SQL Server 2012 Tutorials: Analysis Services - Multidimensional Modeling

SQL Server 2012 Books Online

Step-by-Step

Microsoft

SQL Server 2012 Tutorials: Analysis Services - Multidimensional Modeling

SQL Server 2012 Books Online

Summary: This tutorial describes how to use SQL Server Data Tools to develop and deploy an Analysis Services project, using the fictitious company Adventure Works Cycles for all examples.

Category: Step-by-Step
Applies to: SQL Server 2012
Source: SQL Server Books Online (link to source content)
E-book publication date: June 2012

Contents

Multidimensional Modeling (Adventure Works Tutorial)

Welcome to the Analysis Services Tutorial. This tutorial describes how to use SQL Server Data Tools to develop and deploy an Analysis Services project, using the fictitious company Adventure Works Cycles for all examples.

What You Will Learn

In this tutorial, you will learn the following:

- How to define data sources, data source views, dimensions, attributes, attribute relationships, hierarchies, and cubes in an Analysis Services project within SQL Server Data Tools.
- How to view cube and dimension data by deploying the Analysis Services project to an instance of Analysis Services, and how to then process the deployed objects to populate them with data from the underlying data source.
- How to modify the measures, dimensions, hierarchies, attributes, and measure groups in the Analysis Services project, and how to then deploy the incremental changes to the deployed cube on the development server.
- How to define calculations, Key Performance Indicators (KPIs), actions, perspectives, translations, and security roles within a cube.

A scenario description accompanies this tutorial so that you can better understand the context for these lessons. For more information, see Analysis Services Tutorial Scenario.

Prerequisites

You will need sample data, sample project files, and software to complete all of the lessons in this tutorial. For instructions on how to find and install the prerequisites for this tutorial, see Install Sample Data and Projects for the Analysis Services Multidimensional Modeling Tutorial.

Additionally, the following permissions must be in place to successfully complete this tutorial:

- You must be a member of the Administrators local group on the Analysis Services computer or be a member of the server administration role in the instance of Analysis Services.
- You must have Read permissions in the **AdventureWorksDW2012** sample database.

Lessons

This tutorial includes the following lessons.

Lesson	Estimated time to complete
Lesson 1: Defining a Data Source View within an Analysis Services Project	15 minutes
Lesson 2: Defining and Deploying a Cube	30 minutes
Lesson 3: Modifying Measures, Attributes and Hierarchies	45 minutes
Lesson 4: Defining Advanced Attribute and Dimension Properties	120 minutes
Lesson 5: Defining Relationships Between Dimensions and Measure Groups	45 minutes
Lesson 6: Defining Calculations	45 minutes
Lesson 7: Defining Key Performance Indicators (KPIs)	30 minutes
Lesson 8: Defining Actions	30 minutes
Lesson 9: Defining Perspectives and Translations	30 minutes
Lesson 10: Defining Administrative and User Roles	15 minutes

Note

The cube database that you will create in this tutorial is a simplified version of the Analysis Services multidimensional model project that is part of the Adventure Works sample databases available for download on the codeplex site. The tutorial version of the Adventure Works multidimensional database is simplified to bring greater focus to the specific skills that you will want to master right away. After you complete the tutorial, consider exploring the multidimensional model project on your own to further your understanding of Analysis Services multidimensional modeling.

Next Step

To begin the tutorial, continue to the first lesson: Lesson 1: Defining a Data Source View within an Analysis Services Project.

See Also

Tutorials (Analysis Services)

Analysis Services Tutorial Scenario

This tutorial is based on Adventure Works Cycles, a fictitious company. Adventure Works Cycles is a large, multinational manufacturing company that produces and distributes metal and composite bicycles to commercial markets in North America, Europe, and Asia. The headquarters for Adventure Works Cycles is Bothell, Washington, where the company employs 500 workers. Additionally, Adventure Works Cycles employs several regional sales teams throughout its market base.

In recent years, Adventure Works Cycles bought a small manufacturing plant, Importadores Neptuno, which is located in Mexico. Importadores Neptuno manufactures several critical subcomponents for the Adventure Works Cycles product line. These subcomponents are shipped to the Bothell location for final product assembly. In 2005, Importadores Neptuno became the sole manufacturer and distributor of the touring bicycle product group.

Following a successful fiscal year, Adventure Works Cycles now wants to broaden its market share by targeting advertising to its best customers, extending product availability through an external Web site, and reducing the cost of sales by reducing production costs.

Current Analysis Environment

To support the data analysis needs of the sales and marketing teams and of senior management, the company currently takes transactional data from the AdventureWorks2012 database, and non-transactional information such as sales quotas from spreadsheets, and consolidates this information into the **AdventureWorksDW2012** relational data warehouse. However, the relational data warehouse presents the following challenges:

- Reports are static. Users have no way to interactively explore the data in the reports to obtain more detailed information, such as they could do with a Microsoft Office Excel pivot table. Although the existing set of predefined reports is sufficient for many users, more advanced users need direct query access to the database for interactive queries and specialized reports. However, because of the complexity of the **AdventureWorksDW2012** database, too much time is needed for such users to master how to create effective queries.
- Query performance is widely variable. For example, some queries return results very quickly, in only a few seconds, while other queries take several minutes to return.
- Aggregate tables are difficult to manage. In an attempt to improve query response times, the data warehouse team at Adventure Works built several aggregate tables in the **AdventureWorksDW2012** database. For example, they built a table that summarizes sales by month. However, while these aggregate tables greatly improve query performance, the infrastructure that they built to maintain the tables over time is fragile and prone to errors.

- Complex calculation logic is buried in report definitions and is difficult to share between reports. Because this business logic is generated separately for each report, summary information sometimes is different between reports. Therefore, management has limited confidence in the data warehouse reports.
- Users in different business units are interested in different views of the data. Each group is distracted and confused by data elements that are irrelevant to them.
- Calculation logic is particularly challenging for users who need specialized reports. Because such users must define the calculation logic separately for each report, there is no centralized control over how the calculation logic is defined. For example, some users know that they should use basic statistical techniques such as moving averages, but they do not know how to construct such calculations and so do not use these techniques.
- It is difficult to combine related sets of information. Specialized queries that combine two sets of related information, such as sales and sales quotas, are difficult for business users to construct. Such queries overwhelmed the database, so the company requires that users request cross-subject-area sets of data from the data warehouse team. As a result, only a handful of predefined reports have been defined that combine data from multiple subject areas. Additionally, users are reluctant to try to modify these reports because of their complexity.
- Reports are focused primarily on business information in the United States. Users in the non-U.S. subsidiaries are very dissatisfied with this focus, and want to be able to view reports in different currencies and different languages.
- Information is difficult to audit. The Finance department currently uses the **AdventureWorksDW2012** database only as a source of data from which to query in bulk. They then download the data into individual spreadsheets, and spend significant time preparing the data and manipulating the spreadsheets. The corporate financial reports are therefore difficult to prepare, audit, and manage across the company.

The Solution

The data warehouse team recently performed a design review of the current analysis system. The review included a gap analysis of current issues and future demands. The data warehouse team determined that the **AdventureWorksDW2012** database is a well-designed dimensional database with conformed dimensions and surrogate keys. Conformed dimensions enable a dimension to be used in multiple data marts, such as a time dimension or a product dimension. Surrogate keys are artificial keys that link dimension and fact tables and that are used to ensure uniqueness and to improve performance. Moreover, the data warehouse team determined that there currently are no significant problems with the loading and management of the base tables in the **AdventureWorksDW2012** database. The team has therefore decided to use Microsoft Analysis Services to accomplish the following:

- Provide unified data access through a common metadata layer for analytical analysis and reporting.
- Simplify users' view of data, speeding the development of both interactive and predefined queries and predefined reports.
- Correctly construct queries that combine data from multiple subject areas.
- Manage aggregates.
- Store and reuse complex calculations.
- Present a localized experience to business users outside the United States.

The lessons in the Analysis Services tutorial provide guidance in building a cube database that meets all of these goals. To get started, continue to the first lesson: Lesson 1: Create a New Tabular Model Project.

See Also

Analysis Services Tutorial

Install Sample Data and Projects for the Analysis Services Multidimensional Modeling Tutorial

Use the instructions and links provided in this topic to install all of the data and project files used in the Analysis Services Multidimensional Modeling Tutorial.

Step 1: Install Software

The lessons in this tutorial assume that you have the following software installed. All of the following software is installed using SQL Server installation media. For simplicity of deployment, you can install all of the features on a single computer. To install these features, run SQL Server Setup and select them from the Feature Selection page. For more information, see Install SQL Server 2012 from the Installation Wizard (Setup).

- Database Engine
- Analysis Services
- SQL Server Data Tools (SSDT)
- SQL Server Management Studio

Optionally, consider installing Excel to browse your multidimensional data as you proceed through the tutorial. Installing Excel enables the **Analyze in Excel** feature that starts Excel using a PivotTable field list that is connected to the cube you are building. Using Excel to browse data is recommended because you can quickly build a pivot report that lets you interact with the data.

Alternatively, you can browse data using the built-in MDX query designer that is built into SQL Server Data Tools (SSDT). The query designer returns the same data, except the data is presented as a flat rowset.

Step 2: Install Databases

An Analysis Services multidimensional model uses transactional data that you import from a relational database management system. For the purposes of this tutorial, you will use the following relational database as your data source.

- **AdventureWorksDW2012** – This is a relational data warehouse that runs on a Database Engine instance. It provides the original data that will be used by the Analysis Services databases and projects that you build and deploy throughout the tutorial.

To install this database, do the following:

1. Download the AdventureWorkDW2012 database from the product samples page on codeplex.

   The database file name is AdvntureWorksDW2012_Data.mdf. The file should be in the Downloads folder on your computer.
2. Copy the AdventureWorksDW2012_Data.mdf file to the data directory of the local SQL Server Database Engine instance. By default, it is located at C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Data.
3. Start SQL Server Management Studio and connect to the Database Engine instance.
4. Right-click Databases, click **Attach**.
5. Click **Add**.
6. Select the **AdventureWorksDW2012_Data.mdf** database file and click **OK**. If the file is not listed, check the C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Data folder to be sure the file is there.
7. In database details, remove the Log file entry. The sample download does not include a log file. A new log file will be created automatically when you attach the database. Select the log file and click **Remove**, and then click **OK** to attach just the primary database file.

Step 3: Grant Database Permissions

The sample projects use data source impersonation settings that specify the security context under which data is imported or processed. By default, the impersonation settings specify the Analysis Services service account for accessing the data. To use this default setting, you must ensure that the service account under which Analysis Services runs has data reader permissions on the **AdventureWorksDW2012** database.

Note

For learning purposes, it is recommended that you use the service account impersonation option and grant data reader permissions to the service account in

SQL Server. Although other impersonation options are available, not all of them are suitable for processing operations. Specifically, the option for using the credentials of the current user is not supported for processing.

1. Determine the service account. You can use SQL Server Configuration Manager or the Services console application to view account information. If you installed Analysis Services as the default instance, using the default account, the service is running as **NT Service\MSSQLServerOLAPService**.
2. In Management Studio, connect to the database engine instance.
3. Expand the Security folder, right-click Logins and select **New Login**.
4. On the General page, in Login name, type **NT Service\MSSQLServerOLAPService** (or whatever account the service is running as).
5. Click **User Mapping**.
6. Select the checkbox next to the **AdventureWorksDW2012** database. Role membership should automatically include **db_datareader** and **public**. Click **OK** to accept the defaults.

Step 4: Install Projects

The tutorial includes sample projects so that you can compare your results against a finished project, or start a lesson that is further on in the sequence.

The project file for Lesson 4 is particularly important because it provides the basis for not only that lesson, but all the subsequent lessons that follow. In contrast with the previous project files, where the steps in the tutorial result in an exact copy of the completed project files, the Lesson 4 sample project includes new model information that is not found in the model you built in lessons 1 through 3. Lesson 4 assumes that you are starting with a sample project file that is available in the following download.

1. Download the Analysis Services Tutorial SQL Server 2012 on the product samples page on codeplex.

   The "Analysis Services Tutorial SQL Server 2012.zip" file will be saved to the Downloads folder on your computer.
2. Move the .zip file to a folder just below the root drive (for example, C:\Tutorial). This step mitigates the "Path too long" error that sometimes occurs if you attempt to unzip the files in the Downloads folder.
3. Unzip the sample projects: right-click on the file and select **Extract All**. After you extract the files, you should have the following projects installed on your computer:
   - Lesson 1 Complete
   - Lesson 2 Complete
   - Lesson 3 Complete
   - Lesson 4 Complete
   - Lesson 4 Start

- Lesson 5 Complete
- Lesson 6 Complete
- Lesson 7 Complete
- Lesson 8 Complete
- Lesson 9 Complete
- Lesson 10 Complete

4. Start SQL Server Data Tools (SSDT).
5. Open the solution (.sln) file that corresponds to the lesson you are using.
6. Deploy the solution to verify that database permissions and server location information is set up correctly.

   If Analysis Services and the Database Engine are installed as the default instance (MSSQLServer) and all software is running on the same computer, you can click **Deploy** on the Project menu to build and deploy the sample project to the local Analysis Services instance. During deployment, data will be processed (or imported) from the **AdventureWorksDW2012** database on the local Database Engine instance.

   If you encounter errors, review the previous steps on setting up database permissions. You might also need to change server names, for example, an instance name if you installed either Analysis Services or the Database Engine as a named instance. Additionally, if the servers are on a different server, you might need to configure Windows Firewall to allow access to the servers.
7. In SQL Server Management Studio, connect to Analysis Services. Verify that a database named **Analysis Services Tutorial** is running on the server.

Next Step

You are now ready to use the tutorial. For more information about how to get started, see Multidimensional Modeling (Adventure Works Tutorial).

See Also

Install SQL Server 2012 from the Installation Wizard (Setup)

Configuring the Windows Firewall to Allow Analysis Services Access

Configuring the Windows Firewall to allow SQL Server Access

Lesson 1: Defining a Data Source View within an Analysis Services Project

Designing a business intelligence application in SQL Server starts with creating an Analysis Services project in SQL Server Data Tools (SSDT). Within this project, you define all the elements of your solution, starting with a data source view.

This lesson contains the following tasks:

Creating an Analysis Services Project

In this task, you create the Analysis Services Tutorial project, based on an Analysis Services multidimensional model template.

Defining a Data Source

In this task, you specify the **AdventureWorksDW2012** database as the data source for the Analysis Services dimensions and cubes that you will define in subsequent lessons.

Defining a Data Source View

In this task, you define a single unified view of the metadata from selected tables in the **AdventureWorksDW2012** database.

Modifying Default Table Names

In this task, you modify table names in the data source view, so that the names of subsequent Analysis Services objects that you define will be more user-friendly.

Compare your results against a sample project file that was built for this lesson. For more information about downloading the sample projects that go with this tutorial, see SSAS Multidimensional Model Projects for SQL Server 2012 on the product samples page on codeplex.

Next Lesson

Lesson 2: Defining and Deploying a Cube

See Also

Create an Analysis Services Project
Defining Data Sources (SSAS)
Designing Data Source Views (SSAS)
Analysis Services Tutorial Scenario
Analysis Services Tutorial

Creating an Analysis Services Project

In the following task, you use SQL Server Data Tools (SSDT) to create a new Analysis Services project named **Analysis Services Tutorial**, based on the Analysis Services Project template. A *project* is a collection of related objects. Projects exist within a solution, which includes one or more projects. For more information, see Create an Analysis Services Project.

Procedures

To create a new Analysis Services project

1. Click **Start**, point to **All Programs**, point to **Microsoft SQL Server 2012**, and then click **SQL Server Data Tools**.

   The Microsoft Visual Studio development environment opens.
2. On the Start page of Visual Studio, click **New Project**.
3. In the **New Project** dialog box, in the **Installed Templates** pane, expand **Business Intelligence**, and then select **Analysis Services**. Choose the **Analysis Services Multidimensional and Data Mining Project** template.

   Notice the default project name, location, and the default solution name are generated in the bottom of the dialog box. By default, a new directory is created for the solution.
4. Change the project Name to **Analysis Services Tutorial**, which also changes the **Solution name** box, and then click **OK**.

You have successfully created the **Analysis Services Tutorial** project, based on the **Analysis Services Multidimensional and Data Mining Project** template, within a new solution that is also named **Analysis Services Tutorial**.

Next Task in Lesson

Defining a Data Source

See Also

Developing an Analysis Services Project (SSAS)

Create an Analysis Services Project (SSAS)

Defining a Data Source

After you create an Analysis Services project, you generally start working with the project by defining one or more data sources that the project will use. When you define a data source, you are defining the connection string information that will be used to connect to the data source. For more information, see Defining a Data Source using the Data Source Wizard.

In the following task, you define the AdventureWorksDWSQLServer2012 sample database as the data source for the Analysis Services Tutorial project. While this database is located on your local computer for the purposes of this tutorial, source databases are frequently hosted on one or more remote computers.

Procedures

To define a new data source

1. In Solution Explorer (on the right of the Microsoft Visual Studio window), right-click **Data Sources**, and then click **New Data Source**.
2. On the **Welcome to the Data Source Wizard** page of the **Data Source Wizard**, click **Next** to open the **Select how to define the connection** page.
3. On the **Select how to define the connection** page, you can define a data source based on a new connection, based on an existing connection, or based on a previously defined data source object. In this tutorial, you define a data source based on a new connection. Verify that **Create a data source based on an existing or new connection** is selected, and then click **New**.
4. In the **Connection Manager** dialog box, you define connection properties for the data source. In the **Provider** list box, verify that **Native OLE DB\SQL Server Native Client 11.0** is selected.

   Analysis Services also supports other providers, which are displayed in the **Provider** list.
5. In the **Server name** text box, type **localhost**.

   To connect to a named instance on your local computer, type **localhost\<instance name>**. To connect to the specific computer instead of the local computer, type the computer name or IP address.
6. Verify that **Use Windows Authentication** is selected. In the **Select or enter a database name** list, select **AdventureWorksDW2012**.
7. Click **Test Connection** to test the connection to the database.
8. Click **OK**, and then click **Next**.
9. On the **Impersonation Information** page of the wizard, you define the security credentials for Analysis Services to use to connect to the data source. Impersonation affects the Windows account used to connect to the data source when Windows Authentication is selected. Analysis Services does not support impersonation for processing OLAP objects. Select **Use the service account**, and then click **Next**.
10. On the **Completing the Wizard** page, accept the default name, **Adventure Works DW 2012**, and then click **Finish** to create the new data source.

Note
To modify the properties of the data source after it has been created, double-

click the data source in the **Data Sources** folder to display the data source properties in **Data Source Designer**.

Next Task in Lesson

Defining a Data Source View

See Also

Defining a Data Source using the Data Source Wizard

Defining a Data Source View

After you define the data sources that you will use in an Analysis Services project, the next step is generally to define a data source view for the project. A data source view is a single, unified view of the metadata from the specified tables and views that the data source defines in the project. Storing the metadata in the data source view enables you to work with the metadata during development without an open connection to any underlying data source. For more information, see Working with Data Source Views (SSAS).

In the following task, you define a data source view that includes five tables from the **AdventureWorksDW2012** data source.

Procedures

To define a new data source view

1. In Solution Explorer (on the right of the Microsoft Visual Studio window), right-click **Data Source Views**, and then click **New Data Source View**.
2. On the **Welcome to the Data Source View Wizard** page, click **Next**. The **Select a Data Source** page appears.
3. Under **Relational data sources**, the **Adventure Works DW 2012** data source is selected. Click **Next**.

   Note
   To create a data source view that is based on multiple data sources, first define a data source view that is based on a single data source. This data source is then called the primary data source. You can then add tables and views from a secondary data source. When designing dimensions that contain attributes based on related tables in multiple data sources, you might need to define a Microsoft SQL Server data source as the primary data source to use its distributed query engine capabilities.
4. On the **Select Tables and Views** page, select tables and views from the list of objects that are available from the selected data source. You can filter this list to help you select tables and views.

Note
Click the maximize button in the upper-right corner so that the window covers the full screen. This makes it easier to see the complete list of available objects.

In the **Available objects** list, select the following objects. You can select multiple tables by clicking each while holding down the CTRL key:

- **DimCustomer (dbo)**
- **DimDate (dbo)**
- **DimGeography (dbo)**
- **DimProduct (dbo)**
- **FactInternetSales (dbo)**

5. Click **>** to add the selected tables to the **Included objects** list.
6. Click **Next.**
7. In the Name field, make sure **Adventure Works DW 2012** displays, and then click **Finish**.

   The **Adventure Works DW 2012** data source view appears in the **Data Source Views** folder in Solution Explorer. The content of the data source view is also displayed in Data Source View Designer in SQL Server Data Tools (SSDT). This designer contains the following elements:

   - A **Diagram** pane in which the tables and their relationships are represented graphically.
   - A **Tables** pane in which the tables and their schema elements are displayed in a tree view.
   - A **Diagram Organizer** pane in which you can create subdiagrams so that you can view subsets of the data source view.
   - A toolbar that is specific to Data Source View Designer.
8. To maximize the Microsoft Visual Studio development environment, click the **Maximize** button.
9. To view the tables in the **Diagram** pane at 50 percent, click the **Zoom** icon on the Data Source View Designer toolbar. This will hide the column details of each table.
10. To hide Solution Explorer, click the **Auto Hide** button, which is the pushpin icon on the title bar. To view Solution Explorer again, position your pointer over the Solution Explorer tab along the right side of the development environment. To unhide Solution Explorer, click the **Auto Hide** button again.
11. If the windows are not hidden by default, click **Auto Hide** on the title bar of the Properties and Solution Explorer windows.

    You can now view all the tables and their relationships in the **Diagram** pane.

Notice that there are three relationships between the FactInternetSales table and the DimDate table. Each sale has three dates associated with the sale: an order date, a due date, and a ship date. To view the details of any relationship, double-click the relationship arrow in the **Diagram** pane.

Next Task in Lesson

Modifying Default Table Names

See Also

Working with Data Source Views (SSAS)

Modifying Default Table Names

You can change the value of the **FriendlyName** property for objects in the data source view to make them easier to notice and use.

In the following task, you will change the friendly name of each table in the data source view by removing the "**Dim**" and "**Fact**" prefixes from these tables. This will make the cube and dimension objects (that you will define in the next lesson) easier to notice and use.

Note
You can also change the friendly names of columns, define calculated columns, and join tables or views in the data source view to make them easier to use.

Procedures

To modify the default name of a table

1. In the **Tables** pane of **Data Source View Designer**, right-click the **FactInternetSales** table, and then click **Properties**.
2. If the Properties window on the right side of the Microsoft Visual Studio window is not displayed, click the **Auto Hide** button on the title bar of the Properties window so that this window remains visible.

   It is easier to change the properties for each table in the data source view when the Properties window remains open. If you do not pin the window open by using the **Auto Hide** button, the window will close when you click a different object in the **Diagram** pane.
3. Change the **FriendlyName** property for the **FactInternetSales** object to **InternetSales**.

   When you click away from the cell for the **FriendlyName** property, the change is applied. In the next lesson, you will define a measure group that is based on this fact table. The name of the fact table will be InternetSales instead of FactInternetSales because of the change you made in this lesson.

4. Click **DimProduct** in the **Tables** pane. In the Properties window, change the **FriendlyName** property to **Product**.
5. Change the **FriendlyName** property of each remaining table in the data source view in the same way, to remove the "**Dim**" prefix.
6. When you have finished, click the **Auto Hide** button to hide the Properties window again.
7. On the **File** menu, or on the toolbar of SQL Server Data Tools, click **Save All** to save the changes you have made to this point in the Analysis Services Tutorial project. You can stop the tutorial here if you want and resume it later.

Next Lesson

Lesson 2: Defining and Deploying a Cube

See Also

Designing Data Source Views (SSAS)

Changing Properties in a Data Source View

Lesson 2: Defining and Deploying a Cube

After you define a data source view in your Microsoft Analysis Services project, you are ready to define an initial Analysis Services cube.

You can define a cube and its dimensions in a single pass using the Cube Wizard. Alternatively, you can define one or more dimensions and then use the Cube Wizard to define a cube that uses those dimensions. If you are designing a complex solution, you generally start by defining the dimensions. For more information, see Designing Dimensions or Designing Cubes.

Note
Completed projects for all of the lessons in this tutorial are available online. You can jump ahead to any lesson by using the completed project from the previous lesson as a starting point. Click here to download the sample projects that go with this tutorial.

This lesson contains the following tasks:

Defining a Dimension

In this task, you use the Dimension Wizard to define a dimension.

Defining a Cube

In this task, you use the Cube Wizard to define an initial Analysis Services cube.

Adding Attributes to Dimensions

In this task, you add attributes to the dimensions that you created.

Reviewing Cube and Dimension Properties

In this task, you review the structure of the cube that you defined by using the Cube Wizard.

Deploying an Analysis Services Project

In this task, you deploy the Analysis Services project to your local instance of Analysis Services, and learn about certain deployment properties.

Browsing the Cube

In this task, you browse the cube and dimension data by using Excel or the MDX query designer.

Next Lesson

Lesson 3: Modifying Measures, Attributes and Hierarchies

See Also

Analysis Services Tutorial Scenario
SQL Server Analysis Services Tutorial
Defining and Configuring Dimensions, Attributes and Hierarchies
Defining and Configuring Cubes and Cube Properties
Configuring Analysis Services Project Properties
Building Analysis Services Projects
Deploying Analysis Services Projects

Defining a Dimension

In the following task, you will use the Dimension Wizard to build a Date dimension.

Note
This lesson requires that you have completed all the procedures in Lesson 1.

Procedures

To define a dimension

1. In Solution Explorer (on the right side of Microsoft Visual Studio), right-click **Dimensions**, and then click **New Dimension**. The Dimension Wizard appears.
2. On the **Welcome to the Dimension Wizard** page, click **Next**.
3. On the **Select Creation Method** page, verify that the **Use an existing table** option is selected, and then click **Next**.
4. On the **Specify Source Information** page, verify that the **Adventure Works DW 2012** data source view is selected.

5. In the **Main table** list, select **Date**.
6. Click **Next**.
7. On the **Select Dimension Attributes** page, select the check boxes next to the following attributes:
   - **Date Key**
   - **Full Date Alternate Key**
   - **English Month Name**
   - **Calendar Quarter**
   - **Calendar Year**
   - **Calendar Semester**
8. Change the setting of the **Full Date Alternate Key** attribute's **Attribute Type** column from **Regular** to **Date**. To do this, click **Regular** in the **Attribute Type** column. Then click the arrow to expand the options. Next, click **Date** > **Calendar** > **Date**. Click **OK**. Repeat these steps to change the attribute type of the attributes as follows:
   - **English Month Name** to **Month**
   - **Calendar Quarter** to **Quarter**
   - **Calendar Year** to **Year**
   - **Calendar Semester** to **Half Year**
9. Click **Next**.
10. On the **Completing the Wizard** page, in the Preview pane, you can see the **Date** dimension and its attributes.
11. Click **Finish** to complete the wizard.

    In Solution Explorer, in the Analysis Services Tutorial project, the Date dimension appears in the **Dimensions** folder. In the center of the development environment, Dimension Designer displays the Date dimension.
12. On the **File** menu, click **Save All**.

Next Task in Lesson

Defining a Cube

See Also

Designing Dimensions

Creating a Dimension by Using an Existing Table

How to: Create a Dimension Using the Dimension Wizard

Defining a Cube

The Cube Wizard helps you define the measure groups and dimensions for a cube. In the following task, you will use the Cube Wizard to build a cube.

Procedures

To define a cube and its properties

1. In Solution Explorer, right-click **Cubes**, and then click **New Cube**. The Cube Wizard appears.
2. On the **Welcome to the Cube Wizard** page, click **Next**.
3. On the **Select Creation Method** page, verify that the **Use existing tables** option is selected, and then click **Next**.
4. On the **Select Measure Group Tables** page, verify that the **Adventure Works DW 2012** data source view is selected.
5. Click **Suggest** to have the cube wizard suggest tables to use to create measure groups.

   The wizard examines the tables and suggests **InternetSales** as a measure group table. Measure group tables, also called fact tables, contain the measures you are interested in, such as the number of units sold.
6. Click **Next**.
7. On the **Select Measures** page, review the selected measures in the **Internet Sales** measure group, and then clear the check boxes for the following measures:
   - **Promotion Key**
   - **Currency Key**
   - **Sales Territory Key**
   - **Revision Number**

   By default, the wizard selects as measures all numeric columns in the fact table that are not linked to dimensions. However, these four columns are not actual measures. The first three are key values that link the fact table with dimension tables that are not used in the initial version of this cube.
8. Click **Next**.
9. On the **Select Existing Dimensions** page, make sure the **Date** dimension that you created earlier is selected, and then click **Next**.
10. On the **Select New Dimensions** page, select the new dimensions to be created. To do this, verify that the **Customer**, **Geography**, and **Product** check boxes are selected, and then clear the **InternetSales** check box.
11. Click **Next**.
12. On the **Completing the Wizard** page, change the name of the cube to **Analysis**

Services Tutorial. In the Preview pane, you can see the **InternetSales** measure group and its measures. You can also see the **Date**, **Customer,** and **Product** dimensions.

13. Click **Finish** to complete the wizard.

    In Solution Explorer, in the Analysis Services Tutorial project, the Analysis Services Tutorial cube appears in the **Cubes** folder, and the Customer and Product database dimensions appear in the **Dimensions** folder. Additionally, in the center of the development environment, the Cube Structure tab displays the Analysis Services Tutorial cube.
14. On the toolbar of the Cube Structure tab, change the **Zoom** level to 50 percent, so that you can more easily see the dimensions and fact tables in the cube. Notice that the fact table is yellow and the dimension tables are blue.
15. On the **File** menu, click **Save All**.

Next Task in Lesson

Adding Attributes to Dimensions

See Also

Defining and Configuring Cubes and Cube Properties

Defining and Configuring Dimensions, Attributes and Hierarchies

Adding Attributes to Dimensions

Now that you have defined dimensions, you can populate them with attributes that represent each data element in the dimension. Attributes are commonly based on fields from a data source view. When adding attributes to a dimension, you can include fields from any table in the data source view.

In this task, you will use Dimension Designer to add attributes to the Customer and Product dimensions. The Customer dimension will include attributes based on fields from both the Customer and Geography tables.

Adding Attributes to the Customer Dimension

To add attributes

1. Open Dimension Designer for the Customer dimension. To do this, double-click the **Customer** dimension in the **Dimensions** node of Solution Explorer.
2. In the **Attributes** pane, notice the Customer Key and Geography Key attributes that were created by the Cube Wizard.
3. On the toolbar of the **Dimension Structure** tab, make sure the Zoom icon to view the tables in the **Data Source View** pane is set at 100 percent.
4. Drag the following columns from the **Customer** table in the **Data Source View**

pane to the **Attributes** pane:

- **BirthDate**
- **MaritalStatus**
- **Gender**
- **EmailAddress**
- **YearlyIncome**
- **TotalChildren**
- **NumberChildrenAtHome**
- **EnglishEducation**
- **EnglishOccupation**
- **HouseOwnerFlag**
- **NumberCarsOwned**
- **Phone**
- **DateFirstPurchase**
- **CommuteDistance**

5. Drag the following columns from the **Geography** table in the **Data Source View** pane to the **Attributes** pane:
   - **City**
   - **StateProvinceName**
   - **EnglishCountryRegionName**
   - **PostalCode**
6. On the File menu, click **Save All**.

Adding Attributes to the Product Dimension

To add attributes

1. Open Dimension Designer for the Product dimension. Double-click the **Product** dimension in Solution Explorer.
2. In the **Attributes** pane, notice the Product Key attribute that was created by the Cube Wizard.
3. On the toolbar of the **Dimension Structure** tab, make sure the Zoom icon to view the tables in the **Data Source View** pane is set at 100 percent.
4. Drag the following columns from the **Product** table in the **Data Source View** pane to the **Attributes** pane:
   - **StandardCost**
   - **Color**
   - **SafetyStockLevel**

- **ReorderPoint**
- **ListPrice**
- **Size**
- **SizeRange**
- **Weight**
- **DaysToManufacture**
- **ProductLine**
- **DealerPrice**
- **Class**
- **Style**
- **ModelName**
- **StartDate**
- **EndDate**
- **Status**

5. On the File menu, click **Save All**.

Next Task in Lesson

Reviewing Cube and Dimension Properties

See Also

Configuring Dimension Attribute Properties

Reviewing Cube and Dimension Properties

After you have defined a cube, you can review the results by using Cube Designer. In the following task, you review the structure of the cube in the Analysis Services Tutorial project.

Procedures

To review cube and dimension properties in Cube Designer

1. To open the Cube Designer, double-click the **Analysis Services Tutorial** cube in the **Cubes** node of Solution Explorer.
2. In the **Measures** pane of the **Cube Structure** tab in Cube Designer, expand the **Internet Sales** measure group to reveal the defined measures.

   You can change the order by dragging the measures into the order that you want. The order you create affects how certain client applications order these measures. The measure group and each measure that it contains have properties that you can edit in the Properties window.

3. In the **Dimensions** pane of the **Cube Structure** tab in Cube Designer, review the cube dimensions that are in the Analysis Services Tutorial cube.

   Notice that although only three dimensions were created at the database level, as displayed in Solution Explorer, there are five cube dimensions in the Analysis Services Tutorial cube. The cube contains more dimensions than the database because the Date database dimension is used as the basis for three separate date-related cube dimensions, based on different date-related facts in the fact table. These date-related dimensions are also called *role playing dimensions*. The three date-related cube dimensions let users dimension the cube by three separate facts that are related to each product sale: the product order date, the due date for fulfillment of the order, and the ship date for the order. By reusing a single database dimension for multiple cube dimensions, Analysis Services simplifies dimension management, uses less disk space, and reduces overall processing time.

4. In the **Dimensions** pane of the **Cube Structure** tab, expand **Customer**, and then click **Edit Customer** to open the dimension in Dimension Designer.

   Dimension Designer contains these tabs: **Dimension Structure**, **Attribute Relationships**, **Translations**, and **Browser**. Notice that the **Dimension Structure** tab includes three panes: **Attributes**, **Hierarchies**, and **Data Source View**. The attributes that the dimension contains appear in the **Attributes** pane. For more information, see Defining and Configuring Dimension Attributes, Defining and Configuring a Multilevel Hierarchy, and Defining and Configuring an Attribute Relationship.

5. To switch to Cube Designer, right-click the **Analysis Services Tutorial** cube in the **Cubes** node in Solution Explorer, and then click **View Designer**.

6. In Cube Designer, click the **Dimension Usage** tab.

   In this view of the Analysis Services Tutorial cube, you can see the cube dimensions that are used by the Internet Sales measure group. Also, you can define the type of relationship between each dimension and each measure group in which it is used.

7. Click the **Partitions** tab.

   The Cube Wizard defines a single partition for the cube, by using the multidimensional online analytical processing (MOLAP) storage mode without aggregations. With MOLAP, all leaf-level data and all aggregations are stored within the cube for maximum performance. Aggregations are precalculated summaries of data that improve query response time by having answers ready before questions are asked. You can define additional partitions, storage settings, and writeback settings on the **Partitions** tab. For more information, see Partitions, Aggregations and Aggregation Designs (SSAS), and Designing Partition Storage and Aggregations.

8. Click the **Browser** tab.

Notice that the cube cannot be browsed because it has not yet been deployed to an instance of Analysis Services. At this point, the cube in the Analysis Services Tutorial project is just a definition of a cube, which you can deploy to any instance of Analysis Services. When you deploy and process a cube, you create the defined objects in an instance of Analysis Services and populate the objects with data from the underlying data sources.

9. In Solution Explorer, right-click **Analysis Services Tutorial** in the **Cubes** node, and then click **View Code**. You might need to wait.

   The XML code for the Analysis Services Tutorial cube is displayed on the **Analysis Services Tutorial.cube [XML]** tab. This is the actual code that is used to create the cube in an instance of Analysis Services during deployment. For more information, see How to: View the XML Code for an Analysis Services Project.

10. Close the XML code tab.

Next Task in Lesson

Deploying an Analysis Services Project

See Also

How to: Browse Dimension Data in Dimension Designer

Deploying an Analysis Services Project

To view the cube and dimension data for the objects in the Analysis Services Tutorial cube in the Analysis Services Tutorial project, you must deploy the project to a specified instance of Analysis Services and then process the cube and its dimensions. *Deploying* an Analysis Services project creates the defined objects in an instance of Analysis Services. *Processing* the objects in an instance of Analysis Services copies the data from the underlying data sources into the cube objects. For more information, see Deploying a Project and Configuring Project Properties.

At this point in the development process, you generally deploy the cube to an instance of Analysis Services on a development server. Once you have finished developing your business intelligence project, you will generally use the Analysis Services Deployment Wizard to deploy your project from the development server to a production server. For more information, see Planning an Analysis Services Deployment and Using the Analysis Services Deployment Wizard.

In the following task, you review the deployment properties of the Analysis Services Tutorial project and then deploy the project to your local instance of Analysis Services.

Procedures

To deploy the Analysis Services project

1. In Solution Explorer, right-click the **Analysis Services Tutorial** project, and then

click **Properties**.

The **Analysis Services Tutorial Property Pages** dialog box appears and displays the properties of the Active(Development) configuration. You can define multiple configurations, each with different properties. For example, a developer might want to configure the same project to deploy to different development computers and with different deployment properties, such as database names or processing properties. Notice the value for the **Output Path** property. This property specifies the location in which the XMLA deployment scripts for the project are saved when a project is built. These are the scripts that are used to deploy the objects in the project to an instance of Analysis Services.

2. In the **Configuration Properties** node in the left pane, click **Deployment**.

   Review the deployment properties for the project. By default, the Analysis Services Project template configures an Analysis Services project to incrementally deploy all projects to the default instance of Analysis Services on the local computer, to create an Analysis Services database with the same name as the project, and to process the objects after deployment by using the default processing option. For more information, see Configuring Project Properties.

   Note
   If you want to deploy the project to a named instance of Analysis Services on the local computer, or to an instance on a remote server, change the **Server** property to the appropriate instance name, such as *<ServerName>\<InstanceName>*.

3. Click **OK**.

4. In Solution Explorer, right-click the **Analysis Services Tutorial** project, and then click **Deploy**. You might need to wait.

   Note
   If you get errors during deployment, use SQL Server Management Studio to check the database permissions. The account you specified for the data source connection must have a login on the SQL Server instance. Double-click the login to view User Mapping properties. The account must have db_datareader permissions on the **AdventureWorksDW2012** database.

   SQL Server Data Tools (SSDT) builds and then deploys the Analysis Services Tutorial project to the specified instance of Analysis Services by using a deployment script. The progress of the deployment is displayed in two windows: the **Output** window and the **Deployment Progress – Analysis Services Tutorial** window.

   Open the Output window, if necessary, by clicking **Output** on the **View** menu. The **Output** window displays the overall progress of the deployment. The **Deployment Progress – Analysis Services Tutorial** window displays the detail about each step taken during deployment. For more information, see Building a

Project and Deploying a Project.

5. Review the contents of the **Output** window and the **Deployment Progress – Analysis Services Tutorial** window to verify that the cube was built, deployed, and processed without errors.
6. To hide the **Deployment Progress - Analysis Services Tutorial** window, click the **Auto Hide** icon (it looks like a pushpin) on the toolbar of the window.
7. To hide the **Output** window, click the **Auto Hide** icon on the toolbar of the window.

You have successfully deployed the Analysis Services Tutorial cube to your local instance of Analysis Services, and then processed the deployed cube.

Next Task in Lesson

Browsing the Cube

See Also

Deploying a Project

Configuring Project Properties

Browsing the Cube

After you deploy a cube, the cube data is viewable on the **Browser** tab in Cube Designer, and the dimension data is viewable on the **Browser** tab in Dimension Designer. Browsing cube and dimension data is way to check your work incrementally. You can verify that small changes to properties, relationships, and other objects have the desired effect once the object is processed. While the Browser tab is used to view both cube and dimension data, the tab provides different capabilities based on the object you are browsing.

For dimensions, the Browser tab provides a way to view members or navigate a hierarchy all the way down to the leaf node. You can browse dimension data in different languages, assuming you have added the translations to your model.

For cubes, the Browser tab provides two approaches for exploring data. You can use the built-in MDX Query Designer to build queries that return a flattened rowset from a multidimensional database. Alternatively, you can use an Excel shortcut. When you start Excel from within SQL Server Data Tools, Excel opens with a PivotTable already in the worksheet and a predefined connection to the model workspace database.

Excel generally offers a better browsing experience because you can explore cube data interactively, using horizontal and vertical axes to analyze the relationships in your data. In contrast, the MDX Query Designer is limited to a single axis. Moreover, because the rowset is flattened, you do not get the drilldown that an Excel PivotTable provides. As you add more dimensions and hierarchies to your cube, which you will do in subsequent lessons, Excel will be the preferred solution for browsing data.

Procedures

To browse the deployed cube

1. Switch to **Dimension Designer** for the Product dimension in SQL Server Data Tools (SSDT). To do this, double-click the **Product** dimension in the **Dimensions** node of Solution Explorer.
2. Click the **Browser** tab to display the **All** member of the **Product Key** attribute hierarchy. In lesson three, you will define a user hierarchy for the Product dimension that will let you browse the dimension.
3. Switch to **Cube Designer** in SQL Server Data Tools (SSDT). To do this, double-click the **Analysis Services Tutorial** cube in the **Cubes** node of Solution Explorer.
4. Select the **Browser** tab, and then click the **Reconnect** icon on the toolbar of the designer.

   The left pane of the designer shows the objects in the Analysis Services Tutorial cube. On the right side of the **Browser** tab, there are two panes: the upper pane is the **Filter** pane, and the lower pane is the **Data** pane. In an upcoming lesson, you will use the cube browser to do analysis.

Next Lesson

Lesson 3: Modifying Measures, Attributes and Hierarchies

See Also

MDX Query Editor (Analysis Services - Multidimensional Data)

Lesson 3: Modifying Measures, Attributes and Hierarchies

After defining your initial cube, you are ready to improve the usefulness and friendliness of the cube. You can do this by adding hierarchies that support navigation and aggregation at various levels, by applying formats to specific measure, and by defining calculations and relationships.

Note

Completed projects for all of the lessons in this tutorial are available online. You can jump ahead to any lesson by using the completed project from the previous lesson as a starting point. Click here to download the sample projects that go with this tutorial.

This lesson contains the following tasks:

Modifying Measures

In this task, you specify formatting properties for the currency and percentage measures

in the Analysis Services Tutorial cube.

Modifying the Customer Dimension

In this task, you define a user hierarchy, create named calculations, modify attributes to use named calculations, and group attributes and user hierarchies into display folders.

Modifying the Product Dimension

In this task, you define a user hierarchy, create named calculations, define the All member name, and define display folders.

Modifying the Time Dimension

In this task, you define a user hierarchy, modify attribute member names, and use composite keys to specify unique attribute members.

Browsing the Deployed Cube

In this task, you browse cube data by using the browser in Cube Designer.

See Also

Analysis Services Tutorial Scenario

Analysis Services Tutorial

Modifying Measures

You can use the **FormatString** property to define formatting settings that control how measures are displayed to users. In this task, you specify formatting properties for the currency and percentage measures in the Analysis Services Tutorial cube.

Procedures

To modify the measures of the cube

1. Switch to the **Cube Structure** tab of Cube Designer for the Analysis Services Tutorial cube, expand the **Internet Sales** measure group in the **Measures** pane, right-click **Order Quantity**, and then click **Properties**.
2. In the Properties window, click the **Auto Hide** pushpin icon to pin the Properties window open.

   It is easier to change properties for several items in the cube when the Properties window remains open.
3. In the Properties window, click the **FormatString** list, and then type **#,#**.
4. On the toolbar of the **Cube Structure** tab, click the **Show Measures Grid** icon on the left.

