Assignment 09: Business Intelligence

Part 1: Overview

This purpose of this assignment is to introduce various types of business intelligence tools and some of the common ways you can build out data visualizations such as dashboards, interactive reports and charts from data warehouse data. We'll start with MOLAP development using SQL Server Analysis Services, or SSAS. This will help you understand how the semantic model, where we add additional meaning to our data, helps support business intelligence. After you build your MOLAP cube, you'll visualize and explore the data with a variety of business intelligence tools to create interactive reports, charts, maps, and dashboards.

As such, it should be noted that this lab does not demonstrate everything you need to know about the MOLAP process, nor does it serve as training tool for SSAS, Excel, Power BI, etc. These activities are beyond the scope of this course. Yes, when you're finished with the lab, you'll know your way around these tools, but more importantly you'll understand the purpose and various capabilities of MOLAP databases, the semantic model, and how it connects to data visualization tooling. Over time the tools change, but the concepts stay the same!

Goals

Specifically the goals of this assignment are to:

- Learn how to build a Multidimensional Online Analytical Processing (MOLAP) Database
- Use Excel and Power BI to connect to data warehouse data and build business intelligence
- Realize the benefits the semantic model provides

Effort

This assignment is best done individually. Please work alone.

Technical Requirements

To complete this assignment you will need the following:

- Access to your **ist722_*yournetid*_dw** database on SQL Server.
- Access to the SQL Server 2012 Data Tools (used to create SSAS database objects).
- Microsoft Excel 2013 or higher.
- Microsoft Power BI Desktop.
- You should connect to your computer running your SQL server database before starting this lab.

Understanding Analysis Services

Microsoft SQL Server Analysis Services allows us to build multidimensional OLAP solutions. We can deploy these solutions to a special database, which users can connect to and perform ad-hoc interactive reporting. The OLAP database differs from a traditional database in that it supports:

- Rich metadata in our dimensions—for example, we can sort by month number but display the month's name so that the ordering of months is correct: Jan, Feb, Mar, etc.
- Fixed and ragged hierarchies.
- Detection of measures in our fact tables and their configuration as additive or semi-additive.
- Perspectives for limiting views into the OLAP database.
- Row-level security over our dimensional model.
- Aggregations—pre-calculations for performance improvements.
- Calculations, KPI's, data mining and more!

Analysis services consist of two components:

1) **SQL Server Data Tools—**We use this to build our analysis services databases.
2) **SQL Server Analysis Services**—We deploy our analysis services databases to this server to test and deploy our packages

Terminology

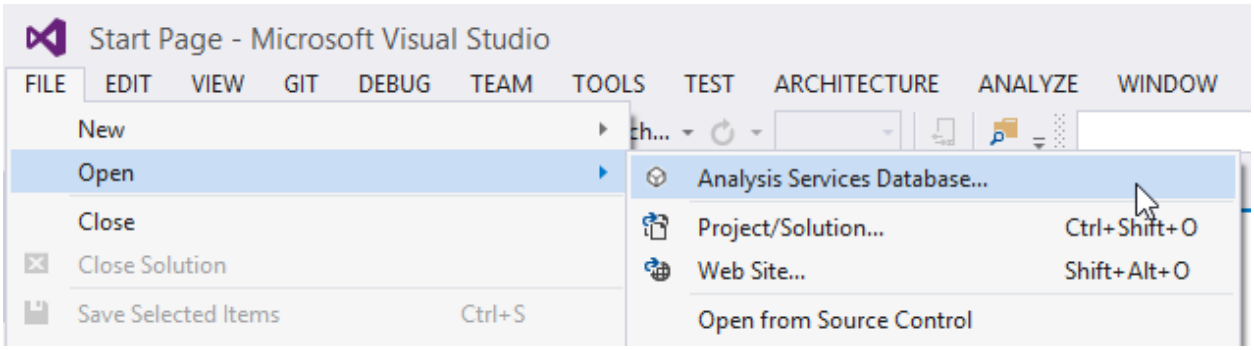
- **Data source**—Connections to relational databases and other sources of data.
- **Data Source View**—The logical model of the schema used by analysis services. Typically comes from one source, but can be derived from several sources. You can create joins in your data sources that do not exist in your ROLAP model.
- **Cubes** —A set of measures and dimensions used to analyze data. This is the primary delivery mechanism for OLAP services.
- **Dimensions**—As the name implies, these are dimensions. Typically, there are additional rows in the dimension tables, such as audit columns and SCD columns, that we do not wish to include in the OLAP presentation. Also, we can rename our columns in our MOLAP dimensions, provide different display and sort values, and create hierarchies.
- **Mining Structures**—Defines a domain for a data mining models.
- **Roles**—Allows you to secure dimensions, measures, and/or values in the cubes. Your NetID requires admin access to your **ist722_*yournetid*** database. Without it you will not be able to deploy the database.
- **Assemblies**—Code libraries that are required by the project. These typically include custom data sources or data mining models.

Quick Walk-Through

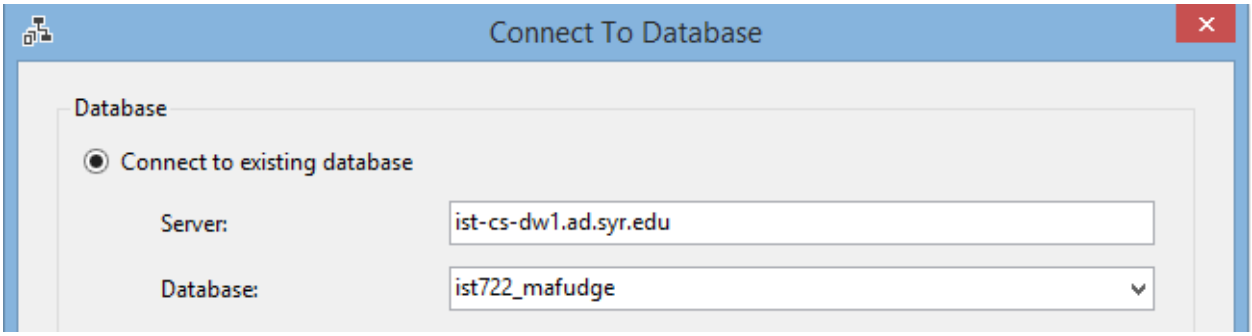
In this section, we'll walk through creating and deploying an empty analysis services database. This will give you a general overview of how the process works and help set up your project for the rest of the lab.

DO THIS: Launch SQL Server Data Tools and connect to your Analysis Services Database.

1) Click the **Windows Start** button.
2) Click the **Visual Studio 2017** icon .
3) From the menu, click **File → Open → Analysis Services**

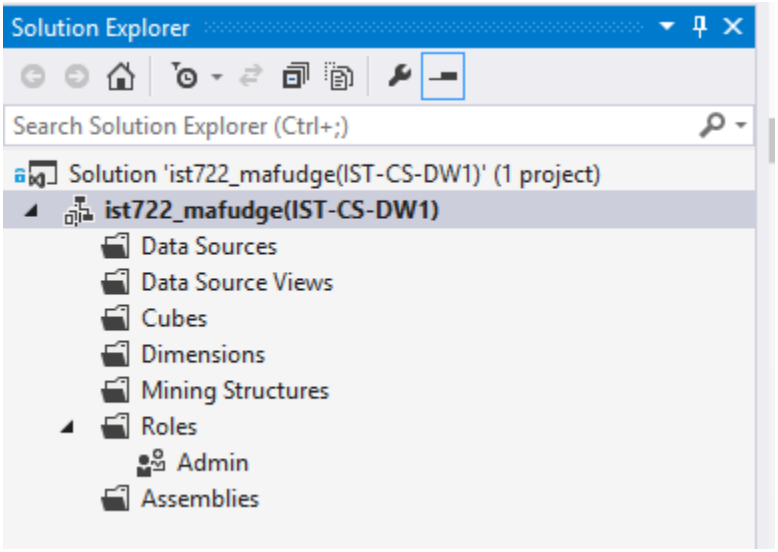


4) From the **Connect to Database** dialog, choose **Connect to existing database** and enter:
Server: **ist-cs-dw1.ad.syr.edu**
Database: **ist722_*yournetid***



Then click **OK.**

5) Under **Solution Explorer** you will see your **Analysis Services Database** on our SQL server.



IMPORTANT: Do not edit the permissions in the database under the Roles folder – if you do, you might remove your access to the database!

If you did these steps successfully, you're ready to begin creating an analysis services database.

Part 2: Walk-Through

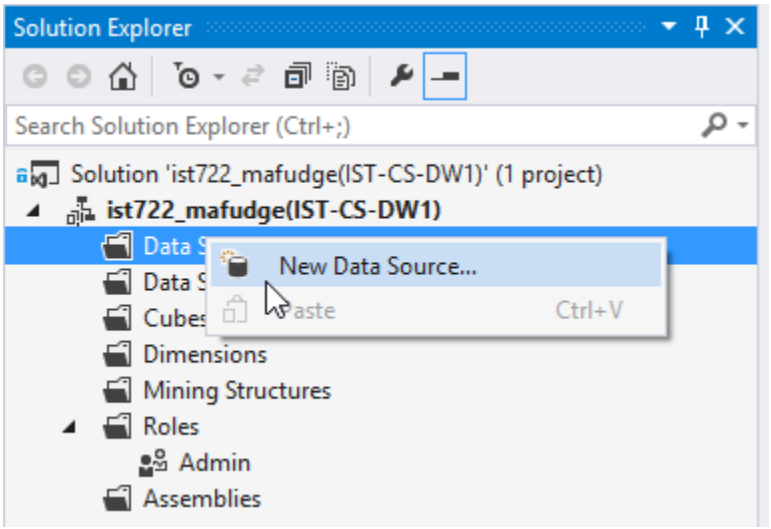
In this part, we will walk through the process to build an analysis services cube for the Northwind inventory dimensional model and access it with Excel 2013. In Part 3 you will repeat the process on your own for the Northwind sales dimensional model.

NOTE: We'll build these cubes based on the Northwind sales and inventory dimensional models we populated via ETL in the previous lab. If you did not complete that lab, you will not have any data!

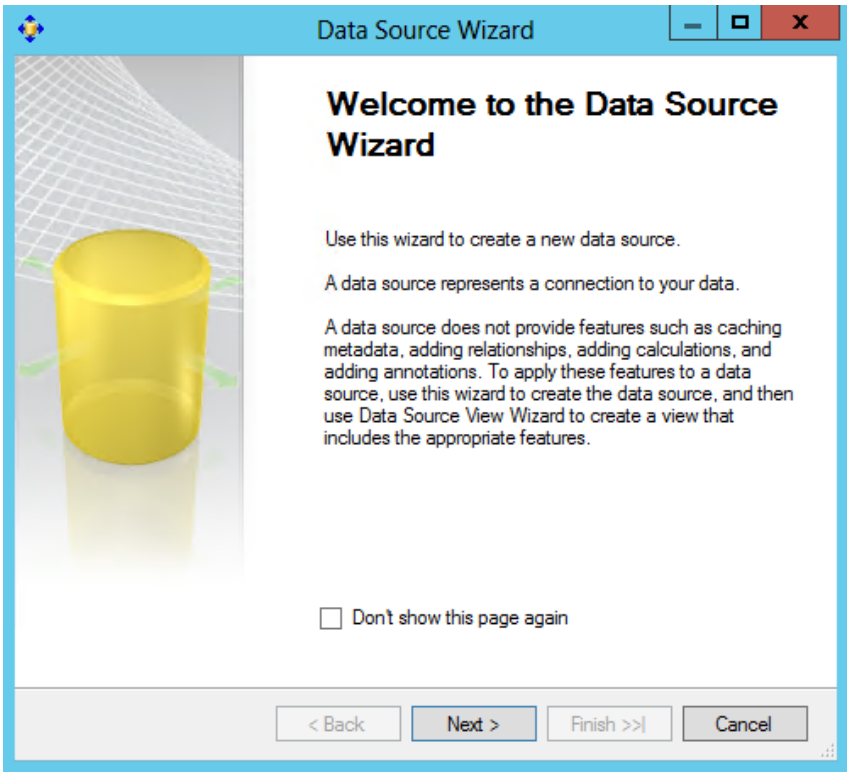
Step 2.1: Add a Data Source to the Project

The first thing you must do in your analysis services project is connect to one or more data sources. Since we're following the Kimball model of storing our MOLAP solution in a ROLAP structure, we will need only one connection to our **ist722_yournetid_dw** database containing the relational version of our dimensional model.

1) Right-click **Data Sources** in **Solution Explorer** then click **New Data Source…**

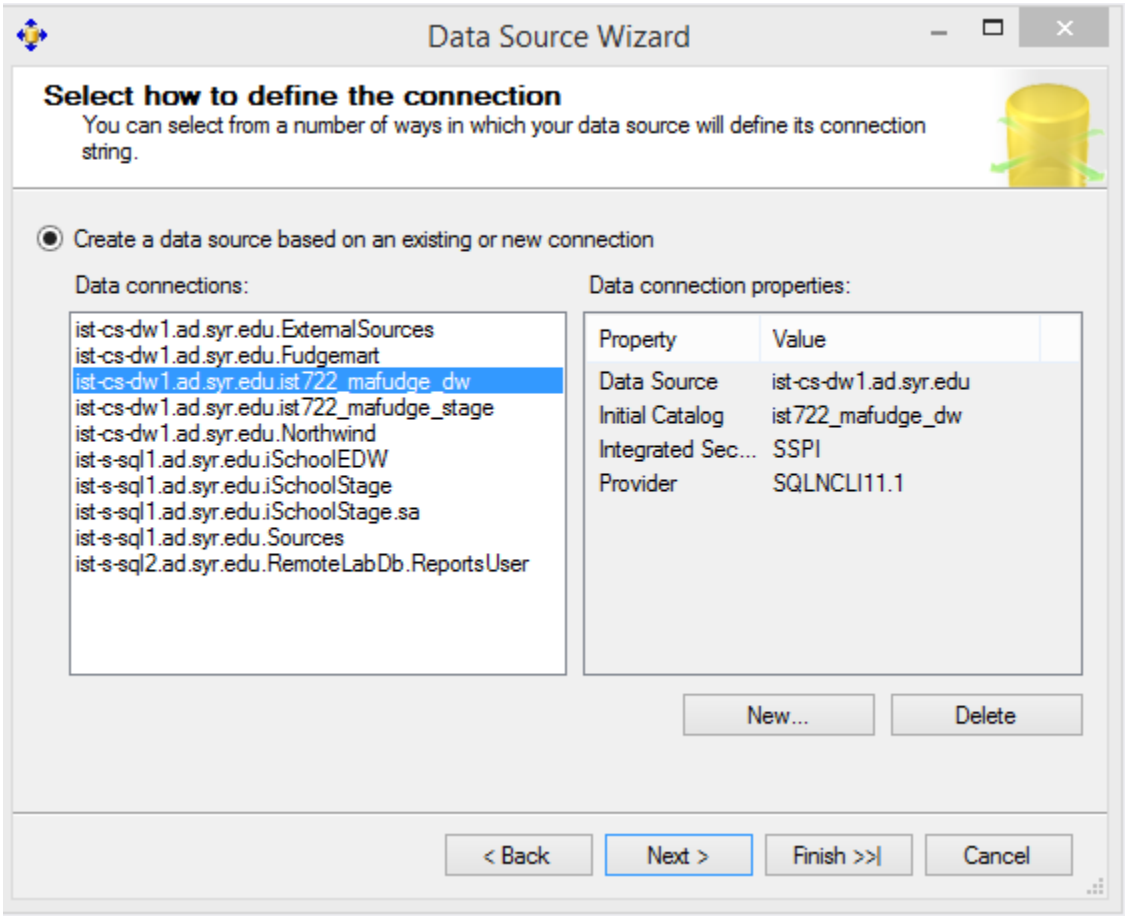


2) You'll see the **Data Source Wizard** dialog.



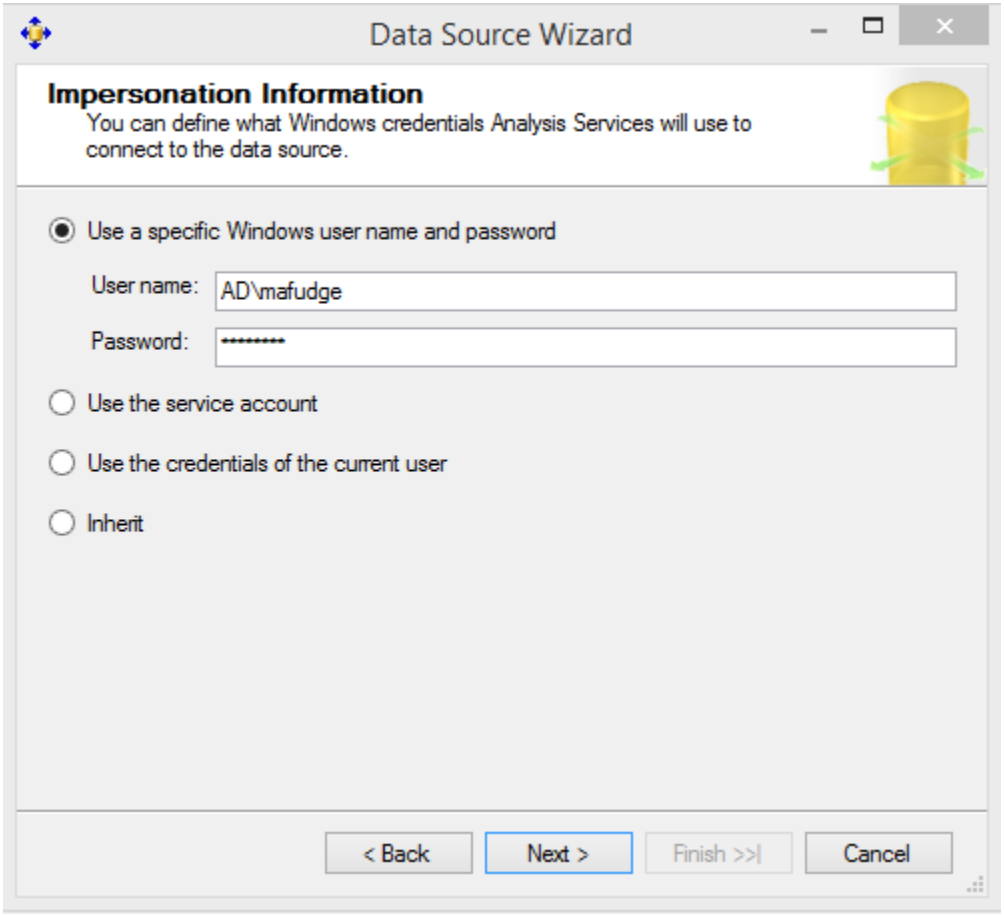
Click **Next>.**

3) This next dialog is familiar from the Integration Services lab. Choose the **ist722_*yournetid*_dw** connection.



Click **Next >** to add the connection.

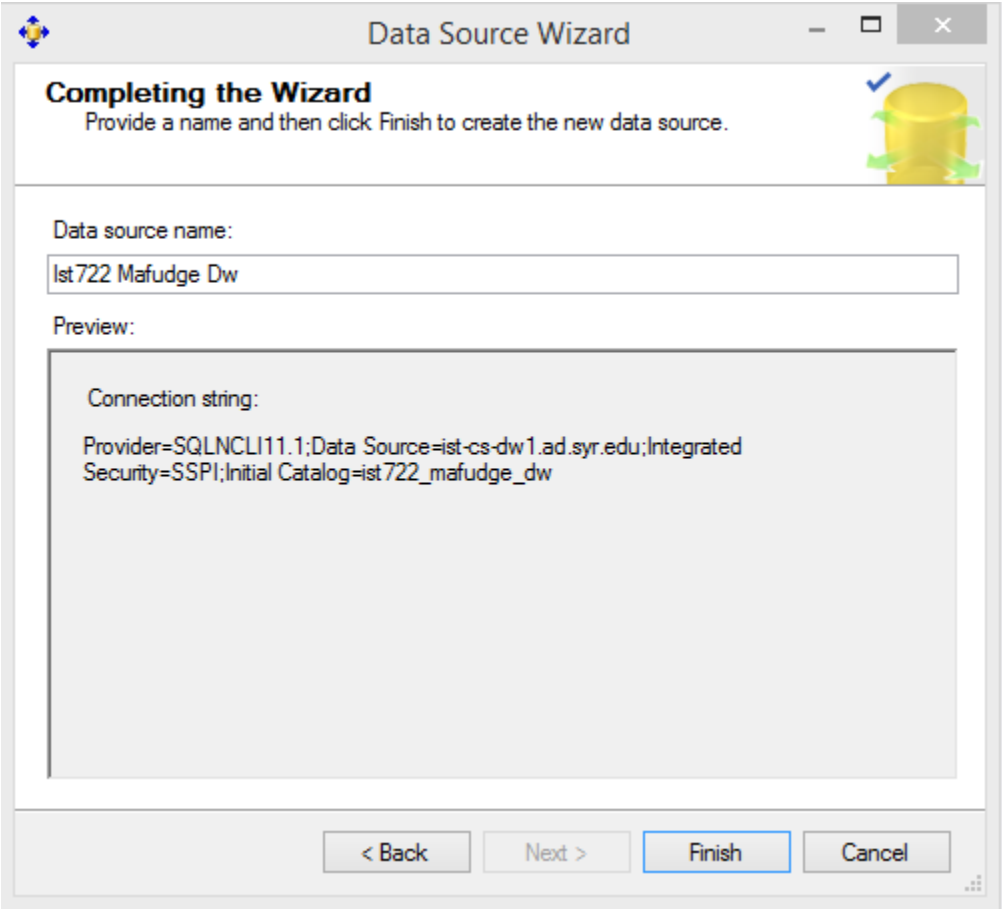
4) You must choose the security credentials for this connection. Under normal circumstances we would inherit this value, but here we need to reenter **your AD\netid and password**. **NOTE**: This is due to our class setup—this is not a common practice!



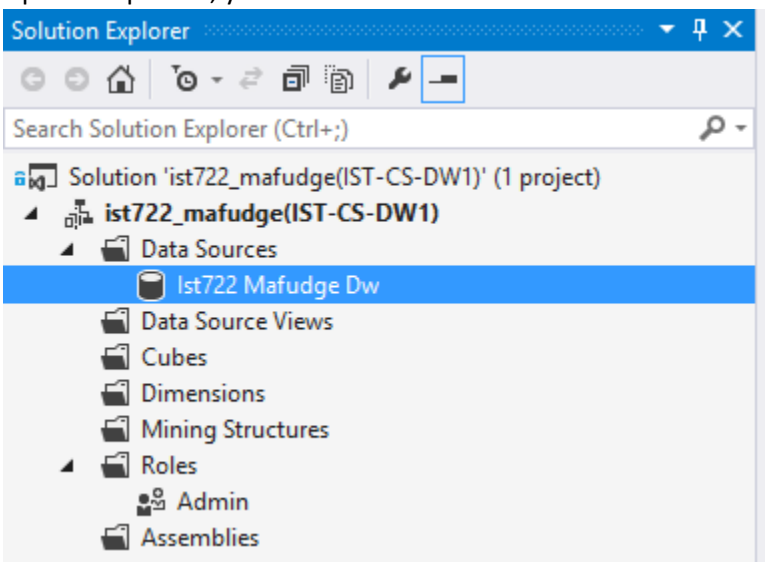
IMPORTANT: Your credentials SHOULD be different from this screenshot. They should match your university credentials—the same information you use to connect to SQL server.

