p

Microsoft

DAT224x

Developing a Multidimensional Data Model

Lab 03 | Creating the Dimensions

Estimated time to complete this lab is 90 minutes

Overview

In this lab, you will create dimensions in the **Reseller Sales** Multidimensional Project.

Note: The four labs in this course are accumulative. You cannot complete this lab if you did not successfully complete **Lab 02**.

Getting Started

In this exercise, you will start the VM provisioned in **Lab 01**. You will then connect to the VM to complete the exercises in this lab.

1. Sign in to the **Azure Portal** by using your subscription.

2. In the left pane, select **Virtual Machines**—do not select **Virtual Machines (Classic)**.
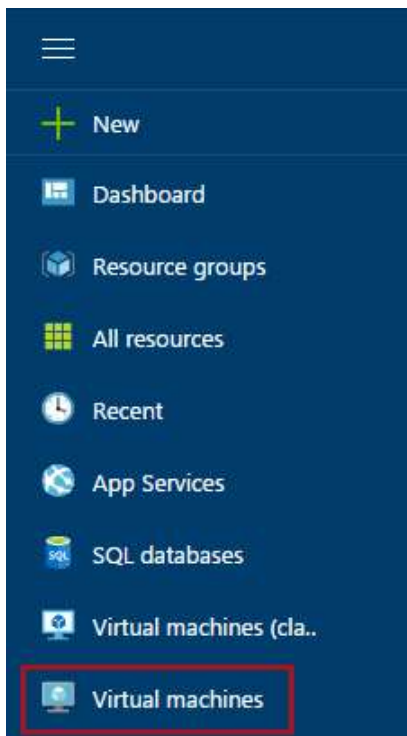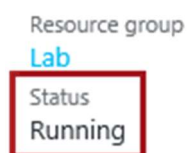

3. In the **Virtual Machines** blade, select the VM you provisioned in **Lab 01**.
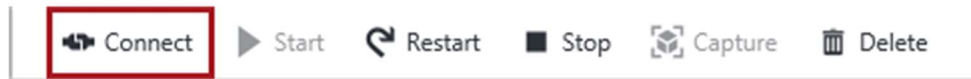
4. In the VM blade, click **Start**.

5. Wait for the VM status to update to **Running**.

   *It usually takes 1-2 minutes for the VM to start.*

6. To connect to the VM, click **Connect**.

    Take care not to use the RDP file downloaded in the previous lab. It is likely that a different IP address has be assigned.

7. When prompted to open the Remote Desktop File, click **Open**.

8. If prompted to connect to the unknown publisher, click **Connect**.

9. If prompted, in the **Windows Security** dialog window, click **Use Another Account**.
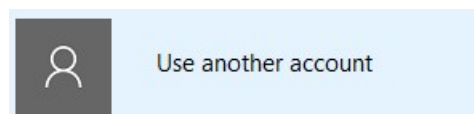
10. Enter the credentials you created for your VM.

11. Check the **Remember My Credentials** checkbox.
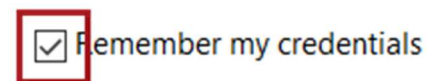
12. Click **OK**.

13. In the **Remote Desktop Connection** dialog window, check the **Don't Ask Me Again for Connections to This Computer** checkbox.
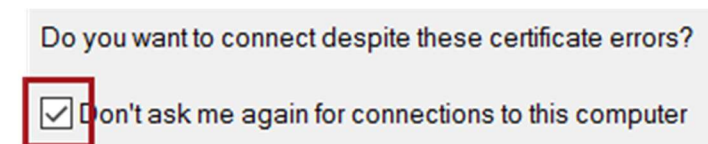
14. Click **Yes**.

Exercise 1: Developing the Date Dimension

In this exercise, you will develop the **Date** dimension. This dimension will consist of four attributes to define fiscal date periods. To support efficient and convenient navigation, you will also define a multi-level hierarchy to provide drill down and drill up paths. These paths will allow users to summarize cube measures at different levels of granularity.

Opening the Project

In this task, you will open the Multidimensional Project. Only complete this task if the project is not already open.

1. To launch SSDT, on the task bar, click the **Visual Studio 2015** shortcut.
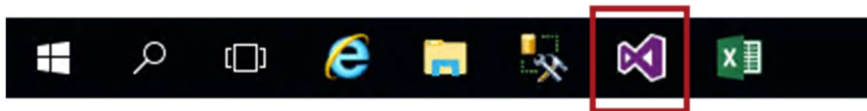
2. On the **File** menu, select **Recent Project and Solutions**, and then select the first project (representing the project you last opened).

   If required, the F:\Labs\Lab02\Solution folder contains the previous lab's solution.

Preparing the Date Dimension Data

In this task, in the **Reseller Sales** data source view, you will replace the **Date** table with a named query.

1. Open the **Reseller Sales** data source view.

2. In the Tables pane, right-click the **Date** table, and then select **Replace Table | With New Named Query**.

3. In the **Create Named Query** window, in the **Name** box, replace the text with **Date**.

4. To hide the diagram and grid panes, click the following toolbar buttons.

---

5. Replace the query in the SQL pane with the following.

*You can copy the query from the **F:\Labs\Lab03\Assets\Snippets.txt** file.*

T-SQL

```
SELECT
  [DateKey],
  [FullDateAlternateKey] AS [Date],
  CAST([CalendarYear] AS NCHAR(4)) + N' ' + LEFT([EnglishMonthName], 3) + N', '
        CASE WHEN [DayNumberOfMonth] < 10 THEN N'0' ELSE N'' END +
        CAST([DayNumberOfMonth] AS NVARCHAR(2)) AS [DateLabel],
  ((([CalendarYear] * 100) + [MonthNumberOfYear]) AS [MonthKey],
  CAST([CalendarYear] AS NCHAR(4)) + N' ' + LEFT([EnglishMonthName], 3) AS [MonthLabel],
  [MonthNumberOfYear] AS [MonthOfYearKey],
  LEFT([EnglishMonthName], 3) AS [MonthOfYearLabel],
  ((([CalendarYear] * 10) + [CalendarQuarter]) AS [CalendarQuarterKey],
  N'CY' + CAST([CalendarYear] AS NCHAR(4)) +
        N' Q' + CAST([CalendarQuarter] AS NCHAR(1)) AS [CalendarQuarterLabel],
  [CalendarQuarter] AS [CalendarQuarterOfYearKey],
  N'CY Q' + CAST([CalendarQuarter] AS NCHAR(1)) AS [CalendarQuarterOfYearLabel],
  [CalendarYear] AS [CalendarYearKey],
  N'CY' + CAST([CalendarYear] AS NCHAR(4)) AS [CalendarYearLabel],
  ((([FiscalYear] * 10) + [FiscalQuarter]) AS [FiscalQuarterKey],
  N'FY' + CAST([FiscalYear] AS NCHAR(4)) + N' Q' +
        CAST([FiscalQuarter] AS NCHAR(1)) AS [FiscalQuarterLabel],
  [FiscalQuarter] AS [FiscalQuarterOfYearKey],
  N'FY Q' + CAST([FiscalQuarter] AS NCHAR(1)) AS [FiscalQuarterOfYearLabel],
  [FiscalYear] AS [FiscalYearKey],
  N'FY' + CAST([FiscalYear] AS NCHAR(4)) AS [FiscalYearLabel]
FROM
  [dbo].[DimDate]
WHERE
  [DateKey] >= 20140101;
```

*The query definition retrieves a subset of date values (from 1 January, 2014), and produces columns containing key and name values for each date, for month, and calendar and fiscal periods. These columns will be required to configure the **Date** dimension attributes in the next task.*

6. Review the query result in the results pane.

> **Lab-based Knowledge Check**
> **Lab 03 ► Date Query Review**
>
> How many date rows were returned?
>
> What is the **FiscalQuarterKey** value for the row with a **DateKey** value of **20140101**?
>
> You may need data from this step to answer a Lab-based Knowledge Check associated with this module.
>
> At this time, we recommend that you open the **Module 2 Lab-based Knowledge Check** portion of the course in EdX to answer the questions as you complete this lab.