   The grid view lets you select multiple measures at the same time.
5. Select the following measures. You can select multiple measures by clicking each

while holding down the CTRL key:

- **Unit Price**
- **Extended Amount**
- **Discount Amount**
- **Product Standard Cost**
- **Total Product Cost**
- **Sales Amount**
- **Tax Amt**
- **Freight**

6. In the Properties window, in the **FormatString** list, select **Currency**.
7. In the drop-down list at the top of the Properties window (right below the title bar), select the measure **Unit Price Discount Pct**, and then select **Percent** in the **FormatString** list.
8. In the Properties window, change the **Name** property for the **Unit Price Discount Pct** measure to **Unit Price Discount Percentage**.
9. In the **Measures** pane, click **Tax Amt** and change the name of this measure to **Tax Amount**.
10. In the Properties window, click the **Auto Hide** icon to hide the Properties window, and then click **Show Measures Tree** on the toolbar of the **Cube Structure** tab.
11. On the **File** menu, click **Save All**.

Next Task in Lesson

Modifying the Customer Dimension

See Also

Define Database Dimensions

Configuring Measure Properties

Modifying the Customer Dimension

There are many different ways that you can increase the usability and functionality of the dimensions in a cube. In the tasks in this topic, you modify the Customer dimension.

Renaming Attributes

You can change attribute names with the **Dimension Structure** tab of Dimension Designer.

To rename an attribute

1. Switch to **Dimension Designer** for the Customer dimension in SQL Server Data Tools (SSDT). To do this, double-click the **Customer** dimension in the **Dimensions** node of Solution Explorer.
2. In the **Attributes** pane, right-click **English Country Region Name**, and then click **Rename**. Change the name of the attribute to **Country-Region**.
3. Change the names of the following attributes in the same manner:
   - **English Education** attribute — change to **Education**
   - **English Occupation** attribute — change to **Occupation**
   - **State Province Name** attribute — change to **State-Province**
4. On the **File** menu, click **Save All**.

Creating a Hierarchy

You can create a new hierarchy by dragging an attribute from the **Attributes** pane to the **Hierarchies** pane.

To create a hierarchy

1. Drag the **Country-Region** attribute from the **Attributes** pane into the **Hierarchies** pane.
2. Drag the **State-Province** attribute from the **Attributes** pane into the **<new level>** cell in the **Hierarchies** pane, underneath the **Country-Region** level.
3. Drag the **City** attribute from the **Attributes** pane into the **<new level>** cell in the **Hierarchies** pane, underneath the **State-Province** level.
4. In the **Hierarchies** pane of the **Dimension Structure** tab, right-click the title bar of the **Hierarchy** hierarchy, select **Rename**, and then type **Customer Geography**.

   The name of the hierarchy is now **Customer Geography**.
5. On the **File** menu, click **Save All**.

Adding a Named Calculation

You can add a named calculation, which is a SQL expression that is represented as a calculated column, to a table in a data source view. The expression appears and behaves as a column in the table. Named calculations let you extend the relational schema of existing tables in a data source view without modifying the table in the underlying data source. For more information, see Creating Named Calculations in a Data Source View (SSAS)

To add a named calculation

1. Open the **Adventure Works DW 2012** data source view by double-clicking it in the **Data Source Views** folder in Solution Explorer.
2. In the **Tables** pane on the left, right-click **Customer**, and then click **New Named**

Calculation.

3. In the **Create Named Calculation** dialog box, type **FullName** in the **Column name** box, and then type or copy and paste the following **CASE** statement in the **Expression** box:

```
CASE
   WHEN MiddleName IS NULL THEN
   FirstName + ' ' + LastName
   ELSE
   FirstName + ' ' + MiddleName + ' ' + LastName
END
```

   The **CASE** statement concatenates the **FirstName**, **MiddleName**, and **LastName** columns into a single column that you will use in the Customer dimension as the displayed name for the **Customer** attribute.

4. Click **OK**, and then expand **Customer** in the **Tables** pane.

   The **FullName** named calculation appears in the list of columns in the Customer table, with an icon that indicates that it is a named calculation.

5. On the **File** menu, click **Save All**.
6. In the **Tables** pane, right-click **Customer**, and then click **Explore Data**.
7. Review the last column in the **Explore Customer Table** view.

   Notice that the **FullName** column appears in the data source view, correctly concatenating data from several columns from the underlying data source and without modifying the original data source.

8. Close the **Explore Customer Table** tab.

Using the Named Calculation for Member Names

After you have created a named calculation in the data source view, you can use the named calculation as a property of an attribute.

To use the named calculation for member names

1. Switch to Dimension Designer for the Customer dimension.
2. In the **Attributes** pane of the **Dimension Structure** tab, click the **Customer Key** attribute.
3. Open the Properties window and click the **Auto Hide** button on the title bar so that it stays open.
4. In the **Name** property field, type **Full Name**.
5. Click in the **NameColumn** property field at the bottom, and then click the browse (**...**) button to open the **Name Column** dialog box.

6. Select **FullName** at the bottom of the **Source column** list, and then click **OK**.
7. In the Dimensions Structure tab, drag the **Full Name** attribute from the **Attributes** pane into the **<new level>** cell in the **Hierarchies** pane, underneath the **City** level.
8. On the **File** menu, click **Save All**.

Defining Display Folders

You can use display folders to group user and attribute hierarchies into folder structures to increase usability.

To define display folders

1. Open the **Dimension Structure** tab for the Customer dimension.
2. In the **Attributes** pane, select the following attributes by holding down the CTRL key while clicking each of them:
   - **City**
   - **Country-Region**
   - **Postal Code**
   - **State-Province**
3. In the Properties window, click the **AttributeHierarchyDisplayFolder** property field at the top (you might need to point to it to see the full name), and then type **Location**.
4. In the **Hierarchies** pane, click **Customer Geography**, and then in the Properties window on the right, select **Location** as the value of the **DisplayFolder** property.
5. In the **Attributes** pane, select the following attributes by holding down the CTRL key while clicking each of them:
   - **Commute Distance**
   - **Education**
   - **Gender**
   - **House Owner Flag**
   - **Marital Status**
   - **Number Cars Owned**
   - **Number Children At Home**
   - **Occupation**
   - **Total Children**
   - **Yearly Income**
6. In the Properties window, click the **AttributeHierarchyDisplayFolder** property field at the top, and then type **Demographic**.
7. In the **Attributes** pane, select the following attributes by holding down the CTRL

key while clicking each of them:

- **Email Address**
- **Phone**

8. In the Properties window, click the **AttributeHierarchyDisplayFolder** property field and type **Contacts**.
9. On the **File** menu, click **Save All**.

Defining Composite KeyColumns

The **KeyColumns** property contains the column or columns that represent the key for the attribute. In this lesson, you create a composite key for the **City** and **State-Province** attributes. Composite keys can be helpful when you need to uniquely identify an attribute. For example, when you define attribute relationships later in this tutorial, a **City** attribute must uniquely identify a **State-Province** attribute. However, there could be several cities with the same name in different states. For this reason, you will create a composite key that is composed of the **StateProvinceName** and **City** columns for the **City** attribute. For more information, see How to: Modify the KeyColumn Property of an Attribute.

To define composite KeyColumns for the City attribute

1. Open the **Dimension Structure** tab for the Customer dimension.
2. In the **Attributes** pane, click the **City** attribute.
3. In the **Properties** window, click in the **KeyColumns** field near the bottom, and then click the browse (**...**) button.
4. In the **Key Columns** dialog box, in the **Available Columns** list, select the column **StateProvinceName**, and then click the **>** button.

   The **City** and **StateProvinceName** columns are now displayed in the **Key Columns** list.
5. Click **OK**.
6. To set the **NameColumn** property of the **City** attribute, click the **NameColumn** field in the Properties window, and then click the browse (**...**) button.
7. In the **Name Column** dialog box, in the **Source column** list, select **City**, and then click **OK**.
8. On the **File** menu, click **Save All**.

To define composite KeyColumns for the State-Province attribute

1. Make sure the **Dimension Structure** tab for the Customer dimension is open.
2. In the **Attributes** pane, click the **State-Province** attribute.
3. In the **Properties** window, click in the **KeyColumns** field, and then click the browse (**...**) button.

4. In the **Key Columns** dialog box, in the **Available Columns** list, select the column **EnglishCountryRegionName**, and then click the **>** button.

   The **EnglishCountryRegionName** and **StateProvinceName** columns are now displayed in the **Key Columns** list.
5. Click **OK**.
6. To set the **NameColumn** property of the **State-Province** attribute, click the **NameColumn** field in the Properties window, and then click the browse (**...**) button.
7. In the **Name Column** dialog box, in the **Source column** list, select **StateProvinceName**, and then click **OK**.
8. On the **File** menu, click **Save All**.

Defining Attribute Relationships

If the underlying data supports it, you should define attribute relationships between attributes. Defining attribute relationships speeds up dimension, partition, and query processing. For more information, see Defining Attribute Relationships and Attribute Relationships.

To define attribute relationships

1. In the **Dimension Designer** for the Customer dimension, click the **Attribute Relationships** tab. You might need to wait.
2. In the diagram, right-click the **City** attribute, and then click **New Attribute Relationship**.
3. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **City**. Set the **Related Attribute** to **State-Province**.
4. In the **Relationship type** list, set the relationship type to **Rigid**.

   The relationship type is **Rigid** because relationships between the members will not change over time. For example, it would be unusual for a city to become part of a different state or province.
5. Click .
6. In the diagram, right-click the **State-Province** attribute and then select **New Attribute Relationship**.
7. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **State-Province**. Set the **Related Attribute** to **Country-Region**.
8. In the **Relationship type** list, set the relationship type to **Rigid**.
9. Click **OK**.
10. On the **File** menu, click **Save All**.

Deploying Changes, Processing the Objects, and Viewing the Changes

After you have changed attributes and hierarchies, you must deploy the changes and reprocess the related objects before you can view the changes.

To deploy the changes, process the objects, and view the changes

1. On the **Build** menu of SQL Server Data Tools, click **Deploy Analysis Services Tutorial**.
2. After you receive the **Deployment Completed Successfully** message, click the **Browser** tab of Dimension Designer for the Customer dimension, and then click the Reconnect button on the left side of the toolbar of the designer.
3. Verify that **Customer Geography** is selected in the **Hierarchy** list, and then in the browser pane, expand **All**, expand **Australia**, expand **New South Wales**, and then expand **Coffs Harbour**.

   The browser displays the customers in the city.
4. Switch to **Cube Designer** for the Analysis Services Tutorial cube. To do this, double-click the **Analysis Services Tutorial** cube in the **Cubes** node of **Solution Explorer**.
5. Click the **Browser** tab, and then click the Reconnect button on the toolbar of the designer.
6. In the **Measure Group** pane, expand **Customer**.

   Notice that instead of a long list of attributes, only the display folders and the attributes that do not have display folder values appear underneath Customer.
7. On the **File** menu, click **Save All**.

Next Task in Lesson

Modifying the Product Dimension

See Also

Configuring Dimension Attribute Properties

Removing Attributes from a Dimension

Renaming an Attribute

Creating Named Calculations in a Data Source View (SSAS)

Modifying the Product Dimension

In the tasks in this topic, you use a named calculation to provide more descriptive names for the product lines, define a hierarchy in the Product dimension, and specify the (All) member name for the hierarchy. You also group attributes into display folders.

Adding a Named Calculation

You can add a named calculation to a table in a data source view. In the following task, you create a named calculation that displays the full product line name.

To add a named calculation

1. To open the **Adventure Works DW 2012** data source view, double-click **Adventure Works DW 2012** in the **Data Source Views** folder in Solution Explorer.
2. In the bottom of the diagram pane, right-click the **Product** table header, and then click **New Named Calculation**.
3. In the **Create Named Calculation** dialog box, type **ProductLineName** in the **Column name** box.
4. In the **Expression** box, type or copy and paste the following **CASE** statement:

```
CASE ProductLine
   WHEN 'M' THEN 'Mountain'
   WHEN 'R' THEN 'Road'
   WHEN 'S' THEN 'Accessory'
   WHEN 'T' THEN 'Touring'
   ELSE 'Components'
END
```

   This **CASE** statement creates user-friendly names for each product line in the cube.
5. Click **OK** to create the **ProductLineName** named calculation. You might need to wait.
6. On the **File** menu, click **Save All**.

Modifying the NameColumn Property of an Attribute

To modify the NameColumn property value of an attribute

1. Switch to Dimension Designer for the Product dimension. To do this, double-click the **Product** dimension in the **Dimensions** node of Solution Explorer.
2. In the **Attributes** pane of the **Dimension Structure** tab, select **Product Line**.
3. In the Properties window on the right side of the screen, click the **NameColumn** property field at the bottom of the window, and then click the browse (**...**) button to open the **Name Column** dialog box. (You might need to click the **Properties** tab on the right side of the screen to open the Properties window.
4. Select **ProductLineName** at the bottom of the **Source column** list, and then click **OK**.

   The NameColumn field now contains the text, **Product.ProductLineName**

(WChar). The members of the **Product Line** attribute hierarchy now display the full name of the product line instead of an abbreviated product line name.

5. In the **Attributes** pane of the **Dimension Structure** tab, select **Product Key**.
6. In the Properties window, click the **NameColumn** property field, and then click the ellipsis browse (**...**) button to open the **Name Column** dialog box.
7. Select **EnglishProductName** in the **Source column** list, and then click **OK**.

   The NameColumn field now contains the text, **Product.EnglishProductName (WChar)**.
8. In the Properties window, scroll up, click the **Name** property field, and then type **Product Name**.

Creating a Hierarchy

To create a hierarchy

1. Drag the **Product Line** attribute from the **Attributes** pane into the **Hierarchies** pane.
2. Drag the **Model Name** attribute from the **Attributes** pane into the **<new level>** cell in the **Hierarchies** pane, underneath the **Product Line** level.
3. Drag the **Product Name** attribute from the **Attributes** pane into the **<new level>** cell in the **Hierarchies** pane, underneath the **Model Name** level. (You renamed Product Key to Product Name in the previous section.)
4. In the **Hierarchies** pane of the **Dimension Structure** tab, right-click the title bar of the **Hierarchy** hierarchy, click **Rename**, and then type **Product Model Lines**.

   The name of the hierarchy is now **Product Model Lines**.
5. On the **File** menu, click **Save All**.

Specifying Folder Names and All Member Names

To specify the folder and member names

1. In the **Attributes** pane, select the following attributes by holding down the CTRL key while clicking each of them:
   - **Class**
   - **Color**
   - **Days To Manufacture**
   - **Reorder Point**
   - **Safety Stock Level**
   - **Size**
   - **Size Range**

- **Style**
- **Weight**

2. In the **AttributeHierarchyDisplayFolder** property field in the Properties window, type **Stocking**.

   You have now grouped these attributes into a single display folder.
3. In the **Attributes** pane, select the following attributes:
   - **Dealer Price**
   - **List Price**
   - **Standard Cost**
4. In the **AttributeHierarchyDisplayFolder** property cell in the Properties window, type **Financial**.

   You have now grouped these attributes into a second display folder.
5. In the **Attributes** pane, select the following attributes:
   - **End Date**
   - **Start Date**
   - **Status**
6. In the **AttributeHierarchyDisplayFolder** property cell in the Properties window, type **History**.

   You have now grouped these attributes into a third display folder.
7. Select the **Product Model Lines** hierarchy in the **Hierarchies** pane, and then change the **AllMemberName** property in the Properties window to **All Products**.
8. Click an open area of the **Hierarchies** pane, and then change the **AttributeAllMemberName** property at the top of the Properties window to **All Products**.

   Clicking an open area lets you modify properties of the Product dimension itself. You could also click **Product** at the top of the attributes list in the **Attributes** pane.
9. On the **File** menu, click **Save All**.

Defining Attribute Relationships

If the underlying data supports it, you should define attribute relationships between attributes. Defining attribute relationships speeds up dimension, partition, and query processing. For more information, see Defining Attribute Relationships and Attribute Relationships.

To define attribute relationships

1. In the **Dimension Designer** for the Product dimension, click the **Attribute Relationships** tab.

2. In the diagram, right-click the **Model Name** attribute, and then click **New Attribute Relationship**.
3. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Model Name**. Set the **Related Attribute** to **Product Line**.

   In the **Relationship type** list, leave the relationship type set to **Flexible** because relationships between the members might change over time. For example, a product model might eventually be moved to a different product line.
4. Click **OK**.
5. On the **File** menu, click **Save All**.

Reviewing Product Dimension Changes

To review the Product dimension changes

1. On the **Build** menu of SQL Server Data Tools (SSDT), click **Deploy Analysis Services Tutorial**.
2. After you have received the **Deployment Completed Successfully** message, click the **Browser** tab of **Dimension Designer** for the **Product** dimension, and then click the Reconnect button on the toolbar of the designer.
3. Verify that **Product Model Lines** is selected in the **Hierarchy** list, and then expand **All Products**.

   Notice that the name of the **All** member appears as **All Products**. This is because you changed the **AllMemberName** property for the hierarchy to **All Products** earlier in the lesson. Also, the members of the **Product Line** level now have user-friendly names, instead of single-letter abbreviations.

Next Task in Lesson

Modifying the Time Dimension

See Also

Creating Named Calculations in a Data Source View (SSAS)

Defining and Configuring a Multilevel Hierarchy

Configuring the All Level for Attribute Hierarchies

Modifying the Date Dimension

In the tasks in this topic, you create a user-defined hierarchy and change the member names that are displayed for the Date, Month, Calendar Quarter, and Calendar Semester attributes. You also define composite keys for attributes, control the sort order of dimension members, and define attribute relationships.

Adding a Named Calculation

You can add a named calculation, which is a SQL expression that is represented as a calculated column, to a table in a data source view. The expression appears and behaves as a column in the table. Named calculations enable you to extend the relational schema of existing tables in a data source view without modifying the table in the underlying data source. For more information, see Creating Named Calculations in a Data Source View (SSAS)

To add a named calculation

1. To open the **Adventure Works DW 2012** data source view, double-click it in the **Data Source Views** folder in Solution Explorer.
2. Near the bottom of the **Tables** pane, right-click **Date**, and then click **New Named Calculation**.
3. In the **Create Named Calculation** dialog box, type **SimpleDate** in the **Column name** box, and then type or copy and paste the following **DATENAME** statement in the **Expression** box:

   ```
   DATENAME(mm, FullDateAlternateKey) + ' ' +
   DATENAME(dd, FullDateAlternateKey) + ', ' +
   DATENAME(yy, FullDateAlternateKey)
   ```

   The **DATENAME** statement extracts the year, month, and day values from the FullDateAlternateKey column. You will use this new column as the displayed name for the FullDateAlternateKey attribute.
4. Click **OK**, and then expand **Date** in the **Tables** pane.

   The **SimpleDate** named calculation appears in the list of columns in the Date table, with an icon that indicates that it is a named calculation.
5. On the **File** menu, click **Save All**.
6. In the **Tables** pane, right-click **Date**, and then click **Explore Data**.
7. Scroll to the right to review the last column in the **Explore Date Table** view.

   Notice that the **SimpleDate** column appears in the data source view, correctly concatenating data from several columns from the underlying data source, without modifying the original data source.
8. Close the **Explore Date Table** view.

Using the Named Calculation for Member Names

After you create a named calculation in the data source view, you can use the named calculation as a property of an attribute.

To use the named calculation for member names

1. Open **Dimension Designer** for the Date dimension in SQL Server Data Tools (SSDT). To do this, double-click the **Date** dimension in the **Dimensions** node of

Solution Explorer.

2. In the **Attributes** pane of the **Dimension Structure** tab, click the **Date Key** attribute.
3. If the Properties window is not open, open the Properties window, and then click the **Auto Hide** button on the title bar so that it stays open.
4. Click the **NameColumn** property field near the bottom of the window, and then click the ellipsis browse (**...**) button to open the **Name Column** dialog box.
5. Select **SimpleDate** at the bottom of the **Source column** list, and then click **OK**.
6. On the **File** menu, click **Save All**.

Creating a Hierarchy

You can create a new hierarchy by dragging an attribute from the **Attributes** pane to the **Hierarchies** pane.

To create a hierarchy

1. In **Dimension Structure** tab of the Dimension Designer for the **Date** dimension, drag the **Calendar Year** attribute from the **Attributes** pane into the **Hierarchies** pane.
2. Drag the **Calendar Semester** attribute from the **Attributes** pane into the **<new level>** cell in the **Hierarchies** pane, underneath the **Calendar Year** level.
3. Drag the **Calendar Quarter** attribute from the **Attributes** pane into the **<new level>** cell in the **Hierarchies** pane, underneath the **Calendar Semester** level.
4. Drag the **English Month Name** attribute from the **Attributes** pane into the **<new level>** cell in the **Hierarchies** pane, underneath the **Calendar Quarter** level.
5. Drag the **Date Key** attribute from the **Attributes** pane into the **<new level>** cell in the **Hierarchies** pane, underneath the **English Month Name** level.
6. In the **Hierarchies** pane, right-click the title bar of the **Hierarchy** hierarchy, cick **Rename**, and then type **Calendar Date**.
7. By using the right-click context menu, in the **Calendar Date** hierarchy, rename the **English Month Name** level to **Calendar Month**, and then rename the **Date Key** level to **Date**.
8. Delete the **Full Date Alternate Key** attribute from the **Attributes** pane because you will not be using it. Click **OK** in the **Delete Objects** confirmation window.
9. On the **File** menu, click **Save All**.

Defining Attribute Relationships

If the underlying data supports it, you should define attribute relationships between attributes. Defining attribute relationships speeds up dimension, partition, and query processing.

To define attribute relationships

1. In the **Dimension Designer** for the **Date** dimension, click the **Attribute Relationships** tab.
2. In the diagram, right-click the **English Month Name** attribute, and then click **New Attribute Relationship**.
3. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **English Month Name**. Set the **Related Attribute** to **Calendar Quarter**.
4. In the **Relationship type** list, set the relationship type to **Rigid**.

   The relationship type is **Rigid** because relationships between the members will not change over time.
5. Click **OK**.
6. In the diagram, right-click the **Calendar Quarter** attribute, and then click **New Attribute Relationship**.
7. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Calendar Quarter**. Set the **Related Attribute** to **Calendar Semester**.
8. In the **Relationship type** list, set the relationship type to **Rigid**.
9. Click **OK**.
10. In the diagram, right-click the **Calendar Semester** attribute, and then click **New Attribute Relationship**.
11. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Calendar Semester**. Set the **Related Attribute** to **Calendar Year**.
12. In the **Relationship type** list, set the relationship type to **Rigid**.
13. Click **OK**.
14. On the **File** menu, click **Save All**.

Providing Unique Dimension Member Names

In this task, you will create user-friendly name columns that will be used by the **EnglishMonthName**, **CalendarQuarter**, and **CalendarSemester** attributes.

To provide unique dimension member names

1. To switch to the **Adventure Works DW 2012** data source view, double-click it in the **Data Source Views** folder in Solution Explorer.
2. In the **Tables** pane, right-click **Date**, and then click **New Named Calculation**.
3. In the **Create Named Calculation** dialog box, type **MonthName** in the **Column name** box, and then type or copy and paste the following statement in the **Expression** box:

```
EnglishMonthName+' '+ CONVERT(CHAR (4), CalendarYear)
```

The statement concatenates the month and year for each month in the table into a new column.

4. Click **OK**.
5. In the **Tables** pane, right-click **Date**, and then click **New Named Calculation**.
6. In the **Create Named Calculation** dialog box, type **CalendarQuarterDesc** in the **Column name** box, and then type or copy and paste the following SQL script in the **Expression** box:

```
'Q' + CONVERT(CHAR (1), CalendarQuarter) +' '+ 'CY ' +
CONVERT(CHAR (4), CalendarYear)
```

   This SQL script concatenates the calendar quarter and year for each quarter in the table into a new column.
7. Click **OK**.
8. In the **Tables** pane, right-click **Date**, and then click **New Named Calculation**.
9. In the **Create Named Calculation** dialog box, type **CalendarSemesterDesc** in the **Column name** box, and then type or copy and paste the following SQL script in the **Expression** box:

```
CASE
WHEN CalendarSemester = 1 THEN 'H1' + ' ' + 'CY' + ' '
       + CONVERT(CHAR(4), CalendarYear)
ELSE
'H2' + ' ' + 'CY' + ' ' + CONVERT(CHAR(4), CalendarYear)
END
```

   This SQL script concatenates the calendar semester and year for each semester in the table into a new column.
10. Click **OK.**
11. On the **File** menu, click **Save All**.

Defining Composite KeyColumns and Setting the Name Column

The **KeyColumns** property contains the column or columns that represent the key for the attribute. In this task, you will define composite **KeyColumns**.

To define composite KeyColumns for the English Month Name attribute

1. Open the **Dimension Structure** tab for the Date dimension.
2. In the **Attributes** pane, click the **English Month Name** attribute.
3. In the **Properties** window, click the **KeyColumns** field, and then click the browse (**...**) button.
4. In the **Key Columns** dialog box, in the **Available Columns** list, select the column

CalendarYear, and then click the **>** button.

5. The **EnglishMonthName** and **CalendarYear** columns are now displayed in the **Key Columns** list.
6. Click **OK**.
7. To set the **NameColumn** property of the **EnglishMonthName** attribute, click the **NameColumn** field in the Properties window, and then click the browse (**...**) button.
8. In the **Name Column** dialog box, in the **Source Column** list, select **MonthName**, and then click **OK**.
9. On the **File** menu, click **Save All**.

To define composite KeyColumns for the Calendar Quarter attribute

1. In the **Attributes** pane, click the **Calendar Quarter** attribute.
2. In the **Properties** window, click the **KeyColumns** field, and then click the browse (**...**) button.
3. In the **Key Columns** dialog box, in the **Available Columns** list, select the column **CalendarYear**, and then click the **>** button.

   The **CalendarQuarter** and **CalendarYear** columns are now displayed in the **Key Columns** list.
4. Click **OK**.
5. To set the **NameColumn** property of the **Calendar Quarter** attribute, click the **NameColumn** field in the Properties window, and then click the browse (**...**) button.
6. In the **Name Column** dialog box, in the **Source Column** list, select **CalendarQuarterDesc**, and then click **OK**.
7. On the **File** menu, click **Save All**.

To define composite KeyColumns for the Calendar Semester attribute

1. In the **Attributes** pane, click the **Calendar Semester** attribute.
2. In the **Properties** window, click the **KeyColumns** field, and then click the browse (**...**) button.
3. In the **Key Columns** dialog box, in the **Available Columns** list, select the column, **CalendarYear**, and then click the **>** button.

   The **CalendarSemester** and **CalendarYear** columns are now displayed in the **Key Columns** list.
4. Click **OK**.
5. To set the **NameColumn** property of the **Calendar Semester** attribute, click the **NameColumn** field in the property window, and then click the browse (**...**)

button.

6. In the **Name Column** dialog box, in the **Source Column** list, select **CalendarSemesterDesc**, and then click **OK**.
7. On the **File** menu, click **Save All**.

Deploying and Viewing the Changes

After you have changed attributes and hierarchies, you must deploy the changes and reprocess the related objects before you can view the changes.

To deploy and view the changes

1. On the **Build** menu of SQL Server Data Tools, click **Deploy Analysis Services Tutorial**.
2. After you have received the **Deployment Completed Successfully** message, click the **Browser** tab of **Dimension Designer** for the **Date** dimension, and then click the Reconnect button on the toolbar of the designer.
3. Select **Calendar Quarter** from the **Hierarchy** list. Review the members in the **Calendar Quarter** attribute hierarchy.

   Notice that the names of the members of the **Calendar Quarter** attribute hierarchy are clearer and easier to use because you created a named calculation to use as the name. Members now exist in the **Calendar Quarter** attribute hierarchy for each quarter in each year. The members are not sorted in chronological order. Instead they are sorted by quarter and then by year. In the next task in this topic, you will modify this behavior to sort the members of this attribute hierarchy by year and then by quarter.
4. Review the members of the **English Month Name** and **Calendar Semester** attribute hierarchies.

   Notice that the members of these hierarchies are also not sorted in chronological order. Instead, they are sorted by month or semester, respectively, and then by year. In the next task in this topic, you will modify this behavior to change this sort order.

Changing the Sort Order by Modifying Composite Key Member Order

In this task, you will change the sort order by changing the order of the keys that make up the composite key.

To modify the composite key member order

1. Open the **Dimension Structure** tab of Dimension Designer for the **Date** dimension, and then select **Calendar Semester** in the **Attributes** pane.
2. In the Properties window, review the value for the **OrderBy** property. It is set to **Key**.

The members of the **Calendar Semester** attribute hierarchy are sorted by their key value. With a composite key, the ordering of the member keys is based first on the value of the first member key, and then on the value of the second member key. In other words, the members of the **Calendar Semester** attribute hierarchy are sorted first by semester and then by year.

3. In the Properties window, click the ellipsis browse button (**...**) to change the **KeyColumns** property value.
4. In the **Key Columns** list of the **Key Columns** dialog box, verify that **CalendarSemester** is selected, and then click the down arrow to reverse the order of the members of this composite key. Click **OK**.

   The members of the attribute hierarchy are now sorted first by year and then by semester.
5. Select **Calendar Quarter** in the **Attributes** pane, and then click the ellipsis browse button (**...**) for the **KeyColumns** property in the Properties window.
6. In the **Key Columns** list of the **Key Columns** dialog box, verify that **CalendarQuarter** is selected, and then click the down arrow to reverse the order of the members of this composite key. Click **OK**.

   The members of the attribute hierarchy are now sorted first by year and then by quarter.
7. Select **English Month Name** in the **Attributes** pane, and then click the ellipsis button (**...**) for the **KeyColumns** property in the Properties window.
8. In the **Key Columns** list of the **Key Columns** dialog box, verify that **EnglishMonthName** is selected, and then click the down arrow to reverse the order of the members of this composite key. Click **OK**.

   The members of the attribute hierarchy are now sorted first by year and then by month.
9. On the **Build** menu of SQL Server Data Tools, click **Deploy Analysis Services Tutorial**. When deployment has successfully completed, click the **Browser** tab in Dimension Designer for the **Date** dimension.
10. On the toolbar of the **Browser** tab, click the Reconnect button.
11. Review the members of the **Calendar Quarter** and **Calendar Semester** attribute hierarchies.

    Notice that the members of these hierarchies are now sorted in chronological order, by year and then by quarter or semester, respectively.
12. Review the members of the **English Month Name** attribute hierarchy.

    Notice that the members of the hierarchy are now sorted first by year and then alphabetically by month. This is because the data type for the EnglishCalendarMonth column in the data source view is a string column, based on the nvarchar data type in the underlying relational database. For information about how to enable the months to be sorted chronologically within each year,

see Sorting Attribute Members Based on a Secondary Attribute.

Next Task in Lesson

Browsing the Deployed Cube

See Also

Defining and Configuring Dimensions, Attributes and Hierarchies

Browsing the Deployed Cube

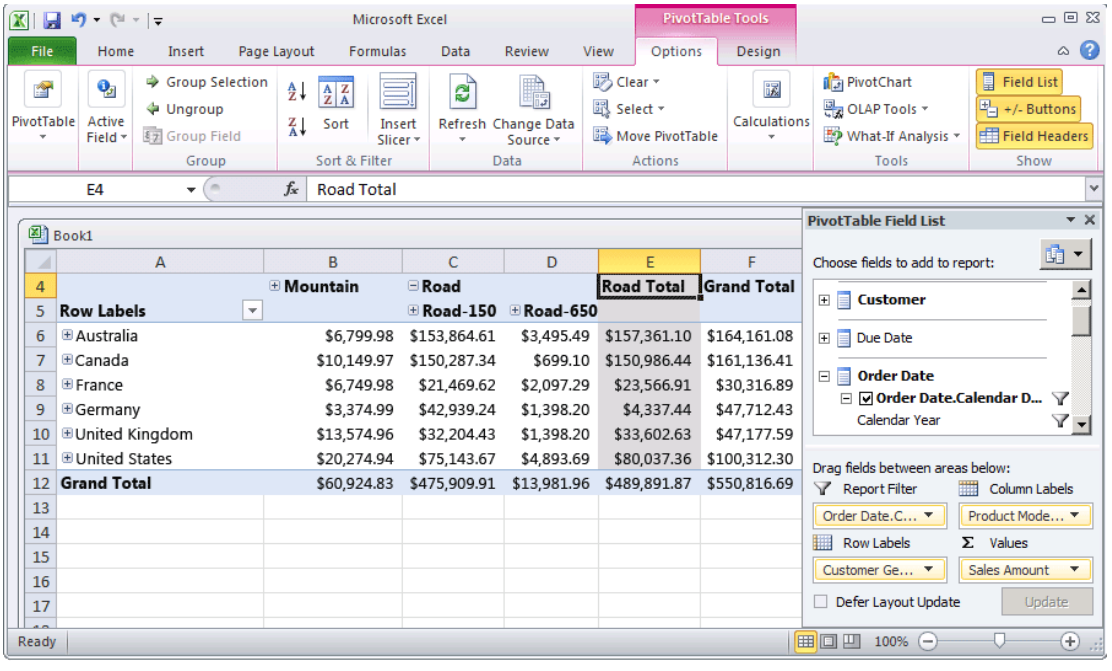
In the following task, you browse the Analysis Services Tutorial cube. Because our analysis compares measure across multiple dimensions, you will use an Excel PivotTable to browse your data. Using a PivotTable lets you place customer, date, and product information on different axes so that you can see how Internet Sales change when viewed across specific time periods, customer demographics, and product lines.

Procedures

To browse the deployed cube

1. To switch to Cube Designer in SQL Server Data Tools, double-click the **Analysis Services Tutorial** cube in the **Cubes** folder of the Solution Explorer.
2. Open the **Browser** tab, and then click the **Reconnect** button on the toolbar of the designer.
3. Click the Excel icon to launch Excel using the workspace database as the data source. When prompted to enable connections, click **Enable**.
4. In the PivotTable Field List, expand **Internet Sales**, and then drag the **Sales Amount** measure to the **Values** area.
5. In the PivotTable Field List, expand **Product**.
6. Drag the **Product Model Lines** user hierarchy to the **Columns** area.
7. In the PivotTable Field List, expand **Customer**, expand **Location**, and then drag the **Customer Geography** hierarchy from the Location display folder in the Customer dimension to the **Rows** area.
8. In the PivotTable Field List, expand **Order Date**, and then drag the **Order Date.Calendar Date** hierarchy to the **Report Filter** area.
9. Click the arrow to the right of the **Order Date.Calendar Date** filter in the data pane, clear the check box for the **(All)** level, expand **2006**, expand **H1 CY 2006**, expand **Q1 CY 2006**, select the check box for **February 2006**, and then click **OK**.

   Internet sales by region and product line for the month of February, 2006 appear as shown in the following image.



	A	B	C	D	E	F
4		Mountain	Road		Road Total	Grand Total
5	Row Labels		Road-150	Road-650		
6	Australia	$6,799.98	$153,864.61	$3,495.49	$157,361.10	$164,161.08
7	Canada	$10,149.97	$150,287.34	$699.10	$150,986.44	$161,136.41
8	France	$6,749.98	$21,469.62	$2,097.29	$23,566.91	$30,316.89
9	Germany	$3,374.99	$42,939.24	$1,398.20	$4,337.44	$47,712.43
10	United Kingdom	$13,574.96	$32,204.43	$1,398.20	$33,602.63	$47,177.59
11	United States	$20,274.94	$75,143.67	$4,893.69	$80,037.36	$100,312.30
12	Grand Total	$60,924.83	$475,909.91	$13,981.96	$489,891.87	$550,816.69
13						
14						
15						
16						
17						

Next Lesson

Lesson 4: Defining Advanced Attribute and Dimension Properties

Lesson 4: Defining Advanced Attribute and Dimension Properties

In this lesson, you will learn how to use some of the advanced properties of attributes, attribute hierarchies, and dimension properties.

Note

This lesson is based on an enhanced version of the Analysis Services Tutorial project that you completed in the first three lessons of this tutorial. The first task in this lesson describes where to locate the appropriate sample project to use for the lesson, and the difference between this project and the project that you created in the first three lessons.

This lesson contains the following tasks:

Using a Modified Version of the Analysis Services Tutorial Project

In this task, you open, review, and deploy a modified version of the Analysis Services Tutorial project, which has multiple measure groups and additional dimensions.

Defining Attribute Properties in a Parent - Child Dimension

In this task, you define level names in a parent-child dimension and specify whether data related to parent members is displayed. For more information, see Defining a Parent-Child Hierarchy and Working with Attributes in Parent-Child Hierarchies.

Automatically Grouping Attribute Members

In this task, you automatically create groupings of attribute members based on the distribution of the members within the attribute hierarchy. For more information, see Grouping Attribute Members (Discretization).

Hiding and Disabling Attribute Hierarchies

In this task, you learn how and when to disable or hide attribute hierarchies.

Sorting Attribute Members Based on a Secondary Attribute

In this task, you learn how to sort dimension members based on a secondary attribute, to achieve the sort order that you want.

Specifying Aggregation Relationships Between Attributes in a User Hierarchy

In this task, you learn how to define member properties for attributes and to specify aggregation relationships between them. For more information, see Defining and Configuring an Attribute Relationship and Configuring User-defined Hierarchy Properties.

Defining Unknown Member and Null Processing Properties

In this task, you configure the **UnknownMember** and **UnknownMemberName** properties to handle error conditions caused by null dimension members.

Next Lesson

Lesson 5: Defining Relationships Between Dimensions and Measure Groups

See Also

Analysis Services Tutorial Scenario

SQL Server 2005 Analysis Services Tutorial

Defining and Configuring Dimensions, Attributes and Hierarchies

Using a Modified Version of the Analysis Services Tutorial Project

The remaining lessons in this tutorial are based on an enhanced version of the Analysis Services Tutorial project that you completed in the first three lessons. Additional tables and named calculations have been added to the **Adventure Works DW 2012** data

source view, additional dimensions have been added to the project, and these new dimensions have been added to the Analysis Services Tutorial cube. In addition, a second measure group has been added, which contains measures from a second fact table. This enhanced project will enable you to continue learning how to add functionality to your business intelligence application without having to repeat the skills you have already learned.

Before you can continue with the tutorial, you must download, extract, load and process the enhanced version of the Analysis Services Tutorial project. Use the instructions in this lesson to ensure you have performed all the steps.

Downloading and Extracting the Project File

1. Click here to go to the download page that provides the sample projects that go with this tutorial. The tutorial projects are included in the **Analysis Services Tutorial SQL Server 2012** download.
2. Click **Analysis Services Tutorial SQL Server 2012** to download the package that contains the projects for this tutorial.

   By default, a .zip file is saved to the Downloads folder. You must move the .zip file to a location that has a shorter path (for example, create a C:\Tutorials folder to store the files). You can then extract the files contained in the .zip file. If you attempt to unzip the files from the Downloads folder, which has a longer path, you will only get Lesson 1.
3. Create a subfolder at or near the root drive, for example, C:\Tutorial.
4. Move the **Analysis Services Tutorial SQL Server 2012.zip** file to the subfolder.
5. Right-click the file and select **Extract All**.
6. Browse to the **Lesson 4 Start** folder to find the **Analysis Services Tutorial.sln** file.

Loading and Processing the Enhanced Project

1. In SQL Server Data Tools, on the **File** menu, click **Close Solution** to close files you won't be using.
2. On the **File** menu, point to **Open**, and then click **Project/Solution**.
3. Browse to the location where you extracted the tutorial project files.

   Find the folder named **Lesson 4 Start**, and then double-click Analysis Services Tutorial.sln.
4. Deploy the enhanced version of the Analysis Services Tutorial project to the local

instance of Analysis Services, or to another instance, and verify that processing completes successfully.

Understanding the Enhancements to the Project

The enhanced version of the project is different from the version of the Analysis Services Tutorial project that you completed in the first three lessons. The differences are described in the following sections. Review this information before continuing with the remaining lessons in the tutorial.

Data Source View

The data source view in the enhanced project contains one additional fact table and four additional dimension tables from the database.

Notice that with ten tables in the data source view, the <All Tables> diagram is becoming crowded. This makes it difficult to easily understand the relationships between the tables and to locate specific tables. To solve this problem, the tables are organized into two logical diagrams, the **Internet Sales** diagram and the **Reseller Sales** diagram. These diagrams are each organized around a single fact table. Creating logical diagrams lets you view and work with a specific subset of the tables in a data source view instead of always viewing all the tables and their relationships in a single diagram.

Internet Sales Diagram

The **Internet Sales** diagram contains the tables that are related to the sale of Adventure Works products directly to customers through the Internet. The tables in the diagram are the four dimension tables and one fact table that you added to the **Adventure Works DW 2012** data source view in Lesson 1. These tables are as follows:

- **Geography**
- **Customer**
- **Date**
- **Product**
- **InternetSales**

Reseller Sales Diagram

The **Reseller Sales** diagram contains the tables that are related to the sale of Adventure Works products by resellers. This diagram contains the following seven dimension tables and one fact table from the database:

- **Reseller**
- **Promotion**
- **SalesTerritory**
- **Geography**
- **Date**
- **Product**

- **Employee**
- **ResellerSales**

Notice that the **DimGeography**, **DimDate**, and **DimProduct** tables are used in both the **Internet Sales** diagram and the **Reseller Sales** diagram. Dimension tables can be linked to multiple fact tables.

Database and Cube Dimensions

The Analysis Services Tutorial project contains five new database dimensions, and the Analysis Services Tutorial cube contains these same five dimensions as cube dimensions. These dimensions have been defined to have user hierarchies and attributes that were modified by using named calculations, composition member keys, and display folders. The new dimensions are described in the following list.

Reseller Dimension

The Reseller dimension is based on the **Reseller** table in the **Adventure Works DW 2012** data source view.

Promotion Dimension

The Promotion dimension is based on the **Promotion** table in the **Adventure Works DW 2012** data source view.

Sales Territory Dimension

The Sales Territory dimension is based on the **SalesTerritory** table in the **Adventure Works DW 2012** data source view.

Employee Dimension

The Employee dimension is based on the **Employee** table in the **Adventure Works DW 2012** data source view.

Geography Dimension

The Geography dimension is based on the **Geography** table in the **Adventure Works DW 2012** data source view.

Analysis Services Cube

The **Analysis Services Tutorial** cube now contains two measure groups, the original measure group based on the **InternetSales** table and a second measure group based on the **ResellerSales** table in the **Adventure Works DW 2012** data source view.

Next Task in Lesson

Defining Attribute Properties in a Parent - Child Dimension

See Also

Deploying an Analysis Services Project

Defining Parent Attribute Properties in a Parent-Child Hierarchy

A parent-child hierarchy is a hierarchy in a dimension that is based on two table columns. Together, these columns define the hierarchical relationships among the members of the dimension. The first column, called the *member key column*, identifies each dimension member. The other column, called the *parent column*, identifies the parent of each dimension member. The **NamingTemplate** property of a parent attribute determines the name of each level in the parent-child hierarchy, and the **MembersWithData** property determines whether data for parent members should be displayed.

For more information, see Defining a Parent-Child Hierarchy, Working with Attributes in Parent-Child Hierarchies

Note
When you use the Dimension Wizard to create a dimension, the wizard recognizes the tables that have parent-child relationships and automatically defines the parent-child hierarchy for you.

In the tasks in this topic, you will create a naming template that defines the name for each level in the parent-child hierarchy in the **Employee** dimension. You will then configure the parent attribute to hide all parent data, so that only the sales for leaf-level members are displayed.

Browsing the Employee Dimension

1. In Solution Explorer, double-click **Employee.dim** in the **Dimensions** folder to open Dimension Designer for the Employee dimension.
2. Click the **Browser** tab, verify that **Employees** is selected in the **Hierarchy** list, and then expand the **All Employees** member.