After you enter your credentials, click **Next>.**

5) At the final step in the wizard, you name the data source. Keep the default name and click **Finish**.



6) Upon completion, you will now see the data source in **Solution Explorer**:

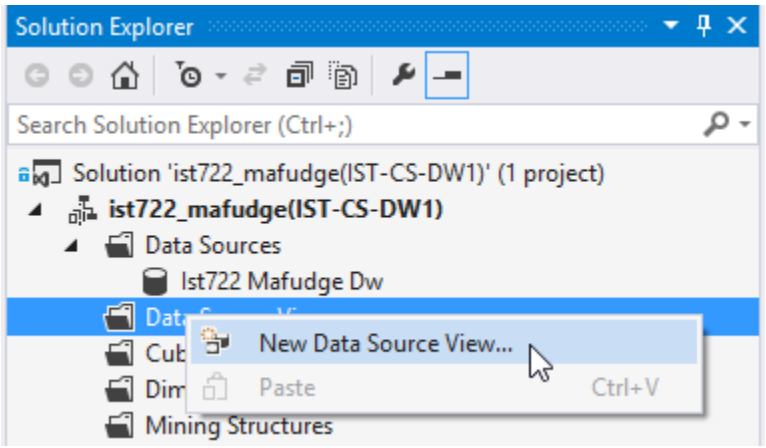


TIP: It's a good idea to save your work periodically! Do so now.

Step 2.2: Create the Data Source View

The data source view contains the logical model of the relational objects (tables, keys, columns, and constraints) that will be used by your OLAP database to build out cubes. We will create the data source view for the inventory daily snapshot dimensional model.

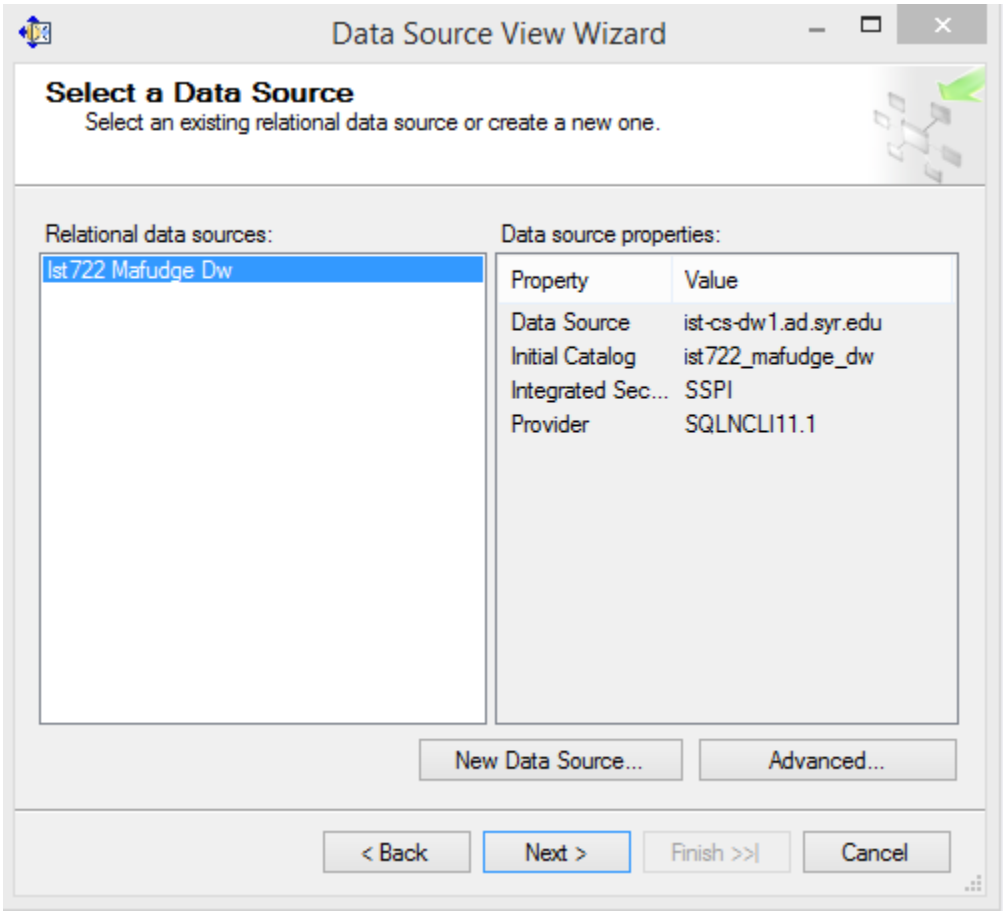
1) Right-click **Data Source Views** from within the **Solution Explorer** and click **New Data Source View…** from the menu.



2) This will launch the **Data Source View Wizard.**

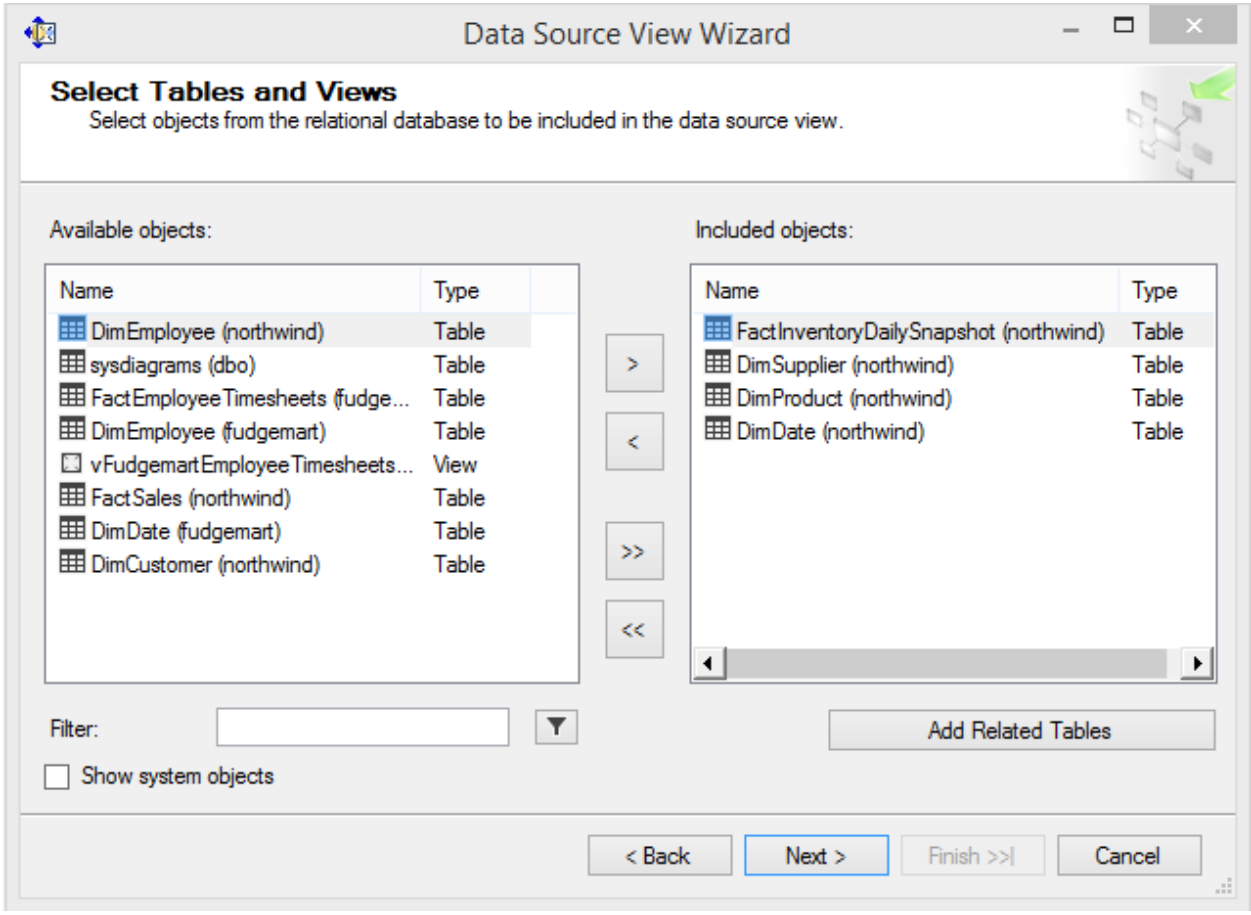
Click **Next>** to begin.

3) Choose your data source. Rather than create one, we'll use the one we built in step 1.



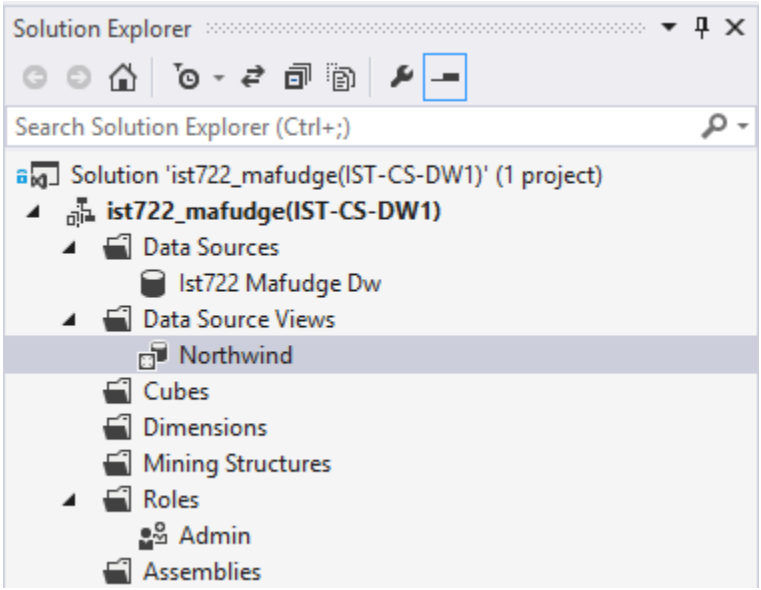
Click **Next>.**

4) Next we must select the tables used by our dimensional model. The easiest way to do this is to add the fact table and then tell the wizard to add the related dimension tables.
   a. Select the **FactInventoryDailySnapshot (northwind)** table from **Available objects** and click the **>** to add it to included objects.
   b. Click **Add Related Tables** to include the dimension tables. This will add the DimSupplier, DimProduct, and DimDate tables to the list of included objects.

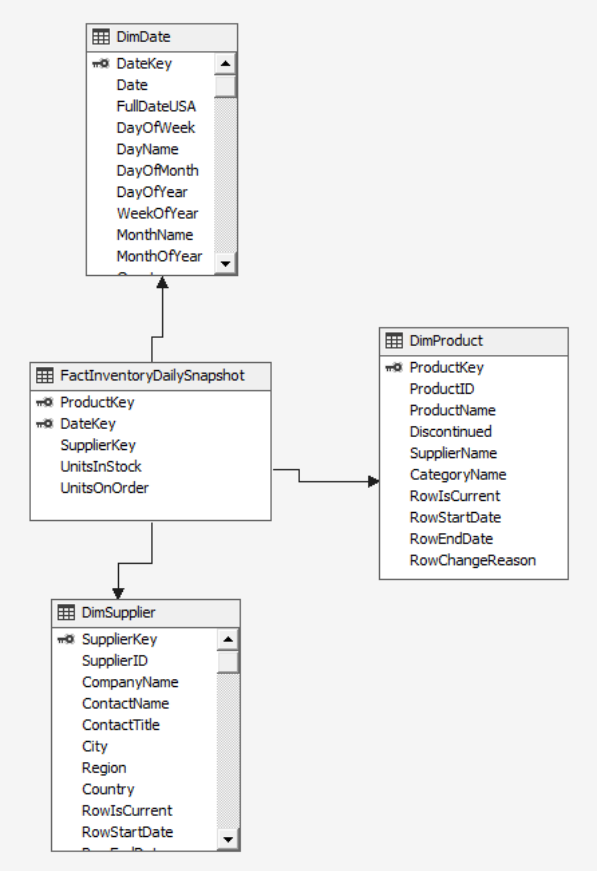


Click **Next >**.

5) You will now see the last step in the wizard. Rename the data source view to **Northwind** and click **Finish**. Under **Solution Explorer,** you will see the **Northwind** data source view:



You will also see a table layout in the data source view:



Step 2.3: Create Dimensions

The next step in OLAP design is to create your dimensions. After you build out and configure your dimensions, you'll move on to creating the cube structure from your fact table.

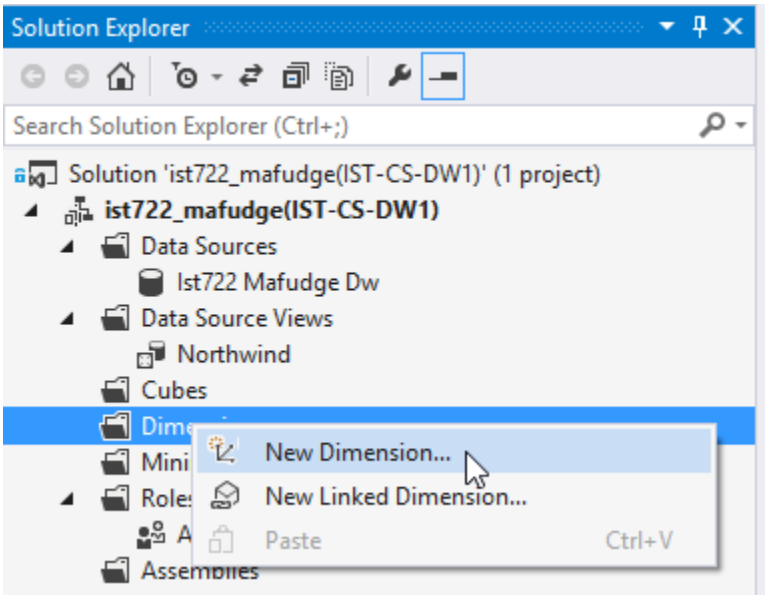
Since OLAP databases are an end-user tool, we want our dimensions to be as self-explanatory and as user friendly as possible. Here are some important guidelines we will follow:

- Use meaningful but descriptive names.
- Don't expose unnecessary attributes.
- Sort attributes in a meaningful way.
- Build hierarchies to help users understand how to navigate through the dimension.

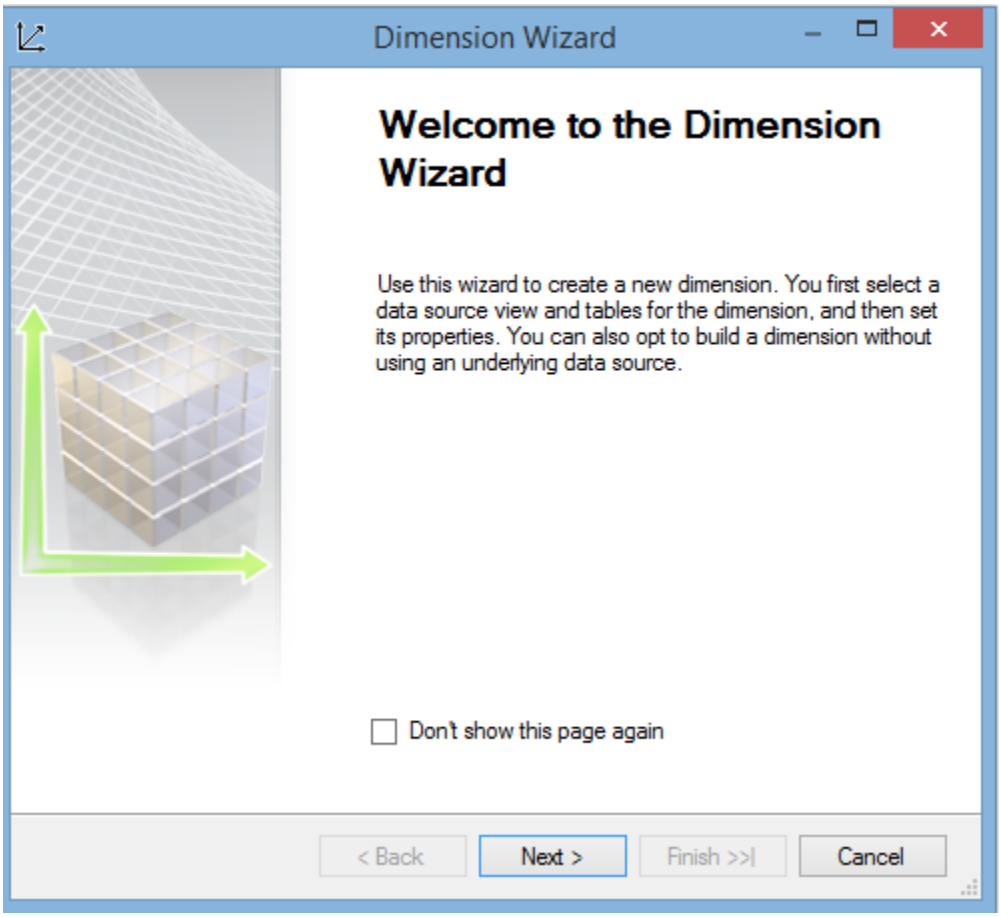
Step 2.3.1: Designing the Date Dimension—Wizard

Let's start out by creating the date dimension using the **Dimension Wizard.**

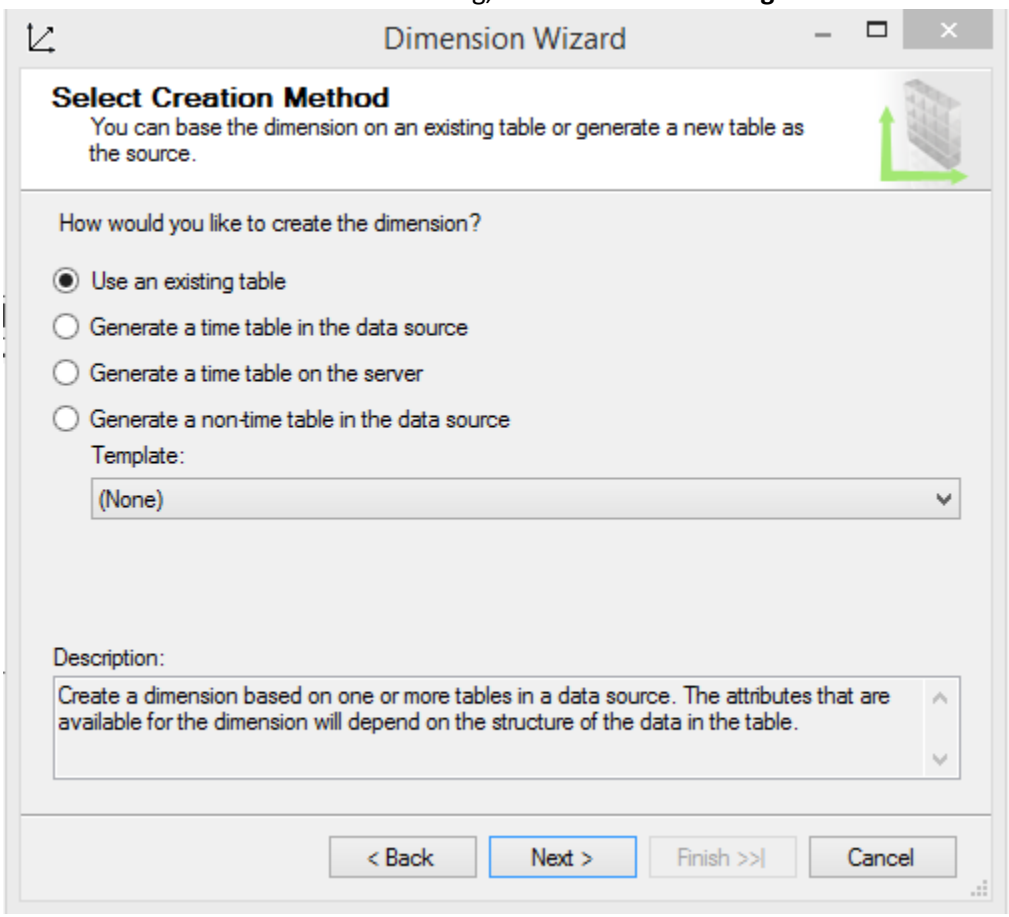
1. From **Solution Explorer,** right-click the **Dimensions** folder and select **New Dimension…** from the menu.



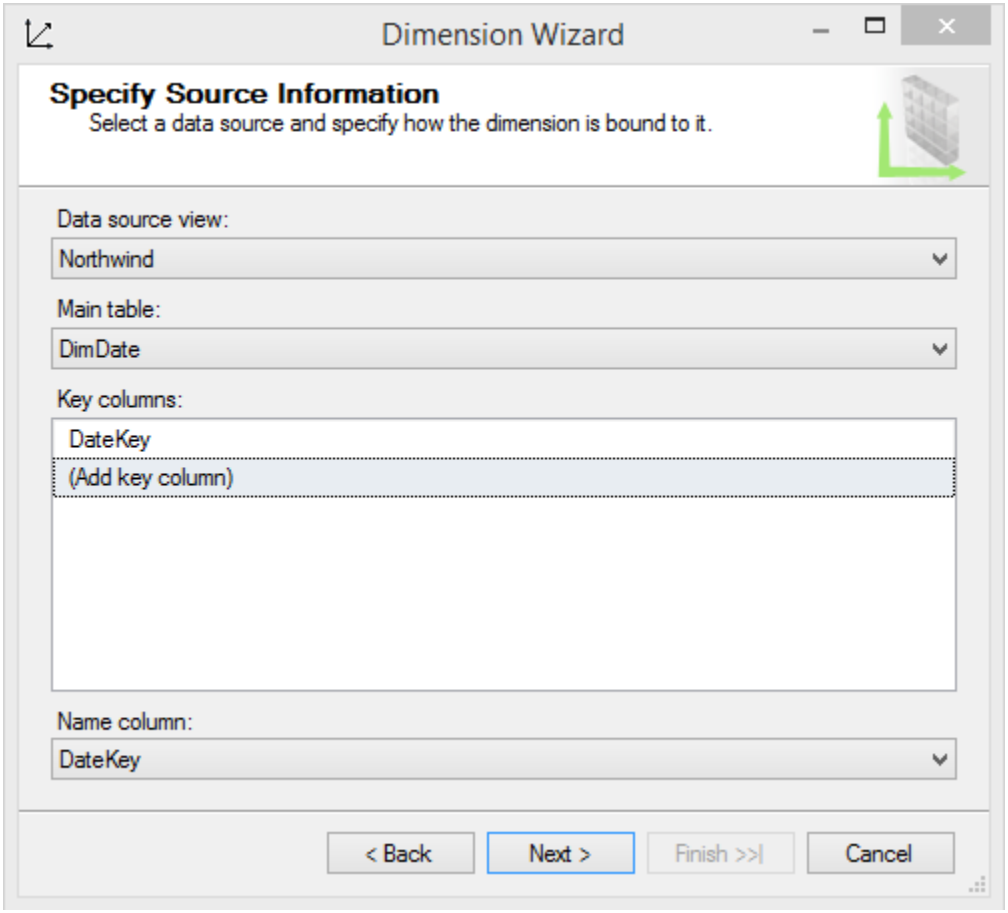
2. This invokes the **Dimension Wizard.** Click **Next >** to get started.