7. Click **OK**.

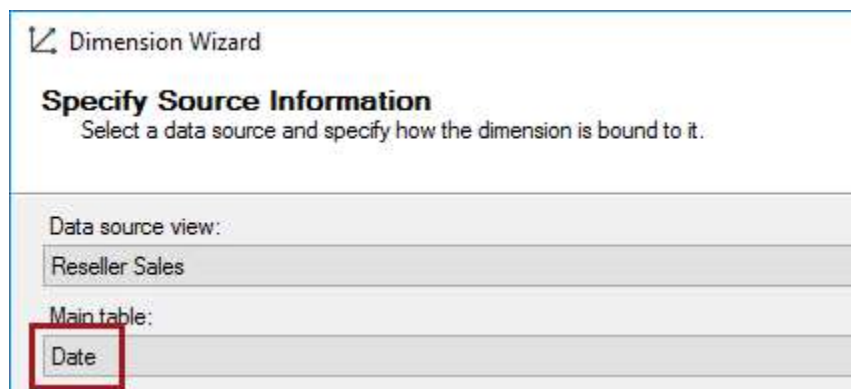8. To save the solution, on the **File** menu, select **Save All**.

Creating the Date Dimension

In this task, you will use the **Dimension Wizard** to create the **Date** dimension.

1.   In **Solution Explorer**, right-click the **Dimensions** folder, and then select **New Dimension**.

2.   In the **Dimension Wizard**, click **Next**.

3.   At the **Select Creation Method** step, notice that the **Use an Existing Table** option is selected.

      This option will base the dimension on a table defined in the data source view. All dimensions created in this lab will use this option.

4.   Click **Next**.
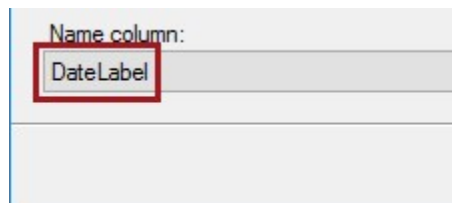
5.   At the **Specify Source Information** step, in the **Main Table** dropdown list, select **Date**.

6.   Notice that the **Key Columns** property has been automatically configured to use the primary (logical) key (**DateKey**) of the main table.

      You will learn about this property later in this exercise.

7.   In the **Name Column** dropdown list, select the **DateLabel** column.

8.   Click **Next**.

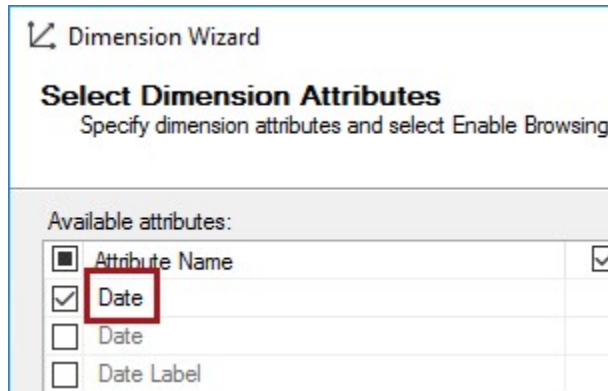9. At the **Select Dimension Attributes** step, in the **Attribute Name** column, replace the **Date Key** text with **Date**.

10. Check the following additional three attributes, and then rename them.

Attribute Name	New Attribute Name
Month Key	Month
Fiscal Quarter Key	Fiscal Quarter
Fiscal Year Key	Fiscal Year

11. Click **Next**.

12. At the **Completing the Wizard** step, in the **Preview** pane, verify that the dimension will consist of four attributes.
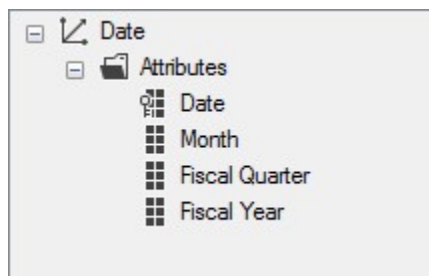
A dimension—by definition—is a collection of attributes. There must be at least one attribute in a dimension, and there must be one key attribute (decorated with the key symbol) which represents the lowest granularity of the dimension.

13. Click **Finish**.

14. In **Solution Explorer**, notice the addition of the **Date** dimension, and that the dimension designer opens automatically.

15. To save the solution, on the **File** menu, select **Save All**.

Configuring the Dimension Attributes

In this task, you will configure the dimension attributes produce appropriately named and ordered hierarchy members.
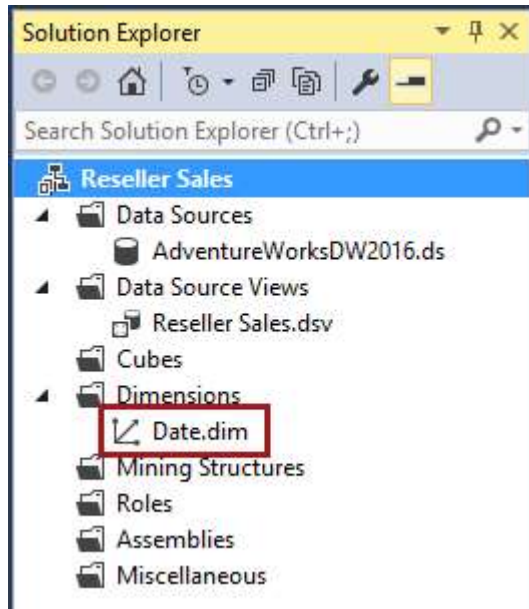
1. In the **Date** dimension designer, in the **Dimension Structure** tab, notice the three panes.

    ▪ The **Attributes** pane displays the dimension (**Date**) and its attributes

    ▪ The **Hierarchies** pane displays optional hierarchies assembled from the dimension attributes. Hierarchies extend beyond the single level of an attribute hierarchy and provide a drill down navigation path for users.

    ▪ The **Data Source View** pane displays the subset of the data source view that the dimension is based on

2. To browse the dimension, in **Solution Explorer**, right-click the **Reseller Sales** project, and then select **Deploy**.

    Every time you create or change an object (dimension or cube), you must deploy the project. The deployment process will send the project definition to the server to create, or modify, the objects. If necessary, objects may be processed. In the dimension designer, you can then browse the updated server object.

3. Notice the **Deployment Progress** window which opens in a small window over the dimension designer.

    *The **Deployment Progress** window will open automatically every time the project is deployed.*

    Tip: You can drag this window to become docked inside any of the panes, or simply close it when deployment has completed.

4. In the dimension designer, select the **Browser** tab.
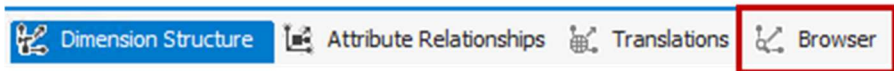
5. In the **Hierarchy** dropdown list, select **Fiscal Year**.

    By default, each dimension attribute produces a hierarchy.

6. In the browser pane, expand the **All** member to reveal all of the **Fiscal Year** hierarchy members.

    *Attribute hierarchies are shallow. By default, an attribute hierarchy consists of an **All** member and a single level beneath. The name of this level has the same name as the attribute.*

7. Notice that the hierarchy member names consist of a year value, however it is not clear whether the name represents a fiscal or calendar year.

8. To provide a name for each hierarchy member, select the **Dimension Structure** tab.

9. In the **Attributes** pane, select the **Fiscal Year** attribute.

    It is important to verify that you have selected the correct object before commencing to configure its properties.

10. In the **Properties** pane (located at the bottom-right), scroll to the bottom of the list, and review the value for the **KeyColumns** property (categorized in the **Source** group).