   Notice that **Ken J. Sánchez** is the top-level manager in this parent-child hierarchy.
3. Select the **Ken J. Sánchez** member.

   Notice that the level name for this member is **Level 02**. (The level name appears after **Current level:** immediately above the **All Employees** member.) In the next task, you will define more descriptive names for each level.
4. Expand **Ken J. Sánchez** to view the names of the employees who report to this manager, and then select **Brian S. Welcker** to view the name of this level.

   Notice that the level name for this member is **Level 03**.
5. In Solution Explorer, double-click **Analysis Services Tutorial.cube** in the **Cubes**

folder to open Cube Designer for the Analysis Services Tutorial cube.

6. Click the **Browser** tab.
7. Click the Excel icon, and then click **Enable** when prompted to enable connections.
8. In the PivotTable Field List, expand **Reseller Sales**. Drag **Reseller Sales-Sales Amount** to the Values area.
9. In the PivotTable Field List, expand **Employee**, and then drag the **Employees** hierarchy to the **Rows** area.

   All the members of the Employees hierarchy are added to column A of the PivotTable report.

   The following image shows the Employees hierarchy expanded.
10.

    Notice that the sales made by each manager in Level 03 are also displayed in Level 04. This is because each manager is also an employee of another manager. In the next task, you will hide these sale amounts.

Modifying Parent Attribute Properties in the Employee Dimension

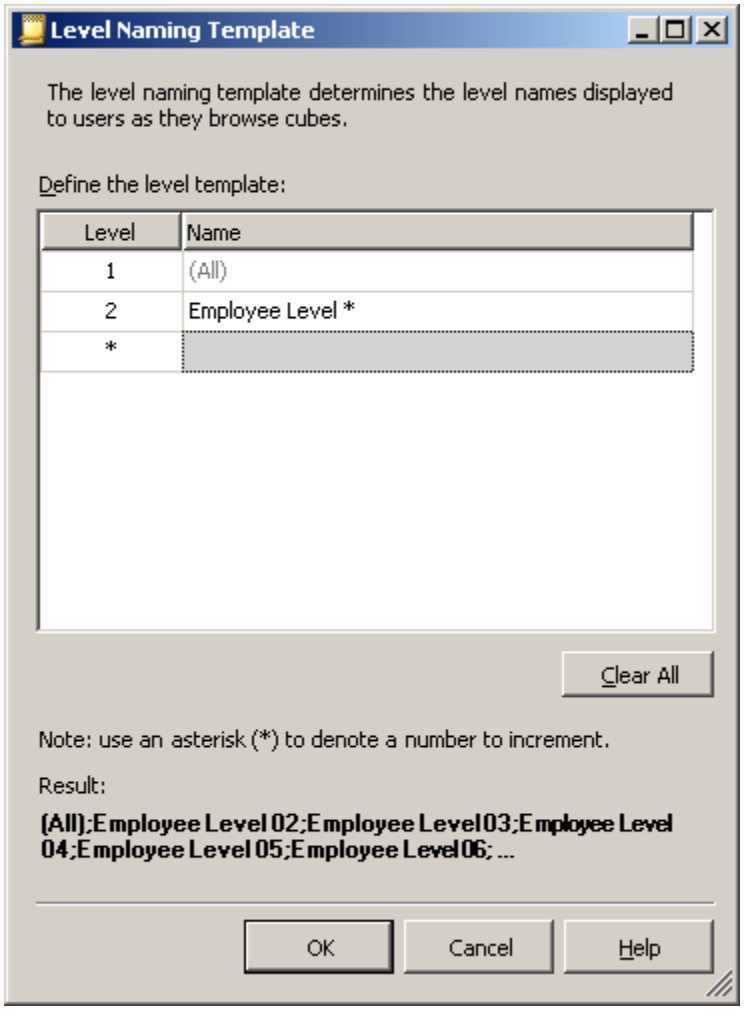
1. Switch to Dimension Designer for the **Employee** dimension.
2. Click the **Dimension Structure** tab, and then select the **Employees** attribute hierarchy in the **Attributes** pane.

   Notice the unique icon for this attribute. This icon signifies that the attribute is the parent key in a parent-child hierarchy. Notice also, in the Properties window, that the **Usage** property for the attribute is defined as **Parent**. This property was set by the Dimension Wizard when the dimension was designed. The wizard automatically detected the parent-child relationship.
3. In the Properties window, click the ellipsis button (**...**) in the **NamingTemplate** property cell.

   In the **Level Naming Template** dialog box, you define the level naming template that determines the level names in the parent-child hierarchy that are displayed to users as they browse cubes.
4. In the second row, the ***** row, type **Employee Level *** in the **Name** column, and then click the third row.

   Notice under **Result** that each level will now be named "Employee Level" followed by a sequentially increasing number.

   The following image shows the changes in the **Level Naming Template** dialog box.



5. Click **OK**.
6. In the Properties window for the **Employees** attribute, in the **MembersWithData** property cell, select **NonLeafDataHidden** to change this value for the **Employees** attribute.

   This will cause data that is related to non-leaf level members in the parent-child hierarchy to be hidden.

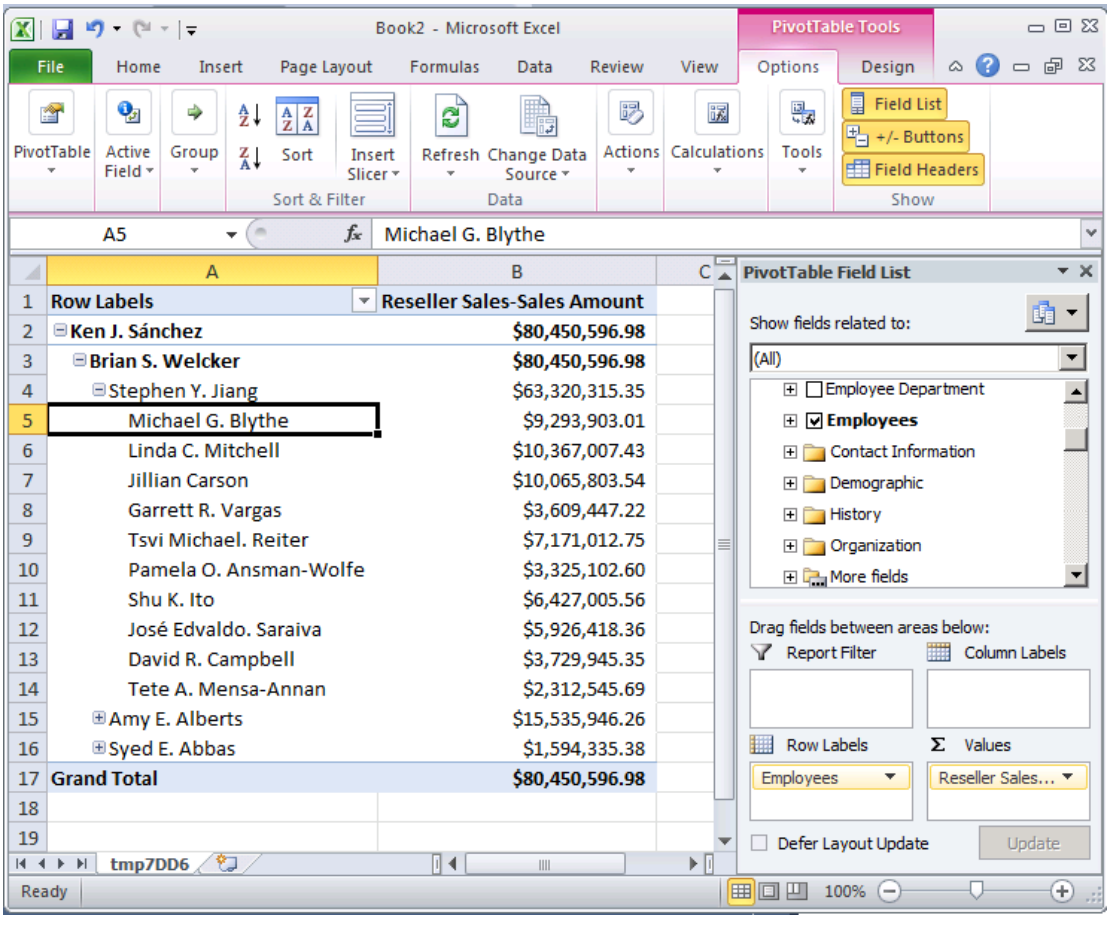
Browsing the Employee Dimension with the Modified Attributes

1. On the **Build** menu of SQL Server Data Tools (SSDT), click **Deploy Analysis Services Tutorial**.
2. When deployment has successfully completed, switch to Cube Designer for the Analysis Services Tutorial cube, and then click **Reconnect** on the toolbar of the

Browser tab.

3. Click the Excel icon, and then click **Enable**.
4. Drag **Reseller Sales-Sales Amount** to the Values area.
5. Drag the **Employees** hierarchy to the Row Labels area.

   The following image shows the changes that you made to the Employees hierarchy. Notice that Stephen Y. Jiang no longer appears as an employee of himself.



Next Task in Lesson

Automatically Grouping Attribute Members

See Also

Defining a Parent-Child Hierarchy

Working with Attributes in Parent-Child Hierarchies

Automatically Grouping Attribute Members

When you browse a cube, you typically dimension the members of one attribute hierarchy by the members of another attribute hierarchy. For example, you might group customer sales by city, by product purchased, or by gender. However, with certain types of attributes, it is useful to have Microsoft Analysis Services automatically create groupings of attribute members based on the distribution of the members within an attribute hierarchy. For example, you can have Analysis Services create groups of yearly income values for customers. When you do this, users who browse the attribute hierarchy will see the names and values of the groups instead of the members themselves. This limits the number of levels that are presented to users, which can be more useful for analysis.

The **DiscretizationMethod** property determines whether Analysis Services creates groupings, and determines the type of grouping that is performed. By default, Analysis Services does not perform any groupings. When you enable automatic groupings, you can allow Analysis Services to automatically determine the best grouping method based on the structure of the attribute, or you can choose one of the grouping algorithms in the following list to specify the grouping method:

EqualAreas

Analysis Services creates group ranges so that the total population of dimension members is distributed equally across the groups.

Clusters

Analysis Services creates groups by performing single-dimensional clustering on the input values by using the K-Means clustering method with Gaussian distributions. This option is valid only for numeric columns.

After you specify a grouping method, you must specify the number of groups, by using the **DiscretizationBucketCount** property. For more information, see Grouping Attribute Members (Discretization)

In the tasks in this topic, you will enable different types of groupings for the following: the yearly income values in the **Customer** dimension; the number of employee sick leave hours in the **Employees** dimension; and the number of employee vacation hours in the **Employees** dimension. You will then process and browse the Analysis Services Tutorial cube to view the effect of the member groups. Finally, you will modify the member group properties to see the effect of the change in grouping type.

Grouping Attribute Hierarchy Members in the Customer Dimension

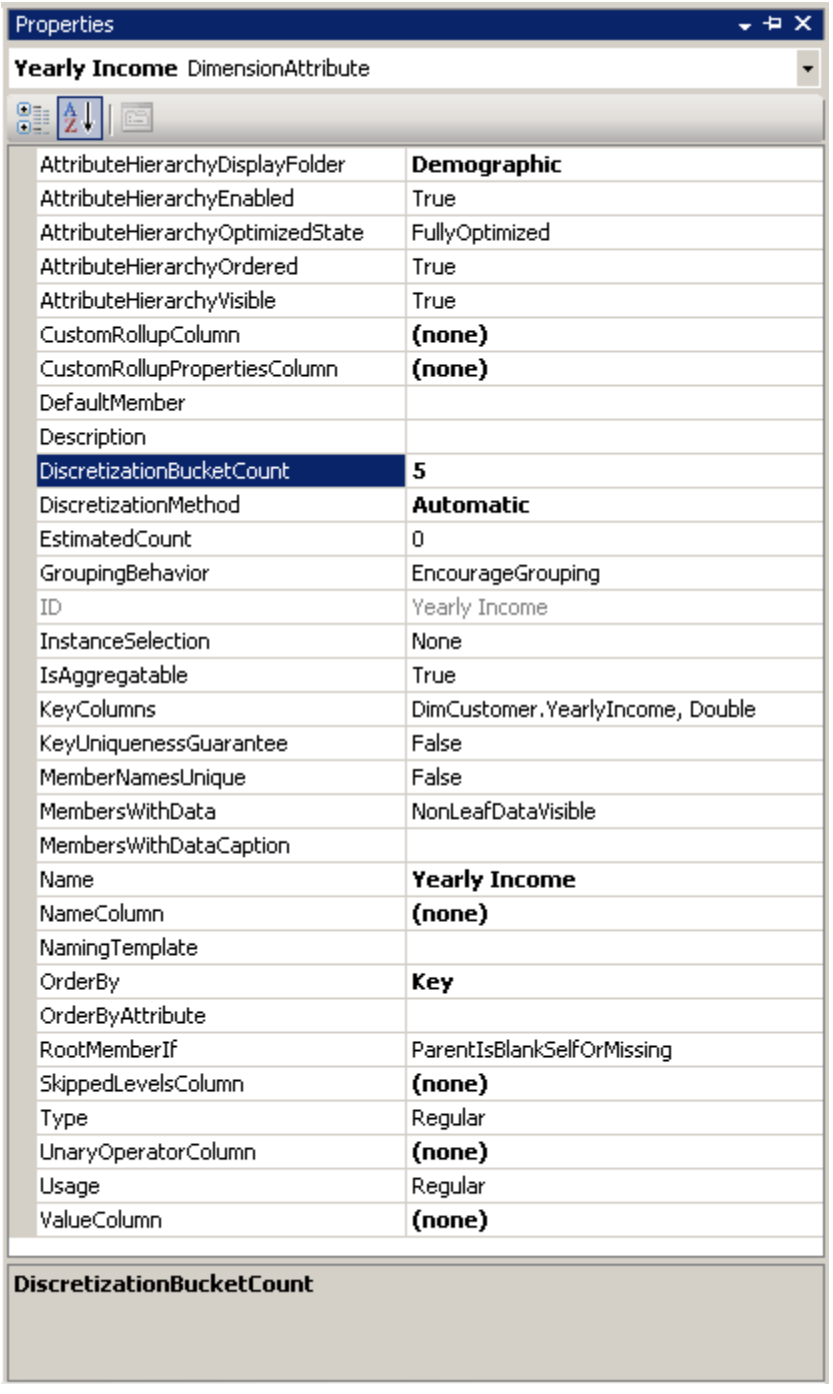
1. In Solution Explorer, double-click **Customer** in the **Dimensions** folder to open Dimension Designer for the Customer dimension.
2. In the **Data Source View** pane, right-click the **Customer** table, and then click

Explore Data.

Notice the range of values for the **YearlyIncome** column. These values become the members of the **Yearly Income** attribute hierarchy, unless you enable member grouping.

3. Close the **Explore Customer Table** tab.
4. In the **Attributes** pane, select **Yearly Income**.
5. In the Properties window, change the value for the **DiscretizationMethod** property to **Automatic** and change the value for the **DiscretizationBucketCount** property to **5**.

   The following image shows the modified properties for **Yearly Income**.



Grouping Attribute Hierarchy Members in the Employee Dimension

1. Switch to Dimension Designer for the Employee dimension.
2. In the **Data Source View** pane, right-click the **Employee** table, and then click **Explore Data**.

   Notice the values for the **SickLeaveHours** column and the **VacationHours** column.
3. Close the **Explore Employee Table** tab.
4. In the **Attributes** pane, select **Sick Leave Hours**.
5. In the Properties window, change the value for the **DiscretizationMethod** property to **Clusters** and change the value for the **DiscretizationBucketCount** property to **5**.
6. In the **Attributes** pane, select **Vacation Hours**.
7. In the Properties window, change the value for the **DiscretizationMethod** property to **Equal Areas** and change the value for the **DiscretizationBucketCount** property to **5**.

Browsing the Modified Attribute Hierarchies

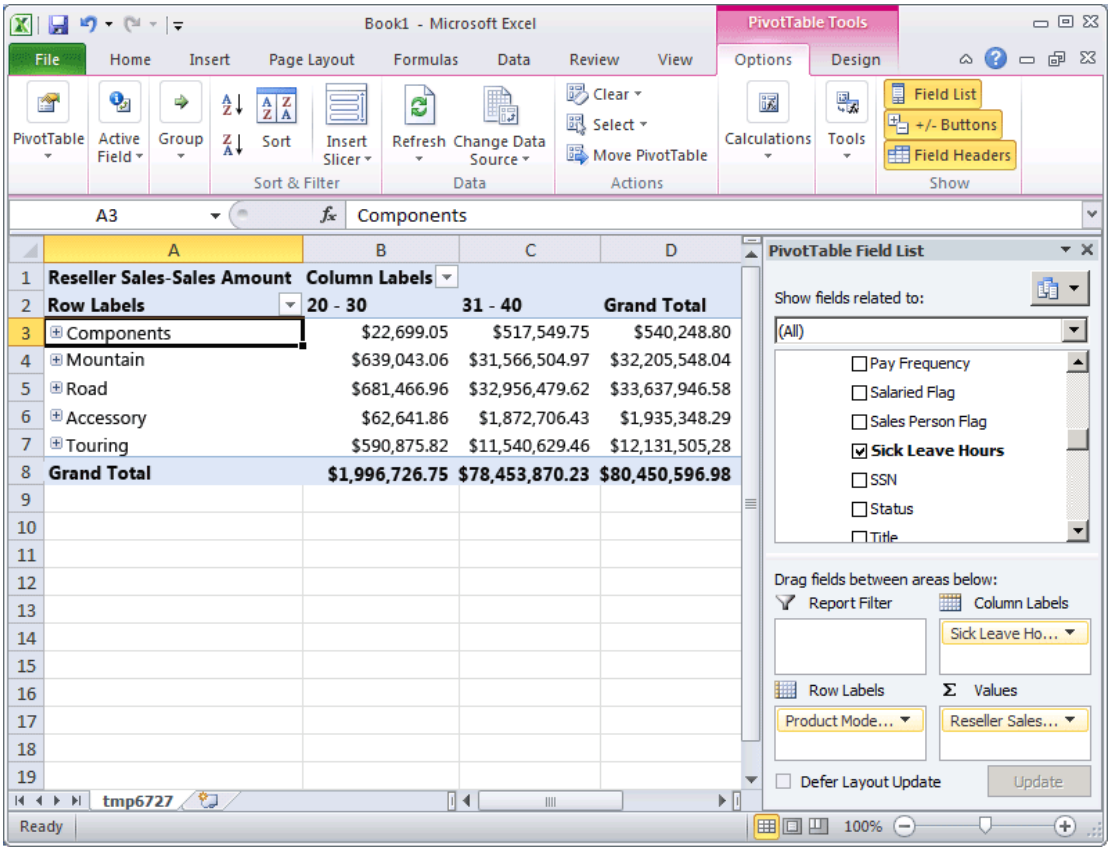
1. On the **Build** menu of SQL Server Data Tools (SSDT), click **Deploy Analysis Services Tutorial**.
2. When deployment has successfully completed, switch to Cube Designer for the Analysis Services Tutorial cube, and then click **Reconnect** on the **Browser** tab.
3. Click the Excel icon, and then click **Enable**.
4. Drag the **Internet Sales-Sales Amount** measure to the Values area of the PivotTable Field List.
5. In the field list, expand the **Product** dimension, and then drag the **Product Model Lines** user hierarchy to the **Row Labels** area of the field list.
6. Expand the **Customer** dimension in the field list, expand the **Demographic** display folder, and then drag the **Yearly Income** attribute hierarchy to the **Column Labels** area.

   The members of the **Yearly Income** attribute hierarchy are now grouped into six buckets, including a bucket for sales to customers whose yearly income is unknown. Not all buckets are displayed.
7. Remove the **Yearly Income** attribute hierarchy from the columns area and remove the **Internet Sales-Sales Amount** measure from the **Values** area.
8. Add the **Reseller Sales-Sales Amount** measure to the data area.
9. In the field list, expand the **Employee** dimension, expand **Organization**, then drag **Sick Leave Hours** to **Column Labels**.

   Notice that all sales are made by employees within one of two groups. Notice

also that the employees with 32 - 42 sick leave hours made significantly more sales than employees with 20 - 31 sick leave hours.

The following image shows sales dimensioned by employee sick leave hours.



10. Remove the **Sick Leave Hours** attribute hierarchy from the column area of the **Data** pane.
11. Add **Vacation Hours** to the column area of the **Data** pane.

    Notice that two groups appear, based on the equal areas grouping method. Three other groups are hidden because they contain no data values.

Modifying Grouping Properties and Reviewing the Effect of the Changes

1. Switch to Dimension Designer for the **Employee** dimension, and then select **Vacation Hours** in the **Attributes** pane.
2. In the Properties window, change the value of the **DiscretizationBucketCount** property to **10.**
3. On the **Build** menu of SQL Server Data Tools, click **Deploy Analysis Services**

Tutorial.

4. When deployment has successfully completed, switch back to Cube Designer for the Analysis Services Tutorial cube.
5. Click **Reconnect** on the **Browser** tab, click the Excel icon, and then reconstruct the PivotTable so that you can view the effect of the change to the grouping method:
   a. Drag Reseller Sales-Sales Amount to Values
   b. Drag Vacation Hours (in the Employees Organization folder) to Columns
   c. Drag Product Model Lines to Rows

   Notice that there are now three groups of members of the **Vacation Hours** attribute that have sales values for products. (The other seven groups contain members with no sales data.)

Next Task in Lesson

Hiding and Disabling Attribute Hierarchies

See Also

Grouping Attribute Members (Discretization)

Hiding and Disabling Attribute Hierarchies

By default, an attribute hierarchy is created for every attribute in a dimension, and each hierarchy is available for dimensioning fact data. This hierarchy consists of an "All" level and a detail level containing all members of the hierarchy. As you have already learned, you can organize attributes into user-defined hierarchies to provide navigation paths in a cube. Under certain circumstances, you may want to disable or hide some attributes and their hierarchies. For example, certain attributes such as social security numbers or national identification numbers, pay rates, birth dates, and login information are not attributes by which users will dimension cube information. Instead, this information is generally only viewed as details of a particular attribute member. You may want to hide these attribute hierarchies, leaving the attributes visible only as member properties of a specific attribute. You may also want to make members of other attributes, such as customer names or postal codes, visible only when they are viewed through a user hierarchy instead of independently through an attribute hierarchy. One reason to do so may be the sheer number of distinct members in the attribute hierarchy. Finally, to improve processing performance, you should disable attribute hierarchies that users will not use for browsing.

The value of the **AttributeHierarchyEnabled** property determines whether an attribute hierarchy is created. If this property is set to **False**, the attribute hierarchy is not created and the attribute cannot be used as a level in a user hierarchy; the attribute hierarchy exists as a member property only. However, a disabled attribute hierarchy can still be

used to order the members of another attribute. If the value of the **AttributeHierarchyEnabled** property is set to **True**, the value of the **AttributeHierarchyVisible** property determines whether the attribute hierarchy is visible independent of its use in a user-defined hierarchy.

When an attribute hierarchy is enabled, you may want to specify values for the following three additional properties:

- **IsAggregatable**

  By default, an (All) level is defined for all attribute hierarchies. To disable the (All) level for an enabled attribute hierarchy, set the value for this property to **False**.

  Note
  An attribute that has its **IsAggregatable** property set to false can only be used as the root of a user-defined hierarchy and must have a default member specified (otherwise, one will be chosen for you by the Analysis Services engine).

- **AttributeHierarchyOrdered**

  By default, Analysis Services orders the members of enabled attribute hierarchies during processing, and then stores the members by the value of the **OrderBy** property, such as by Name or Key. If you do not care about ordering, you can increase processing performance by setting the value of this property to **False**.

- **AttributeHierarchyOptimizedState**

  By default, Analysis Services creates an index for each enabled attribute hierarchy during processing, to improve query performance. If you do not plan to use an attribute hierarchy for browsing, you can increase processing performance by setting the value of this property to **NotOptimized**. However, if you use a hidden hierarchy as the key attribute for the dimension, creating an index of the attribute members will still improve performance.

These properties do not apply if an attribute hierarchy is disabled.

In the tasks in this topic, you will disable social security numbers and other attributes in the Employee dimension that will not be used for browsing. You will then hide the customer name and postal code attribute hierarchies in the Customer dimension. The large number of attribute members in these hierarchies will make browsing these hierarchies very slow independent of a user hierarchy.

Setting Attribute Hierarchy Properties in the Employee Dimension

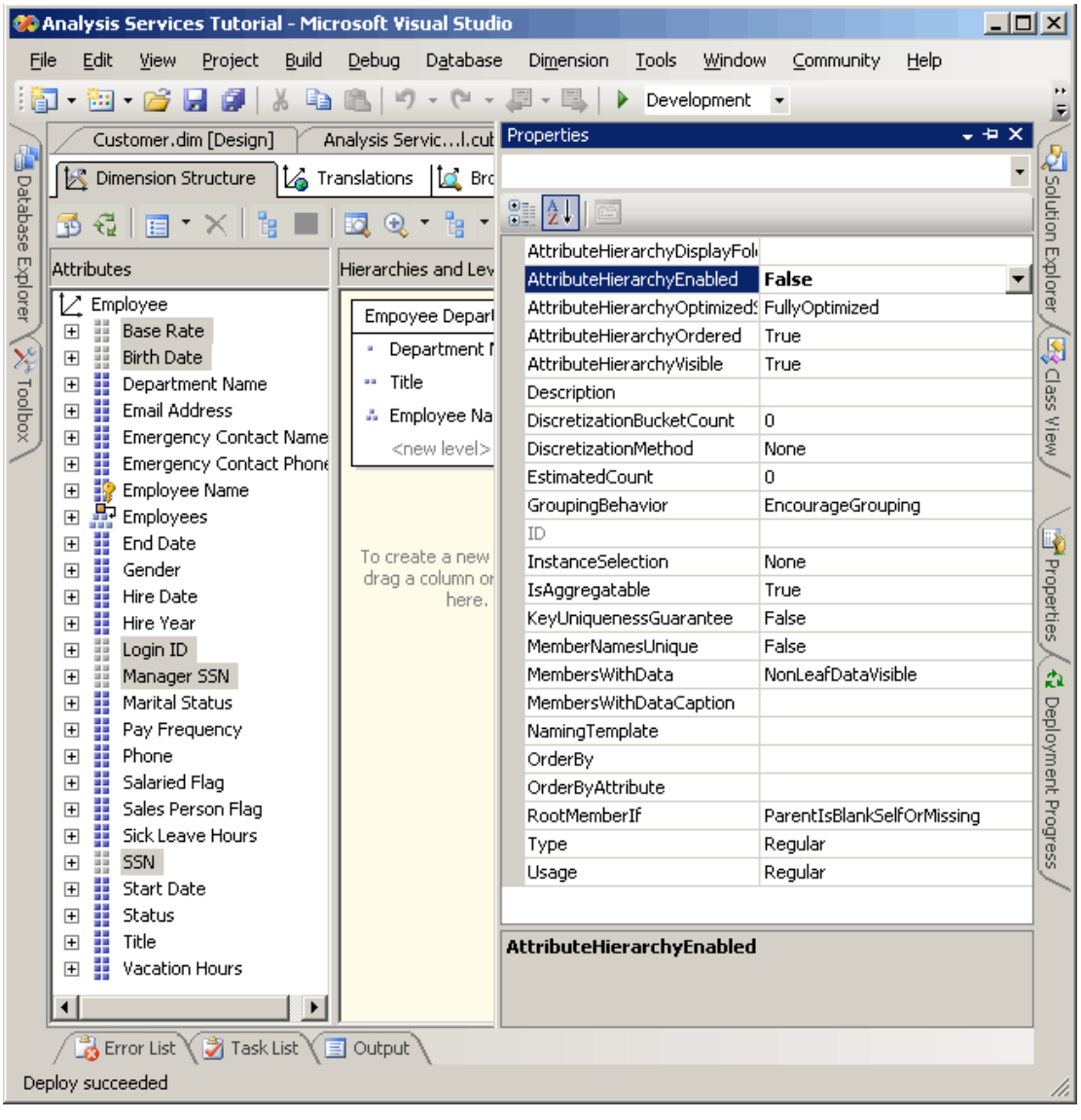
1. Switch to Dimension Designer for the Employee dimension, and then click the **Browser** tab.
2. Verify that the following attribute hierarchies appear in the **Hierarchy** list:

- **Base Rate**
- **Birth Date**
- **Login ID**
- **Manager SSN**
- **SSN**

3. Switch to the **Dimension Structure** tab, and then select the following attributes in the **Attributes** pane. You can select multiple measures by clicking each while holding down the CTRL key:
   - **Base Rate**
   - **Birth Date**
   - **Login ID**
   - **Manager SSN**
   - **SSN**
4. In the Properties window, set the value of the **AttributeHierarchyEnabled** property to **False** for the selected attributes.

   Notice in the **Attributes** pane that the icon for each attribute has changed to indicate that the attribute is not enabled.

   The following image shows the **AttributeHierarchyEnabled** property set to False for the selected attributes.



5. On the **Build** menu, click **Deploy Analysis Services Tutorial**.
6. When processing has successfully completed, switch to the **Browser** tab, click **Reconnect**, and then try to browse the modified attribute hierarchies.

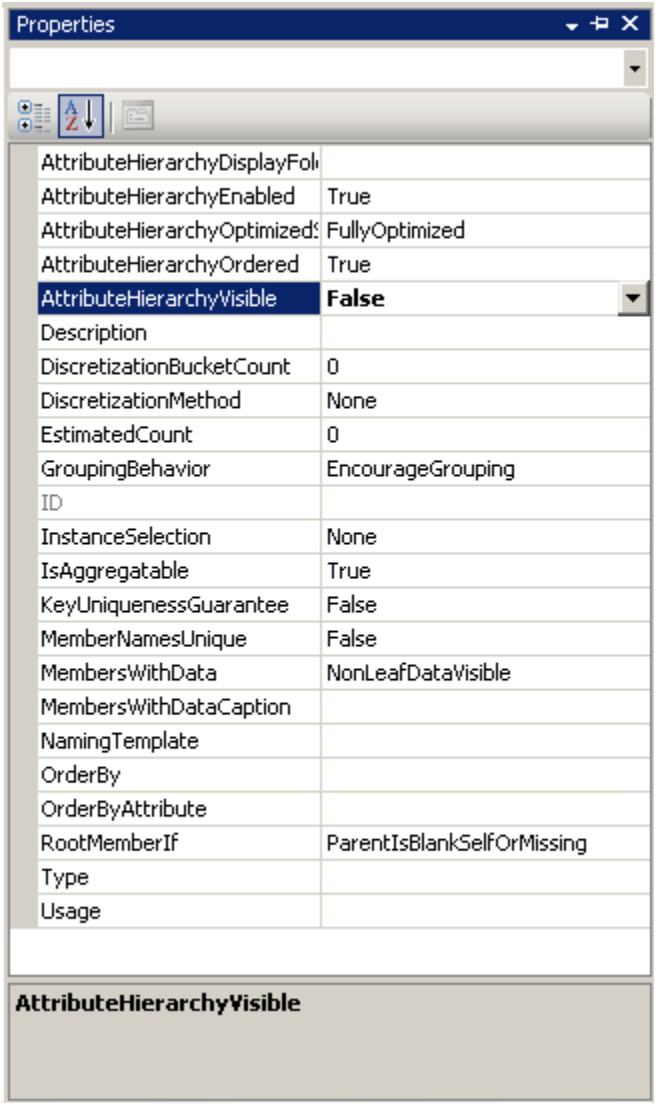
   Notice that the members of the modified attributes are not available for browsing as attribute hierarchies in the **Hierarchy** list. If you try to add one of the disabled attribute hierarchies as a level in a user hierarchy, you will receive an error notifying you that the attribute hierarchy must be enabled to participate in a user-defined hierarchy.

Setting Attribute Hierarchy Properties in the Customer Dimension

1. Switch to Dimension Designer for the Customer dimension, and then click the **Browser** tab.
2. Verify that the following attribute hierarchies appear in the **Hierarchy** list:
   - **Full Name**
   - **Postal Code**
3. Switch to the **Dimension Structure** tab, and then select the following attributes in the **Attributes** pane by using the CTRL key to select multiple attributes at the same time:
   - **Full Name**
   - **Postal Code**
4. In the Properties window, set the value of the **AttributeHierarchyVisible** property to **False** for the selected attributes.

   Because the members of these attribute hierarchies will be used for dimensioning fact data, ordering and optimizing the members of these attribute hierarchies will improve performance. Therefore, the properties of these attributes should not be changed.

   The following image shows the **AttributeHierarchyVisible** property set to False.



5. Drag the **Postal Code** attribute from the **Attributes** pane into the **Customer Geography** user hierarchy in the **Hierarchies and Levels** pane, immediately under the **City** level.

   Notice that a hidden attribute can still become a level in a user hierarchy.
6. On the **Build** menu, click **Deploy Analysis Services Tutorial**.
7. When deployment has successfully completed, switch to the **Browser** tab for the Customer dimension, and then click **Reconnect**.
8. Try to select either of the modified attribute hierarchies from the **Hierarchy** list.

   Notice that neither of the modified attribute hierarchies appears in the **Hierarchy** list.
9. In the **Hierarchy** list, select **Customer Geography**, and then browse each level in

the browser pane.

Notice that the hidden levels, **Postal Code** and **Full Name**, are visible in the user-defined hierarchy.

Next Task in Lesson

Sorting Attribute Members Based on a Secondary Attribute

See Also

Designing and Implementing How-to Topics (Analysis Services - Multidimensional Data)

Sorting Attribute Members Based on a Secondary Attribute

In Lesson 3, you learned how to sort attribute members based on either their name or key value. You also learned how to use a composite member key to affect attribute members and sort order. For more information, see Modifying the Time Dimension. However, if neither the name nor the key of the attribute provide the sort order that you want, you can use a secondary attribute to achieve the desired sort order. By defining a relationship between the attributes, you can use the second attribute to sort the members of the first attribute.

Attribute relationships define the relationships or dependencies between attributes. In a dimension that is based on a single relational table, all attributes are typically related to each other through the key attribute. This is because all the attributes for a dimension provide information about the members linked by the key attribute of the dimension to the facts in the fact table for each related measure group. In a dimension that is based on multiple tables, attributes are typically linked based on the join key between the tables. If the underlying data supports it, related attributes can be used to specify a sort order. For example, you might create a new attribute that provides the sort logic for a related attribute.

Dimension Designer lets you define additional relationships between attributes or change the default relationships to increase performance. The main constraint when you create an attribute relationship is to make sure that the attribute referred to has no more than one value for any member in the attribute to which it is related. When you define a relationship between two attributes, you can define the relationship as rigid or flexible, based on whether the relationships between members will change over time. For example, an employee might move to a different sales region, but a city will not move to a different state. If a relationship is defined as rigid, attribute aggregations are not recalculated every time the dimension is incrementally processed. However, if the relationship between members does change, the dimension must be fully processed. For more information, see Attribute Relationships, Defining and Configuring an Attribute

Relationship, Configuring Attribute Relationship Properties, and Specifying Attribute Relationships Between Attributes in a User-defined Hierarchy.

In the tasks in this topic, you will define a new attribute in the **Date** dimension based on an existing column in the underlying dimension table. You will use this new attribute to sort calendar month members chronologically instead of alphabetically. You will also define a new attribute in the **Customer** dimension based on the named calculation that you will use to sort the **Commute Distance** attribute members. In the tasks in the next topic, you will learn to use attribute relationships to increase query performance.

Defining an Attribute Relationship and Sort Order in the Date Dimension

1. Open Dimension Designer for the **Date** dimension, and then review the **OrderBy** property for the **Month Name** attribute in the Properties window.

   Notice that the **Month Name** attribute members are ordered by their key values.
2. Switch to the **Browser** tab, verify that **Calendar Date** is selected in the **Hierarchy** list, and then expand the levels in the user-defined hierarchy to review the sort order for the calendar months.

   Notice that the members of the attribute hierarchy are sorted based on the ASCII values of their member keys, which are month and year. In this case, sorting by the attribute name or key does not sort calendar months chronologically. To solve this, you will sort the members of the attribute hierarchy based on a new attribute, the **MonthNumberOfYear** attribute. You will create this attribute based on a column that conveniently exists in the **Date** dimension table.
3. Switch to the **Dimension Structure** tab for the Date dimension, right-click **MonthNumberOfYear** in the **Data Source View** pane, and then click **New Attribute from Column**.
4. In the **Attributes** pane, select **Month Number Of Year**, and then set the **AttributeHierarchyEnabled** property to **False** in the Properties window, set the **AttributeHierarchyOptimizedState** property to **NotOptimized**, and set the **AttributeHierarchyOrdered** property to **False**.

   These settings will hide the attribute from users and will improve processing time. This attribute will not be used for browsing. It will only be used for ordering the members of another attribute.

    **Note**
   Sorting properties in the Properties window alphabetically will simplify this task as these three properties will be sorted adjacent to each other.
5. Click the **Attribute Relationships** tab.

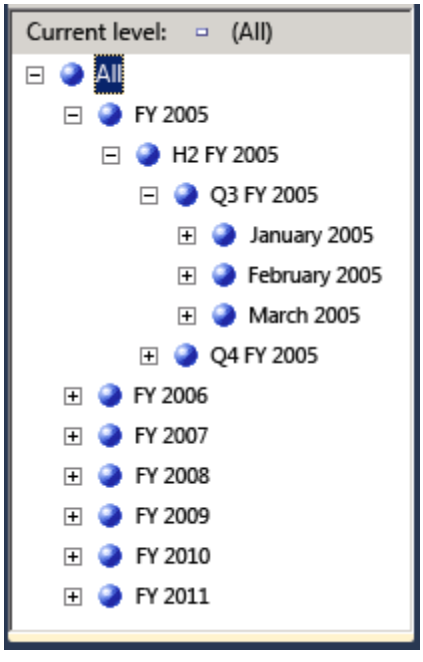
   Notice that all the attributes in the **Date** dimension are related directly to the **Date** attribute, which is the member key that relates the dimension members to

the facts in the related measure groups. There is no relationship defined between the **Month Name** attribute and the **Month Number Of Year** attribute.

6. In the diagram, right-click the **Month Name** attribute and then select **New Attribute Relationship**.
7. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Month Name**. Set the **Related Attribute** to **Month Number Of Year**.
8. In the **Relationship type** list, set the relationship type to **Rigid**.

   The relationships between the members of the **Month Name** attribute and the **Month Number Of Year** attribute will not change over time. As a result, Analysis Services will not drop aggregations for this relationship during incremental processing. If a change does occur, a processing error will occur during incremental processing and you will need to perform a full process of the dimension. You are now ready to set the sort order for the members of **Month Name**.
9. Click .
10. Click the **Dimension Structure** tab.
11. Select **Month Name** in the **Attributes** pane, and then change the value of the **OrderBy** property in the Properties window to **AttributeKey** and change the value of the **OrderByAttribute** property to **Month Number Of Year**.
12. On the **Build** menu, click **Deploy Analysis Services Tutorial**.
13. When deployment has successfully completed, switch to the **Browser** tab for the Date dimension, click **Reconnect**, and then browse the **Calendar Date** and **Fiscal Date** user hierarchies to verify that months now sort in chronological order.

    Notice that the months are now sorted in chronological order, as shown in the following image.

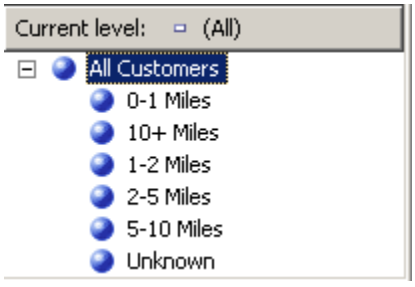


Defining Attribute Relationships and Sort Order in the Customer Dimension

1. Switch to the **Browser** tab in Dimension Designer for the Customer dimension, and then browse the members of the **Commute Distance** attribute hierarchy.

   Notice that the members of this attribute hierarchy are sorted based on the ASCII values of the member key. In this case, sorting by the attribute name or key does not sort the commute distances from least to most. In this task, you sort the members of the attribute hierarchy based on the **CommuteDistanceSort** named calculation that ascribes the appropriate sort number to each distinct value in the column. To save time, this named calculation has already been added to the **Customer** table in the Adventure Works DW data source view. You can switch to this data source view to view the SQL script that is used in this named calculation. For more information, see Creating Named Calculations in a Data Source View (SSAS).

   The following image shows the members of the **Commute Distance** attribute hierarchy, sorted by the ASCII values of the member key.



2. Switch to the **Dimension Structure** tab in Dimension Designer for the Customer dimension, right-click **CommuteDistanceSort** in the **Customer** table in the **Data Source View** pane, and then click **New Attribute from Column**.
3. In the **Attributes** pane, select **Commute Distance Sort**, and then set the **AttributeHierarchyEnabled** property for this attribute to **False** in the Properties window, set the **AttributeHierarchyOptimizedState** property to **NotOptimized**, and set the **AttributeHierarchyOrdered** property to **False**.

   These settings will hide the attribute from users and will improve processing time. This attribute will not be used for browsing. It will only be used for ordering the members of another attribute.
4. Select **Geography**, and then set its **AttributeHierarchyVisible** property to **False** in the Properties window, set its **AttributeHierarchyOptimizedState** property to **NotOptimized**, and set its **AttributeHierarchyOrdered** property to **False**.

   These settings will hide the attribute from users and will improve processing time. This attribute will not be used for browsing. It will be only be used for ordering the members of another attribute. Because **Geography** has member properties, its **AttributeHierarchyEnabled** property must be set to **True**. Therefore, to hide the attribute, you set the **AttributeHierarchyVisible** property to **False**.
5. Click the **Attribute Relationships** tab.
6. In the attributes list, right-click the **Commute Distance** attribute and then select **New Attribute Relationship**.
7. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Commute Distance**. Set the **Related Attribute** to **Commute Distance Sort**.
8. In the **Relationship type** list, set the relationship type to **Rigid**.

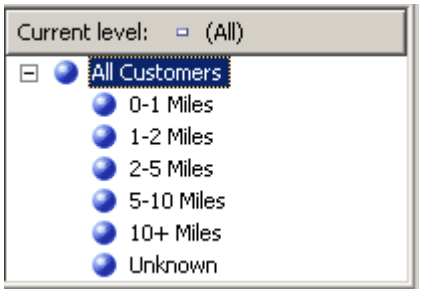
   The relationship between the members of the **Commute Distance** attribute and the **Commute Distance Sort** attribute will not change over time.
9. Click .

   You are now ready to set the sort order for the **Commute Distance** attribute.
10. Click the **Dimension Structure** tab.
11. In the **Attributes** pane, select **Commute Distance**, and then change the value of the **OrderBy** property in the Properties window to **AttributeKey**, and change the

value of the **OrderByAttribute** property to **Commute Distance Sort**.

12. On the **Build** menu, click **Deploy Analysis Services Tutorial**.
13. When deployment has successfully completed, switch to the **Browser** tab of Dimension Designer for the Customer dimension, click **Reconnect**, and then browse the **Commute Distance** attribute hierarchy.

    Notice that the attribute hierarchy members are now sorted in a logical order based on increasing distance, as shown in the following image.



Next Task in Lesson

Specifying Aggregation Relationships Between Attributes in a User-defined Hierarchy

Specifying Attribute Relationships Between Attributes in a User-Defined Hierarchy

As you have already learned in this tutorial, you can organize attribute hierarchies into levels within user hierarchies to provide navigation paths for users in a cube. A user hierarchy can represent a natural hierarchy, such as city, state, and country, or can just represent a navigation path, such as employee name, title, and department name. To the user navigating a hierarchy, these two types of user hierarchies are the same.