3. Dimensions can be created “top down” or “bottom up.” In top-down design, you design the dimension, and SSAS generates the ROLAP table. Since we have the ROLAP schema already, we are doing a “bottom up” design.
   From the **Select Creation Method** dialog, choose **Use an existing table**.

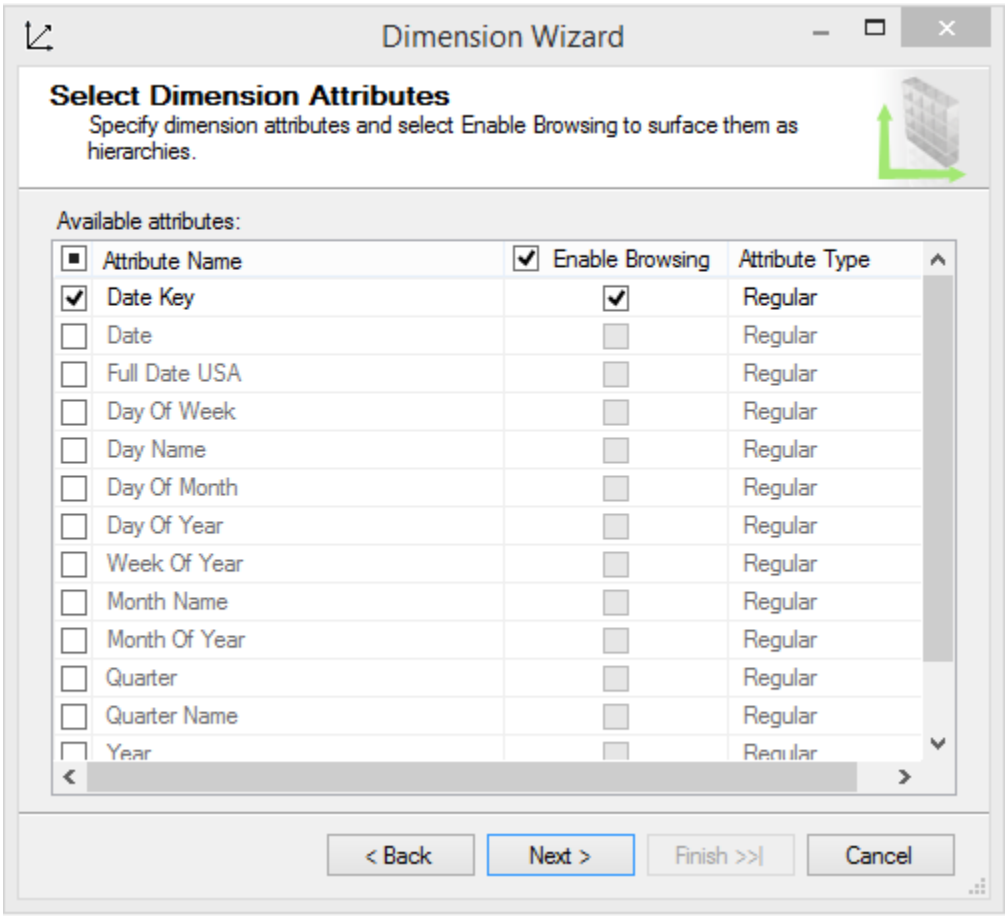


   Click **Next >**.
4. This next dialog allows us to select the dimension from the data source view. You should select the **DimDate** table from the **Northwind** data source view. The key column should be detected automatically as **DateKey,** and you can leave the name column as **DateKey**. We will change this later.



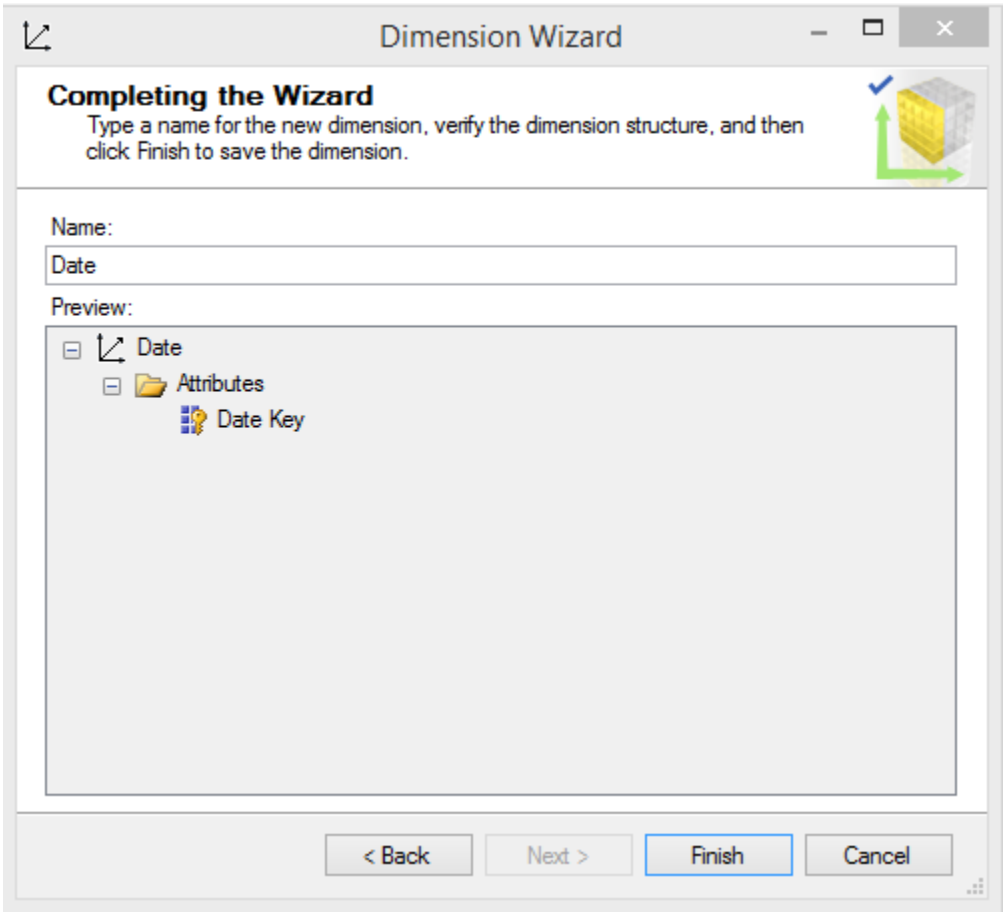
Click **Next >** to continue.

5. On the next screen you will be asked which attribute columns to include in the dimension. Unlike the ROLAP schema, we are not required to include every column in the dimension table. Only **Date Key** should be selected at this time. We will configure the other dimension attributes later.



Click **Next >** to continue.

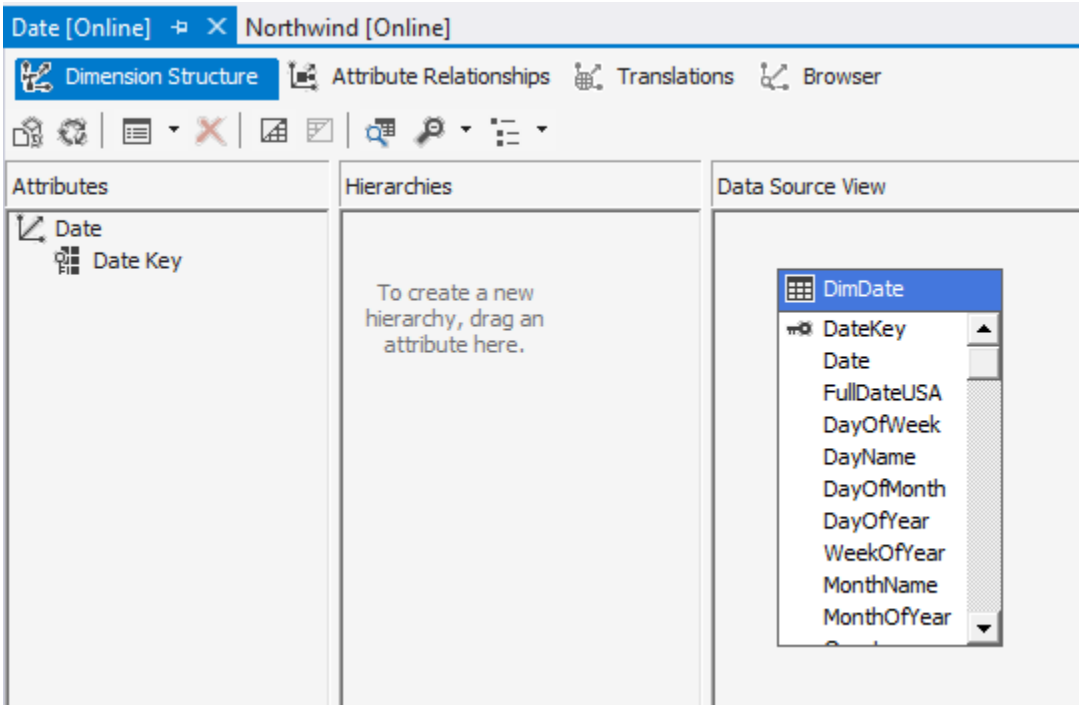
6. On the final screen of the wizard, you will be asked to name the dimension. In the ROLAP schema, the best practice is to add the prefix “Dim” to the dimension. Since the MOLAP schema is user-facing, we drop this convention and just call the dimension “Date.” Name the dimension **Date.**



Click **Finish.**

Step 2.3.2: Designing the Date Dimension—Attributes

After you complete the wizard, you should now see the **Dimension Designer** screen, which is the main user interface for building out and exploring dimensions:



It consists of four tabs along the top:

- **Dimension Structure**—For building out your dimension
- **Attribute Relationships**—For configuring performance-based indexes in your dimension
- **Translations** —For supporting multiple languages
- **Browser**—For viewing the data in your dimension. **NOTE**: You must process the dimension before you can view it. More on that later.

The **Dimension Structure** page in Dimension Designer contains three panes:

- To the far right is the **Data Source View.** This represents the available attributes from the source table for which you can populate your dimension.
- To the far left is the **Attributes** pane, where you build out the visible elements of the dimension.
- In the middle is the **Hierarchies** pane, where you build out navigational hierarchies of your dimension.

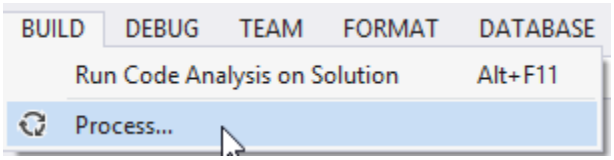
Tangent: Processing and Browsing Dimensions

A key part of the dimension and cube development procedure is browsing data to see the results of your design changes. **Before you can browse a change you make, you must first process it.** Processing simply loads the source ROLAP data into the MOLAP database.

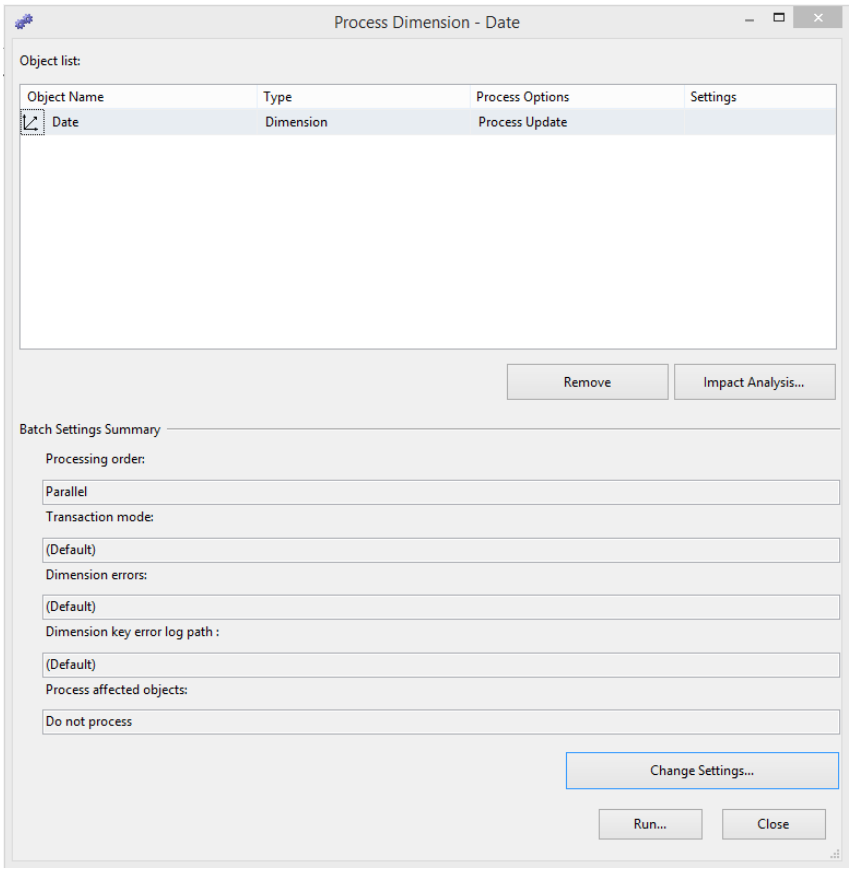
As you are building out your MOLAP database, you will iterate the "Develop → Process → Browse Data" cycle several times. In large MOLAP databases, processing can take a considerable amount of time,

which is why this procedure is not automated. It's a good idea to become accustomed to the procedure. **Try this now.**

1. You process your database by selecting **Build → Process…** from the menu.

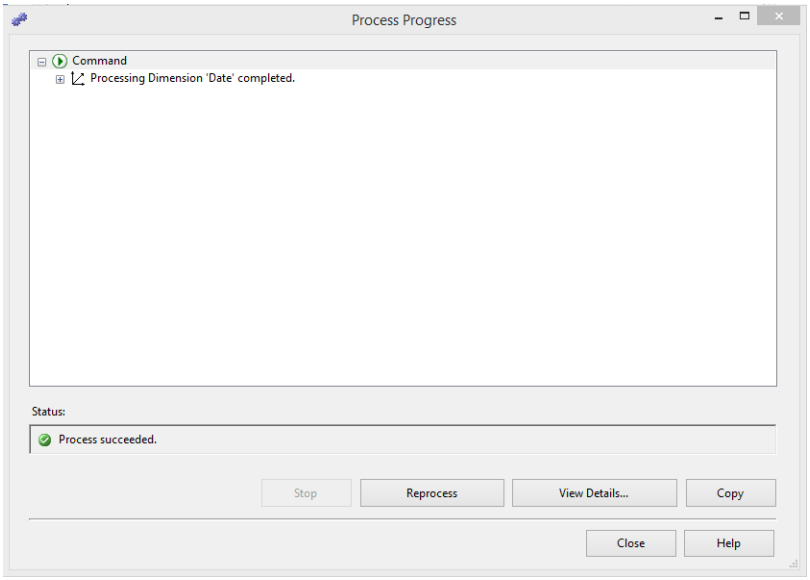


2. Next you'll see a Process Options dialog. This informs you of the changes that will be processed.



Click **Run…** to execute the changes.

3. Finally, you'll see the **Process Progress** dialog where you can view the status of the processing.

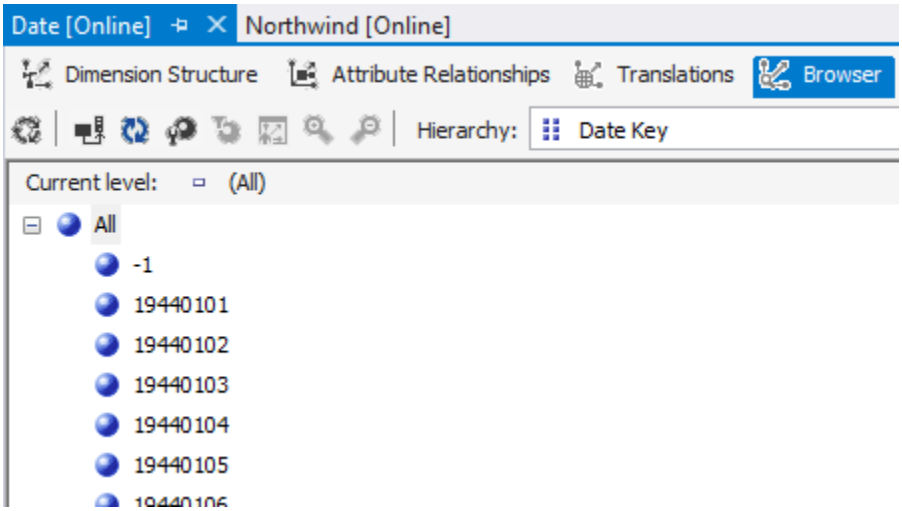


4. You can **close** both dialogs after the processing is complete.

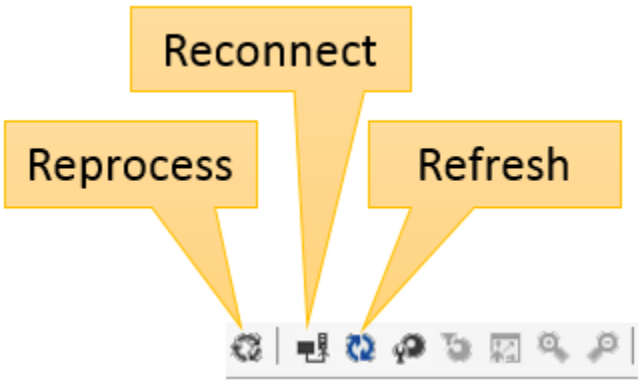
In future steps in the lab, when you're asked to **Process** your database, you should follow steps 1-4 above: **Menu → Build → Process. → Run… → Close → Close**.

After you process the database you can click the **Browser** 

tab to view the data in the dimension. **Try this now.** You should see the **Date Key** hierarchy and be able to view Date Keys under **All.**



IMPORTANT: If the information you see is not accurate or up-to-date, you might need to reconnect to the database and refresh the data. Both of these icons are in the browser toolbar:

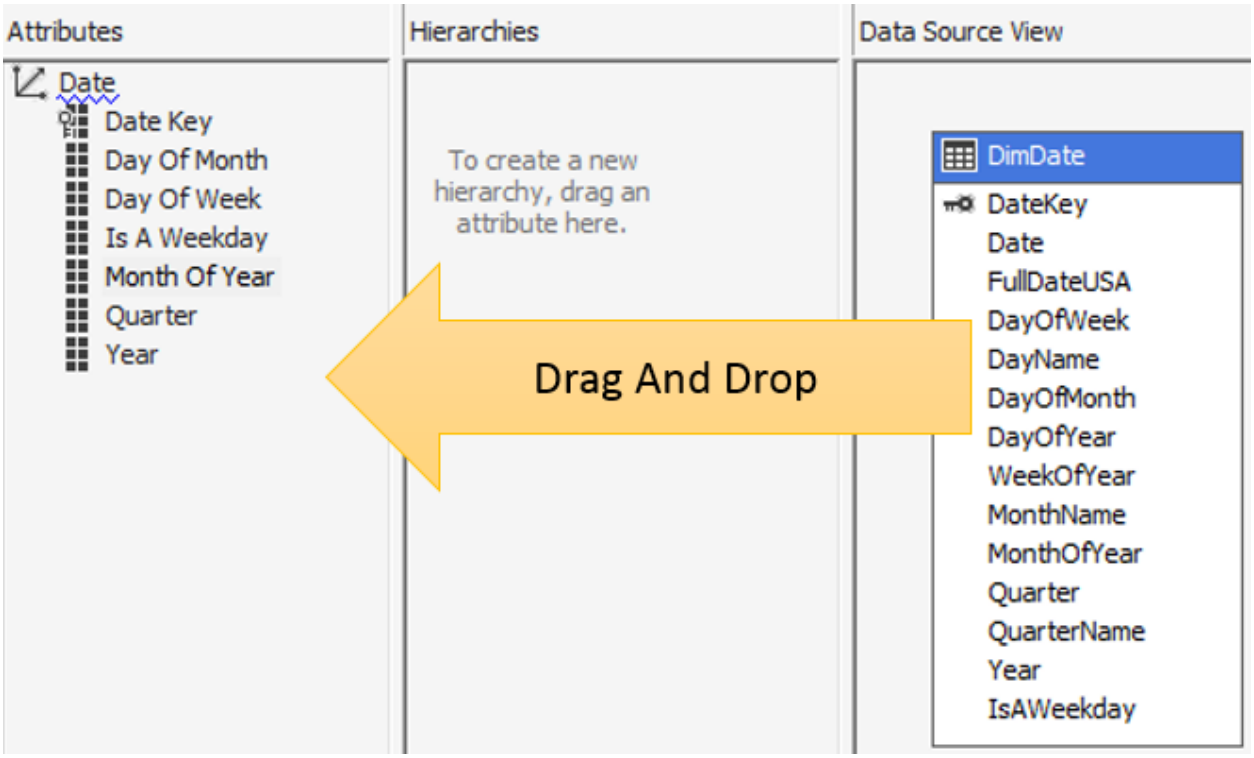


Get to know these three icons. You'll be clicking them a lot!

DO THIS: Switch back to **Dimension Structure** before continuing with the lab.

End of Tangent. Back to Dimension Building

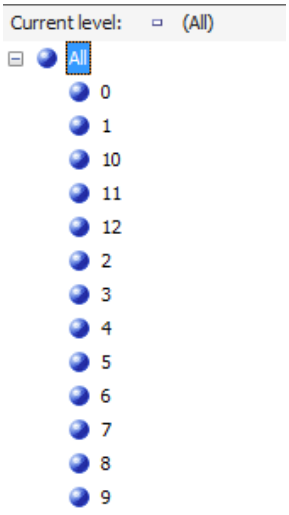
1. The first step in the dimension building process is to add attributes we need in our dimension from the **Data Source View**. From **DimDate** in the **Data Source View,** drag and drop the following columns into the **Attributes** section: **DayOfMonth, DayOfWeek, IsAWeekday, MonthOfYear, Quarter, Year**. When you are finished, they should appear under the **Date** dimension under **Attributes:**



2. Next we need to configure each dimension attribute. For each attribute we should configure the following:
    - Set the **Name.**
    - Set the **Key Column** (the source column that represents the **internal** values of the attribute).
    - Set the **Name column** (the source column that represents the **visible** values of the attribute).
    - Set the **Order by** for the attribute (Should we sort by the key column or the visible column?).

    You might be wondering, "What do each of these attributes mean and why are they important?"

