**The laboratory work 7**

**Defining Advanced Attribute and Dimension Properties**

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](https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks-analysis-services) to go to the download page that provides the sample projects that go with this tutorial. The tutorial projects are included in the **adventure-works-multidimensional-tutorial-projects.zip** download.
2. Click **adventure-works-multidimensional-tutorial-projects.zip** 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.

1. Create a subfolder at or near the root drive, for example, C:\Tutorial.
2. Move the **adventure-works-multidimensional-tutorial-projects.zip** file to the subfolder.
3. Right-click the file and select **Extract All**.
4. 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.

1. 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 **AdventureWorksDW2012** database.

Notice that with ten tables in the data source view, the 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 **AdventureWorksDW2012** 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.

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 [Parent-Child Dimensions](https://docs.microsoft.com/en-us/analysis-services/multidimensional-models/parent-child-dimension?view=asallproducts-allversions), [Attributes in Parent-Child Hierarchies](https://docs.microsoft.com/en-us/analysis-services/multidimensional-models/parent-child-dimension-attributes?view=asallproducts-allversions)

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.

1. 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.

1. 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**.

1. In Solution Explorer, double-click **Analysis Services Tutorial.cube** in the **Cubes** folder to open Cube Designer for the Analysis Services Tutorial cube.
2. Click the **Browser** tab.
3. Click the Excel icon, and then click **Enable** when prompted to enable connections.
4. In the PivotTable Field List, expand **Reseller Sales**. Drag **Reseller Sales-Sales Amount** to the Values area.
5. 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.

1. 

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.

1. 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.

1. 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.



1. Click **OK**.
2. 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, 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.



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 [Group Attribute Members (Discretization)](https://docs.microsoft.com/en-us/analysis-services/multidimensional-models/attribute-properties-group-attribute-members?view=asallproducts-allversions)

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.

1. Close the **Explore Customer Table** tab.
2. In the **Attributes** pane, select **Yearly Income**.
3. 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.

1. Close the **Explore Employee Table** tab.
2. In the **Attributes** pane, select **Sick Leave Hours**.
3. In the Properties window, change the value for the **DiscretizationMethod** property to **Clusters** and change the value for the **DiscretizationBucketCount** property to **5**.
4. In the **Attributes** pane, select **Vacation Hours**.
5. 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, 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.

1. Remove the **Yearly Income** attribute hierarchy from the columns area and remove the **Internet Sales-Sales Amount** measure from the **Values** area.
2. Add the **Reseller Sales-Sales Amount** measure to the data area.
3. 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.



1. Remove the **Sick Leave Hours** attribute hierarchy from the column area of the **Data** pane.
2. 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:
	1. Drag Reseller Sales-Sales Amount to Values
	2. Drag Vacation Hours (in the Employees Organization folder) to Columns
	3. 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.)

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.



1. On the **Build** menu, click **Deploy Analysis Services Tutorial**.
2. 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.



1. 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.

1. On the **Build** menu, click **Deploy Analysis Services Tutorial**.
2. When deployment has successfully completed, switch to the **Browser** tab for the Customer dimension, and then click **Reconnect**.
3. 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.

1. 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.

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 Date Dimension](https://docs.microsoft.com/en-us/analysis-services/multidimensional-tutorial/lesson-3-4-modifying-the-date-dimension?view=asallproducts-allversions). 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](https://docs.microsoft.com/en-us/analysis-services/multidimensional-models-olap-logical-dimension-objects/attribute-relationships?view=asallproducts-allversions), [Define Attribute Relationships](https://docs.microsoft.com/en-us/analysis-services/multidimensional-models/attribute-relationships-define?view=asallproducts-allversions), [Configure Attribute Relationship Properties](https://docs.microsoft.com/en-us/analysis-services/multidimensional-models/attribute-relationships-configure-attribute-properties?view=asallproducts-allversions), and [Specifying Attribute Relationships Between Attributes in a User-Defined Hierarchy](https://docs.microsoft.com/en-us/analysis-services/multidimensional-tutorial/lesson-4-6-specifying-attribute-relationships-in-user-defined-hierarchy?view=asallproducts-allversions).

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.

1. 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.

1. 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**.
2. 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.

1. 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.

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 **Month Number Of Year**.
3. 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**.

1. Click **OK**.
2. Click the **Dimension Structure** tab.
3. 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**.
4. On the **Build** menu, click **Deploy Analysis Services Tutorial**.
5. 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 [Define Named Calculations in a Data Source View (Analysis Services)](https://docs.microsoft.com/en-us/analysis-services/multidimensional-models/define-named-calculations-in-a-data-source-view-analysis-services?view=asallproducts-allversions).

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



1. 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**.
2. 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.

1. 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**.

1. Click the **Attribute Relationships** tab.
2. In the attributes list, right-click the **Commute Distance** attribute and then select **New Attribute Relationship**.
3. In the **Create Attribute Relationship** dialog box, the **Source Attribute** is **Commute Distance**. Set the **Related Attribute** to **Commute Distance Sort**.
4. 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.