    *Tip: By default, the **Properties** pane categorizes the properties by functionality, and then orders the properties alphabetically within each category. Most developers find this helpful. You can, however, sort all properties alphabetically by changing the sort preference.*


    *The **KeyColumns** property determines the membership of the attribute, and can consist of multiple columns. At processing time, the membership is defined by the distinct values of the key columns.*

    *For the **Fiscal Year** attribute, the **KeyColumns** property is bound to a single column—the **FiscalYearKey** column from the **Date** table. This explains why the hierarchy consists of each distinct fiscal year value stored in the table.*

11. In the **Properties** pane, select the **NameColumn** property (categorized in the **Source** group).

    *The **NameColumn** allows binding to a different column for the purpose of producing a user-friendly name for each of the hierarchy members. If a **NameColumn** binding is not set, the value of the **KeyColumns** property will be used, providing the **KeyColumns** property is bound to a single column. Multi-column keys must set an explicit name column.*

12. To set the **NameColumn** property, click the ellipsis.
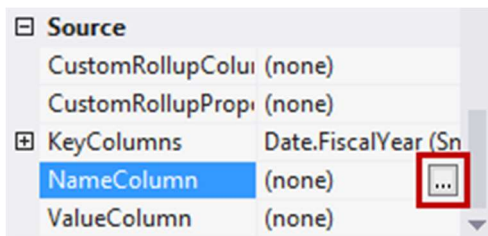
13. In the **Name Column** window, in the **Source Column** list, scroll to the bottom of the list, select the **FiscalYearLabel** column, and then click **OK**.

14. To deploy the project, in **Solution Explorer**, right-click the **Reseller Sales** project, and then select **Deploy**.

15. When the deployment completes, select the **Browser** tab.

16. Notice the warning at the bottom of the browser pane.

17. On the **Dimension** menu, select **Reconnect**.

Whenever you deploy changes, the browser must reconnect to obtain the definition of the updated server object.

*Tip: All commands available on the **Dimension** menu are also available on the toolbar. You can hover over a toolbar button to reveal a tooltip that describes its function.*

18. Notice that the hierarchy members are now prefixed with **FY**.

19. In the hierarchy dropdown list, select **Fiscal Quarter**.

20. Expand the **All** member to reveal the **Fiscal Quarter** hierarchy members.

21. Notice that the hierarchy level consists of each distinct fiscal quarter key value.

22. Set the **Fiscal Quarter** attribute's **NameColumn** property to **FiscalQuarterLabel**, and then redeploy, reconnect and browse the **Fiscal Quarter** hierarchy members.

23. In the hierarchy dropdown list, select **Month**.

24. Expand the **All** member to reveal the **Month** hierarchy members.

25. Notice that the hierarchy level consists of each distinct month key value.

26. Set the **Month** attribute's **NameColumn** property to **MonthLabel**, and then redeploy, reconnect and browse the **Month** hierarchy members.

27. In the hierarchy dropdown list, select **Date**.

28. Expand the **All** member to reveal the **Date** hierarchy members.

29. Notice that the hierarchy level consists of friendly date labels.

*The **NameColumn** property for the **Date** attribute (key attribute) was set by you in the Dimension Wizard.*

30. Notice that the **Day** hierarchy consists of hundreds of members.

*Clearly, the **Date** hierarchy does not provide a user-friendly experience to search and filter by a specific member selection. Also, it does not provide a convenient drill down navigation path (e.g. Fiscal Year > Fiscal Quarter > Month) for users.*

31. Select the **Dimension Structure** tab.

32. In the **Attributes** pane, notice the blue squiggly underlining the **Date** dimension.
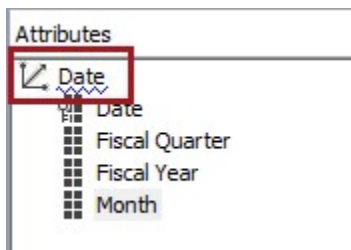
33. Hover over the **Date** dimension to reveal the tooltip.

*The tooltip is a best practice alert that advises that hierarchies should be created in non-parent child dimensions. What it means to suggest is that a multi-level hierarchy should be created.*

Creating a Multi-Level Hierarchy

In this task, you will create the **Fiscal** multi-level hierarchy to allow a drill down path for the fiscal periods.

1. To create a multi-level hierarchy, from the **Attributes** pane, drag the **Fiscal Year** attribute into the **Hierarchies** pane.
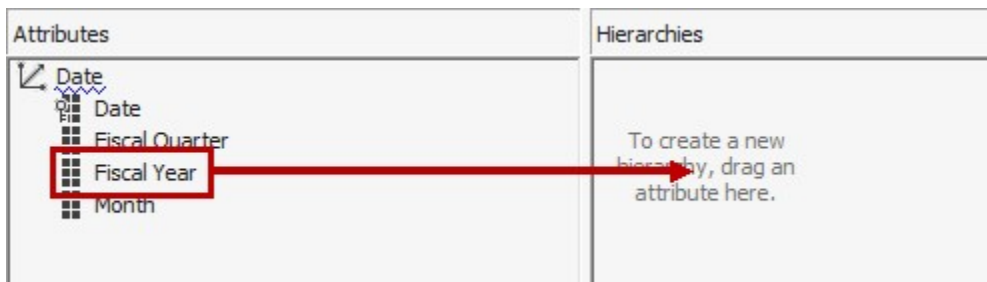
2. Select the header of the hierarchy, and then in the **Properties** pane, modify the **Name** property to **Fiscal**.

3. To add a new level to the hierarchy, drag the **Fiscal Quarter** attribute to the **<new level>** of the hierarchy.

4. Repeat the last step to add the **Month** attribute, and then the **Date** attribute, to the hierarchy.

5. To rename the hierarchy level, select the **Fiscal Year** <u>level of the hierarchy</u> (do not select the attribute), and then in the **Properties** pane, modify the **Name** property to **Year**.

6. Repeat the last step to modify the name of the **Fiscal Quarter** level to **Quarter**.

7. Verify that the hierarchy consists of four levels: **Year**, **Quarter**, **Month** and **Date**.
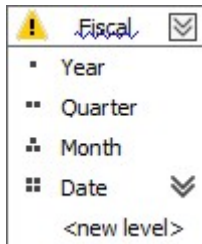
8. Deploy the project, then browse and reconnect, and then review the **Fiscal** hierarchy members.

    *In the **Hierarchy** dropdown list, the triangular icon (⁂) represents a multi-level hierarchy.*

9. In the browser pane, expand **All > FY2014 > FY2014 Q3**.
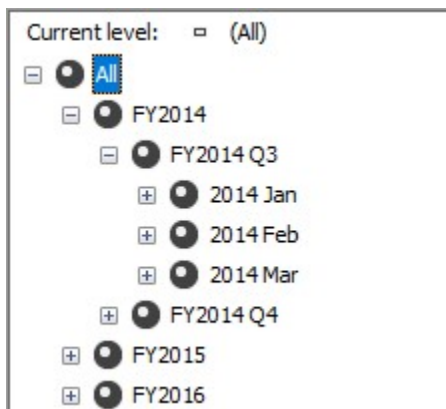
10. Select a member from each level of the hierarchy, and notice the name of the hierarchy appears at the top of the browser pane.

11. Select the **Dimension Structure** tab.

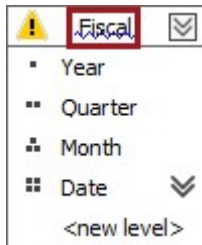12. In the **Hierarchies** pane, notice the hazard icon in the header of the **Fiscal** hierarchy, and the blue squiggly line underlining the hierarchy name.

13. Hover over the blue squiggly line to reveal a tooltip.

    *The best practice alert advises that attribute relationships do not exist between the levels of the hierarchy. Presently relationships exist between each of the attributes and the **Date** (key) attribute. This is both inaccurate and inefficient. You will configure optimized attribute relationships in the next task.*

Optimizing the Dimension Design

In this task, you will optimize the dimension by defining attribute relationships, by configuring appropriate attribute relationship types, and by hiding attributes used in the levels of the multi-level hierarchy.