    As an example, let's look at the current **Month of Year** attribute. Before you can see what the data in this dimension looks like, you'll need to **Process** your database. When you switch to the **Browser** tab and select the **Month of Year** hierarchy, you will see the following:



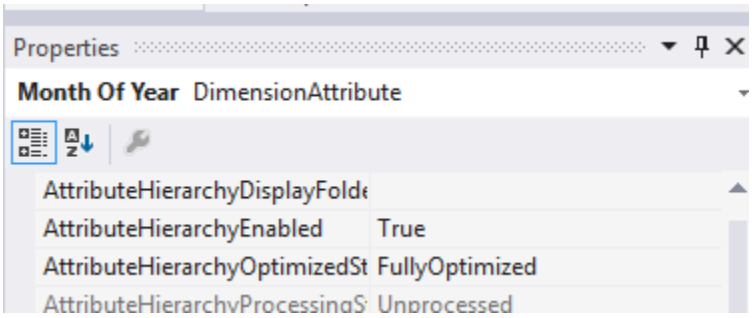
    This attribute is not very useful in its current form:
    - Month names make more sense than month numbers.
    - The months are not sorted correctly.
    - It makes more sense to show the name of the month but sort by the number.
    - What is month 0?

    Luckily we can fix these issues by simply configuring the dimension attributes.

    Switch back to the **Dimension Structure** tab.

    Click the **Month Of Year** attribute in the **Attributes** section.

    On the right side of the screen, you will see a properties window for this attribute:



- Find the **Name** property and change it to **Month.**
- Find the **NameColumn** property and change it to **MonthName.**
- Find the **Order By** property and change it to **Key.**

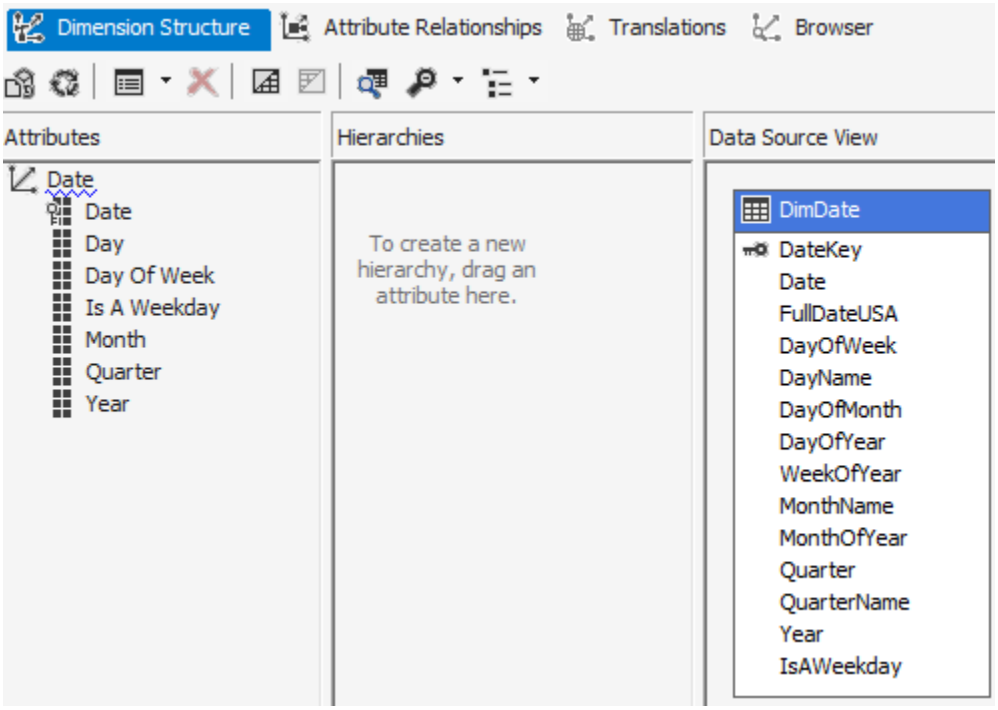
Now save your changes, process the database, and view your new **Month** dimension!

3. Now that you understand how to configure dimension attributes, complete the remaining attributes for the **Date** dimension:

Attribute	Property	Value
Date Key	Name	Date
	KeyColumn	DateKey
	NameColumn	FullDateUSA
	ValueColumn	Date
	Order By	Key
Day Of Month	Name	Day
	KeyColumn	DayOfMonth
	Order By	Key
Day Of Week	Name	Day Of Week
	KeyColumn	DayOfWeek
	NameColumn	DayName
	Order By	Key
Is A Weekday	Name	Is A Weekday
	KeyColumn	IsAWeekday
	Order By	Key
Quarter	Name	Quarter
	KeyColumn	Quarter

	NameColumn	QuarterName
	Order By	Key
Year	Name	Year
	KeyColumn	Year
	Order By	Key

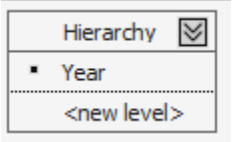
When you are finished, you should have the following attributes in your **Date** dimension: **Date, Day, Day of Week, Is A Weekday, Month, Quarter,** and **Year.**



Step 2.3.3: Designing the Date Dimension—Hierarchies

The final step in configuring a dimension is to configure the hierarchies. These help the user to navigate/drill down through the data in the dimension. Together let's create the classic Year → Quarter → Month → Date hierarchy.

1. Drag and drop **Year** from the **Attributes** section into the **Hierarchies** section. This creates the initial hierarchy:

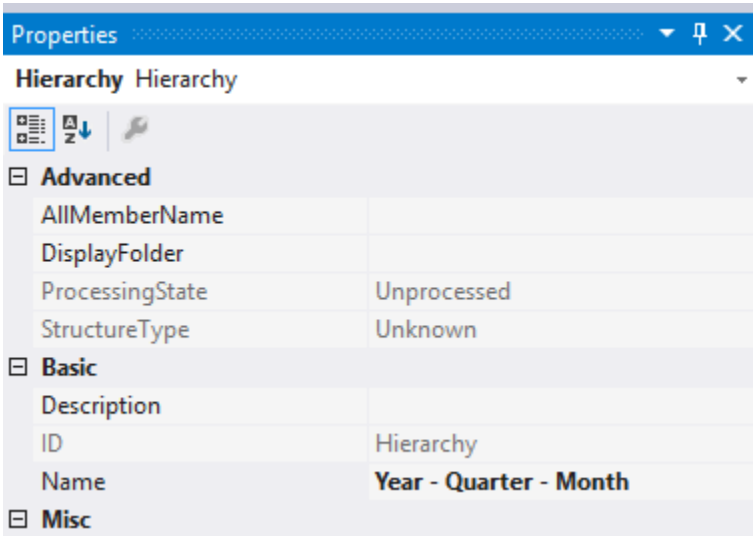


2. Next drag and drop **Quarter** under **Year** in the hierarchy:

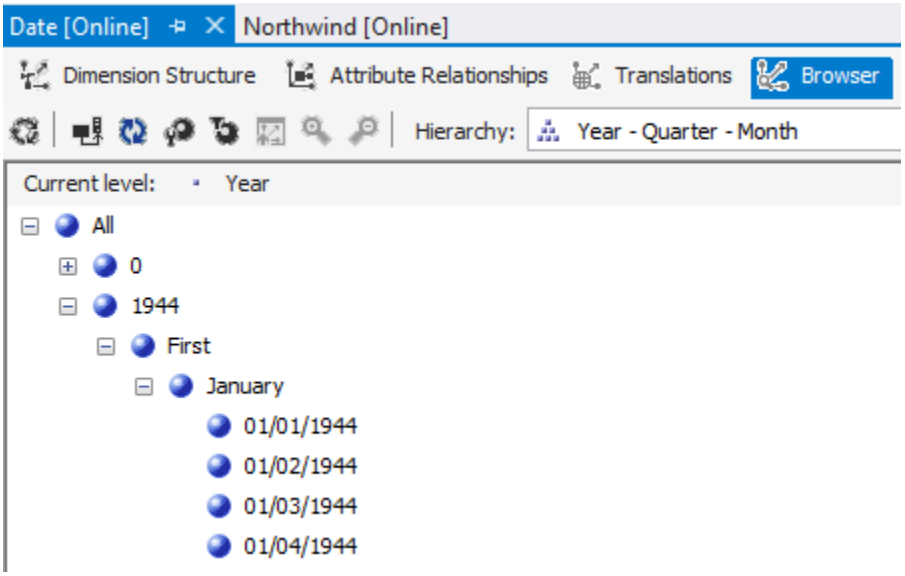
3. Repeat the process for **Month** and **Date**, completing the hierarchy:

NOTE: If you build the hierarchy incorrectly, you can always delete items from it and recreate it.

4. Finally, name the hierarchy by setting the name property to **Year – Quarter – Month.**



5. **Save** your work and **process** the database so you can view your new hierarchy in the browser:



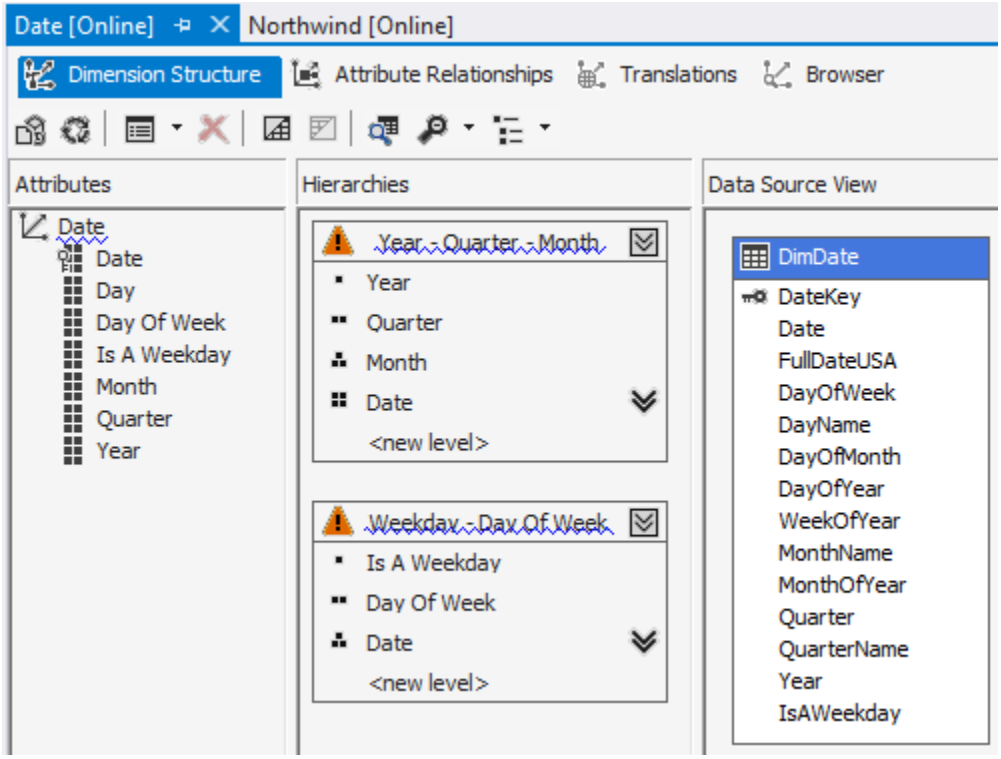
6. Return to **Dimension Structure** when you're done.

NOTE: You'll notice blue squiggly lines in various places within the Dimension Designer. These squiggles are best practice warnings. These are suggestions for building a better dimension. You can ignore them as they are beyond the scope of our lab.

7. Now you create this hierarchy on your own:

Name	Value
Hierarchy Name	Weekday – Day Of Week
Hierarchy	Is A Weekday → Day Of Week → Date

When completed, you should have two hierarchies like this:



Step 2.3.4: Other Dimensions

It's time to put your knowledge to work and complete the **Product** and **Supplier** dimensions required for our inventory snapshot cube. Here are the specifications:

Product Dimension

Dimension: **Product**
Table Source: **DimProduct**

Attribute	Property	Value
Product Key	Name	Product
	KeyColumn	ProductKey
	NameColumn	ProductName

	Order By	Name
Category Name	Name	Category
	KeyColumn	CategoryName
	Order By	Key
Discontinued	Name	Discontinued
	KeyColumn	Discontinued
	Order By	Key

Hierarchy Name	Hierarchy
Category – Product	Category → Product

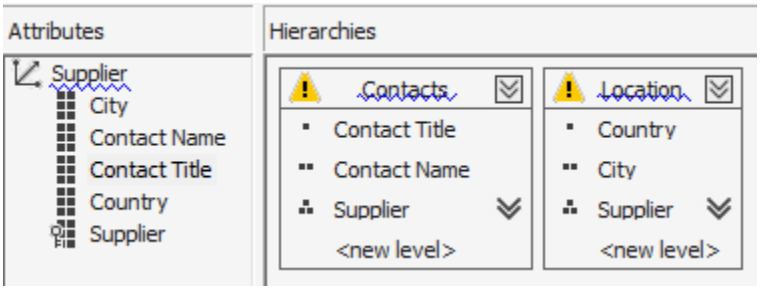
Supplier Dimension

Dimension: **Supplier**

Table Source: **DimSupplier**

Attribute	**Property**	**Value**
Supplier Key	Name	Supplier
	KeyColumn	SupplierKey
	NameColumn	CompanyName
	Order By	Name
City	Name	City
	KeyColumn	City
	Order By	Key
Country	Name	Country
	KeyColumn	Country
	Order By	Key
Contact Name	Name	Contact Name
	KeyColumn	ContactName
	Order By	Key
Contact Title	Name	Contact Title
	KeyColumn	ContactTitle
	Order By	Key

Hierarchy Name	Hierarchy
Contacts	Contact Title → Contact Name → Supplier
Locations	Country → City → Supplier



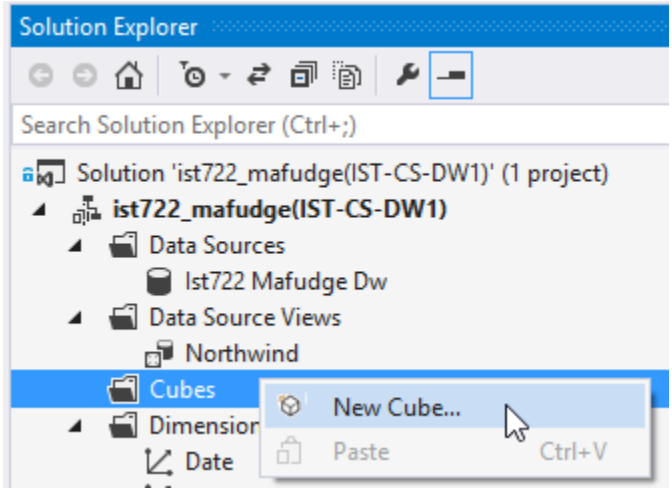
Step 2.4: Create the Cube

Now that the dimensions are complete, it's time to create the cube. The cube is the MOLAP version of the dimensional model. In this step we will create a cube from our data source view. Eventually, this cube will be used by end users for ad-hoc queries.

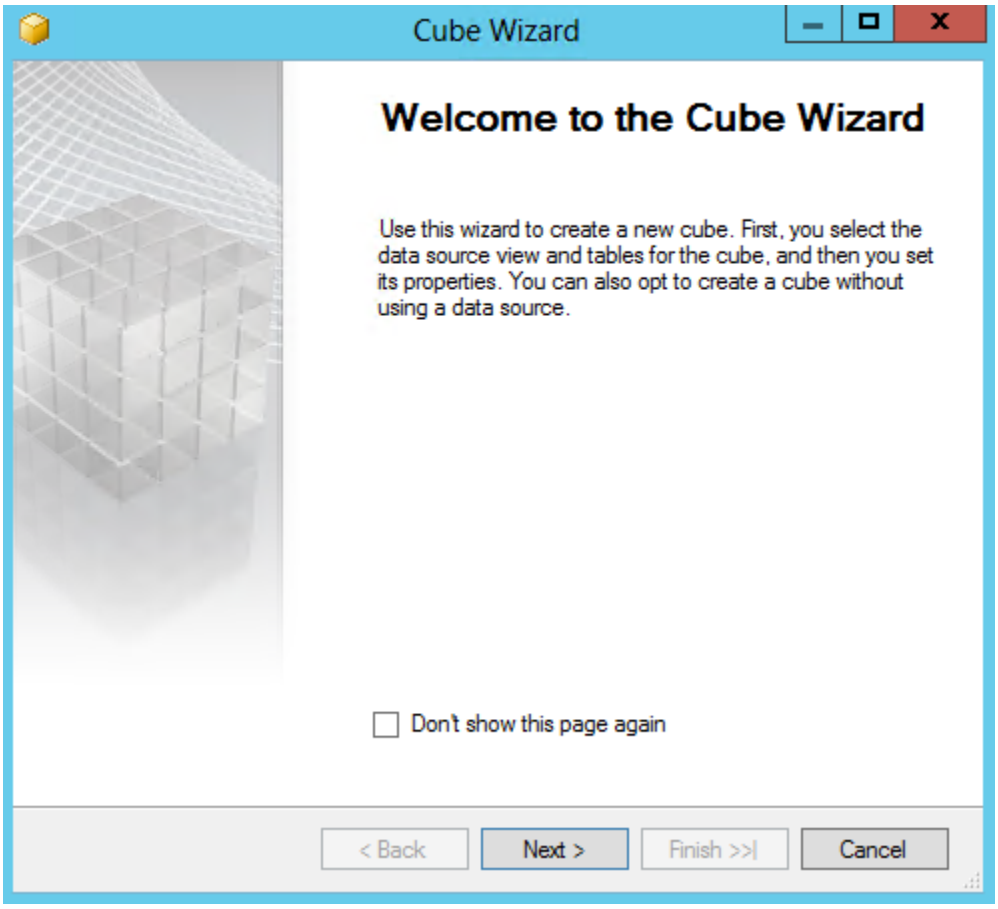
Step 4.3.1: Cube Wizard

To get started building a cube, use the wizard.

1) Right-click on **Cubes** from **Solution Explorer** and select **New Cube…**

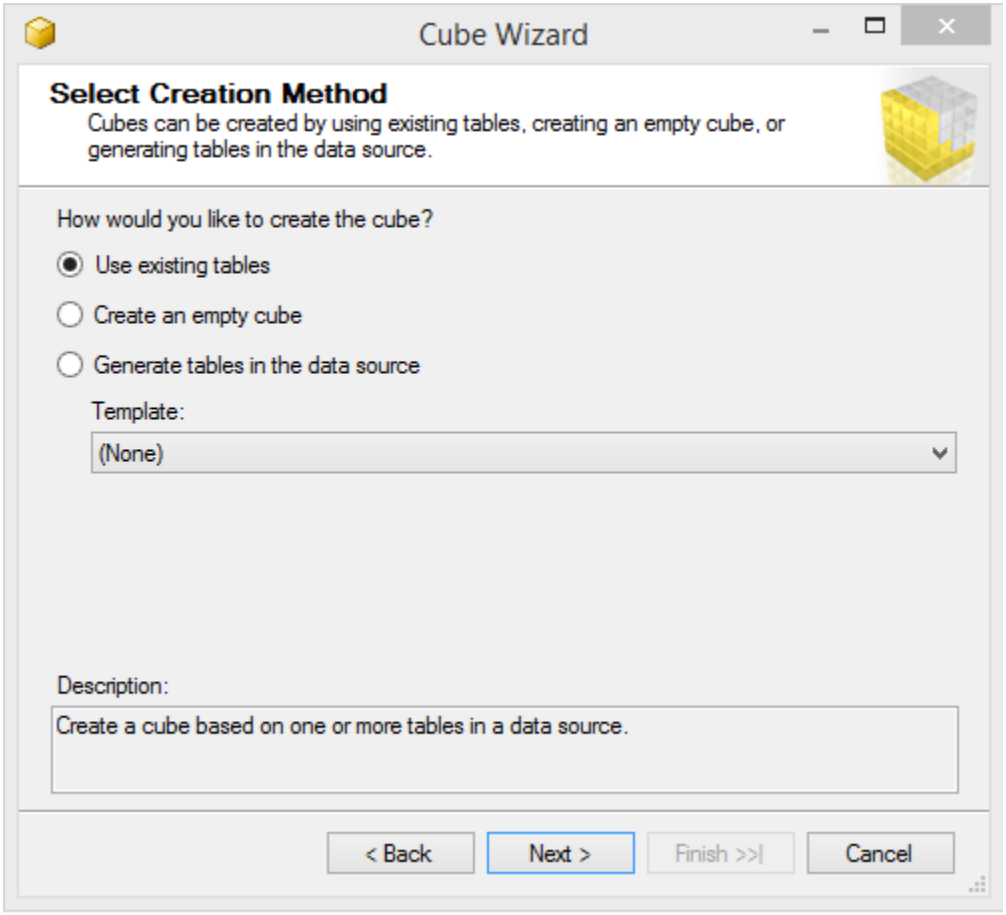


2) This will start the **Cube Wizard**:



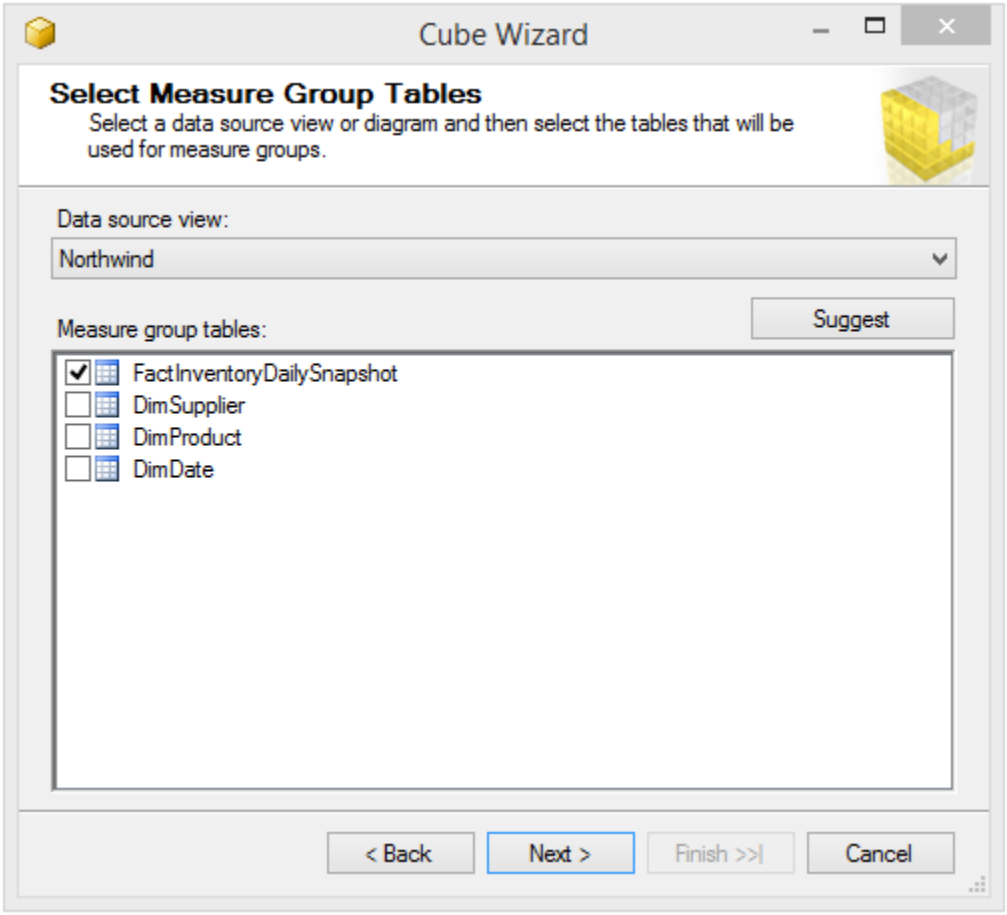
Click **Next>.**

3) We'll build our cube from the "bottom up" using the fact table we have already added to our data source view:



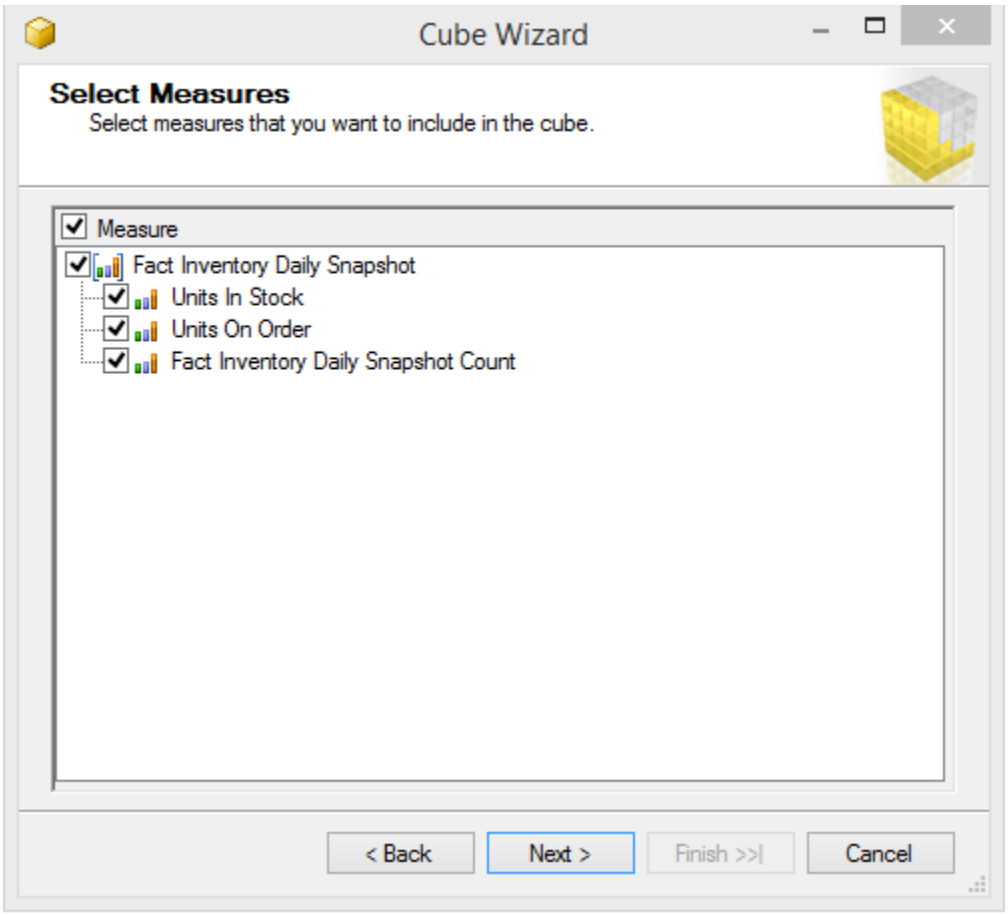
Select **Use existing tables** and click **Next>.**

4) In this step, you should select the fact table as it contains the measurements required by our cube:



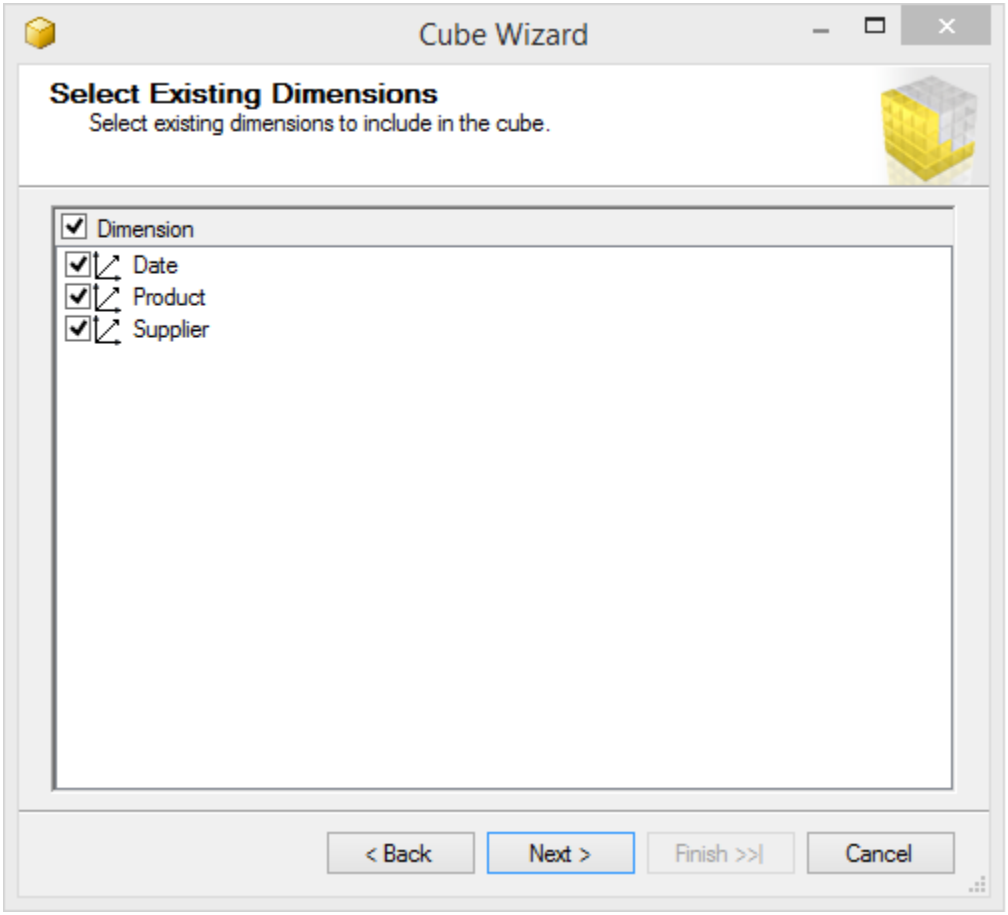
Select on **FactInventoryDailySnapshot** and click **Next>**

5) Next we select measures from our fact table that we would like to include in our cube. While most of the time you'll select all the columns in your fact table, this screen gives you the opportunity to omit things "detected as facts which are not" like degenerate dimensions and audit columns from your cube.



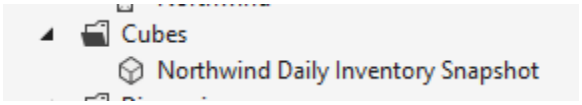
Select all columns and click **Next>.**

6) At this step, you will get to select dimensions. The dimensions are discovered automatically from the foreign keys in your fact table. If the same dimension (like a date dimension) is used more than once, then it will be added multiple times as a role-playing dimension.

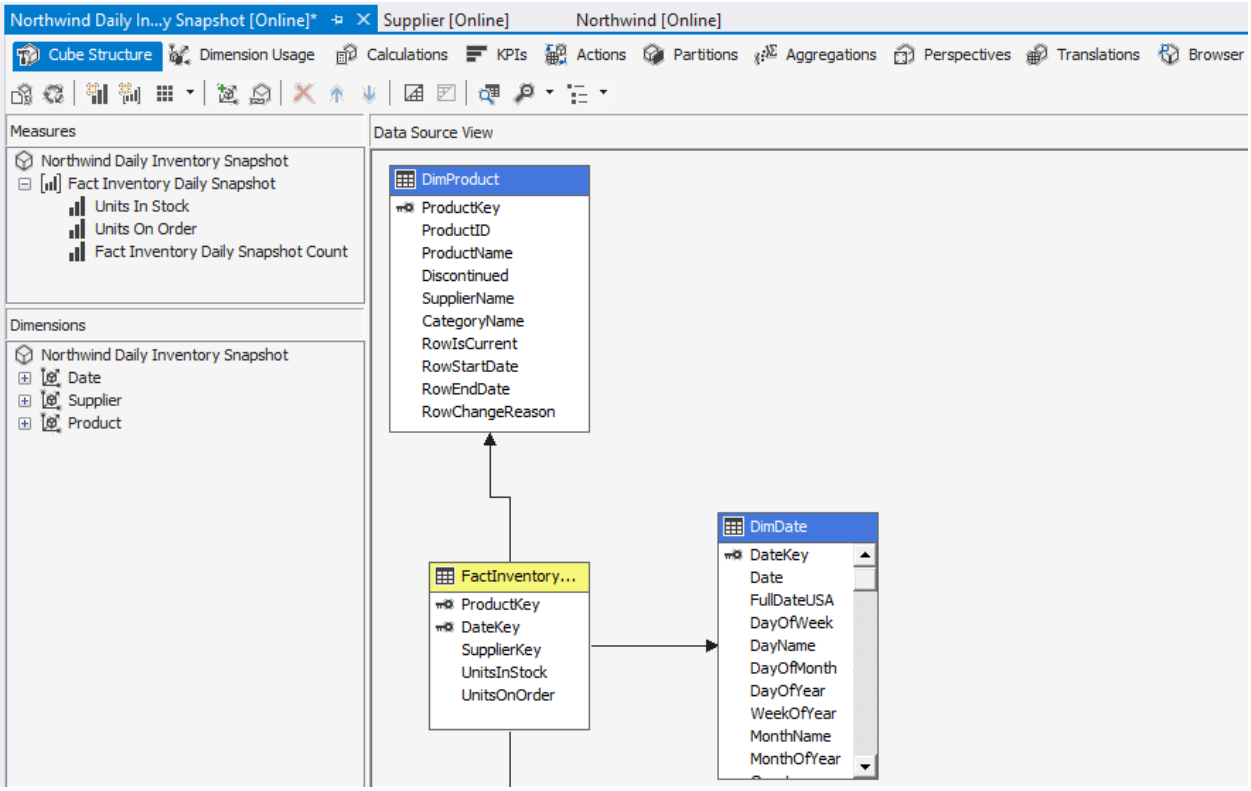


Select all the dimension tables and click **Next>.**

7) In the last step, you can name your cube. Type in **Northwind Daily Inventory Snapshot** and click **Finish.** You should see your cube loaded into the project:



After you complete the wizard, you should now see the **Cube Designer** screen, which is the main user interface for building out your cube:



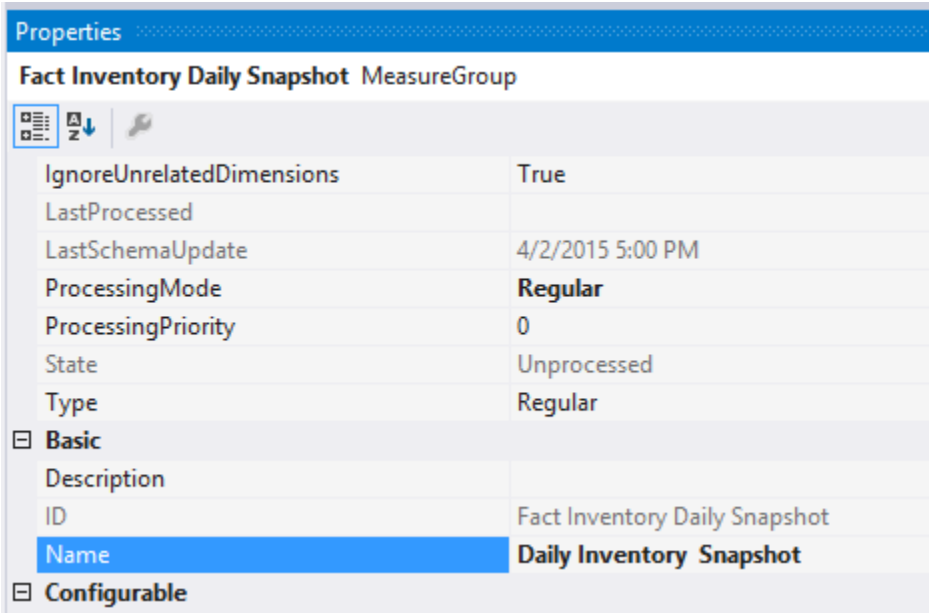
There are 10 tabs along the top:

- **Cube Structure**—for building out and configuring the measures in the cube
- **Dimension Usage**—for configuring how the dimensions are used in the cube. Since we started from an ROLAP schema this will be configured for us automatically.
- **Calculation—**for adding additional calculations to our cube beyond those in the ROLAP schema.
- **KPI's**—configuring Key Performance Indicators for goal and trend setting.
- **Actions**—configure drill-thru to operational data and integration with reports
- **Partitions**—physical divisions of the cube into logical boundaries. For example you can partition the cube by location so each sales branch only receives their own data.
- **Aggregations**—define which calculations should be stored within the cube versus calculated on demand.
- **Perspectives**—set up "views" within the cube to limit access to attributes or measures based on user permissions.
- **Translations—**multi-language support
- **Browser—**view the cube data in a basic pivot-table like format.

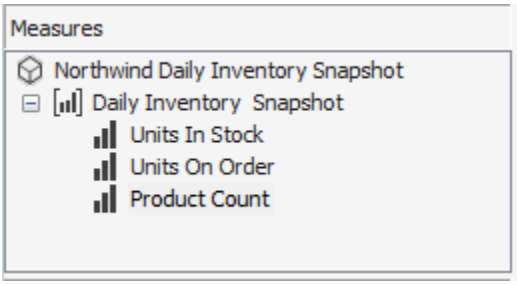
Step 2.4.2: Configure Measure Properties

After the cube wizard is finished, our next step is to configure our measures. From the **Cube Structure** tab you can configure each measure similar to how you configured dimension attributes.

1. First rename the measure group from **Fact Inventory Daily Snapshot** to **Daily Inventory Snapshot**. “Fact” is not a user-facing term, so it should be removed.



2. Rename **Fact Inventory Daily Snapshot Count** to **Product Count**. This better represents the purpose of this measure. When you’re finished, your measures should look like this:



3. Save your changes and process your database.

Step 2.4.3: Building Key Performance Indicators (KPI’s)

In this next step, let’s build a really basic KPI so you can realize the benefits of this feature.

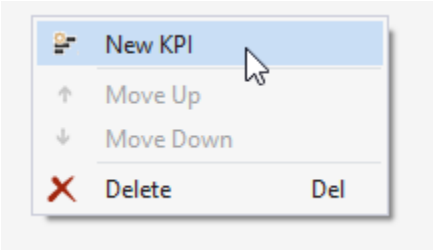
Let’s assume that the inventory manager adheres to the following heuristic for maintaining ideal stock levels of inventory:

Units In Stock	**Status**
>=20 and <=50	Good (ideal stock levels)
>=10 and <20 or >50 and <=60	Okay (not ideal, but not horrible)
Below 10 or more than 60	Bad (too much or too little inventory)

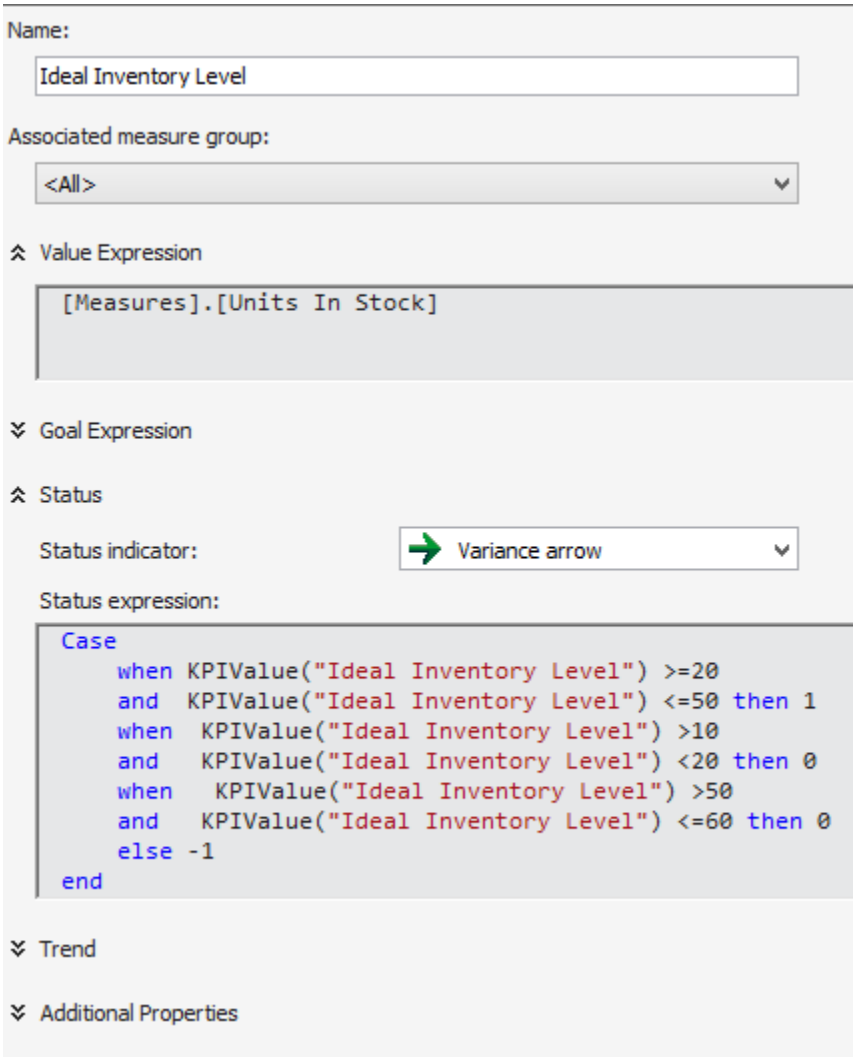
Let’s add this into our cube as a KPI.

1. Click the **KPIs** tab.

2. In the **KPI** organizer section, right-click and select **New KPI** from the menu.



3. Fill out the KPI form as follows:



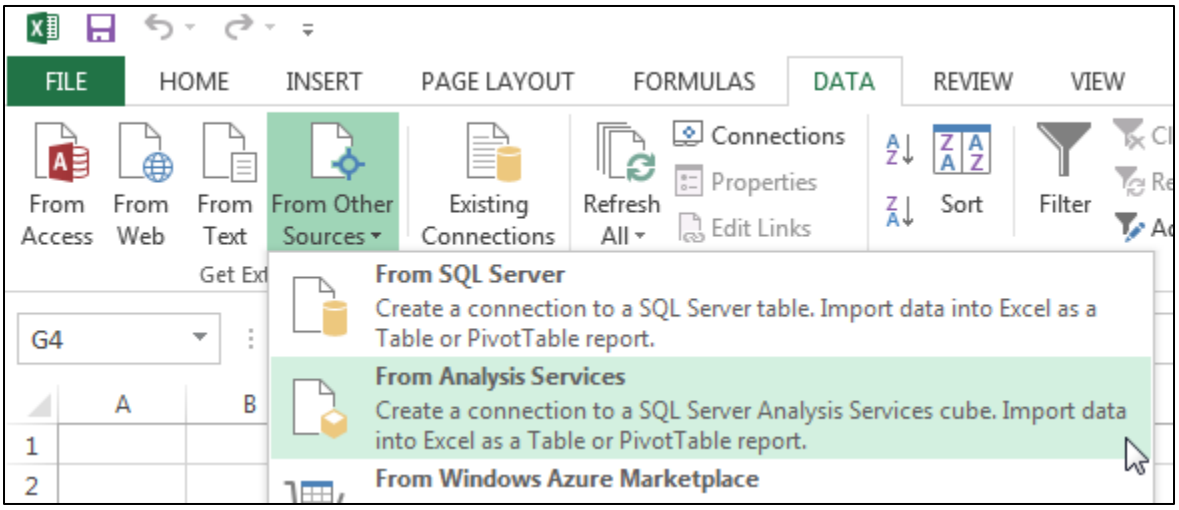
4. Save your changes and process your database.

Step 2.5: Browse Your Cube in Excel 2013 With Pivot Tables

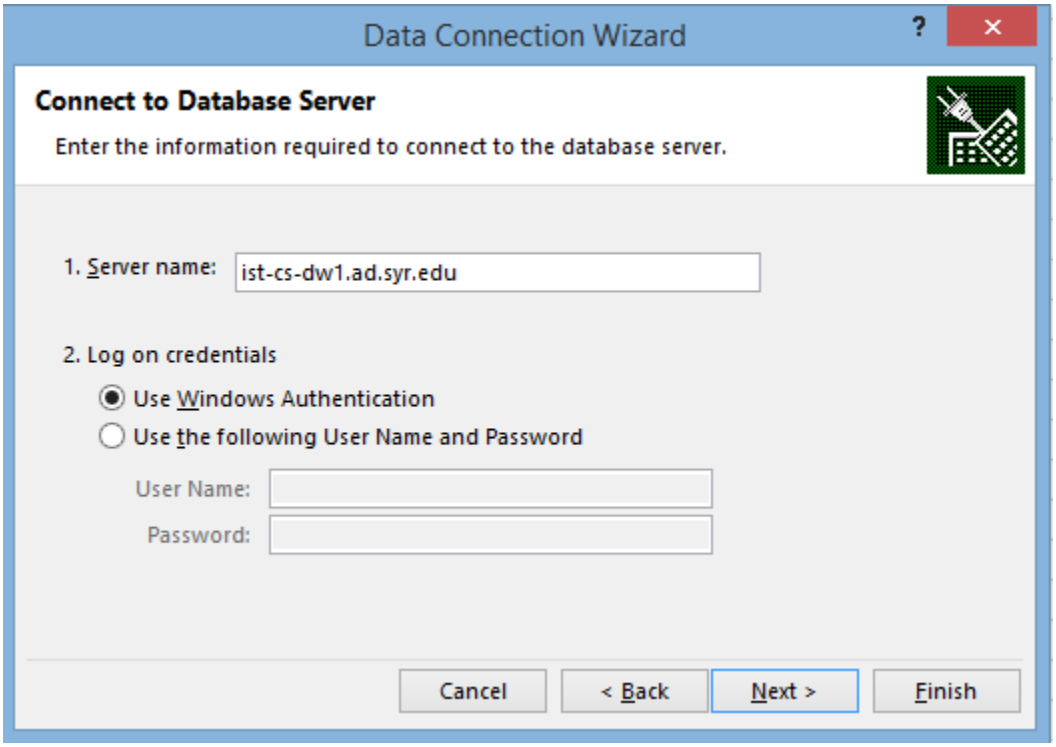
It's time explore our cube and experience all the semantic intelligence we've added, in contrast to the ROLAP schemas we've been using to date. We will use Excel 2013 pivot tables to accomplish this.

1) Open **Microsoft Excel 2013**.

