24 FEBRUARY 2016

Creating a Date Dimension in an Analysis Services (SSAS) Cube

> A walk-through can be a great help to understanding SQL Server Analysis Services, especially when it solves the problem of providing an effective date dimension to a cube. Dennes takes Adventureworks, and demonstrates all the twists and turns of creating a cube that provides meaningful historical information.

★★★★★ 19

7

Dennes Torres

The Date Dimension is one of the most important dimensions in an Analysis Server (SSAS) cube. Cubes need a date dimension in order to analyze historical information.

There are already many examples of date dimensions on the web, but, in this article, I think I can add my two-cents-worth of advice about creating a more complete and flexible date dimension.

I will do this by walking you through an example in all its twists and turns. I will also demonstrate many features of SSAS, along the way, that can assist you to build cube dimensions, and features needed by the date dimension.

The Example

I will build the example (see the link at the head of the article) based on AdventureWorksDW database. You can download AdventureWorksDW from http://msftdbprodsamples.codeplex.com/ but I will also show some additional fields that AdventureWorksDW doesn't have, and which you could add to your own data warehouse (DW).

Starting the Project

You can download a sample start project here. The zip file has two folders: one with the start project and another with the completed project.

The sample is a Visual Studio 2013 solution with a single multidimensional project that uses the fact table FactInternetSales and the dimensions 'Dim Promotion', 'Dim sales territory', ' Dim customer' and 'Dim product'. We could create a few more dimensions, but our objective here is to focus on the date dimension.

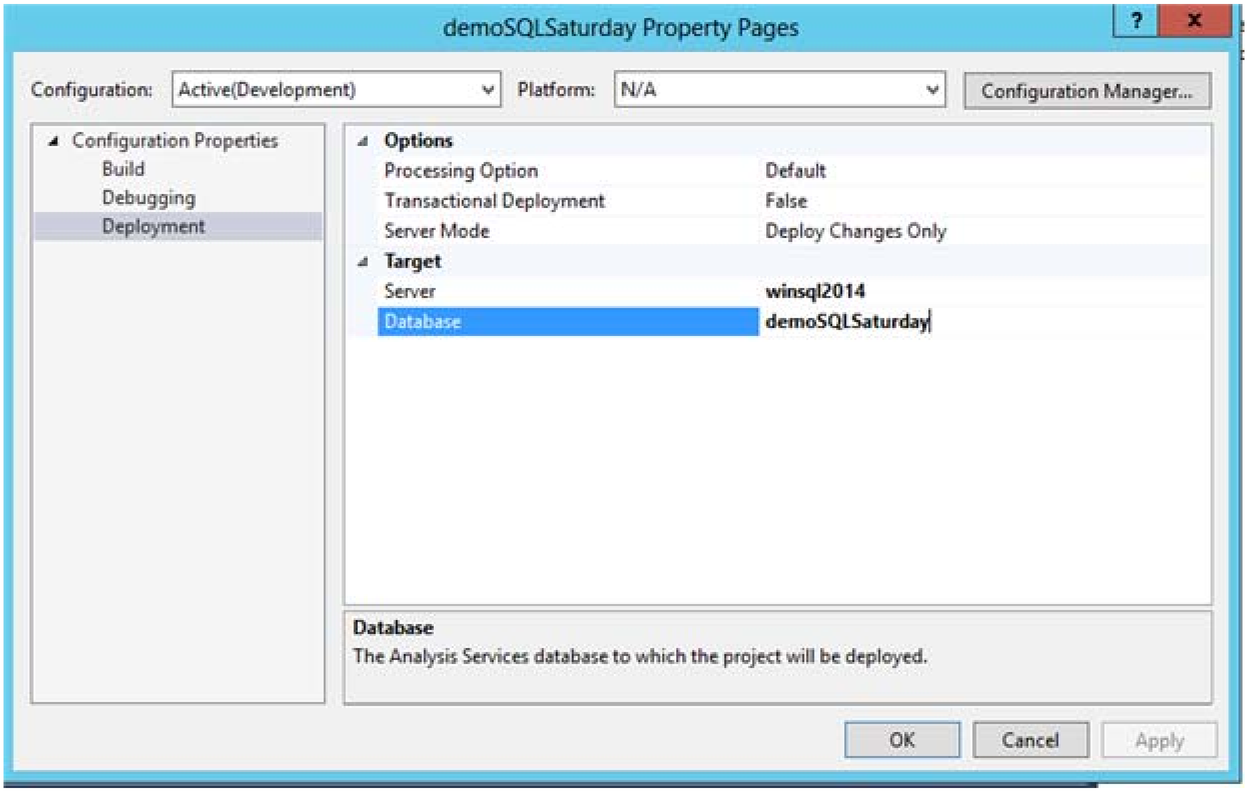
Before we start, we need to configure the project.

The project needs to be tested and deployed to a SSAS server instance installed in multidimensional mode. You need to configure the sample project, inside the solution, pointing to a SSAS instance in your environment.

Another important configuration is the Data Source View. The project has a Data Source named *AdventureWorksDS* and pointing the source database (*AdventureWorksDW*). You need to change AdventureWorksDS configuration to point to the server where you installed this database.

Firstly, let's configure the deployment server. After you have opened the solution in VS 2013, you will see the *'Solution Explorer'* window on the right side of the screen. Right-click the project in *"Solution Explorer"* and click the ' *Properties'* menu item.

In the *'demoSQLSaturday Property Pages'* window you need to configure the *'Server'* property to specify the name of your SSAS instance. The database will be created when you process the cube.



demoSQLSaturday Property Pages

Configuration: Active(Development)

Platform: N/A

Options

Processing Option	Default
Transactional Deployment	False
Server Mode	Deploy Changes Only

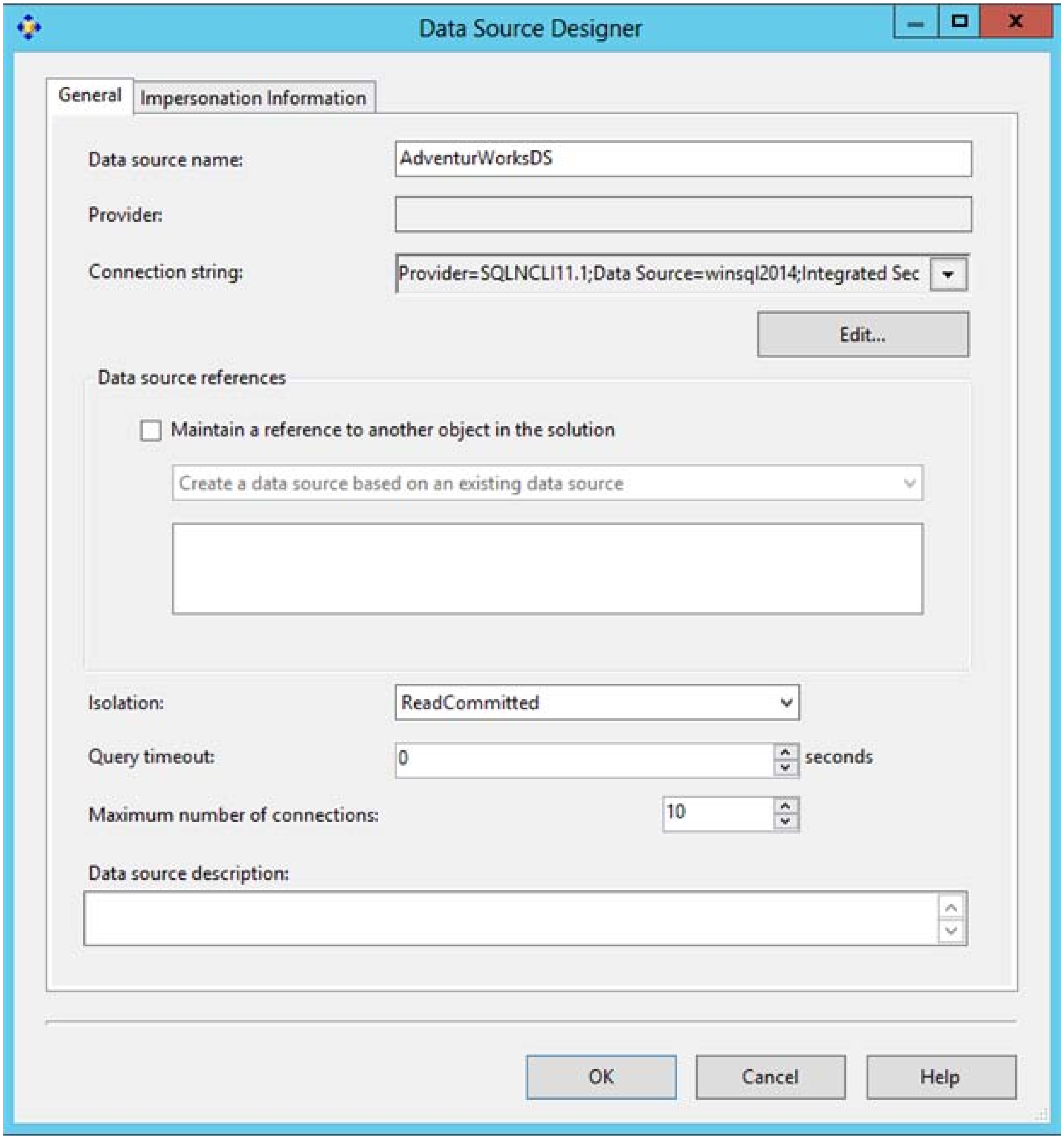
Target

Server	winsql2014
Database	demoSQLSaturday

Database
The Analysis Services database to which the project will be deployed.

Now we need to configure the data source. Follow these steps:

- Double click *'AdventureWorksDS'* inside Solution Explorer window



Data source name: AdventurWorksDS

Provider:

Connection string: Provider=SQLNCLI11.1;Data Source=winsql2014;Integrated Sec

□ Maintain a reference to another object in the solution

Create a data source based on an existing data source

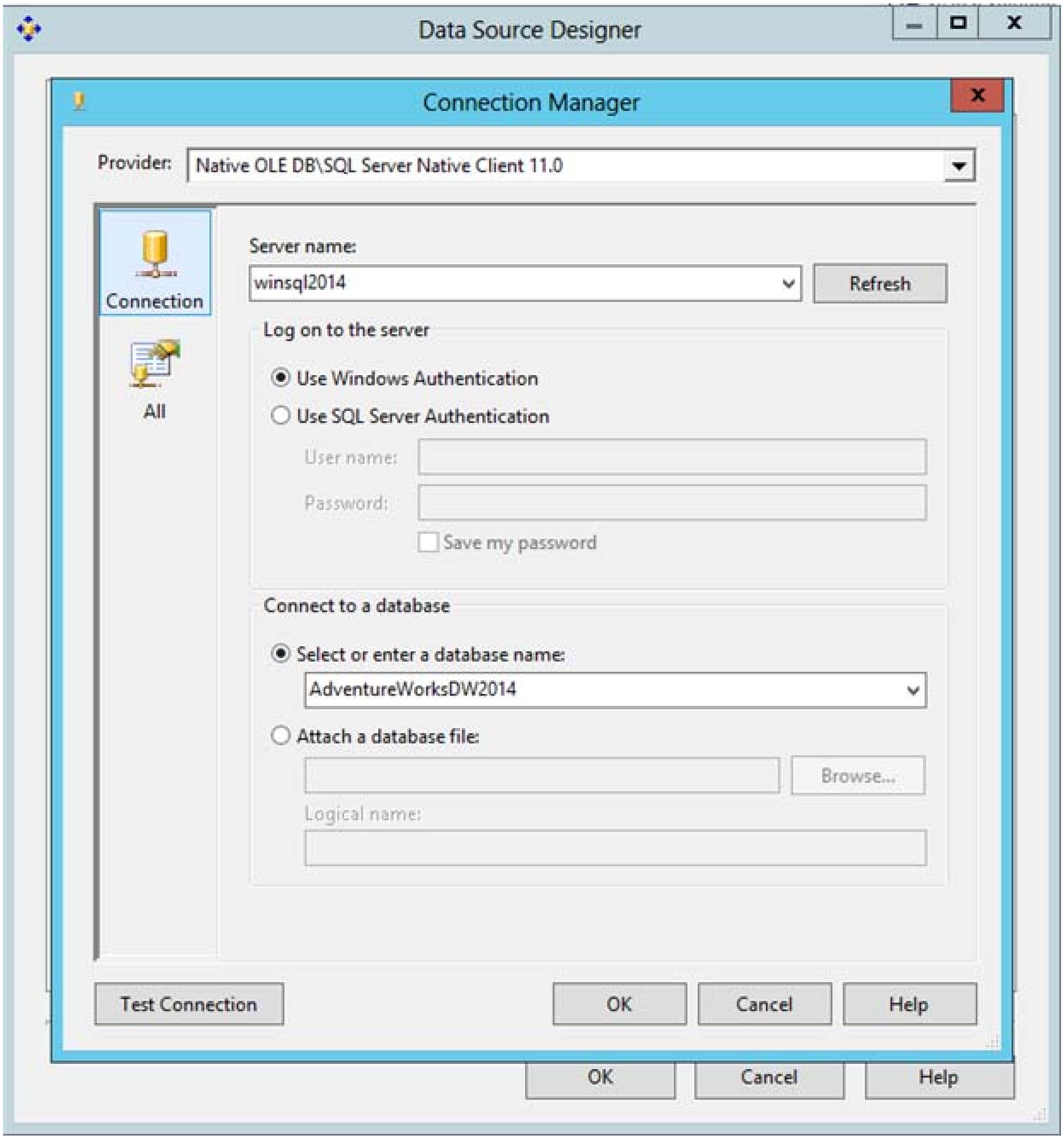
Isolation: ReadCommitted

Query timeout: 0 seconds

Maximum number of connections: 10

Data source description:

- In *‘Data Source Designer window’*, click the *‘Edit’* button to change the connection string.