With a natural hierarchy, if you define attribute relationships between the attributes that make up the levels, Analysis Services can use an aggregation from one attribute to obtain the results from a related attribute. If there are no defined relationships between attributes, Analysis Services will aggregate all non-key attributes from the key attribute. Therefore, if the underlying data supports it, you should define attribute relationships between attributes. Defining attribute relationships improves dimension, partition, and query processing performance. For more information, see Defining Attribute Relationships and Attribute Relationships.

When you define attribute relationships, you can specify that the relationship is either flexible or rigid. If you define a relationship as rigid, Analysis Services retains aggregations when the dimension is updated. If a relationship that is defined as rigid actually changes, Analysis Services generates an error during processing unless the dimension is fully processed. Specifying the appropriate relationships and relationship

properties increases query and processing performance. For more information, see Defining and Configuring an Attribute Relationship, and Configuring User-defined Hierarchy Properties.

In the tasks in this topic, you define attribute relationships for the attributes in the natural user hierarchies in the Analysis Services Tutorial project. These include the **Customer Geography** hierarchy in the **Custome**r dimension, the **Sales Territory** hierarchy in the **Sales Territory** dimension, the **Product Model** Lines hierarchy in the **Product** dimension, and the **Fiscal Date** and **Calendar Date** hierarchies in the **Date** dimension. These user hierarchies are all natural hierarchies.

Defining Attribute Relationships for Attributes in the Customer Geography Hierarchy

1. Switch to Dimension Designer for the Customer dimension, and then click the **Dimension Structure** tab.

   In the **Hierarchies** pane, notice the levels in the **Customer Geography** user-defined hierarchy. This hierarchy is currently just a drill-down path for users, as no relationship between levels or attributes have been defined.
2. Click the **Attribute Relationships** tab.

   Notice the four attribute relationships that link the non-key attributes from the **Geography** table to the key attribute from the **Geography** table. The **Geography** attribute is related to the **Full Name** attribute. The **Postal Code** attribute is indirectly linked to the **Full Name** attribute through the **Geography** attribute, because the **Postal Code** is linked to the **Geography** attribute and the **Geography** attribute is linked to the **Full Name** attribute. Next, we will change the attribute relationships so that they do not use the **Geography** attribute.
3. In the diagram, right-click the **Full Name** attribute and then select **New Attribute Relationship**.
4. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Full Name**. Set the **Related Attribute** to **Postal Code**. In the **Relationship type** list, leave the relationship type set to **Flexible** because relationships between the members might change over time.
5. Click .

   A warning icon appears in the diagram because the relationship is redundant. The relationship **Full Name** -> **Geography**-> **Postal Code** already existed, and you just created the relationship **Full Name** -> **Postal Code**. The relationship **Geography**-> **Postal Code** is now redundant, so we will remove it.
6. In the **Attribute Relationships** pane, right-click **Geography**-> **Postal Code** and then click **Delete**.

7. When the **Delete Objects** dialog box appears, click **OK**.
8. In the diagram, right-click the **Postal Code** attribute and then select **New Attribute Relationship**.
9. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Postal Code**. Set the **Related Attribute** to **City**. In the **Relationship type** list, leave the relationship type set to **Flexible**.
10. Click .

    The relationship **Geography**-> **City** is now redundant so we will delete it.
11. In the Attribute Relationships pane, right-click **Geography**-> **City** and then click **Delete**.
12. When the **Delete Objects** dialog box appears, click **OK**.
13. In the diagram, right-click the **City** attribute and then select **New Attribute Relationship**.
14. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **City**. Set the **Related Attribute** to **State-Province**. In the **Relationship type** list, set the relationship type to **Rigid** because the relationship between a city and a state will not change over time.
15. Click .
16. Right-click the arrow between **Geography** and **State-Province** and then click **Delete**.
17. When the **Delete Objects** dialog box appears, click **OK**.
18. In the diagram, right-click the **State-Province** attribute and then select **New Attribute Relationship**.
19. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **State-Province**. Set the **Related Attribute** to **Country-Region**. In the **Relationship type** list, set the relationship type to **Rigid** because the relationship between a state-province and a country-region will not change over time.
20. Click .
21. In the Attribute Relationships pane, right-click **Geography**-> **Country-Region** and then click **Delete**.
22. When the **Delete Objects** dialog box appears, click **OK**.
23. Click the **Dimension Structure** tab.

    Notice that when you delete the last attribute relationship between **Geography** and other attributes, that **Geography** itself is deleted. This is because the attribute is no longer used.
24. On the File menu, click **Save All**.

Defining Attribute Relationships for Attributes in the Sales Territory Hierarchy

1. Open Dimension Designer for the **Sales Territory** dimension, and then click the **Attribute Relationships** tab.
2. In the diagram, right-click the **Sales Territory Country** attribute and then select **New Attribute Relationship**.
3. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Sales Territory Country**. Set the **Related Attribute** to **Sales Territory Group**. In the **Relationship type** list, leave the relationship type set to **Flexible**.
4. Click .

   Sales Territory Group is now linked to **Sales Territory Country**, and **Sales Territory Country** is now linked to **Sales Territory Region**. The **RelationshipType** property for each of these relationships is set to **Flexible** because the groupings of regions within a country might change over time and because the groupings of countries into groups might change over time.

Defining Attribute Relationships for Attributes in the Product Model Lines Hierarchy

1. Open Dimension Designer for the **Product** dimension, and then click the **Attribute Relationships** tab.
2. In the diagram, right-click the **Model Name** attribute and then select **New Attribute Relationship**.
3. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Model Name**. Set the **Related Attribute** to **Product Line**. In the **Relationship type** list, leave the relationship type set to **Flexible**.
4. Click .

Defining Attribute Relationships for Attributes in the Fiscal Date Hierarchy

1. Switch to Dimension Designer for the **Date** dimension, and then click the **Attribute Relationships** tab.
2. In the diagram, right-click the **Month Name** attribute and then select **New Attribute Relationship**.
3. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Month Name**. Set the **Related Attribute** to **Fiscal Quarter**. In the **Relationship type** list, set the relationship type to **Rigid**.

4. Click .
5. In the diagram, right-click the **Fiscal Quarter** attribute and then select **New Attribute Relationship**.
6. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Fiscal Quarter**. Set the **Related Attribute** to **Fiscal Semester**. In the **Relationship type** list, set the relationship type to **Rigid**.
7. Click .
8. In the diagram, right-click the **Fiscal Semester** attribute and then select **New Attribute Relationship**.
9. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Fiscal Semester**. Set the **Related Attribute** to **Fiscal Year**. In the **Relationship type** list, set the relationship type to **Rigid**.
10. Click .

Defining Attribute Relationships for Attributes in the Calendar Date Hierarchy

1. In the diagram, right-click the **Month Name** attribute and then select **New Attribute Relationship**.
2. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Month Name**. Set the **Related Attribute** to **Calendar Quarter**. In the **Relationship type** list, set the relationship type to **Rigid**.
3. Click .
4. In the diagram, right-click the **Calendar Quarter** attribute and then select **New Attribute Relationship**.
5. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Calendar Quarter**. Set the **Related Attribute** to **Calendar Semester**. In the **Relationship type** list, set the relationship type to **Rigid**.
6. Click .
7. In the diagram, right-click the **Calendar Semester** attribute and then select **New Attribute Relationship**.
8. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Calendar Semester**. Set the **Related Attribute** to **Calendar Year**. In the **Relationship type** list, set the relationship type to **Rigid**.
9. Click .

Defining Attribute Relationships for Attributes in the Geography Hierarchy

1. Open Dimension Designer for the Geography dimension, and then click the **Attribute Relationships** tab.
2. In the diagram, right-click the **Postal Code** attribute and then select **New Attribute Relationship**.
3. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Postal Code**. Set the **Related Attribute** to **City**. In the **Relationship type** list, set the relationship type to **Flexible**.
4. Click .
5. In the diagram, right-click the **City** attribute and then select **New Attribute Relationship**.
6. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **City**. Set the **Related Attribute** to **State-Province**. In the **Relationship type** list, set the relationship type to **Rigid**.
7. Click .
8. In the diagram, right-click the **State-Province** attribute and then select **New Attribute Relationship**.
9. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **State-Province**. Set the **Related Attribute** to **Country-Region**. In the **Relationship type** list, set the relationship type to **Rigid**.
10. Click .
11. In the diagram, right-click the **Geography Key** attribute and then select **Properties**.
12. Set the **AttributeHierarchyOptimizedState** property to **NotOptimized**, set the **AttributeHierarchyOrdered** property to **False**, and set the **AttributeHierarchyVisible** property to **False**.
13. On the **File** menu, click **Save All**.
14. On the **Build** menu of SQL Server Data Tools (SSDT), click **Deploy Analysis Services Tutorial**.

Next Task in Lesson

Defining Unknown Member and Null Processing Properties

See Also

Defining and Configuring an Attribute Relationship

Configuring User-defined Hierarchy Properties

Defining the Unknown Member and Null Processing Properties

When Analysis Services processes a dimension, all the distinct values from the underlying columns in the tables, or views in the data source view, populate the attributes in the dimension. If Analysis Services encounters a null value during processing, by default, it converts this null to a zero for numeric columns or to an empty string for string columns. You can modify the default settings or convert null values in your extract, transform, and load process (if any) of the underlying relational data warehouse. Additionally, you can have Analysis Services convert the null value to a designated value by configuring three properties: the **UnknownMember** and **UnknownMemberName** properties for the dimension, and the **NullProcessing** property for the dimension's key attribute.

The Dimension Wizard and the Cube Wizard will enable these properties for you based on whether the key attribute of a dimension is nullable or the root attribute of a snowflake dimension is based on a nullable column. In these cases, the **NullProcessing** property of the key attribute will be set to **UnknownMember** and the **UnknownMember** property will be set to **Visible**.

However, when you build snowflaked dimensions incrementally, as we are doing with the Product dimension in this tutorial, or when you define dimensions using Dimension Designer and then incorporate these existing dimensions into a cube, the **UnknownMember** and **NullProcessing** properties might need to be set manually.

In the tasks in this topic, you will add the product category and product subcategory attributes to the Product dimension from snowflaked tables that you will add to the Adventure Works DW data source view. You will then enable the **UnknownMember** property for the Product dimension, specify **Assembly Components** as the value for the **UnknownMemberName** property, relate the **Subcategory** and **Category** attributes to the product name attribute, and then define custom error handling for the member key attribute that links the snowflaked tables.

Note
If you have added the Subcategory and Category attributes when you originally defined the Analysis Services Tutorial cube using the Cube Wizard, these steps would have been performed for you automatically.

Reviewing Error Handling and Unknown Member Properties in the Product Dimension

1. Switch to Dimension Designer for the **Product** dimension, click the **Dimension Structure** tab, and then select **Product** in the **Attributes** pane.

   This enables you to view and modify the properties of the dimension itself.

2. In the Properties window, review the **UnknownMember** and **UnknownMemberName** properties.

   Notice that the **UnknownMember** property is not enabled, because its value is set to **None** instead of **Visible** or **Hidden**, and that no name is specified for the **UnknownMemberName** property.

3. In the Properties window, select **(custom)** in the **ErrorConfiguration** property cell, and then expand the **ErrorConfiguration** properties collection.

   Setting the **ErrorConfiguration** property to **(custom)** allows you to view the default error configuration settings - it does not change any settings.

4. Review the key and null key error configuration properties, but do not make any changes.

   Notice that, by default, when null keys are converted to the unknown member and the processing error associated with this conversion is ignored.

   The following image shows the property settings for the **ErrorConfiguration** properties collection.

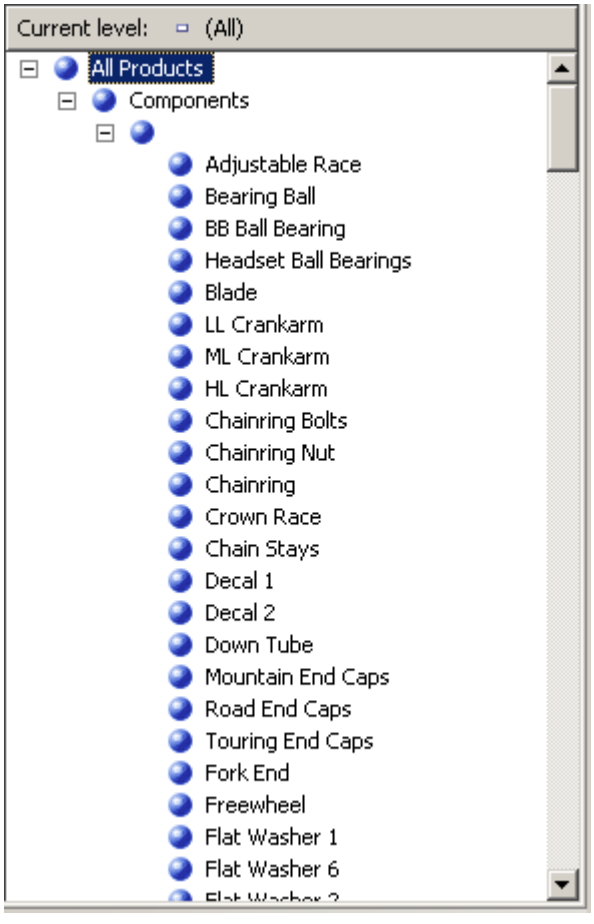
ErrorConfiguration	(custom)
KeyDuplicate	IgnoreError
KeyErrorAction	ConvertToUnknown
KeyErrorLimit	0
KeyErrorLimitAction	StopProcessing
KeyErrorLogFile	
KeyNotFound	ReportAndContinue
NullKeyConvertedToUnknown	IgnoreError
NullKeyNotAllowed	ReportAndContinue

5. Click the **Browser** tab, verify that **Product Model Lines** is selected in the **Hierarchy** list, and then expand **All Products**.

   Notice the five members of the Product Line level.

6. Expand **Components**, and then expand the unlabeled member of the **Model Name** level.

   This level contains the assembly components that are used when building other components, starting with the **Adjustable Race** product, as shown in the following image.



Defining Attributes from Snowflaked Tables and a Product Category User-Defined Hierarchy

1. Open Data Source View Designer for the Adventure Works DW data source view, select **Reseller Sales** in the **Diagram Organizer** pane, and then click **Add/Remove Objects** on the **Data Source View** menu of SQL Server Data Tools (SSDT).

   The **Add/Remove Tables** dialog box opens.
2. In the **Included objects** list, select **DimProduct (dbo)**, and then click **Add Related Tables**.

   Both **DimProductSubcategory (dbo)** and **FactProductInventory (dbo)** are added. Remove **FactProductInventory (dbo)** so that just the **DimProductSubcategory (dbo)** table is added to the **Included objects** list.
3. With the **DimProductSubcategory (dbo)** table selected by default as the table

most recently added, click **Add Related Tables** again.

The **DimProductCategory (dbo)** table is added to the **Included objects** list.

4. Click **OK**.
5. On the **Format** menu of SQL Server Data Tools, point to **Auto Layout**, and then click **Diagram**.

   Notice that the **DimProductSubcategory (dbo)** table and **DimProductCategory (dbo)** table are linked to each other, and also to the **ResellerSales** table through the **Product** table.
6. Switch to Dimension Designer for the **Product** dimension, and then click the **Dimension Structure** tab.
7. Right-click anywhere in the **Data Source View** pane, and then click **Show All Tables**.
8. In the **Data Source View** pane, locate the **DimProductCategory** table, right-click **ProductCategoryKey** in that table, and then click **New Attribute from Column**.
9. In the **Attributes** pane, change the name of this new attribute to **Category**.
10. In the Properties window, click in the **NameColumn** property field and then click the browse (**...**) button to open the **Name Column** dialog box.
11. Select **EnglishProductCategoryName** in the **Source column** list and then click **OK**.
12. In the **Data Source View** pane, locate the **DimProductSubcategory** table, right-click **ProductSubcategoryKey** in that table, and then click **New Attribute from Column**.
13. In the **Attributes** pane, change the name of this new attribute to **Subcategory**.
14. In the Properties window, click in the **NameColumn** property field and then click the browse **(...)** button to open the **Name Column** dialog box.
15. Select **EnglishProductSubcategoryName** in the **Source column** list and then click **OK**.
16. Create a new user-defined hierarchy called **Product Categories** with the following levels, in order from top to bottom: **Category**, **Subcategory**, and **Product Name**.
17. Specify **All Products** as the value for the **AllMemberName** property of the Product Categories user-defined hierarchy.

Browsing the User-Defined Hierarchies in the Product Dimension

1. On the toolbar of the **Dimension Structure** tab of **Dimension Designer** for the **Product** dimension, click **Process**.
2. Click **Yes** to build and deploy the project, and then click **Run** to process the

Product dimension.

3. When processing has succeeded, expand **Processing Dimension 'Product' completed successfully** in the **Process Progress** dialog box, expand **Processing Dimension Attribute 'Product Name' completed**, and then expand **SQL queries 1**.
4. Click the SELECT DISTINCT query and then click **View Details**.

   Notice that a WHERE clause has been added to the SELECT DISTINCT clause that removes those products that have no value in the ProductSubcategoryKey column, as shown in the following image.

5. Click **Close** three times to close all processing dialog boxes.
6. Click the **Browser** tab in Dimension Designer for the **Product** dimension, and then click **Reconnect**.
7. Verify that **Product Model Lines** appears in the **Hierarchy** list, expand **All Products**, and then expand **Components**.
8. Select **Product Categories** in the **Hierarchy** list, expand **All Products**, and then expand **Components**.

   Notice that none of the assembly components appear.

To modify the behavior mentioned in the previous task, you will enable the **UnknownMember** property of the Products dimension, set a value for the **UnknownMemberName** property, set the **NullProcessing** property for the **Subcategory** and **Model Name** attributes to **UnknownMember**, define the **Category** attribute as a related attribute of the **Subcategory** attribute, and then define the **Product Line** attribute as a related attribute of the **Model Name** attribute. These steps will cause Analysis Services to use the unknown member name value for each product that does not have a value for the **SubcategoryKey** column, as you will see in the following task.

Enabling the Unknown Member, Defining Attribute Relationships, and Specifying Custom Processing Properties for Nulls

1. Click the **Dimension Structure** tab in Dimension Designer for the **Product** dimension, and then select **Product** in the **Attributes** pane.
2. In the **Properties** window, change the **UnknownMember** property to **Visible**, and then change the value for the **UnknownMemberName** property to **Assembly Components**.

   Changing the **UnknownMember** property to either **Visible** or **Hidden** enables the **UnknownMember** property for the dimension.
3. Click the **Attribute Relationships** tab.
4. In the diagram, right-click the **Subcategory** attribute and then select **New Attribute Relationship**.
5. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Subcategory**. Set the **Related Attribute** to **Category**. Leave the relationship type set to **Flexible**.
6. Click .
7. In the **Attributes** pane, select **Subcategory.**
8. In the Properties window, expand the **KeyColumns** property and then expand the **DimProductSubcategory.ProductSubcategoryKey (Integer)** property.

9. Change the **NullProcessing** property to **UnknownMember**.
10. In the **Attributes** pane, select **Model Name**.
11. In the Properties window, expand the **KeyColumns** property and then expand the **Product.ModelName (WChar)** property.
12. Change the **NullProcessing** property to **UnknownMember**.

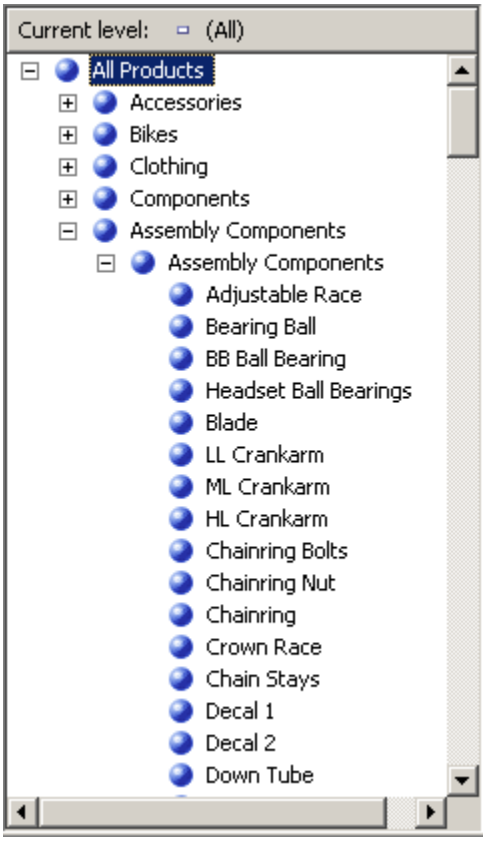
    Because of these changes, when Analysis Services encounters a null value for the **Subcategory** attribute or the **Model Name** attribute during processing, the unknown member value will be substituted as the key value, and the user-defined hierarchies will be constructed correctly.

Browsing the Product Dimension Again

1. On the **Build** menu, click **Deploy Analysis Services Tutorial**.
2. When deployment has successfully completed, click the **Browser** tab in Dimension Designer for the **Product** dimension, and then click **Reconnect**.
3. Verify that **Product Categories** is selected in the **Hierarchy** list, and then expand **All Products**.

   Notice that Assembly Components appears as a new member of the Category level.
4. Expand the **Assembly Components** member of the **Category** level and then expand the **Assembly Components** member of the **Subcategory** level.

   Notice that all the assembly components now appear at the **Product Name** level, as shown in the following image.



Next Lesson

Lesson 5: Defining Relationships Between Dimensions and Measure Groups

Lesson 5: Defining Relationships Between Dimensions and Measure Groups

In the previous lessons in this tutorial, you learned that database dimensions added to a cube can be used as the basis for one or more cube dimensions. In this lesson, you learn to define different types of relationships between cube dimensions and measure groups, and to specify the properties of these relationships.

For more information, see Dimension Relationships (SSAS) and Defining and Configuring Dimension Usage and Dimension Relationships.

Note

Completed projects for all of the lessons in this tutorial are available online. You can jump ahead to any lesson by using the completed project from the previous

lesson as a starting point. Click here to download the sample projects that go with this tutorial.

This lesson contains the following tasks:

Defining a Referenced Relationship

In this task, you learn to link a dimension to a fact table indirectly through a dimension that is linked directly through a primary key–foreign key relationship.

Defining a Fact Relationship

In this task, you learn to define a dimension based on data in the fact table, and to define the dimension relationship as a fact relationship.

Defining a Many-to-Many Relationship

In this task, you learn to relate a fact to multiple dimension members through the definition of a many-to-many relationship between dimension tables and fact tables.

Defining Dimension Granularity within a Measure Group

In this task, you learn to modify the granularity of a dimension for a specific measure group.

Next Lesson

Lesson 6: Defining Calculations

See Also

Analysis Services Tutorial Scenario
SQL Server 2005 Analysis Services Tutorial
Dimension Relationships (SSAS)
Defining and Configuring Dimension Usage and Dimension Relationships

Defining a Referenced Relationship

Up to this point in the tutorial, each cube dimension that you defined was based on a table that was directly linked to the fact table for a measure group by a primary key to foreign key relationship. In the tasks in this topic, you link the **Geography** dimension to the fact table for reseller sales through the **Reseller** dimension, which is called a *reference dimension*. This enables users to dimension reseller sales by geography. For more information, see Defining a Referenced Relationship and Referenced Relationship Properties.

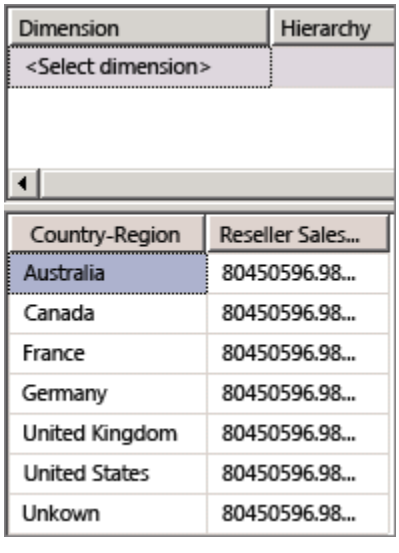
Dimensioning Reseller Sales by Geography

1. In Solution Explorer, right-click **Analysis Services Tutorial** in the **Cubes** folder,

and then click **Browse**.

2. Remove all hierarchies from the data pane, and then verify that the **Reseller Sales-Sales Amount** measure appears in the data area of the data pane. Add it to the data pane if it is not already there.
3. From the **Geography** dimension in the metadata pane, drag the **Geographies** user-defined hierarchy to the **Drop Row Fields Here** area of the data pane.

   Notice that the **Reseller Sales-Sales Amount** measure is not correctly dimensioned by the **Country-Region** attribute members in the **Regions** hierarchy. The value for **Reseller Sales-Sales Amount** repeats for each **Country-Region** attribute member.



Dimension	Hierarchy
<Select dimension>	

Country-Region	Reseller Sales...
Australia	80450596.98...
Canada	80450596.98...
France	80450596.98...
Germany	80450596.98...
United Kingdom	80450596.98...
United States	80450596.98...
Unkown	80450596.98...

4. Open Data Source View Designer for the **Adventure Works DW 2012** data source view.
5. In the **Diagram Organizer** pane, view the relationship between the **Geography** table and the **ResellerSales** table.

   Notice that there is no direct link between these tables. However, there is an indirect link between these tables through either the **Reseller** table or the **SalesTerritory** table.
6. Double-click the arrow that represents the relationship between the **Geography** table and the **Reseller** table.

   In the **Edit Relationship** dialog box, notice that the **GeographyKey** column is the primary key in the **Geography** table and the foreign key in the **Reseller** table.
7. Click **Cancel**, switch to Cube Designer for the Analysis Services Tutorial cube, and then click the **Dimension Usage** tab.

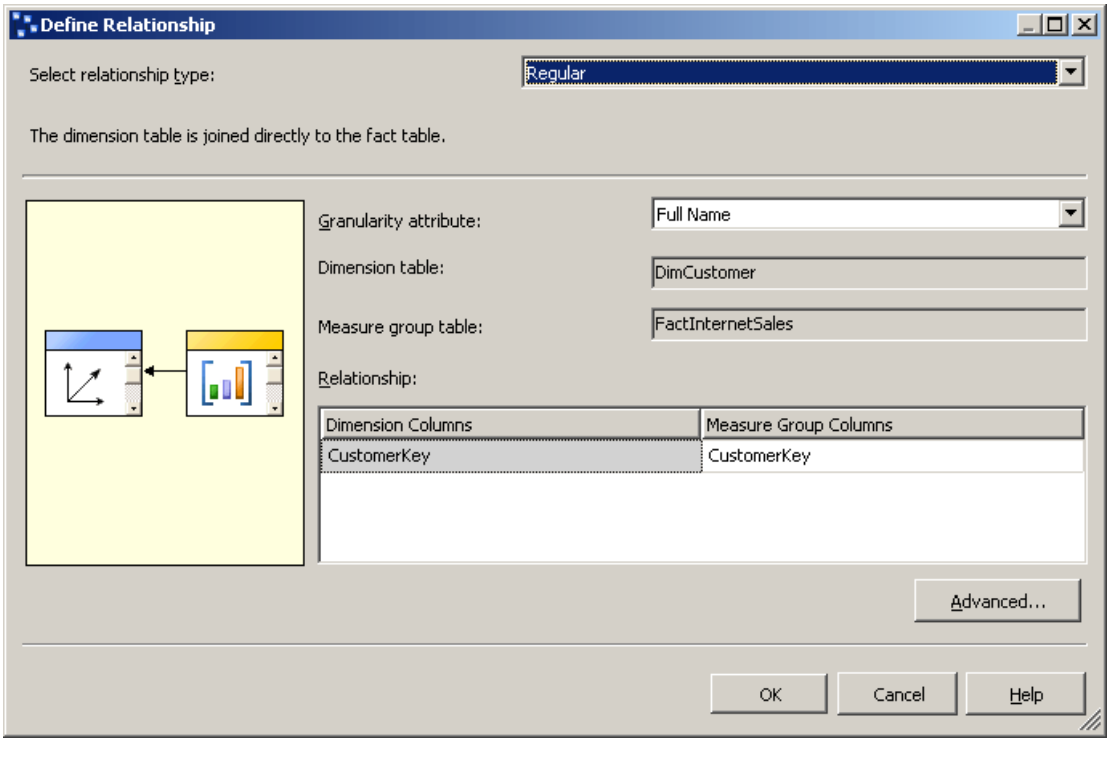
   Notice that the **Geography** cube dimension does not currently have a relationship with either the **Internet Sales** measure group or the **Reseller Sales**

measure group.

8. Click the ellipsis button (**...**) in the **Full Name** cell at the intersection of the **Customer** dimension and the **Internet Sales** measure group.

   In the **Define Relationship** dialog box, notice that a **Regular** relationship is defined between the **DimCustomer** dimension table and the **FactInternetSales** measure group table based on the **CustomerKey** column in each of these tables. All the relationships that you have defined within this tutorial up to this point have been regular relationships.

   The following image shows the **Define Relationship** dialog box with a regular relationship between the **DimCustomer** dimension table and the **FactInternetSales** measure group table.



9. Click **Cancel**.
10. Click the ellipsis button (**...**) in the unnamed cell at the intersection of the **Geography** dimension and the **Reseller Sales** measure group.

    In the **Define Relationship** dialog box, notice that no relationship is currently defined between the Geography cube dimension and the Reseller Sales measure group. You cannot define a regular relationship because there is no direct relationship between the dimension table for the Geography dimension and the fact table for the Reseller Sales measure group.
11. In the **Select relationship type** list, select **Referenced**.

You define a referenced relationship by specifying a dimension that is directly connected to the measure group table, called an *intermediate dimension*, that Analysis Services can use to link the reference dimension to the fact table. You then specify the attribute that links the reference dimension to the intermediate dimension.

12. In the **Intermediate dimension** list, select **Reseller**.

    The underlying table for the Geography dimension is linked to the fact table through the underlying table for the Reseller dimension.

13. In the **Reference dimension attribute** list, select **Geography Key**, and then try to select **Geography Key** in the **Intermediate dimension attribute** list.

    Notice that **Geography Key** does not appear in the **Intermediate dimension attribute** list. This is because the **GeographyKey** column is not defined as an attribute in the **Reseller** dimension.

14. Click **Cancel**.

    In the next task, you will solve this problem by defining an attribute that is based on the GeographyKey column in the Reseller dimension.

Defining the Intermediate Dimension Attribute and the Referenced Dimension Relationship

1. Open Dimension Designer for the **Reseller** dimension, and view the columns in the **Reseller** table in the **Data Source View** pane, and view the defined attributes in the **Reseller** dimension in the **Attributes** pane.

   Notice that although GeographyKey is defined as a column in the Reseller table, no dimension attribute is defined in the Reseller dimension based on this column. Geography is defined as a dimension attribute in the Geography dimension because it is the key column that links the underlying table for that dimension to the fact table.

2. To add a **Geography Key** attribute to the **Reseller** dimension, right-click **GeographyKey** in the **Data Source View** pane, and then click **New Attribute from Column**.

3. In the **Attributes** pane, select **Geography Key**, and then, in the Properties window, set the **AttributeHierarchyOptimizedState** property to **NotOptimized**, the **AttributeHierarchyOrdered** property to **False**, and the **AttributeHierarchyVisible** property to **False.**

   The Geography Key attribute in the Reseller dimension will only be used to link the Geography dimension to the Reseller Sales fact table. Because it will not be used for browsing, there is no value in defining this attribute hierarchy as visible. Additionally, ordering and optimizing the attribute hierarchy will only negatively

affect processing performance. However, the attribute must be enabled to serve as the link between the two dimensions.

4. Switch to Cube Designer for the Analysis Services Tutorial cube, click the **Dimension Usage** tab, and then click the ellipsis button (**...**) at the intersection of the **Reseller Sales** measure group and the **Geography** cube dimension.
5. In the **Select relationship type** list, select **Referenced**.
6. In the **Intermediate dimension** list, select **Reseller**.
7. In the **Reference dimension attribute** list, select **Geography Key**, and then select **Geography Key** in the **Intermediate dimension attribute** list.

   Notice that the **Materialize** check box is selected. This is the default setting for MOLAP dimensions. Materializing the dimension attribute link causes the value of the link between the fact table and the reference dimension for each row to be materialized, or stored, in the dimension's MOLAP structure during processing. This will have a minor effect on processing performance and storage requirements, but will increase query performance (sometimes significantly).
8. Click **OK**.

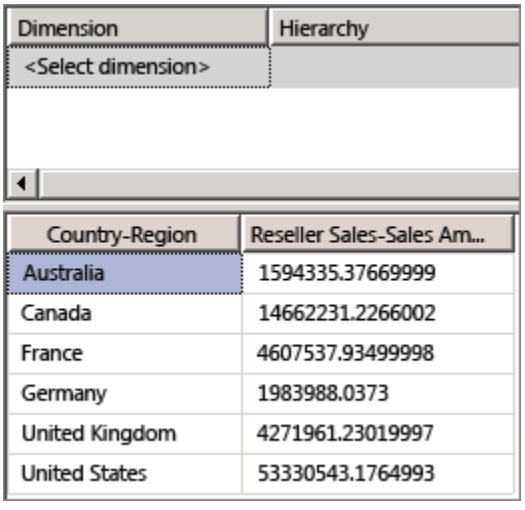
   Notice that the **Geography** cube dimension is now linked to the **Reseller Sales** measure group. The icon indicates that the relationship is a referenced dimension relationship.
9. In the **Dimensions** list on the **Dimension Usage** tab, right-click **Geography**, and then click **Rename**.
10. Change the name of this cube dimension to **Reseller Geography**.

    Because this cube dimension is now linked to the **Reseller Sales** measure group, users will benefit from explicitly defining its use in the cube, to avoid possible user confusion.

Successfully Dimensioning Reseller Sales by Geography

1. On the **Build** menu, click **Deploy Analysis Services Tutorial**.
2. When deployment has successfully completed, click the **Browser** tab in Cube Designer for the Analysis Services Tutorial cube, and then click the **Reconnect** button.
3. In the metadata pane, expand **Reseller Geography**, right-click **Geographies**, and then click **Add to Row Area**.

   Notice that the **Reseller Sales-Sales Amount** measure is now correctly dimensioned by the **Country-Region** attribute of the **Geographies** user-defined hierarchy, as shown in the following image.



Next Task in Lesson

Defining a Fact Relationship

See Also

Attribute Relationships

Defining a Referenced Relationship and Referenced Relationship Properties

Defining a Fact Relationship

Users sometimes want to be able to dimension measures by data items that are in the fact table or to query the fact table for specific additional related information, such as invoice numbers or purchase order numbers related to specific sales facts. When you define a dimension based on such a fact table item, the dimension is called a *fact dimension*. Fact dimensions are also known as degenerate dimensions. Fact dimensions are useful for grouping together related fact table rows, such as all the rows that are related to a particular invoice number. Although you can put this information in a separate dimension table in the relational database, creating a separate dimension table for the information provides no benefit because the dimension table would grow at the same rate as the fact table, and would just create duplicate data and unnecessary complexity.

Within Analysis Services, you can determine whether to duplicate the fact dimension data in a MOLAP dimension structure for increased query performance, or whether to define the fact dimension as a ROLAP dimension to save storage space at the expense of query performance. When you store a dimension with the MOLAP storage mode, all the dimension members are stored in the instance of Analysis Services in a highly compressed MOLAP structure, in addition to being stored in the measure group's partitions. When you store a dimension with the ROLAP storage mode, only the

dimension definition is stored in the MOLAP structure—the dimension members themselves are queried from the underlying relational fact table at query time. You decide the appropriate storage mode based on how frequently the fact dimension is queried, the number of rows returned by a typical query, the performance of the query, and the processing cost. Defining a dimension as ROLAP does not require that all cubes that use the dimension also be stored with the ROLAP storage mode. The storage mode for each dimension can be configured independently.

When you define a fact dimension, you can define the relationship between the fact dimension and the measure group as a fact relationship. The following constraints apply to fact relationships:

- The granularity attribute must be the key column for the dimension, which creates a one-to-one relationship between the dimension and the facts in the fact table.
- A dimension can have a fact relationship with only a single measure group.

Note
Fact dimensions must be incrementally updated after every update to the measure group that the fact relationship references.

For more information, see Dimension Relationships, and Defining a Fact Relationship and Fact Relationship Properties.

In the tasks in this topic, you add a new cube dimension based on the **CustomerPONumber** column in the **FactInternetSales** fact table. You then define the relationship between this new cube dimension and the **Internet Sales** measure group as a fact relationship.

Defining the Internet Sales Orders Fact Dimension

1. In Solution Explorer, right-click **Dimensions**, and then click **New Dimension**.
2. On the **Welcome to the Dimension Wizard** page, click **Next**.
3. On the **Select Creation Method** page, verify that the **Use an existing table** option is selected, and then click **Next**.
4. On the **Specify Source Information** page, verify that the **Adventure Works DW 2012** data source view is selected.
5. In the **Main table** list, select **InternetSales**.
6. In the **Key columns** list, verify that **SalesOrderNumber** and **SalesOrderLineNumber** are listed.
7. In the **Name column** list, select **SalesOrderLineNumber**.
8. Click **Next**.
9. On the **Select Related Tables** page, clear the check boxes beside all of the tables, and then click **Next**.

10. On the **Select Dimension Attributes** page, click the check box in the header twice to clear all of the check boxes. The **Sales Order Number** attribute will remain selected because it is the key attribute.
11. Select the **Customer PO Number** attribute, and then click **Next**.
12. On the **Completing the Wizard** page, change the name to **Internet Sales Order Details** and then click **Finish** to complete the wizard.
13. On the **File** menu, click **Save All**.
14. In the **Attributes** pane of the Dimension Designer for the **Internet Sales Order Details** dimension, select **Sales Order Number**, and then change the **Name** property in the Properties window to **Item Description.**
15. In the **NameColumn** property cell, click the browse button **(...)**. In the **Name Column** dialog box, select **Product** from the **Source table** list, select **EnglishProductName** for the **Source column**, and then click **OK**.
16. Add the **Sales Order Number** attribute to the dimension by dragging the **SalesOrderNumber** column from the **InternetSales** table in the **Data Source View** pane to the **Attributes** pane.
17. Change the **Name** property of the new **Sales Order Number** attribute to **Order Number**, and change the **OrderBy** property to **Key**.
18. In the **Hierarchies** pane, create an **Internet Sales Orders** user hierarchy that contains the **Order Number** and **Item Description** levels, in that order.
19. In the **Attributes** pane, select **Internet Sales Order Details**, and then review the value for the **StorageMode** property in the Properties window.

    Notice that, by default, this dimension is stored as a MOLAP dimension. Although changing the storage mode to ROLAP will save processing time and storage space, it occurs at the expense of query performance. For the purposes of this tutorial, you will use MOLAP as the storage mode.
20. To add the newly created dimension to the Analysis Services Tutorial cube as a cube dimension, switch to **Cube Designer**. On the **Cube Structure** tab, right-click in the **Dimensions** pane and select **Add Cube Dimension**.
21. In the **Add Cube Dimension**.dialog box, select **Internet Sales Order Details** and then click **OK**.

Defining a Fact Relationship for the Fact Dimension

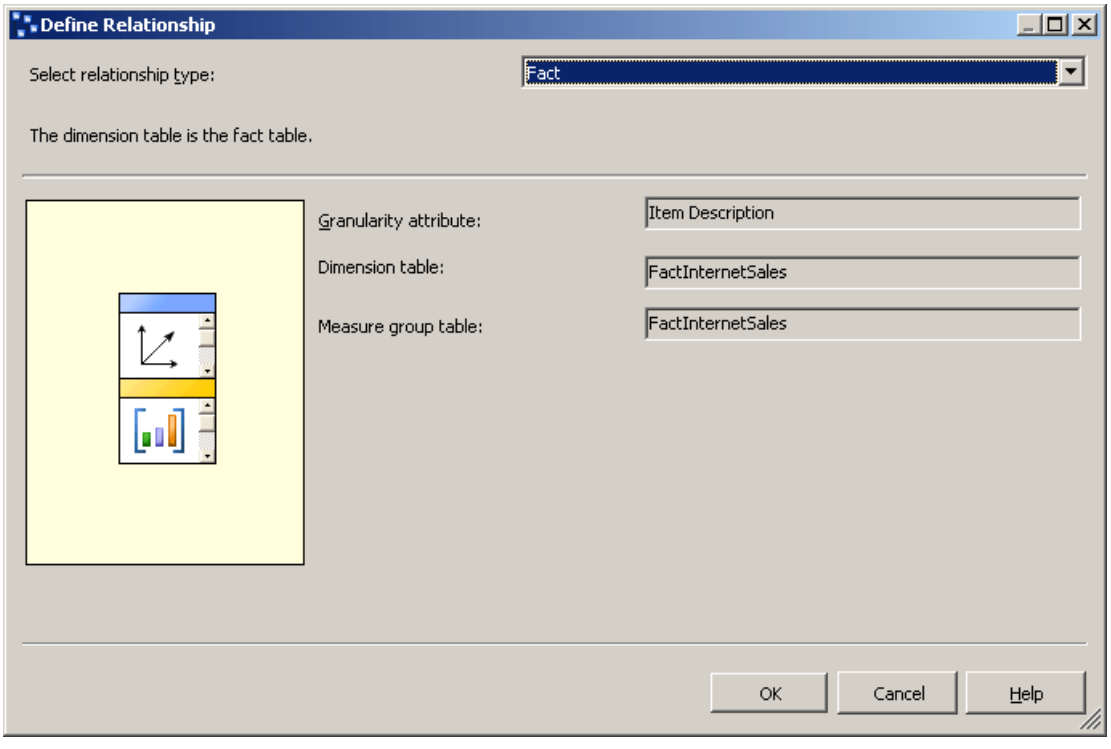
1. In the Cube Designer for the Analysis Services Tutorial cube, click the **Dimension Usage** tab.

   Notice that the **Internet Sales Order Details** cube dimension is automatically configured as having a fact relationship, as shown by the unique icon.
2. Click the browse button (**...**) in the **Item Description** cell, at the intersection of

the **Internet Sales** measure group and the **Internet Sales Order Details** dimension, to review the fact relationship properties.

The **Define Relationship** dialog box opens. Notice that you cannot configure any of the properties.

The following image shows the fact relationship properties in the **Define Relationship** dialog box.



3. Click **Cancel**.

Browsing the Cube by Using the Fact Dimension

1. On the **Build** menu, click **Deploy Analysis Services Tutorial** to deploy the changes to the instance of Analysis Services and process the database.
2. After deployment has successfully completed, click the **Browser** tab in Cube Designer for the Analysis Services Tutorial cube, and then click the **Reconnect** button.
3. Clear all measures and hierarchies from the data pane, and then add the **Internet Sales-Sales Amount** measure to the data area of the data pane.
4. In the metadata pane, expand **Customer**, expand **Location**, expand **Customer Geography**, expand **Members**, expand **All Customers**, expand **Australia**,

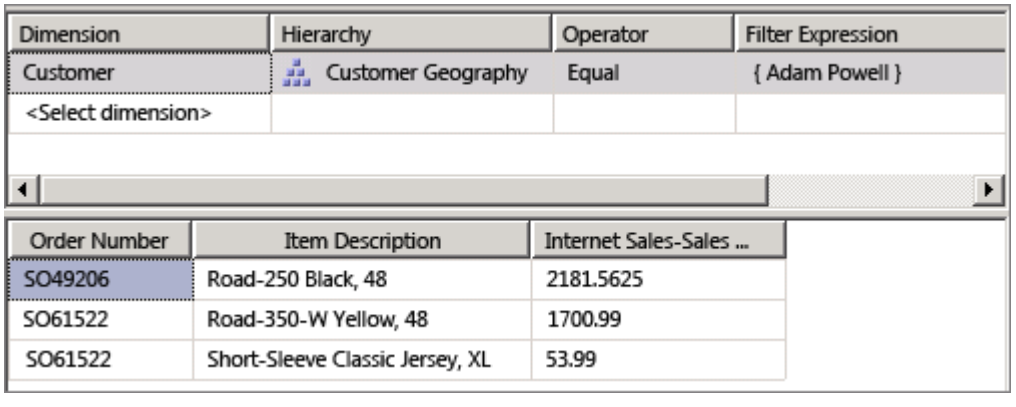
expand **Queensland**, expand **Brisbane**, expand **4000**, right-click **Adam Powell**, and then click **Add to Filter**.

