Hong Kong Baptist University

Department of Computer Science

COMP 7810/4096 Business Intelligence (2019-20)

Multi-Dimensional eXpressions (MDX)

Introduction

Multi-Dimensional eXpressions (MDX) is a language that expresses selections, calculations, and some metadata definitions against an Online Analytical Processing (OLAP) database, and provides some capabilities for specifying how query results are to be represented. While SQL is used for relational queries, MDX is used for multidimensional queries. You have learnt how to create a cube with dimensions (e.g. Time, Product, Customer) and measures (e.g. Internet sales amount). In this lab, you will learn how to use MDX queries to extract useful data from the cube.

MDX requires multidimensional thinking. In MDX, the SELECT statement specifies a result set that contains a subset of multidimensional data that has been returned from a cube. To specify a result set, an MDX query must contain the following information:

- The *number of axes* that you want the result set to contain. You can specify up to 128 axes in an MDX query.
- The *set of members* or tuples to include *on each axis* of the MDX query.
- The *name of the cube*.
- The *set of members* or tuples to include *on the slicer axis*.

The syntax of MDX is:

```
SELECT axis specification ON COLUMNS,
axis specification ON ROWS
FROM cube_name
WHERE slicer_specification (optional)
```

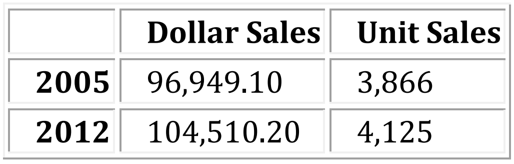
Here is an example:

```
SELECT
{ [Measures].[Dollar Sales], [Measures].[Unit Sales] }
on columns,
{ [Time].[CalendarYear].[2005], [Time].[ CalendarYear].[2012] }
on rows
FROM [Sales]
WHERE [Customer].[MA]
```

In this example:

- **SELECT** keyword starts the clause that specifies what you want to retrieve.
- The **ON** keyword is used with an axis name to specify where dimensions from your database are displayed. The two measures *Dollar sales* and *Unit sales* are put on the columns axis and *time periods* are put on the row axis.
- MDX uses curly braces, {and}, to enclose a set of elements from a particular dimension or set of dimensions.
- The **FROM** clause in an MDX query names the cube from which the data is being queried.
- The **WHERE** clause provides a place to specify members for other cube dimensions that don't appear in the columns or rows (or other axes).

The result grid of this example is:



	Dollar Sales	**Unit Sales**
2005	96,949.10	3,866
2012	104,510.20	4,125

Learning Outcomes

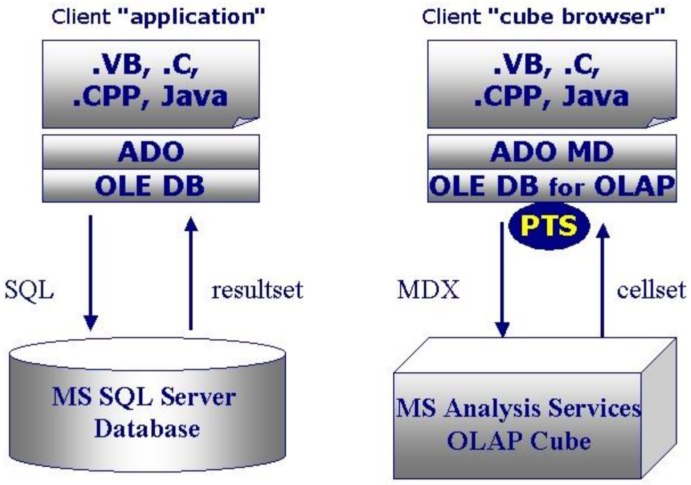
This tutorial is based on the cube you built (`Adventure Works Cube`) in Lab 2B.

By finishing this lab session, you should be able to

- Understand MDX Concepts.
- Learn MDX Syntax.
- Learn some MDX functions.

Tools

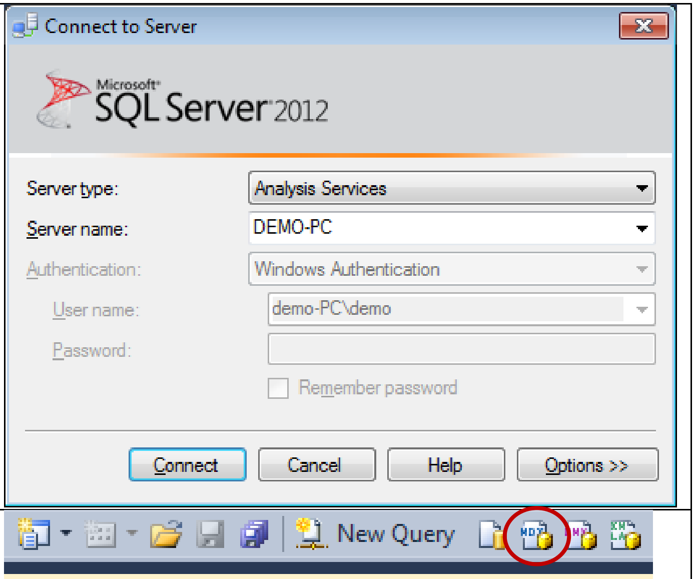
- Microsoft SQL Server Management Studio 2012
- Visual Studio 2010 with SQL Server Data Tools (SSDT)



Part A: Multi-Dimensional eXpressions (MDX)