2) Create a new blank workbook.
3) From the ribbon, select **DATA → Get External Data → From Other Sources → From Analysis Services.**

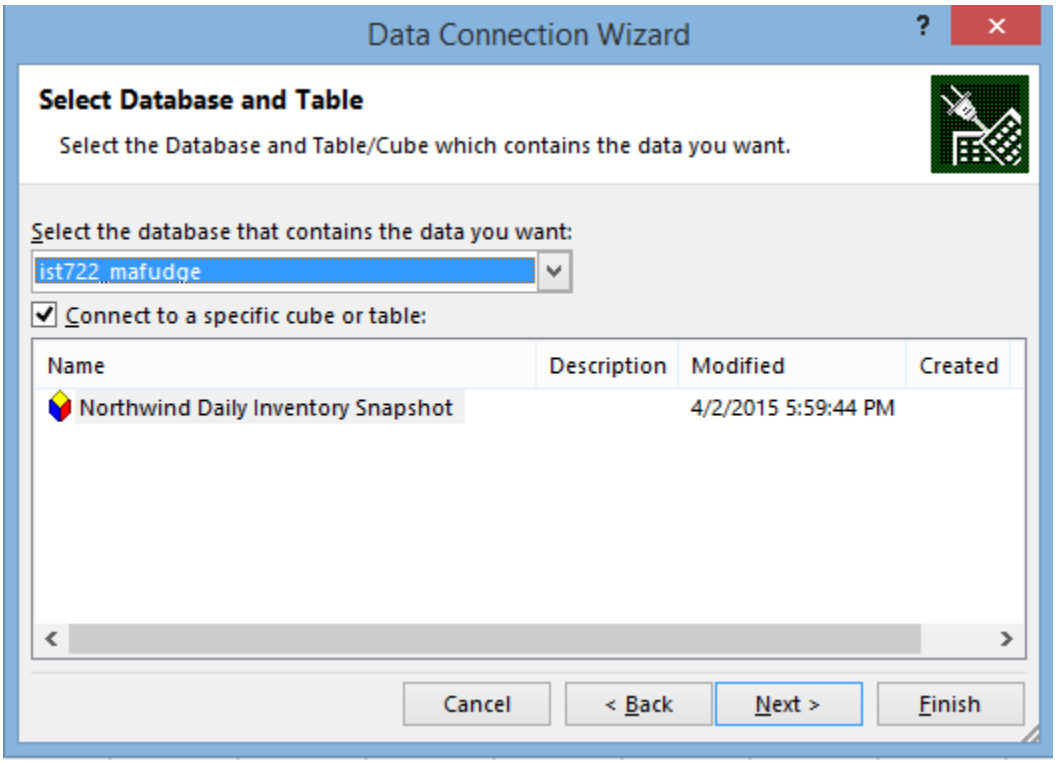


4) Under Server name, type in our SQL server: **ist-cs-dw1.ad.syr.edu**, and under Log on credentials select **Use Windows Authentication.**



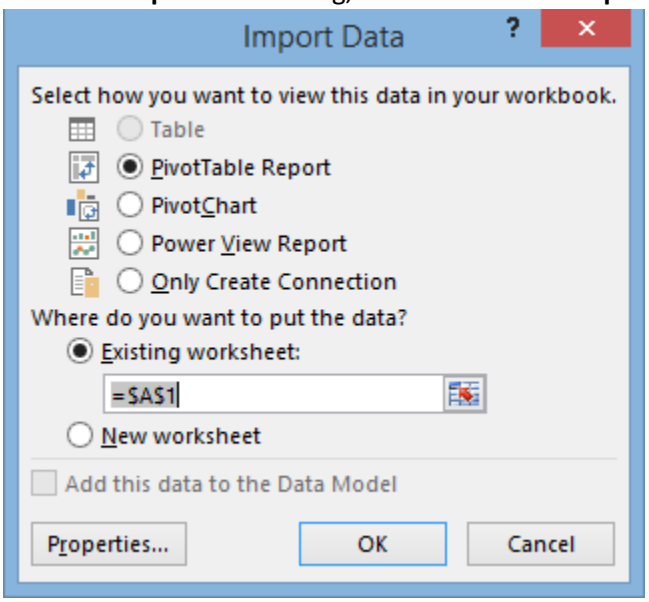
When you've completed the **Data Connection Wizard,** click **Next>.**

5) Select the **ist722_*yournetid*** Analysis Services database and the **Northwind Daily Inventory Snapshot** cube.



After you've make your selections, click **Next>, and** then click **Finish** to save the data connection.

6) From the **Import Data** dialog, select **PivotTable Report** and **Existing worksheet.**

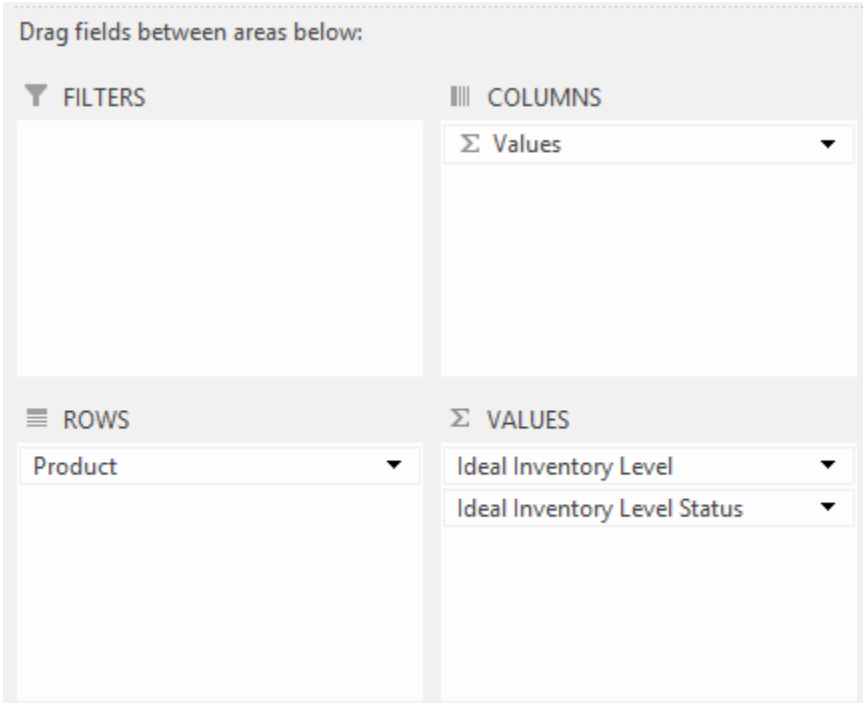


Click **OK** after you have made your selections.

7) Let's check out our KPI.
Drag **Product** to **ROWS.**

Drag the KPI **Ideal Inventory Level Value** to **VALUES.**
Drag the KPI **Ideal Inventory Level Status** to **VALUES.**

Your pivot table fields should look like this:

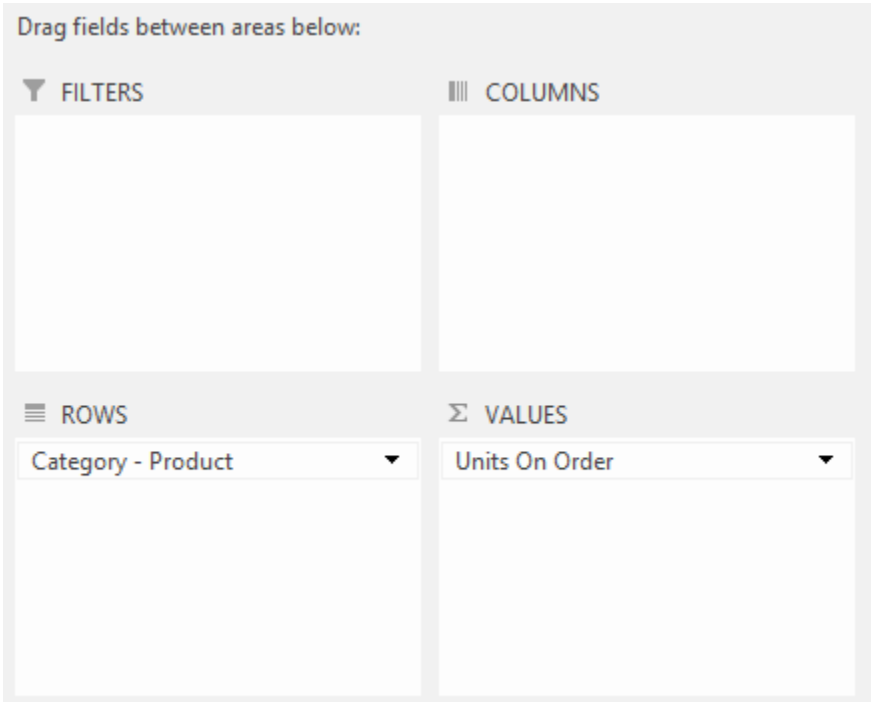


And your pivot table will look like this:

Row Labels	Ideal Inventory Level	Ideal Inventory Level Status
Alice Mutton	0	
Aniseed Syrup	28	
Boston Crab Meat	102	
Camembert Pierrot	12	
Carnarvon Tigers	33	
Chai	30	
Chang	32	
Chartreuse verte	64	
Chef Anton's Cajun Seasoning	41	
Chef Anton's Gumbo Mix	0	
Chocolade	32	
Côte de Blaye	12	
Escargots de Bourgogne	57	
Filo Mix	33	
Flotemysost	20	
Geitost	72	
Genen Shouyu	27	
Gnocchi di nonna Alice	23	
Gorgonzola Telino	21	

8) Let's try another one using hierarchy.
Remove all fields from the pivot table.
Add **Units On Order** to **VALUES.**
-Add **Category – Product** to **ROWS.**

9) Your pivot table fields should look like this:



10) Expanding a **Category** should automatically drill down into products! Welcome to Semantic BI Models!

Row Labels	Units On Order
⊞ **Beverages**	**38**
⊞ **Condiments**	**119**
⊟ **Confections**	**115**
Chocolade	49
Gumbär Gummibärchen	0
Maxilaku	30
NuNuCa Nuß-Nougat-Creme	0
Pavlova	0
Schoggi Schokolade	0
Scottish Longbreads	8
Sir Rodney's Marmalade	0
Sir Rodney's Scones	28
Tarte au sucre	0
Teatime Chocolate Biscuits	0
Valkoinen suklaa	0
Zaanse koeken	0
⊞ **Dairy Products**	**94**
⊞ **Grains/Cereals**	**70**
⊞ **Meat/Poultry**	**0**
⊞ **Produce**	**14**
⊞ **Seafood**	**67**
Grand Total	**517**

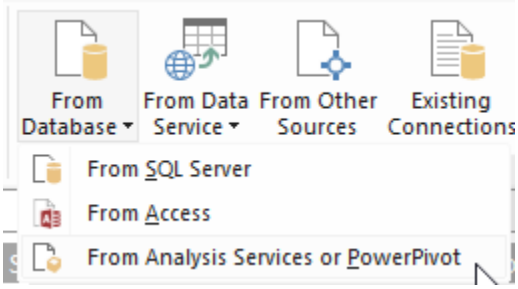
Step 2.6: Excel as an Analytic Tool: PowerPivot and Power View

Next let's explore how you can use Excel to build rapid ad-hoc data models from our data warehouse data. We'll use Excel's PowerPivot tool to do this. After we complete the in-memory model in Excel PowerPivot, we will create a dashboard with Power View.

We'll use them to display a map of units in stock based on supplier country. This will give us an idea of the origin of our products based on their quantities.

1. Launch PowerPivot. **Menu → PowerPivot → Manage** this opens the PowerPivot Add-in, which will allow us to load the SSAS database into memory. **NOTE:** PowerPivot opens in a new window.

2. From the PowerPivot ribbon, select **Get External Data → From Other Sources → Analysis Services.**



3. The **Table Import Wizard** will appear.
   For Server or File Name enter **ist-cs-dw1.ad.syr.edu**
   Select **Use Windows Authentication.**
   For Database name enter **ist722_*yournetid***

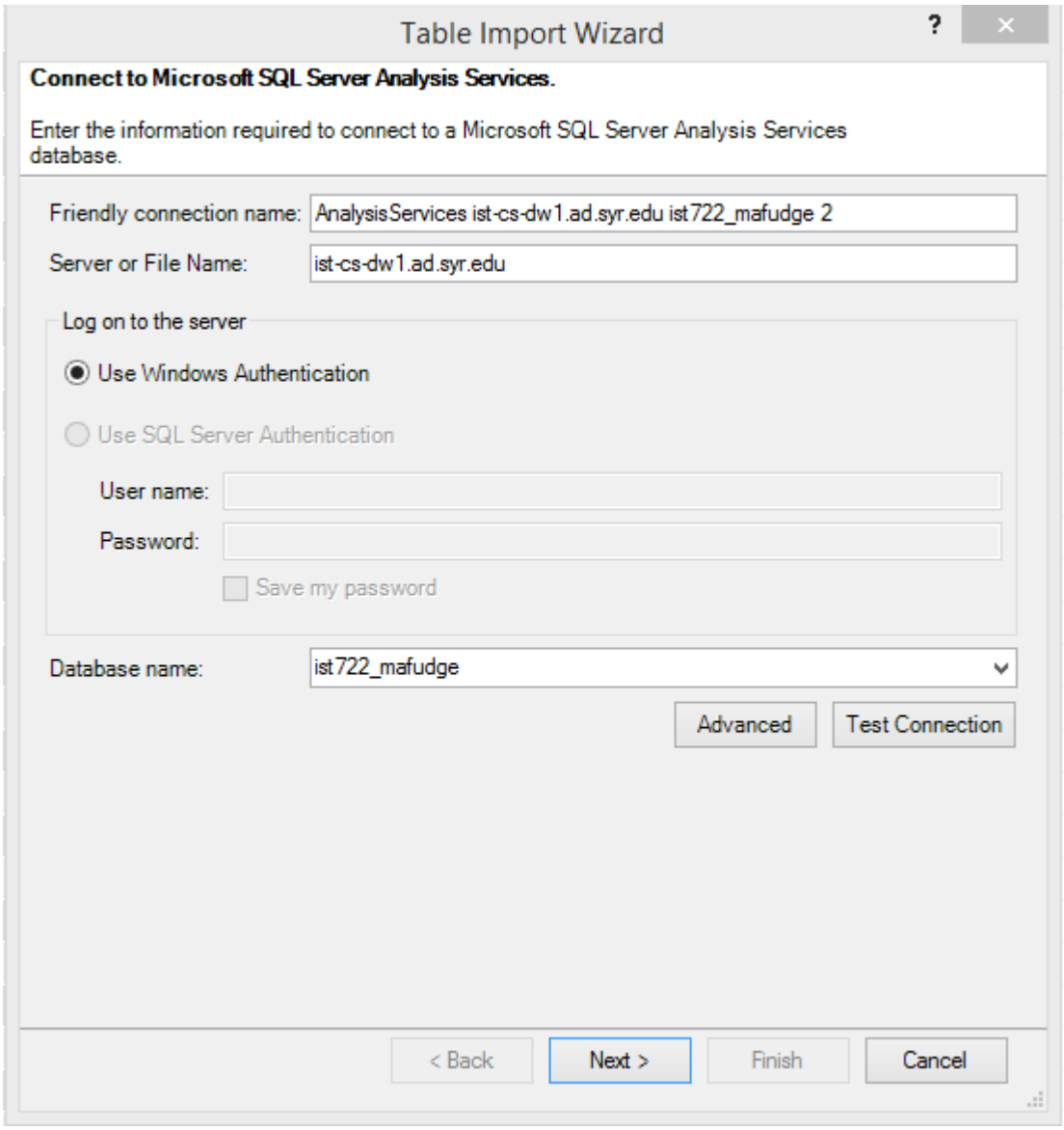
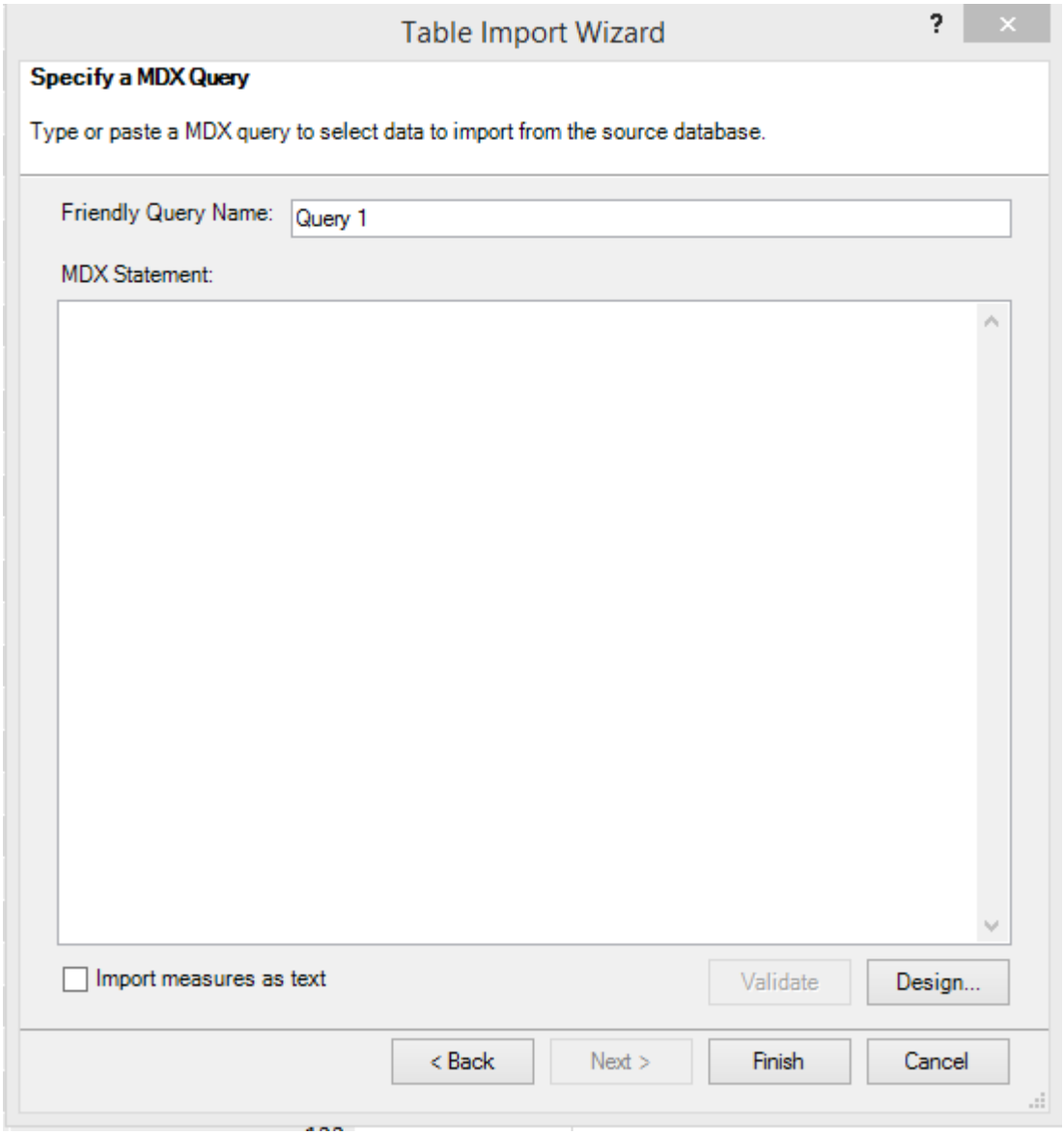


Table Import Wizard
Connect to Microsoft SQL Server Analysis Services.
Enter the information required to connect to a Microsoft SQL Server Analysis Services database.
Friendly connection name:
AnalysisServices ist-cs-dw1.ad.syr.edu ist722_mafudge 2
Server or File Name:
ist-cs-dw1.ad.syr.edu
Log on to the server
Use Windows Authentication
Use SQL Server Authentication
User name:
Password:
Save my password
Database name:
ist722_mafudge
Advanced
Test Connection
< Back
Next >
Finish
Cancel

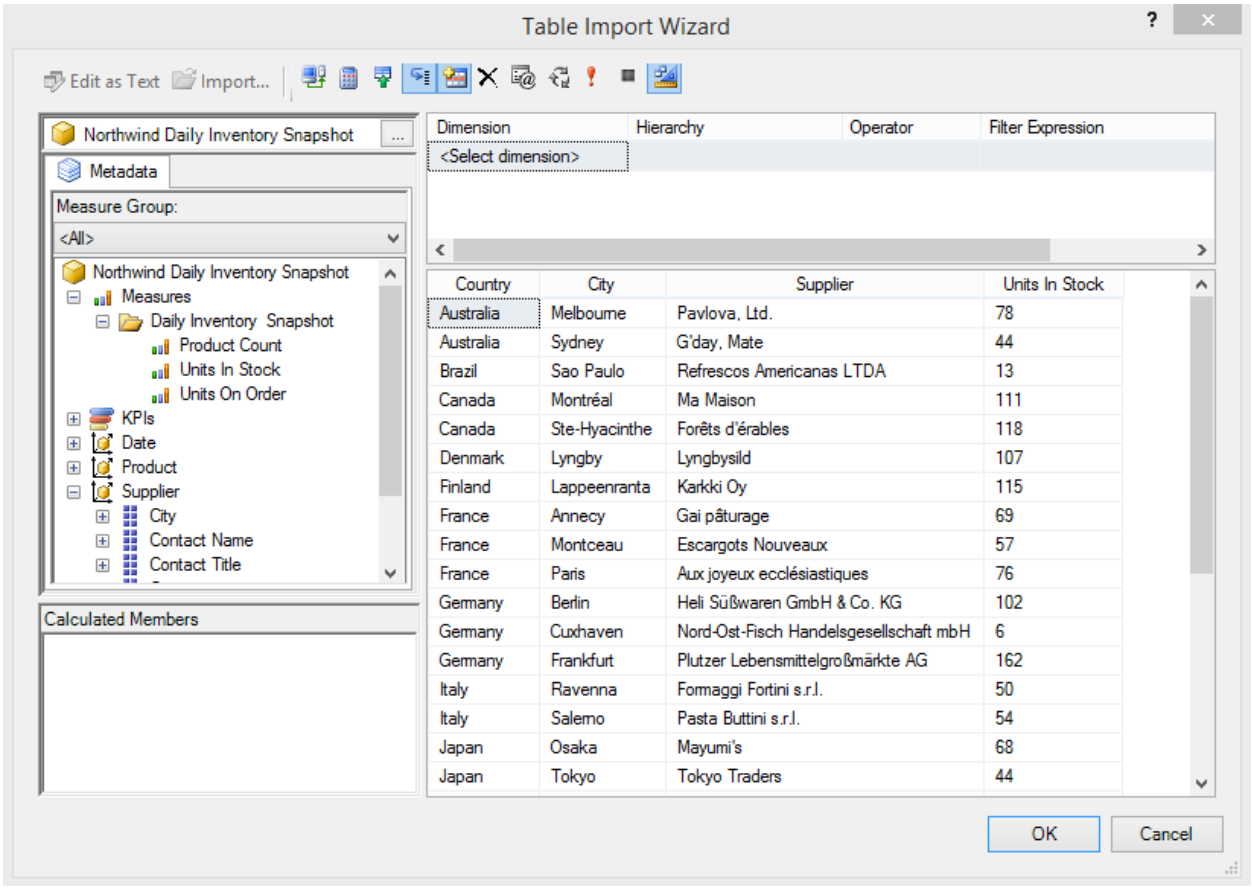
Click **Next >**.