1. To configure attribute relationships, select the **Attribute Relationships** tab.

2. To configure the correct relationship, drag the **Month** attribute on top of the **Fiscal Quarter** attribute.
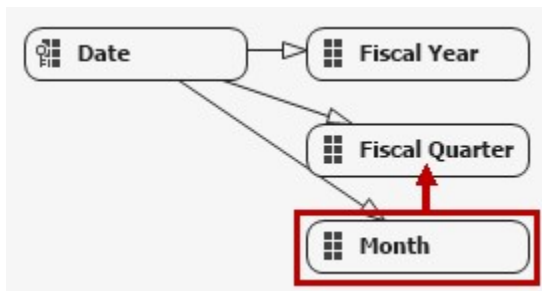
3. Drag the **Fiscal Quarter** attribute on top of the **Fiscal Year** attribute.

4. Verify that the attribute relationships form a chain, from **Date** to **Month**, to **Fiscal Quarter**, to **Fiscal Year**.
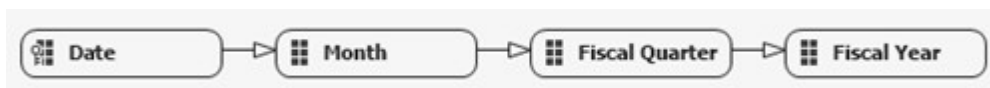
5. Select the **Dimension Structure** tab.

6. Notice that the hazard icon and the blue squiggly have disappeared from the multi-level hierarchy.

7. In the **Attributes** pane, hover over the blue squiggly underlining the **Date** dimension.

    The tooltip advises two alerts. The first is to avoid visible attributes if they are used as levels in a user-defined hierarchy (also known as a multi-level hierarchy). The second is to define rigid attribute relationships.

8. To address the second alert first, select the **Attribute Relationships** tab.
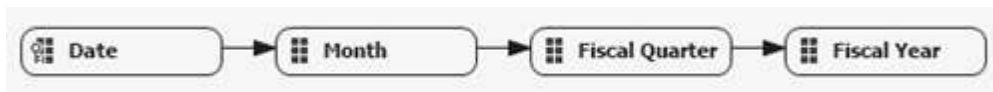
9. Right-click each of the three attribute relationships (represented as arrows), and then select **Relationship Type > Rigid**.

10. Verify that each attribute relationship is represented by a solid black arrow head.

    *Rigid relationships can be configured for relationships which you know at design time will never change—and this is certainly the case for the **Date** dimension. You would never expect the **FY2014 Q3** member to relocate beneath the **FY2015** member!*

    Defining rigid relationships can allow certain optimizations that take place at processing time. But take care! If a rigid relationship is configured and the dimension data changes, the dimension would need to be fully processed to allow the change. It is important to understand that a fully processed dimension invalidates all cube data that references it. All dependent cube measure groups would then need to be reprocessed.

11. Select the **Dimension Structure** tab.

12. To address the remaining alert, in the **Attributes** pane, select the **Date** attribute (not dimension), and then while pressing the **Control** key, select the **Month** attribute also.
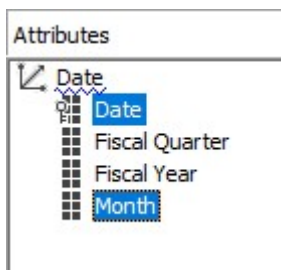
13. In the **Properties** pane, set the **AttributeHierarchyVisible** property (categorized in the **Advanced** group) to **False**.

    Hiding these attributes makes sense most of the time. Rather than having a user expand the ***Date*** *attribute to reveal all dates (in this case 1,461 members), they could access the members in a more controlled way by drilling down to a particular month first. This can save significant time and server and client resources when users browse a cube.*

    The ***Fiscal Year*** *and* ***Fiscal Quarter*** *attributes have been deliberately left visible so that they can be used to provide available values to Reporting Services report parameters.*

14. Deploy the project, then browse and reconnect.

15. In the **Hierarchy** dropdown list, notice that there are now only three hierarchies.

16. Select the **Dimension Structure** tab.

17. Notice that a blue squiggly continues to underline the **Date** dimension.

    Best practice alerts are only suggestions. They can be dismissed if necessary.

Dismissing Best Practice Alerts

In this task, you will dismiss an alert for the dimension, and then for the entire database.

1. To dismiss the alert, on the **View** menu, select **Error List**.

2. In the **Error List**, right-click the warning, and then select **Dismiss**.

3. In the **Dismiss Warning** window, in the **Comments** box, enter **Attributes can remain visible for SSRS reports**.

    The comments allow the developer to document the reason for dismissing the alert.

4. Click **OK**.

5. To close the **Error List**, click the **X** located in the top right corner of the list.

6. Notice that the blue squiggly underlining the **Date** dimension has disappeared.

7. To dismiss alerts for the entire project, on the **Database** menu, select **Edit Database**.

8. Select the **Warnings** tab.

9. Expand the **Dimension Design** group.

10. Scroll down the list to locate the alert **Avoid Visible Attribute Hierarchies for Attributes Used as Levels in User-Defined Hierarchies** (the second warning in the **Low** importance group).

11. Uncheck the warning, and in the corresponding **Comment** box, enter **Attributes can remain visible for SSRS reports**.

12. To revert the warning already dismissed, in the **Dismissed Warnings** list, select the warning, and then click **Re-enable** (located at the bottom-left).

13. To save the database properties, on the **File** menu, select **Save All**.

14. To close the database properties, on the **File** menu, select **Close**.

Configuring Additional Properties

In this task, you will use the **Business Intelligence Wizard** to configure time properties. You will also set the **ValueColumn** property of the **Day** attribute.

*An important development task for the **Date** dimension is to define the dimension and attribute types. This metadata is required to distinguish the dimension and its attributes, and is referred to by certain time features, for example semi-additive cube measures, and MDX time intelligence functions.*

1. To bulk configure attribute **Type** properties, in **Solution Explorer**, right-click the **Date.dim** dimension, and then select **Add Business Intelligence**.

2. In the **Business Intelligence Wizard**, click **Next**.

3. At the **Choose Enhancement** step, select **Define Dimensions Intelligence**.
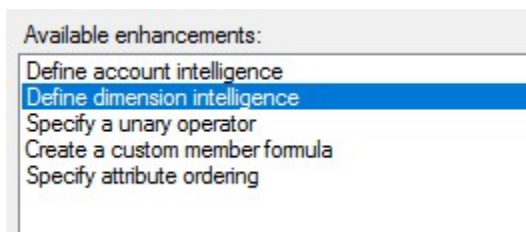
4. Click **Next**.

5. At the **Define Dimension Intelligence** step, in the **Dimension Type** dropdown list, select **Time**.

6. Select to include the **Year** type, and in the corresponding **Dimension Attribute** dropdown list, select **Fiscal Year**.

7. Configure the remaining three dimension attributes as follows.

8. Click **Next**.

9. At the **Completing the Wizard** step, review the changes.

   *Each change represents the setting of the dimension or attribute **Type** property. The **Business Intelligence Wizard** is a convenient and easy way to update many properties in a single operation.*

10. Click **Finish**.

11. In the **Attributes** pane, select the **Date** attribute.

12. In the **Properties** pane, review the **Type** property (categorized in the **Basic** group) which is now set to **Days**.

13. In the **Properties** pane, set the **ValueColumn** property (categorized in the **Source** group) to bind to the **Date** column (second column in the list).

   *The **Date** column contains date/time values. This configuration will improve the Excel PivotTable experience by allowing users to define date-related filters by using a calendar control. It will also allow the creation of timelines (a capability added in Excel 2013) to allow visual filtering by contiguous calendar time periods.*

14. To save the solution, on the **File** menu, select **Save All**.

15. To close the **Date** dimension designer, on the **File** menu, select **Close**.

Exercise 2: Developing the Product Dimension

In this exercise, you will develop the **Product** dimension. This dimension will be based on the three product dimension tables: **DimProduct**, **DimProductSubcategory** and **DimProductCategory**.