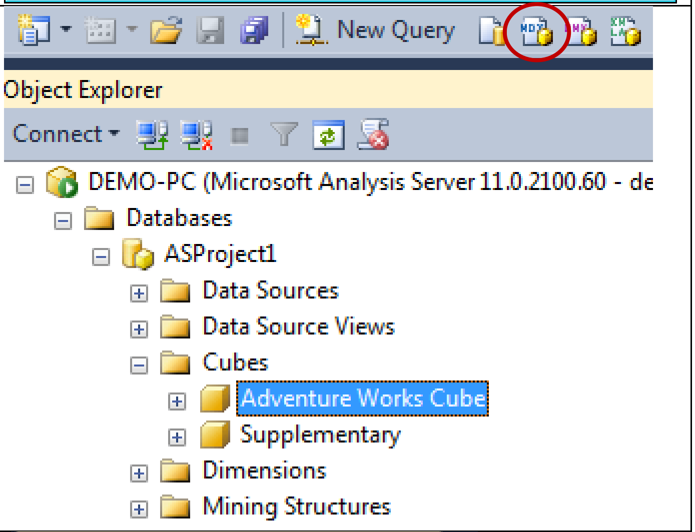
I. Connect to the localhost and make MDX query

1. Open **SQL Server Management Studio**, change the Server type to **Analysis Services.** Press **Connect** button.

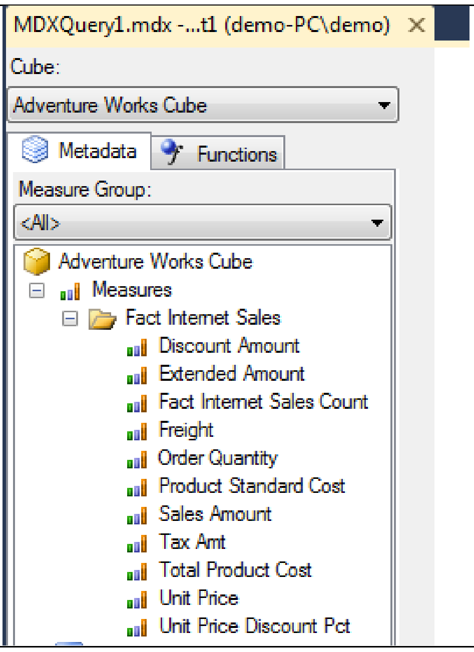


Server type: Analysis Services
Server name: DEMO-PC
Authentication: Windows Authentication
User name: demo-PC\demo
Password:
□ Remember password

2. Select the **Adventure Works Cube**. Click the **MDX** button to write the first MDX query. Press **Connect** again.



3. You may see Measures and Dimensions under Metadata tab.



Cube: Adventure Works Cube
Measure Group: <All>

II. Putting members onto columns or rows, use of [], {} and comma(,)

1. To calculate total sales amount, type the expression as shown here and press Execute button.

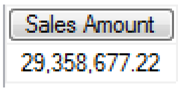
Note:

- It is the simplest form of axis specification involves taking a particular measure e.g. sales amount.
- MDX is **neither case-sensitive nor line-oriented**

```
select Measures.[Sales Amount] on columns
from [Adventure Works Cube]
```

or

```
SELECT MEASURES.[SALES AMOUNT]
ON COLUMNS
FROM [ADVENTURE WORKS CUBE]
```



Sales Amount
29,358,677.22

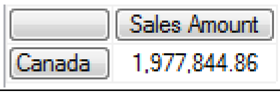
- To calculate sales amount in *Canada* only, change the expression to the one as shown here.

Note: You can use axis numbers to specify the axis in the query. 0 for columns, 1 for rows

```
select Measures.[Sales Amount]
on columns,
[Dim Customer].[English Country Region
Name].Canada
on rows
from [Adventure Works Cube]
```

or

```
select Measures.[Sales Amount]
on 0,
[Dim Customer].[English Country Region
Name].Canada
on 1
from [Adventure Works Cube]
```



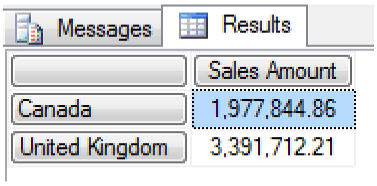
	Sales Amount
Canada	1,977,844.86

2. To calculate sales amount in *Canada* and *United Kingdom*, change the expression as shown here and press Execute button.

Note:

- The square brackets are optional, except for identifiers with embedded spaces, you can remove the **[]** for Canada but not for United Kingdom.
- The axis definition is enclosed in braces **{}**, which are used to denote sets (more than one single member of a dimension).
- The two members Canada and United Kingdom are separated by comma **(,)**
- In the result set, Canada row comes first.

```
select Measures.[Sales Amount] on 0,
{[Dim Customer].[English Country Region
Name]. [Canada],
[Dim Customer].[English Country Region
Name].[United Kingdom]}
on 1
from [Adventure Works Cube]
```