Provider: Native OLE DB\SQL Server Native Client 11.0

Server name: winsql2014

● Use Windows Authentication

○ Use SQL Server Authentication

● Select or enter a database name: AdventureWorksDW2014

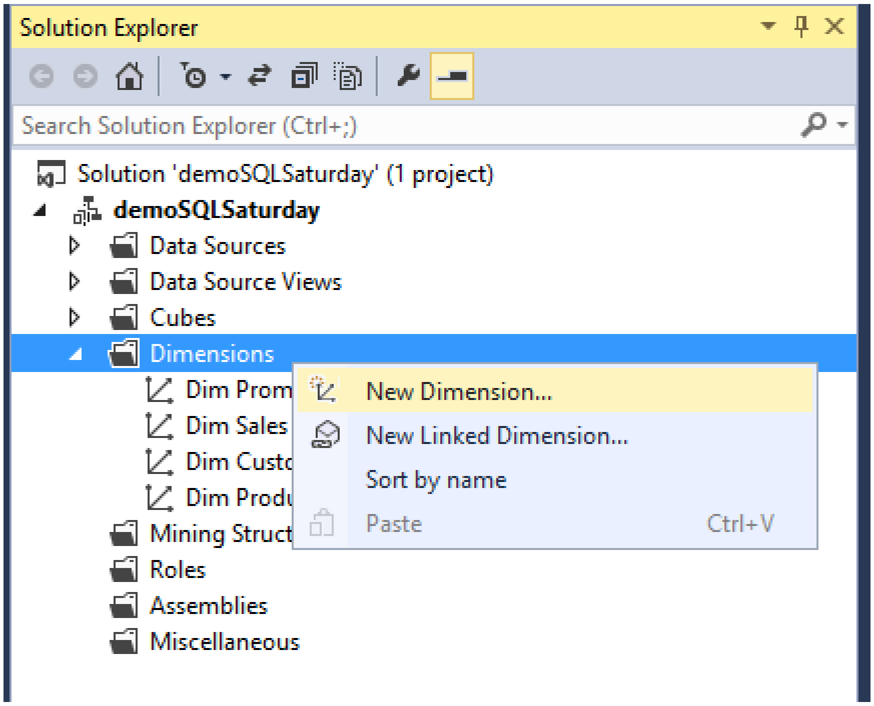
○ Attach a database file:

- In the ‘ *Connection Manager’* window, configure the server name and authentication as needed for your environment.
- In the textbox bellow *‘Select or enter a database name’*, type *‘AdventureWorksDW’*
- Click the *‘Test Connection’* button
- After the *‘Test connection succeed’* message appears, click the ‘*Ok’* button to close
- In the *‘Connection Manager’* window, click the *‘Ok’* button
- In the *‘Data Source Designer*’ window, click the *‘Ok’* button

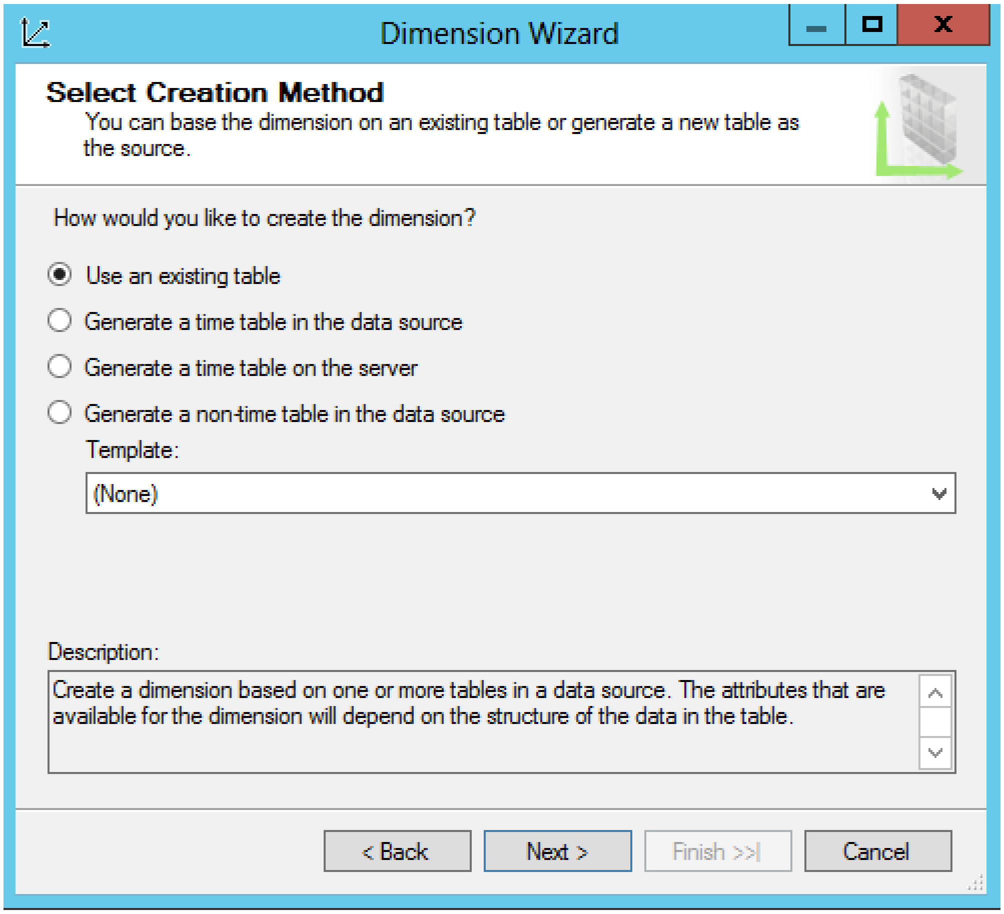
Start the creation of the date dimension

Please follow these steps:

- In Solution Explorer, right-click on the *'Dimensions'* folder in the list of folders for the sample project, left-click on the *'New Dimension…'* menu item



- On the *'Dimensions Wizard'*, Click the *'Next '* button
- Keep *'Use an existing table'* selected and click *'next'* again



Select Creation Method

You can base the dimension on an existing table or generate a new table as the source.

How would you like to create the dimension?

● Use an existing table

○ Generate a time table in the data source

○ Generate a time table on the server

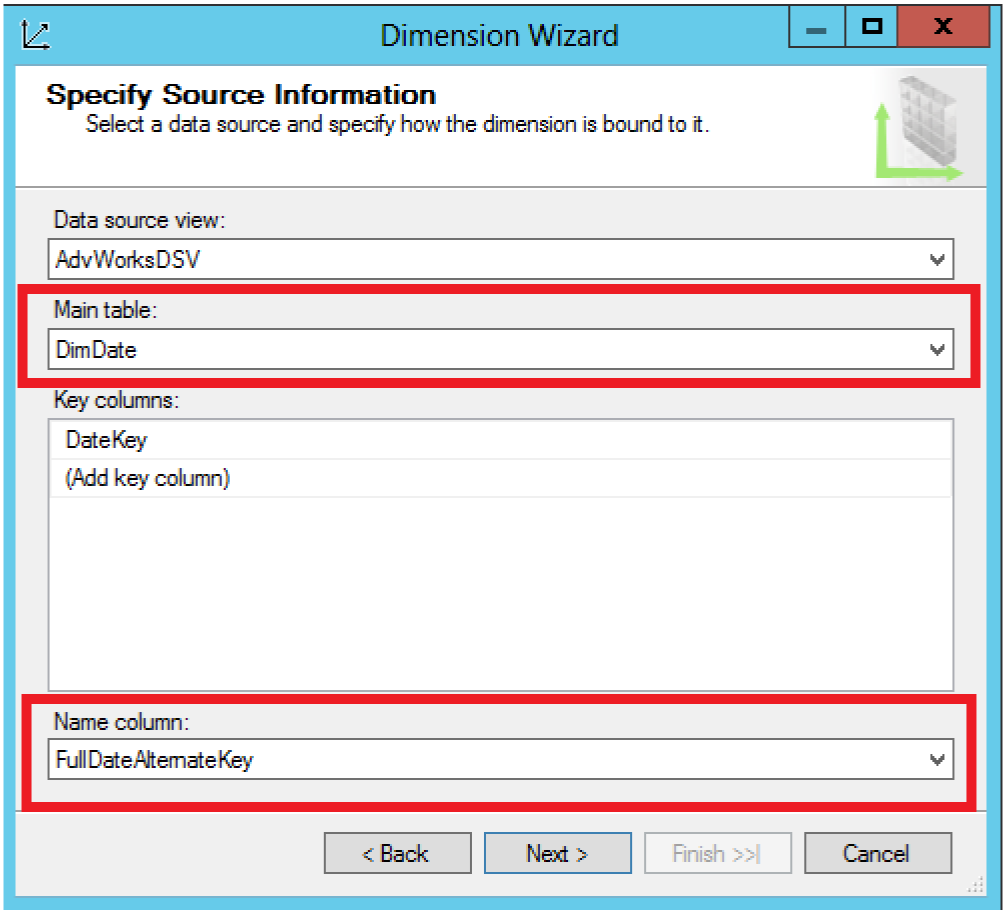
○ Generate a non-time table in the data source

Template: (None)

Description: Create a dimension based on one or more tables in a data source. The attributes that are available for the dimension will depend on the structure of the data in the table.

- In the *'Main table'* combobox, select *'Dim Date'*
- keep *'Key Columns'* unchanged *('Date Key'* is already selected)
- In *'Name Column'* combobox select *'FullDateAlternateKey'*

This means that *'Date Key'* will be the Key Column that will identify each value of this attribute (the first attribute of the dimension) while *'FullDateAlternateKey'* will be displayed for this attribute, instead of *'Date Key'*.



Dimension Wizard

Specify Source Information

Select a data source and specify how the dimension is bound to it.

Data source view: AdvWorksDSV

Main table: DimDate

Key columns:

DateKey

(Add key column)

Name column: FullDateAlternateKey

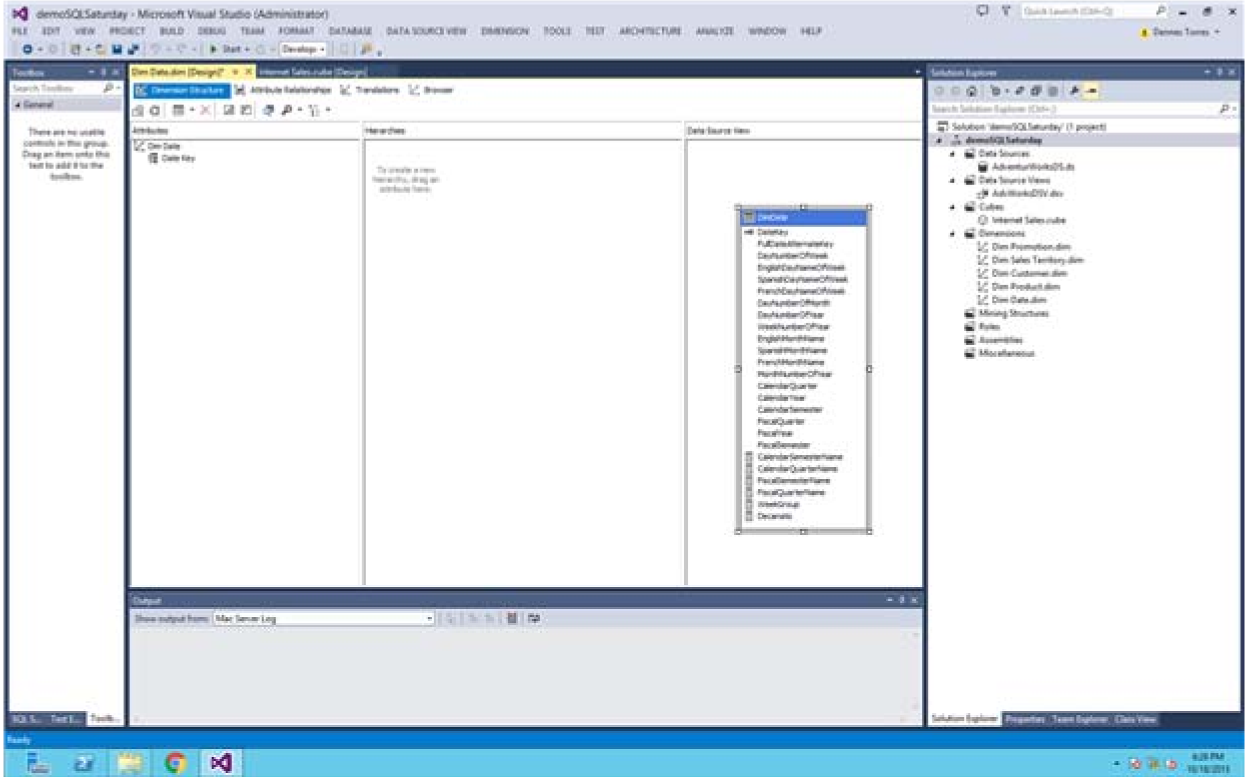
- Click *'Next'*; and click *'Next'* again on the *'Select Dimension Attributes'* window. We will choose the attributes more carefully during this article.
- Check the name of the dimension, *'Dim Date'*, and click 'Finish' to close the dimension wizard

Now you have the date dimension and the dimension editor open, let's start editing the dimension. We have the following areas in the editor:

Attributes: We will add and configure attributes in this area. This area will keep the collection of attributes we create for this dimension.