Creating the Product Dimension

In this task, you will use the **Dimension Wizard** to create the **Product** dimension.

1. In **Solution Explorer**, right-click the **Dimensions** folder, and then select **New Dimension**.

2. In the **Dimension Wizard**, click **Next**.

3. At the **Select Creation Method** step, click **Next**.

4. At the **Specify Source Information** step, in the **Main Table** dropdown list, select **Product**.

5. In the **Name Column** dropdown list, select **EnglishProductName**.

    *This conveniently configures the **NameColumn** property for the key attribute.*

6. Click **Next**.

7. At the **Select Related Tables** step, notice that the two related tables are selected.

    A collection of normalized dimension tables is described as a snowflake dimension. Columns from all three tables can be used to create attributes for a single Analysis Services dimension.

8. Click **Next**.

9. At the **Select Dimension Attributes** step, notice that the **Product Key**, **Product Subcategory Key** and **Product Category Key** attributes are automatically selected (you will need to scroll down the list to see all selected columns).

    *The wizard suggests that each logical key column should be a dimension attribute, and each would be created with its **KeyColumns** property based on the source column. Note that the first attribute, based on the logical key of the **Product** table, is the key attribute, and was configured at the **Specify Source Information** step of the wizard.*

10. Check the **Color** attribute.

11. Rename the following attributes.

Attribute Name	Rename Attribute To
Product Key	Product
Product Subcategory Key	Subcategory
Product Category Key	Category

*This step of the wizard does not support the configuration of a **NameColumn** property. Therefore, if necessary, the attribute name columns must be set in the dimension designer.*

*You will need to configure the **NameColumn** properties of the **Subcategory** and **Category** attributes, as they are numeric key values (as were fiscal year, fiscal quarter and month).*

12. Click **Next**.

13. At the **Completing the Wizard** step, in the **Preview** pane, verify that the dimension will consist of four attributes.
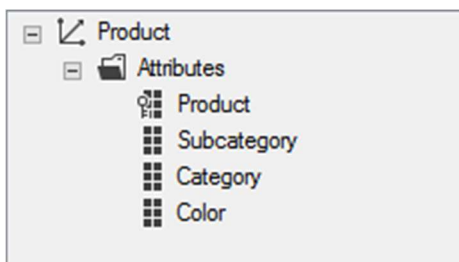
14. Click **Finish**.

15. To save the solution, on the **File** menu, select **Save All**.

Configuring the Attribute NameColumn Properties

In this task, you will configure the **NameColumn** property for two attributes.

1. In the dimension designer, in the **Attributes** pane, select the **Category** attribute.

2. In the **Properties** pane, bind the **NameColumn** property to the **EnglishProductCategoryName** column.

3. Repeat the last two steps to bind the **NameColumn** property for the **Subcategory** attribute to the **EnglishProductSubcategoryName** column.

Optimizing the Dimension Design

In this task, you will optimize the dimension design by adding a multi-level hierarchy, and by configuring the attribute relationship types. You will also explore the concept of the unknown member.

1. Create a multi-level hierarchy named **Products** with three levels: **Category**, **Subcategory** and **Product**.

    *Unlike with the **Fiscal** hierarchy created in the **Date** dimension, there is no alert advising that attribute relationships need to be created.*

2. Select the **Attribute Relationships** tab.

3. Notice that the attribute relationships have already been correctly configured.

    *The **Dimension Wizard** can automatically configure the correct relationships for snowflake dimensions, as the related tables represents a clear chain of one-to-many relationships.*

4. Configure both attribute relationship types as rigid.

    *Recall that the **DimProduct** table supports Type 2 Slowly Changing Dimensions. At Adventure Works, when a product characteristic changes, it is either assigned a new stock-keeping unit (**ProductID**) or is implemented as a new Type 2 version. Either way, a new product record is added to the table. Therefore, it is safe to configure the relationship type between the **Product** and **Subcategory** attributes as rigid.*

5. Verify that the attribute relationships form a chain, from **Product** to **Subcategory**, to **Category**.
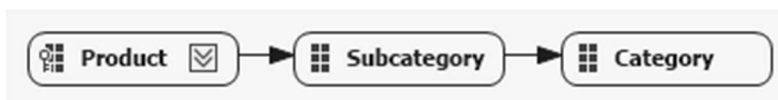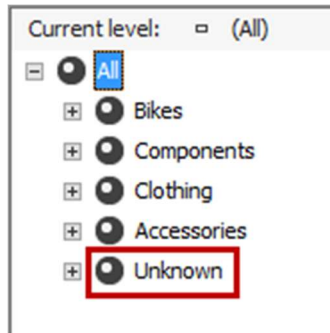

6. Select the **Dimension Structure** tab.

7. Hide the **Product** and **Subcategory** attribute hierarchies.

8. Deploy the project, and then browse the dimension.

    *If you are prompted to overwrite the project, while in development, you should click **OK**.*

---

9.   In the **Hierarchy** dropdown list, verify that there are only three hierarchies.

Hierarchy: ⁙ Products

    ⁂ Products
    ⁙ Category
    ⁙ Color
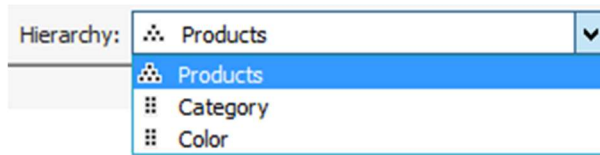
10. Browse the membership of the **Products** hierarchy.

11. Notice that the hierarchy **Category** level includes an **Unknown** member.

Current level: ⬚   (All)

⊟ ⬤ All
    ⊞ ⬤ Bikes
    ⊞ ⬤ Components
    ⊞ ⬤ Clothing
    ⊞ ⬤ Accessories
    ⊞ ⬤ Unknown

*Unknown members do not come from the tables in the source database. The **Unknown** member is a dimension feature that compensates for a possible lack of referential integrity between the dimension tables or between the dimension and the cube. This way, the b member represents any values that cannot be assigned to a member in the attribute hierarchy.*

*This member can be added by configuring the **UnknownMember** property of the dimension. The **Dimension Wizard** automatically configures this property to **Visible** for snowflake dimensions (i.e. dimensions based on more than one table) or dimensions that include a parent-child hierarchy.*

12. Expand **All > Unknown > Unknown**.

*In the **AdventureWorksDW2016** database, the **DimProduct** table contains products that have not been assigned a **ProductSubcategoryKey** value. These products are not finished goods that are sold; rather, they are components that are used in finished goods. You will filter out these products so that you can disable the unknown member.*

13. Switch to the **Reseller Sales** data source view designer.

14. In the data source view designer, in the **Tables** pane, right-click the **Product** table, and then select **Replace Table | With New Named Query**.

15. In the **Create Named Query** window, in the **Name** box, replace the text with **Product**.

16. To hide the **Diagram** pane, click the following toolbar button.

17. In the grid pane, locate the **FinishedGoodsFlag** column, then in the corresponding **Filter** box, enter **1**, and then press **Enter**.

18. In the SQL pane, notice the addition of the WHERE clause.

19. Click **OK**.

20. Switch to the **Product** dimension designer.

21. Select the **Dimension Structure** tab.

22. In the **Attributes** pane, select the **Product** dimension (not the **Product** attribute).
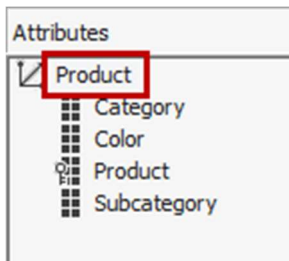
23. In the **Properties** pane, set the **UnknownMember** property (categorized in the **Advanced** group) to **None**.