	Sales Amount
Canada	1,977,844.86
United Kingdom	3,391,712.21

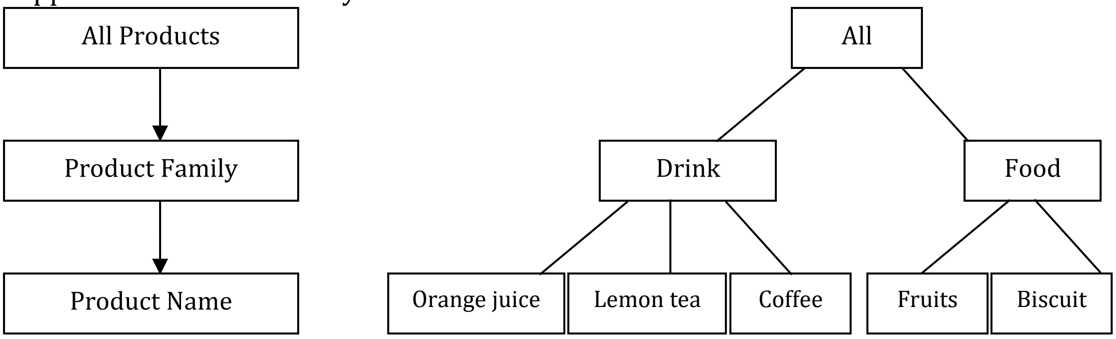
3. To calculate sales amount and order quantity in *Canada* and *United Kingdom.* Note: Must use **braces {}** to enclose two different measures.	select { Measures.[Sales Amount], Measures.[Order Quantity] } on 0, {[Dim Customer].[English Country Region Name].[Canada], [Dim Customer].[English Country Region Name].[United Kingdom]} on 1 from [Adventure Works Cube]

III. The use of .Members, .Children and NON EMPTY

Getting the set of members for a dimension, hierarchy or level is very common for retrievals. The .Members operator takes a dimension, hierarchy or level on its left-hand side, and returns a set of all members associated with that metadata scope.

Another kind of selection that is very frequent is to get the children of a member. We may want to use this to implement drill down operation.

Suppose there is a hierarchy in Product dimension:



Returns a set of members:

- [Product Family].[Product Name].**MEMBERS** = { Orange juice, Lemon tea, Coffee, Fruits , Biscuit }
- [Product Family].[Food].**CHILDREN** = { Fruits , Biscuit }

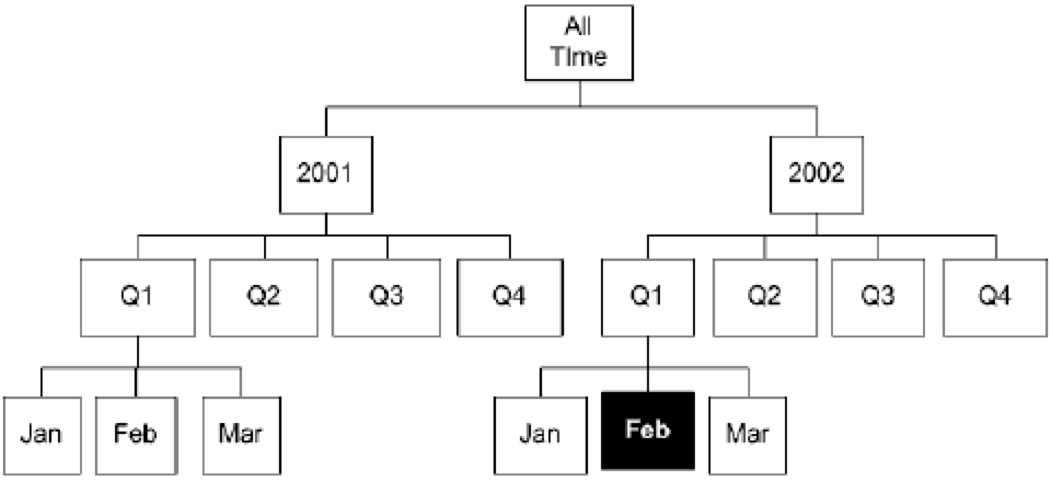


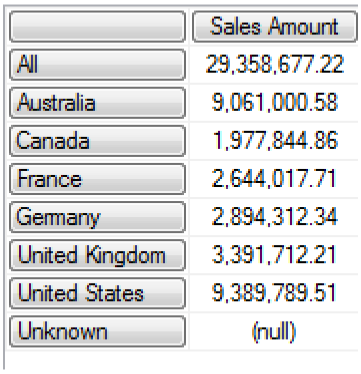
Figure 1. An example hierarchy of the time dimension.

- [2002].[Month].[Feb] is illegal as omitting quarter parent
- [Year].[2002].[Q2].[Feb] is illegal

1. To take all MEMBERS of a dimension (to display sales amount in all countries), you can use **`.Members`**

   Note:
   - In running this expression, a row member named "**All**" will be generated by default.
   - There is an **UNKNOWN** row with null value in the query result.

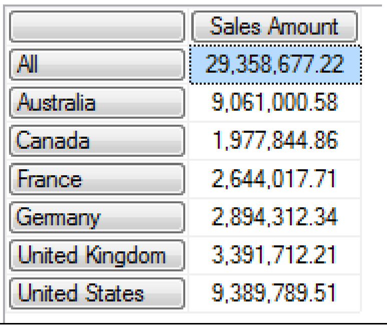
```
select Measures.[Sales Amount] on 0,
[Dim Customer].[English Country Region
Name].Members on 1
from [Adventure Works Cube]
```



	Sales Amount
All	29,358,677.22
Australia	9,061,000.58
Canada	1,977,844.86
France	2,644,017.71
Germany	2,894,312.34
United Kingdom	3,391,712.21
United States	9,389,789.51
Unknown	(null)

2. To remove empty slice from query result, add **NON EMPTY** expression to the query.

```
select Measures.[Sales Amount] on 0,
non empty [Dim Customer].[English
Country Region Name].Members on 1
from [Adventure Works Cube]
```