Filtering to limit the sales orders returned to a single customer lets the user drill down to the underlying detail in a large fact table without suffering a significant loss in query performance.

5. Add the **Internet Sales Orders** user-defined hierarchy from the **Internet Sales Order Details** dimension to the row area of the data pane.

   Notice that the sales order numbers and the corresponding Internet sales amounts for Adam Powell appear in the data pane.

   The following image shows the result of the previous steps.



Dimension	Hierarchy	Operator	Filter Expression
Customer	Customer Geography	Equal	{ Adam Powell }
<Select dimension>			

Order Number	Item Description	Internet Sales-Sales ...
SO49206	Road-250 Black, 48	2181.5625
SO61522	Road-350-W Yellow, 48	1700.99
SO61522	Short-Sleeve Classic Jersey, XL	53.99

Next Task in Lesson

Defining a Many-to-Many Relationship

See Also

Dimension Relationships

Defining a Fact Relationship and Fact Relationship Properties

Defining a Many-to-Many Relationship

When you define a dimension, typically each fact joins to one and only one dimension member, whereas a single dimension member can be associated with many different facts. For example, each customer can have many orders but each order belongs to a single customer. In relational database terminology, this is referred to as a *one-to-many relationship*. However, sometimes a single fact can join to multiple dimension members. In relational database terminology, this is referred to as a *many-to-many relationship*. For example, a customer may have multiple reasons for making a purchase, and a purchase reason can be associated with multiple purchases. A join table is used to define the sales reasons that relate to each purchase. A Sales Reason dimension constructed from such relationships would then have multiple members that relate to a single sales transaction.

Many-to-many dimensions expand the dimensional model beyond the classic star schema and support complex analytics when dimensions are not directly related to a fact table.

In Analysis Services, you define a many-to-many relationship between a dimension and a measure group by specifying an intermediate fact table that is joined to the dimension table. An intermediate fact table is joined, in turn, to an intermediate dimension table to which the fact table is joined. The many-to-many relationships between the intermediate fact table and both the dimension tables in the relationship and the intermediate dimension creates the many-to-many relationships between members of the primary dimension and measures in the measure group that is specified by the relationship. In order to define a many-to-many relationship between a dimension and a measure group through an intermediate measure group, the intermediate measure group must share one or more dimensions with the original measure group.

With a many-to-many dimension, values are distinct summed, which means that they do not aggregate more than once to the All member.

Note

In order to support a many-to-many dimension relationship, a primary key–foreign key relationship must be defined in the data source view between all the tables that are involved. Otherwise, you will not be able to select the correct intermediate measure group when you establish the relationship in the **Dimension Usage** tab of Cube Designer.

For more information, see Dimension Relationships, and Defining a Many-to-Many Relationship and Many-to-Many Relationship Properties.

In the tasks in this topic, you define the Sales Reasons dimension and the Sales Reasons measure group, and you define a many-to-many relationship between the Sales Reasons dimension and the Internet Sales measure group through the Sales Reasons measure group.

Adding Required Tables to the Data Source View

1. Open Data Source View Designer for the **Adventure Works DW 2012** data source view.
2. Right-click anywhere in the **Diagram Organizer** pane, click **New Diagram**, and specify **Internet Sales Order Reasons** as the name for this new diagram.
3. Drag the **InternetSales** table to the **Diagram** pane from the **Tables** pane.
4. Right-click anywhere in the **Diagram** pane, and then click **Add/Remove Tables**.
5. In the **Add/Remove Tables** dialog box, add the **DimSalesReason** table and the **FactInternetSalesReason** table to the **Included objects** list, and then click **OK**.

   Notice that the primary key–foreign key relationships between the tables that are

involved are established automatically because those relationships are defined in the underlying relational database. If these relationships were not defined in the underlying relational database, you would have to define them in the data source view.

6. On the **Format** menu, point to **Auto Layout**, and then click **Diagram**.
7. In the Properties window, change the **FriendlyName** property of the **DimSalesReason** table to **SalesReason**, and then change the **FriendlyName** property of the **FactInternetSalesReason** table to **InternetSalesReason**.
8. In the **Tables** pane, expand **InternetSalesReason (dbo.FactInternetSalesReason)**, click **SalesOrderNumber**, and then review the **DataType** property for this data column in the Properties window.

   Notice that the data type for the **SalesOrderNumber** column is a string data type.
9. Review the data types for the other columns in the **InternetSalesReason** table.

   Notice that the data types for the other two columns in this table are numeric data types.
10. In the **Tables** pane, right-click **InternetSalesReason (dbo.FactInternetSalesReason)**, and then click **Explore Data**.

    Notice that, for each line number within each order, a key value identifies the sales reason for the purchase of that line item, as shown in the following image.

SalesOrderNu	SalesOrderLin	SalesReasonK
SO43697	1	5
SO43697	1	9
SO43702	1	5
SO43702	1	9
SO43703	1	5
SO43703	1	9
SO43706	1	5
SO43706	1	9
SO43707	1	5
SO43707	1	9
SO43709	1	5
SO43709	1	9
SO43710	1	5
SO43710	1	9
SO43711	1	5
SO43711	1	9
SO43712	1	5

Defining the Intermediate Measure Group

1. Switch to Cube Designer for the Analysis Services Tutorial cube, and then click the **Cube Structure** tab.
2. Right-click anywhere in the **Measures** pane, and then click **New Measure Group**. For more information, see Defining and Configuring a Measure Group.
3. In the **New Measure Group** dialog box, select **InternetSalesReason** in the **Select a table from the data source view** list, and then click **OK**.

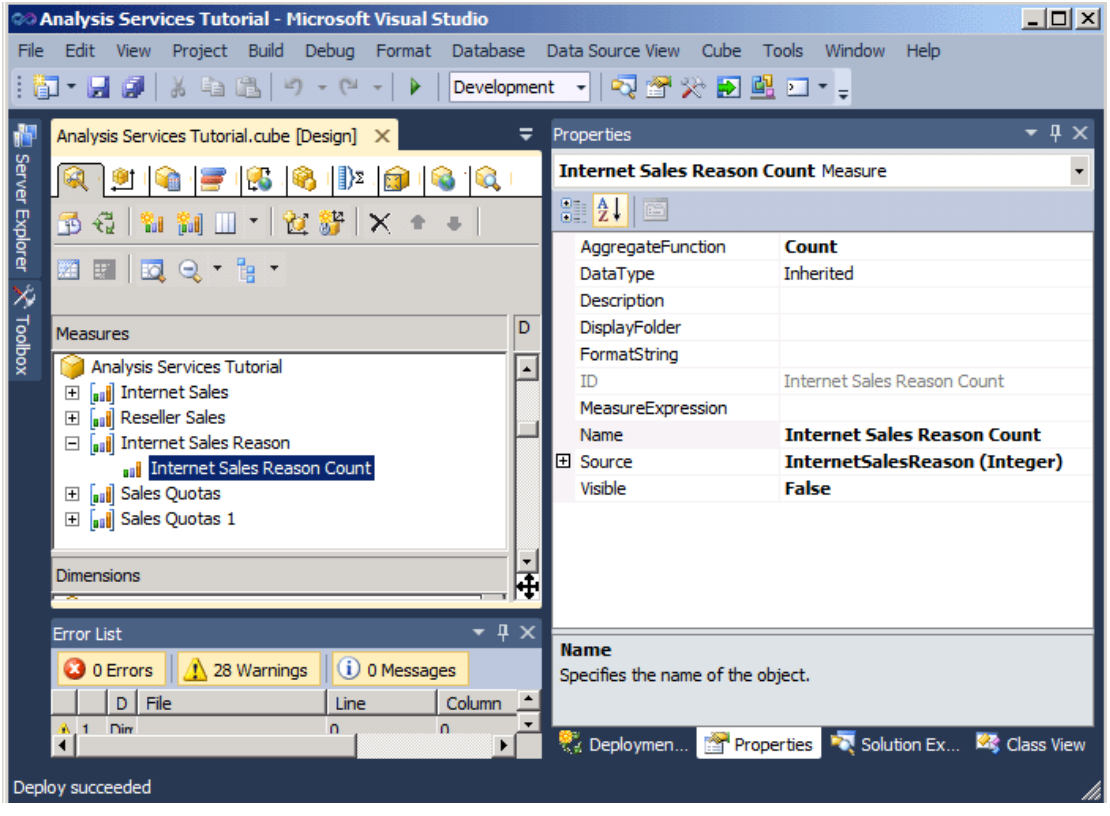
   Notice that the **Internet Sales Reason** measure group now appears in the **Measures** pane.
4. Expand the **Internet Sales Reason** measure group.

   Notice that only a single measure is defined for this new measure group, the **Internet Sales Reason Count** measure.
5. Select **Internet Sales Reason Count** and review the properties of this measure in the Properties window.

   Notice that the **AggregateFunction** property for this measure is defined as **Count** instead of **Sum**. Analysis Services chose **Count** because the underlying data type is a string data type. The other two columns in the underlying fact table were not selected as measures because Analysis Services detected them as numeric keys instead of as actual measures. For more information, see Defining Semiadditive Behavior.
6. In the Properties window, change the **Visible** property of the **Internet Sales Reason Count** measure to **False**.

   This measure will only be used to join the Sales Reason dimension that you will define next to the Internet Sales measure group. Users will not browse this measure directly.

   The following image shows the properties for the **Internet Sales Reason Count** measure.



Defining the Many-to-Many Dimension

1. In Solution Explorer, right-click **Dimensions**, and then click **New Dimension**.
2. On the **Welcome to the Dimension Wizard** page, click **Next**.
3. On the **Select Creation Method** page, verify that the **Use an existing table** option is selected, and then click **Next**.
4. On the **Specify Source Information** page, verify that the Adventure Works DW 2012 data source view is selected.
5. In the **Main table** list, select **SalesReason**.
6. In the **Key columns** list, verify that **SalesReasonKey** is listed.
7. In the **Name column** list, select **SalesReasonName**.
8. Click **Next**.
9. On the **Select Dimension Attributes** page, the **Sales Reason Key** attribute is automatically selected because it is the key attribute. Select the check box beside the **Sales Reason Reason Type** attribute, change its name to **Sales Reason Type**, and then click **Next**.

10. On the **Completing the Wizard** page, click **Finish** to create the Sales Reason dimension.
11. On the **File** menu, click **Save All**.
12. In the **Attributes** pane of the Dimension Designer for the **Sales Reason** dimension, select **Sales Reason Key**, and then change the **Name** property in the Properties window to **Sales Reason.**
13. In the **Hierarchies** pane of the Dimension Designer, create a **Sales Reasons** user hierarchy that contains the **Sales Reason Type** level and the **Sales Reason** level, in that order.
14. In the Properties window, define **All Sales Reasons** as the value for the **AllMemberName** property of the Sales Reasons hierarchy.
15. Define **All Sales Reasons** as the value for **AttributeAllMemberName** property of the Sales Reason dimension.
16. To add the newly created dimension to the Analysis Services Tutorial cube as a cube dimension, switch to **Cube Designer**. On the **Cube Structure** tab, right-click in the **Dimensions** pane and select **Add Cube Dimension**.
17. In the **Add Cube Dimension** dialog box, select **Sales Reason** and then click **OK**.
18. On the **File** menu, click **Save All**.

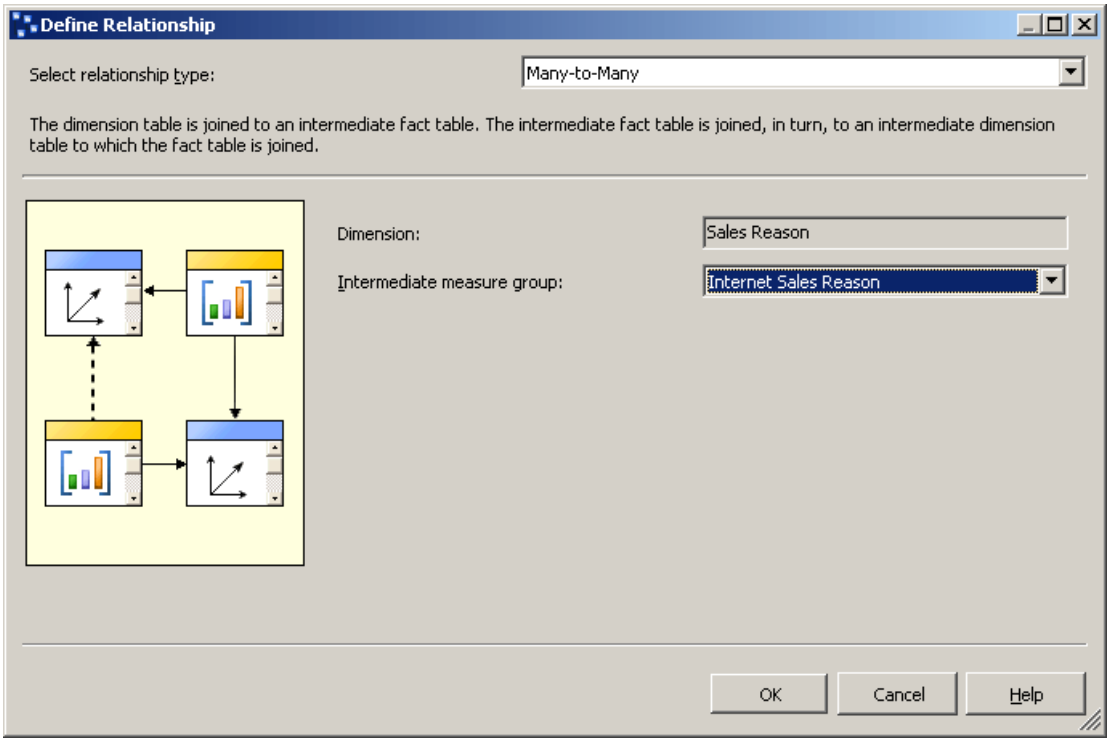
Defining the Many to Many Relationship

1. Switch to Cube Designer for the Analysis Services Tutorial cube, and then click the **Dimension Usage** tab.

   Notice that the **Sales Reason** dimension has a regular relationship defined with the **Internet Sales Reason** measure group, but has no relationship defined with the **Internet Sales** or **Reseller Sales** measure groups. Notice also that the **Internet Sales Order Details** dimension has a regular relationship defined with the **Internet Sales Reason** dimension, which in turn has a **Fact Relationship** with the **Internet Sales** measure group. If this dimension was not present (or another dimension with a relationship with both the **Internet Sales Reason** and the **Internet Sales** measure group were not present), you would not be able to define the many-to-many relationship.
2. Click the cell at the intersection of the **Internet Sales** measure group and the **Sales Reason** dimension and then click the browse button (**...**).
3. In the **Define Relationship** dialog box, select **Many-to-Many** in the **Select relationship type** list.

   You have to define the intermediate measure group that connects the Sales Reason dimension to the Internet Sales measure group.
4. In the **Intermediate measure group** list, select **Internet Sales Reason**.

The following image shows the changes in the **Define Relationship** dialog box.



5. Click **OK**.

   Notice the many-to-many icon that represents the relationship between the Sales Reason dimension and the Internet Sales measure group.

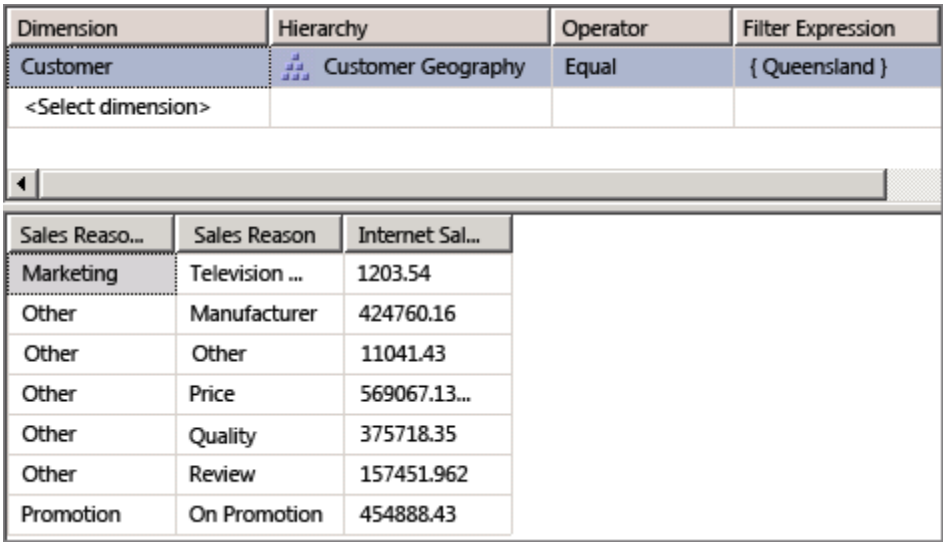
Browsing the Cube and the Many-to-Many Dimension

1. On the **Build** menu, click **Deploy Analysis Services Tutorial**.
2. When deployment has successfully completed, switch to the **Browser** tab in Cube Designer for the Analysis Services Tutorial cube, and then click **Reconnect**.
3. Add the **Internet Sales-Sales Amount** measure to the data area of the data pane.
4. Add the **Sales Reasons** user-defined hierarchy from the **Sales Reason** dimension to the row area of the data pane.
5. In the metadata pane, expand **Customer**, expand **Location**, expand **Customer Geography**, expand **Members**, expand **All Customers**, expand **Australia**, right-click **Queensland**, and then click **Add to Filter**.
6. Expand each member of the **Sales Reason Type** level to review the dollar values that are associated with each reason a customer in Queensland gave for their

purchase of an Adventure Works product over the Internet.

Notice that the totals that are associated with each sales reason add up to more than the total sales. This is because some customers cited multiple reasons for their purchase.

The following image shows the **Filter** pane and **Data** pane of Cube Designer.



Dimension	Hierarchy	Operator	Filter Expression
Customer	Customer Geography	Equal	{ Queensland }
<Select dimension>			

Sales Reaso...	Sales Reason	Internet Sal...
Marketing	Television ...	1203.54
Other	Manufacturer	424760.16
Other	Other	11041.43
Other	Price	569067.13...
Other	Quality	375718.35
Other	Review	157451.962
Promotion	On Promotion	454888.43

Next Task in Lesson

Defining Dimension Granularity within a Measure Group

See Also

Working with Diagrams in a Data Source View (SSAS)

Dimension Relationships

Defining a Many-to-Many Relationship and Many-to-Many Relationship Properties

Defining Dimension Granularity within a Measure Group

Users will want to dimension fact data at different granularity or specificity for different purposes. For example, sales data for reseller or internet sales may be recorded for each day, whereas sales quota information may only exist at the month or quarter level. In these scenarios, users will want a time dimension with a different grain or level of detail for each of these different fact tables. While you could define a new database dimension as a time dimension with this different grain, there is an easier way with Analysis Services.

By default in Analysis Services, when a dimension is used within a measure group, the grain of the data within that dimension is based on the key attribute of the dimension.

For example, when a time dimension is included within a measure group and the default grain of the time dimension is daily, the default grain of that dimension within the measure group is daily. Many times this is appropriate, such as for the **Internet Sales** and **Reseller Sales** measure groups in this tutorial. However, when such a dimension is included in other types of measure groups, such as in a sales quota or budget measure group, a monthly or quarterly grain is generally more appropriate.

To specify a grain for a cube dimension other than the default grain, you modify the granularity attribute for a cube dimension as used within a particular measure group on the **Dimension Usage** tab of Cube Designer. When you change the grain of a dimension within a specific measure group to an attribute other than the key attribute for that dimension, you must guarantee that all other attributes in the measure group are directly or indirectly related to new granularity attribute. You do this by specifying attribute relationships between all other attributes and the attribute that is specified as the granularity attribute in the measure group. In this case, you define additional attribute relationships rather than move attribute relationships. The attribute that is specified as the granularity attribute effectively becomes the key attribute within the measure group for the remaining attributes in the dimension. If you do not specify attribute relationships appropriately, Analysis Services will not be able to aggregate values correctly, as you will see in the tasks in this topic.

For more information, see Dimension Relationships, Defining a Regular Relationship and Regular Relationship Properties.

In the tasks in this topic, you add a Sales Quotas measure group and define the granularity of the Date dimension in this measure group to be monthly. You then define attribute relationships between the month attribute and other dimension attributes to ensure that Analysis Services aggregates values correctly.

Adding Tables and Defining the Sales Quotas Measure Group

1. Switch to the **Adventure Works DW 2012** data source view.
2. Right-click anywhere in the **Diagram Organizer** pane, click **New Diagram**, and then name the diagram **Sales Quotas**.
3. Drag the **Employee**, **Sales Territory**, and **Date** tables from the **Tables** pane to the **Diagram** pane.
4. Add the **FactSalesQuota** table to the **Diagram** pane by right-clicking anywhere in the **Diagram** pane and selecting **Add/Remove Tables**.

   Notice that the **SalesTerritory** table is linked to the **FactSalesQuota** table through the **Employee** table.
5. Review the columns in the **FactSalesQuota** table and then explore the data in this table.

   Notice that the grain of the data within this table is the calendar quarter, which is

the lowest level of detail in the FactSalesQuota table.

6. In Data Source View Designer, change the **FriendlyName** property of the **FactSalesQuota** table to **SalesQuotas**.
7. Switch to the Analysis Services Tutorial cube, and then click the **Cube Structure** tab.
8. Right-click anywhere in the **Measures** pane, click **New Measure Group**, click **SalesQuotas** in the **New Measure Group** dialog box, and then click **OK**.

   The **Sales Quotas** measure group appears in the **Measures** pane. In the **Dimensions** pane, notice that a new **Date** cube dimension is also defined, based on the **Date** database dimension. A new time-related cube dimension is defined because Analysis Services does not know which of the existing time-related cube dimensions to relate to the **DateKey** column in the **FactSalesQuota** fact table that underlies the Sales Quotas measure group. You will change this later in another task in this topic.
9. Expand the **Sales Quotas** measure group.
10. In the **Measures** pane, select **Sales Amount Quota**, and then set the value for the **FormatString** property to **Currency** in the Properties window.
11. Select the **Sales Quotas Count** measure, and then type **#,#** as the value for the **FormatString** property in the Properties window.
12. Delete the **Calendar Quarter** measure from the **Sales Quotas** measure group.

    Analysis Services detected the column that underlies the Calendar Quarter measure as a column that contains measures. However, this column and the CalendarYear column contain the values that you will use to link the Sales Quotas measure group to the Date dimension later in this topic.
13. In the **Measures** pane, right-click the **Sales Quotas** measure group, and then click **New Measure**. For more information, see Defining and Configuring a Measure.

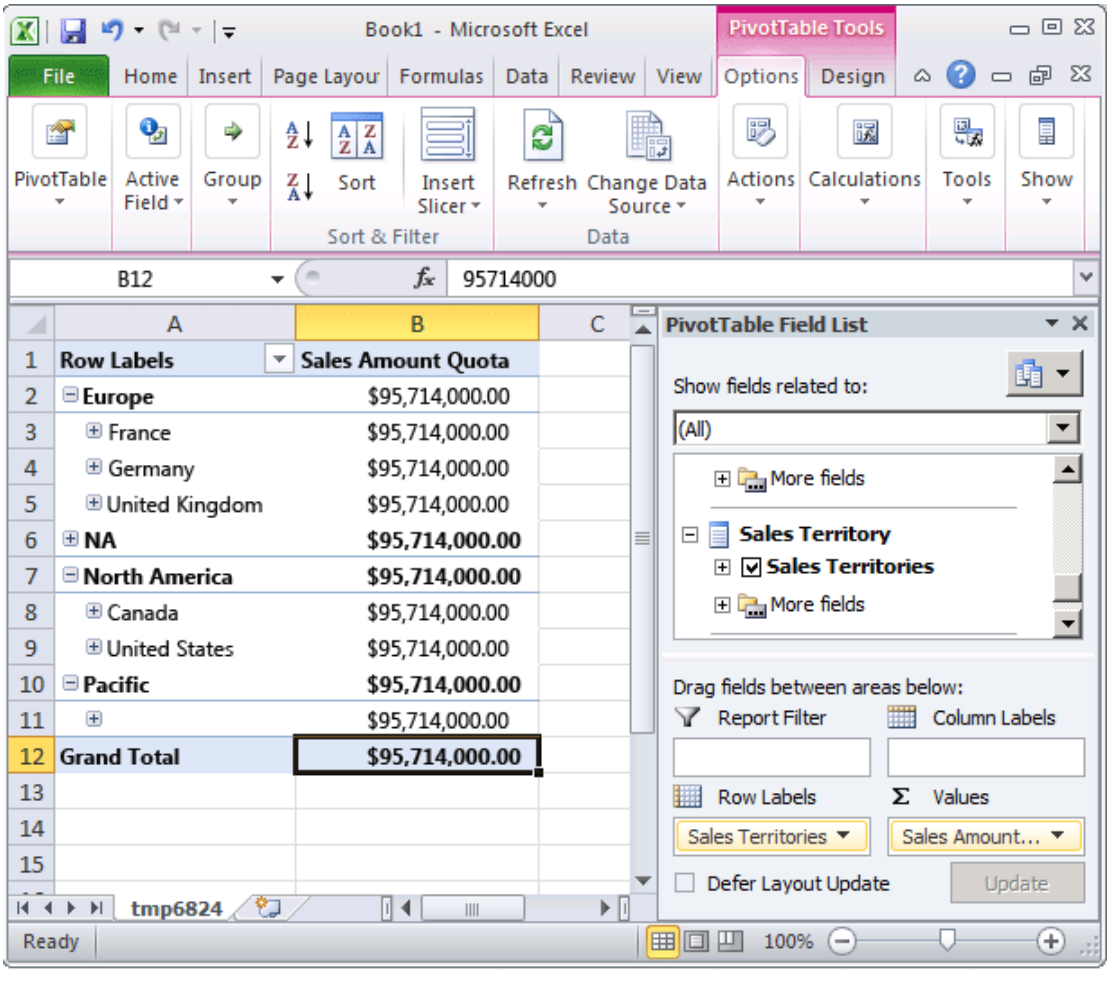
    The **New Measure** dialog box opens, containing the available source columns for a measure with a usage type of **Sum**.
14. In the **New Measure** dialog box, select **Distinct count** in the **Usage** list, verify that **SalesQuotas** is selected in the **Source table** list, select **EmployeeKey** in the **Source column** list, and then click **OK**.

    Notice that the measure is created in a new measure group named **Sales Quotas 1**. Distinct count measures in SQL Server are created in their own measure groups to maximize processing performance.
15. Change the value for the **Name** property for the **Employee Key Distinct Count** measure to **Sales Person Count**, and then type **#,#** as the value for the **FormatString** property.

Browsing the Measures in the Sales Quota Measure Group by Date

1. On the **Build** menu, click **Deploy Analysis Services Tutorial**.
2. When deployment has successfully completed, click the **Browser** tab in Cube Designer for the Analysis Services Tutorial cube, and then click the **Reconnect** button.
3. Click the Excel shortcut, and then click **Enable**.
4. In the PivotTable Field List, expand the **Sales Quotas** measure group, and then drag the **Sales Amount Quota** measure to the Values area.
5. Expand the **Sales Territory** dimension, and then drag the **Sales Territories** user-defined hierarchy to Row Labels.

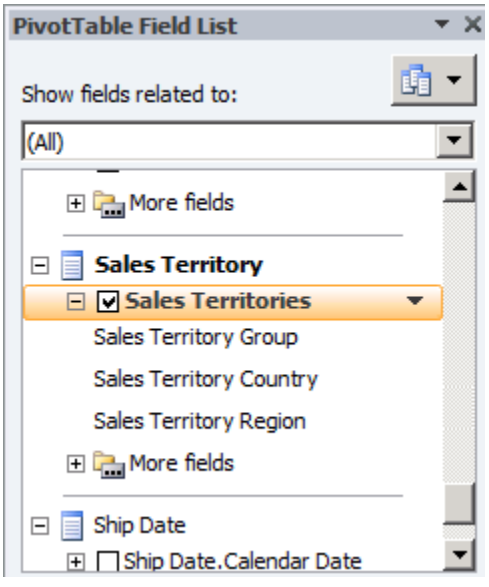
   Notice that the Sales Territory cube dimension is not related, directly or indirectly, to the Fact Sales Quota table, as shown in the following image.



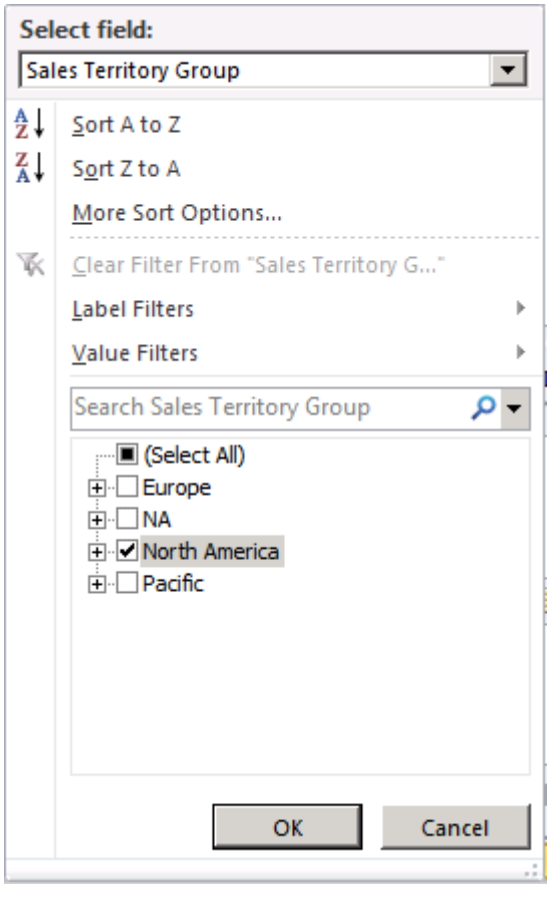
In the next series of steps in this topic you will define a reference dimension

relationship between this dimension and this fact table.

6. Move the **Sales Territories** user hierarchy from the Rows Labels area to the Column Labels area.
7. In the PivotTable Field list, select the **Sales Territories** user-defined hierarchy, and then click the down arrow to the right.



8. In the filter, click the Select All checkbox to clear all the selections, and then choose just **North America**.

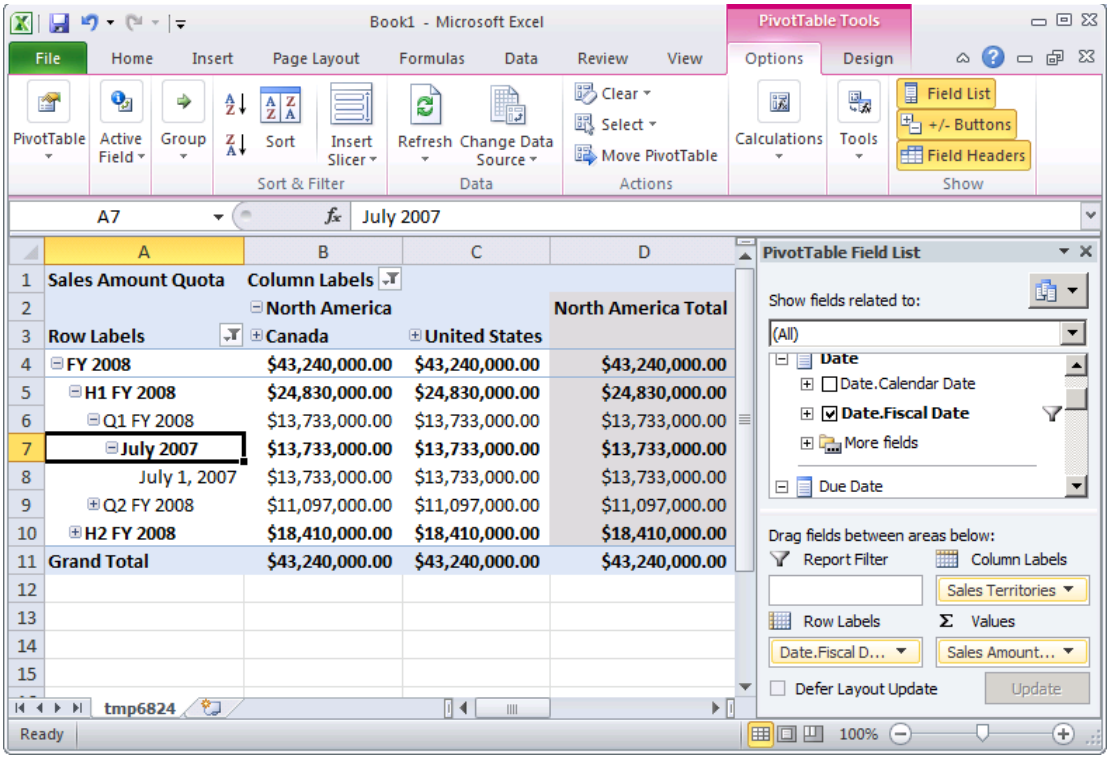


9. In the PivotTable Field List, expand **Date**.
10. Drag the **Date.Fiscal Date** user hierarchy to Row Labels
11. On the PivotTable, click the down arrow next to Row Labels. Clear all of the years except for **FY 2008**.

    Notice that only the **July 2007** member of the **Month** level appears, instead of the **July, 2007**, **August, 2007**, and **September, 2007** members of **Month** level, and that only the **July 1, 2007** member of the **Date** level appears, instead of all 31 days. This behavior occurs because the grain of the data in the fact table is at the quarter level and the grain of the **Date** dimension is the daily level. You will change this behavior in the next task in this topic.

    Notice also that the **Sales Amount Quota** value for the month and day levels is the same value as for the quarter level, $13,733,000.00. This is because the lowest level of data in the Sales Quotas measure group is at the quarter level. You will change this behavior in Lesson 6.

    The following image shows the values for **Sales Amount Quota**.



Defining Dimension Usage Properties for the Sales Quotas Measure Group

1. Open Dimension Designer for the **Employee** dimension, right-click **SalesTerritoryKey** in the **Data Source View** pane, and then click **New Attribute from Column**.
2. In the **Attributes** pane, select **SalesTerritoryKey**, and then set the **AttributeHierarchyVisible** property to **False** in the Properties window, set the **AttributeHierarchyOptimizedState** property to **NotOptimized**, and set the **AttributeHierarchyOrdered** property to **False**.

   This attribute is required to link the **Sales Territory** dimension to the **Sales Quotas** and **Sales Quotas 1** measure groups as a referenced dimension.
3. In Cube Designer for the Analysis Services Tutorial cube, click the **Dimension Usage** tab, and then review the dimension usage within the **Sales Quotas** and **Sales Quotas 1** measure groups.

   Notice that the **Employee** and **Date** cube dimensions are linked to the **Sales Quotas and Sales Quotas 1** measure groups through regular relationships. Notice also that the **Sales Territory** cube dimension is not linked to either of these measure groups.

4. Click the cell at the intersection of the **Sales Territory** dimension and the **Sales Quotas** measure group and then click the browse button (**...**). The **Define Relationship** dialog box opens.
5. In the **Select relationship type** list, select **Referenced**.
6. In the **Intermediate dimension** list, select **Employee**.
7. In the **Reference dimension attribute** list, select **Sales Territory Region.**
8. In the **Intermediate dimension attribute** list, select **Sales Territory Key**. (The key column for the Sales Territory Region attribute is the SalesTerritoryKey column.)
9. Verify that the **Materialize** check box is selected.
10. Click **OK**.
11. Click the cell at the intersection of the **Sales Territory** dimension and the **Sales Quotas 1** measure group and then click the browse button (**...**). The **Define Relationship** dialog box opens.
12. In the **Select relationship type** list, select **Referenced**.
13. In the **Intermediate dimension** list, select **Employee**.
14. In the **Reference dimension attribute** list, select **Sales Territory Region.**
15. In the **Intermediate dimension attribute** list, select **Sales Territory Key**. (The key column for the Sales Territory Region attribute is the SalesTerritoryKey column.)
16. Verify that the **Materialize** check box is selected.
17. Click **OK**.
18. Delete the **Date** cube dimension.

    Instead of having four time-related cube dimensions, you will use the **Order Date** cube dimension in the **Sales Quotas** measure group as the date against which sales quotas will be dimensioned. You will also use this cube dimension as the primary date dimension in the cube.
19. In the **Dimensions** list, rename the **Order Date** cube dimension to **Date**.

    Renaming the **Order Date** cube dimension to **Date** makes it easier for users to understand its role as the primary date dimension in this cube.
20. Click the browse button (**...**) in the cell at the intersection of the **Sales Quotas** measure group and the **Date** dimension.
21. In the **Define Relationship** dialog box, select **Regular** in the **Select relationship type** list.
22. In the **Granularity attribute** list, select **Calendar Quarter**.

    Notice that a warning appears to notify you that because you have selected a non-key attribute as the granularity attribute, you must make sure that all other attributes are directly or indirectly related to the granularity attribute by

specifying them as member properties.

23. In the **Relationship** area of the **Define Relationship** dialog box, link the **CalendarYear** and **CalendarQuarter** dimension columns from the table that underlies the Date cube dimension to the **CalendarYear** and **CalendarQuarter** columns in the table that underlies the Sales Quota measure group, and then click **OK**.

    Note
    The Calendar Quarter is defined as the granularity attribute for the Date cube dimension in the Sales Quotas measure group, but the Date attribute continues to be the granularity attribute for the Internet Sales and Reseller Sales measure groups.

24. Repeat the previous four steps for the **Sales Quotas 1** measure group.

Defining Attribute Relationships Between the Calendar Quarter Attribute and the Other Dimension Attributes in the Date Dimension

1. Switch to **Dimension Designer** for the **Date** dimension, and then click the **Attribute Relationships** tab.

   Notice that although **Calendar Year** is linked to **Calendar Quarter** through the **Calendar Semester** attribute, the fiscal calendar attributes are linked only to one another; they are not linked to the **Calendar Quarter** attribute and therefore will not aggregate correctly in the **Sales Quotas** measure group.
2. In the diagram, right-click the **Calendar Quarter** attribute and then select **New Attribute Relationship**.
3. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Calendar Quarter**. Set the **Related Attribute** to **Fiscal Quarter**.
4. Click **OK**.

   Notice that a warning message appears stating that the **Date** dimension contains one or more redundant attribute relationships that may prevent data from being aggregated when a non-key attribute is used as a granularity attribute.
5. Delete the attribute relationship between the **Month Name** attribute and the **Fiscal Quarter** attribute.
6. On the **File** menu, click **Save All**.

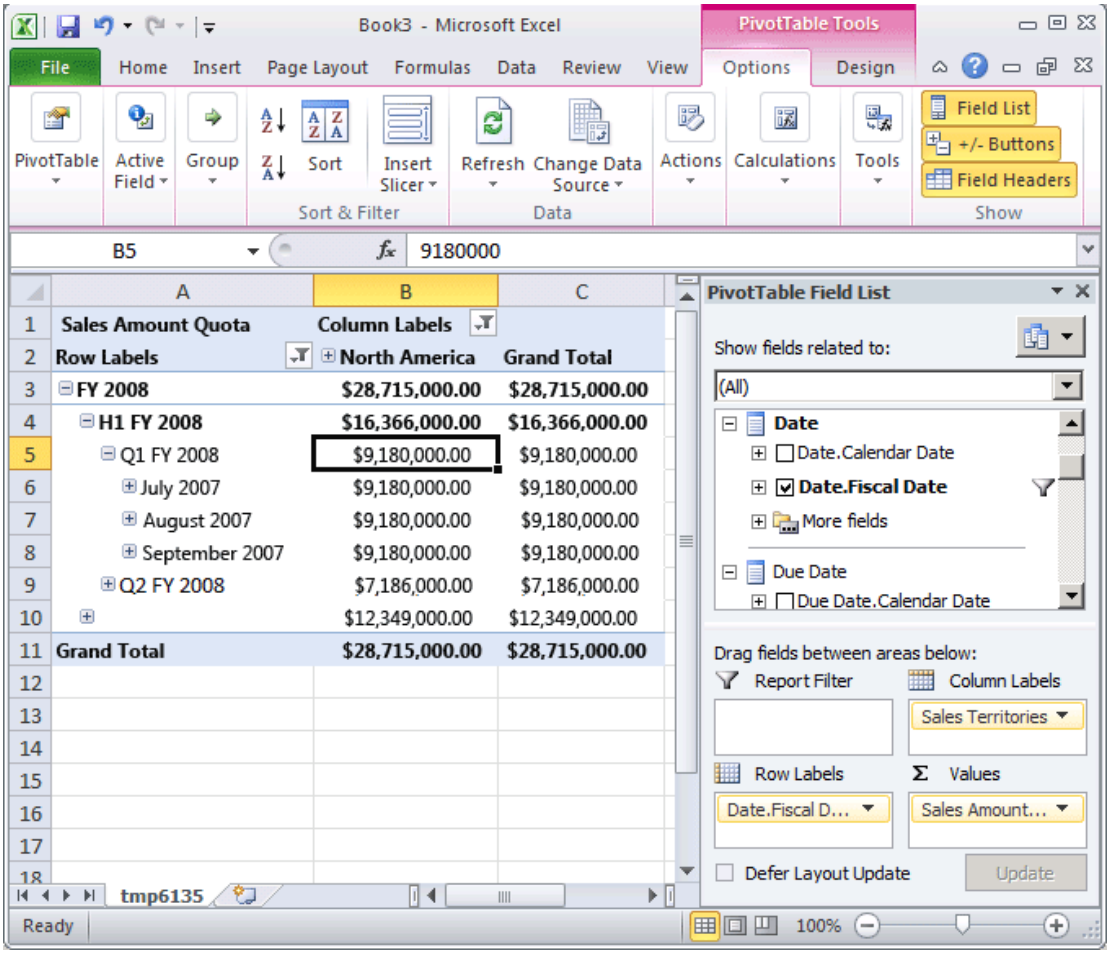
Browsing the Measures in the Sales Quota Measure Group by Date

1. On the **Build** menu, click **Deploy Analysis Services Tutorial**.

2. When deployment has successfully completed, click the **Browser** tab in Cube Designer for the Analysis Services Tutorial cube, and then click **Reconnect**.
3. Click the Excel shortcut, and then click **Enable**.
4. Drag the **Sales Amount Quota** measure to the Values area.
5. Drag the **Sales Territories** user hierarchy to the Column Labels, and then filter on **North America**.
6. Drag the **Date.FiscalDate** user hierarchy to the Row Labels, and then click the down arrow next to **Row Labels** on the PivotTable, and clear all check boxes other than **FY 2008**, to display only fiscal year 2008.
7. Click OK.
8. Expand **FY 2008**, expand **H1 FY 2008**, and then expand **Q1 FY 2008**.

   The following image shows a PivotTable for the Analysis Services Tutorial cube, with the Sales Quota measure group dimensioned correctly.

   Notice that each member of the fiscal quarter level has the same value as the quarter level. Using **Q1 FY 2008** as an example, the quota of $9,180,000.00 for **Q1 FY 2008** is also the value for each of its members. This behavior occurs because the grain of the data in the fact table is at the quarter level and the grain of the Date dimension is also at the quarter level. In Lesson 6, you will learn how to allocate the quarterly amount proportionally to each month.



Next Lesson

Lesson 6: Defining Calculations

See Also

Dimension Relationships

Defining a Regular Relationship and Regular Relationship Properties

Working with Diagrams in a Data Source View (SSAS)

Lesson 6: Defining Calculations

In this lesson, you learn to define calculations, which are Multidimensional Expressions (MDX) expressions or scripts. Calculations enable you to define calculated members, named sets, and execute other script commands to extend the capabilities of an Analysis

Services cube. For example, you can run a script command to define a subcube and then assign a calculation to the cells in the subcube.