24. In the **Attributes** pane, notice that the **Category** and **Subcategory** attributes are underlined with a red squiggly.

    Unlike a blue squiggly that indicate a best practice alert, a red squiggly indicates an error. The project cannot be built, and therefore cannot be deployed, while errors exist.

    *In this instance, the errors represent incongruous settings for the **NullProcessing** property of two attributes, which is set to assign a member with a null key to use the unknown member.*

25. To address the design errors, select the **Category** attribute.

26. In the **Properties** pane, expand the **KeyColumns** property.

27. Expand the **ProductCategory.ProductCategoryKey** column, and then update the **NullProcessing** property to **Automatic**.

28. Repeat the last two steps to similarly configure the **Subcategory** attribute.

29. Deploy the project, browse and then reconnect.

30. In the dimension browser, in the **Products** hierarchy, notice that the unknown members are no longer in the hierarchy.

31. To save the solution, on the **File** menu, select **Save All**.

32. Close the **Product** dimension designer.

Exercise 3: Developing the Reseller Dimension

In this exercise, you will develop the **Reseller** dimension. This dimension will be based on the **DimReseller** and **DimGeography** tables.

## Developing the Reseller Dimension

In this task, you will create and optimize the **Reseller** dimension.

1. Use the **Dimension Wizard** to create a dimension developing the following requirements.

    -   The main table is **Reseller**

    -   The name column of the key attribute is the **ResellerName** column

    -   The dimension attributes must also be based on the **Geography** table

    -   The dimension attributes must include six attributes: **Reseller** (based on the **ResellerKey** column), **Phone**, **Business Type**, **City**, **State** (based on the **StateProvinceName** column), and **Country** (based on the **EnglishCountryRegionName** column)—do not include an attribute based on the **GeographyKey** column

    -   The name of the dimension must be **Reseller**

2. Before finishing the **Dimension Wizard**, in the **Preview** pane, verify that the design consists of six attributes.

    *If the dimension preview does not contain six attributes, click **Back** to adjust the wizard configuration.*

---
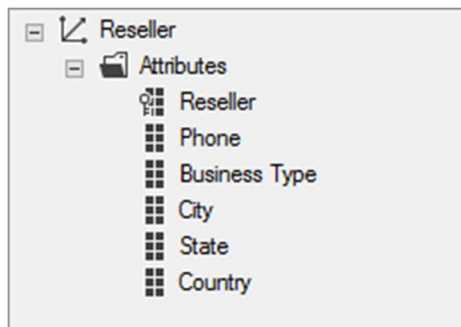
3. Use the dimension designer to configure the following additional requirements.

    ▪   A hierarchy named **Resellers** to navigate **Business Type > Reseller**

    ▪   A second hierarchy named **Geography** to navigate
        Country > State > City > Reseller

    ▪   Two attribute relationship paths **Reseller > Business Type**, and
        Reseller > City > State > Country

    ▪   Rigid attribute relationships types between the **City** and **State** attributes, and
        between the **State** and **Country** attributes (do <u>not</u> configure rigid relationships for
        the **Reseller** and **City** attributes, or the **Reseller** and **Business Type** attributes—it is
        possible that resellers could relocate cities, or be reclassified to a different business
        type)

    ▪   The **City**, **State** and **Reseller** attribute hierarchies must be hidden

4. Verify that the attribute relationships form two chains, from **Reseller** to **City**, to **State**, to
    Country, and also from **Reseller** to **Business Type**.
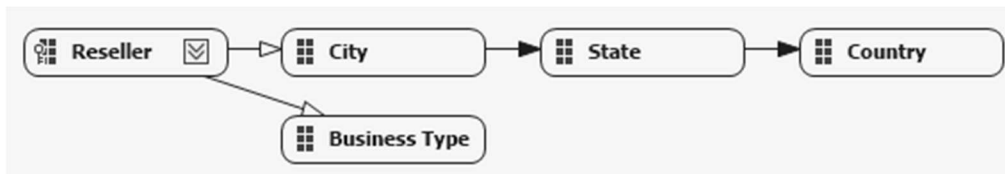

5. Deploy the project.

    The dimension process will fail, due to a data discrepancy.

6. In the **Error List**, review the error description.

    *At processing time, the storage engine has encountered a duplicate value in the **City**
    attribute. According to the error description, the city name is **Augsburg**, and a closer
    inspection of the data in the **DimGeography** table reveals that this city name does in fact
    exist twice in **Germany**, but in different states (provinces). According to the attribute
    relationships configured for this dimension, however, a city must be assigned to only one
   state.*

    *To address this problem, you will modify the **KeyColumns** property for the **City** attribute to
    be sourced from both the **State** and the **City** attributes (recall that attribute members are
    based on the distinct combination of the key column values). This will then ensure a member
    is created for each city of Augsburg.*

7. In the **Dimension Structure** tab, in the **Attributes** pane, select the **City** attribute.

8. In the **Properties** pane, update the **KeyColumns** property based on the **City** and **StateProvinceCode** attributes.
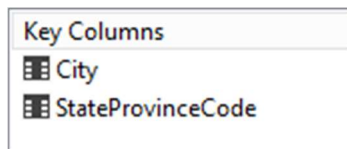
9. Click **OK**.

10. In the **Attributes** pane, notice that the **City** attribute is underlined with a red squiggly.

11. Hover over the **City** attribute to reveal the tooltip error message.

    Attributes with multiple key columns must define an explicit name column.

12. Configure the **NameColumn** property of the **City** attribute to bind to the **City** column.

13. Close the **Error List**.

14. Deploy the project, and then browse the dimension.

15. Verify that there are five hierarchies.
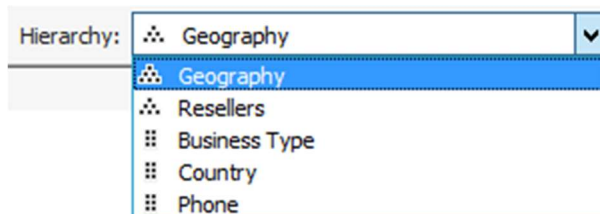
16. Browse the membership of the five hierarchies.

17. Notice that the members of the **Phone** hierarchy consist of telephone numbers.

    *It is an unusual requirement to include a hierarchy of telephone numbers as it would be surprising that users would need to filter cube data by a telephone number, and more unusual that they would need to aggregate a measure by a phone number. It is, however, understandable that users would want to retrieve telephone numbers when reporting on resellers. Recall that you added the salesperson's phone number as a property to the PivotTable report in **Lab 01**.*

    This requirement can be implemented in the dimension as a member property. A member property is implemented by disabling the attribute hierarchy.

18. Select the **Dimension Structure** tab.

19. In the **Attributes** pane, select the **Phone** attribute.

20. In the **Properties** pane, set the **AttributeHierarchyEnabled** property (categorized in the **Advanced** group) to **False**.

21. In the **Attributes** pane, notice that the icon decorating to the **Phone** attribute has changed to indicate a disabled attribute.
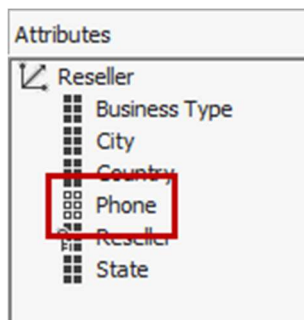
22. Deploy the project, browse and then reconnect.