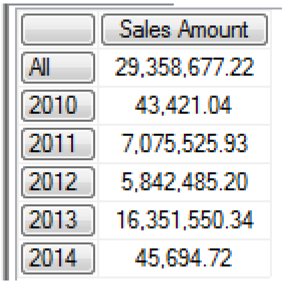


	Sales Amount
All	29,358,677.22
Australia	9,061,000.58
Canada	1,977,844.86
France	2,644,017.71
Germany	2,894,312.34
United Kingdom	3,391,712.21
United States	9,389,789.51

3. To calculate sales amount in every year, you can use the expression as shown here.

   (How to remove the grand total?)

```
select Measures.[Sales Amount] on 0,
NON EMPTY [Dim Date].[Calendar
Year].Members on 1
from [Adventure Works Cube]
```

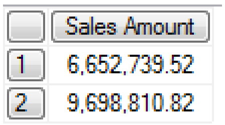


	Sales Amount
All	29,358,677.22
2010	43,421.04
2011	7,075,525.93
2012	5,842,485.20
2013	16,351,550.34
2014	45,694.72

4. *Hierarchy Nagvigation:*
   `.Children` function returns the child members for a particular member within the dimension. The expression here shows the sales amount in the two semesters in 2013.

   (How to select child members of Australia? How to select the parent member for Tasmania?)

```
select Measures.[Sales Amount] on 0,
[Dim Date].[2013].Children on 1
from [Adventure Works Cube]
```



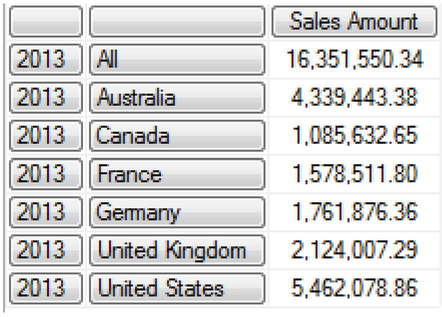
	Sales Amount
1	6,652,739.52
2	9,698,810.82

Note: **[]** is used if it starts with number

5. To calculate sales amount in all countries in 2013, you can use **()** [but not {}] to enclose a set of members (tuples) from different dimensions.

   Note: **Remember to use (), not {},** otherwise, you will get error.

```
select Measures.[Sales Amount] on 0,
non empty(
[Dim Date].[Calendar Year].[2013],
[Dim Customer].[English Country Region
Name].Members
)
on rows
from [Adventure Works Cube]
```



		Sales Amount
2013	All	16,351,550.34
2013	Australia	4,339,443.38
2013	Canada	1,085,632.65
2013	France	1,578,511.80
2013	Germany	1,761,876.36
2013	United Kingdom	2,124,007.29
2013	United States	5,462,078.86

IV. Slicer specification using WHERE clause

The **WHERE** clause specifies rule for limiting the results of the query to a subspace of the data. The process of limiting the results is called **slicing**.

You can define the **slicer specification** with the **WHERE** clause, outlining the slice of the cube to be viewed. Usually, the WHERE clause is used to define the **measure** that is being queried. Because the cube's measures are just another dimension, selecting the desired measure is achieved by selecting the appropriate slice of the cube.

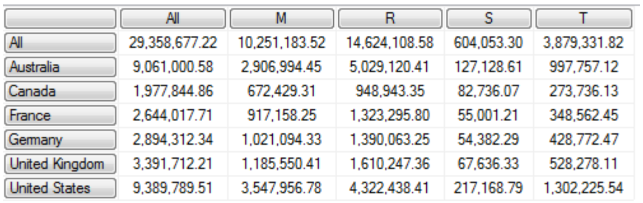
1. To calculate the sales amount *by country in 2013 only*.

   Note: The result is as same as the previous example but it *does not show the calendar year* before the country names.

```
select Measures.[Sales Amount] on 0,
NON EMPTY
[Dim Customer].[English Country Region
Name].Members on 1
from [Adventure Works Cube]
where
[Dim Date].[Calendar Year].[2013]
```

2. You may put the measure in WHERE clause to calculate the sales amount *by country & by product line*, you can use expression as show here.

```
select
NON EMPTY
[Dim Product].[Product Line].Members
on 0,
NON EMPTY
[Dim Customer].[English Country Region
Name].Members
on 1
from [Adventure Works Cube]

where Measures.[Sales Amount]
```



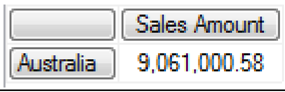
	All	M	R	S	T
All	29,358,677.22	10,251,183.52	14,624,108.58	604,053.30	3,879,331.82
Australia	9,061,000.58	2,906,994.45	5,029,120.41	127,128.61	997,757.12
Canada	1,977,844.86	672,429.31	948,943.35	82,736.07	273,736.13
France	2,644,017.71	917,158.25	1,323,295.80	55,001.21	348,562.45
Germany	2,894,312.34	1,021,094.33	1,390,063.25	54,382.29	428,772.47
United Kingdom	3,391,712.21	1,185,550.41	1,610,247.36	67,636.33	528,278.11
United States	9,389,789.51	3,547,956.78	4,322,438.41	217,168.79	1,302,225.54

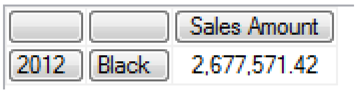
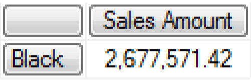
V. Tuples

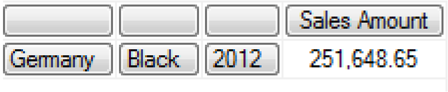
Analysis Services presents cubes as n-dimensional spaces (cube space), within the cube space, data are accessible through cells, each identified by a tuple.

