MICROSOFT
TRAINING
AND CERTIFICATION

Microsoft Official
Curriculum

Module 5: Creating Simple MDX Queries

Contents

Microsoft

Instructor Notes

Presentation:
60 Minutes

Lab:
15 Minutes

In this module, students will learn how to create simple multidimensional expression (MDX) query statements that generate a report. Students will learn the terminology that is used to create a query, and then they will be introduced to an interface that can be used to process a query and create a report. Finally, students will learn about different types of simple query statements and how to create them.

At the end of this module, students will be able to:

- Understand MDX query statement terminology.
- Write simple MDX query statements by using the MDX2093A application.

Materials and Preparation

This section provides you with the required materials and preparation tasks that are needed to teach this module.

Required Materials

To teach this module, you need the following materials:

- Microsoft® PowerPoint® file 2093A_05ppt

Preparation Tasks

To prepare for this module, you should:

- Read all of the materials for this module.
- Read the instructor notes and margin notes.
- Practice the lecture and demonstration.
- Complete the labs.
- Review the Teacher Preparation materials on the Teacher Preparation compact disc.

Instructor Setup for Group Activities and Labs

This section provides setup instructions that are required to prepare the instructor computer or classroom configuration for group activities and labs.

All group activities and labs use the same database setup, which requires restoring a database archive.

► To prepare for group activities and labs

In this procedure, you restore the **Market** database, which is a .cab file type.

1. Start Analysis Manager.
2. In the left pane, expand the Analysis Services folder.
3. Expand the **Server** icon and verify that the **Market** database does not exist.
4. Right-click the **Server** icon, and then click **Restore Database**.
5. Navigate to the C:\Moc\2093A\Batches folder.
6. Select **Market.cab**, click **Open**, and then click **Restore**.

If the **Market** database already exists from a previous group exercise or lab, and cubes within the database contain extraneous information, you can return the **Market** database and its cubes to a beginning position by either:

- Deleting any calculated members that were created in a specific cube, and then saving the cube.

  - or -

- Repeating the preceding restore database procedure.

Other Activities

Difficult Questions

Following are difficult questions that students might ask you during the delivery of this module and answers to the questions. These materials delve into subjects that are within the scope of the module but are not specifically addressed in the content of the student notes.

1. Sometimes the WHERE clause is called a Filter *axis*. If it is an axis, why can't you put a set on it?

   The WHERE clause is sometimes called an axis, but that is technically incorrect. The technical term for the Filter clause is *slicer specification*.

2. What is the difference between a slicer, a filter, and the WHERE clause?

   They can all be terms for the same thing. The tuple from the WHERE clause is called a *slicer specification* in the Analysis Services documentation, but the same thing is called *Filter area* in the Microsoft Office PivotTable® list control, and *Page area* in a Microsoft Excel PivotTable report.

3. If I will never write a custom client application, why should I create MDX statements?

   You might never use MDX to create a report, but many complex calculated members require aggregations over multiple values. To create those calculations, you must be familiar with sets. Creating MDX query statements is the best way to learn how to work with sets.

4. Do all the tuples in a set have to come from the same dimension?

   If each tuple contains only one member, then all the members must come from the same dimension. If the tuples in a set contain two members, then all the tuples must have two members, with the first member of each tuple coming from one dimension and the second member of each tuple coming from a different dimension.

5. When would you create a report using more than two axes?

   For a printed, tabular report, you will invariably use only two axes—row and column. Most client applications, however, will allow you to place multiple nested dimensions on either the Row or Column axis. In a charting application, you might have three or more axes represented.

6. When would you use the range operator in a set?

   MDX has a large number of functions that return sets—many of which allow you to manipulate other sets. Those functions, which will be introduced in Module 6, are simpler and less likely to create an error than the range operator. You might want to create a set by using the range operator when testing a particular set of members.

7. A query with only one axis must have a Columns axis. What if you want only one axis, but want it to appear on as rows?

   The terms *Columns* and *Rows* are simply aliases for the underlying axis names—Axis(0) and Axis(1). A client application can display the Columns axis on the rows of a report.

Module Strategy

Major sections of this module begin with or include a group activity followed by a review lecture and then a lab. The following are guidelines for delivering materials in the context of group activities:

- Using group activities to introduce new content

  You often introduce new concepts or functionality while delivering the procedures in a group activity as a live demonstration. For example, you can present a new MDX function by showing first its construction and then its result set as an actual calculated member formula or in a query statement.

  Use the topic slides that follow the group activity as a review of the content—for example, the syntax of a specific function.

- Interaction with students

  A group activity flows best when you deliver it as a shared exploration. Ask students such questions as: "What would happen if we…?" "Why did this happen?" "Was that what you expected?" Encourage students to ask you questions about the results being tested.

- Students follow along

  In some cases, you might want to encourage students to follow your live demonstration on their own computers. This practice works best for simpler group activities or for a group activity that is not replicated by a later lab.

  It is not a problem if a student does not follow your demonstration, or if a student starts following and then stops before the group activity is completed. There is no file or structure dependency between group activities or between a group activity and a later lab.

- Lab replication of group activity

  The exercises in the labs closely follow the group activity procedures but do not define each step or show the code answer. Encourage students to write and test the MDX expressions on their own, referring back to the group activity procedures for clarification. Students can also refer to answer files that are available for each procedure within exercises.

  Labs are generally more challenging when students have not followed the instructor on their own computers during the group activity. However, many students benefit from the two hands-on experiences of following the group activity and completing the labs.

- Answer files for group activities

  Where applicable, answer files are provided for each procedure in a group activity. If necessary to facilitate your demonstration, copy and paste the correct expression from the answer file into the Calculated Member Builder.

Use the following additional strategies to present this module:

- Understanding MDX Query Statements

  Begin with an explanation of why query statements are needed—particularly if the end goal is to create calculated members: Complex calculated members require aggregating the values from multiple members—a set. A query statement is the best way to learn how to use sets. Use the Office PivotTable report to introduce concepts that will later appear in the MDX syntax. Help students understand that the PivotTable report creates MDX statements behind the scenes and that they will create similar statements themselves.

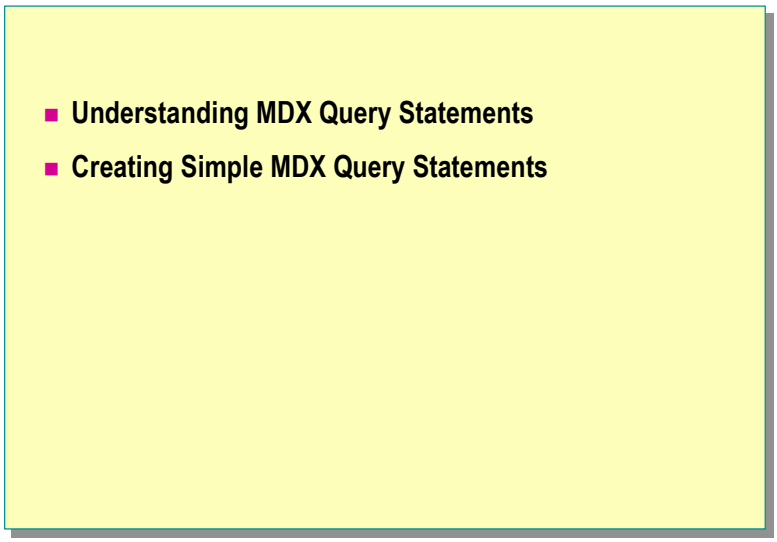
- Creating Simple MDX Query Statements

  Start with a description of a very simple query, consisting only of the keywords SELECT and FROM. Next, describe the use of a WHERE clause. Explain how to display multiple values in a query by putting sets on axes, and then explain how to retrieve all the members from a level by using the **Level Members** function. Describe the rules for managing axes and how axes are named. Give an overview of sets, and then finish with a description of how sets can be used to return a specific range of members.

Overview

Topic Objective
To provide an overview of the module topics and objectives.

Lead-in
In this module, you will learn how to create query statements by using MDX.



In this module, you will learn how to create simple multidimensional expression (MDX) query statements that generate a report. You will learn the terminology that is used to create a query, and then explore an interface that can be used to process a query and create a report. Finally, you will learn about different types of simple query statements and how to create them.

At the end of this module, you will be able to:

- Understand MDX query statement terminology.
- Write simple MDX query statements by using the MDX2093A application.

◆ Understanding MDX Query Statements

Topic Objective
To introduce the concept of an MDX query statement.

Lead-in
This section introduces the basic terminology and concepts of an MDX query statement.

- **MDX Expressions vs. MDX Query Statements**
- **Values: Discrete and Aggregated**
- **The Function of an MDX Query Statement**
- **Group Activity: Using MDX Terms in a Report**
- **MDX Terms**

Delivery Tip
Briefly explain the bullets on this diamond slide to provide context for the upcoming group activity where the items are functionally demonstrated.

This section introduces the basic terminology and concepts of an MDX query statement.

The following topics are included in this section:

- MDX Expressions versus MDX Query Statements

  This topic reviews the difference between MDX expressions and MDX query statements.

- Values: Discrete and Aggregated

  This topic introduces why it is important to learn how to create an MDX query statement, even if your ultimate purpose is to create calculated members.