23. In the **Hierarchy** dropdown list, notice that the **Phone** hierarchy is no longer listed.

24. Select the **Resellers** hierarchy.

25. Right-click the **All** member, and then select **Expand All**.

26. To display member properties, on the **Dimension** menu, select **Member Properties**.

27. In the member properties list, check **Phone**, and then click **OK**.

28. Notice that the member property values are available for reporting.
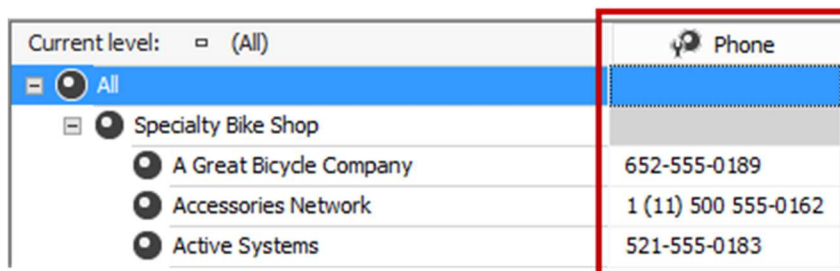
29. To save the solution, on the **File** menu, select **Save All**.

30. Close the **Reseller** dimension designer.

Exercise 4: Developing the Salesperson Dimension

In this exercise, you will develop the **Salesperson** dimension. This dimension will be based on the **DimEmployee** table, and will include a parent-child hierarchy to navigate the sales organization structure defined by the table's self-referencing relationship.

Developing the Salesperson Dimension

In this task, you will create and optimize the **Salesperson** dimension.

1. Use the **Dimension Wizard** to create a dimension developing the following requirements.

    ▪ The main table is **Salesperson**

    ▪ The name column of the key attribute is the **Salesperson_Name** column

    ▪ A dimension attribute must be created for every column in the **Salesperson** table

    ▪ The attribute based on the **EmployeeKey** column must be named **Salesperson**, and the attribute based on the **ParentEmployeeKey** must be named **Sales Organization**

    ▪ The attributes based on the **LoginID**, **EmailAddress** and **Phone** columns must have their hierarchies disabled

    *Tip: The **AttributeHierarchyEnabled** property can be conveniently set to **False** in the **Dimension Wizard** by deselecting the **Enable Browsing** checkbox.*

    ▪ The dimension must be named **Salesperson**

2. Before finishing the **Dimension Wizard**, in the **Preview** pane, verify that the design consists of eight attributes.

    *If the dimension preview does not contain eight attributes, click **Back** to fix the wizard configuration.*

    Note that the preview pane has a bug. The attributes configured to not enable browsing should be decorated with a slightly different icon.

---
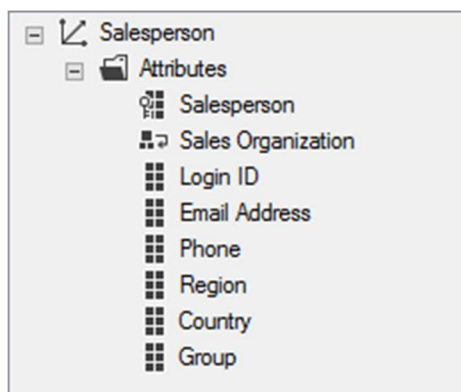
*Notice the icon used for the **Sales Organization** attribute which represents a parent-child hierarchy. The **Dimension Wizard** detected the self-referencing relationship defined in the data source view and has configured that attribute as a parent attribute.*

A parent attribute delivers a parent-child hierarchy. A maximum of one parent-hierarchy can be defined in a dimension.

3. Use the dimension designer to implement the following requirements.

    ▪ A hierarchy named **Salespeople** to navigate
       Group > Country > Region > Salesperson

    ▪ One attribute relationship chain: **Salesperson > Region > Country > Group**

    ▪ Rigid attribute relationships types can be configured between all attributes

    ▪ The **Country**, **Region** and **Salesperson** attributes must be hidden (the **Salesperson** attribute will already be hidden because the **Dimension Wizard** hides the key attribute when a parent-child hierarchy is defined)

4. Verify that the attribute relationship form a chain, from **Salesperson** to **Region**, to **Country**, to **Group**.

*Recall that the **DimEmployee** table supports Type 2 Slowly Changing Dimensions. At Adventure Works, when a salesperson changes sales territory a new version is created. Therefore, it is safe to configure the relationship type between the **Salesperson** and **Region** attributes as rigid.*

5. Deploy the project, and then browse the dimension.

6. Verify that there are three hierarchies.
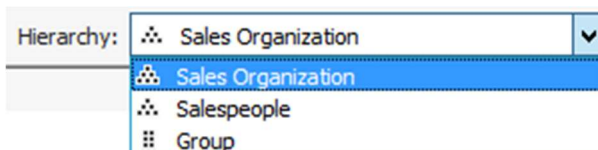
7. Browse the membership of the **Salespeople** hierarchy first.

8. Right-click the **All** member, and then select **Expand All**.

9. Review the **Email Address**, **Login ID** and **Phone** member properties for the salespeople.

10. Notice that the last member on the **Group** level is the **Unknown** member.

    *The **Dimension Wizard** automatically configures the unknown member for dimensions with a parent-child hierarchy.*

    *There is, however, integrity between all records involved in the self-referencing relationship on the **DimEmployee** table. The unknown member for the dimension can be turned off.*

11. Select the **Dimension Structure** tab.

12. In the **Attributes** pane, select the **Salesperson** dimension (<u>not</u> the **Salesperson** attribute).

13. In the **Properties** pane, set the **UnknownMember** property (categorized in the **Advanced** group) to **None**.

14. In the **Attributes** pane, select the **Salesperson** attribute.

15. In the **Properties** pane, expand the **KeyColumns** property.

16. Expand the **Salesperson.EmployeeKey** column, and then set the **NullProcessing** property to **Automatic**.

17. Deploy the project, reconnect and then browse the dimension.

18. In the dimension browser, notice that the unknown member is no longer included in the hierarchy.

19. In the **Hierarchy** dropdown list, select **Sales Organization**.

20. Right-click the **All** member, and then select **Expand All**.

21. Notice that there are repeat members in the hierarchy.
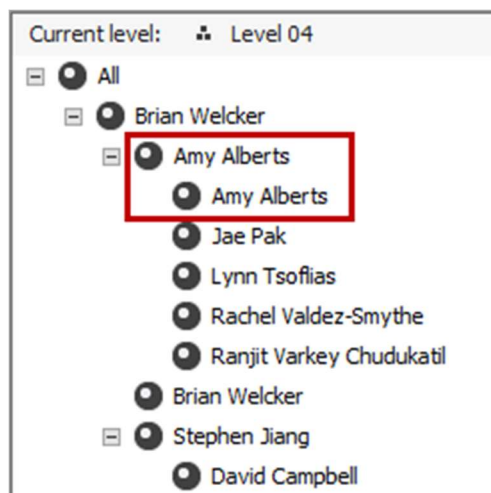
22. Select members from different levels of the hierarchy, and notice the name of the hierarchy level at the top of the browser pane.

*An interesting characteristic of a parent-child hierarchy is that all members are members from the key attribute (in this case, the **Salesperson** attribute). As such, data can be attached to any member, and at any level, of the hierarchy (e.g. any salesperson can sell products, not just those at the lowest level of the hierarchy). This is reason why there are repeat members—the lowest member is there to allow the aggregation of its values into itself. Contrast this with an attribute hierarchy or multi-level hierarchy. Data is attached at the lowest level and aggregated up the hierarchy.*

*Another interesting characteristic is that the number of levels in a parent-child hierarchy is determined by the relationships between the records in the table. As such, the level names are generic and sequential (e.g. **Level 02**, **Level 03** etc.). Contrast this with an attribute hierarchy or multi-level hierarchy where the levels are fixed at design time and can be explicitly named.*

There are two optimizations to the design of a parent-child hierarchy that will make them more user-friendly.

Configuring the Parent-Child Hierarchy

In this task, you will configure the **Sales Organization** hierarchy.

1. Select the **Dimension Structure** tab.

2. In the **Attributes** pane, select the **Sales Organization** attribute.

3. In the **Properties** pane, set the **MembersWithDataCaption** property (categorized in the **Parent-Child** group) to **(* data)**—include the parentheses.