Points within the cube space can be referenced using a tuple. A tuple is represented by one member from each dimension, separated by a comma, and is enclosed within parentheses.

In MDX, a tuple is an ordered collection of one or more members from different dimensions.

1. One dimensional (1D) tuple.	`select Measures.[Sales Amount] on 0,`
	`[Dim Customer].[English Country Region Name].[Australia] on 1`
	`from [Adventure Works Cube]`
	
	Sales Amount
	Australia 9,061,000.58

2. Two dimensional (2D) tuple: is an intersection of 2 members from 2 dimensions
 	
Use ( ), not curly braces { }
	`Option 1: use ()`
	`select Measures.[Sales Amount] on 0,`
	`(`
	` [Dim Date].[Calendar Year].[2012] ,`
	` [Dim Product].[Color].[Black]`
	`)`
	`on 1`
	`from [Adventure Works Cube]`
	
	Sales Amount
	2012 Black 2,677,571.42
	
	`Option 2: use where clause`
	`select Measures.[Sales Amount] on 0,`
	`[Dim Product].[Color].[Black] on 1`
	`from [Adventure Works Cube]`
	
	`where [Dim Date].[Calendar Year].[2012]`
	
	Sales Amount
	Black 2,677,571.42

3. Three Dimensional (3D) tuple: 3 members from 3 dimensions	`Option 1: use ()`
	`select Measures.[Sales Amount] on 0,`
	`( [Dim Customer].[English Country Region Name].[Germany] ,`
	`[Dim Product].[Color].[Black],`
	`[Dim Date].[Calendar Year].[2012]`
	`) on 1`
	`from [Adventure Works Cube]`
	
	Sales Amount
	Germany Black 2012 251,648.65
	
	`Option 2: Use where clause:`
	`select Measures.[Sales Amount] on 0,`
	`( [Dim Customer].[English Country Region Name].[Germany] ,`
	`[Dim Product].[Color].[Black]`
	`) on 1`
	`from [Adventure Works Cube]`
	`where [Dim Date].[Calendar Year].[2012]`

VI. Sets

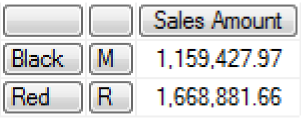
A Set is an ordered list of members, or tuples that are defined using the exact same set of dimensions. We can create a **collection of tuples (1D, 2D, 3D), use curly brace {}**

1. To return a collection of 2D tuples.

   Use combination of ( ) and { }

   These two tuples have the same dimensionality because they both have two members

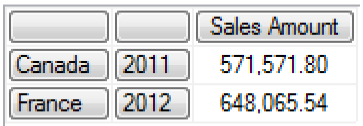
```
select Measures.[Sales Amount] on 0,
{
([Dim Product].[Color].[Black] ,
[Dim Product].[Product Line].[M])
 ,
([Dim Product].[Color].[Red] ,
[Dim Product].[Product Line].[R])
}
on 1

from [Adventure Works Cube]

where [Dim Date].[Calendar Year].[2012]
```



		Sales Amount
Black	M	1,159,427.97
Red	R	1,668,881.66

2. Short exercise!!

```
To show sales amount for Canada in
2011, and sales amount for France in
2012.
Remove ALL and NULL values
```



		Sales Amount
Canada	2011	571,571.80
France	2012	648,065.54

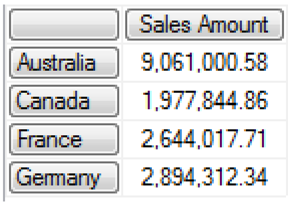
VII. Another method to return sets - use of colon (:)

The comma operator (,) is used to separate a list of members or to construct **sets - a collection of tuples**. How about colon (:) operator?

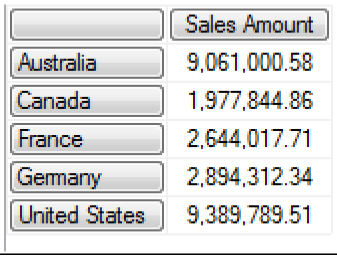
At every level in every dimension, the members of that level are arranged in a particular order. You can specify a set as a range of members in that order by listing two members from the same level as endpoints and putting **colon (:)** between them.

1. To calculate sales amount for the first four members in the same level.

```
select Measures.[Sales Amount] on 0,

{[Dim Customer].[English Country Region
Name].[Australia]:
[Dim Customer].[English Country Region
Name].[Germany]} on 1

from [Adventure Works Cube]
```



	Sales Amount
Australia	9,061,000.58
Canada	1,977,844.86
France	2,644,017.71
Germany	2,894,312.34

2. You can use comma and colon at the same time to specify a set of members together with another single member.	`select Measures.[Sales Amount] on 0,`
	
	`{[Dim Customer].[English Country Region Name].[Australia]:`
	`[Dim Customer].[English Country Region Name].[Germany],`
	`[Dim Customer].[English Country Region Name].[United States]} on 1`
	
	`from [Adventure Works Cube]`
	
	
	Sales Amount
	Australia 9,061,000.58
	Canada 1,977,844.86
	France 2,644,017.71
	Germany 2,894,312.34
	United States 9,389,789.51

VIII. Comments in MDX

To facilitate documentation, three variations of comment syntax are allowed in MDX.