- The Function of an MDX Query Statement

  This topic explains that the function of an MDX query is to extract values from an online analytical processing (OLAP) cube and put them into a cell set, or report.

- Group Activity: Using MDX Terms in a Report

  The group activity uses the Microsoft® Office PivotTable® list browser control to introduce MDX terminology.

- MDX Terms

  This topic reviews the important terms used in an MDX query statement.

MDX Expressions vs. MDX Query Statements

Topic Objective
To review the differences between MDX expressions and MDX queries.

Lead-in
To review, MDX has two major functions.

- **MDX Expression**
  - Multidimensional formula
  - Similar to a spreadsheet formula
- **MDX Query Statement**
  - Query language for browsers
  - Similar to SQL query

Delivery Tip
This slide is for review and reorientation. You have already covered MDX expressions in preceding modules, and now you are moving on to MDX query statements. Do not spend much time on this slide.

MDX has two major functions: creating expressions and creating query statements.

MDX Expression

Creating expressions is one of the major tasks for MDX:

- An MDX expression is a multidimensional formula.
- MDX expressions are essentially analogous to formulas in a spreadsheet.

You use MDX expressions to create calculated members.

MDX Query Statement

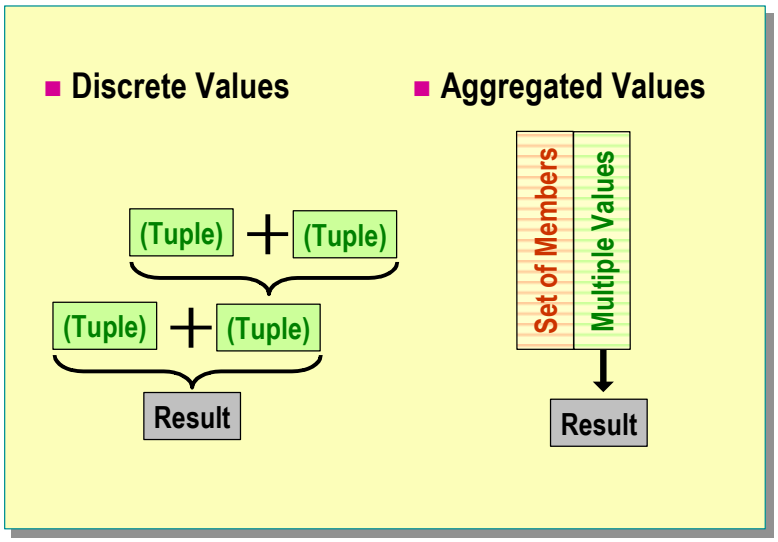
The second major task for MDX is to create query statements:

- A query statement is what a browser uses to generate a report based on an OLAP cube.
- MDX queries are analogous to Structured Query Language (SQL) query statements.

Values: Discrete and Aggregated

Topic Objective
To show the two types of values that can go into an MDX expression.

Lead-in
You know how to create MDX expressions that use discrete value



Delivery Tip
Carefully explain the difference between discrete and aggregated values because this distinction is important for helping students understand why they are learning MDX query statements.

By learning about calculated members, you know how to create MDX expressions that use discrete values. There is an additional type of MDX expression that requires multiple values.

Discrete Values

A discrete value is a single value retrieved from a cube. To retrieve a discrete value from a cube, you use a complete tuple. You can create many useful MDX expressions—for example, percent of total and growth calculations—simply by using discrete values combined with arithmetic operators.

Aggregated Values

The single value needed by a calculated member can come from a large number of values aggregated together. For example, to calculate a year-to-date value, you must add the values for all the months leading up to the current month. Rather than specify each month individually, you specify which set of months you want to retrieve the values from.

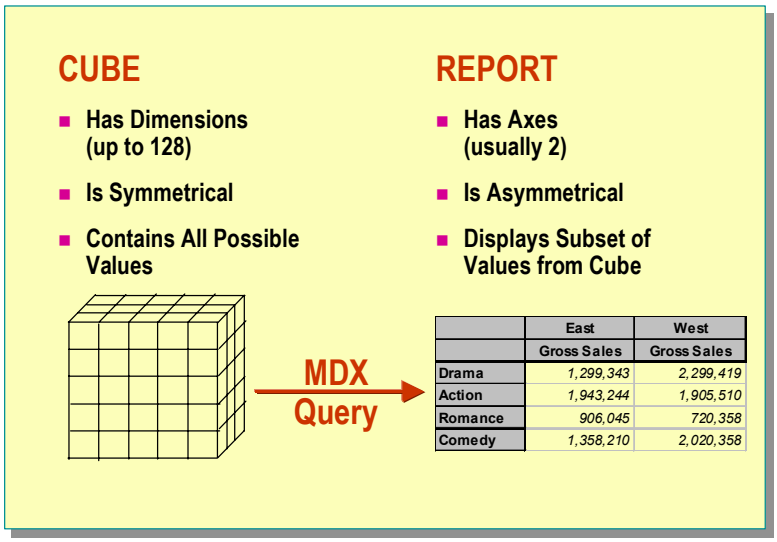
Calculated members that use aggregated values are very useful. To create them effectively, you must first understand how to create *sets* of members. The best way to learn how to work with sets of members is to create MDX query statements.

Even if you never intend to create a custom application or to use an MDX query statement, learning how to create them will be extremely useful as you create sophisticated calculated members that require dynamically aggregated values.

The Function of an MDX Query Statement

Topic Objective
To show why MDX queries are used and to introduce some MDX terminology.

Lead-in
The purpose of an MDX query is to extract values from an online analytical processing (OLAP) cube and put them into a report.



Delivery Tip
Build Slide: The slide initially shows only the two graphics. As you click the mouse, the Cube bullets appear, and then the Report bullets appear, and finally the MDX arrow appears.

The purpose of an MDX query is to extract values from an online analytical processing (OLAP) cube and put them into a cell set, or report. Cubes and reports are different kinds of structures.

Cube

A cube is the source of data for a report. It has the following characteristics:

- A cube has dimensions—up to 128.
- A cube is symmetrical. Every member of every dimension intersects with every member of every other dimension.
- A cube contains all possible values for all members of all levels of all dimensions.

Report

A report is the output that the user sees. It is based on data from the cube. It has the following characteristics:

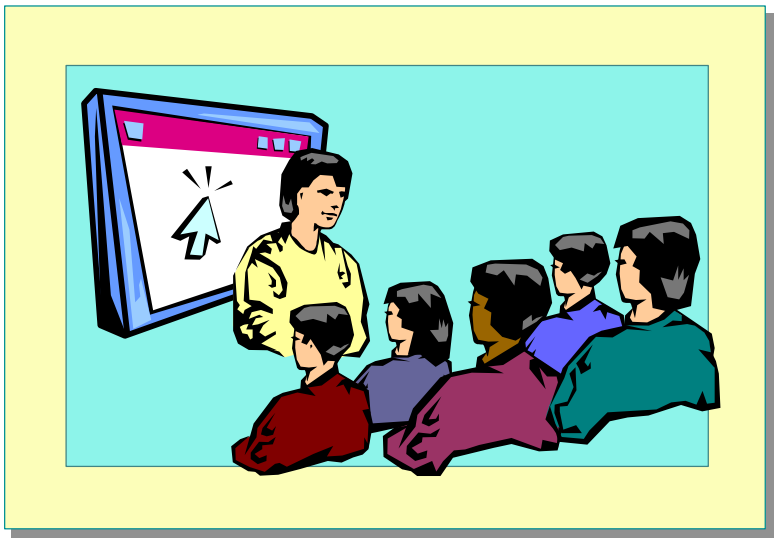
- A report does not have dimensions; it has axes—typically, a Rows axis and a Columns axis. An axis can include members from more than one dimension.
- A report can be asymmetrical. That is, a report contains only selected values from selected levels of selected dimensions from the cube. For example, in a report, you can display all the states under USA, but none of the states under Canada or Mexico.
- A report can have a slicer specification. A report typically displays only a small fraction of the cells in a cube.

An MDX query statement consists of the instructions for extracting a report from a cube.

Group Activity: Using MDX Terms in a Report

Topic Objective
To demonstrate how to use MDX in a report.

Lead-in
In this group activity, you will learn MDX terminology in the context of using a browser.



In this group activity, you can follow along on your own computer with your instructor or observe the demonstration.

You will learn how to use MDX terminology in the context of a report in the Office PivotTable list browser control. Browsers such as the Office PivotTable list use MDX query statements to populate a report.

The terminology you will learn includes:

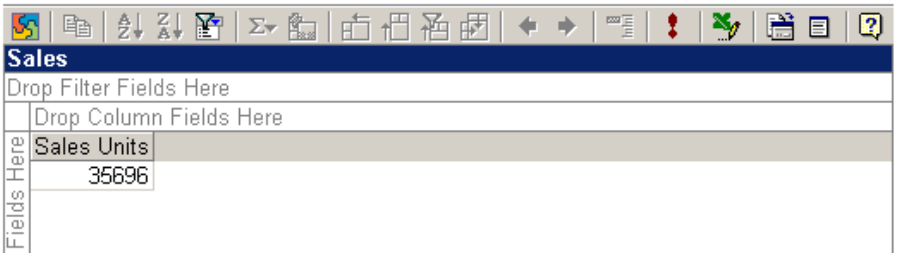
- Default members of a dimension
- The Filter clause tuple overriding default members
- Sets on an axis
- Selecting individual members for a set
- Selecting all the members of a level or dimension
- Putting multiple dimensions on one axis

► To show a single cell with a default tuple

In this procedure, you will learn that a single cell retrieved from a cube always uses a complete tuple and how the Filter clause overrides the default member to create the tuple.