   *The **MembersWithDataCaption** property is used to name the repeated member in the hierarchy. The asterisk symbol (*) represents the member name, and is substituted into the member caption. For example, based on the configured template, the member **Brian Welcker** will be expressed as **(Brian Welker data)**.*

4. In the **Properties** pane, select the **NamingTemplate** property (categorized in the **Parent-Child** group), and then click the corresponding ellipsis.

5. In the **Level Naming Template** window, in the ***** level row, click inside the **Name** box, and then enter **Director**.

6.	Configure the following additional level names.

Level	Name
3	Manager
4	Representative

Level	Name
1	(All)
2	Director
3	Manager
4	Representative
*	

7.	Click **OK**.

8.	Deploy the project, reconnect and then browse the dimension.

9.	Browse the **Sales Organization** hierarchy again and review the hierarchy level names and member names.
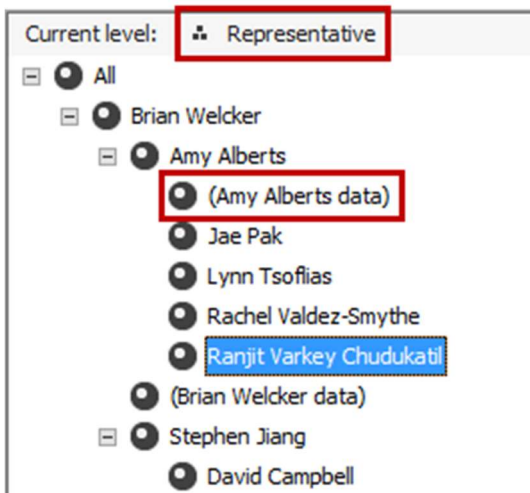
10.	To save the solution, on the **File** menu, select **Save All**.

11.	Close the **Salesperson** dimension designer.

Exercise 5: Developing the Sales Order Dimension

In this exercise, you will develop the **Sales Order** dimension. This dimension will be based on the **FactResellerSales** table. Dimensions based on fact tables are referred to as fact—or degenerate—dimensions. The dimension will be used to support a drillthrough action that you will create in **Lab 04**.

Preparing the Data Source View

In this task, you will prepare the data source view by creating a named calculation.

1. In the **Reseller Sales** data source view designer, in the **Sales** table, create a named calculation named **SalesOrderLineNumber_Name** by using the following expression.

    *You can copy the expressions from the **F:\Labs\Lab03\Assets\Snippets.txt** file.*

    T-SQL
    ```
    [SalesOrderNumber] + N'-' + FORMAT([SalesOrderLineNumber], N'00')
    ```

    *The logic in this expression ensures that the **SalesOrderLineNumber** value is formatted to two characters, padding with a leading zero if necessary.*

2. In the diagram view, in the **Sales** table, notice that the primary (logical) keys are the **SalesOrderNumber** and the **SalesOrderLineNumber** columns (the first two columns).

3. To save the solution, on the **File** menu, select **Save All**.

Developing the Sales Order Dimension

In this task, you will develop the **Sales Order** dimension.

1. Use the **Dimension Wizard** to create a dimension implementing the following requirements.

    - The main table is **Sales**

    - Only one attribute must be created (which must be the key attribute). The key columns should be sourced from the two **Sales** table primary key columns, and the name column should be sourced the **SalesOrderLineNumber_Name** column.

    - No related tables should be added to the dimension

    - The name of the key attribute must be **Sales Order Line**

    - Ensure that no other attributes are checked

    *Recall that the **Dimension Wizard** always automatically selects attributes where **Key** is part of the name of the column.*

    Hint: Scroll to the bottom of the available attributes list and deselect all "key" attributes that have been automatically selected.

2. The dimension must be named **Sales Order**

3. Before finishing the **Dimension Wizard**, in the **Preview** pane, verify that the design consists of a single attribute.

    *If the dimension preview does not consist of a single attribute, click **Back** to fix the wizard configuration.*
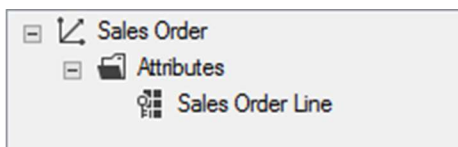
4. Deploy the project.

5. Browse the membership of **Sales Order Line** hierarchy.

    *As the **KeyColumns** property of the **Sales Order Line** attribute is based on the two logical keys of the **Sales** table, the number of members in the hierarchy will always be the same as the number of rows in the table, plus one, for the **All** member. This is not a practical hierarchy to allow users to browse, yet it is needed to meet the requirements of the drillthrough action that you will create in **Lab 04**.*

    *When you create the cube in **Lab 04**, you will hide the cube dimension so that users cannot browse by it.*

6. To save the solution, on the **File** menu, select **Save All**.

7. Close the **Sales Order** dimension designer.

8. Close the **Reseller Sales** data source view designer.

    You have now completed the lab. In the next lab, you will create the cube.

    *If you are not immediately continuing with the next lab, you should complete the **Finishing Up** exercise to shut down and stop the VM.*
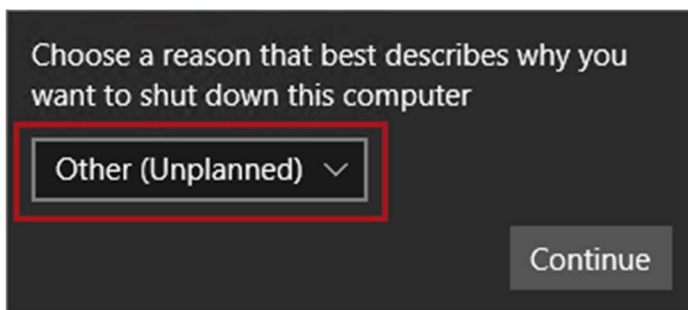
Finishing Up

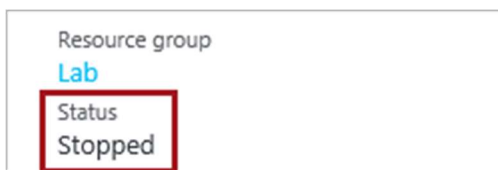In this exercise, you will shut down and stop the VM.

1. Close all open applications.

2. Press the **Windows** key, and then in the **Start** page, located at the bottom-left, click the **Power** button, and then select **Shut Down**.
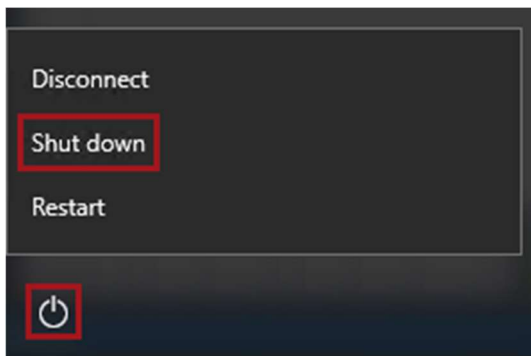
3. When prompted to choose a reason, to accept the default.

4. Click **Continue**.

5. In the **Azure Portal** Web browser page, wait until the status of the VM updates to **Stopped**.

*In this state, however, the VM is still billable.*

6. Optionally, to deallocate the VM, click **Stop**.

    Deallocation will take some minutes to complete, and also extends the time required to restart the VM. Consider deallocating the VM if you want to reduce costs, or if you choose to complete the next lab after an extended period.

    The deallocation can take several minutes to complete.

7. Verify that the VM status updates to **Stopped (Deallocated)**.
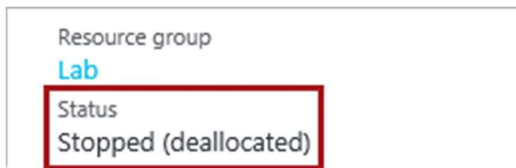
    In this state, the VM is now not billable—except for a relatively smaller storage cost.

    Note that a deallocated VM will likely acquire a different IP address the next time it is started.

8. Sign out of the **Azure Portal**.