Hierarchies: We will create custom hierarchies in this area

Data Source View: We will get the fields from this area and drag to the *'Attributes'* area to create attributes for our dimension



Hierarchies

The date dimension allows us to aggregate the fact information by month, year, semester and many other options. In our example, the fact information is the amount of sales, so we can see the total of sales by month, semesters and many other options.

These aggregation options are hierarchical. We can aggregate by year and month, for example, creating a hierarchy.

When using Excel to access the cube, the users can, of course, build their own hierarchies by dragging individual attributes to a pivot table, one by one. Besides being a boring, hard-working task for the user, this can also lead to errors if the user drag the attributes in an order that makes no sense.

Our objective while designing the dimension is to make things easier for the user by creating useful pre-defined hierarchies.

Firstly, we must decide what hierarchies will we create and how to configure them.

You will notice that, for the date dimension, a single hierarchy isn't enough. For example, we cannot mix the fiscal year with calendar year. We also cannot mix month with weeks. We will need several hierarchies in our date dimension.

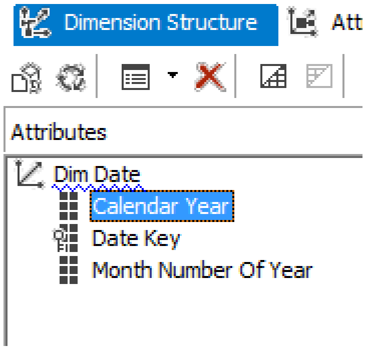
Calendar Hierarchies

Let’s start with calendar hierarchies, following the traditional calendar. We can have a simple hierarchy such as Year/Month/Day or we can have a more complete hierarchy such as Year/Semester/Quarter/Month/Day.

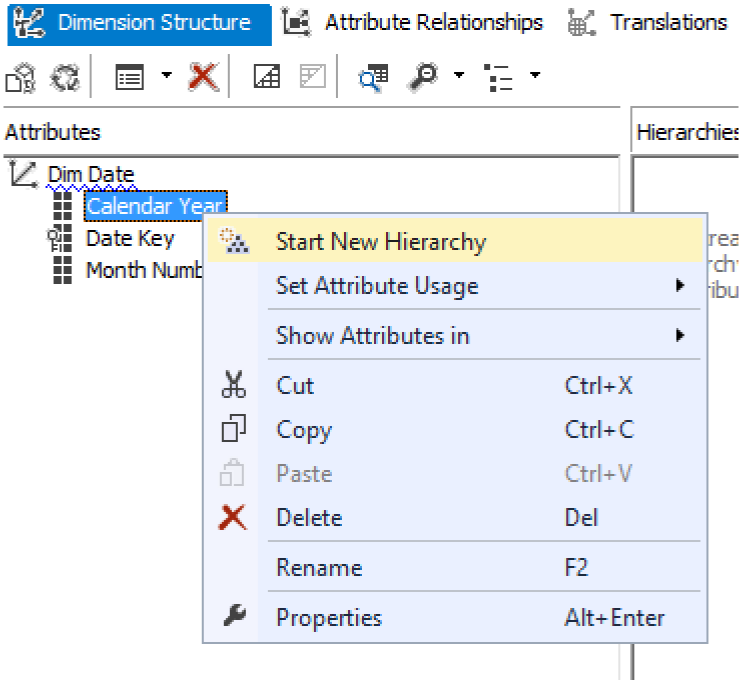
We can create both options as two different hierarchies and allow the user to choose the best for each situation, or even use perspectives to choose which user will see which hierarchy. We will talk about perspectives later in this article.

Let’s do a step-by-step walkthrough to create the ‘Simple Calendar ‘ Hierarchy:

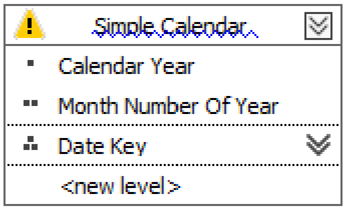
- Drag the field *‘CalendarYear’* from *‘Data Source View’* to *‘Attributes’* area.
- Drag the field *‘MonthNumberofYear’* from *‘Data Source View* to *‘Attributes’* area



- Right-click *‘Calendar Year’* in the attributes area and click on the ‘Start New Hierarchy ‘menu item. A new hierarchy will be created in the *‘Hierarchies’* area



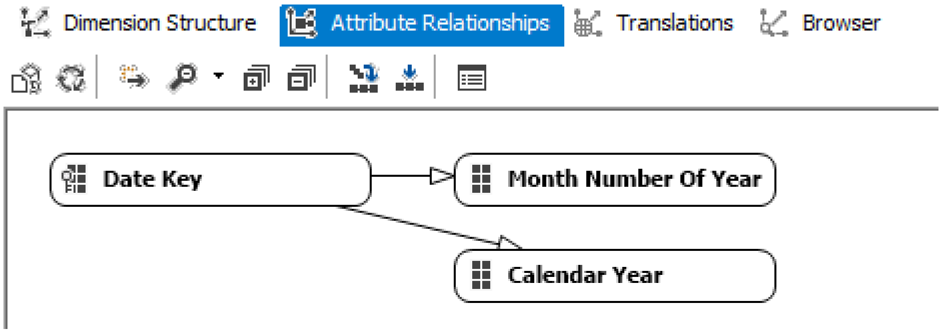
- Right-click the hierarchy you just created and rename it to *'Simple Calendar'*
- Drag the attribute *'Month Number of Year'* to the hierarchy, under ' *Calendar Year'* attribute.
- Drag the attribute *'Date Key'* to the hierarchy, under *'Month Number of Year'* attribute.



At this point, you will notice an alert beside the hierarchy name. Pointing to the alert, the message explains to us the problem: We need to configure the attribute relationships to optimize the dimension processing.

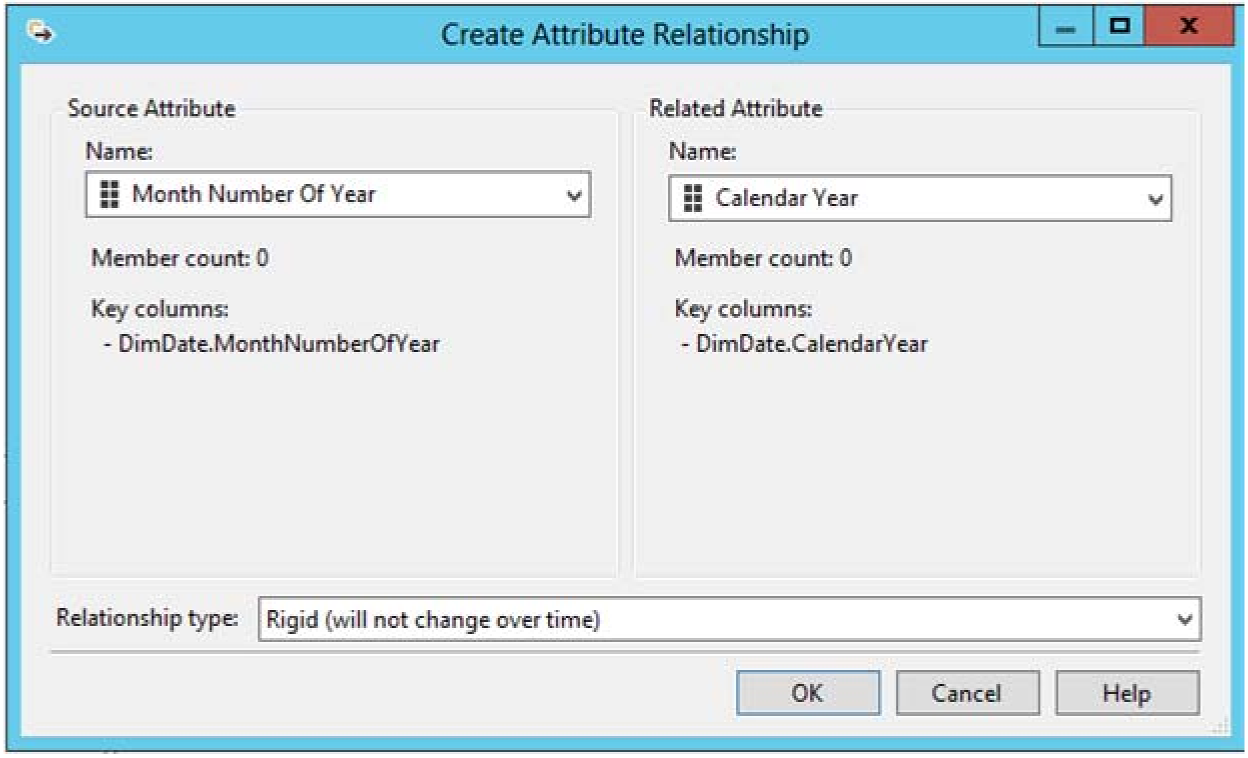
Before the user can use our cube, SSAS will process the cube and the dimensions on it, pre-aggregating the information and storing aggregations in the server. Some configurations, like the attribute relationships, can optimize the dimension processing.

- Click the *'Attribute Relationships'* tab



- Right-click *'Month Number of Year'* and click the *'New Relationship'* menu item
- In *'Create Attribute Relationship'* window, check if *'Calendar Year'* is selected in the *'Related Attribute'* combobox
- Select *'Rigid'* as the relationship type.
- Click *'Ok'* button

For each hierarchy that we create, we will need to fix the attribute relationship again.



Create Attribute Relationship

Source Attribute

Name: Month Number Of Year

Member count: 0

Key columns:
- DimDate.MonthNumberOfYear

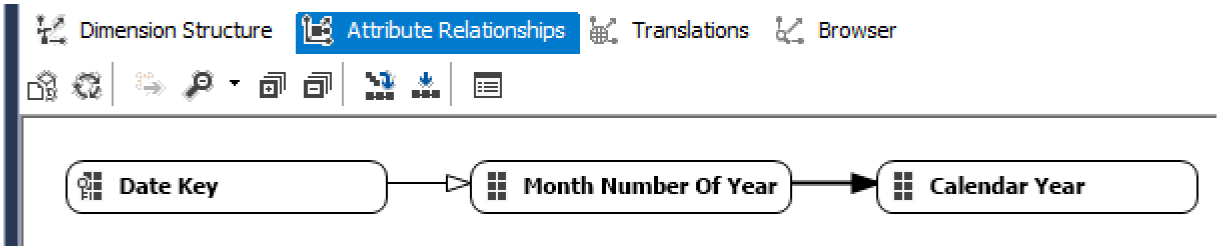
Related Attribute

Name: Calendar Year

Member count: 0

Key columns:
- DimDate.CalendarYear

Relationship type: Rigid (will not change over time)



The relationship type is the answer for the question: “Can the month be changed to belong to another year?” If the answer is “yes”, the relationship needs to be flexible. If the answer is “no”, the relationship can be rigid. A rigid relationship offers better processing time than a flexible relationship because SSAS will not try to find relations that are already well-known.

For example, let’s take the relation between employee and department. This is easy to figure out because an employee can change to another department during his work in the company, so this relation should be flexible.

On the other hand, how can one answer that same question for the relation between ‘month‘ and ‘year’? January exists in every year, so what’s the answer?

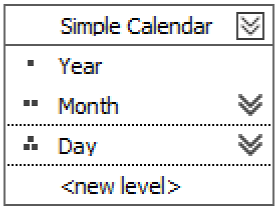
The problem is that a hierarchy member can‘t repeat across the hierarchy. Month either can’t repeat every year or else can’t be in the hierarchy. Because of that, we need to change our attribute just a little, so that the key attribute isn’t only the month but the month/year. This change makes the answer a lot easier: January/2014 is always part of the 2014 year and will never be changed to the 2015 year. The answer is “rigid”, as it will be for most relationships in the date hierarchy.