The **Explore.htm** file contains an Office PivotTable list control that points at the **Sales** cube of the **Market** database.

1. In Microsoft Windows® Explorer, navigate to the C:\Moc\MDX2093A\LabFiles\L05 folder.
2. Double-click the **Explore.htm** file.



Ask what member is used for each dimension—the default member.

The initial report in the **Explore.htm** file shows only a single cell that shows the total value for the **Sales Units** measure. This grand total is a single value from the cube. To retrieve this value, Microsoft SQL Server™ 2000 Analysis Services requires a tuple.

► To show a single cell with a slicer specification tuple

In this procedure, you will learn how the slicer specification overrides the default member to create the tuple.

1. Click the **Field List** toolbar button. Drag the **Product**, **State**, **Employee**, and **Time Calendar** dimensions from the **Field List** window to the slicer specification of the report. The slicer area is identified by the words **Drop Filter Fields Here**.

   Each dimension displays the default member in the slicer specification, although the total value never changes as you move the dimension.

Note Because the **Time Fiscal** and **Time Calendar** dimensions are hierarchies of the same dimension, you can include only one in the report at a time. However, the report still uses the default member of the unused hierarchy.

Ask how useful a report with a single member would be.

Ask students what they might want on the report—for example, more than one member from a dimension on the rows and columns axes of a report.

The default member allows a query to ignore some of the dimensions in a cube, because if a dimension does not appear in the slicer specification, it always uses the default member.

If a dimension does appear in the slicer specification, it can use either the default member or a different member. Putting a dimension in the slicer specification of a report overrides the default member for that dimension.

2. Click the arrow next to the **Product** dimension list, select **Meat**, and then click **OK.** Now only the total for meat shows.

Sales			
Employee	Product	State	Time Fiscal
Sheri Nowmer	Meat	All State	All Time
	Drop Column Fields Here		
Fields Here	Sales Units		
	6156		

The current report has dimension members only in the slicer specification, which creates a report displaying only a single value.

Note According to Microsoft SQL Server 2000 Books Online, the term for the tuple that overrides the default tuple is *slicer specification*. In the Microsoft Office PivotTable list, the term for the slicer specification is *Filter area*. In the Microsoft Excel PivotTable report, the term for the slicer specification is *Page field*. The terms s*licer*, *Filter*, and *Page* are essentially synonyms.

► To put sets on an axis

Tell students that more than one member from a single dimension is called a *set*.

In this procedure, you will learn how multiple members go into a set on an axis.

1. Drag the **State** dimension to the Rows axis in the area identified by the words **Drop Row Fields Here**.

Sales			
Employee	Product	Time Fiscal	
Sheri Nowmer	Meat	All Time	
		Drop Column Fields Here	
Country	Region State	Sales Units	
⊞ Canada		785	
⊞ Mexico		1470	
⊞ USA		3901	
Grand Total		6156	

The three countries—plus the total—appear. The Rows axis now displays a set that contains four *positions*. Each position in the set corresponds to a member from the **State** dimension.

2. Click the **Country** level label, and click the **Expand** toolbar button.

Sales			
Employee ▾	Product ▾	Time Fiscal ▾	
Sheri Nowmer	Meat	All Time	
		Drop Column Fields Here	
Country ▾	Region	State	Sales Units
⊟ Canada	⊞ Canada		785
	Total		785
⊟ Mexico	⊞ Mexico		1470
	Total		1470
⊟ USA	⊞ North West		2795
	⊞ South West		1106
	Total		3901
Grand Total			6156

Ask how many positions are now in the set on the Rows axis—eight.

The regions appear. Even though the labels are split into two columns—to show the levels of the dimension—each position in the set still contains only a single member from the **State** dimension.

Notice that the **State** and **Region** columns show *all* the members from the respective levels. In other words, you did not put these members on the axis by adding individual members; you added all the members of the level at one time.

► To add individual members of the Measures dimension

In this procedure, you put members from the **Measures** dimension on the Columns axis by adding them one at a time.

1. Drag the **Sales Dollars** measure to the Data area, which is in the center of the report where the Sales Units column resides. You must drag the measure icon to an existing measure data section. A bold line appears where you should place the measure.
2. Drag the **Cost Dollars** measure to the Data area.

Sales					
Employee ▾	Product ▾	Time Fiscal ▾			
Sheri Nowmer	Meat	All Time			
		Drop Column Fields Here			
Country ▾	Region	State	Sales Units	Sales Dollars	Cost Dollars
⊟ Canada	⊞ Canada		785	1678.57	943.07
	Total		785	1678.57	943.07
⊟ Mexico	⊞ Mexico		1470	3281.11	1851.37
	Total		1470	3281.11	1851.37
⊟ USA	⊞ North West		2795	6233.27	3514.64
	⊞ South West		1106	2414.89	1375.71
	Total		3901	8648.16	4890.35
Grand Total			6156	13607.84	7684.79

Ask how many positions are in the set—three. Ask which dimension each member came from—the Measures dimension.

The Columns axis displays a set.

You individually select members from the **Measures** dimension. You add or remove them one at a time. This is different from the way that you typically add members to a non-measures dimension such as State. You can, however, remove individual members from a set on an axis.

► To remove individual members of a non-measures dimension

In this procedure, you remove members of the **State** dimension from the Rows axis by deselecting them one at a time.

1. On the **Country** button, click the drop-down arrow on the **Country** buttonand then clear the **Canada** check box.
2. Expand USA, and then clear the **South West** check box.

Ask students what the tuple is that retrieves the value 6233.27—the first cell in the Sales Dollars column. Product = Meat, Employee = Sheri Nowmer, Time Calendar = All Time, Region = North West.

3. Click **OK**.

Sales					
Employee ▼	Product ▼	Time Fiscal ▼			
Sheri Nowmer	Meat	All Time			
			Drop Column Fields Here		
Country ▼	Region	State	Sales Units	Sales Dollars	Cost Dollars
⊟ Mexico	⊞ Mexico		1470	3281.11	1851.37
	Total		1470	3281.11	1851.37
⊟ USA	⊞ North West		2795	6233.27	3514.64
	Total		2795	6233.27	3514.64
Grand Total			4265	9514.38	5366.01

4. To restore the members you removed, click the drop-down arrow on the **Country** button. Select the **Show All** check box, and click OK.

   With a non-measures dimension, you add all the members of the level, and then remove members one at a time by using the drop-down list. With the **Measures** dimension, you add members one at a time.

   In essence, you can create a set in any of the three following ways:

   - Display all the members from a level.
   - Include only explicit members—typically the **Measures** dimension.
   - Exclude explicit members from the members on a level.

► To put two dimensions on a single axis

In this procedure, you will learn how to create a set that contains two dimensions.

It is not unusual for a cube to contain at least six or seven dimensions. However, a report based on an OLAP cube typically contains only two axes—Columns and Rows. It is often useful to combine multiple dimensions from the cube on a single axis of the report.

1. In the PivotTable list control, drag the **Region** and **State** level buttons away from the Rows axis, leaving only the countries and **Grand Total** on the axis.
2. Drag the **Product** level from the slicer specification to the Rows axis, to the left of the **State** level. Drag the **Subcategory** and **Product** labels away from the report.

Sales				
Employee ▾	Time Fiscal ▾			
Sheri Nowmer	All Time			
		Drop Column Fields Here		
Country ▾	Category ▾	Sales Units	Sales Dollars	Cost Dollars
⊟ Canada	Bread	1046	2180.74	1243.02
	Dairy	1884	4014.91	2328.83
	Meat	785	1678.57	943.07
	Total	3715	7874.22	4514.92
⊟ Mexico	Bread	1289	2697.92	1558.79
	Dairy	2407	5159.38	2979.94
	Meat	1470	3281.11	1851.37
	Total	5166	11138.41	6390.1
⊟ USA	Bread	12197	25721.76	14710.97
	Dairy	10717	23358.73	13523.86
	Meat	3901	8648.16	4890.35
	Total	26815	57728.65	33125.18
Grand Total		35696	76741.28	44030.2

The report now shows both categories and countries on the Rows axis. The Rows axis still contains a single set—a set with 13 positions—but each position in the set now contains a two-member tuple—a coordinate from more than one dimension.

Technically, a set always contains tuples. Even when there is only one dimension on an axis, the set consists of single-member tuples.

► To put three dimensions on a single axis

In this procedure, you will learn how to create a set that contains three dimensions.

1. Drag the **Time Calendar** dimension from the slicer specification to the Rows axis, to the left of the Category label.
2. Drag the **Calendar Quarter** and **Calendar Month** labels away from the report, select the **Calendar Year** label, and then click the **Expand** toolbar button.

Ask how many sets are on the Rows axis—still just one.

Ask how many positions are in the set—23 positions.

Ask how many members are in the tuple in each position—3 members.