- /* and */
- -- (A pair of dashes)
- //

1. Try the expression.	`select`
	`Measures.[Sales Amount] on 0,`
	`//NON EMPTY`
	`{[Dim Customer].[English Country Region Name].Members } on 1`
	`from [Adventure Works Cube]`

2. Try the expression.	`select`
	`Measures.[Sales Amount] on 0,`
	`--NON EMPTY`
	`{[Dim Customer].[English Country Region Name].Members } on 1`
	`from [Adventure Works Cube]`

3. Try the expression.	`select`
	`Measures.[Sales Amount] on 0 /*,`
	`NON EMPTY`
	`{[Dim Customer].[English Country Region Name].Members } on 1`
	`*/`
	`from [Adventure Works Cube]`

IX. Calculated members

Calculated member allows you to define formula and treat the formula as a new member of a specified parent.

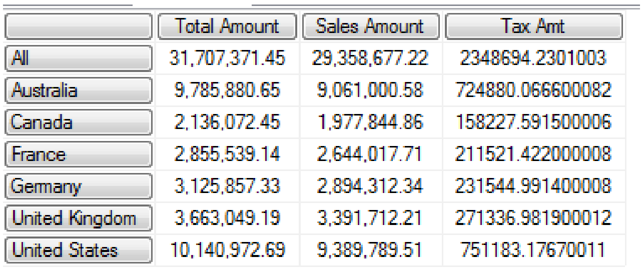
If a calculated member is only required for a MDX query, you can define that calculated member by using the **WITH** keyword in front of the SELECT statement. The syntax is:

```
WITH MEMBER parent.name AS 'expression'
```

FORMAT_STRING is used to change the display format of the new calculated member.

1. To calculate the total amount [sales amount + tax amt] for all countries.

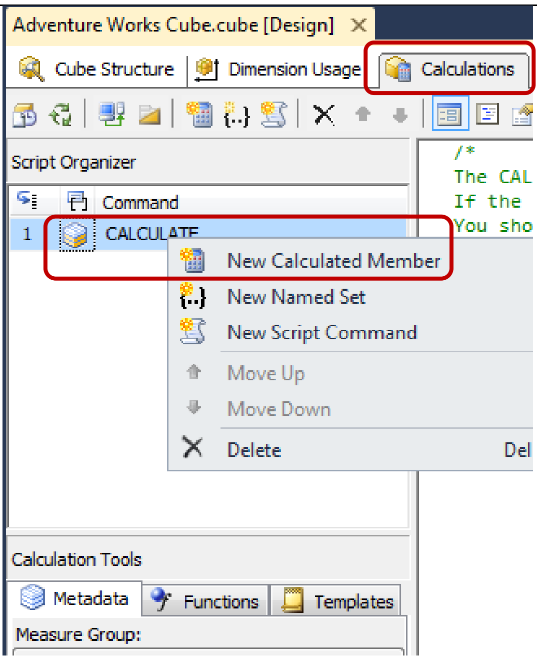
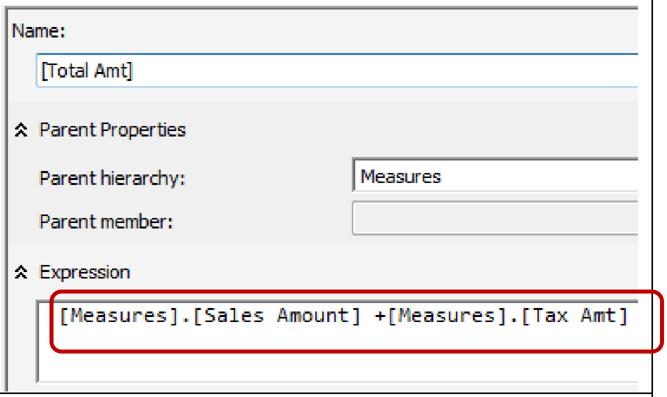
```
WITH
   MEMBER [Measures].[Total Amount] AS
   [Measures].[Sales Amount] +
  [Measures].[Tax Amt]

select
   {[Measures].[Total Amount],
    [Measures].[Sales Amount],
    [Measures].[Tax Amt]}
on 0,
NON EMPTY
[Dim Customer].[English Country Region
Name].Members on 1

from [Adventure Works Cube]
```



	Total Amount	Sales Amount	Tax Amt
All	31,707,371.45	29,358,677.22	2348694.2301003
Australia	9,785,880.65	9,061,000.58	724880.066600082
Canada	2,136,072.45	1,977,844.86	158227.591500006
France	2,855,539.14	2,644,017.71	211521.422000008
Germany	3,125,857.33	2,894,312.34	231544.991400008
United Kingdom	3,663,049.19	3,391,712.21	271336.981900012
United States	10,140,972.69	9,389,789.51	751183.17670011

2. Change the display format for total amount to *currency with 2 d.p.*

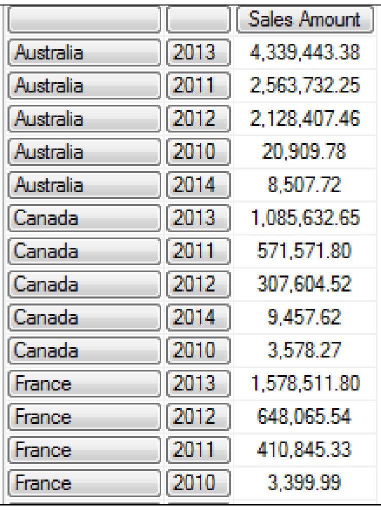
```
WITH
   MEMBER [Measures].[Total Amount] AS
   [Measures].[Sales Amount] +
  [Measures].[Tax Amt],
   FORMAT_STRING = 'currency'

select
   {[Measures].[Total Amount],
    [Measures].[Sales Amount],
    [Measures].[Tax Amt]}
on 0,
NON EMPTY
[Dim Customer].[English Country Region
Name].Members on 1

from [Adventure Works Cube]
```