1. Click **OK**.

You are now ready to set the sort order for the **Commute Distance** attribute.

1. Click the **Dimension Structure** tab.
2. 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**.
3. On the **Build** menu, click **Deploy Analysis Services Tutorial**.
4. 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.



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 [Define Attribute Relationships](https://docs.microsoft.com/en-us/analysis-services/multidimensional-models/attribute-relationships-define?view=asallproducts-allversions) and [Attribute Relationships](https://docs.microsoft.com/en-us/analysis-services/multidimensional-models-olap-logical-dimension-objects/attribute-relationships?view=asallproducts-allversions).

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 [Define Attribute Relationships](https://docs.microsoft.com/en-us/analysis-services/multidimensional-models/attribute-relationships-define?view=asallproducts-allversions), and [User Hierarchy Properties](https://docs.microsoft.com/en-us/analysis-services/multidimensional-models-olap-logical-dimension-objects/user-hierarchies-properties?view=asallproducts-allversions).

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.

1. 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.

1. In the diagram, right-click the **Full Name** attribute and then select **New Attribute Relationship**.
2. 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.
3. Click **OK**.

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.

1. In the **Attribute Relationships** pane, right-click **Geography**-> **Postal Code** and then click **Delete**.
2. When the **Delete Objects** dialog box appears, click **OK**.
3. In the diagram, right-click the **Postal Code** attribute and then select **New Attribute Relationship**.
4. 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**.
5. Click **OK**.

The relationship **Geography**-> **City** is now redundant so we will delete it.

1. In the Attribute Relationships pane, right-click **Geography**-> **City** and then click **Delete**.
2. When the **Delete Objects** dialog box appears, click **OK**.
3. In the diagram, right-click the **City** attribute and then select **New Attribute Relationship**.
4. 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.
5. Click **OK**.
6. Right-click the arrow between **Geography** and **State-Province** and then click **Delete**.
7. When the **Delete Objects** dialog box appears, click **OK**.
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** because the relationship between a state-province and a country-region will not change over time.
10. Click **OK**.
11. In the Attribute Relationships pane, right-click **Geography**-> **Country-Region** and then click **Delete**.
12. When the **Delete Objects** dialog box appears, click **OK**.
13. 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.

1. 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 **OK**.

**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 **OK**.

## 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 **OK**.
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 **OK**.
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 **OK**.

## 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 **OK**.
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 **OK**.
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 **OK**.

## 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 **OK**.
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 **OK**.
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 **OK**.
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, click **Deploy Analysis Services Tutorial**.

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.

1. 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.

1. 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.

1. 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.



1. 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.

1. 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.

The **Add/Remove Tables** dialog box opens.

1. 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.

1. 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.

1. Click **OK**.
2. 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.

1. Switch to Dimension Designer for the **Product** dimension, and then click the **Dimension Structure** tab.
2. Right-click anywhere in the **Data Source View** pane, and then click **Show All Tables**.
3. In the **Data Source View** pane, locate the **DimProductCategory** table, right-click **ProductCategoryKey** in that table, and then click **New Attribute from Column**.
4. In the **Attributes** pane, change the name of this new attribute to **Category**.
5. In the Properties window, click in the **NameColumn** property field and then click the browse (**...**) button to open the **Name Column** dialog box.
6. Select **EnglishProductCategoryName** in the **Source column** list and then click **OK**.
7. In the **Data Source View** pane, locate the **DimProductSubcategory** table, right-click **ProductSubcategoryKey** in that table, and then click **New Attribute from Column**.
8. In the **Attributes** pane, change the name of this new attribute to **Subcategory**.
9. In the Properties window, click in the **NameColumn** property field and then click the browse **(...)** button to open the **Name Column** dialog box.
10. Select **EnglishProductSubcategoryName** in the **Source column** list and then click **OK**.
11. Create a new user-defined hierarchy called **Product Categories** with the following levels, in order from top to bottom: **Category**, **Subcategory**, and **Product Name**.
12. 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.



1. Click **Close** three times to close all processing dialog boxes.
2. Click the **Browser** tab in Dimension Designer for the **Product** dimension, and then click **Reconnect**.
3. Verify that **Product Model Lines** appears in the **Hierarchy** list, expand **All Products**, and then expand **Components**.
4. 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.

1. Click the **Attribute Relationships** tab.
2. In the diagram, right-click the **Subcategory** attribute and then select **New Attribute Relationship**.
3. 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**.
4. Click **OK**.
5. In the **Attributes** pane, select **Subcategory.**
6. In the Properties window, expand the **KeyColumns** property and then expand the **DimProductSubcategory.ProductSubcategoryKey (Integer)** property.
7. Change the **NullProcessing** property to **UnknownMember**.
8. In the **Attributes** pane, select **Model Name**.
9. In the Properties window, expand the **KeyColumns** property and then expand the **Product.ModelName (WChar)** property.
10. 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.

1. 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.