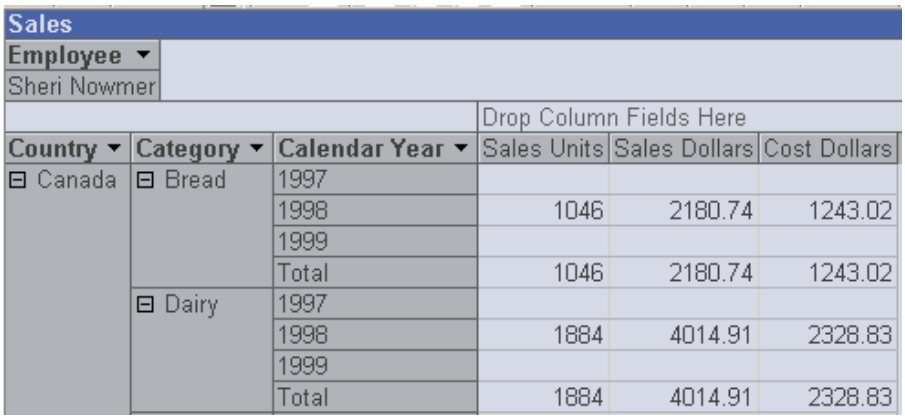
Sales					
Employee					
Sheri Nowmer					
			Drop Column Fields Here		
Country	Category	Calendar Year	Sales Units	Sales Dollars	Cost Dollars
⊟ Canada	⊟ Bread	1998	1046	2180.74	1243.02
		Total	1046	2180.74	1243.02
	⊟ Dairy	1998	1884	4014.91	2328.83
		Total	1884	4014.91	2328.83
	⊟ Meat	1998	785	1678.57	943.07
		Total	785	1678.57	943.07
	Total		3715	7874.22	4514.92
⊟ Mexico	⊟ Bread	1998	1289	2697.92	1558.79
		Total	1289	2697.92	1558.79
	⊟ Dairy	1998	2407	5159.38	2979.94
		Total	2407	5159.38	2979.94
	⊟ Meat	1998	1470	3281.11	1851.37
		Total	1470	3281.11	1851.37
	Total		5166	11138.41	6390.1
⊟ USA	⊟ Bread	1997	5559	11523.31	6596.6
		1998	6638	14198.45	8114.37
		Total	12197	25721.76	14710.97
	⊟ Dairy	1998	10717	23358.73	13523.86
		Total	10717	23358.73	13523.86
	⊟ Meat	1998	3901	8648.16	4890.35
		Total	3901	8648.16	4890.35
	Total		26815	57728.65	33125.18
Grand Total			35696	76741.28	44030.2

The report now shows countries, categories, and calendar years on the Rows axis.

► To show empty rows on an axis

In this procedure, you will learn how to control whether empty rows appear in the PivotTable list.

1. Click the caption bar of the PivotTable list, which has the label **Sales**.
2. In the PivotTable list toolbar, click the **Property Toolbox** button.
3. Expand the **Display Empty Items** section and select the **Row** check box.



Sales					
Employee ▾					
Sheri Nowmer					
			Drop Column Fields Here		
Country ▾	Category ▾	Calendar Year ▾	Sales Units	Sales Dollars	Cost Dollars
⊟ Canada	⊟ Bread	1997			
		1998	1046	2180.74	1243.02
		1999			
		Total	1046	2180.74	1243.02
	⊟ Dairy	1997			
		1998	1884	4014.91	2328.83
		1999			
		Total	1884	4014.91	2328.83

The blank rows for 1997 and 1999 appear. In an MDX query, you can choose whether to display empty rows. In the PivotTable list, the default is not to display them.

4. Close Internet Explorer.

Summary

Each position in a set consists of a tuple—typically a partial tuple. A partial tuple is a way to specify coordinates from more than one dimension on a single axis.

Again, contrast the slicer specification with the Rows and Columns axes:

- The Rows and Columns axes can contain a set—a set of tuples.
- The slicer specification can contain only a single tuple—the same as a single position in a set on one of the other axes.

For each value in the report, there is a single member from each dimension that contributed to the complete coordinate for that cell in the cube. The member for a dimension can come from any of the axes, and more than one dimension can be represented on a single axis. Any dimension not found on the Columns axis, Rows axis, or slicer specification uses the default member.

MDX Terms

Topic Objective
To review MDX terms.

Lead-in
Let's review the terms commonly used in MDX query statements.

- **Axis – The Row and Column Headings of a Report**
- **Set – Multiple Members on an Axis**
- **Slicer – The Filter Specification for a Report**
- **Tuple – One Member from Each Dimension Defined by Slicer Specification and Row/Column Sets**
- **Member – A Single Item from a Dimension**

Delivery Tip
Because students have already been exposed to the basic MDX terms in the previous group activity, treat this topic as a review—that is, do not spend a lot of time.

An MDX query statement shares some terms with a standard MDX expression and has some additional terms.

- Axis

  A report typically has two axes, one for rows and one for columns. The axes are the mechanism for *dicing* values in a report—as is frequently used in the term *slicing and dicing* to describe OLAP user interface elements and analysis methods.

- Set

  A set is one or more members of a dimension. A set appears on the Rows or Columns axis of a report.

- Slicer

  The slicer specification for a report appears above the report in a PivotTable list. The slicer specification contains only a single tuple—typically a partial tuple. This is the mechanism for *slicing* values in a report—as is frequently used in the term *slicing and dicing* to describe OLAP user interface elements and analysis methods.

- Tuple

  Each value retrieved from a cube must have a complete tuple. The slicer specification can contain a partial tuple that contributes to the complete tuple for a cell. The sets on the Rows and Columns axes contain partial tuples that also contribute to the complete tuple for each cell.

- Member

  A member is a single item from a dimension. A tuple consists of one member from one or more dimensions. A set consists of one or more tuples. A specific member for each dimension is necessary to retrieve a value from a cube.

◆ Creating Simple MDX Query Statements

Topic Objective
To introduce the subjects involved in learning to write MDX query statements.

Lead-in
In this section, you will learn basic MDX query syntax.

- **Group Activity: Using the MDX2093A Application**
- **Group Activity: Creating Simple MDX Query Statements**
- **Level Members Function**
- **Sets with Explicit Members**
- **Sets and Axes**
- **Simplest Possible Query**
- **WHERE Clause**
- **Query with Axes**

In this section, you will learn basic MDX query syntax.

The first group activity introduces you to the MDX2093A application, which is a customer interface for constructing and viewing MDX query statements.

The second group activity introduces you to basic MDX query statements—starting with a simple query and then advancing by using sets, members, and tuples to narrow the query results.

Delivery Tip
Briefly explain the bullets on this diamond slide to provide context for the upcoming group activity where the items are functionally demonstrated.

The following topics follow the group activities:

- Level Members Function

  This topic explains the Level Members function, which returns the set of all members of a level.
- Sets with Explicit Members

  This topic explains how to enclose specific member names in braces to create a set.
- Sets and Axes

  This topic explains the rules for using sets with axes in a report.
- Simplest Possible Query

  This topic explains how to write the simplest possible query statement.
- WHERE Clause

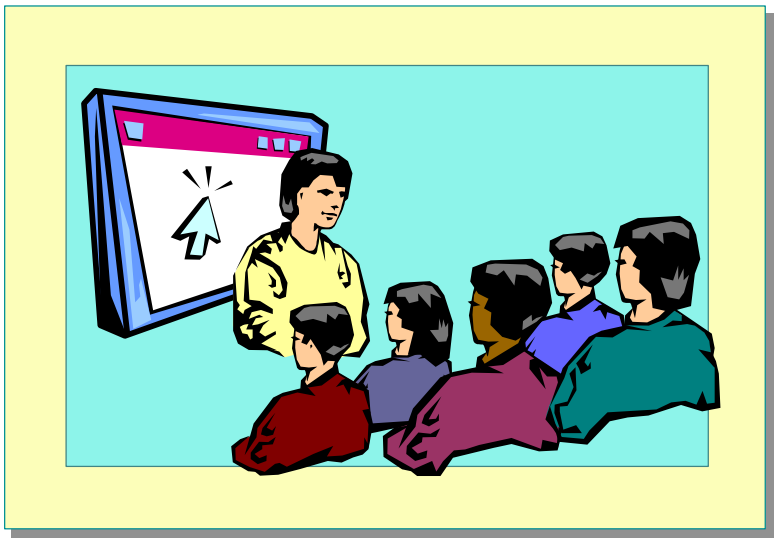
  This topic explains the use of the WHERE clause, which is equivalent to the slicer specification of the Office PivotTable list.
- Query with Axes

  This topic explains how to display multiple values in a query by adding sets to the Rows and Columns axes.

Group Activity: Using the MDX2093A Application

Topic Objective
To learn how to use the MDX2093A application and view metadata.

Lead-in
In this group activity, you will learn how to use the MDX 2093A application.



In this group activity, you can follow along on your own computer with your instructor or observe the demonstration

You will learn the interface to the MDX2093A application.

► **To open and tour the MDX2093A application**

In this procedure, you learn how to start the MDX2093A application and use it to view metadata.

1. In Windows Explorer, navigate to the folder C:\MOC\2093A\LabFiles. Double-click the file MDX2093A.exe.
2. Click **OK** to connect to the server. When prompted to open a query file, click **Cancel**, and then to create a new query file, click the **New Query File** toolbar button.