Let ‘s continue with a few more steps to see what happens if we choose the relationship type to *“rigid”* without changing the key of the attribute.

Some customizations and the first test

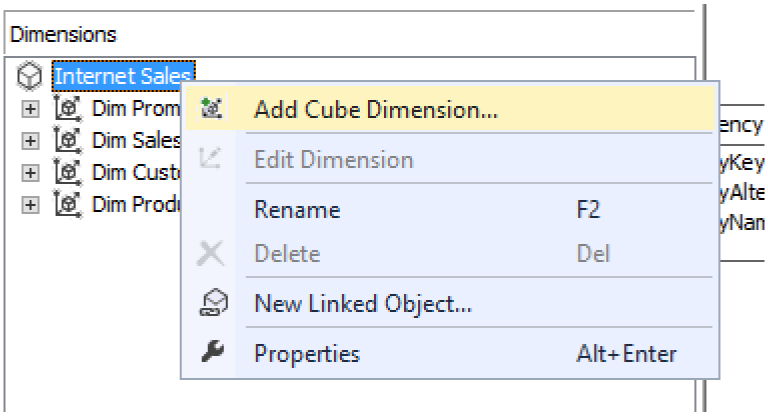
Now we will customize the dimension, making it simpler for the user:

- Return to the dimension structure tab
- In the *‘simple calendar’* hierarchy, right-click the *‘Calendar Year’* attribute and rename it to ‘Year‘
- In the *‘simple calendar’* hierarchy, right-click *‘Month Number of Year’* and rename to ‘Month‘
- In the *‘simple calendar’* hierarchy, right-click *‘Date Key’* and rename it to *‘Day’*



With this complete, we ‘ll test our new dimension, and we will find and correct a few problems. We need to add the dimension to the cube, process the cube and test it.

- If the tab *‘Internet sales.cube’* is not already opened, open it double-clicking *‘Internet sales.cube’* in the *‘Cubes’* folder of the *‘Solution Explorer’* window
- Select the *‘Internet sales.cube’* tab
- In *‘Dimensions’*, right-click *‘Internet Sales’*, click *‘add cube dimension’*

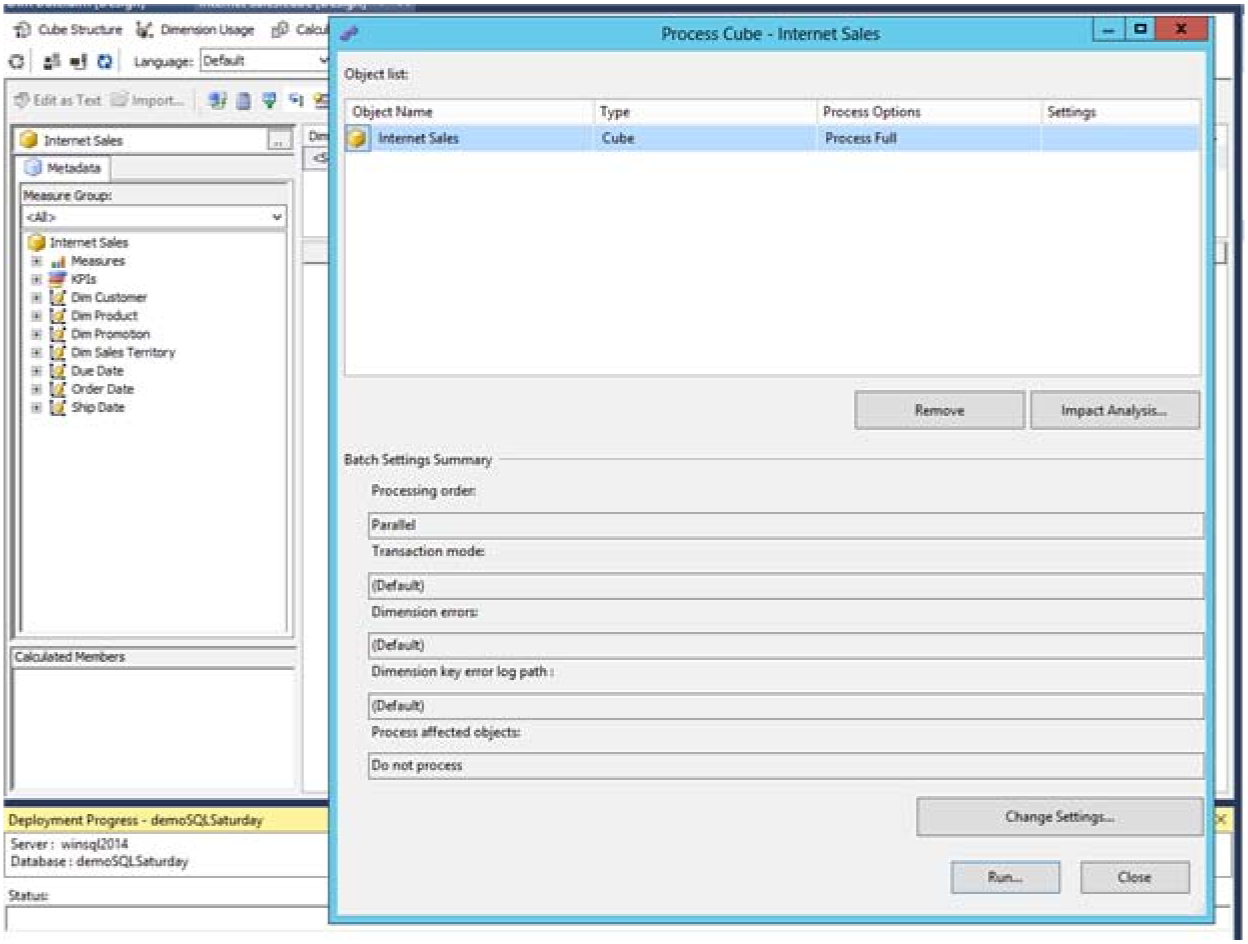


- Select *‘Dim Date’* and click *‘OK’*

You will notice that not just one but three dimensions are added to the cube. This happens because there are three relations between the fact table and the date dimension table. This feature is called *‘role playing* *dimension'*. Any change to the date dimension will affect all three dimensions added to the cube.

Now we will process the cube and test it in Excel. You will repeat these steps several times during this article.

- Click *'Browser'* tab
- Click *'Process'* button
- Click *'Yes'* to confirm when needed
- Type the administrator password for your server



Process Cube - Internet Sales

Object list:

Object Name	Type	Process Options	Settings
Internet Sales	Cube	Process Full	

Batch Settings Summary

Processing order: Parallel

Transaction mode: (Default)

Dimension errors: (Default)

Dimension key error log path : (Default)

Process affected objects: Do not process

Deployment Progress - demoSQLSaturday

Server : winsql2014
Database : demoSQLSaturday

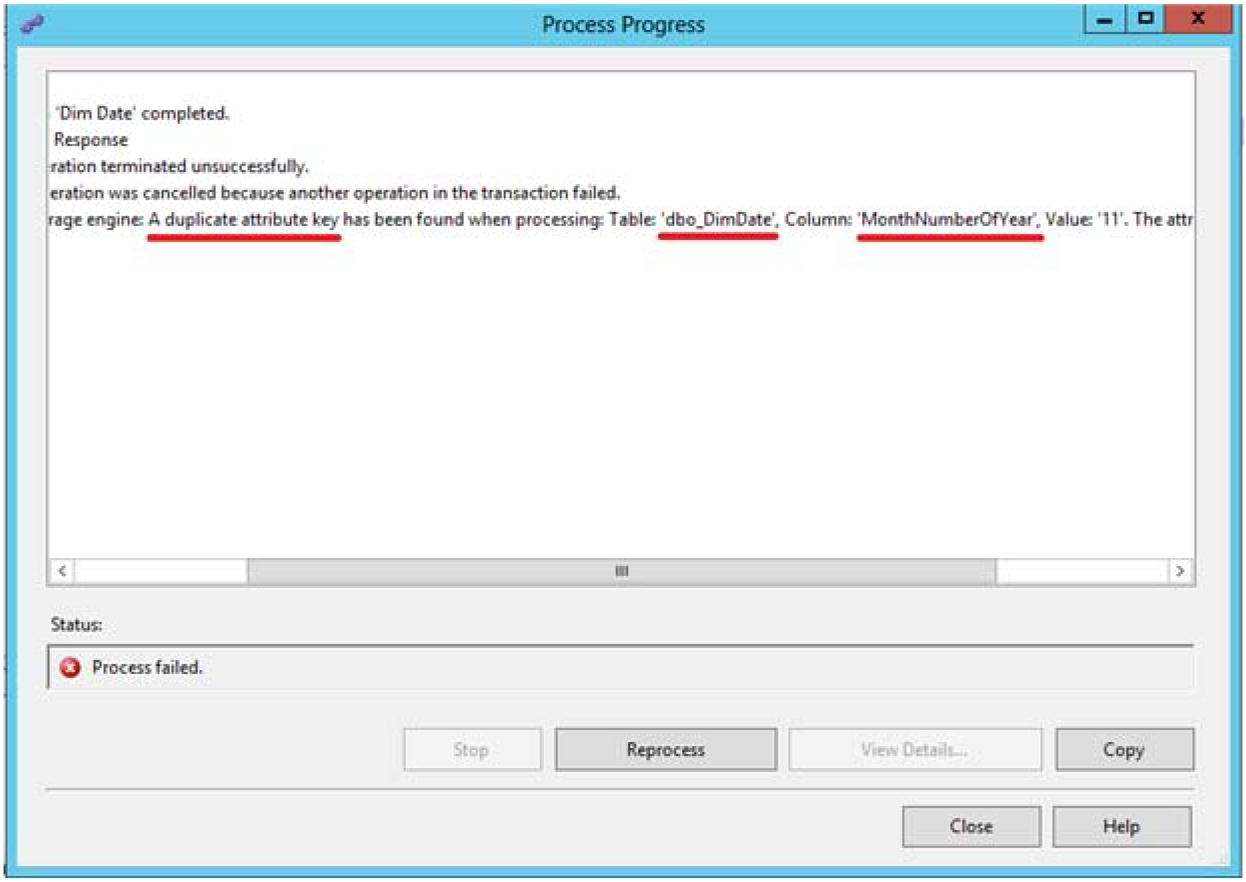
Status:

- Click *'Run'*

You will notice the following error message:

A duplicate attribute key has been found when processing: Table 'Dbo_DimDate', column: 'MonthNumberOfYear'

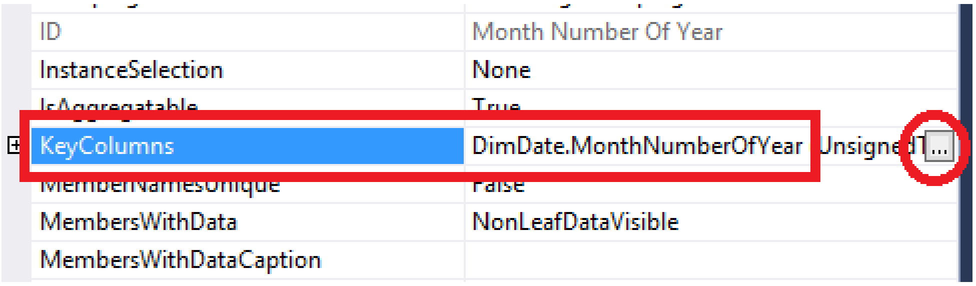
This happens exactly for the reason I explained before: Month can't repeat every year, the hierarchy doesn't accept this. We need to change the 'Month Number of Year' attribute just a bit to be accepted in the hierarchy.