3. You may open **ASProject1** in the Visual Studio with SSDT, **right click** the **Adventure Works Cube.cube** you have built in lab 2, and choose **Browse.** 4. Select **Calcuations** tabs, right click **CALCULATE** and choose **New Calculated Member** 5. Type **[Total Amt]** as Name, type **[Measures].[Sales Amount] +[Measures].[Tax Amt]** as the Expression. 6. Select **File → Save All** to save the change.	 Adventure Works Cube.cube [Design] Cube Structure Dimension Usage Calculations Script Organizer Command 1 CALCULATE New Calculated Member New Named Set New Script Command Move Up Move Down Delete Del Calculation Tools Metadata Functions Templates Measure Group: /* The CAL If the You sho  Name: [Total Amt] Parent Properties Parent hierarchy: Measures Parent member: Expression [Measures].[Sales Amount] +[Measures].[Tax Amt]
7. **Deploy** the cube again and calculate the total amount for all countries. (using new calculated member [Total Amt] directly)	(Think about the code yourself)

X. ORDER function with DESC

The **Order** function first arranges the members **according to their position in the hierarchy**, and then orders each level.

1. Sort the sales amount by country from largest to smaller.	select [Measures].[Sales Amount] on 0, non empty order([Dim Customer].[English Country Region Name].children, [Measures].[Sales Amount], DESC)on 1 from [Adventure Works Cube]
2. Sort the sales amount by country by calendar year from largest to smaller.	select [Measures].[Sales Amount] on 0, non empty order(([Dim Customer].[English Country Region Name].children ,[Dim Date].[Calendar Year].children) , [Measures].[Sales Amount], DESC)on 1 from [Adventure Works Cube]



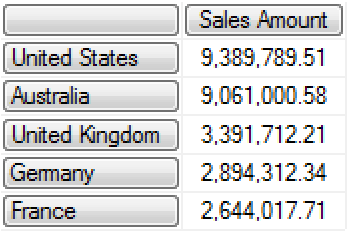
		Sales Amount
Australia	2013	4,339,443.38
Australia	2011	2,563,732.25
Australia	2012	2,128,407.46
Australia	2010	20,909.78
Australia	2014	8,507.72
Canada	2013	1,085,632.65
Canada	2011	571,571.80
Canada	2012	307,604.52
Canada	2014	9,457.62
Canada	2010	3,578.27
France	2013	1,578,511.80
France	2012	648,065.54
France	2011	410,845.33
France	2010	3,399.99

XI. TOP Performance Analysis

When displaying information such as the best-selling cities based on sales amount, it may be beneficial to limit the query to, say, the top dozen. MDX can support this operation using functions like HEAD or TOPCOUNT.

1. To find the top five sales countries.

```
select Measures.[Sales Amount] on 0,

HEAD (ORDER ([Dim Customer].[English
Country Region Name].children,
Measures.[Sales Amount], desc),5)
on 1

from [Adventure Works Cube]
```



	Sales Amount
United States	9,389,789.51
Australia	9,061,000.58
United Kingdom	3,391,712.21
Germany	2,894,312.34
France	2,644,017.71

2. To find the top five sales countries.

```
select Measures.[Sales Amount] on 0,

TOPCOUNT
([Dim Customer].[English Country Region
Name].children
, 5, [Measures].[Sales Amount]) on 1

from [Adventure Works Cube]
```

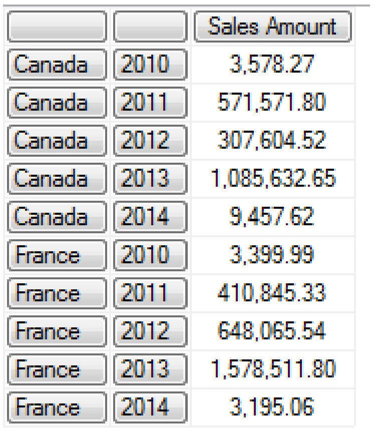
XII. Crossjoin function and filtering

In many cases, you will want to take the cross-product of members (or tuples) in two different sets (that is, specify all of their possible combinations). The CrossJoin() function is the most direct way of combining the two sets in this way. The syntax is:

```
CrossJoin (set1, set2)
```

1. To show the sales amount for both *Canada* and *France* in different calendar years

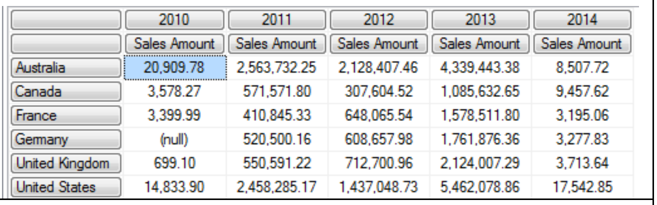
```
select [Measures].[Sales Amount] on 0,

Non empty( Crossjoin (
{[Dim Customer].[English Country Region
Name].[Canada]
,[Dim Customer].[English Country Region
Name].[France]},
[Dim Date].[Calendar Year].children))
on 1

from [Adventure Works Cube]
```



		Sales Amount
Canada	2010	3,578.27
Canada	2011	571,571.80
Canada	2012	307,604.52
Canada	2013	1,085,632.65
Canada	2014	9,457.62
France	2010	3,399.99
France	2011	410,845.33
France	2012	648,065.54
France	2013	1,578,511.80
France	2014	3,195.06