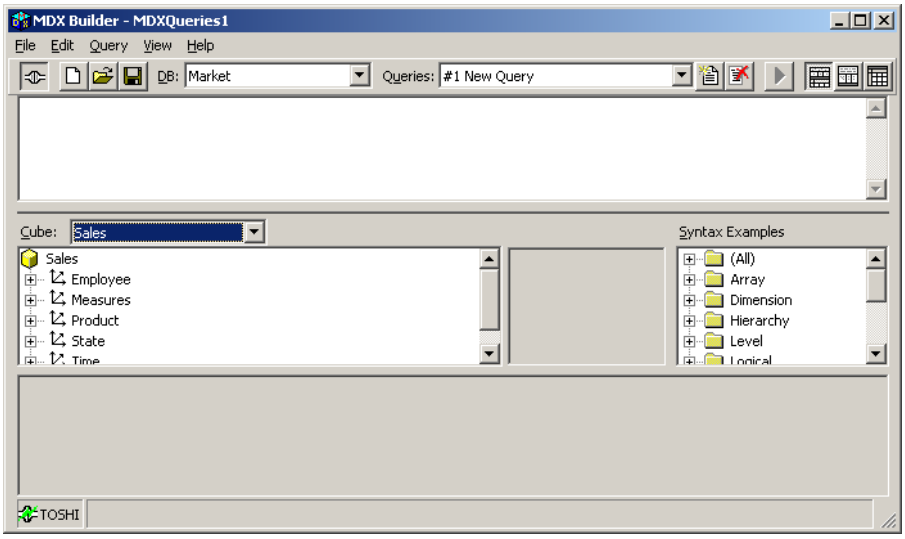
   One query file can contain multiple queries.

   The MDX Sample application window is divided into three vertical panes.

   - The top pane is the Query pane, where you build an MDX query statement.
   - The middle pane is the Metadata pane, where you inspect information in a selected cube.
   - The bottom pane is the Results pane, which shows the output of an MDX query.
   - Across the top of the application are a menu bar and a toolbar.

Tip If the Results pane does not appear, click the **Split View** button to the right of the **Run Query** button to redisplay the Results pane below the Metadata pane.

3. In the toolbar **DB** list, select **Market** as the database name. Then, at the top of the Metadata pane, click **Sales** as the cube name.



After you select a cube, the Data list—the leftmost section of the Metadata pane—displays the cube name as the top-level parent, plus the names of all the dimensions in the cube. The Data list in the Metadata pane is essentially equivalent to the Data list in the Calculated Member Builder. The Syntax Examples list is essentially equivalent to the Functions list in the Calculated Member Builder.

The MDX2093A application interface consists of three vertical panes:

- The toolbar

  The toolbar allows you to create and save a new query file. It also allows you to create and select new queries in the query file. A query file has the extension MDX, but it is a simple text file that you can view by using Microsoft Notepad or any simple text editor.

Caution The MDX2093A application will not allow you to save a query file until you have created at least one query. Be sure to use the **Save As** command on the **File** menu when you first save the file. Until you have saved the file and given it a name, the **Save** toolbar button does nothing.

- The query pane

  This is where you enter an MDX statement. Before executing a query, you must first select the correct database in the DB list.

- The metadata pane

  This area includes the Data tree, the Member list, and the Functions list. This is where you can browse the hierarchies and members of a cube, inserting dimension, level, and member names in the MDX expression. Some of the controls in the Metadata pane are similar to those in the Calculated Member Builder.

To use the Data list in the Metadata pane, you must select a cube from the Cube list. Selecting the cube has no effect on the query, but it determines the metadata you can view. To avoid confusion, be sure to select the cube you will be using as the source for your queries.

- The Results pane

  This is where you view the results.

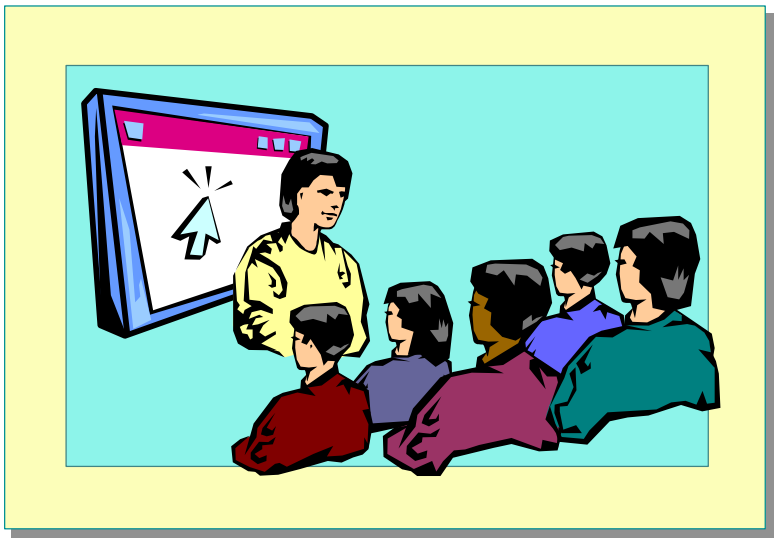
Toolbar buttons allow you to control which panes are currently displayed. Often, the most useful option is to display all three panes together.

Note The MDX2093A application was developed by modifying the MDX Sample application, which is included with Microsoft SQL Server 2000. The MDX Sample application also comes with complete source files for the Microsoft Visual Basic® project used to create it. If you are a Visual Basic programmer, you can customize or add enhancements to MDX Sample or the MDX2093A version.

Group Activity: Creating Simple MDX Query Statements

Topic Objective
To learn how to create basic query statements.

Lead-in
In this group activity, you will learn how to create simple MDX queries.



In this group activity, you can follow along on your own computer with your instructor or observe the demonstration

You will learn how to create simple MDX queries.

► To create the simplest possible MDX query

In this procedure, you learn how to create the simplest possible query from the **Sales** cube.

1. In the MDX2093A.exe application, click the **Query** menu, point at **Insert Template**, and click **No Axes**.
2. Click in the **«Cube»** token and double-click **Sales** in the **Data** list. The resulting query should be

   ```
   SELECT FROM Sales
   ```

   This is the simplest possible MDX query.
3. Click the **Run Query** button on the toolbar.

Delivery Tip
QuerySimpleD.txt in C:\MOC\2093A\Demo\D05\Answers contains the completed MDX expression for this procedure.

Note You can also press F5 to run a query.

The single number that appears in the report—35,696.00—is the total **Sales Units** for the entire cube. This is the original number from the Office PivotTable list control in the **Explore.htm** file. The query uses the default member from each dimension because the query does not specify a member from any dimension.

► To add a comment and save a query file

In this section you will learn how to add comments and save a query file.

1. Move the insertion point before the word SELECT. Type **//Simple Query**, and press ENTER. Then click the **Run Query** button to run the query again.

   You can add a comment to an MDX statement simply by entering two adjacent slash marks (//). MDX will ignore everything from that point to the end of that line.

   Verify that the query did run—that is, that nothing changed and no error message was generated.

   Note Instead of using two sequential slash marks (//) to comment the remainder of the line, you can use two sequential hyphens (--) for the same purpose. If you want to create a comment that is either a partial line or spans multiple lines, begin the comment with a slash mark followed by an asterisk (/*) and end it with an asterisk followed by a slash mark (*/).

2. Click the **Save Query File** button on the toolbar, type **Queries** as the name of the file, and then click **OK**.

   You can add multiple queries to this single file, but you cannot save a file until you have entered something in the Query pane.

► To add a set to the Columns axis

Delivery Tip
QueryOneAxisD.txt in C:\MOC\2093A\Demo\D05\Answers contains the completed MDX expression for this procedure.

In this procedure, you will learn how to add sets of members to the Columns or Rows axis.

To display more than one cell in the Results pane, you must place a set of members on an axis. Before creating the new query, you add a new query to the query file so that you can keep both queries intact.

To the right of the member list, in the Metadata pane, there is a tree labeled **Syntax Examples**. This list is the equivalent to the Functions list in the Calculated Member Builder.

Ask students which group to use—the Set group. Hint: You must put a *Set* on an axis.

1. Click the **New Query** button. On the **Query** menu, point at **Insert Template**, and click **Columns Only**.
2. Click in the **«Cube»** token, and then in the Data list, double-click **Sales**.
3. Click in the **«Set»** token. Then expand the Set group, and double-click the **Members - Level** function.

Ask students:

What you should use to replace the *«Level»* token—a level.

Where you can find one—in the **Data** list

What icon to look for—the clustered dots.

4. Click in the **«Level»** token**.** In the **Data** list, expand the **Product** dimension, and double-click the **Category** level node in the **Product** dimension. The final query should look like this:

```
SELECT
[Product].[Category].Members ON COLUMNS
FROM Sales
```

5. To view the results of the query, click the **Run Query** button.

   The labels from the **Subcategory** level appear as column headings in the report.

6. Save the query file.

   The tokens in the function templates help you know what kind of item to add to a specific location of the query. For example, you must replace the *«Level»* token with the name of a level.

   The following are the three most common types of tokens:

   - The tokens *«Set»* and *«Tuple»* are MDX constructs that do not exist in the OLAP database. Therefore, you typically replace them with functions from the appropriate groups.
   - The *«Cube»*, *«Dimension»*, *«Hierarchy»*, and *«Level»* tokens all refer to OLAP objects. Therefore, you typically replace these tokens with items from the metadata tree.
   - The *«Member»* token is the place where MDX and the cube meet. You can replace the «Member» token either with a constant member from the metadata tree or with a function from the Member group.