4. You will be asked to write an **MDX** statement.



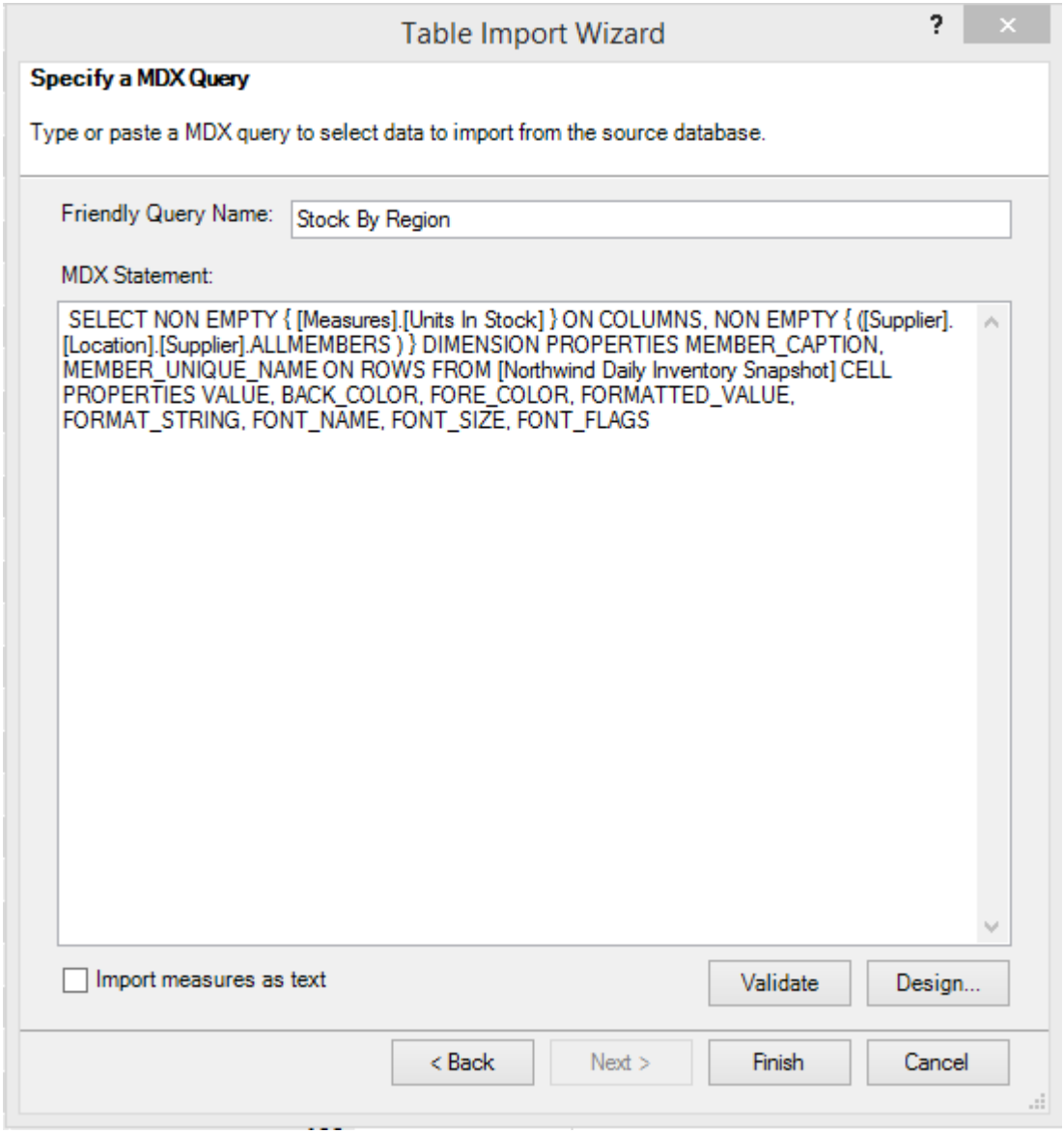
Click **Design…** to visually create the MDX query.

5. Using the Design mode, add the **Location** hierarchy to the main window, and then drag the **Units In Stock** measure to complete the query.



Click **OK** if your data matches the screenshot.

6. You will now see the MDX query text in the Import dialog.



Name your query **Stock By Region** and click **Finish** to save.

7. You should now see your data in PowerPivot. This means the data is in the Excel data model and you can close PowerPivot now and return to Excel.

8. [OPTIONAL] Let's add the Power View report: **Menu → Insert → Power View**

9. [OPTIONAL] The Power View UI looks a lot like a pivot table. Under **Stock By Region,** drag and drop **SupplierLocationCountry** and **MeasuresUnits In Stock** to the **Fields** area:

10. [OPTIONAL] This will create a table in your power view:

SupplierLocationCountry	MeasuresUnits In Stock
Australia	122
Brazil	13
Canada	229
Denmark	107
Finland	115
France	202
Germany	270
Italy	104
Japan	112
Netherlands	60
Norway	112
Singapore	45
Spain	88
Sweden	372
UK	157
USA	561

11. [OPTIONAL] Convert it to a map by clicking **Menu → Switch Visualization → Map**

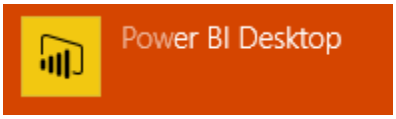
12. [OPTIONAL] Your data is now displayed on a map! Each country has a bubble representing the supplier's inventory level. The larger the bubble the higher the level.

13. **Save your Spreadsheet** as **Northwind-Daily-Inventory-Snapshot.xlsx**

Step 2.7: Power BI Desktop

In this final step we'll explore Microsoft's latest offering in the Business Intelligence tooling space: Power BI Desktop. Power BI Desktop allows you to create interactive reports, dashboards, and visualizations. Unlike Power View and Power Map, the artifacts you create with Power BI can be built and explored without Microsoft Excel and even published to Office 365, making this the future design tool for Business Intelligence. You'll find the features and user interface similar to Power View.

Get Started: Launch the Power BI desktop application from the **Start** menu.

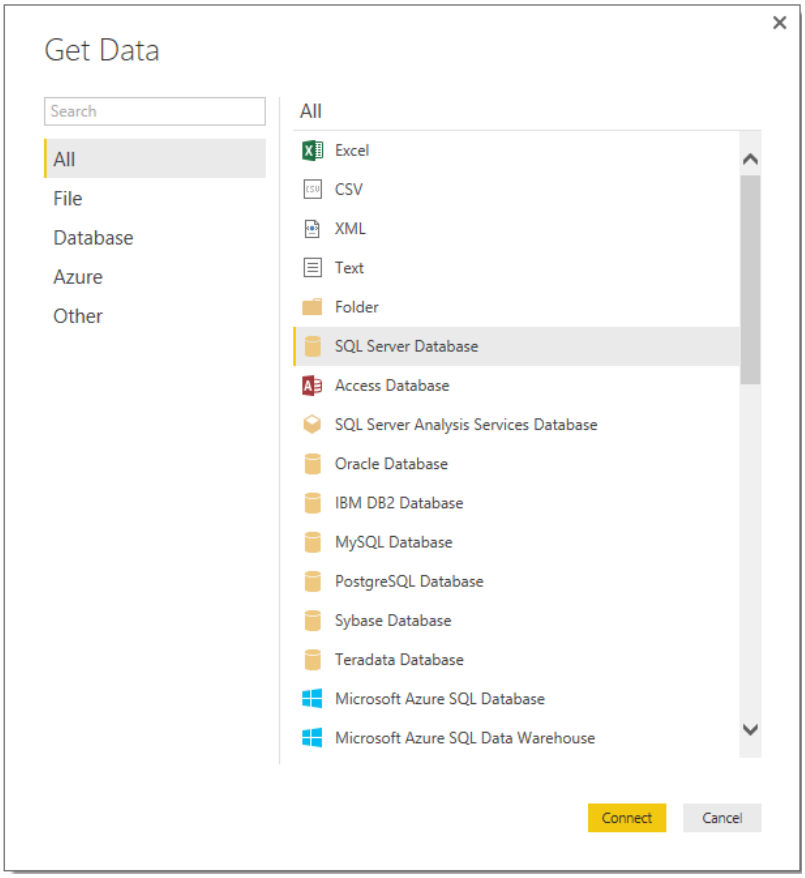


A single Power BI notebook can connect to a variety of sources, like traditional databases and cloud services. Similar to Excel PowerPivot, every source you connect into the Power BI notebook is considered a database table. Under this model, you can combine data sources from disparate sources. For example, you could pull competitor product data from a spreadsheet and join that data to a table from SQL server on a business key. In our example, we will connect a single source.

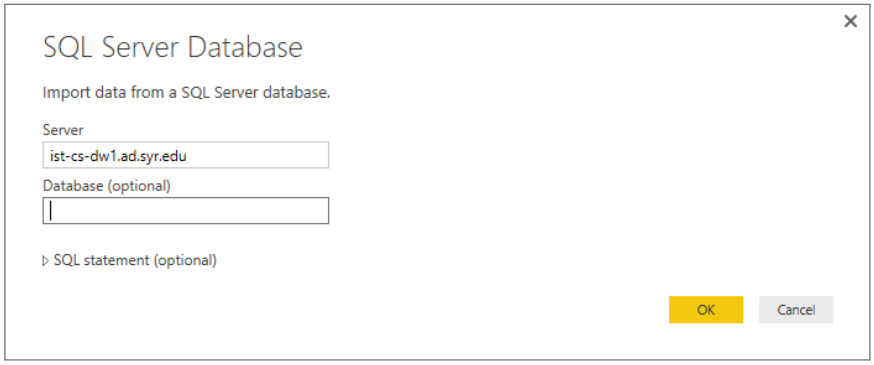
Get the ROLAP Schema for FactInventoryDailySnapshot:

First let's load our fact table for the inventory daily snapshot dimension into Power BI. We'll add the related dimensions, too. It should be noted that there are several data sources you can connect to in Power BI, and this example just chooses one such source. The rest are left as an exercise for the reader.

1. From the Power BI Menu, select **Get Data → More…**
2. Select **SQL Server Database** from the **Get Data** menu, and click **Connect**.

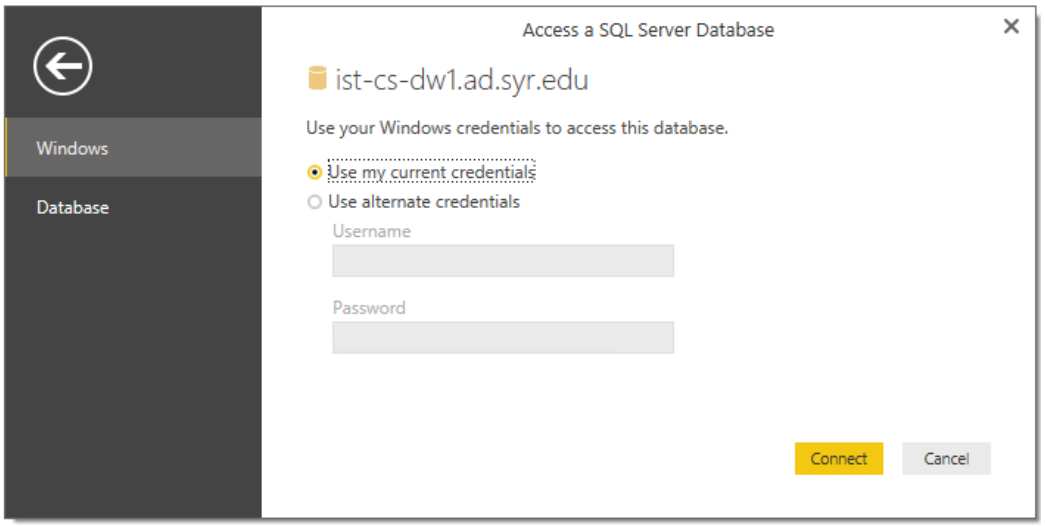


3. From the **SQL Server Database** window, connect to our data warehouse server: **ist-cs-dw1.ad.syr.edu.**



Click **OK**.

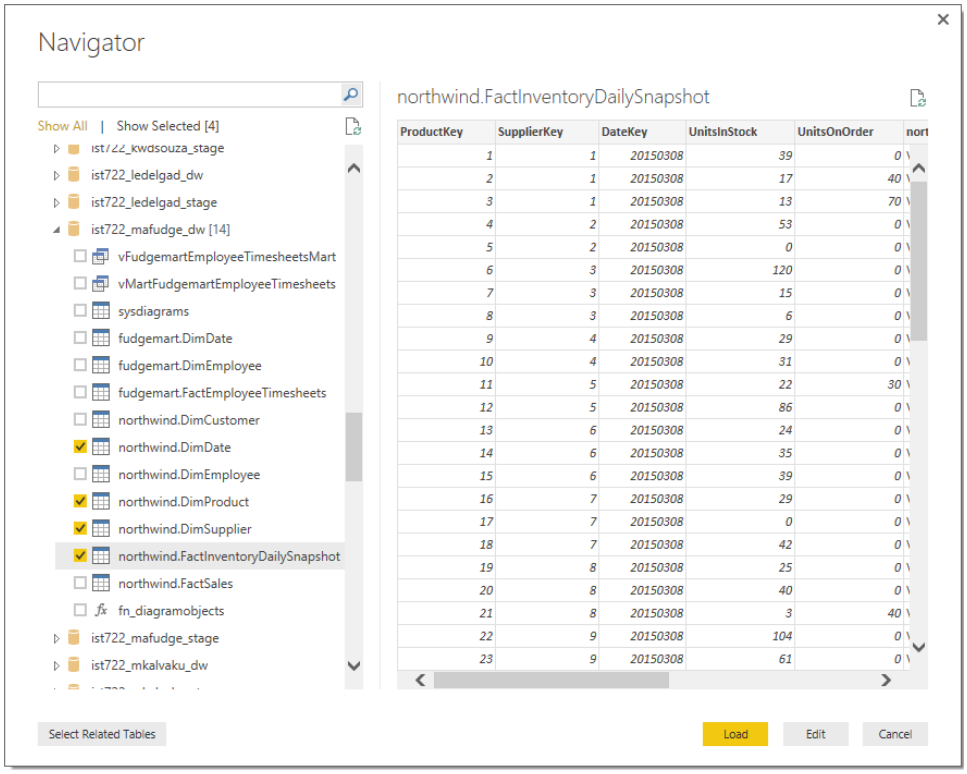
4. Next, you need to enter your credentials.
   Choose **Use my current credentials.**



   Click **Connect**

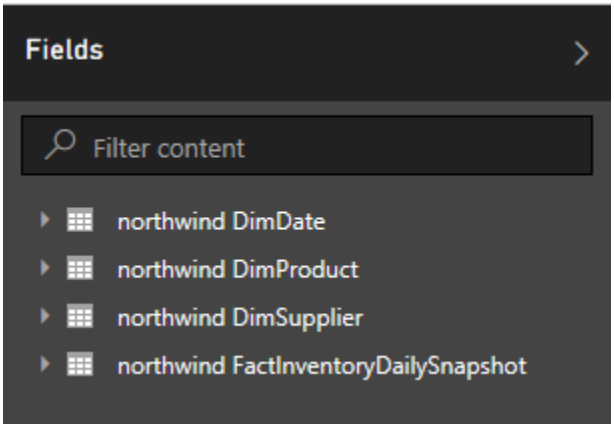
   NOTE: You might get a warning about Encryption support. Click **OK.**

5. Next you will be allowed to **Navigate** for tables to add.
   Browse for the **northwind.FactInventoryDailySnapshot** table
   in your **ist722_*yournetid*_dw** database.
   Click the **Select Related Tables** button to add the dimensions.



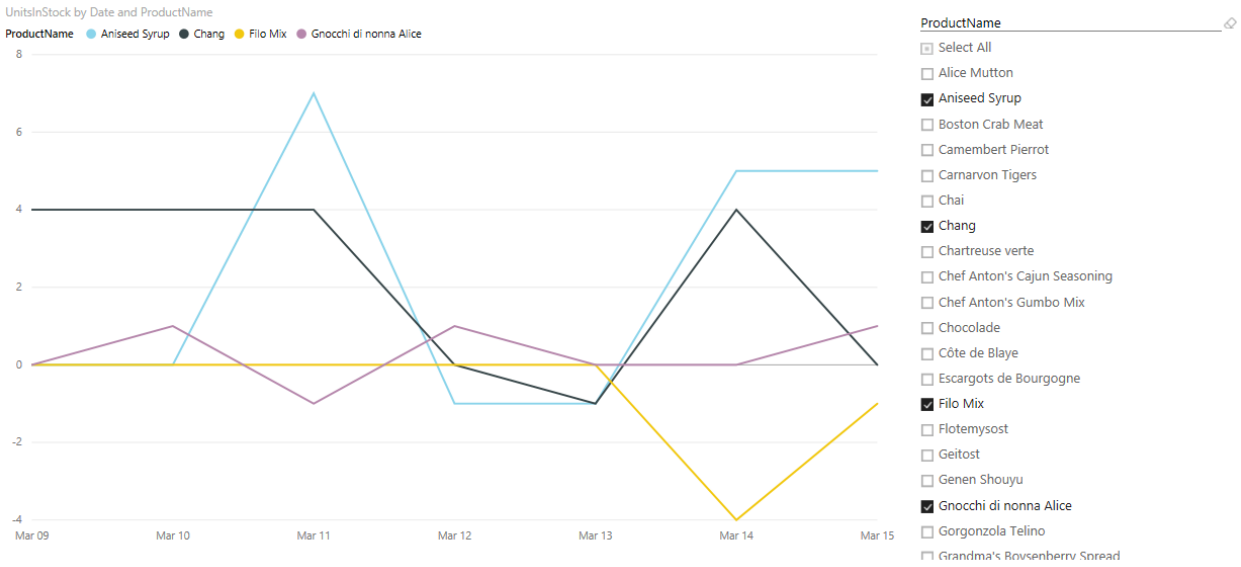
When you're ready, click **Load** to import the database tables into Power BI.
NOTE: It might take a minute or two for the tables to load into Power BI.

When it's complete, you will find the tables in the **Fields** section of the Power BI workspace.

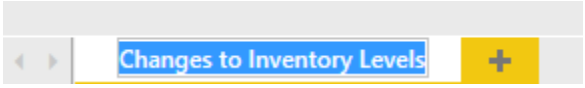


View #1: Changes to Inventory Levels

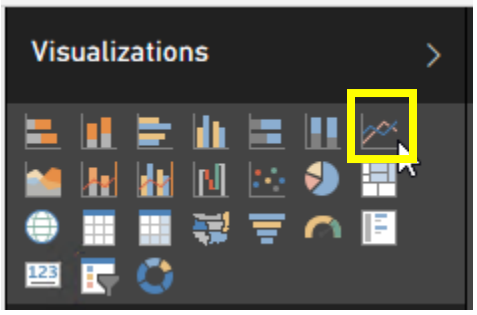
Next, let's build a dashboard to display changes to inventory levels over time by product name. For example, the finished product will look something like this:



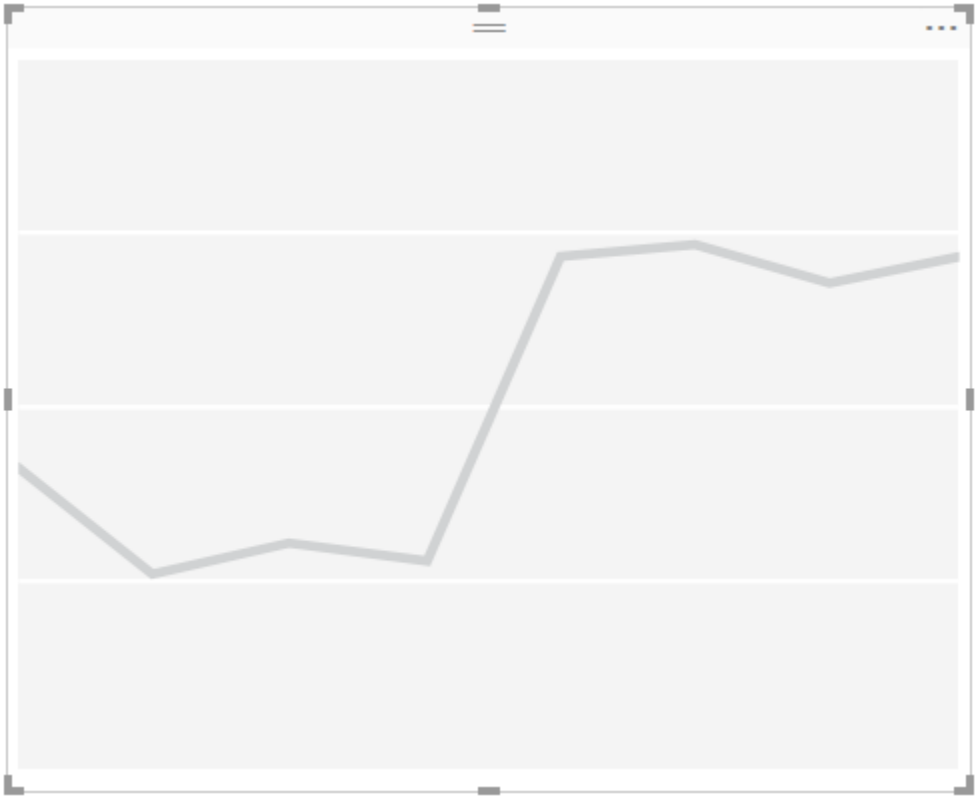
1. First let's change the workbook page to say **Changes to Inventory Levels.**
   Double-click the page tab at the bottom to rename it.