2. To show the sales amount for all countries in different calendar years

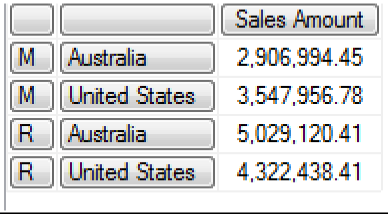
```
select NON EMPTY
[Dim Date].[Calendar Year].children
*
[Measures].[Sales Amount]
on 0 ,

NON EMPTY
[Dim Customer].[English Country Region
Name].children on 1

from [Adventure Works Cube]
```



	2010	2011	2012	2013	2014
	Sales Amount	Sales Amount	Sales Amount	Sales Amount	Sales Amount
Australia	20,909.78	2,563,732.25	2,128,407.46	4,339,443.38	8,507.72
Canada	3,578.27	571,571.80	307,604.52	1,085,632.65	9,457.62
France	3,399.99	410,845.33	648,065.54	1,578,511.80	3,195.06
Germany	(null)	520,500.16	608,657.98	1,761,876.36	3,277.83
United Kingdom	699.10	550,591.22	712,700.96	2,124,007.29	3,713.64
United States	14,833.90	2,458,285.17	1,437,048.73	5,462,078.86	17,542.85

3. To show sales amount for each product line in each country, and returns only the products that have the sales amount greater than 2,000,000.

```
select  Measures.[Sales Amount] on 0,

Filter (CrossJoin (
[Dim Product].[Product Line].children,
[Dim Customer].[English Country Region
Name]. children) ,
Measures.[Sales Amount] >2000000) on 1

from [Adventure Works Cube]
```



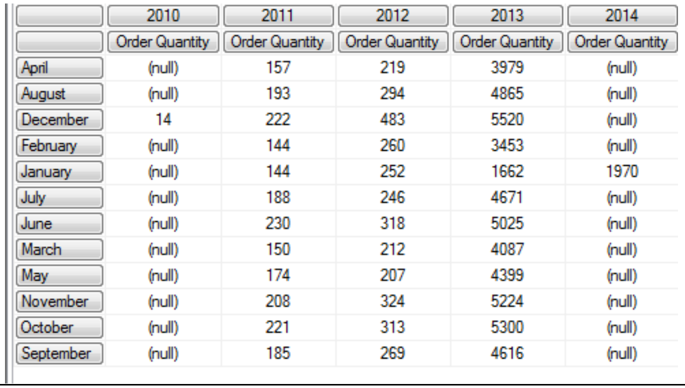
		Sales Amount
M	Australia	2,906,994.45
M	United States	3,547,956.78
R	Australia	5,029,120.41
R	United States	4,322,438.41

XIII. PrevMember, NextMember

PrevMember returns the previous member of a level for a supplied member. **NextMember** returns the next member of a level for a supplied member.

1. To calculate the order quantity for each month in each year.

```
select [Dim Date].[English Month
Name].children on 1,
NON EMPTY
([Dim Date].[Calendar Year].children,
measures.[order quantity]) on 0
from [Adventure Works Cube]
```



	2010	2011	2012	2013	2014
	Order Quantity	Order Quantity	Order Quantity	Order Quantity	Order Quantity
April	(null)	157	219	3979	(null)
August	(null)	193	294	4865	(null)
December	14	222	483	5520	(null)
February	(null)	144	260	3453	(null)
January	(null)	144	252	1662	1970
July	(null)	188	246	4671	(null)
June	(null)	230	318	5025	(null)
March	(null)	150	212	4087	(null)
May	(null)	174	207	4399	(null)
November	(null)	208	324	5224	(null)
October	(null)	221	313	5300	(null)
September	(null)	185	269	4616	(null)

2. To compare order quantity for each month in 2013 and 2012, and calculate the difference. (using **prevMember**)

```
WITH
MEMBER [Order quantity 2012] AS
    ([Measures].[Order quantity], [Dim
Date].[Calendar Year].[2013].prevMember)

MEMBER [Order quantity 2013] AS
    ([Measures].[Order quantity], [Dim
Date].[Calendar Year].[2013])

MEMBER [Difference] AS ([Order quantity
2013]- [Order quantity 2012])

select { [Order quantity 2012], [Order
quantity 2013],[Difference] } on 0,
{[Dim Date].[English Month Name].children}
on 1

from [Adventure Works Cube]
```

3. To compare order quantity for each month in 2013 and 2012, and calculate the difference. (using **nextMember**)

```
(Think about the codes yourself)
```

XIV. Exercise 1

Write MDX expression for the following queries and save the MDX codes in a MS Word file (just copy & paste the expression). Rename the file to **lab3A-ans.docx.** (You need to **remove empty slice** in all questions) [Please refer to your lab2B answers (A to H) to see if the results are the same]

A. Total order quantity for each country
B. Total order quantity by country and product line
C. Total order quantity for each product color
D. Total sales amount by country and product line for married customers only
E. Total sales amount by country by product line in 2014 only
F. Total sales amount by country in 2012 Quarter 2 only
G. Compare sales amount quarter-by-quarter in each year
H. Compare sales amount for Q1 and Q2 in 2012
I. Compare order quantities for Semester 1 and 2 in 2012, and calculate the difference using **nextMember** or **prevMember** function.
J. Find the top three sales year
K. Find the bottom three sales countries

XV. Answer Submission

1. Submit the Word file **lab3A-ans.docx** to the site **http://buelearning.hkbu.edu.hk/**