► To add sets to two axes

Delivery Tip
QueryTwoAxesD.txt in C:\MOC\2093A\Demo\D05\Answers contains the completed MDX expression for this procedure.

In this procedure, you add sets to both the Columns and Rows axes. If you create a query that has only one axis, it must be the Columns axis. If you create two axes, one must be Columns and one must be Rows, although it does not matter in which order they appear in the query.

1. Click the **New Query** button. On the **Query** menu, point at **Insert Template**, and click **Rows and Columns**.
2. Click in the **«Cube»** token and double-click **Sales** in the **Data** list.
3. Click in the first **«Set»** token. Double-click the **Members - Level** function in the **Set** group of the **Syntax Examples** list. Click in the **«Level»** token and double-click the **Category** level of the **Product** dimension.
4. Click in the second **«Set»** token. Double-click the **Members - Level** function in the **Set** group of the **Syntax Examples** list. Click in the **«Level»** token and double-click the **Country** level of the **State** dimension. The statement will look like this:

```
SELECT
 [Product].[Category].Members ON COLUMNS,
 [State].[Country].Members ON ROWS
FROM Sales
```

5. Click the **Run Query** button to run the query.

	Bread	Dairy	Meat
Canada	1,046.00	1,884.00	785.00
Mexico	1,289.00	2,407.00	1,470.00
USA	12,197.00	10,717.00	3,901.00

Each axis has a set of members, and the members in each set come from a level.

Note The terms *Columns* and *Rows* are simply aliases for the true names of the axes, Axis(0), and Axis(1), respectively. The underlying names make it clearer to understand why a single-axis report must include a Columns axis but not a Rows axis. Technically, an MDX query can have up to 128 axes, with alias names for the first few. There are, however, essentially no situations in which it is necessary to use more than two heading axes for a report.

Delivery Tip
QueryExplicitD.txt in C:\MOC\2093A\Demo\D05\Answers contains the completed MDX expression for this procedure.

► To create a set by using explicit member names

In this procedure, you will learn how to put one or more individual members into a set so that they can be put on an axis **QueryExplicit.txt**

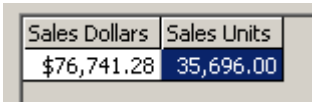
The only thing you can put on an axis is a set. So far, you have used only a set function (**Members**) to form a set to put on the axis. Remember that in the Office PivotTable list control, you can select measures individually. There is no function for selecting individual members. You must be able to put the explicit members in a set. To include members in a set, you must enclose the set in braces.

1. Click the **New Query** button. On the **Query** menu, point at **Insert Template**, and click **Columns Only**.
2. Click in the **«Cube»** token and double-click **Sales** in the **Data** list.
3. Click in the **«Set»** token and type **{ [Sales Dollars], [Sales Units] }**. The final query should be

```
SELECT
{[Sales Dollars], [Sales Units]} ON COLUMNS
FROM Sales
```

Note When you create a set without using a **Set** function, you need to enclose the set in braces, even if it contains only a single member.

4. Run the query



Observe the following about the query and the result:

- The set on the Columns axis consists of two positions.
- Each position corresponds to a tuple.
- Each tuple contains a single member.

Consequently, you could write the set as `{([Sales Dollars]), ([Sales Units])}` to show that each member is a separate tuple within the set.

Note When you create a set using a set function, you do not need to enclose the set in braces, but braces around a set are always acceptable. When you create a set by using explicit members, you must enclose the set in braces.

If the first tuple in a set consists of a single member, each subsequent tuple in that set must be a member of the same dimension.

Delivery Tip
QueryWhereD.txt in C:\MOC\2093A\Demo\D05\Answers contains the completed MDX expression for this procedure.

► To add a slicer specification

In this procedure, you will learn how to add a tuple to a slicer specification in a WHERE clause.

A slicer specification overrides the default member for each dimension that does not appear on an axis. In a PivotTable list, the slicer specification appears above the report. In an MDX query statement, the slicer specification takes the form of a WHERE clause. This is very different from a WHERE clause in an SQL statement.

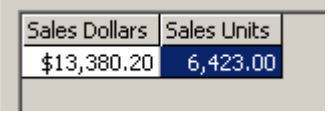
1. On the **Query** menu, point at **Insert Template** and click **WHERE Clause**.

   A **«Tuple»** token appears after the word **WHERE**, indicating that only a tuple can go in the **WHERE** clause.

2. Click in the **«Tuple»** token and type **([Bread],[1998])**. The final query will be

```
SELECT
{[Sales Dollars], [Sales Units]} ON COLUMNS
FROM Sales
WHERE ([Bread],[1998])
```

3. Run the query.



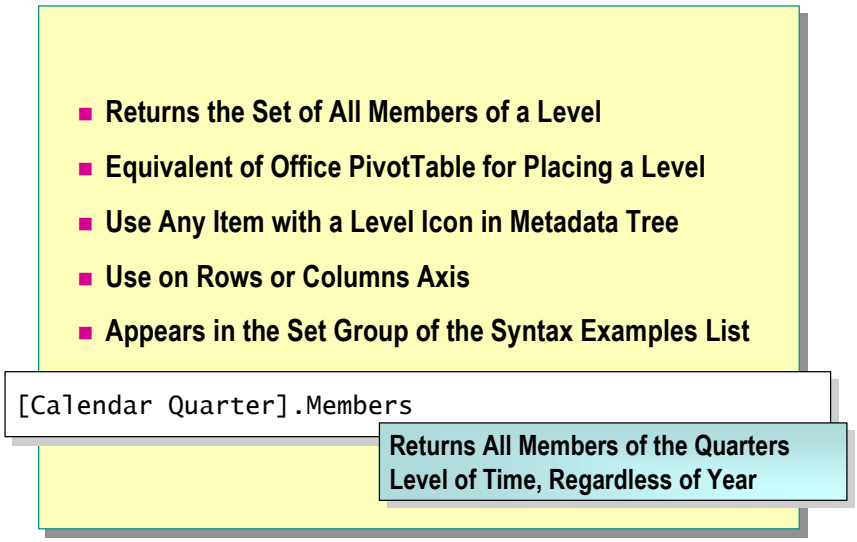
   The labels do not change, but the values do. The slicer specification in the **WHERE** clause does not appear on an axis.

   The only thing that can go into a **WHERE** clause is a single tuple. As with all tuples, if the tuple contains more than one member, it must be enclosed in parentheses. If the tuple contains only a single member, it does not need parentheses.

Level Members Function

Topic Objective
To show how to create a query that returns a set of members.

Lead-in
The **Level Members** function returns the set of all members of a level.



The most useful tool for creating a set of members is to retrieve all the members from a level. The **Level Members** function returns the set of all members of a level, without respect to any members higher or lower in the hierarchy.

Syntax

«*Level*».Members

Example

The following example returns all members of the **Quarters** level of the **Time** dimension, regardless of year.

```
[Calendar Quarter].Members
```

Delivery Tip
Because students have already been exposed to the **Level Members** function in the previous group activity, treat this topic as a review—that is, do not spend a lot of time.

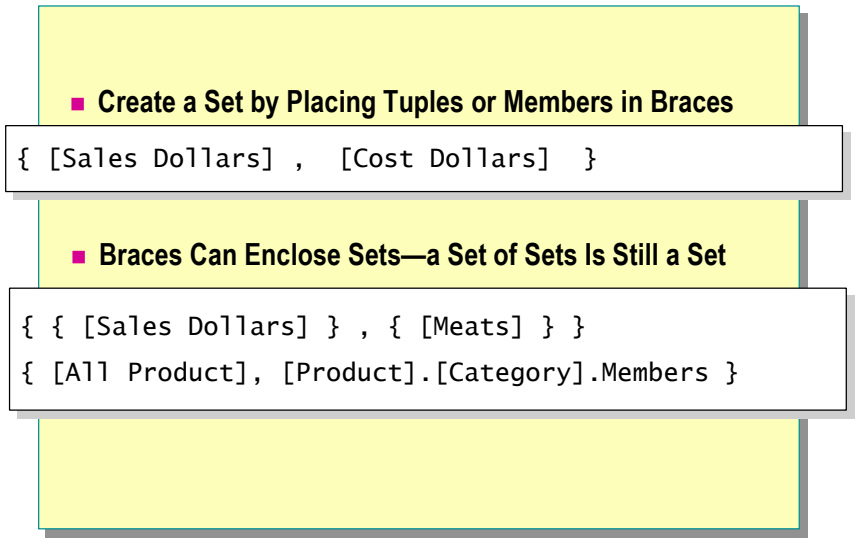
When you use the **Level Members** function, consider the following facts and guidelines:

- The **Level Members** function does essentially what the Office PivotTable list does—when you drag a level to the report, all the members of the level appear.
- For the level argument, you can use any item in the metadata tree that is preceded by a level—that is, clustered dot—icon.
- You can use the **Level Members** function on a Rows or Columns axis—for example,[Calendar Quarter].Members ON COLUMNS.The **Level Members** function appears in the Set group of the **Syntax Examples** list.