2. Next let's add a line chart visualization. Under **Visualizations** select **line chart.**

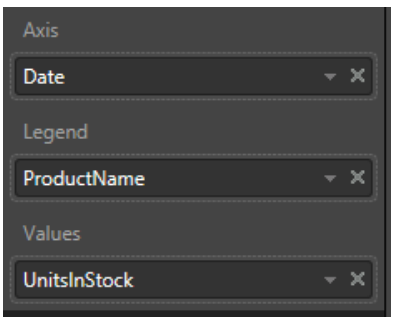


This adds a blank line chart to the page:

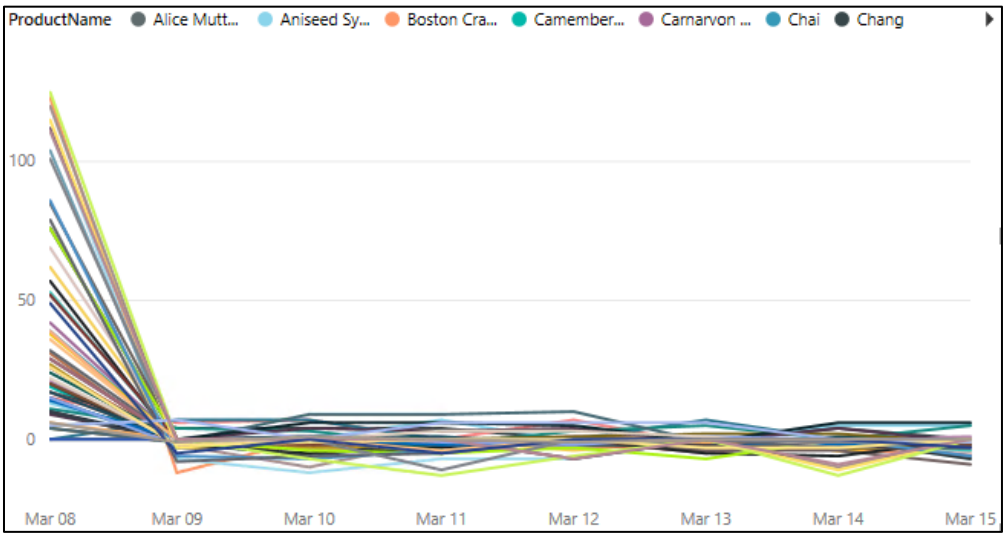


3. Set up the visualization as follows:
   Axis: **Date**
   Legend: **ProductName**
   Values: **UnitsInStock**

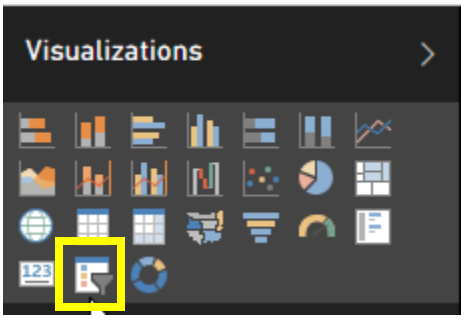
   This will produce an unreadable visualization for two reasons: 1) too many products, and 2) Initial inventory



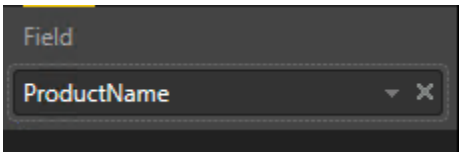
levels. We will fix both issues with filters.



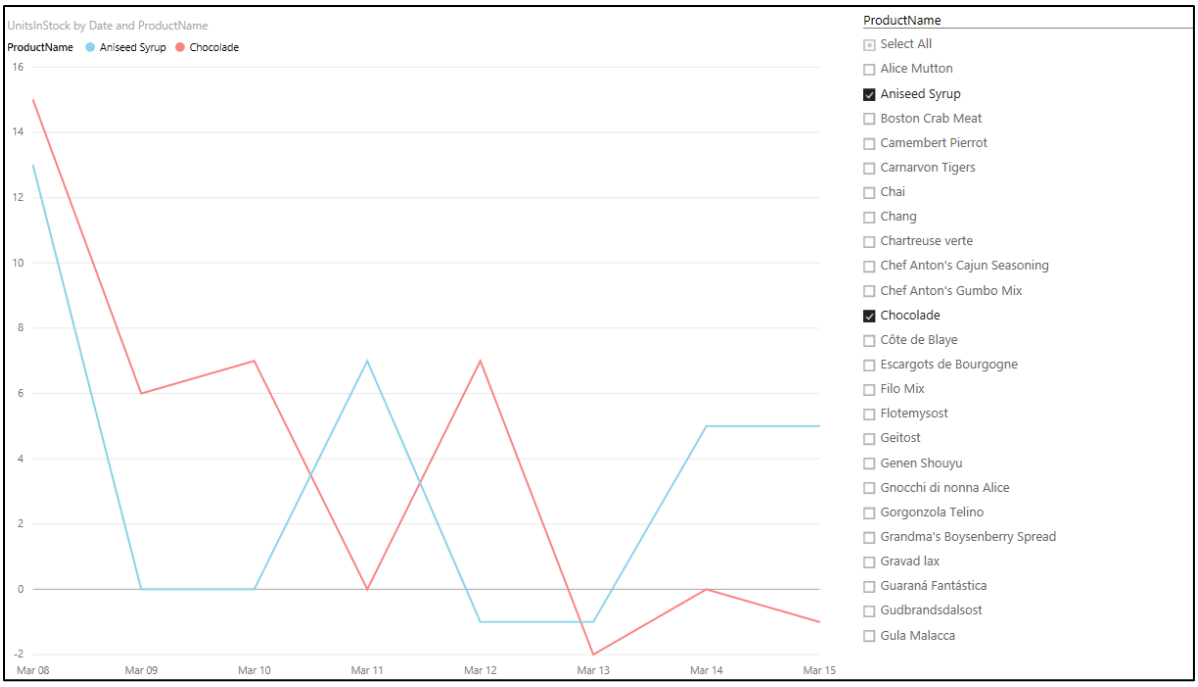
4. Let's add a **slicer** to allow us to select which products to display.
   First click off the line chart to deselect it.
5. From **Visualizations** choose **slicer.**



6.
7. Set up the slicer to display ProductName:

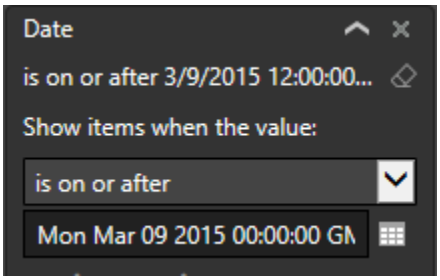


8. To verify the slicer is working, select **Chocolate** and **Aniseed Syrup**. Your line chart should look like this.

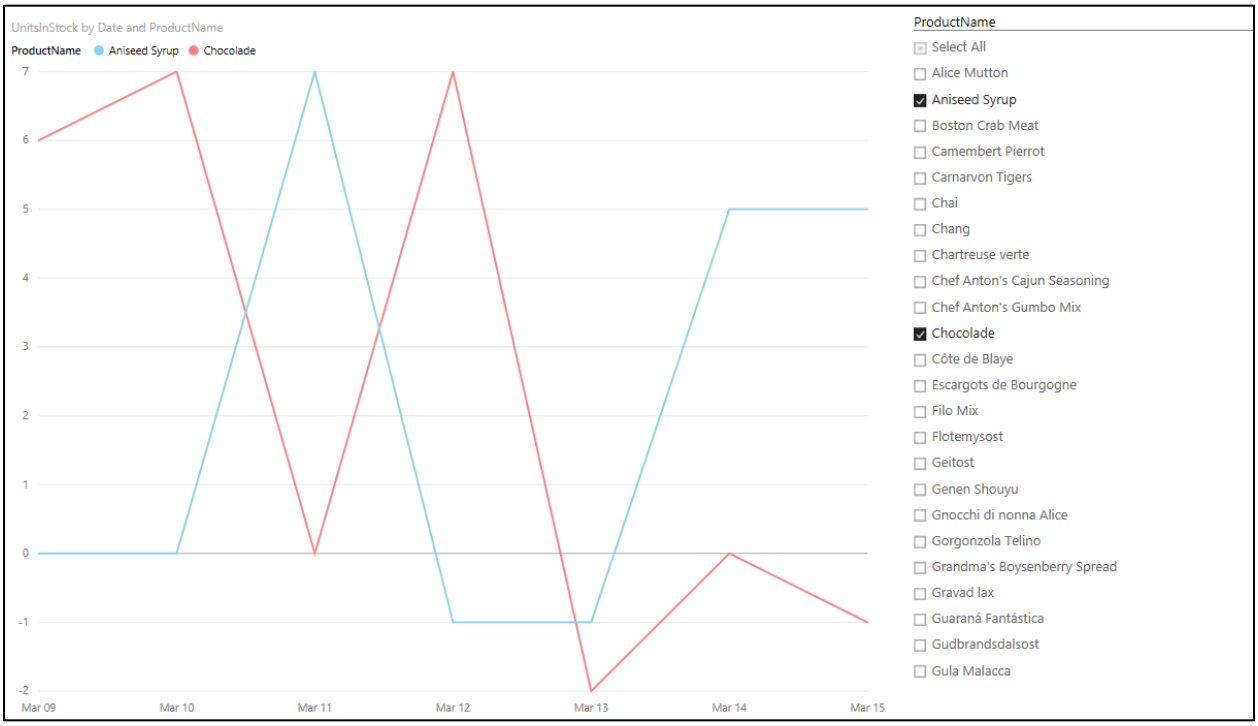


Better, but we still need to remove the March 8 initial inventory level from the chart to remove the skew of initial inventory counts prior to us tracking changes in the data warehouse. This is a common issue with periodic snapshots that add time variance to non-time variant data.

9. Let's add a page level filter to omit data before March 8, 2015.
   Place **Date** in the page level filter.
   Select **Advanced Filtering.**
   Show items where value **is on or after 3/9/2015.**
   Click **Apply Filter.**



10. Now your inventory levels' fluctuations will be represented as a variance over time:

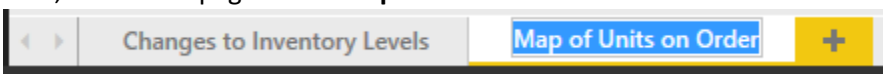


View #2: Interactive Map of Units on Order

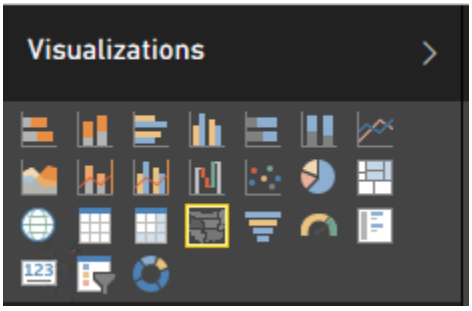
This next page will be an interactive map of units on order. When you click on a country, it will list the products on order from that company and the quantity of units on order. Here's an example with Italy selected:

To demonstrate that Power BI can use data from disparate sources, let's get the Analysis Services Cube for Daily Inventory Periodic Snapshot.

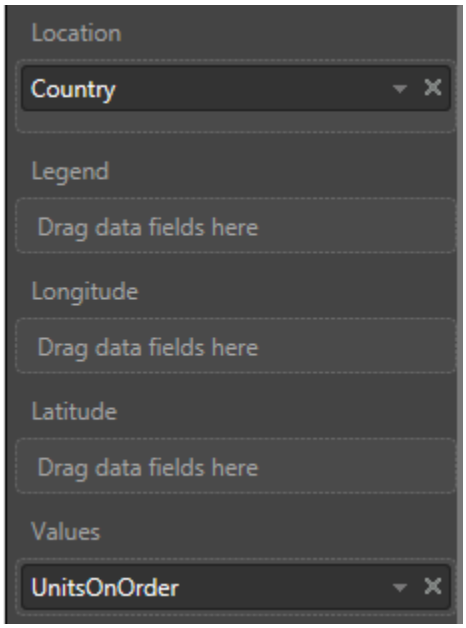
1. First, add a new page called **Map of Units on Order.**



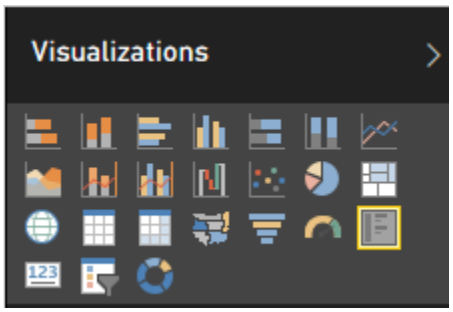
2. Next, add the **filled map visualization** to the page:



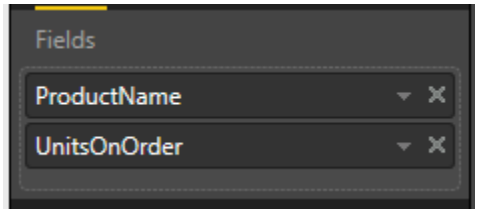
3. Set it up to use **Country** for location and **UnitsOnOrder** for values.



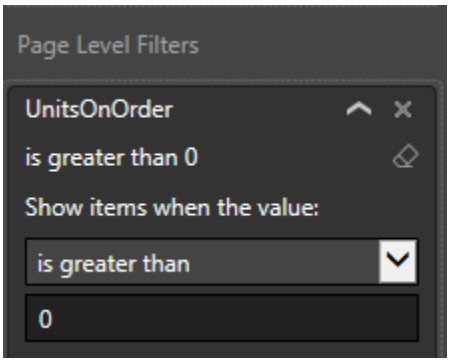
4. Add another visualization the **Multi-row Card** to the page:



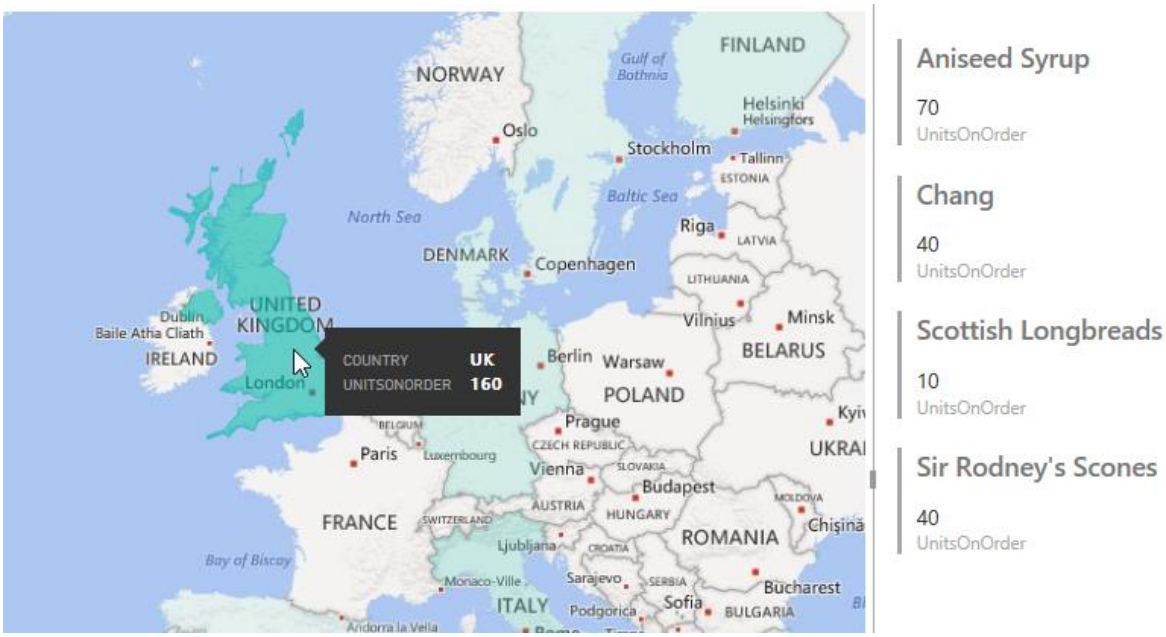
5. Add the following fields to the Card: **ProductName** and **UnitsOnOrder**



6. Finally, configure a page-level filter to only include data where **UnitsOnOrder is > 0.**



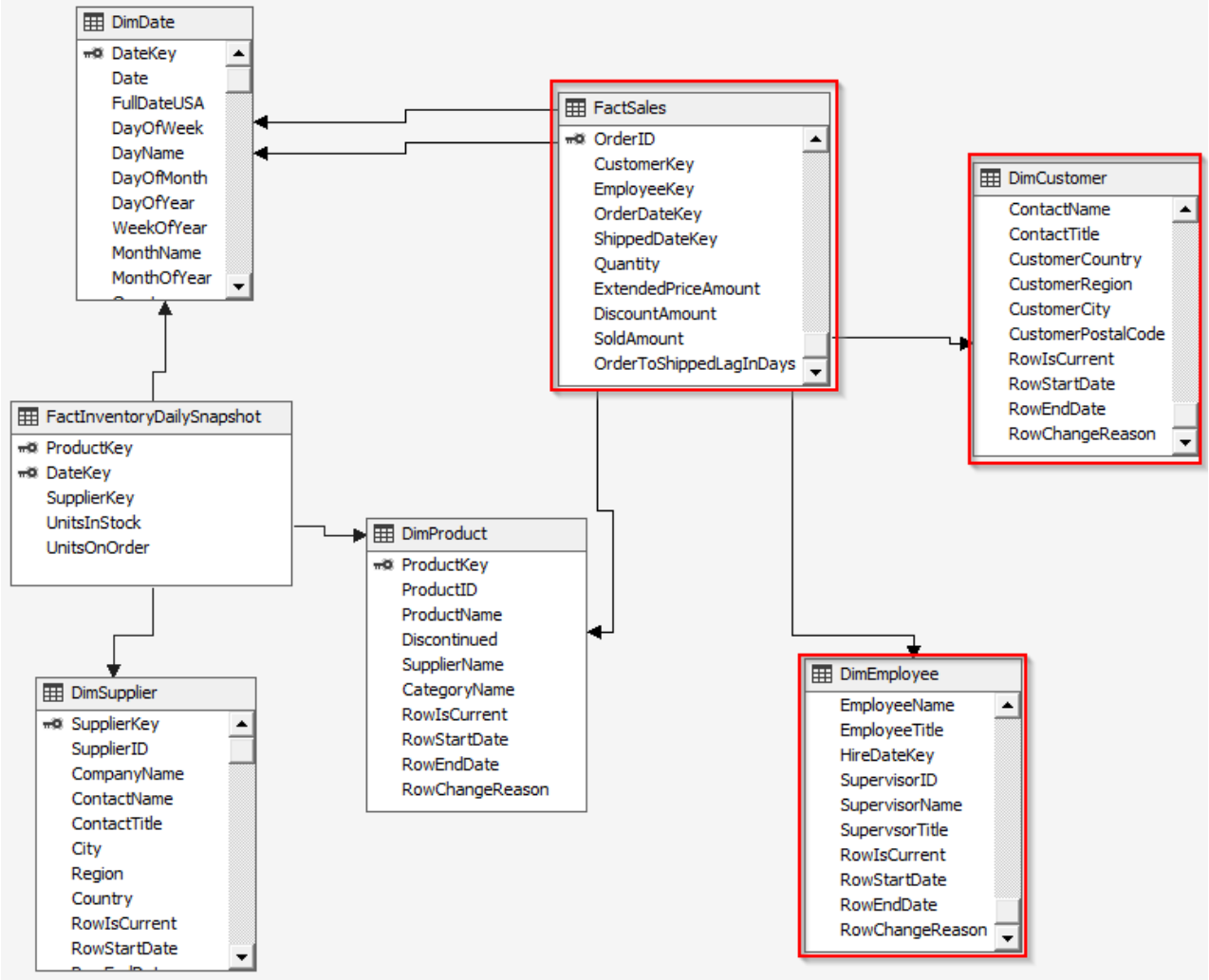
7. You should now be able to click on a country on the map and see which products and how many units are on order from that country!



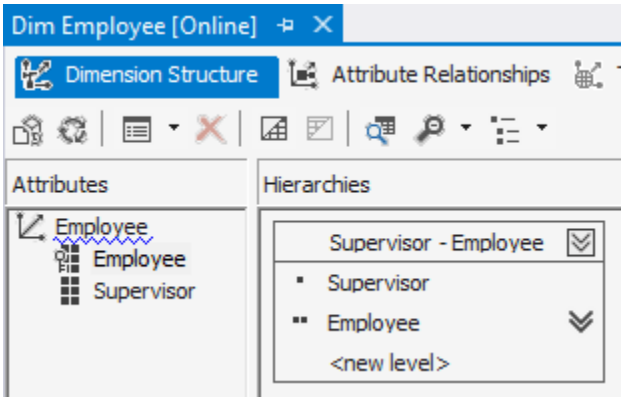
Part 3: On Your Own

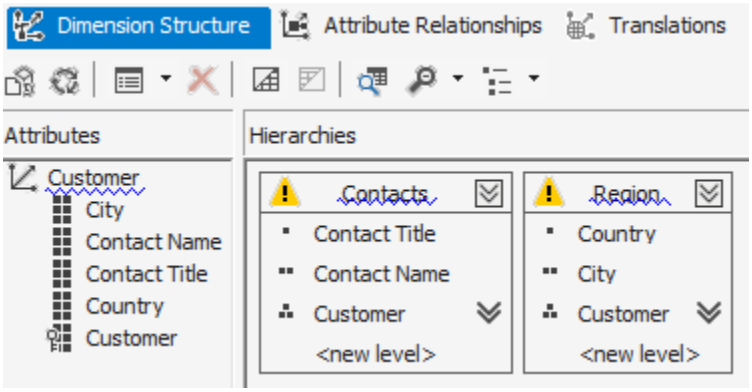
In this part, build out an SQL Server Analysis Services cube project for Northwind Sales. As you're accustomed to doing in Part 3 of assignments, you must work out many of the details on your own. After you complete the cube, create an Excel pivot table and dashboard in Power View and Power BI Desktop.

1. Add the **FactSales, DimCustomer and DimEmployee** tables to the current **Northwind** data source view:



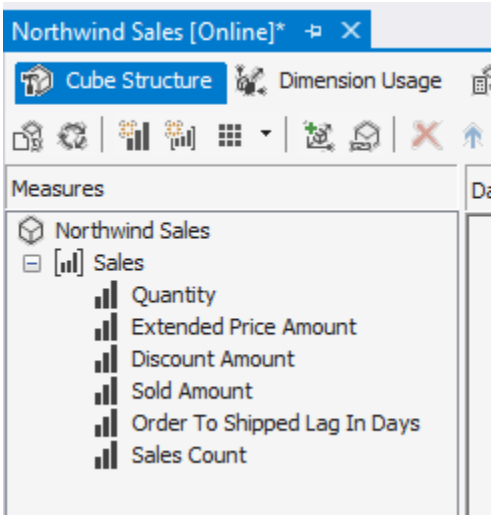
2. Build out the **Customer** and **Employee** dimensions as follows:





NOTE: **Customer** should display the company name.

3. Create the **Northwind Sales** cube as follows:



4. Create an Excel worksheet **Northwind-Sales.xlsx** with:
    a. A pivot table set up to show interesting data with drill downs – connect to ROLAP
    b. A chart of your choosing – connect to MOLAP. For example, you could choose to show sales by day of the week. Do not create a map.
5. Create a **Northwind-Sales.pibx** Power BI Desktop dashboard that is different from the view you created in Excel. Some criteria:
    a. Try using a tools not available in Excel like the map, treemap, or doughnut chart.
    b. Make it interactive so selecting portions of the chart updates another chart, matrix, or card.
    c. Include titles and instructions on the dashboard where it makes sense to do so.
    d. Create one dashboard – connect to ROLAP.
    e. Create one dashboard – connect to MOLAP.

Turning It In

Please turn in a **Word document** with your name, NetID, and date at the top. Paste **screenshots** of the following:

1. Star Schema after constructing the MOLAP (Cube)
2. Excel pivot table or chart output from Step 4a above (Get Data from SQL Server Database)
3. Excel pivot table or chart output from Step 4b above (Get Data from Analysis Services)
4. Power BI dashboard output from Step 5d above (Get Data from SQL Server Database)
5. Power BI dashboard output from Step 5e above (Get Data from Analysis Services)

You may optionally provide the **Northwind_Sales.xlsx** and the **Northwind-Sales.pibx** files.

The SQL Server Analysis Services database will be checked online to verify that your MOLAP models were created.