When you define a new calculation in Cube Designer, the calculation is added to the **Script Organizer** pane of the **Calculations** tab of Cube Designer, and the fields for the particular calculation type are displayed in a calculations form in the **Calculation Expressions** pane. Calculations are executed in the order in which they are listed in the **Script Organizer** pane. You can reorder the calculations by right-clicking on a particular calculation and then selecting **Move Up** or **Move Down**, or by clicking a particular calculation and then using the **Move Up** or **Move Down** icons on the toolbar of the **Calculations** tab.

On the **Calculations** tab, you can add new calculations and view or edit existing calculations in the following views in the **Calculation Expressions** pane:

- Form view. This view shows the expressions and properties for a single command in a graphical format. When you edit an MDX script, an expression box fills the Form view.
- Script view. This view displays all calculation scripts in a code editor, which lets you easily change the calculation scripts. When the **Calculation Expressions** pane is in Script view, the **Script Organizer** is hidden. The Script view provides color coding, parenthesis matching, auto-complete, and MDX code regions. You can expand or collapse the MDX code regions to make editing easier.

To switch between these views in the **Calculation Expressions** pane, click **Form View** or **Script View** on the toolbar of the **Calculations** tab.

Note

If Analysis Services detects a syntax error in any calculation, the Form view will not display until the error is corrected in the Script view.

You can also use the Business Intelligence Wizard to add certain calculations to a cube. For example, you can use this wizard to add time intelligence to a cube, which means defining calculated members for time-related calculations such as period-to-date, moving averages, or period over period growth. For more information, see Defining Time Intelligence Calculations using the Business Intelligence Wizard.

Important

On the **Calculations** tab, the calculation script starts with the CALCULATE command. The CALCULATE command controls the aggregation of the cells in the cube and you should edit this command only if you intend to manually specify how the cube cells should be aggregated.

For more information, see Calculations (SSAS), and Defining and Configuring a Calculation.

Note

Completed projects for all of the lessons in this tutorial are available online. You can jump ahead to any lesson by using the completed project from the previous

lesson as a starting point. Click here to download the sample projects that go with this tutorial.

This lesson contains the following tasks:

Defining Calculated Members

In this task, you learn to define calculated members.

Defining Named Sets

In this task, you learn to define named sets.

Next Lesson

Lesson 7: Defining Key Performance Indicators (KPIs)

See Also

Analysis Services Tutorial Scenario
SQL Server 2005 Analysis Services Tutorial
Defining a Named Set
Defining a Calculated Member

Defining Calculated Members

Calculated members are members of a dimension or a measure group that are defined based on a combination of cube data, arithmetic operators, numbers, and functions. For example, you can create a calculated member that calculates the sum of two physical measures in the cube. Calculated member definitions are stored in cubes, but their values are calculated at query time.

To create a calculated member, use the **New Calculated Member** command on the **Calculations** tab of Cube Designer. You can create a calculated member within any dimension, including the measures dimension. You can also place a calculated member within a display folder in the **Calculation Properties** dialog box. For more information, see Calculations (SSAS), Defining and Configuring a Calculation, and Defining a Calculated Member.

In the tasks in this topic, you define calculated measures to let users view the gross profit margin percentage and sales ratios for Internet sales, reseller sales, and for all sales.

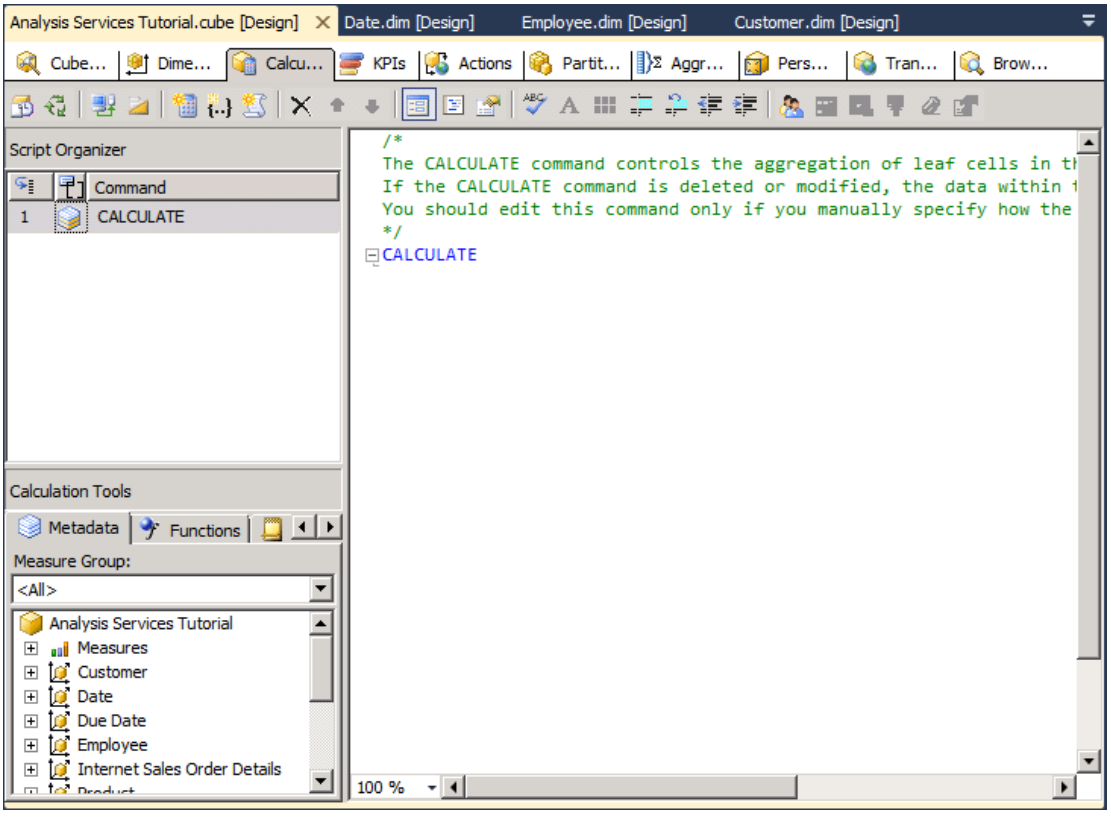
Defining Calculations to Aggregate Physical Measures

1. Open Cube Designer for the Analysis Services Tutorial cube, and then click the **Calculations** tab.

   Notice the default CALCULATE command in the **Calculation Expressions** pane

and in the **Script Organizer** pane. This command specifies that the measures in the cube should be aggregated according to the value that is specified by their AggregateFunction properties. Measure values are generally summed, but may also be counted or aggregated in some other manner.

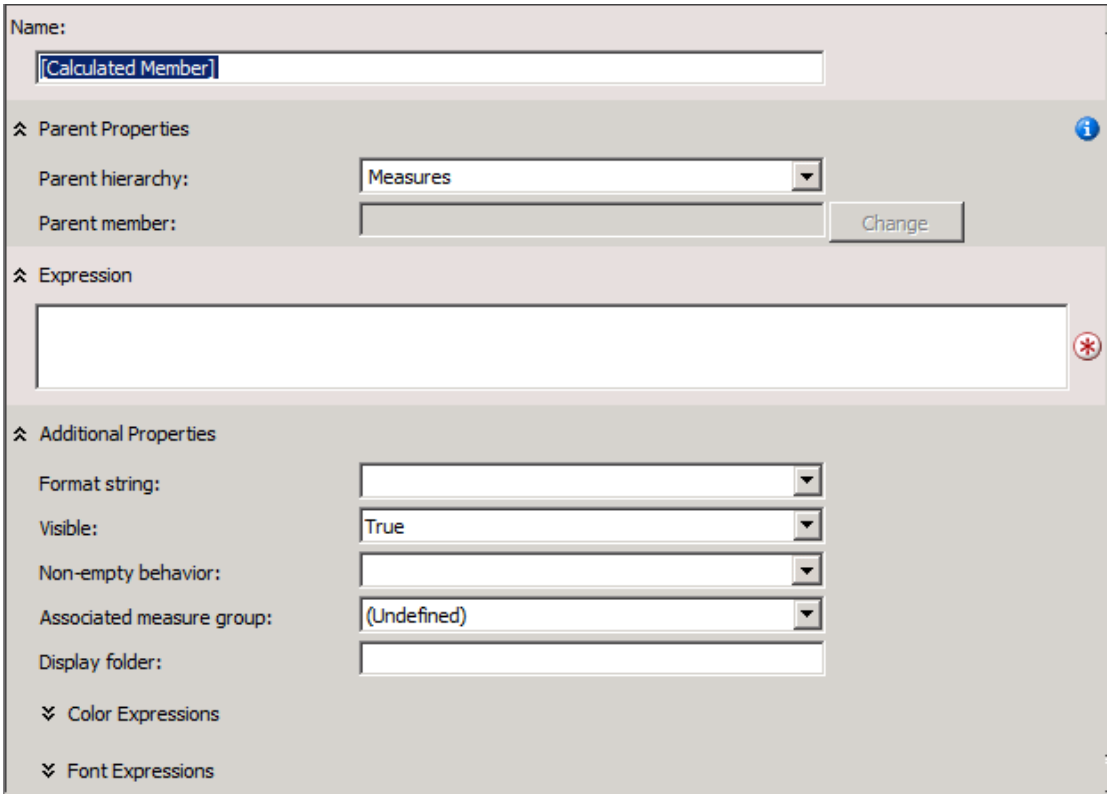
The following image shows the **Calculations** tab of Cube Designer.



2. On the toolbar of the **Calculations** tab, click **New Calculated Member**.

   A new form appears in the **Calculation Expressions** pane within which you define the properties of this new calculated member. The new member also appears in the **Script Organizer** pane.

   The following image shows the form that appears in the **Calculation Expressions** pane when you click **New Calculated Member**.



3. In the **Name** box, change the name of the calculated measure to **[Total Sales Amount]**.

   If the name of a calculated member contains a space, the calculated member name must be enclosed in square brackets.

   Notice in the **Parent hierarchy** list that, by default, a new calculated member is created in the **Measures** dimension. A calculated member in the Measures dimension is also frequently called a calculated measure.

4. On the **Metadata** tab in the **Calculation Tools** pane of the **Calculations** tab, expand **Measures** and then expand **Internet Sales** to view the metadata for the **Internet Sales** measure group.

   You can drag metadata elements from the **Calculation Tools** pane into the **Expression** box and then add operators and other elements to create Multidimensional Expressions (MDX) expressions. Alternatively, you can type the MDX expression directly into the **Expression** box.

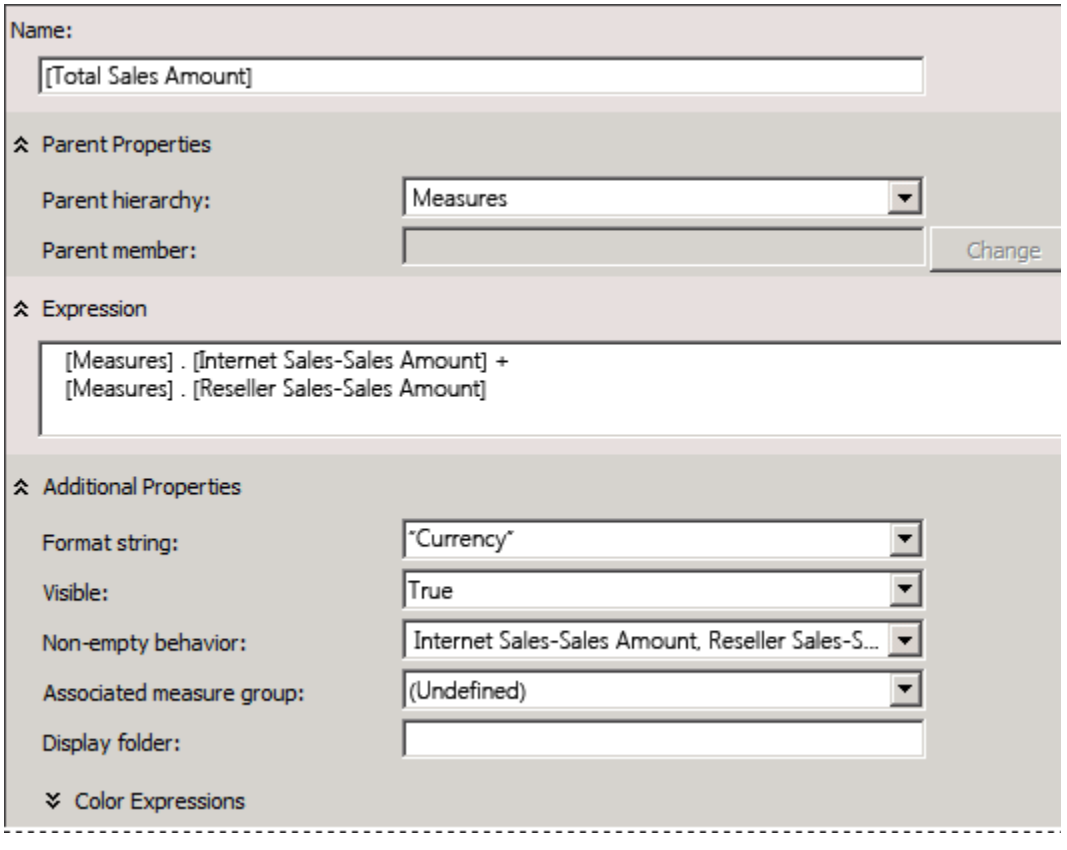
   Note

   If you cannot view any metadata in the **Calculation Tools** pane, click **Reconnect** on the toolbar. If this does not work, you may have to process the cube or start the instance of Analysis Services.

5. Drag **Internet Sales-Sales Amount** from the **Metadata** tab in the **Calculation Tools** pane into the **Expression** box in the **Calculation Expressions** pane.
6. In the **Expression** box, type a plus sign (**+**) after **[Measures].[Internet Sales-Sales Amount]**.
7. On the **Metadata** tab in the **Calculation Tools** pane, expand **Reseller Sales**, and then drag **Reseller Sales-Sales Amount** into the **Expression** box in the **Calculation Expressions** pane after the plus sign (+).
8. In the **Format string** list, select **"Currency".**
9. In the **Non-empty behavior** list, select the check boxes for **Internet Sales-Sales Amount** and **Reseller Sales-Sales Amount**, and then click **OK**.

   The measures you specify in the **Non-empty behavior** list are used to resolve NON EMPTY queries in MDX. When you specify one or more measures in the **Non-empty behavior** list, Analysis Services treats the calculated member as empty if all the specified measures are empty. If the **Non-empty behavior** property is blank, Analysis Services must evaluate the calculated member itself to determine whether the member is empty.

   The following image shows the **Calculation Expressions** pane populated with the settings that you specified in the previous steps.



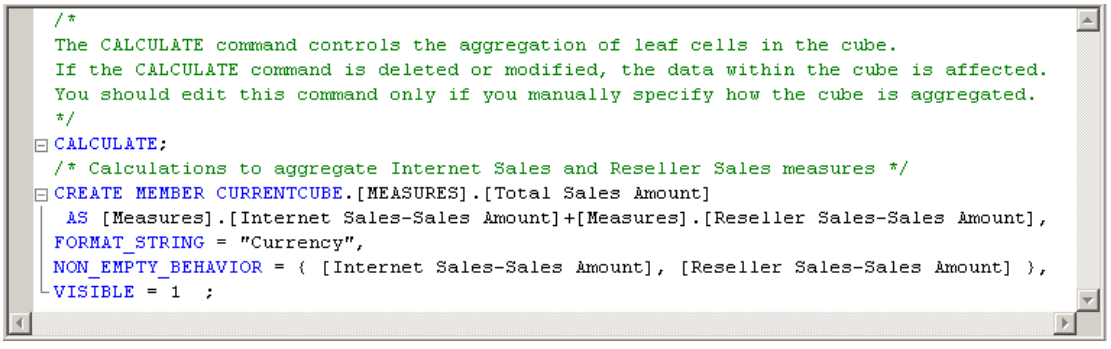
10. On the toolbar of the **Calculations** tab, click **Script View**, and then review the calculation script in the **Calculation Expressions** pane.

    Notice that the new calculation is added to the initial CALCULATE expression; each individual calculation is separated by a semicolon. Notice also that a comment appears at the beginning of the calculation script. Adding comments within the calculation script for groups of calculations is a good practice, to help you and other developers understand complex calculation scripts.

11. Add a new line in the calculation script after the **Calculate;** command and before the newly added calculation script, and then add the following text to the script on its own line:

    ```
    /* Calculations to aggregate Internet Sales and Reseller
    Sales measures */
    ```

    The following image shows the calculation scripts as they should appear in the **Calculation Expressions** pane at this point in the tutorial.



```
/*
The CALCULATE command controls the aggregation of leaf cells in the cube.
If the CALCULATE command is deleted or modified, the data within the cube is affected.
You should edit this command only if you manually specify how the cube is aggregated.
*/
CALCULATE;
/* Calculations to aggregate Internet Sales and Reseller Sales measures */
CREATE MEMBER CURRENTCUBE.[MEASURES].[Total Sales Amount]
 AS [Measures].[Internet Sales-Sales Amount]+[Measures].[Reseller Sales-Sales Amount],
FORMAT_STRING = "Currency",
NON_EMPTY_BEHAVIOR = { [Internet Sales-Sales Amount], [Reseller Sales-Sales Amount] },
VISIBLE = 1  ;
```

12. On the toolbar of the **Calculations** tab, click **Form View**, verify that **[Total Sales Amount]** is selected in the **Script Organizer** pane, and then click **New Calculated Member**.
13. Change the name of this new calculated member to **[Total Product Cost]**, and then create the following expression in the **Expression** box:

    ```
    [Measures].[Internet Sales-Total Product Cost] +
    [Measures].[Reseller Sales-Total Product Cost]
    ```

14. In the **Format string** list, select **"Currency"**.
15. In the **Non-empty behavior** list, select the check boxes for **Internet Sales-Total Product Cost** and **Reseller Sales-Total Product Cost**, and then click **OK**.

    You have now defined two calculated members, both of which are visible in the **Script Organizer** pane. These calculated members can be used by other calculations that you define later in the calculation script. You can view the definition of any calculated member by selecting the calculated member in the **Script Organizer** pane; the definition of the calculated member will appear in the **Calculation Expressions** pane in the Form view. Newly defined calculated

members will not appear in the **Calculation Tools** pane until these objects have been deployed. Calculations do not require processing.

Defining Gross Profit Margin Calculations

1. Verify that **[Total Product Cost]** is selected in the **Script Organizer** pane, and then click **New Calculated Member** on the toolbar of the **Calculations** tab.
2. In the **Name** box, change the name of this new calculated measure to **[Internet GPM]**.
3. In the **Expression** box, create the following MDX expression:

```
([Measures].[Internet Sales-Sales Amount] -
[Measures].[Internet Sales-Total Product Cost]) /
[Measures].[Internet Sales-Sales Amount]
```

4. In the **Format string** list, select **"Percent"**.
5. In the **Non-empty behavior** list, select the check box for **Internet Sales-Sales Amount**, and then click **OK**.
6. On the toolbar of the **Calculations** tab, click **New Calculated Member**.
7. In the **Name** box, change the name of this new calculated measure to **[Reseller GPM]**.
8. In the **Expression** box, create the following MDX expression:

```
([Measures].[Reseller Sales-Sales Amount] -
[Measures].[Reseller Sales-Total Product Cost]) /
[Measures].[Reseller Sales-Sales Amount]
```

9. In the **Format string** list, select **"Percent"**.
10. In the **Non-empty behavior** list, select the check box for **Reseller Sales-Sales Amount**, and then click **OK**.
11. On the toolbar of the **Calculations** tab, click **New Calculated Member**.
12. In the **Name** box, change the name of this calculated measure to **[Total GPM]**.
13. In the **Expression** box, create the following MDX expression:

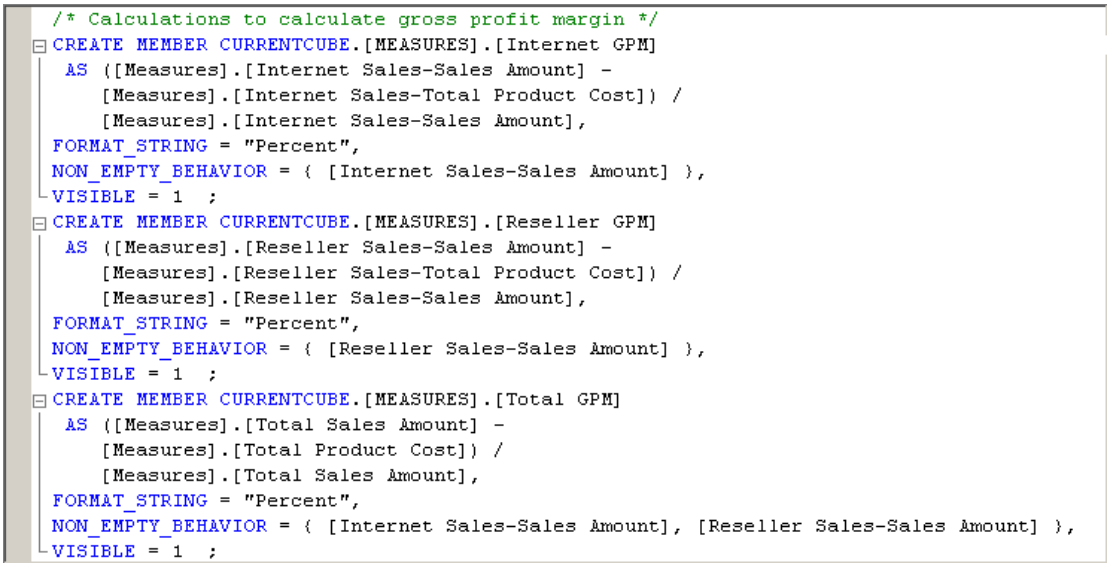
```
([Measures].[Total Sales Amount] -
[Measures].[Total Product Cost]) /
[Measures].[Total Sales Amount]
```

    Notice that this calculated member is referencing other calculated members. Because this calculated member will be calculated after the calculated members that it references, this is a valid calculated member.
14. In the **Format string** list, select **"Percent"**.

15. In the **Non-empty behavior** list, select the check boxes for **Internet Sales-Sales Amount** and **Reseller Sales-Sales Amount**, and then click **OK**.
16. On the toolbar of the **Calculations** tab, click **Script View** and review the three calculations you just added to the calculation script.
17. Add a new line in the calculation script immediately before the **[Internet GPM]** calculation, and then add the following text to the script on its own line:

```
/* Calculations to calculate gross profit margin */
```

The following image shows the **Expressions** pane with the three new calculations.



```
/* Calculations to calculate gross profit margin */
CREATE MEMBER CURRENTCUBE.[MEASURES].[Internet GPM]
 AS ([Measures].[Internet Sales-Sales Amount] -
    [Measures].[Internet Sales-Total Product Cost]) /
    [Measures].[Internet Sales-Sales Amount],
FORMAT_STRING = "Percent",
NON_EMPTY_BEHAVIOR = { [Internet Sales-Sales Amount] },
VISIBLE = 1  ;
CREATE MEMBER CURRENTCUBE.[MEASURES].[Reseller GPM]
 AS ([Measures].[Reseller Sales-Sales Amount] -
    [Measures].[Reseller Sales-Total Product Cost]) /
    [Measures].[Reseller Sales-Sales Amount],
FORMAT_STRING = "Percent",
NON_EMPTY_BEHAVIOR = { [Reseller Sales-Sales Amount] },
VISIBLE = 1  ;
CREATE MEMBER CURRENTCUBE.[MEASURES].[Total GPM]
 AS ([Measures].[Total Sales Amount] -
    [Measures].[Total Product Cost]) /
    [Measures].[Total Sales Amount],
FORMAT_STRING = "Percent",
NON_EMPTY_BEHAVIOR = { [Internet Sales-Sales Amount], [Reseller Sales-Sales Amount] },
VISIBLE = 1  ;
```

Defining the Percent of Total Calculations

1. On the toolbar of the **Calculations** tab, click **Form View**.
2. In the **Script Organizer** pane, select **[Total GPM]**, and then click **New Calculated Member** on the toolbar of the **Calculations** tab.

   Clicking the final calculated member in the **Script Organizer** pane before you click **New Calculated Member** guarantees that the new calculated member will be entered at the end of the script. Scripts execute in the order that they appear in the **Script Organizer** pane.
3. Change the name of this new calculated member to **[Internet Sales Ratio to All Products]**.
4. Type the following expression in the **Expression** box:

```
Case
```

```
When IsEmpty( [Measures].[Internet Sales-Sales Amount] )
Then 0
Else ( [Product].[Product Categories].CurrentMember,
       [Measures].[Internet Sales-Sales Amount]) /
     ( [Product].[Product Categories].[(All)].[All],
       [Measures].[Internet Sales-Sales Amount] )
End
```

This MDX expression calculates the contribution to total Internet sales of each product. The Case statement together with the IS EMPTY function ensures that a divide by zero error does not occur when a product has no sales.

5. In the **Format string** list, select **"Percent"**.
6. In the **Non-empty behavior** list, select the check box for **Internet Sales-Sales Amount**, and then click **OK**.
7. On the toolbar of the **Calculations** tab, click **New Calculated Member**.
8. Change the name of this calculated member to **[Reseller Sales Ratio to All Products]**.
9. Type the following expression in the **Expression** box:

```
Case
    When IsEmpty( [Measures].[Reseller Sales-Sales Amount] )
    Then 0
    Else ( [Product].[Product Categories].CurrentMember,
           [Measures].[Reseller Sales-Sales Amount]) /
         ( [Product].[Product Categories].[(All)].[All],
           [Measures].[Reseller Sales-Sales Amount] )
    End
```

10. In the **Format string** list, select **"Percent"**.
11. In the **Non-empty behavior** list, select the check box for **Reseller Sales-Sales Amount**, and then click **OK**.
12. On the toolbar of the **Calculations** tab, click **New Calculated Member**.
13. Change the name of this calculated member to **[Total Sales Ratio to All Products]**.
14. Type the following expression in the **Expression** box:

```
Case
    When IsEmpty( [Measures].[Total Sales Amount] )
    Then 0
```

```
Else ( [Product].[Product Categories].CurrentMember,
       [Measures].[Total Sales Amount]) /
     ( [Product].[Product Categories].[(All)].[All],
       [Measures].[Total Sales Amount] )
End
```

15. In the **Format string** list, select **"Percent"**.
16. In the **Non-empty behavior** list, select the check boxes for **Internet Sales-Sales Amount** and **Reseller Sales-Sales Amount**, and then click **OK**.
17. On the toolbar of the **Calculations** tab, click **Script View**, and then review the three calculations that you just added to the calculation script.
18. Add a new line in the calculation script immediately before the **[Internet Sales Ratio to All Products]** calculation, and then add the following text to the script on its own line:

    ```
    /* Calculations to calculate percentage of product to total
    product sales */
    ```

    You have now defined a total of eight calculated members, which are visible in the **Script Organizer** pane when you are in Form view.

Browsing the New Calculated Members

1. On the **Build** menu of SQL Server Data Tools (SSDT), click **Deploy Analysis Services Tutorial**.
2. When deployment has successfully completed, switch to the **Browser** tab, click **Reconnect**.
3. Click the Excel icon, and then click **Enable**.
4. In the **PivotTable Field List** pane, expand **Values** folder to view the new calculated members in the Measures dimension.
5. Drag the **Total Sales Amount** to the Values area, and then review the results.

   Drag **Internet Sales-Sales Amount** and **Reseller Sales-Sales Amount** measures from the **Internet Sales** and **Reseller Sales** measure groups to the Values area.

   Notice that the **Total Sales Amount** measure is the sum of the **Internet Sales-Sales Amount** measure and the **Reseller Sales-Sales Amount** measure.
6. Add the **Product Categories** user-defined hierarchy to the filter area of the **Report Filter** area, and then filter the data by **Mountain Bikes**.

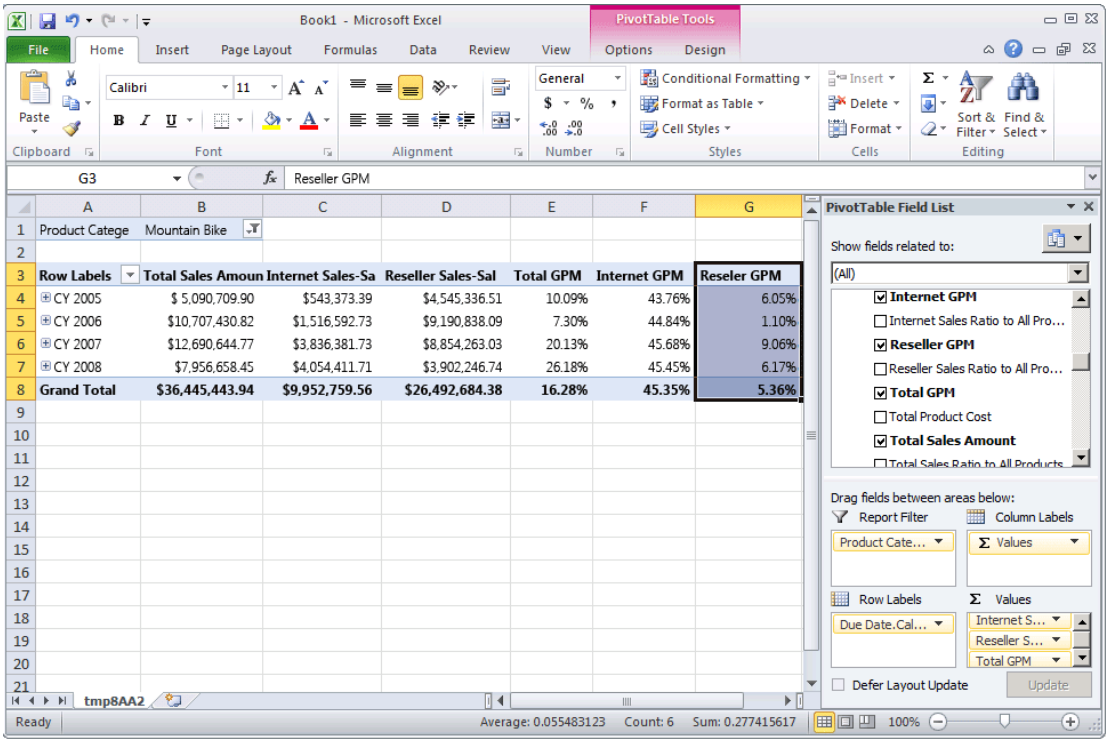
   Notice that the **Total Sales Amount** measure is calculated for the **Mountain Bikes** category of product sales based on the **Internet Sales-Sales Amount** and the **Reseller Sales-Sales Amount** measures for **Mountain Bikes**.

7. Add the **Date.Calendar Date** user-defined hierarchy to the Row labels area, and then review the results.

   Notice that the **Total Sales Amount** measure for each calendar year is calculated for the **Mountain Bikes** category of product sales based on the **Internet Sales-Sales Amount** and the **Reseller Sales-Sales Amount** measures for **Mountain Bikes**.

8. Add the **Total GPM**, **Internet GPM**, and **Reseller GPM** measures to the Values area, and then review the results.

   Notice that the gross profit margin for reseller sales is significantly lower than for sales over the Internet, as shown in the following image.



9. Add the **Total Sales Ratio to All Products**, **Internet Sales Ratio to All Products**, and **Reseller Sales Ratio to All Products** measures to the Values area.

   Notice that the ratio of the sales of mountain bikes to all products has increased over time for Internet sales, but is decreasing over time for reseller sales. Notice also that the ratio of the sale of mountain bikes to all products is lower from sales through resellers than it is for sales over the Internet.

10. Change the filter from **Mountain Bikes** to **Bikes**, and review the results.

    Notice that the gross profit margin for all bikes sold through resellers is negative, because touring bikes and road bikes are being sold at a loss.

11. Change the filter to **Accessories**, and then review the results.

Notice that the sale of accessories is increasing over time, but that these sales make up only a small fraction of total sales. Notice also that the gross profit margin for sales of accessories is higher than for bikes.

Next Task in Lesson

Defining Named Sets

See Also

Calculations (SSAS)

Defining and Configuring a Calculation

Defining a Calculated Member

Defining Named Sets

A named set is a Multidimensional Expressions (MDX) expression that returns a set of dimension members. You can define named sets and save them as part of the cube definition; you can also create named sets in client applications. You create named sets by combining cube data, arithmetic operators, numbers, and functions. Named sets can be used by users in MDX queries in client applications and can also be used to define sets in subcubes. A subcube is a collection of crossjoined sets that restricts the cube space to the defined subspace for subsequent statements. Defining a restricted cube space is a fundamental concept to MDX scripting.

Named sets simplify MDX queries and provide useful aliases for complex, typically used, set expressions. For example, you can define a named set called Large Resellers that contains the set of members in the Reseller dimension that have the most employees. End users could then use the Large Resellers named set in queries, or you could use the named set to define a set in a subcube. Named set definitions are stored in cubes, but their values exist only in memory. To create a named set, use the **New Named Set** command on the **Calculations** tab of Cube Designer. For more information, see Calculations, Defining a Named Set.

In the tasks in this topic, you will define two named sets: a Core Products named set and a Large Resellers named set.

Defining a Core Products Named Set

1. Switch to the **Calculations** tab of Cube Designer for the Analysis Services Tutorial cube, and then click **Form View** on the toolbar.
2. Click **[Total Sales Ratio to All Products]** in the **Script Organizer** pane, and then click **New Named Set** on the toolbar of the **Calculations** tab.

   When you define a new calculation on the **Calculations** tab, remember that

calculations are resolved in the order in which they appear in the **Script Organizer** pane. Your focus within that pane when you create a new calculation determines the order of the execution of the calculation; a new calculation is defined immediately after the calculation on which you are focused.

3. In the **Name** box, change the name of the new named set to **[Core Products]**.

   In the **Script Organizer** pane, notice the unique icon that differentiates a named set from a script command or a calculated member.

4. On the **Metadata** tab in the **Calculation Tools** pane, expand **Product**, expand **Category**, expand **Members**, and then expand **All Products**.

   Note
   If you cannot view any metadata in the **Calculation Tools** pane, click **Reconnect** on the toolbar. If this does not work, you may have to process the cube or start the instance of Analysis Services.

5. Drag **Bikes** into the **Expression** box.

   You now have created a set expression that will return the set of members that are in the Bike category in the Product dimension.

Defining a Large Resellers Named Set

1. Right-click **[Core Products]** in the **Script Organizer** pane, and then click **New Named Set**.
2. In the **Name** box, change the name of this named set to **[Large Resellers]**.
3. In the **Expression** box, type **Exists()**.

   You will use the **Exists** function to return the set of members from the Reseller Name attribute hierarchy that intersects with the set of members in the Number of Employees attribute hierarchy that has the largest number of employees.

4. On the **Metadata** tab in the **Calculation Tools** pane, expand the **Reseller** dimension, and then expand the **Reseller Name** attribute hierarchy.
5. Drag the **Reseller Name** level into the parenthesis for the **Exists** set expression.

   You will use the **Members** function to return all members of this set. For more information, see Members (Set) (MDX).

6. After the partial set expression, type a period, and then add the **Members** function. Your expression should look like the following:

```
Exists([Reseller].[Reseller Name].[Reseller Name].Members)
```

   Now that you have defined the first set for the **Exists** set expression, you are ready to add the second set—the set of members of the Reseller dimension that contains the largest number of employees.

7. On the **Metadata** tab in the **Calculation Tools** pane, expand **Number of**

Employees in the Reseller dimension, expand **Members**, and then expand **All Resellers**.

Notice that the members of this attribute hierarchy are not grouped.

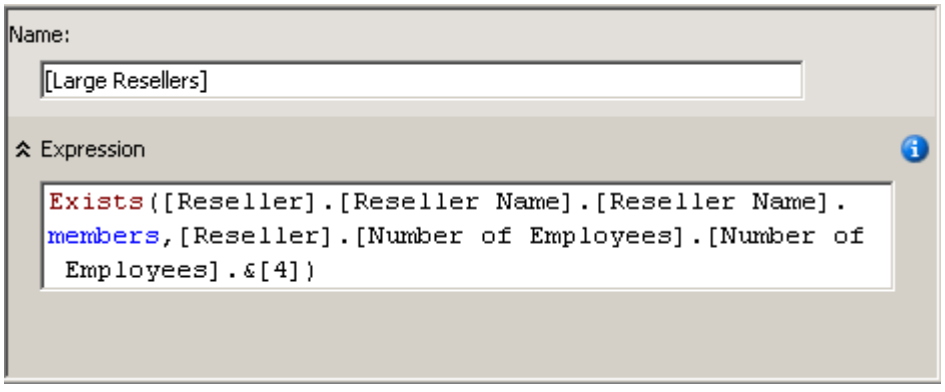
8. Open Dimension Designer for the **Reseller** dimension, and then click **Number of Employees** in the **Attributes** pane.
9. In the Properties window, change the **DiscretizationMethod** property to **Automatic**, and then change the **DiscretizationBucketCount** property to **5**. For more information, see Grouping Attribute Members (Discretization).
10. On the **Build** menu of SQL Server Data Tools (SSDT), click **Deploy Analysis Services Tutorial**.
11. When deployment has successfully completed, switch to Cube Designer for the Analysis Services Tutorial cube, and then click **Reconnect** on the toolbar of the **Calculations** tab.
12. On the **Metadata** tab in the **Calculation Tools** pane, expand **Number of Employees** in the **Reseller** dimension, expand **Members**, and then expand **All Resellers**.

    Notice that the members of this attribute hierarchy are now contained in five groups, numbered 0 through 4. To view the number of a group, pause the pointer over that group to view an InfoTip. For the range `2 -17`, the InfoTip should contain `[Reseller].[Number of Employees].&[0]`.

    The members of this attribute hierarchy are grouped because the **DiscretizationBucketCount** property is set to **5** and the **DiscretizationMethod** property is set to **Automatic**.
13. In the **Expression** box, add a comma in the **Exists** set expression after the **Members** function and before the closing parenthesis, and then drag **83 - 100** from the **Metadata** pane and position it after the comma.

    You have now completed the **Exists** set expression that will return the set of members that intersects with these two specified sets, the set of all resellers and the set of resellers who have 83 to 100 employees, when the Large Resellers named set is put on an axis.

    The following image shows the **Calculation Expressions** pane for the **[Large Resellers]** named set.



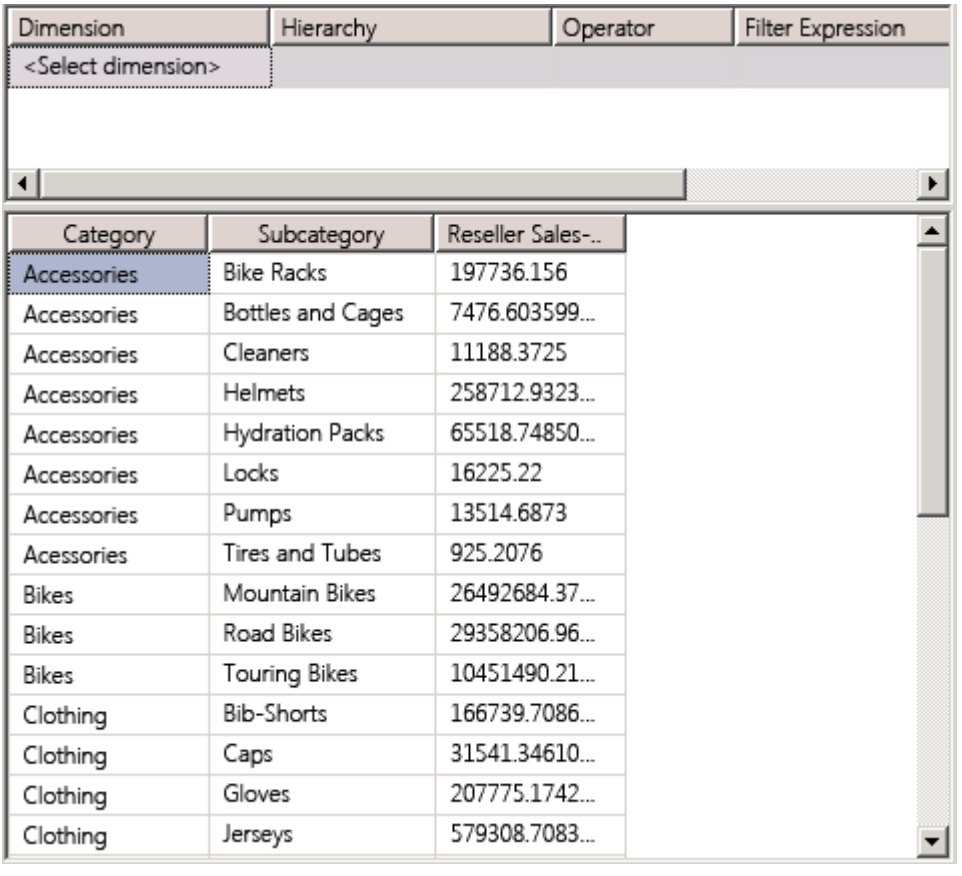
14. On the toolbar of the **Calculations** tab, click **Script View**, and then review the two named sets that you have just added to the calculation script.
15. Add a new line in the calculation script immediately before the first CREATE SET command, and then add the following text to the script on its own line:

    ```
    /* named sets */
    ```

    You have now defined two named sets, which are visible in the **Script Organizer** pane. You are now ready to deploy these named sets, and then to browse these measures in the Analysis Services Tutorial cube.

Browsing the Cube by Using the New Named Sets

1. On the **Build** menu of SQL Server Data Tools, click **Deploy Analysis Services Tutorial**.
2. When deployment has successfully completed, click the **Browser** tab, and then click **Reconnect**.
3. Clear the grid in the data pane.
4. Add the **Reseller Sales-Sales Amount** measure to the data area.
5. Expand the Product dimension, and then add Category and Subcategory to the row area, as shown in the following image.

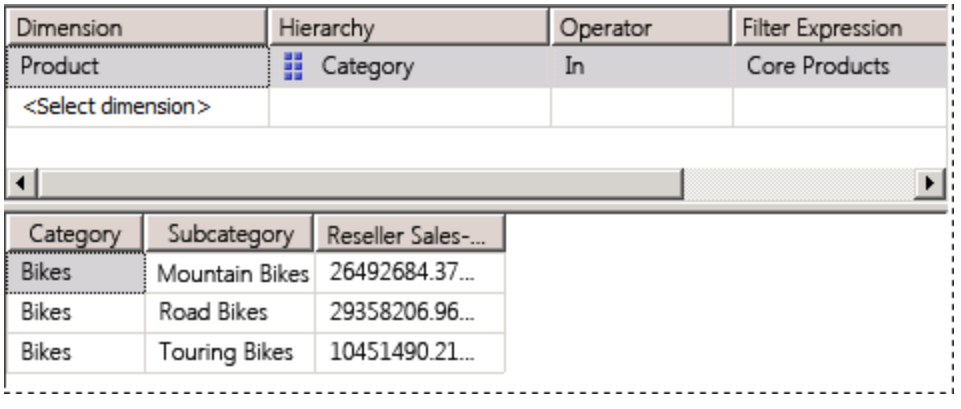


Dimension	Hierarchy	Operator	Filter Expression
<Select dimension>			

Category	Subcategory	Reseller Sales-..
Accessories	Bike Racks	197736.156
Accessories	Bottles and Cages	7476.603599...
Accessories	Cleaners	11188.3725
Accessories	Helmets	258712.9323...
Accessories	Hydration Packs	65518.74850...
Accessories	Locks	16225.22
Accessories	Pumps	13514.6873
Acessories	Tires and Tubes	925.2076
Bikes	Mountain Bikes	26492684.37...
Bikes	Road Bikes	29358206.96...
Bikes	Touring Bikes	10451490.21...
Clothing	Bib-Shorts	166739.7086...
Clothing	Caps	31541.34610...
Clothing	Gloves	207775.1742...
Clothing	Jerseys	579308.7083...