'Dim Date' completed.
Response
ration terminated unsuccessfully.
eration was cancelled because another operation in the transaction failed.
rage engine: A duplicate attribute key has been found when processing: Table: 'dbo_DimDate', Column: 'MonthNumberOfYear', Value: '11'. The attr

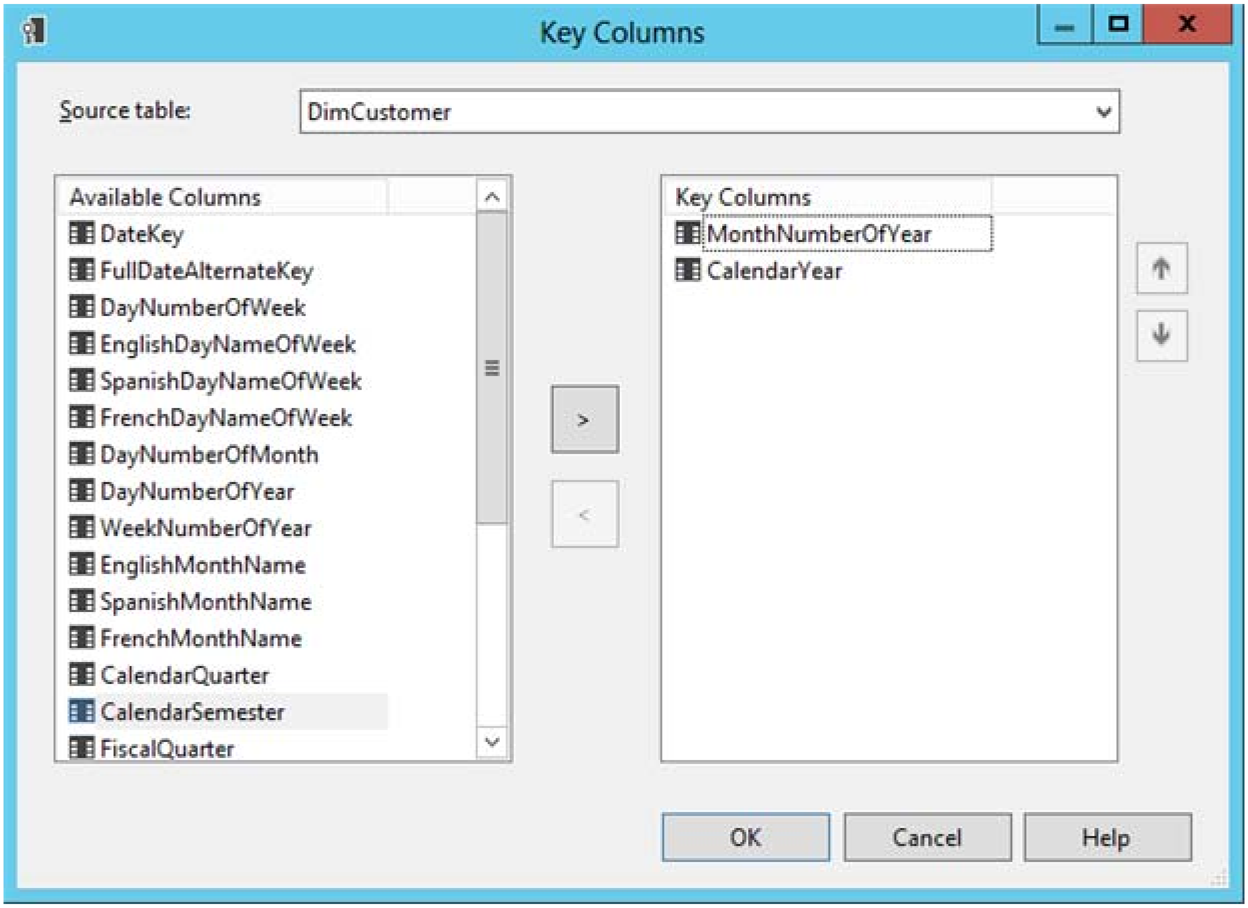
Status:
Process failed.

- Click *'Close'* button on the *'Process Progress'*window and *'Close'* button again in the *'Process Cube'* window, closing both windows.
- Click *'Dim Date'* tab
- In *'Attributes*' area, select *'Month Number of Year'* attribute.
- Right-click *'Month Number of Year'* attribute and click *'Properties'* menu item
- In the *'Properties'* window, click *'KeyColumns'* property and click the '… ' button



ID	Month Number Of Year
InstanceSelection	None
IsAggregatable	True
⊞ KeyColumns	DimDate.MonthNumberOfYear Unsigne ...
MemberNamesUnique	False
MembersWithData	NonLeafDataVisible
MembersWithDataCaption	

- In the *'Key Columns'* window, add *'Calendar Year'* to the collection of key columns



Key Columns

Source table: DimCustomer

Available Columns
DateKey
FullDateAlternateKey
DayNumberOfWeek
EnglishDayNameOfWeek
SpanishDayNameOfWeek
FrenchDayNameOfWeek
DayNumberOfMonth
DayNumberOfYear
WeekNumberOfYear
EnglishMonthName
SpanishMonthName
FrenchMonthName
CalendarQuarter
CalendarSemester
FiscalQuarter

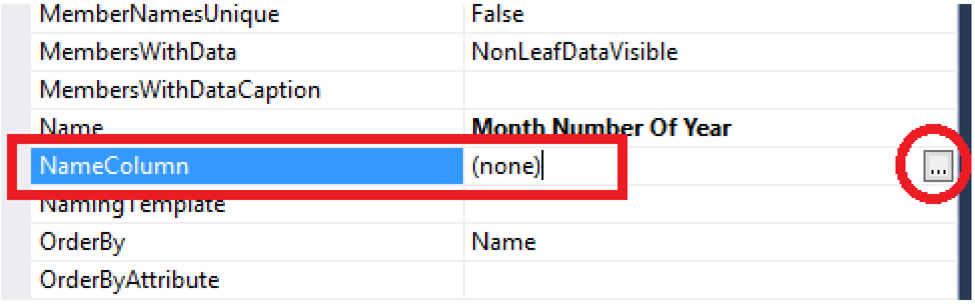
Key Columns
MonthNumberOfYear
CalendarYear

Now each Month information refers to Month/Year, so the information isn‘t duplicated any more. Month “1” (January) was duplicated because every year has month “1”, but there is only one month “1/2008”

Since the attribute has two keys, we need to specify the *‘NameColumn’* property. The *‘NameColumn’* specifies the field that will be displayed for each month.

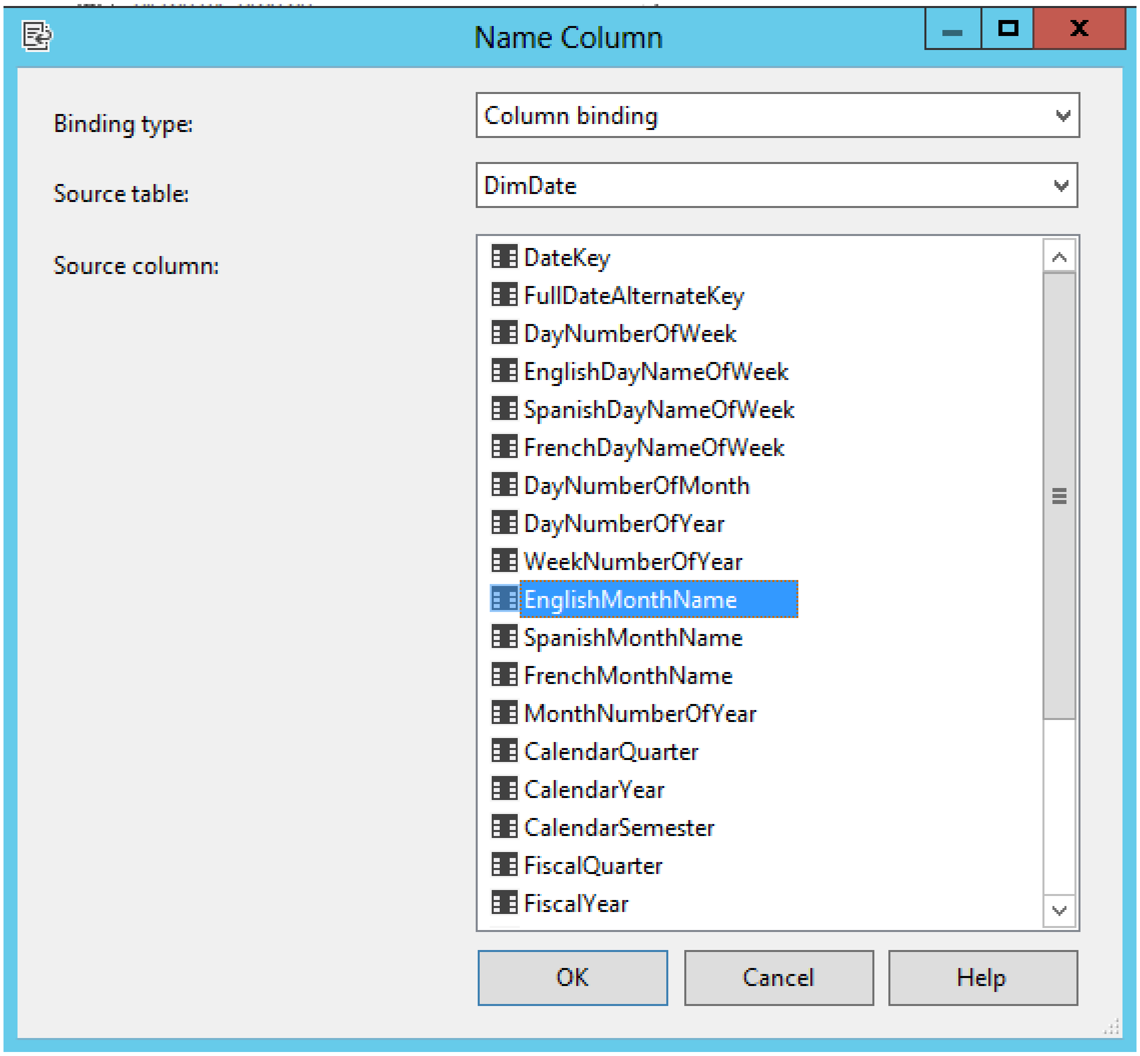
We do not need to choose any of the key columns. We can choose any column in the dimension table. For the *‘Month Number of Year’* attribute, the best column to choose as name column is *‘EnglishMonthName’*.

- Click *‘Ok’* button
- Click *‘NameColumn’* property and click the ‘… ‘ button



MemberNamesUnique	False
MembersWithData	NonLeafDataVisible
MembersWithDataCaption	
Name	**Month Number Of Year**
NameColumn	(none)
NamingTemplate	
OrderBy	Name
OrderByAttribute	

- In *'Name Column'* window, select *'EnglishMonthName'*field



Name Column

Binding type: Column binding

Source table: DimDate

Source column:

- DateKey
- FullDateAlternateKey
- DayNumberOfWeek
- EnglishDayNameOfWeek
- SpanishDayNameOfWeek
- FrenchDayNameOfWeek
- DayNumberOfMonth
- DayNumberOfYear
- WeekNumberOfYear
- EnglishMonthName
- SpanishMonthName
- FrenchMonthName
- MonthNumberOfYear
- CalendarQuarter
- CalendarYear
- CalendarSemester
- FiscalQuarter
- FiscalYear

- Click the *'Ok'* button
- Click the *'Save'* button on the toolbar

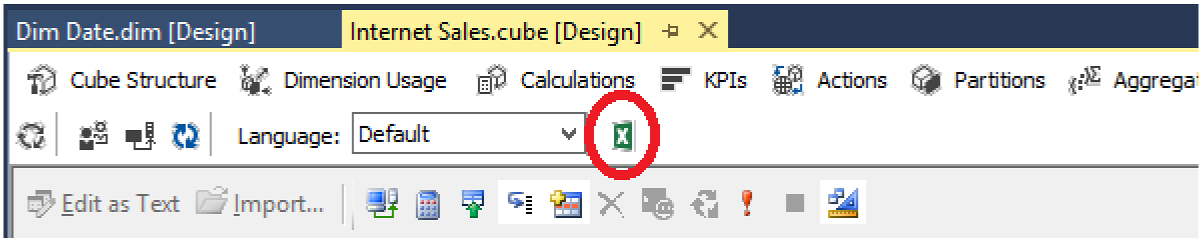
Let's process the cube again:

- Select the *'Internet Sales.cube'* tab
- Select the *'Browser'* tab
- Click *'Process'* button on the toolbar
- Click *'Yes'* button
- Click *'Run'* button

Now the process works.

