the beginning of the query to define query scoped calculated member.

```
WITH MEMBER [Measures].[Special Discount] AS  
    [Measures].[Discount Amount] * 1.5  
SELECT  
    [Measures].[Special Discount] on COLUMNS,  
    NON EMPTY [Product].[Product].MEMBERS ON ROWS  
FROM [Adventure Works]
```

- Average sales amount per order

```
WITH MEMBER [Measures].[Average Sales] AS  
    [Measures].[Sales Amount] / [Measures].[Order Count]  
SELECT  
    [Measures].[Average Sales] on COLUMNS,  
    NON EMPTY [Sales Territory].[Sales Territory Country].Members  
ON ROWS  
FROM [Adventure Works]
```


Ordering Results



- ORDER function
 - Used to order the results
 - Reference: <https://docs.microsoft.com/en-us/sql/mdx/order-mdx?view=sql-server-2017>

```
SELECT [Measures].[Tax Amount] ON 0,  
ORDER([Sales Territory].[Sales Territory].[Group].MEMBERS,  
      [Measures].[Tax Amount], DESC) ON 1  
FROM [Adventure Works]
```

- More advanced order examples
 - <http://www.mssqltips.com/sqlservertip/3129/order-and-sort-with-mdx-in-sql-server-analysis-services/>

Ranking Results



- TOPCOUNT function
 - Used to return the top number of results only
 - Reference: <https://docs.microsoft.com/en-us/sql/mdx/topcount-mdx?view=sql-server-2017>

```
SELECT [Measures].[Order Quantity] ON 0,  
TOPCOUNT([Date].[Calendar].[Calendar Year],3, [Measures].[Order Quantity])  
    ON 1  
FROM [Adventure Works];
```

- Can also use ORDER + HEAD functions
 - <http://www.ssas-info.com/analysis-services-faq/27-mdx/793-how-in-mdx-query-can-i-get-top-3-sales-years-based-on-order-quantity>

MDX Learning Resources



- Key concepts
 - <https://docs.microsoft.com/en-us/sql/analysis-services/multidimensional-models/mdx/key-concepts-in-mdx-analysis-services?view=sql-server-2017>
- MDX Query Fundamentals
 - <https://docs.microsoft.com/en-us/sql/analysis-services/multidimensional-models/mdx/mdx-query-fundamentals-analysis-services?view=sql-server-2017>
- MDX references
 - <https://docs.microsoft.com/en-us/sql/mdx/multidimensional-expressions-mdx-reference>
 - <https://docs.microsoft.com/en-us/sql/mdx/mdx-syntax-elements-mdx>
 - <https://docs.microsoft.com/en-us/sql/mdx/mdx-language-reference-mdx>