6. In the **Metadata** pane, in the **Product** dimension, drag **Core Products** to the filter area.

   Notice that only the **Bike** member of the **Category** attribute and members of the **Bike** subcategories remain in the cube. This is because the **Core Products** named set is used to define a subcube. This subcube limits the members of the **Category** attribute in the **Product** dimension within the subcube to those members of the **Core Product** named set, as shown in the following image.

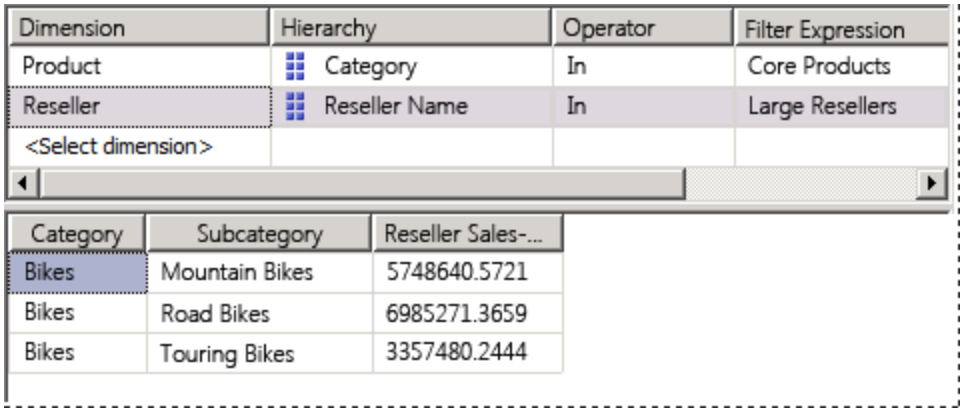


Dimension	Hierarchy	Operator	Filter Expression
Product	Category	In	Core Products
<Select dimension>			

Category	Subcategory	Reseller Sales-...
Bikes	Mountain Bikes	26492684.37...
Bikes	Road Bikes	29358206.96...
Bikes	Touring Bikes	10451490.21...

7. In the **Metadata** pane, expand **Reseller**, add **Large Resellers** to the filter area.

   Notice that the Reseller Sales Amount measure in the Data pane only displays sales amounts for large resellers of bikes. Notice also that the Filter pane now displays the two named sets that are used to define this particular subcube, as shown in the following image.



Dimension	Hierarchy	Operator	Filter Expression
Product	Category	In	Core Products
Reseller	Reseller Name	In	Large Resellers
<Select dimension>			

Category	Subcategory	Reseller Sales-...
Bikes	Mountain Bikes	5748640.5721
Bikes	Road Bikes	6985271.3659
Bikes	Touring Bikes	3357480.2444

Next Task in Lesson

Defining Subcubes and Assigning Calculations Using Script Commands

See Also

Calculations

Defining a Named Set

Lesson 7: Defining Key Performance Indicators (KPIs)

In this lesson, you learn to define Key Performance Indicators (KPIs) in your Analysis Services project. KPIs provide a framework for defining server-side calculations that measure your business, and they standardize how the resulting information is displayed. KPIs can be displayed in reports, portals, and dashboards, through data access APIs, and through Microsoft tools and third-party tools. KPIs are metadata wrappers around regular measures and other Multidimensional Expressions (MDX) expressions. For more information, see Key Performance Indicators (SSAS).

Note

Completed projects for all of the lessons in this tutorial are available online. You can jump ahead to any lesson by using the completed project from the previous

lesson as a starting point. Click here to download the sample projects that go with this tutorial.

This lesson contains the following task:

Defining and Browsing KPIs

In this task, you define KPIs in the Form view and then switch to the Browser view to browse the cube data by using the KPIs.

Next Lesson

Lesson 8: Defining Actions

See Also

Analysis Services Tutorial Scenario

SQL Server 2005 Analysis Services Tutorial

Key Performance Indicators (SSAS)

Defining and Browsing KPIs

To define key performance indicators (KPIs), you first define a KPI name and the measure group to which the KPI is associated. A KPI can be associated with all measure groups or with a single measure group. You then define the following elements of the KPI:

- The value expression

  A value expression is a physical measure such as Sales, a calculated measure such as Profit, or a calculation that is defined within the KPI by using a Multidimensional Expressions (MDX) expression.

- The goal expression

  A goal expression is a value, or an MDX expression that resolves to a value, that defines the target for the measure that the value expression defines. For example, a goal expression could be the amount by which the business managers of a company want to increase sales or profit.

- The status expression

  A status expression is an MDX expression that Analysis Services uses to evaluate the current status of the value expression compared to the goal expression. A goal expression is a normalized value in the range of -1 to +1, where -1 is very bad, and +1 is very good. The status expression displays a graphic to help you easily determine the status of the value expression compared to the goal expression.

- The trend expression

  A trend expression is an MDX expression that Analysis Services uses to evaluate the current trend of the value expression compared to the goal expression. The trend expression helps the business user to quickly determine whether the value expression

is becoming better or worse relative to the goal expression. You can associate one of several graphics with the trend expression to help business users be able to quickly understand the trend.

In addition to these elements that you define for a KPI, you also define several properties of a KPI. These properties include a display folder, a parent KPI if the KPI is computed from other KPIs, the current time member if there is one, the weight of the KPI if it has one, and a description of the KPI.

Note

For more examples of KPIs, see the KPI examples on the Templates tab in the Calculation Tools pane or in the examples in the **Adventure Works DW 2012** sample data warehouse. For more information about how to install this database, see Install Sample Data and Projects for the Analysis Services Multidimensional Modeling Tutorial.

In the task in this lesson, you define KPIs in the Analysis Services Tutorial project, and you then browse the Analysis Services Tutorial cube by using these KPIs. You will define the following KPIs:

- Reseller Revenue

  This KPI is used to measure how actual reseller sales compare to sales quotas for reseller sales, how close the sales are to the goal, and what the trend is toward reaching the goal.
- Product Gross Profit Margin

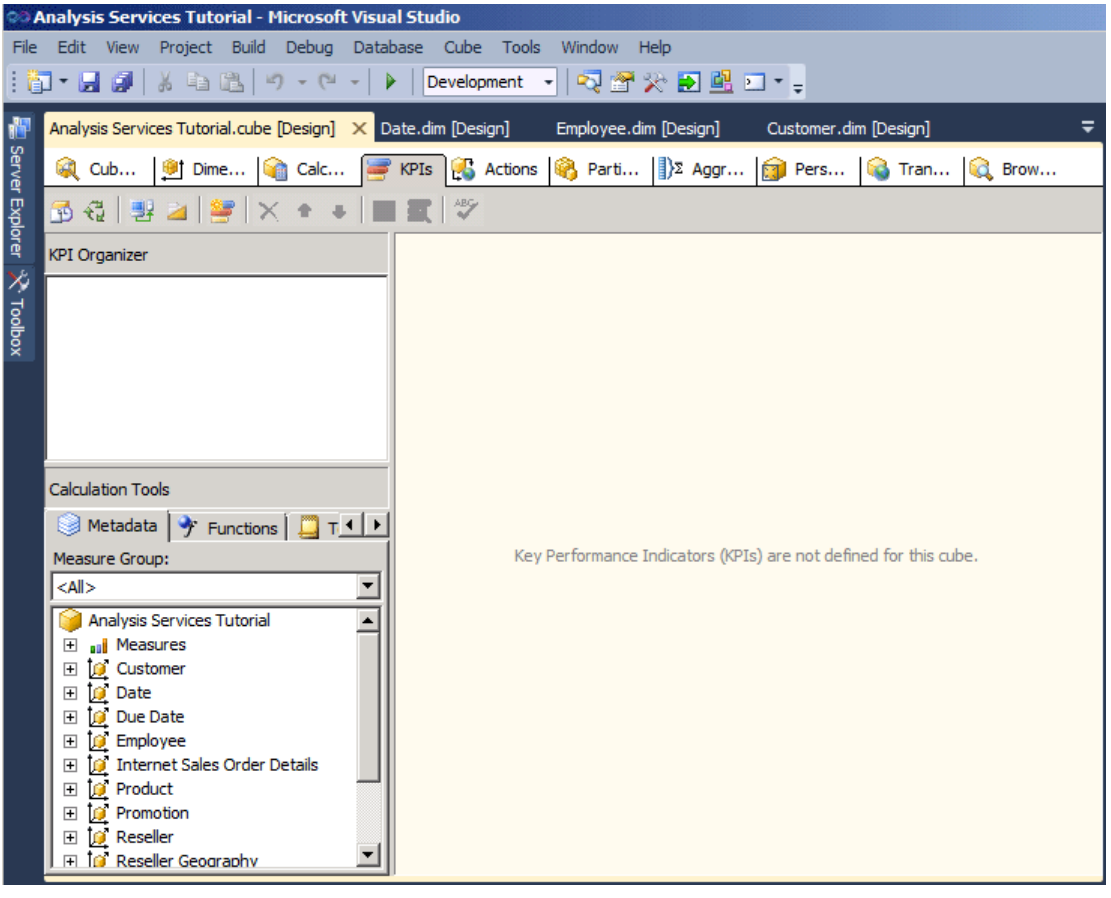
  This KPI is used to determine how close the gross profit margin is for each product category to a specified goal for each product category, and also to determine the trend toward reaching this goal.

Defining the Reseller Revenue KPI

1. Open Cube Designer for the Analysis Services Tutorial cube, and then click the **KPIs** tab.

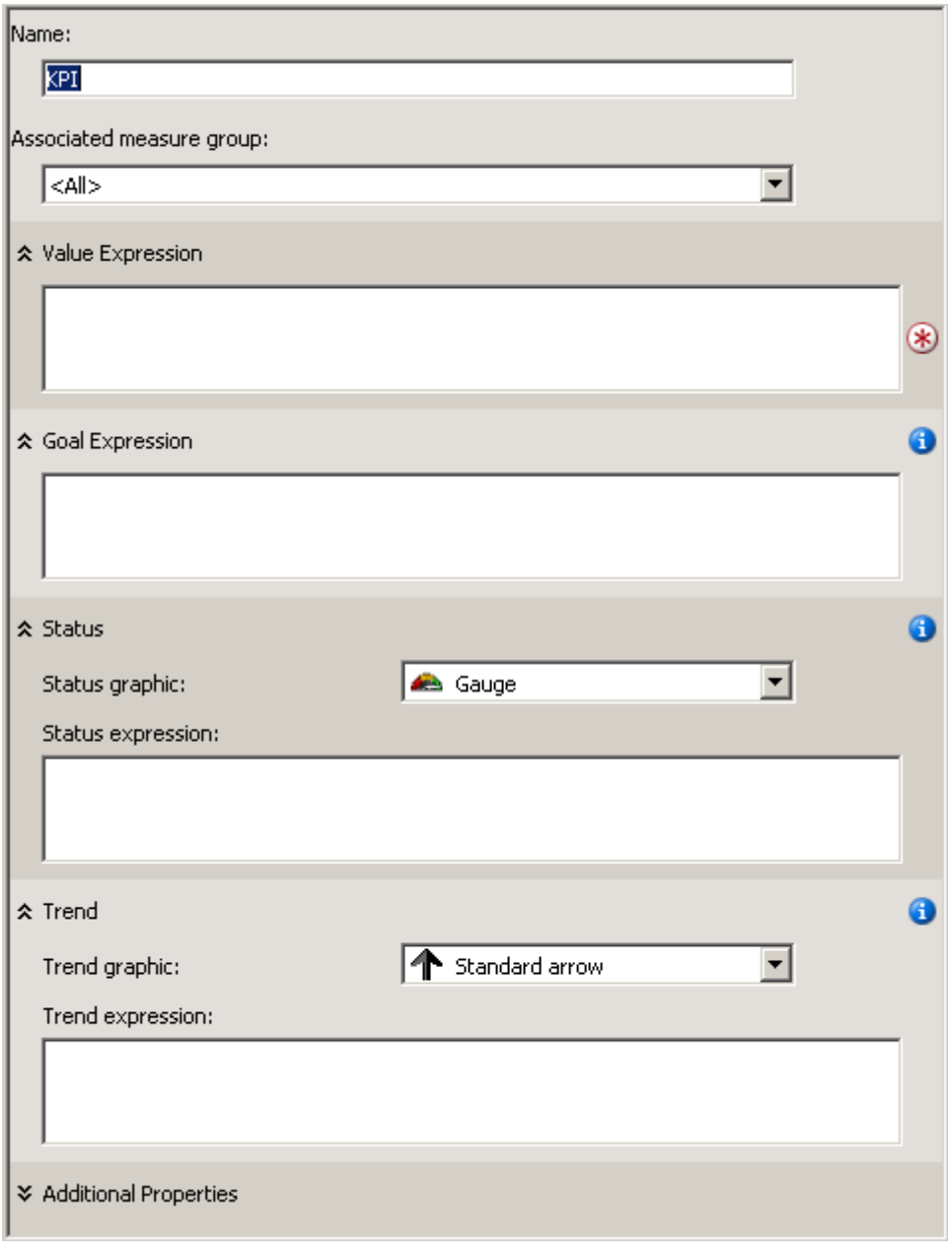
   The **KPIs** tab includes several panes. On the left side of the tab are the **KPI Organizer** pane and the **Calculation Tools** pane. The display pane in the middle of the tab contains the details of the KPI that is selected in the **KPI Organizer** pane.

   The following image shows the **KPIs** tab of Cube Designer.



2. On the toolbar of the **KPIs** tab, click the **New KPI** button.

   A blank KPI template appears in the display pane, as shown in the following image.



3. In the **Name** box, type **Reseller Revenue**, and then select **Reseller Sales** in the **Associated measure group** list.
4. On the **Metadata** tab in the **Calculation Tools** pane, expand **Measures**, expand **Reseller Sales**, and then drag the **Reseller Sales-Sales Amount** measure to the **Value Expression** box.
5. On the **Metadata** tab in the **Calculation Tools** pane, expand **Measures**, expand **Sales Quotas**, and then drag the **Sales Amount Quota** measure to the **Goal**

Expression box.

6. Verify that **Gauge** is selected in the **Status indicator** list, and then type the following MDX expression in the **Status expression** box:

```
Case
 When
  KpiValue("Reseller Revenue")/KpiGoal("Reseller
Revenue")>=.95
   Then 1
 When
  KpiValue("Reseller Revenue")/KpiGoal("Reseller
Revenue")<.95
   And
  KpiValue("Reseller Revenue")/KpiGoal("Reseller
Revenue")>=.85
   Then 0
  Else-1
End
```

This MDX expression provides the basis for evaluating the progress toward the goal. In this MDX expression, if actual reseller sales are more than 85 percent of the goal, a value of 0 is used to populate the chosen graphic. Because a gauge is the chosen graphic, the pointer in the gauge will be half-way between empty and full. If actual reseller sales are more the 90 percent, the pointer on the gauge will be three-fourths of the way between empty and full.

7. Verify that **Standard arrow** is selected in the **Trend indicator** list, and then type the following expression in the **Trend expression** box:

```
Case
 When IsEmpty
  (ParallelPeriod
   ([Date].[Calendar Date].[Calendar Year],1,
       [Date].[Calendar Date].CurrentMember))
  Then 0
 When  (
  KpiValue("Reseller Revenue") -
   (KpiValue("Reseller Revenue"),
    ParallelPeriod
```

```
        ([Date].[Calendar Date].[Calendar Year],1,
          [Date].[Calendar Date].CurrentMember))
         /
         (KpiValue ("Reseller Revenue"),
          ParallelPeriod
           ([Date].[Calendar Date].[Calendar Year],1,
            [Date].[Calendar Date].CurrentMember)))
          >=.02
    Then 1
     When(
      KpiValue("Reseller Revenue") -
       (KpiValue ( "Reseller Revenue" ),
        ParallelPeriod
         ([Date].[Calendar Date].[Calendar Year],1,
          [Date].[Calendar Date].CurrentMember))
         /
          (KpiValue("Reseller Revenue"),
           ParallelPeriod
            ([Date].[Calendar Date].[Calendar Year],1,
              [Date].[Calendar Date].CurrentMember)))
          <=.02
    Then -1
     Else 0
  End
```

This MDX expression provides the basis for evaluating the trend toward achieving the defined goal.

Browsing the Cube by Using the Reseller Revenue KPI

1. On the **Build** menu of SQL Server Data Tools (SSDT), click **Deploy Analysis Service Tutorial**.
2. When deployment has successfully completed, on the toolbar of the **KPIs** tab, click the **Browser View** button, and then click **Reconnect**.

   The status and trend gauges are displayed in the **KPI Browser** pane for reseller

sales based on the values for the default member of each dimension, together with the value for the value and the goal. The default member of each dimension is the All member of the All level, because you have not defined any other member of any dimension as the default member.

3. In the filter pane, select **Sales Territory** in the **Dimension** list, select **Sales Territories** in the **Hierarchy** list, select **Equal** in the **Operator** list, select the **North America** check box in the **Filter Expression** list, and then click **OK**.
4. In the next row in the **Filter** pane, select **Date** in the **Dimension** list, select **Calendar Date** in the **Hierarchy** list, select **Equal** in the **Operator** list, select the **Q3 CY 2007** check box in the **Filter Expression** list, and then click **OK**.
5. Click anywhere in the **KPI Browser** pane to update the values for the **Reseller Revenue KPI**.

   Notice that the **Value**, **Goal**, and **Status** sections of the KPI reflect the values for the new time period

Defining the Product Gross Profit Margin KPI

1. Click the **Form View** button on the toolbar of the **KPIs** tab, and then click the **New KPI** button.
2. In the **Name** box, type **Product Gross Profit Margin**, and then verify that **<All>** appears in the **Associated measure group** list.
3. In the **Metadata** tab in the **Calculation Tools** pane, drag the **Total GPM** measure to the **Value Expression** box.
4. In the **Goal Expression** box, type the following expression:

```
Case
    When [Product].[Category].CurrentMember Is
      [Product].[Category].[Accessories]
    Then .40
    When [Product].[Category].CurrentMember
      Is [Product].[Category].[Bikes]
    Then .12
    When [Product].[Category].CurrentMember Is
      [Product].[Category].[Clothing]
    Then .20
    When [Product].[Category].CurrentMember Is
      [Product].[Category].[Components]
```

```
    Then .10
    Else .12
End
```

5. In the **Status indicator** list, select **Cylinder**.
6. Type the following MDX expression in the **Status expression** box:

```
Case
    When KpiValue( "Product Gross Profit Margin" ) /
         KpiGoal ( "Product Gross Profit Margin" ) >= .90
    Then 1
    When KpiValue( "Product Gross Profit Margin" ) /
         KpiGoal ( "Product Gross Profit Margin" ) <  .90
         And
         KpiValue( "Product Gross Profit Margin" ) /
         KpiGoal ( "Product Gross Profit Margin" ) >= .80
    Then 0
    Else -1
End
```

This MDX expression provides the basis for evaluating the progress toward the goal.

7. Verify that **Standard arrow** is selected in the **Trend indicator** list, and then type the following MDX expression in the **Trend expression** box:

```
Case
When IsEmpty
  (ParallelPeriod
   ([Date].[Calendar Date].[Calendar Year],1,
       [Date].[Calendar Date].CurrentMember))
  Then 0
   When VBA!Abs
    (
      KpiValue( "Product Gross Profit Margin" ) -
       (
         KpiValue ( "Product Gross Profit Margin" ),
          ParallelPeriod
          (
```

```
            [Date].[ Calendar Date].[ Calendar Year],
            1,
            [Date].[ Calendar Date].CurrentMember
          )
        ) /
        (
          KpiValue ( "Product Gross Profit Margin" ),
          ParallelPeriod
          (
            [Date].[ Calendar Date].[ Calendar Year],
            1,
            [Date].[ Calendar Date].CurrentMember
          )
        )
      ) <=.02
Then 0
When KpiValue( "Product Gross Profit Margin" ) -
      (
        KpiValue ( "Product Gross Profit Margin" ),
        ParallelPeriod
        (
          [Date].[ Calendar Date].[ Calendar Year],
          1,
          [Date].[ Calendar Date].CurrentMember
        )
      ) /
      (
        KpiValue ( "Product Gross Profit Margin" ),
        ParallelPeriod
        (
          [Date].[Calendar Date].[Calendar Year],
          1,
          [Date].[Calendar Date].CurrentMember
```

```
            )
          )  >.02
    Then 1
    Else -1
  End
```

This MDX expression provides the basis for evaluating the trend toward achieving the defined goal.

Browsing the Cube by Using the Total Gross Profit Margin KPI

1. On the **Build** menu, click **Deploy Analysis Service Tutorial**.
2. When deployment has successfully completed, click **Reconnect** on the toolbar of the **KPIs** tab, and then click **Browser View**.

   The **Product Gross Profit Margin** KPI appears and displays the KPI value for **Q3 CY 2007** and the **North America** sales territory.
3. In the **Filter** pane, select **Product** in the **Dimension** list, select **Category** in the **Hierarchy** list, select **Equal** in the **Operator** list, and then select **Bikes** in the **Filter Expression** list, and then click **OK**.

   The gross profit margin for the sale of Bikes by resellers in North America in Q3 CY 2007 appears.

Next Lesson

Lesson 8: Defining Actions

Lesson 8: Defining Actions

In this lesson, you will learn to define actions in your Analysis Services project. An action is just a Multidimensional Expressions (MDX) statement that is stored in Analysis Services and which can be incorporated into client applications and started by a user.

Note

Completed projects for all of the lessons in this tutorial are available online. You can jump ahead to any lesson by using the completed project from the previous lesson as a starting point. Click here to download the sample projects that go with this tutorial.

Analysis Services supports the types of actions that are described in the following table.

CommandLine	Executes a command at the command prompt
Dataset	Returns a dataset to a client application.
Drillthrough	Returns a drillthrough statement as an expression, which the client executes to return a rowset
Html	Executes an HTML script in an Internet browser
Proprietary	Performs an operation by using an interface other than those listed in this table.
Report	Submits a parameterized URL-based request to a report server and returns a report to a client application.
Rowset	Returns a rowset to a client application.
Statement	Runs an OLE DB command.
URL	Displays a dynamic Web page in an Internet browser.

Actions let users start an application or perform other steps within the context of a selected item. For more information, see Actions (SSAS), Defining and Configuring an Action

Note
For examples of actions, see the action examples on the Templates tab in the Calculation Tools pane or in the examples in the Adventure Works DW sample data warehouse. For more information about installing this database, see Install Sample Data and Projects for the Analysis Services Multidimensional Modeling Tutorial.

This lesson includes the following task:

Defining and Using a Drillthrough Action

In this task, you define, use, and then modify a drillthrough action through the fact dimension relationship that you defined earlier in this tutorial.

Next Lesson

Lesson 9: Defining Perspectives, Translations, and Currency Conversions

See Also

Analysis Services Tutorial Scenario

SQL Server 2005 Analysis Services Tutorial

Actions (SSAS)

Defining and Configuring an Action

Defining and Using a Drillthrough Action

Dimensioning fact data by a fact dimension without correctly filtering the data that the query returns can cause slow query performance. To avoid this, you can define a drillthrough action that restricts the total number of rows that are returned. This will significantly improve query performance.

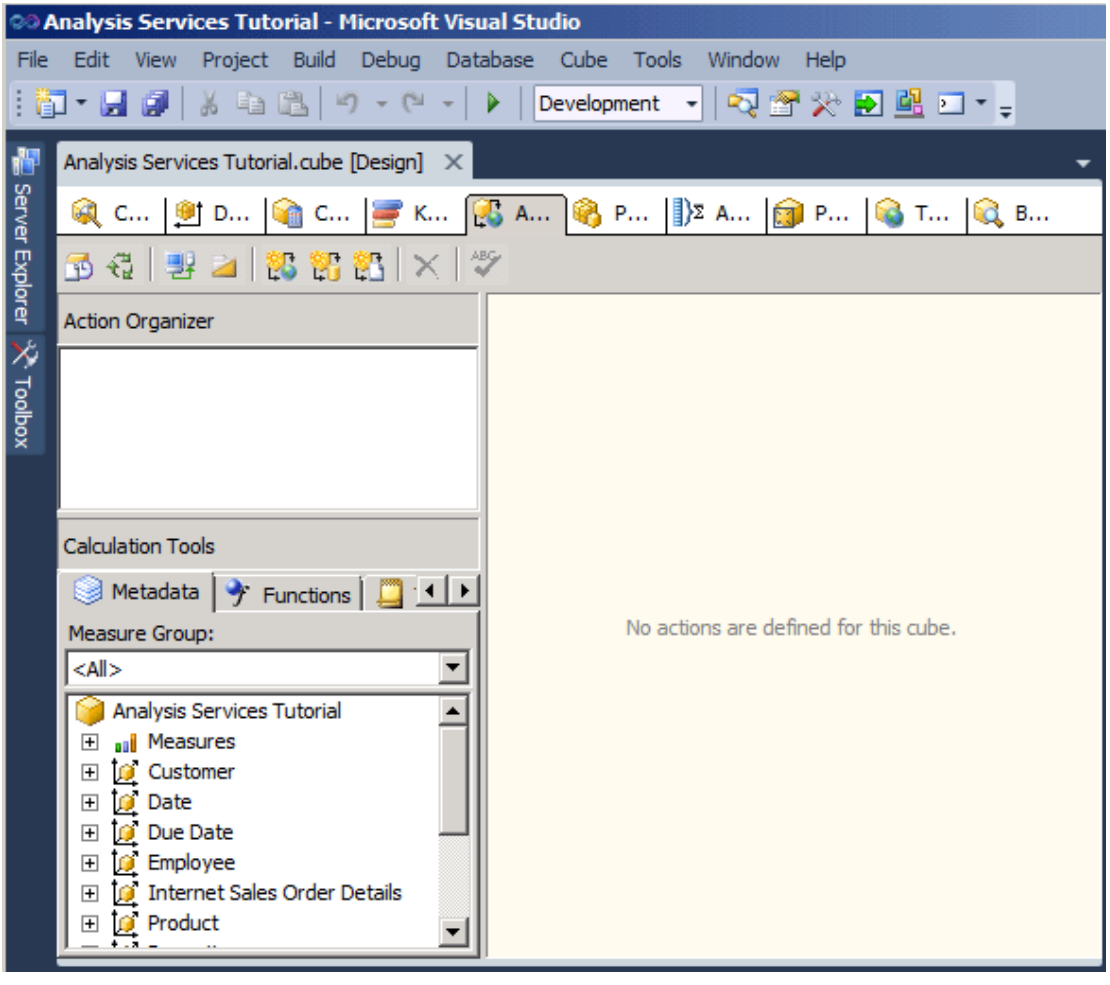
In the tasks in this topic, you define a drillthrough action to return order detail information for sales to customers over the Internet.

Defining the Drillthrough Action Properties

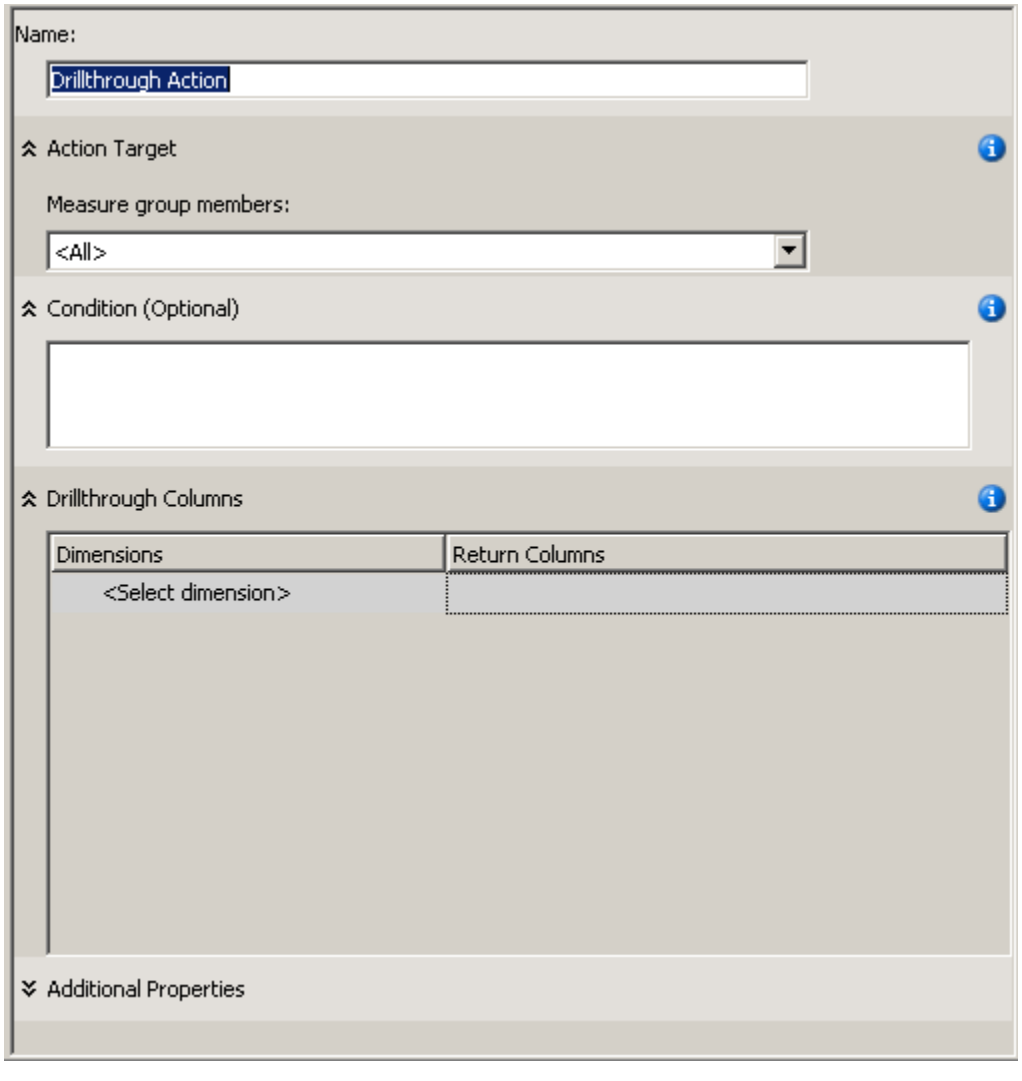
1. In Cube Designer for the Analysis Services Tutorial cube, click the **Actions** tab.

   The **Actions** tab includes several panes. On the left side of the tab are the **Action Organizer** pane and the **Calculation Tools** pane. The pane to the right of these two panes is the **Display** pane, which contains the details of the action that is selected in the **Action Organizer** pane.

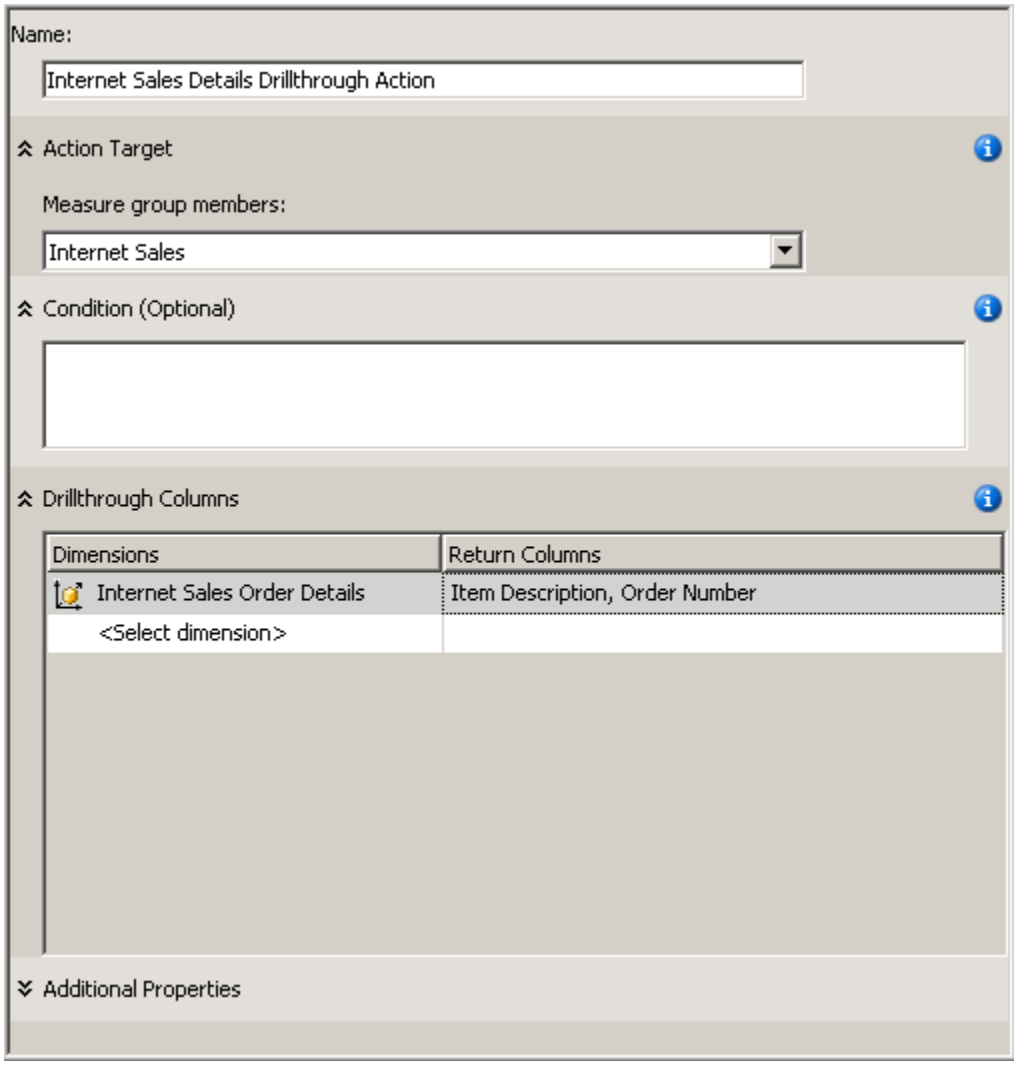
   The following image shows the **Actions** tab of Cube Designer.



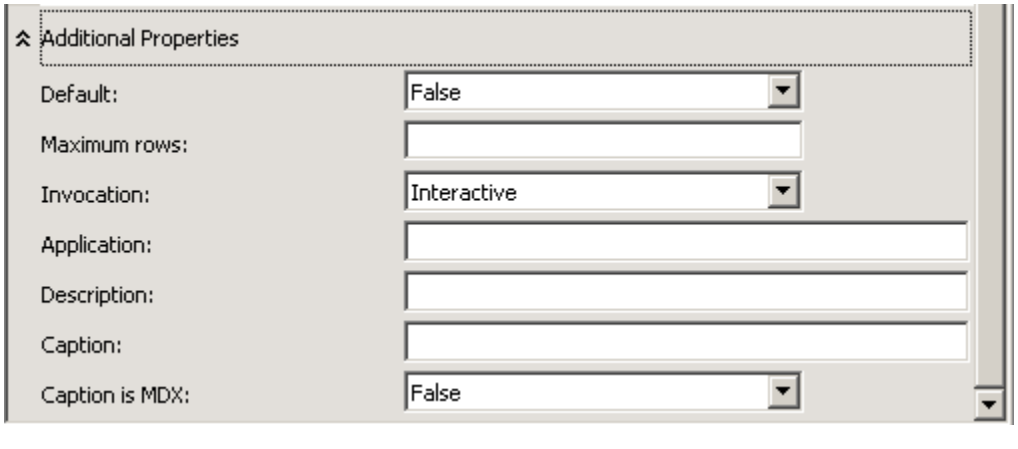
2. On the toolbar of the **Actions** tab, click the **New Drillthrough Action** button. A blank action template appears in the display pane.



3. In the **Name** box, change the name of this action to **Internet Sales Details Drillthrough Action**.
4. In the **Measure group members** list, select **Internet Sales**.
5. In the **Drillthrough Columns** box, select **Internet Sales Order Details** in the **Dimensions** list.
6. In the **Return Columns** list, select the **Item Description** and the **Order Number** check boxes, and then click **OK**. The following image shows the Action template as it should appear at this point in this procedure.

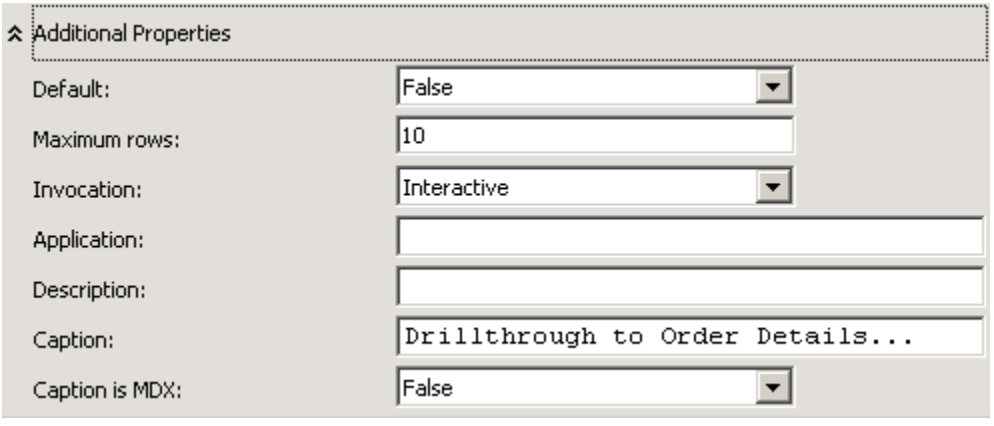


7. Expand the **Additional Properties** box, as shown in the following image.



8. In the **Maximum Rows** box, type **10**.
9. In the **Caption** box, type **Drillthrough to Order Details...**.

   These settings limit the number of rows returned and specify the caption that appears in the client application menu. The following image shows these settings in the **Additional Properties** box.



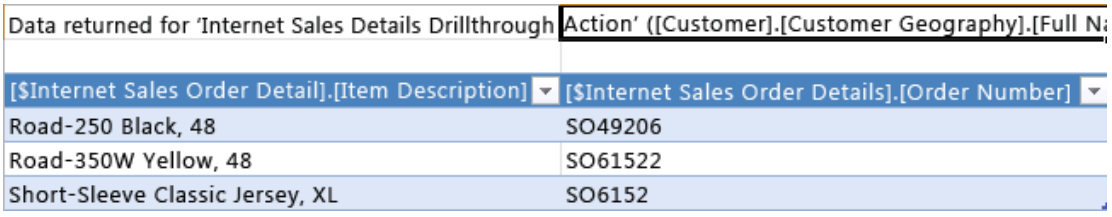
Using the Drillthrough Action

1. On the **Build** menu, click **Deploy Analysis Services Tutorial**.
2. When deployment has successfully completed, click the **Browser** tab in Cube Designer for the Analysis Services Tutorial cube, and then click the **Reconnect** button.
3. Start Excel.
4. Add the **Internet Sales-Sales Amount** measure to the Values area.
5. Add the **Customer Geography** user-defined hierarchy from the **Location** folder in the **Customer** dimension to the **Report Filter** area.
6. On the PivotTable, in **Customer Geography**, add a filter that selects a single customer. Expand **All Customers**, expand **Australia**, expand **Queensland**, expand **Brisbane**, expand **4000**, select the check box for **Adam Powell**, and then click **OK**.

   The total sales of products by Adventure Works Cycles to Adam Powell are displayed in the data area.
7. Right-click on the sales amount, point to **Additional Actions**, and then click **Drillthrough to Order Details**.

   The details of the orders that were shipped to Adam Powell are displayed in the **Data Sample Viewer**, as shown in the following image. However, some

additional details would also be useful, such as the order date, due date, and ship date. In the next procedure, you will add these additional details.



Data returned for 'Internet Sales Details Drillthrough	Action' ([Customer].[Customer Geography].[Full Na
[$Internet Sales Order Detail].[Item Description]	[$Internet Sales Order Details].[Order Number]
Road-250 Black, 48	SO49206
Road-350W Yellow, 48	SO61522
Short-Sleeve Classic Jersey, XL	SO6152

8. Close Excel/

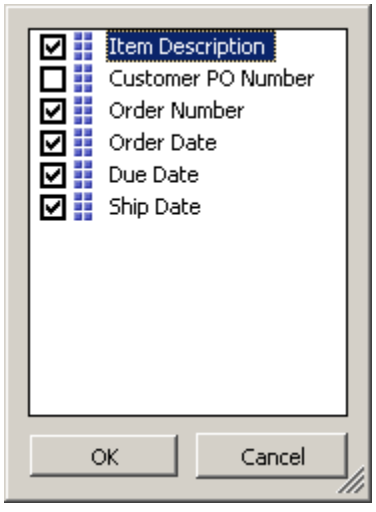
Modifying the Drillthrough Action

1. Open Dimension Designer for the **Internet Sales Order Details** dimension.
   Notice that only three attributes have been defined for this dimension.
2. In the **Data Source View** pane, right-click an open area, and then click **Show All Tables**.
3. On the **Format** menu, point to **Autolayout** and then click **Diagram**.
4. Locate the **InternetSales (dbo.FactInternetSales)** table by right-clicking in an open area of the **Data Source View** pane. Then click **Find Table,** click **InternetSales,** and click **OK**.
5. Create new attributes based on the following columns:
   - OrderDateKey
   - DueDateKey
   - ShipDateKey
6. Change the **Name** property for the **Order Date Key** attribute to **Order Date** Then, click the browse button for the **Name Column** property, and in the **Name Column** dialog box, select **Date** as the source table and select SimpleDate as the source column. Click .
7. Change the **Name** property for the **Due Date Key** attribute to **Due Date**, and then, by using the same method as the **Order Date Key** attribute, change the **Name Column** property for this attribute to **Date.SimpleDate (WChar)**.
8. Change the **Name** property for the **Ship Date Key** attribute to **Ship Date**, and then change the **Name Column** property for this attribute to **Date.SimpleDate (WChar)**.
9. Switch to the **Actions** tab of Cube Designer for the Analysis Services Tutorial cube.
10. In the **Drillthrough Columns** box, select the check boxes to add the following

columns to the **Return Columns** list, and then click **OK**:

- Order Date
- Due Date
- Ship Date

The following image shows these columns selected.



Reviewing the Modified Drillthrough Action

1. On the **Build** menu, click **Deploy Analysis Services Tutorial**.
2. When deployment has successfully completed, switch to the **Browser** tab in Cube Designer for the Analysis Services Tutorial cube, and then click the **Reconnect** button.
3. Start Excel.
4. Recreate the PivotTable using **Internet Sales-Sales Amount** in the Values area and **Customer Geography** in the Report Filter.

   Add a filter that selects from **All Customers**, **Australia**, **Queensland**, **Brisbane**, **4000**, **Adam Powell**.
5. Click the **Internet Sales-Sales Amount** data cell, point to **Additional Actions**, and then click **Drillthrough to Order Details**.

   The details of these orders shipped to Adam Powell are displayed in a temporary worksheet. This includes item description, order number, order date, due date, and ship date information, as shown in the following image.

[Item Description]	[Order Number]	[Order Date]	[Due Date]	[Ship Date]
Road-250 Black, 48	SO49206	20070204	20070216	20070211
Road-350-W Yellow, 48	SO61522	20080105	20080117	20080112
Short-Sleeve Clasic Jersey, XL	SO61522	20080105	20080117	20080112

Next Lesson

Lesson 9: Defining Perspectives and Translations

See Also

Actions

Defining and Configuring an Action

Dimension Relationships

Defining a Fact Relationship

Defining a Fact Relationship and Fact Relationship Properties

Lesson 9: Defining Perspectives and Translations

In this lesson, you learn to define perspectives and translations. You can define perspectives to reduce the apparent complexity of a cube, and define translations that let users view the cube metadata in the language of their choice.