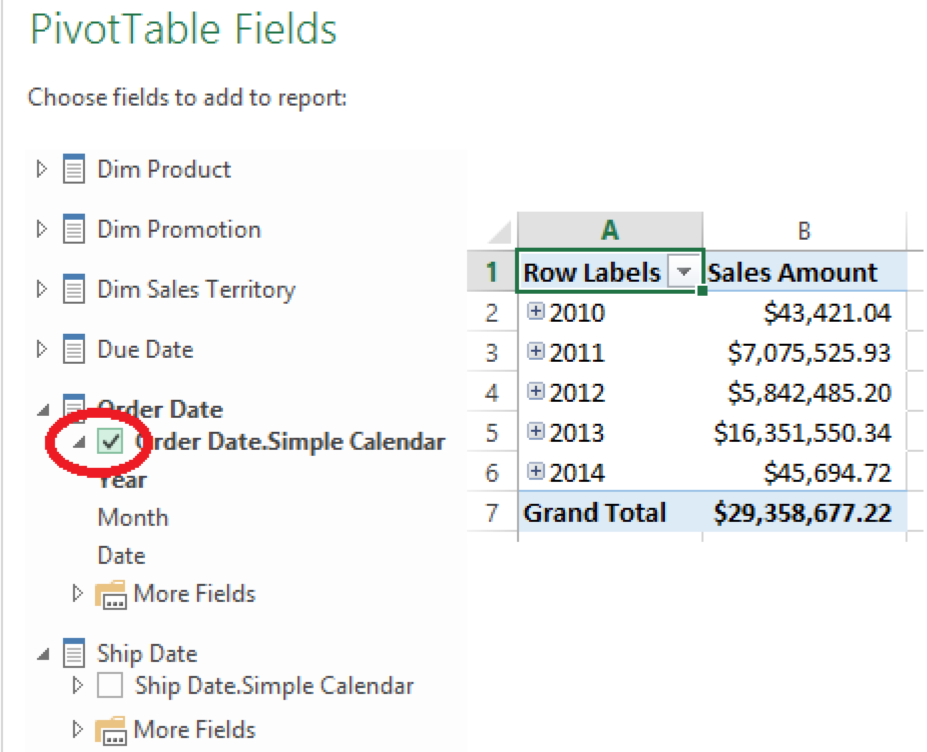
- Click *'Close'* button, and *'Close'* button again in the next window

Click *'Analyze in Excel'* button on the toolbar



- In Excel window, click *'Enable'*
- In the *'Pivot Table Fields'* Panel, in *'Fact Internet Sales'*, select *'sales amount'*
- In *'Order date'*dimension, select *'Simple Calendar'*hierarchy

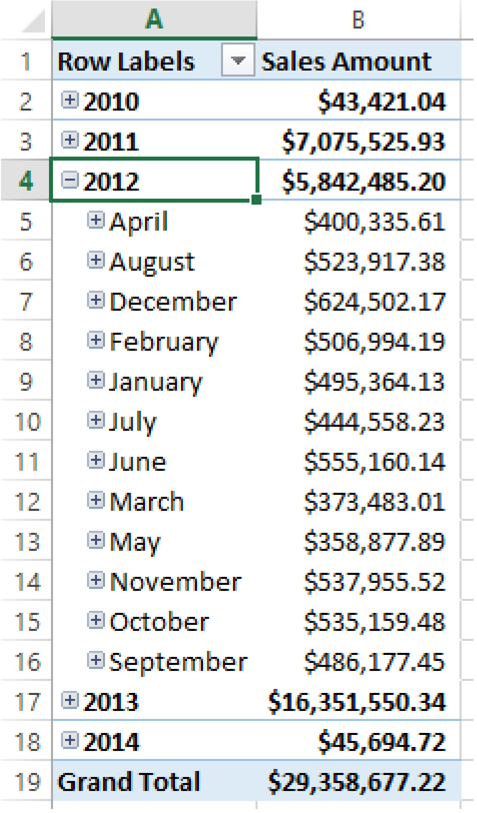
Our dimension is finally taking shape. After you selected *'Simple Calendar'*, you can see the pivot table with the hierarchy we created and the sales amount, allowing the user to drill down across the members of our hierarchy. Our hierarchy makes the construction of the pivot table easier, otherwise we would need to drag the attributes one by one, *'Year'*, *'Month'* and *'Date'*.



	A	B
1	Row Labels	Sales Amount
2	⊞2010	$43,421.04
3	⊞2011	$7,075,525.93
4	⊞2012	$5,842,485.20
5	⊞2013	$16,351,550.34
6	⊞2014	$45,694.72
7	Grand Total	$29,358,677.22

On the other hand, our dimension is far from ready yet. There are some mistakes what we need to fix. Let's identify what's wrong:

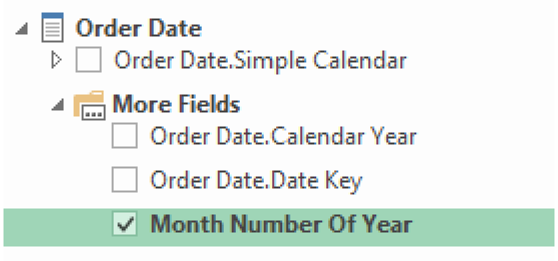
- In the pivot table, click the plus sign besides 2012 year and notice that the order of the months is wrong



	A	B
1	Row Labels	Sales Amount
2	⊞2010	$43,421.04
3	⊞2011	$7,075,525.93
4	⊟2012	$5,842,485.20
5	⊞April	$400,335.61
6	⊞August	$523,917.38
7	⊞December	$624,502.17
8	⊞February	$506,994.19
9	⊞January	$495,364.13
10	⊞July	$444,558.23
11	⊞June	$555,160.14
12	⊞March	$373,483.01
13	⊞May	$358,877.89
14	⊞November	$537,955.52
15	⊞October	$535,159.48
16	⊞September	$486,177.45
17	⊞2013	$16,351,550.34
18	⊞2014	$45,694.72
19	Grand Total	$29,358,677.22

- Click the plus sign besides one of the months and you’ll notice that all the days are displayed, exactly as we asked for and this isn‘t good, because we have more than thirty days. Let ‘s solve this later.
- In *‘Pivot Table Fields’*, under *‘Order Date’*, uncheck the *‘Simple Calendar’* hierarchy
- Open the folder *‘more fields’*.

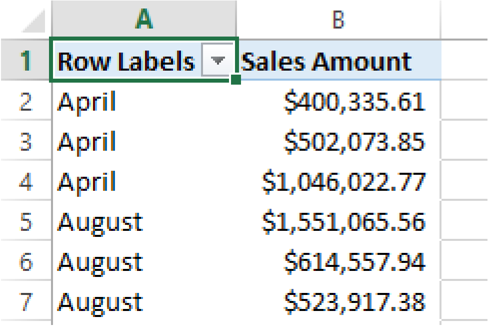
Even after we created the hierarchy, all the attributes are available to use inside the *‘more fields’* folder.



Order Date
- □ Order Date.Simple Calendar
- More Fields
  - □ Order Date.Calendar Year
  - □ Order Date.Date Key
  - ☑ Month Number Of Year

- Select the *‘Month Number of Year’* attribute.

This is the result now:



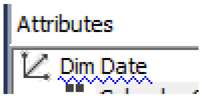
	A	B
1	Row Labels	Sales Amount
2	April	$400,335.61
3	April	$502,073.85
4	April	$1,046,022.77
5	August	$1,551,065.56
6	August	$614,557.94
7	August	$523,917.38

It would be useful to analyze the information by month instead of year/month. For example, if we would like to know in which month the company has the biggest or the smallest sales amount, the hierarchy will not help, because the hierarchy separates the values by year.

As you would have noticed, using the *'Month Number of Year'* without the hierarchy doesn't work anymore. Every month is repeated, one for each year. This is happening because we included the year in the *'KeyColumns'* property of the *'Month Number of Year'* attribute.

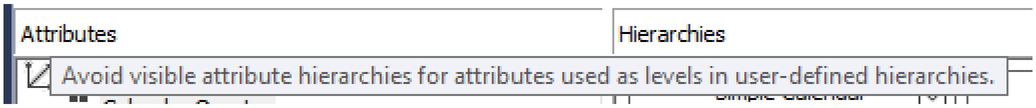
The conclusion is that we can't use the same attribute for both a hierarchy and an individual analysis.

Exactly for this reason, the dimension editor warn us with a blue underline on the dimension name, *"Dim Date"*, pointing that fields that are part of a hierarchy should not be visible.



Attributes

Dim Date



Attributes

Hierarchies

Avoid visible attribute hierarchies for attributes used as levels in user-defined hierarchies.

To solve this problem you don't need to close Excel; you only need to return to Visual Studio.

In the 'Dim Date Design' tab, in the *'attributes'* area, let's change the visibility of the three attributes. We need to change the property 'AttributeHierarchyVisible' to false. This will hide the attributes from the *'More fields'* folder without affecting our custom hierarchy.

Interesting point: All the attributes that we dragged from the data source view to the attributes area of the dimension generated their own hierarchy. Excel only displays hierarchies. The name of the property is *'AttributeHierarchyVisible'*, so if you change this property to false, the hierarchy generated by the attribute will not be displayed in Excel, but will still work inside our custom hierarchy, *'Simple Calendar'*. It 's easy, but not accurate, to say that this property will hide the attribute.

"Why hide all the three attributes?" you might ask. Let's see the details:

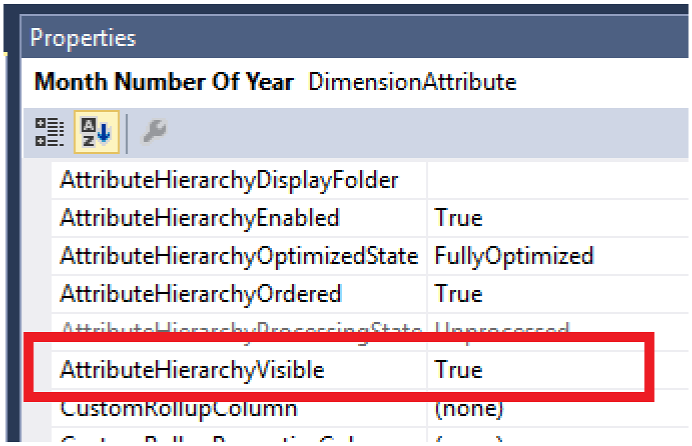
Calendar Year: This is the first attribute in the hierarchy, so we don't need this attribute to be visible. If we drag the *'Calendar Year'* attribute to a pivot table or drag the hierarchy, we will see the same information, so we can hide *'Calendar Year'* attribute.

Date Key: The days are the last level, the leaf level of our dimension. It 's not usual to aggregate the sales by days; it results in too many detail lines.

Month Number of Year: We can't use the same attribute for the hierarchy and individual analysis. We tried this a few steps before and you saw the wrong result.

Therefore, we can hide all the three attributes.

In the image below, you see the property that you need to change for the three attributes:



Properties

Month Number Of Year DimensionAttribute

AttributeHierarchyDisplayFolder	
AttributeHierarchyEnabled	True
AttributeHierarchyOptimizedState	FullyOptimized
AttributeHierarchyOrdered	True
AttributeHierarchyProcessingState	Unprocessed
AttributeHierarchyVisible	True
CustomRollupColumn	(none)

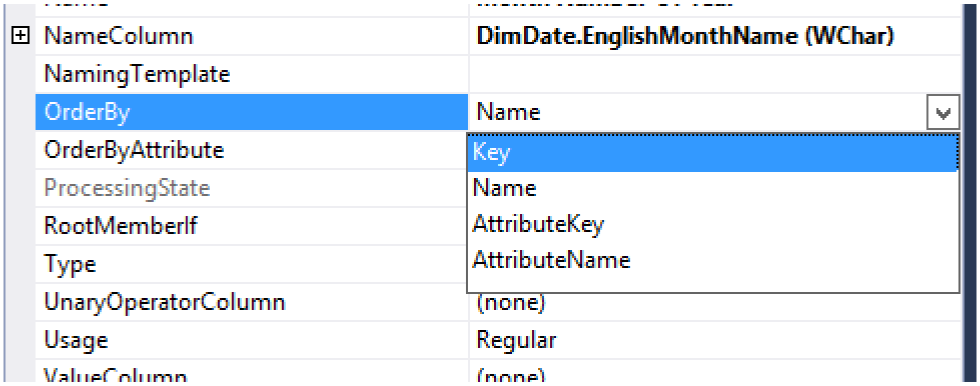
The Difference between AttributeHierarchyVisible and AttributeHierarchyEnabled

When we hide the attribute hierarchy, we can still use the attribute in our custom hierarchies. If we disable the hierarchy, we can't do so; instead, we can only use the attribute as a member to order other hierarchies or as name column in other hierarchies.

Next problem: The order of the month

We just need to change the property *'OrderBy'* of the attribute *'Month Number of Year'* to *'Key'*, problem solved. Instead of ordering by *'EnglishMonthName'*, the name column, the order will now be by *'MonthNumberOfYear'*, the key column.

There are other solutions. For example, we could use *'EnglishMonthName'* as an attribute, instead of *'Month Number of Year'*. The *'KeyColumns'* should be *'Month Number of Year'* and *'Calendar Year'*, the 'NameColumn ' should be *'EnglishMonthName'* and the *'OrderBy'* property should be 'Key '. '



NameColumn	DimDate.EnglishMonthName (WChar)
NamingTemplate	
OrderBy	Name
OrderByAttribute	
ProcessingState	
RootMemberIf	
Type	
UnaryOperatorColumn	(none)
Usage	Regular
ValueColumn	(none)

Next problem: The total number of days

Each month has around thirty days. If we display them all it means that there will be too many detail lines in the pivot table, making the visualization much less useful. It would be better to find a way to group the days below the month level. How can we create clusters among the thirty days in a month?

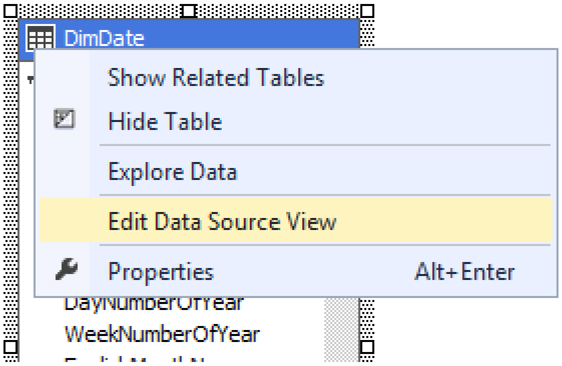
The attributes has the property *'DiscretizationMethod'* that we could use to create these clusters. This property aggregates attribute values when there isn't one hierarchy between them, that's the case of the days. On the other hand, this isn't a good solution, because *'DiscreatizationMethod'* create dynamic aggregations, we haven't control of how the values will be aggregated.