Sets with Explicit Members

Topic Objective
To describe how to use sets to return a specific range of members.

Lead-in
You can create sets that include explicit members.



You can create sets that include explicit members. This is what the Office PivotTable list control does when you add measures to a report.

Example 1

In an MDX query statement, you include explicit member names inside braces. The braces convert the members into a set. For example:

```
{ [Sales Dollars] , [Cost Dollars] }
```

Delivery Tip
Because students have already been exposed to using sets to return explicit members in the previous group activity, treat this topic as a review—that is, do not spend a lot of time.

Because each element of a set is actually a tuple, you can enclose each of the member names in parentheses. If the tuple includes members from only a single dimension, the parentheses are not necessary.

If you want to put a single member on an axis, you must put braces around the member, because a set is the only thing that can go on an axis.

Example 2

You can place extra brackets around any set to create a set of sets, which is still a set. You combine two sets by putting a pair of brackets around them. For example:

```
{ { [Sales Dollars] } , { [Meats] } }
```

Example 3

You can use braces to add an individual member, such as a total, to a set of all the members of a level. For example:

```
{ [All Product], Catgory.Members }
```

Note You can also specify a range of members by using a colon, much as when creating a range in Microsoft Excel. For example, {[Bread]:[Meat]} returns a set containing all the members between Bread and Meat, inclusive. The beginning and ending members of a range must be on the same level of the same dimension. This notation is used only infrequently, because there are usually set functions that can create the desired result. When using the range operator, you must still enclose the members in braces.

Sets and Axes

Topic Objective
To review implications of using sets with axes in a query

Lead-in
Let's talk about how sets work with axes in an MDX query.

- **Use a Set to Display Multiple Members on an Axis**
- **A Set Is a Collection of One or More Tuples**
- **A Set Appears Only on an Axis—Not in WHERE Clause**
- **Each Axis Must Include a Set**
- **You Cannot Skip an Axis**
- **A Dimension Can Appear Once on Axes and WHERE**
- **Query Statement May Have up to 128 Axes**
- **Columns = Axis(0); Rows = Axis(1)**

Delivery Tip
Because students have already been exposed to how sets work with axes in the previous group activity, treat this topic as a review—that is, do not spend a lot of time.

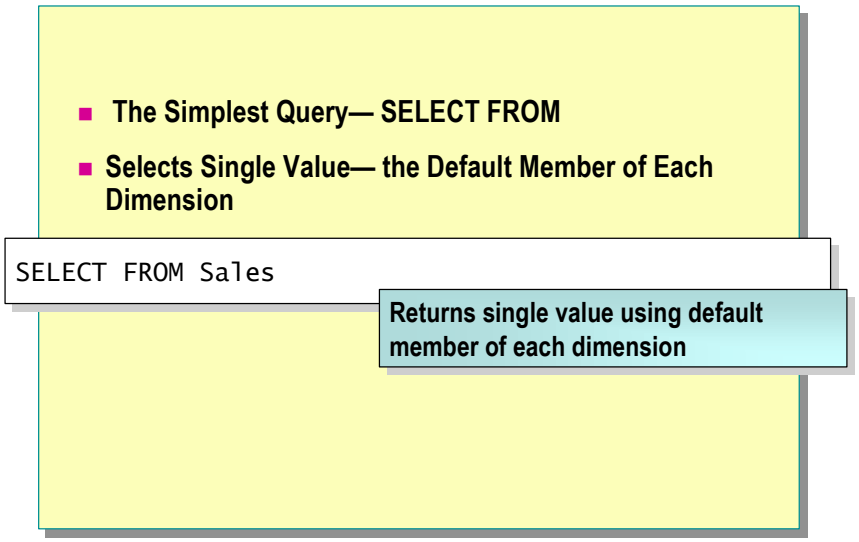
Sets work with axes in an MDX query. When using sets and axes, consider the following facts and guidelines:

- If you want to want to display multiple members from a single dimension in a report, you must put the members into a set and place the set on an axis
- A set is a collection of one or more tuples. Each tuple may contain a member from only one dimension, in which case the set appears to be a collection of one or more members of a dimension, but, technically, the set always contains tuples.
- A set appears only on an axis. You must never put a set in the slicer specification—that is, in a WHERE clause. The slicer specification requires a tuple.
- Each axis must include a set. You cannot have an axis without a set, and each axis contains one and only one set.
- You cannot skip an axis—that is, if there is a Rows axis, there must be a Columns axis, but the order of axes does not matter.
- A dimension can appear only one time on the axes and the WHERE clause. If you include the same dimension on more than one axis, the dimension would appear in more than one in a tuple, which would then be an invalid coordinate.
- An MDX query statement can have as many as 128 axes, a fraction of the 65,535 dimensions of a cube, but far more than you could ever use.
- The official name for each axis consists of the word *Axis* followed by the axis number in parentheses. The first axis is numbered zero. The term *Columns* is equivalent to *Axis(0)*. The term *Rows* is equivalent to *Axis(1)*.

Simplest Possible Query

Topic Objective
To review the simplest possible MDX statement.

Lead-in
A simple query consists of only a single cell—the result of a single tuple.



A simple query consists of only a single cell—the result of a single tuple. The simplest possible query consists of only the keywords SELECT and FROM, along with the name of the cube.

This selects a single value by using the default member of each dimension.

Syntax

```
SELECT FROM «Cube»
```

Example

```
SELECT FROM Sales
```

Delivery Tip
Because students have already been exposed to the simplest query in the previous group activity, treat this topic as a review—that is, do not spend a lot of time.

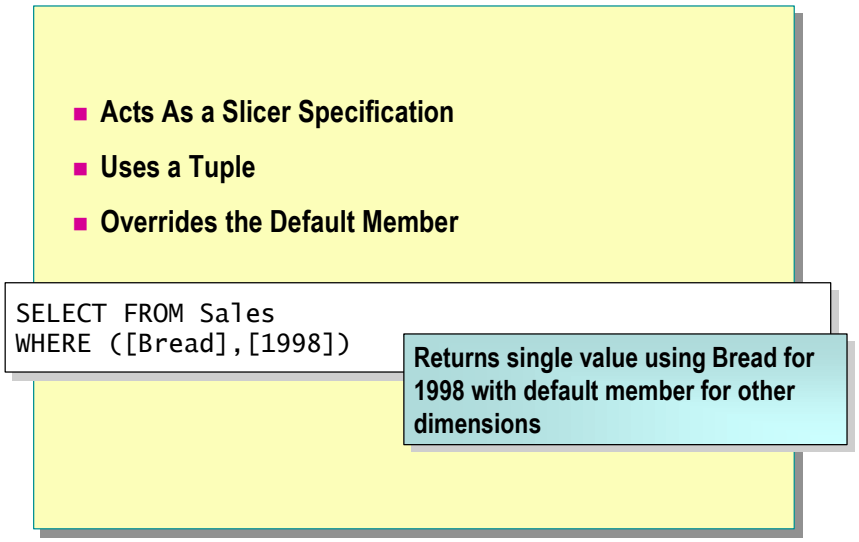
This returns a report that contains a single value. The tuple for the value consists of the default member of each dimension.

Note You can add a comment to an MDX statement simply by entering two adjacent slash marks (//). MDX ignores everything from that point to the end of that line. You can also add a comment by entering two adjacent hyphens (--). Again, MDX ignores everything from the hyphens to the end of the line. To control the beginning and ending point of a comment, use a slash mark followed by an asterisk (/*) to begin the comment, and an asterisk followed by a slash mark (*/) to end the comment.

WHERE Clause

Topic Objective
To describe the use of a WHERE clause.

Lead-in
A WHERE clause is equivalent to the slicer specification of a PivotTable list report.



A WHERE clause is equivalent to the slicer specification of a Microsoft Office PivotTable list report. You can think of WHERE as a Filter clause. The technical term for the WHERE clause is the *slicer specification*.

Syntax

```
SELECT
FROM «Cube»
WHERE «Tuple»
```

Example

```
SELECT
FROM Sales
WHERE ([Bread], [1998])
```

Delivery Tip
Because students have already been exposed to the WHERE clause in the previous group activity, treat this topic as a review—that is, do not spend a lot of time.

This query statement returns a report containing a single value, with **Bread** for the **Product** dimension, 1998 for the **Time.Calendar** dimension, and the default member from all remaining dimensions.

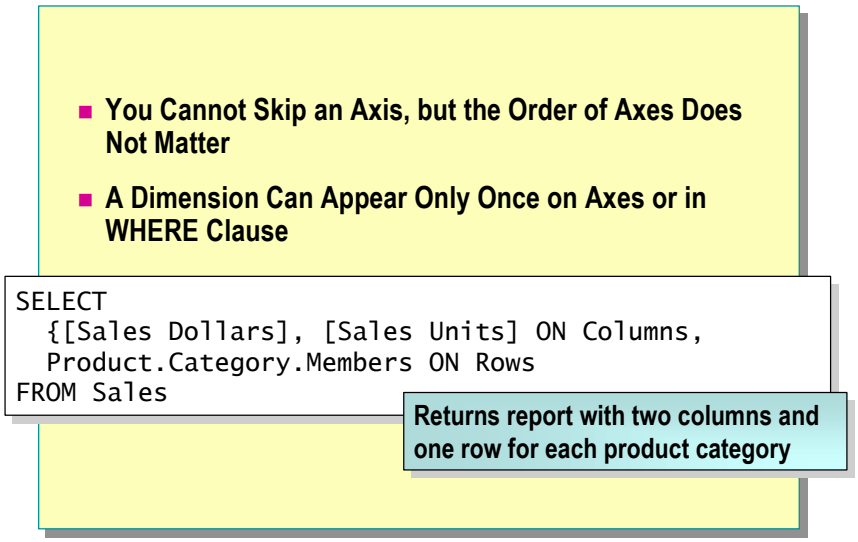
When using a WHERE clause, consider the following facts and guidelines:

- The WHERE clause accepts a tuple—with one or more dimensions represented.
- In the context of the query statement, the WHERE clause overrides the default member for each dimension represented in the tuple.