Note
Completed projects for all of the lessons in this tutorial are available online. You can jump ahead to any lesson by using the completed project from the previous lesson as a starting point. Click here to download the sample projects that go with this tutorial.

This lesson contains the following tasks:

Defining and Browsing Perspectives

In this task, you define and browse perspectives to simplify the view of the cube for specific users or uses.

Defining and Browsing Translations

In this task, you define and browse translations of specific metadata to certain languages.

Next Lesson

Lesson 10: Defining Administrative Roles

See Also

Analysis Services Tutorial Scenario
SQL Server 2005 Analysis Services Tutorial
Perspectives
Defining and Configuring a Perspective
Dimension Translations
Cube Translations
Working with Translations (SSAS)

Defining and Browsing Perspectives

A perspective can simplify the view of a cube for specific purposes. By default, users can see all of the elements in a cube to which they have permissions. What users are viewing when they view an entire Analysis Services cube is the default perspective for the cube. A view of the whole cube can be very complex for users to navigate, especially for users who only need to interact with a small part of the cube to satisfy their business intelligence and reporting requirements.

To reduce the apparent complexity of a cube, you can create viewable subsets of the cube, called *perspectives*, which show users only a part of the measure groups, measures, dimensions, attributes, hierarchies, Key Performance Indicators (KPIs), actions, and calculated members in the cube. This can be particularly useful for working with client applications that were written for a previous release of Analysis Services. These clients have no concept of display folders or perspectives, for example, but a perspective appears to older clients as if it were a cube. For more information, see Perspectives (SSAS), and Defining and Configuring a Perspective.

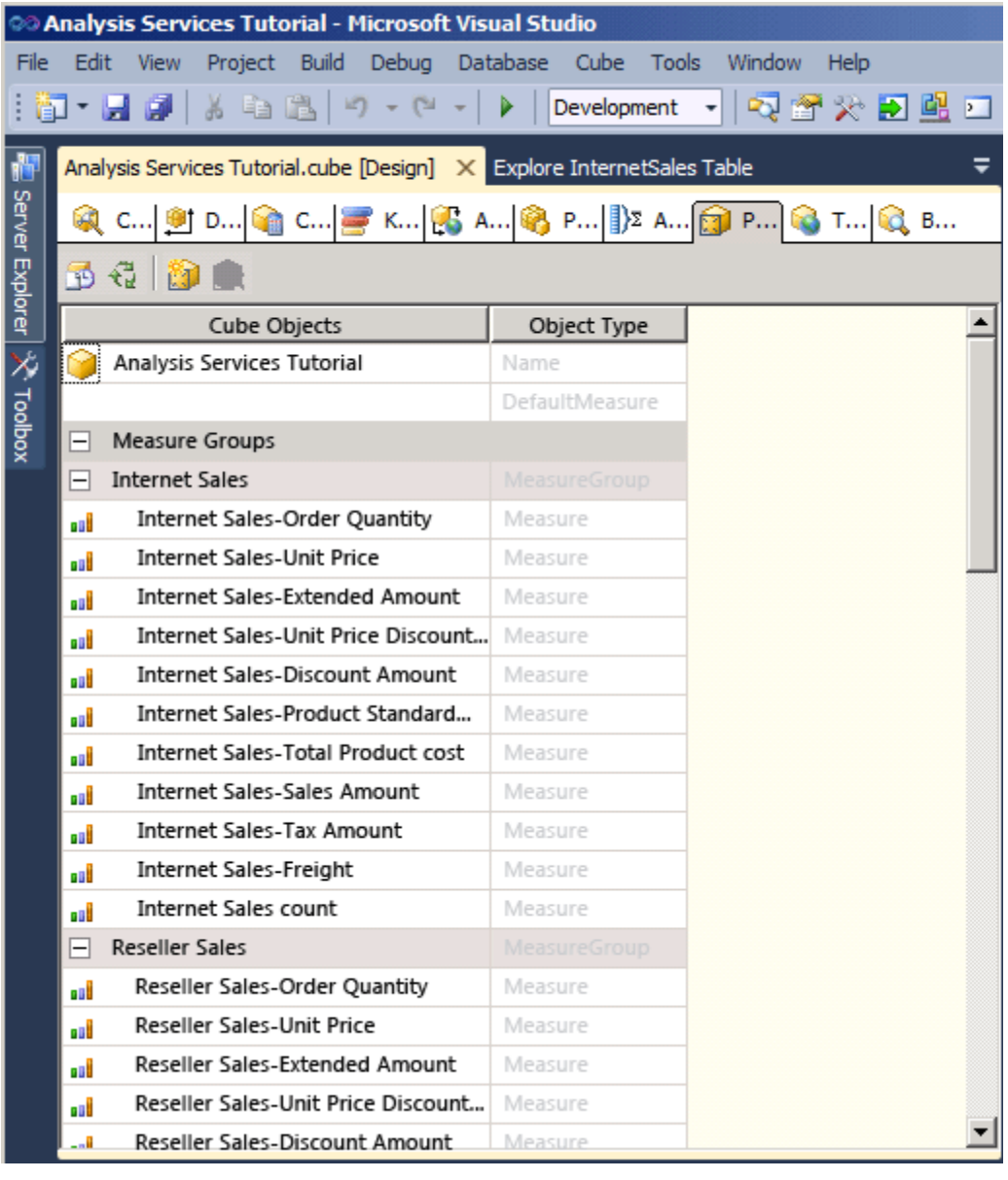
Note
A perspective is not a security mechanism, but instead is a tool for providing a better user experience. All security for a perspective is inherited from the underlying cube.

In the tasks in this topic, you will define several different perspectives and then browse the cube through each of these new perspectives.

Defining an Internet Sales Perspective

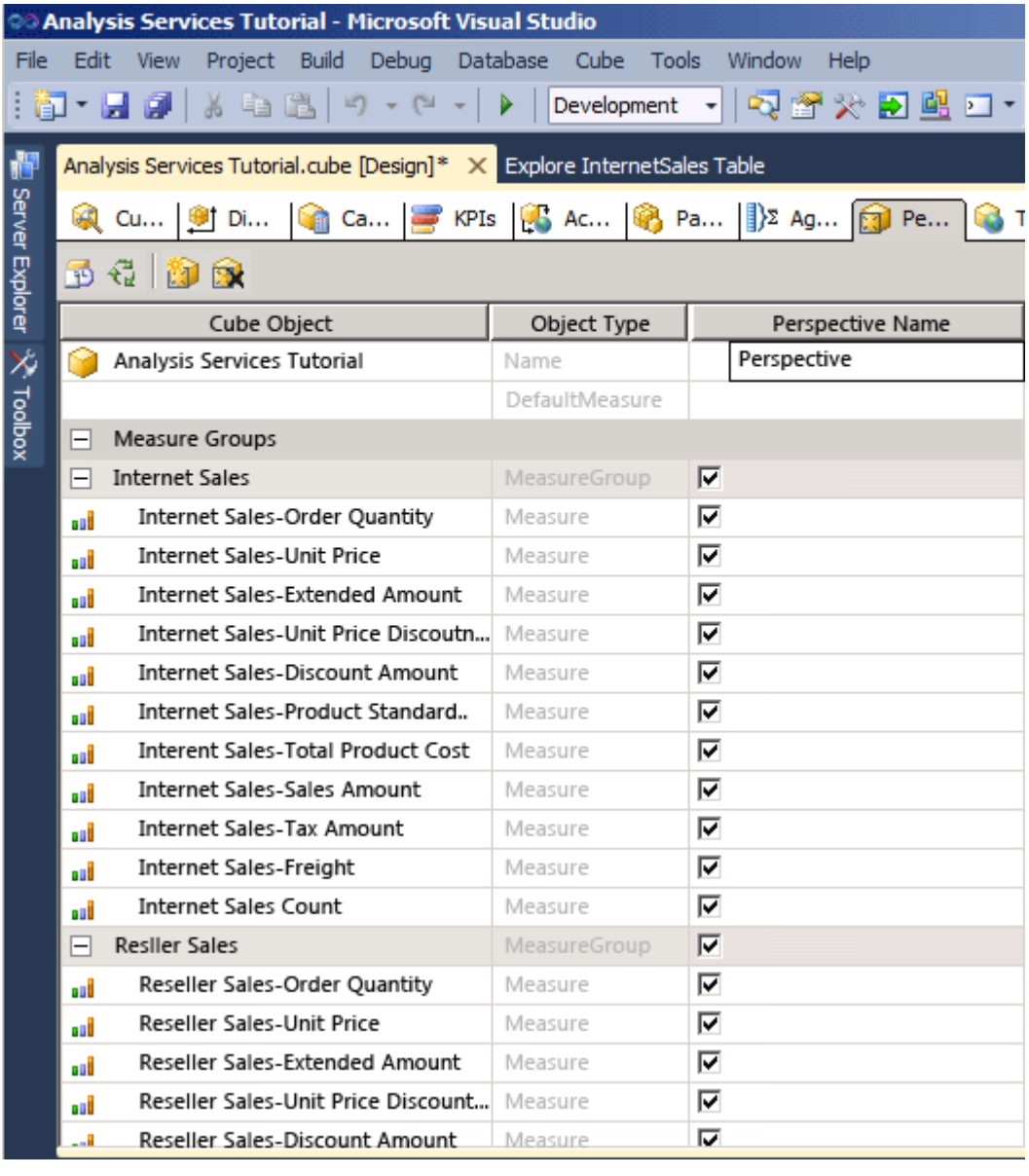
1. Open Cube Designer for the Analysis Services Tutorial cube, and then click the **Perspectives** tab.

   All the objects and their object types appear in the **Perspectives** pane, as shown in the following image.



2. On the toolbar of the **Perspectives** tab, click the **New Perspective** button.

   A new perspective appears in the **Perspective Name** column with a default name of **Perspective**, as shown in the following image. Notice that the check box for every object is selected; until you clear the check box for an object, this perspective is identical to the default perspective of this cube.



3. Change the perspective name to **Internet Sales**.
4. On the next row, set the DefaultMeasure to **Internet Sales-Sales Amount**.

   When users browse the cube by using this perspective, this will be the measure that the users see unless they specify some other measure.

> **Note**
> You can also set the default measure for the whole Analysis Services Tutorial cube in the Properties window on the **Cube Structure** tab for the cube.

5. Clear the check box for the following objects:
   - **Reseller Sales** measure group
   - **Sales Quotas** measure group
   - **Sales Quotas 1** measure group
   - **Reseller** cube dimension
   - **Reseller Geography** cube dimension
   - **Sales Territory** cube dimension
   - **Employee** cube dimension
   - **Promotion** cube dimension
   - **Reseller Revenue** KPI
   - **Large Resellers** named set
   - **Total Sales Amount** calculated member
   - **Total Product Cost** calculated member
   - **Reseller GPM** calculated member
   - **Total GPM** calculated member
   - **Reseller Sales Ratio to All Products** calculated member
   - **Total Sales Ratio to All Products** calculated member

   These objects do not relate to Internet sales.

   
   Note
   Within each dimension, you can also individually select the user-defined hierarchies and attributes that you want to appear in a perspective.

Defining a Reseller Sales Perspective

1. On the toolbar of the **Perspectives** tab, click the **New Perspective** button.
2. Change the name of the new perspective to **Reseller Sales**.
3. Set **Reseller Sales-Sales Amount** as the default measure.

   When users browse the cube by using this perspective, this measure will be the measure that the users will see unless they specify some other measure.
4. Clear the check box for the following objects:
   - **Internet Sales** measure group
   - **Internet Sales Reason** measure group
   - **Customer** cube dimension
   - **Internet Sales Order Details** cube dimension
   - **Sales Reason** cube dimension

- **Internet Sales Details Drillthrough Action** drillthrough action
- **Total Sales Amount** calculated member
- **Total Product Cost** calculated member
- **Internet GPM** calculated member
- **Total GPM** calculated member
- **Internet Sales Ratio to All Products** calculated member
- **Total Sales Ratio to All Products** calculated member

These objects do not relate to resellers sales.

Defining a Sales Summary Perspective

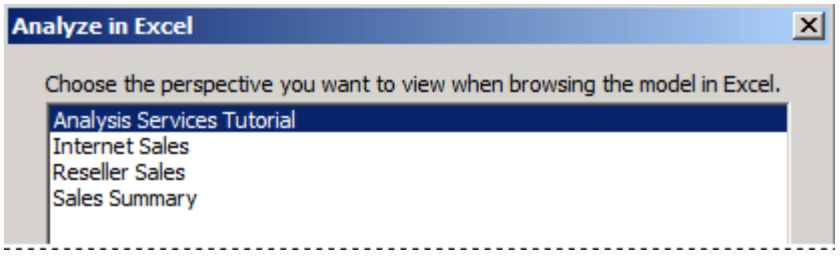
1. On the toolbar of the **Perspectives** tab, click the **New Perspective** button.
2. Change the name of the new perspective to **Sales Summary**.

   
   Note
   You cannot specify a calculated measure as the default measure.
3. Clear the check box for the following objects:
   - **Internet Sales** measure group
   - **Reseller Sales** measure group
   - **Internet Sales Reason** measure group
   - **Sales Quotas** measure group
   - **Sales Quotas1** measure group
   - **Internet Sales Order Details** cube dimension
   - **Sales Reason** cube dimension
   - **Internet Sales Details Drillthrough Action** drillthrough action
4. Select the check box for the following objects:
   - **Internet Sales Count** measure
   - **Reseller Sales Count** measure

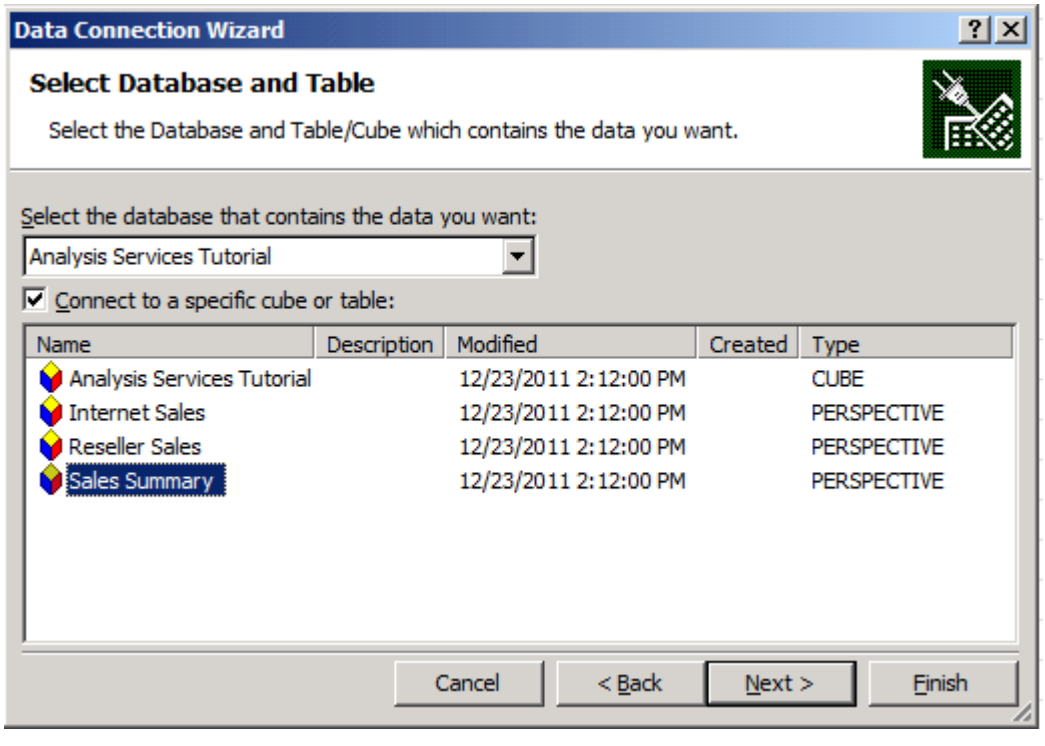
Browsing the Cube Through Each Perspective

1. On the **Build** menu, click **Deploy Analysis Services Tutorial**.
2. When deployment has successfully completed, switch to the **Browser** tab, and then click the **Reconnect** button.
3. Start Excel.
4. Analyze in Excel prompts you to choose which perspective to use when browsing

the model in Excel, as shown in the following image.



5. Alternatively, you can start Excel from the Windows Start menu, define a connection to the Analysis Services Tutorial database on localhost, and choose a perspective in the Data Connection wizard, as shown in the following image.



6. Select **Internet Sales** in the **Perspective** list and then review the measures and dimensions in the metadata pane.

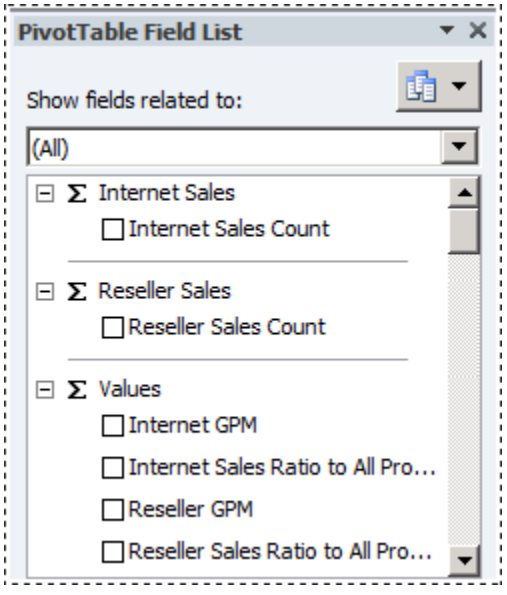
   Notice that only those objects that are specified for the Internet Sales perspective appear.

7. In the metadata pane, expand **Measures**.

   Notice that only the **Internet Sales** measure group appears, together with the **Internet GPM** and **Internet Sales Ratio to All Products** calculated members.

8. In the model, select Excel again. Select **Sales Summary**.

Notice that in each of these measure groups, only a single measure appears, as shown in the following image.



Next Task in Lesson

Defining and Browsing Translations

See Also

Perspectives (SSAS)

Defining and Configuring a Perspective

Defining and Browsing Translations

A translation is a representation of the names of Analysis Services objects in a specific language. Objects include measure groups, measures, dimensions, attributes, hierarchies, KPIs, actions, and calculated members. Translations provide server support for client applications that can support multiple languages. By using such a client, the client passes the locale identifier (LCID) to the instance of Analysis Services, which uses the LCID to determine which set of translations to use when it provides metadata for Analysis Services objects. If an Analysis Services object does not contain a translation for that language, or does not contain a translation for a specified object, the default language is used in returning the object metadata back to the client. For example, if a business user in France accesses a cube from a workstation that has a French locale setting, the business user will see the member captions and member property values in French if a French translation exists. However, if a business user in Germany accesses the same cube from a workstation that has a German locale setting, the business user will see the

captions names and member property values in German. For more information, see Dimension Translations, Cube Translations, Working with Translations (SSAS).

In the tasks in this topic, you define metadata translations for a limited set of dimension objects in the Date dimension and cube objects in the Analysis Services Tutorial cube. You will then browse these dimension and cube objects to examine the metadata translations.

Specifying Translations for the Date Dimension Metadata

1. Open Dimension Designer for the **Date** dimension, and then click the **Translations** tab.

   The metadata in the default language for each dimension object appears. The default language in the Analysis Services Tutorial cube is English.
2. On the toolbar of the **Translations** tab, click the **New Translation** button.

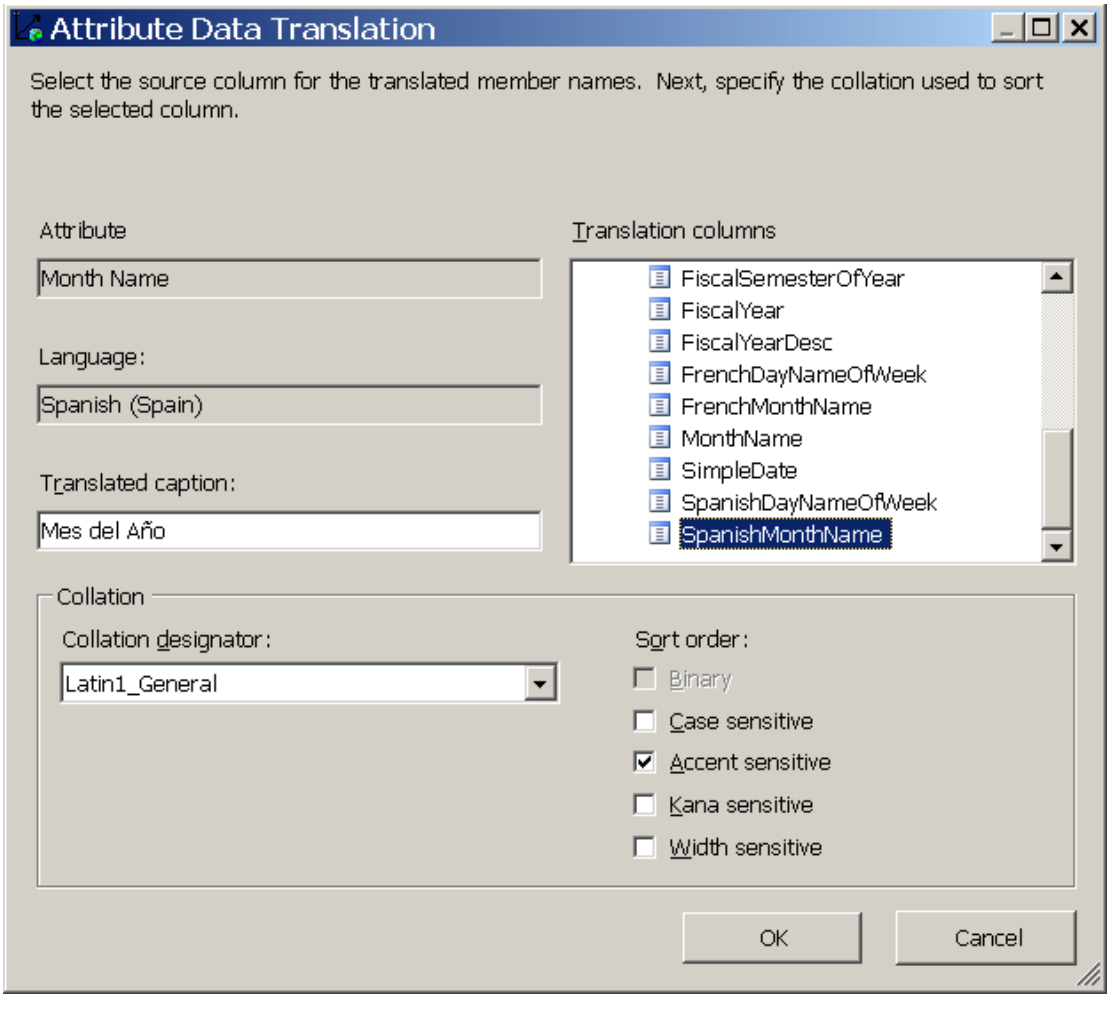
   A list of languages appears in the **Select Language** dialog box.
3. Click **Spanish (Spain)**, and then click **OK**.

   A new column appears in which you will define the Spanish translations for the metadata objects you want to translate. In this tutorial, we will only translate a few objects just to illustrate the process.
4. On the toolbar of the **Translations** tab, click the **New Translation** button, click **French (France)** in the **Select Language** dialog box, and then click **OK**.

   Another language column appears in which you will define French translations.
5. In the row for the **Caption** object for the **Date** dimension, type **Fecha** in the **Spanish (Spain)** translation column and **Temps** in the **French (France)** translation column.
6. In the row for the **Caption** object for the **Month Name** attribute, type **Mes del Año** in the **Spanish (Spain)** translation column and **Mois d'Année** in the **French (France)** translation column.

   Notice that when you enter these translations, an ellipsis (**...**) appears. Clicking this ellipsis will enable you to specify a column in the underlying table that provides translations for each member of the attribute hierarchy.
7. Click the ellipsis (**...**) for the **Spanish (Spain)** translation for the **Month Name** attribute.

   The **Attribute Data Translation** dialog box appears.
8. In the **Translation columns** list, select **SpanishMonthName**, as shown in the following image.



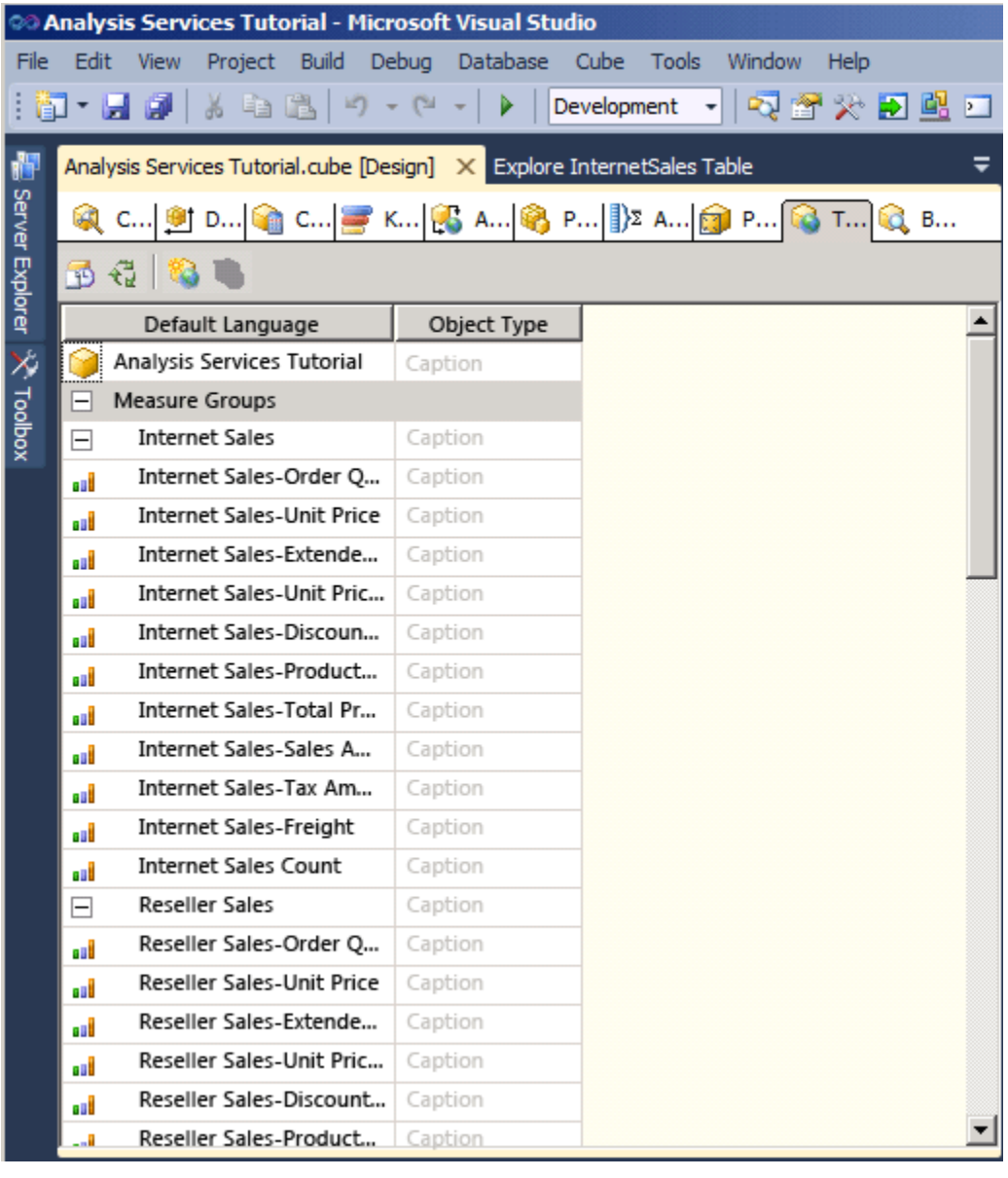
9. Click **OK**, and then click the ellipsis (**...**) for the **French (France)** translation for the **Month Name** attribute.
10. In the **Translation columns** list, select **FrenchMonthName**, and then click **OK**.

    The steps in this procedure illustrate the process of defining metadata translations for dimension objects and members.

Specifying Translations for the Analysis Services Tutorial Cube Metadata

1. Switch to Cube Designer for the Analysis Services Tutorial cube, and then switch to the **Translations** tab.

   The metadata in the default language for each cube object appears, as shown in the following image. The default language in the Analysis Services Tutorial cube is English.



2. On the toolbar of the **Translations** tab, click the **New Translation** button.

   A list of languages appears in the **Select Language** dialog box.

3. Select **Spanish (Spain)**, and then click **OK**.

   A new column appears in which you will define the Spanish translations for the metadata objects you want to translate. In this tutorial, we will only translate a few objects just to illustrate the process.

4. On the toolbar of the **Translations** tab, click the **New Translation** button, select **French (France)** in the **Select Language** dialog box, and then click **OK**.

Another language column appears in which you will define French translations.

5. In the row for the **Caption** object for the **Date** dimension, type **Fecha** in the **Spanish (Spain)** translation column and **Temps** in the **French (France)** translation column.
6. In the row for the **Caption** object for the **Internet Sales** measure group, type **Ventas del Internet** in the **Spanish (Spain)** translation column and **Ventes D'Internet** in the **French (France)** translation column.
7. In the row for the **Caption** object for the Internet Sales-Sales Amount measure, type **Cantidad de las Ventas del Internet** in the **Spanish (Spain)** translation column and **Quantité de Ventes d'Internet** in the **French (France)** translation column.

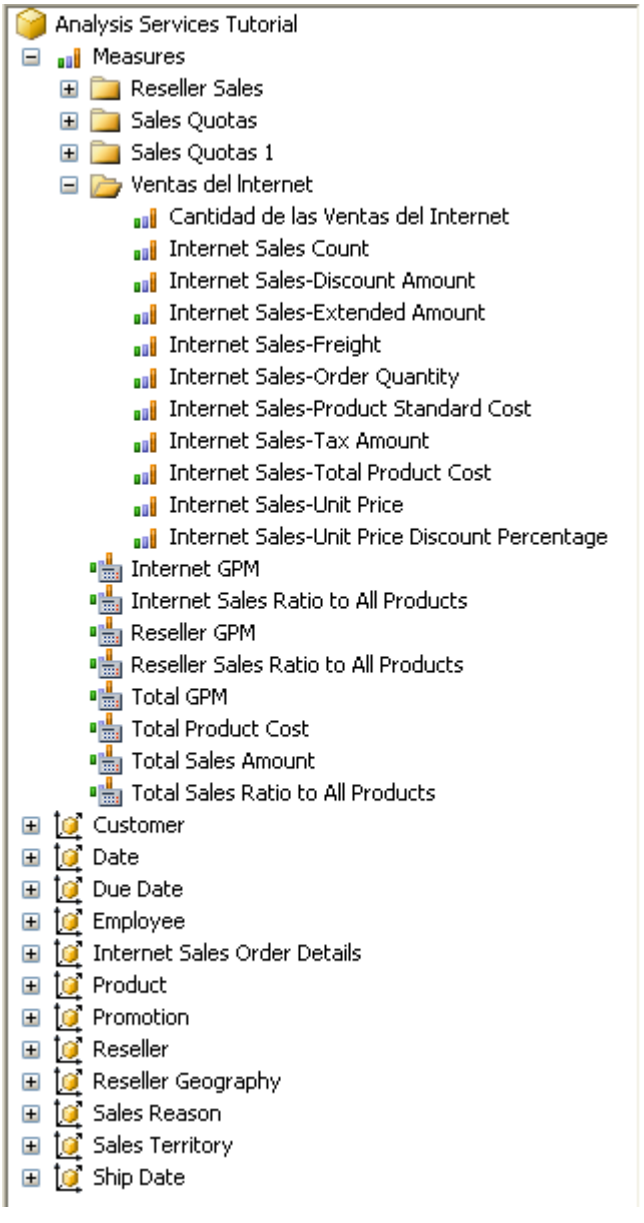
   The steps in this procedure illustrate the process of defining metadata translations for cube objects.

Browsing the Cube By Using Translations

1. On the **Build** menu, click **Deploy Analysis Services Tutorial**.
2. When deployment has successfully completed, switch to the **Browser** tab, and then click **Reconnect**.
3. Remove all hierarchies and measures from the **Data** pane and select Analysis Services Tutorial in the **Perspectives** list.
4. In the metadata pane, expand **Measures** and then expand **Internet Sales**.

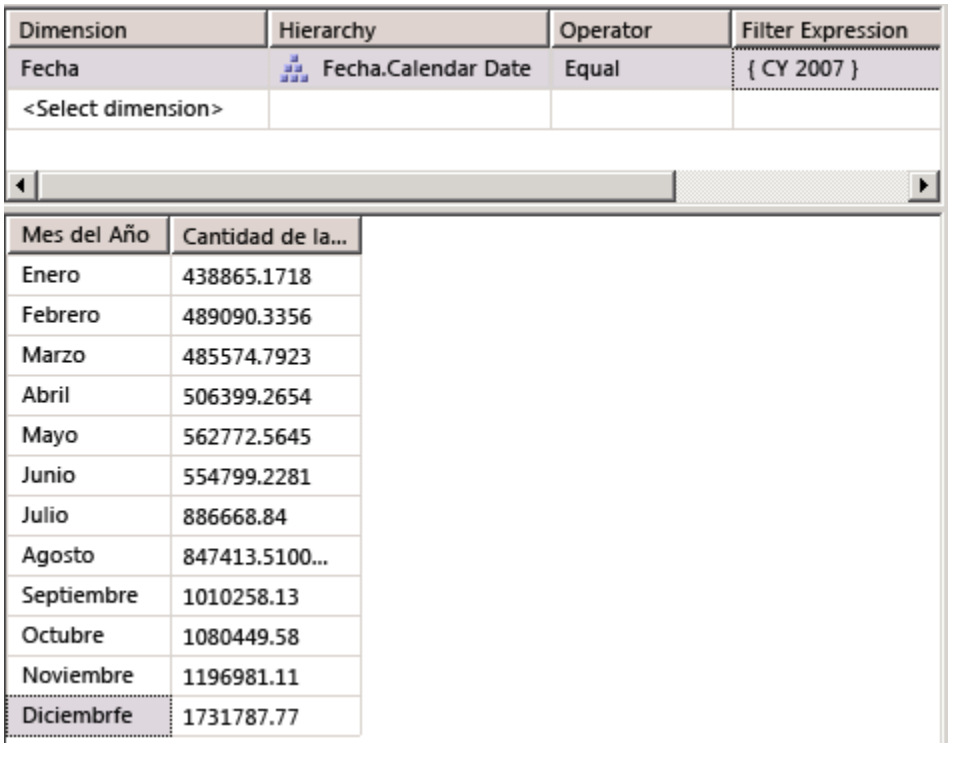
   Notice that the **Internet Sales-Sales Amount** measure appears in English in this measure group.
5. On the toolbar, select **Spanish (Spain)** in the **Language** list.

   Notice that the items in the metadata pane are repopulated. After the items in the metadata pane are repopulated, notice that the Internet Sales-Sales Amount measure no longer appears in the Internet Sales display folder. Instead, it appears in Spanish in a new display folder named **Ventas del Internet**, as shown in the following image.



6. In the metadata pane, right-click **Cantidad de las Ventas del Internet** and then select **Add to Query**.
7. In the metadata pane, expand **Fecha**, expand **Fecha.Calendar Date**, right-click **Fecha.Calendar Date**, and then select **Add to Filter**.
8. In the **Filter** pane, select **CY 2007** as the filter expression.
9. In the metadata pane, right-click **Mes del Ano** and select **Add to Query**.

   Notice that the month names appear in Spanish, as shown in the following image.



Dimension	Hierarchy	Operator	Filter Expression
Fecha	Fecha.Calendar Date	Equal	{ CY 2007 }
<Select dimension>			

Mes del Año	Cantidad de la...
Enero	438865.1718
Febrero	489090.3356
Marzo	485574.7923
Abril	506399.2654
Mayo	562772.5645
Junio	554799.2281
Julio	886668.84
Agosto	847413.5100...
Septiembre	1010258.13
Octubre	1080449.58
Noviembre	1196981.11
Diciembrfe	1731787.77

10. On the toolbar, select **French (France)** in the **Language** list.

    Notice that the month names now appear in French and that the measure name now also appears in French.

Next Lesson

Lesson 10: Defining Administrative Roles

See Also

Dimension Translations

Cube Translations

Working with Translations (SSAS)

Lesson 10: Defining Administrative Roles

In this lesson, you learn to define security roles for administrative tasks.

Note

Completed projects for all of the lessons in this tutorial are available online. You can jump ahead to any lesson by using the completed project from the previous

lesson as a starting point. Click here to download the sample projects that go with this tutorial.

This lesson contains the following task:

Granting Process Database Permissions

In this task, you define a security role that has permissions to process the Analysis Services database, and then you test this security role.

See Also

Analysis Services Tutorial Scenario

Analysis Services Tutorial

Granting Process Database Permissions

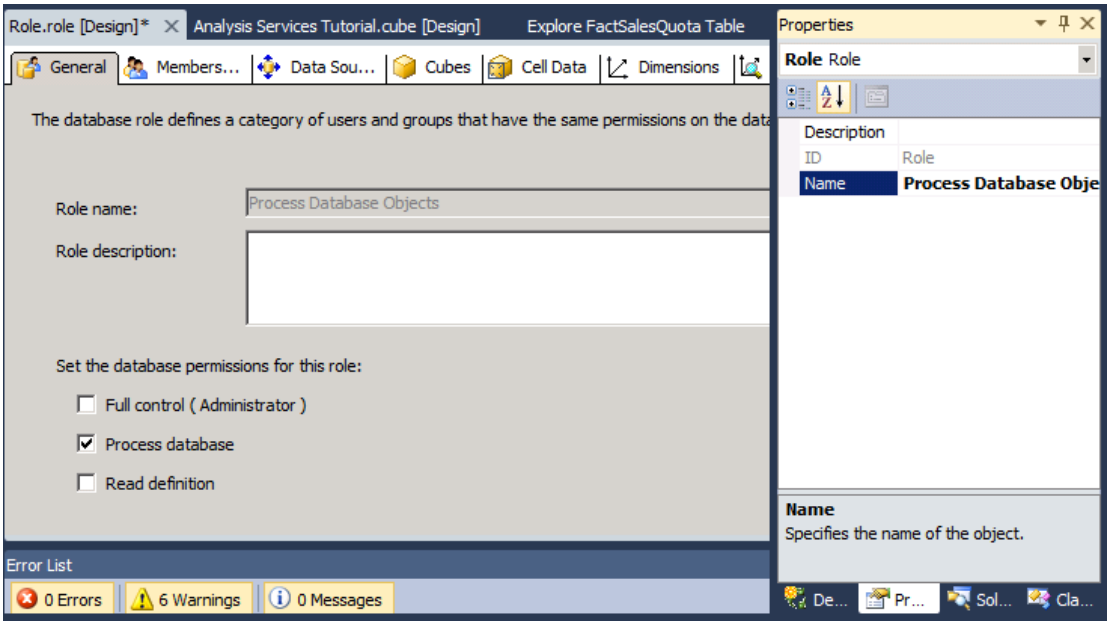
After you install an instance of Analysis Services, all members of the Analysis Services server administrator role in that instance have server-wide permissions to perform any task within the instance of Analysis Services. By default, no other users have any permission to administer or view any objects in the instance of Analysis Services.

A member of the server administrator role can grant users administrative access on a server-wide basis by making them members of the role. A member of the server administrator role can also grant users access on a more limited basis by granting them limited or complete administrative or access permissions at the database level. Limited administrative permissions include process or read definition permissions at the database, cube, or dimension level.

In the tasks in this topic, you will define a Process Database Objects security role that grants members of the role permission to process all database objects, but no permission to view data within the database.

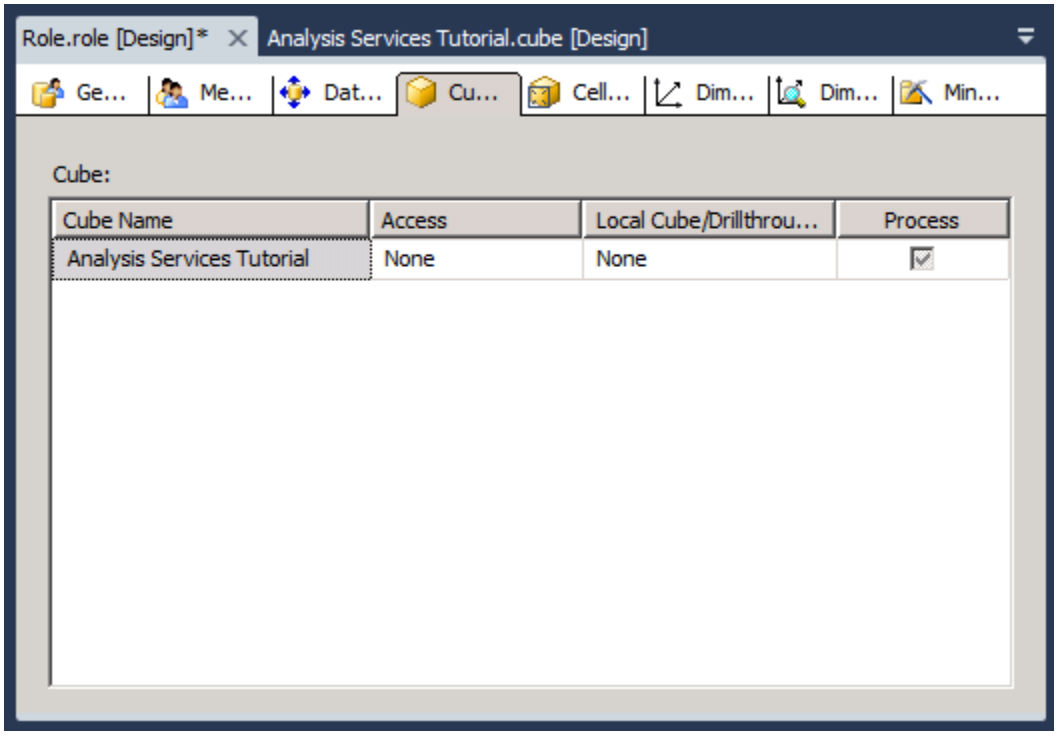
Defining a Process Database Objects Security Role

1. In Solution Explorer, right-click **Roles** and then click **New Role** to open the Role Designer.
2. Click the **Process database** check box.
3. In the Properties window, change the **Name** property for this new role to **Process Database Objects Role**.



4. Switch to the **Membership** tab of Role Designer and click **Add**.
5. Enter the accounts of the Windows domain users or groups who will be members of this role. Click **Check Names** to verify the account information, and then click **OK**.
6. Switch to the **Cubes** tab of Role Designer.

   Notice that members of this role have permissions to process this database, but have no permission to access the data in the Analysis Services Tutorial cube and have no local cube/drillthrough access, as shown in the following image.



7. Switch to the **Dimensions** tab of Role Designer.

   Notice that members of this role have permissions to process all dimension objects in this database, and, by default, have read permissions to access each dimension object in the Analysis Services Tutorial database.

8. On the **Build** menu, click **Deploy Analysis Services Tutorial**.

   You have now successfully defined and deployed the Process Database Objects security role. After a cube is deployed to the production environment, the administrators of the deployed cube can add users to this role as required to delegate processing responsibilities to specific users.

Note

A completed project for Lesson 10 is available by downloading and installing the samples. For more information, see Install Sample Data and Projects for the Analysis Services Multidimensional Modeling Tutorial.

See Also

Multidimensional Model Roles and Permissions