Another solution to aggregate the days is to use groups of ten days. It will result in two groups of ten days and one final varying group for each month. We will have fewer detail lines and the user will be able to identify which part of the month get the best sales result. For example, it would be easy to identify if the most amount of sales happens in the beginning or in the end of the month.

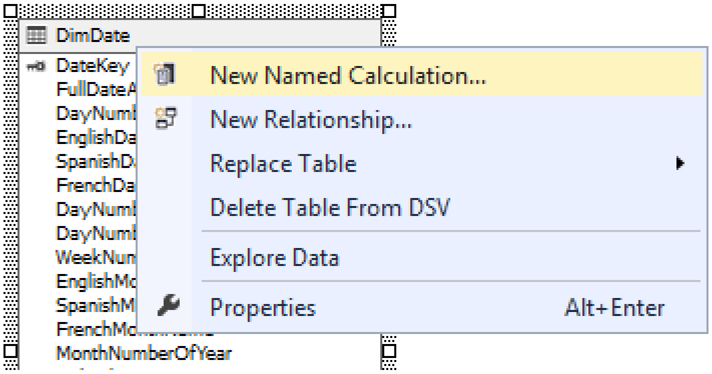
Let's call these groups *'Decans'* (yes, I found this name in astrology). We need to create a calculated member named *'Decans'* to classify each record in 1st, 2nd or 3rd decan.

Let's start the steps to create the 'Decans':

- Right-click the *'DimDate'* table in *'data source view'* area.
- Click the *'Edit data source view'* menu item



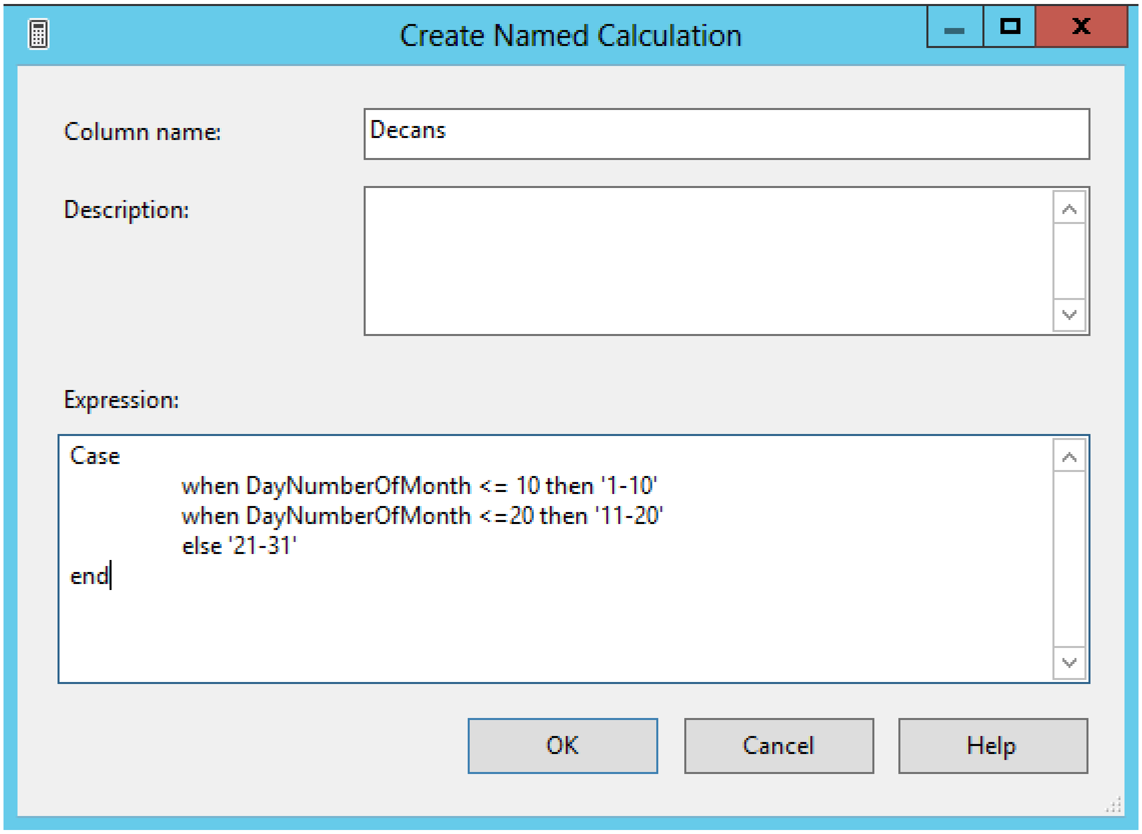
- In the ‘Data Source View Editor’ window, right-click the *‘DimDate’* table
- Click the *‘New Named Calculation’* menu item



In the expression box, inside the *‘Create Named Calculation’* window, use the following expression:

```
Case
        when DayNumberOfMonth <= 10 then '1-10'
        when DayNumberOfMonth <=20 then '11-20'
        else '21-31'
end
```

- In *‘Column Name’* field in the same window, type *‘Decans’* as the column name



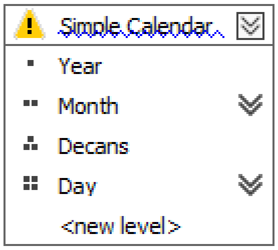
Column name: Decans

Description:

Expression:

```
Case
        when DayNumberOfMonth <= 10 then '1-10'
        when DayNumberOfMonth <=20 then '11-20'
        else '21-31'
end
```

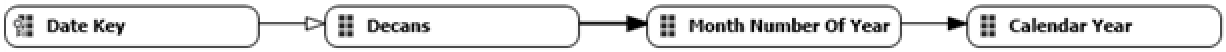
- Click the *'Ok'* button
- Click the 'Save' button on the toolbar
- Select *'Dim Date'* tab in Visual Studio
- Drag and drop *'Decans'* field from *'DimDate'* table to attribute collection
- Drag and drop *'Decans'* attribute to the simple calendar hierarchy, above *'Day'* attribute



Notice the error message again in the *'Simple Calendar'* hierarchy. We need to correct the attribute relationship.

- Click *'Attribute Relationships'* tab
- Right-click *'Decans'* attribute and click *'New attribute relationship'* menu item
- In the *'Related Attribute'*, inside the *'Create Attribute Relationship'*window, select *'Month Number of Year'* as the related attribute
- In the *'Relationship Type'*combobox, select *'Rigid'*

- Click *'Ok'* button

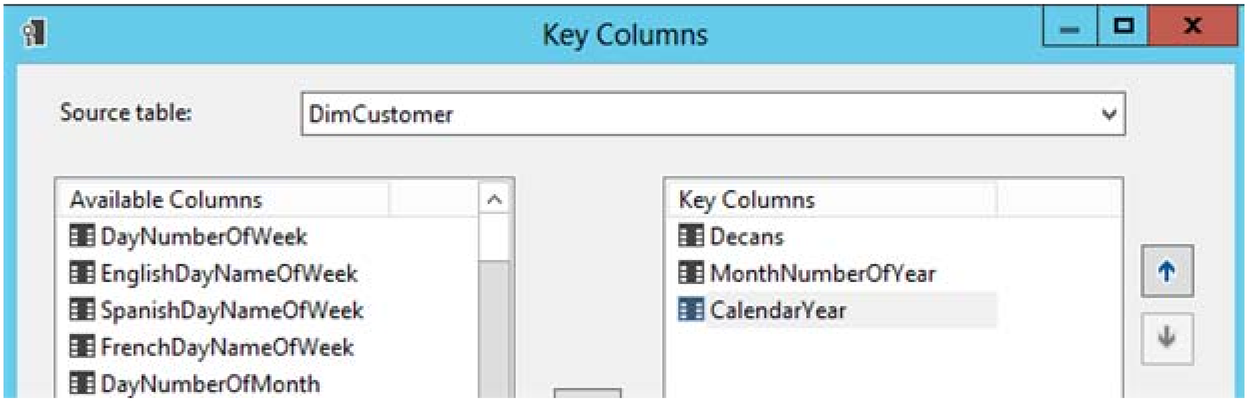


- Select *'Dimension Structure'*tab

The *'Decans'* values repeats each month and the months repeats each year. This repetition isn't allowed inside a hierarchy, we already faced this problem before with the *'Month Number of Year'* attribute.

The solution is the same we used for *'Month Number of Year'*: include additional key columns.

- Right-click the *'Decans'* attribute and click the 'Properties ' menu item
- In the *'Properties'* window, select the *'KeyColumns'* property and click the *'…'* button
- Add *'Month Number of Year'* and *'Calendar Year'* to the key properties, moving these columns from the *'Available Columns'* area to the *'Key Columns'* area



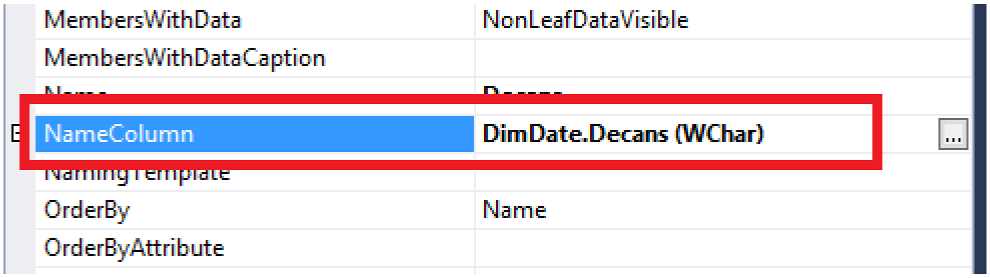
Key Columns

Source table: DimCustomer

Available Columns
- DayNumberOfWeek
- EnglishDayNameOfWeek
- SpanishDayNameOfWeek
- FrenchDayNameOfWeek
- DayNumberOfMonth

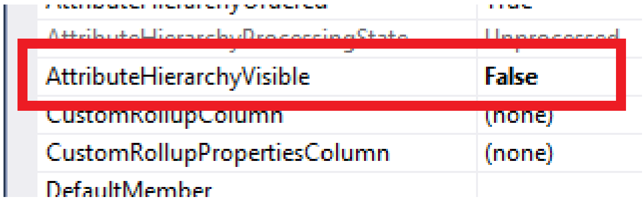
Key Columns
- Decans
- MonthNumberOfYear
- CalendarYear

- Click *'Ok'* button
- In *'NameColumn'* property, select *'Decans'* field



MembersWithData	NonLeafDataVisible
MembersWithDataCaption	
Name	Decans
NameColumn	DimDate.Decans (WChar)
NamingTemplate	
OrderBy	Name
OrderByAttribute	

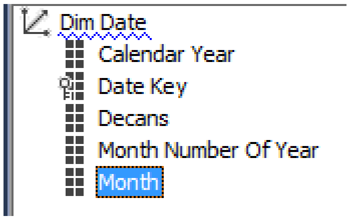
- Change *'AttributeHierarchyVisible'* property to false



AttributeHierarchyOrdered	True
AttributeHierarchyProcessingState	Unprocessed
AttributeHierarchyVisible	**False**
CustomRollupColumn	(none)
CustomRollupPropertiesColumn	(none)
DefaultMember	

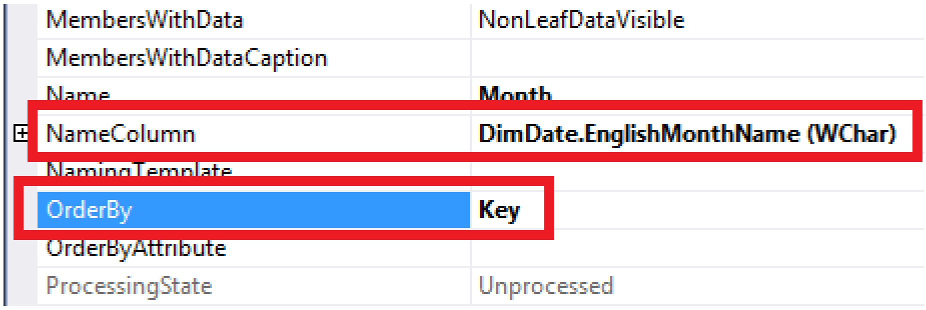
Both information, the month and the ‘decans ‘ are useful for analysis independently of the *‘Calendar Year’*, but, as I explained before, we can’t use the same attribute in a hierarchy and for individual attribute analysis. Instead, we can add the same field multiple times, as different attributes.