Query with Axes

Topic Objective
To describe how to display multiple values in a query.

Lead-in
To display multiple values in a query, you must put members into a set on an axis.



To display multiple values in a query, you must put members into a *set* on an *axis*. A typical query contains two axes: Columns and Rows.

Syntax

```
SELECT
  «Set» ON COLUMNS,
  «Set» ON ROWS
  FROM «Cube»
  WHERE «Tuple»
```

Example

```
SELECT
  {[Sales Dollars], [Sales Units] ON COLUMNS,
  Product.Category.Members ON ROWS
  FROM Sales
```

Delivery Tip
Because students have already been exposed to the queries with axes in the previous group activity, treat this topic as a review—that is, do not spend a lot of time.

This query statement returns a report with two columns—one for **Sales Dollars** and one for **Sales Units**. The report has one row for each member of the **Category** level of the **Product** dimension.

The Rows and Columns axes definitions follow the SELECT keyword, and a comma separates the axes.

Lab A: Creating Simple MDX Query Statements

Topic Objective
To introduce the lab.

Lead-in
In this lab, you will create simple MDX query statements by using the MDX2093A application.

Objectives

Explain the lab objectives.

After completing this lab, you will be able to:

- Create simple MDX query statements.

Prerequisites

Before working on this lab, you must have successfully completed modules 1 through 4 in course 2093A *Implementing Business Logic with MDX in Microsoft® SQL Server™ 2000*.

Estimated time to complete this lab: 15 minutes

Exercise 1
Creating Simple MDX Query Statements

Delivery Tip
The procedures in this exercise essentially replicate the previous group activities, but without the answers.

In this exercise, you will create simple queries by putting sets on axes, using the **Level Members** function, and creating sets by using explicit members.

As you complete each procedure, compare the result as shown in the procedure screen shot to the result on your own computer. If there is a difference, recheck your entry and refer back to the related group activity procedures as necessary.

If you still cannot reconcile your result, then refer to the answer file in which is located in the following folder: C:\MOC\2093A\Labfiles\L05\Answers.

You can copy and paste expressions from this file into the MDX2093A application for any given procedure.

Before beginning the exercises in this lab, verify that the following objects exist in Analysis Manager:

- The **Market** database
- The **Basic Sales** virtual cube

If these objects do not exist in Analysis Manager, you must perform the following procedure before you continue with this lab.

► To restore the Market database

1. In the left pane of Analysis Manager, expand the **Analysis Services** folder.
2. Right-click the server icon, and then click **Restore Database**.
3. Navigate to the C:\Moc\2093A\Batches folder.
4. Select **Market.cab**, click **Open**, and then click **Restore**.

► To create a simple query with a comment

In this procedure, you will create a very simple MDX query and add a comment in the **Query** pane. NoAxes.txt in the Answer folder is the completed MDX expression used in this procedure.

1. In Windows Explorer, navigate to the folder C:\MOC\2093A\LabFiles, and double-click the file **MDX2093A.exe**. When prompted for the server name, click **OK**. When prompted for a query file name, click **Cancel**, and then click the **New Query File** button.
2. In the MDX2093A application, enter an MDX statement that creates a report with no axes and no WHERE clause, which shows the value of the default member of each dimension in the **Sales** cube, and add a comment describing the query to the Query pane.

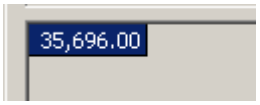
What MDX query statement did you use?

```
SELECT
FROM Sales
```

What comment did you add?

```
//Simple Query
```

3. Run the query. If the comment is entered correctly, you do not get an error message. Verify that the Results pane content is similar to the following:



► To add a set to the Columns axis

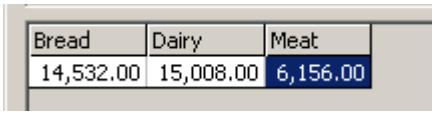
In this procedure, you will create an MDX expression with one axis by using the **Level Members** function. OneAxis.txt in the Answer folder is the completed MDX expression used in this procedure.

1. To create a new query, click the **New Query** button. Enter an MDX statement that creates a report showing product category names across the Columns axis.

   What MDX query statement did you use?

```
SELECT
 Category.Members ON COLUMNS
FROM Sales
```

2. Run the query. Verify that the Results pane content is similar to the following:



► To add sets to two axes

In this procedure, you will create an MDX query with two axes by using the **Level Members** function. TwoAxes.txt in the Answer folder is the completed MDX expression used in this procedure.

1. Create a new query by entering an MDX statement that creates a report showing Region names from the **State** dimension across the Columns axis and Subcategory names from the **Product** dimension down the Rows axis.

   What MDX query statement did you use?

```
SELECT
 Region.Members ON COLUMNS,
 Subcategory.Members ON ROWS
FROM Sales
```

   _____

   _____

2. Run the query. Verify that the Results pane content is similar to the following:

	Canada	Mexico	North West	South West
Bagels	118.00	114.00	1,080.00	229.00
Muffins	484.00	598.00	4,358.00	1,005.00
Sliced Bread	444.00	577.00	4,446.00	1,079.00
Cheese	958.00	1,187.00	3,904.00	1,444.00
Milk	489.00	713.00	2,174.00	775.00
Sour Cream	239.00	250.00	797.00	321.00
Yogurt	198.00	257.00	930.00	372.00
Deli Meats	249.00	435.00	900.00	393.00
Fresh Chicken	55.00	161.00	202.00	67.00
Frozen Chicken	202.00	333.00	614.00	189.00
Hamburger	124.00	238.00	452.00	201.00
Hot Dogs	155.00	303.00	627.00	256.00

► To create a set by using explicit member names

In this procedure, you will create an MDX query by using explicit members in a set. ExplicitMembers.txt in the Answer folder is the completed MDX expression used in this procedure.

1. Create a new query by entering an MDX statement that creates a report showing both the **Sales Dollars** and **Sales Units** measures on the Columns axis.

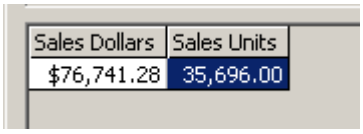
   What MDX query statement did you use?

   ```
   SELECT
    {[Sales Dollars], [Sales Units]} ON COLUMNS
   FROM Sales
   ```

   _____

   _____

2. Run the query. Verify that the Results pane content is similar to the following:



► To add a slicer specification

In this procedure, you will create an MDX query with an explicit set on the Columns axis and a set generated by the **Members** function on the Rows axis, and a WHERE clause. Report.txt in the Answer folder is the completed MDX expression used in this procedure.

1. Create a new query by entering an MDX statement that creates a report showing **Sales Dollars** and **Sales Units** on the Columns axis and the categories from the **Product** dimension on the Rows axis. The report should show values only for **USA** in **1998**, but those labels should not appear on the report.

   What MDX query statement did you use?

```
SELECT
 {[Sales Dollars], [Sales Units]} ON COLUMNS,
 Category.Members ON ROWS
FROM Sales
WHERE ([1998],[USA])
```

   _____

   _____

2. Run the query. Verify that the Results pane content is similar to the following:

	Sales Dollars	Sales Units
Bread	$14,198.45	6,638.00
Dairy	$23,358.73	10,717.00
Meat	$8,648.16	3,901.00

Review

Topic Objective
To reinforce module objectives by reviewing key points.

Lead-in
The review questions cover some of the key concepts taught in the module.

- Understanding MDX Query Statements
- Creating Simple MDX Query Statements

1. What are some of the differences between a cube and an MDX report?

   A cube has dimension, whereas a report has axes—and multiple dimensions can appear on one axis.

   A cube contains all possible values at all aggregation levels, whereas a report is a small selection of values.

2. What is the difference between an MDX expression and an MDX query statement?

   An expression returns a single value; a query returns a report.

3. To retrieve a value from a cell in the cube requires a complete tuple. What are the three sources for providing members for that complete tuple?

   The default member of a dimension.

   The tuple in the WHERE clause.

   The tuples in the sets on the axes.

4. What is the simplest possible MDX query statement for retrieving a value from a cube named Finance?

   SELECT FROM Finance.

5. What is the keyword that begins the slicer specification, and what can you put in a slicer specification?

   WHERE is the keyword, and you can put only a tuple in the slicer specification.

6. What can you put on an axis? Can you put a single member on an axis?

   A set. You can put a member on an axis as long as you put braces around it to turn it into a set.

7. Can you have a Rows axis without a Columns axis? Can you have a Rows axis before a Columns axis?

   No. Yes.