- Drag *‘Month Number of Year’* field to the collection of attributes
- Right-click *‘Month Number of Year 1’* attribute and rename it as *‘Month’*



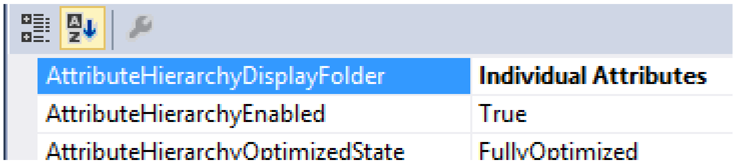
Dim Date
- Calendar Year
- Date Key
- Decans
- Month Number Of Year
- Month

- Right-click *‘Month’* attribute and click *‘Properties’* menu item
- Fill *‘NameColumn’* property with ‘EnglishMonthName ‘ column
- Fill *‘OrderBy’* property with the *‘Key’* value



MembersWithData	NonLeafDataVisible
MembersWithDataCaption	
Name	**Month**
⊞ NameColumn	**DimDate.EnglishMonthName (WChar)**
NamingTemplate	
OrderBy	**Key**
OrderByAttribute	
ProcessingState	Unprocessed

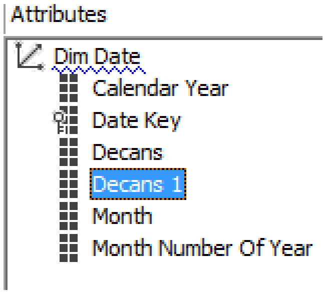
- Fill the property *‘AttributeHierarchyDisplayFolder’* with *‘Individual Attributes’*



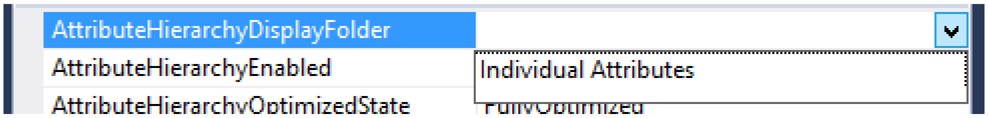
AttributeHierarchyDisplayFolder	**Individual Attributes**
AttributeHierarchyEnabled	True
AttributeHierarchyOptimizedState	FullyOptimized

This will create a custom folder named *‘Individual Attributes’* for this attribute, instead of the folder *‘More Fields’* we saw before in Excel.

- Drag *‘Decans’* field from *‘DimDate’* table to the collection of attributes



- Right-click *'Decans 1'* attribute and rename it as *'Decan'*
- Change the property *'AttributeHierarchyDisplayFolder'* of the *'Decan'* attribute to *'Individual Attributes'*. This time the value is already there, you only need to select.

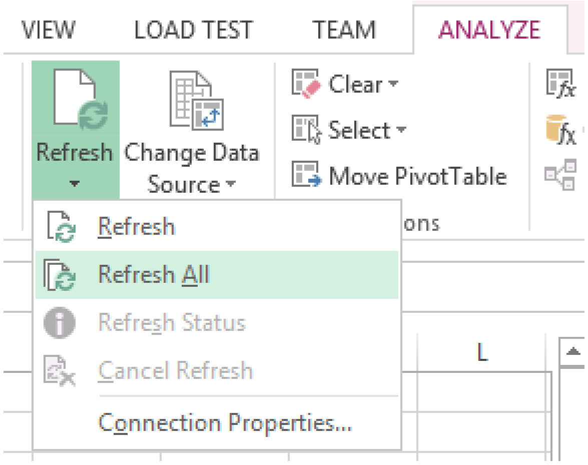


AttributeHierarchyDisplayFolder
AttributeHierarchyEnabled — Individual Attributes
AttributeHierarchyOptimizedState — FullyOptimized

2nd Test in Excel

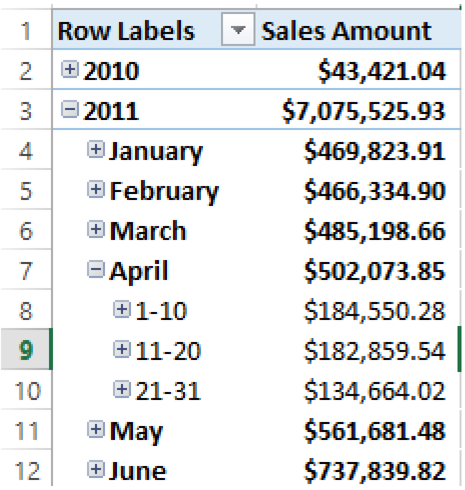
Let 's process the cube again using the same steps we already did before:

- Click *'Internet Sales'* tab
- Click *'Browser'* tab
- Click *'Process'* button on the toolbar
- Click *'Yes'*
- Click *'Run'* button
- Click the *'Close'* button and *'Close'* button again in the next window
- Select the Excel window (It 's still open, right?)
- Uncheck all checkboxes in *'Pivot Table'* fields, cleaning the table
- Click *'Refresh'*-> *'Refresh All'* on the toolbar



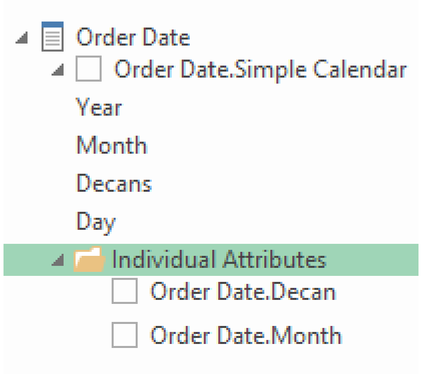
- In *‘Pivot Table’* Fields, Select (check) the *‘Sales amount’* measure
- Select *‘Simple Calendar*’ hierarchy in *‘Order Date’* dimension
- In the pivot table, now filled with the hierarchy and sales amount, click plus sign beside *‘2011’* year and again beside *‘April’*

Notice the months in the correct order and the ‘decans ‘ aggregating the days.



1	Row Labels ▾	Sales Amount
2	⊞ 2010	$43,421.04
3	⊟ 2011	$7,075,525.93
4	⊞ January	$469,823.91
5	⊞ February	$466,334.90
6	⊞ March	$485,198.66
7	⊟ April	$502,073.85
8	⊞ 1-10	$184,550.28
9	⊞ 11-20	$182,859.54
10	⊞ 21-31	$134,664.02
11	⊞ May	$561,681.48
12	⊞ June	$737,839.82

- In *‘Pivot Table’* fields, uncheck *‘Simple Calendar’*
- Open *‘Individual Attributes’* folder, below *‘Simple Calendar*’, and check *‘Month’* attribute



◢ Order Date
◢ □ Order Date.Simple Calendar
Year
Month
Decans
Day
◢ Individual Attributes
□ Order Date.Decan
□ Order Date.Month

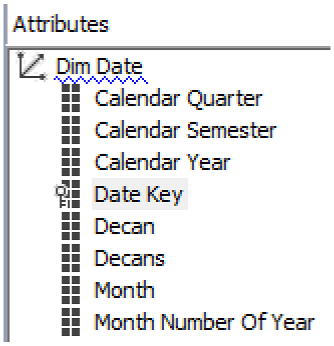
Now we can analyze the information by *‘Month’* and *‘Decan’*.

- Uncheck *‘Month’* attribute, check *‘Decan’* attribute and observe the pivot table

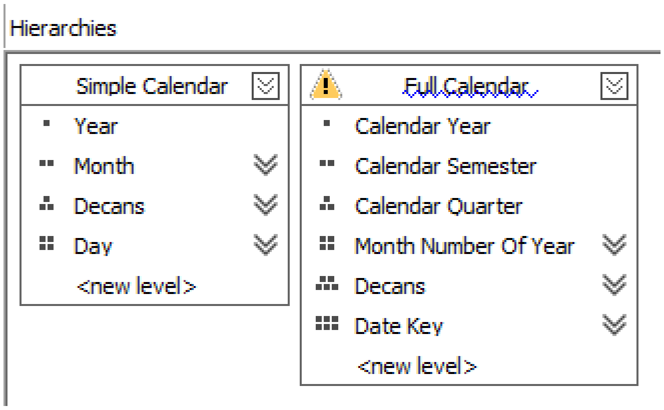
Full Calendar Hierarchy

Now that we already have our first hierarchy, let ‘s create a more complete calendar hierarchy. We left behind the ‘Calendar Semester‘ and ‘Calendar Quarter‘ Fields. We can create a new hierarchy, ‘Full Calendar‘, including these fields. Some users will use ‘Simple Calendar‘ hierarchy, it will be enough for them, and other users will need the ‘Full Calendar’ hierarchy.

- Switch to Visual Studio window
- Select *‘Dim Date’* tab
- Drag *‘CalendarSemester’* field to the *‘Attributes’* area
- Drag *‘CalendarQuarter’* field to the *‘Attributes’* area



- Drag *‘CalendarYear’* field from the *‘Attributes’* area to the *‘hierarchies’* area, creating a new hierarchy
- Right-click the new hierarchy and rename it to *‘Full Calendar’*
- Drag *‘Calendar Semester’* attribute to the *‘Full Calendar’* hierarchy
- Drag *‘Calendar Quarter’* attribute to the *‘Full Calendar’* hierarchy
- Drag *‘Month Number of Year’* attribute to the *‘Full Calendar’* hierarchy
- Drag *‘Decans’* attribute to the *‘Full Calendar’* Hierarchy
- Drag *‘Date Key’* attribute to the *‘Full Calendar’* Hierarchy



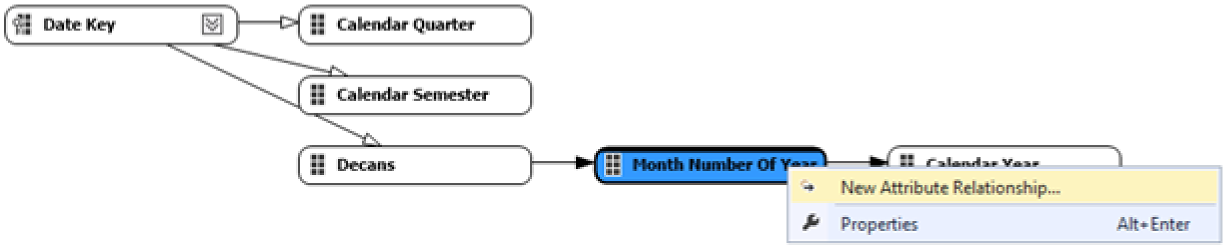
- Rename the hierarchy members to ‘Year ‘, ‘Semester ‘, ‘Quarter ‘, ‘Month ‘, ‘Decan ‘ and ‘Day ‘, from top to bottom, in this order.

We can notice the warning alert again, so we need to configure attribute relationship

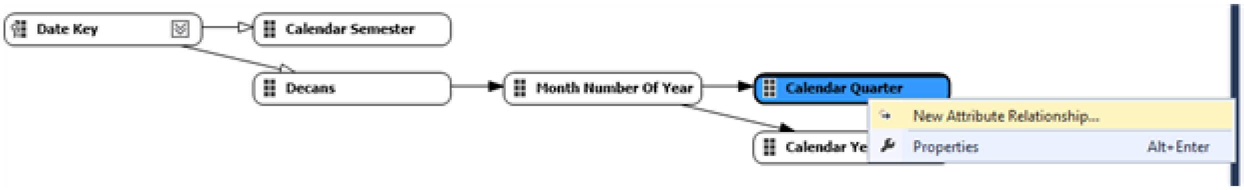
- Click *‘Attribute Relationship’* tab

Notice that some relationships were created for us, but we need to correct them

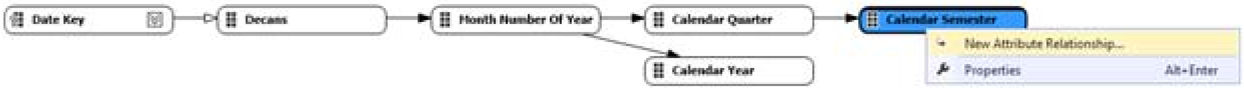
- Right-click *'Month Number of Year'* attribute and click *'New Attribute Relationship'*



- Ensure that *'Related Attribute'* combo box points to *'Calendar Quarter'*
- Change *'Relationship Type'* combobox to *'Rigid'*
- Click *'Ok'* button
- Right-click *'Calendar Quarter'* attribute and click *'New Attribute Relationship'*

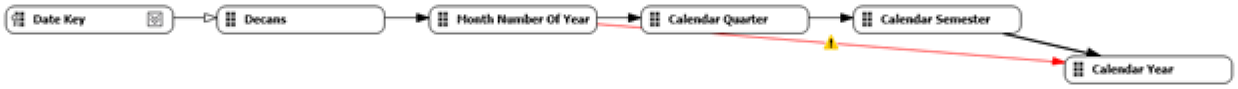


- Ensure that the *'Related Attribute'* combo box points to *'Calendar Semester'*
- Change *'Relationship Type'* combobox to *'Rigid'*
- Click the *'Ok'* button
- Right-click *'Calendar Semester'* and click *'New Attribute Relationship'*



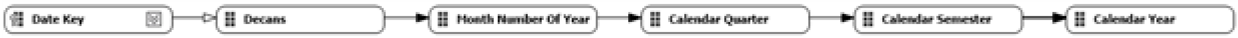
- Ensure that the *'Related Attribute'* combo box points to *'Calendar Year'*
- Change *'Relationship Type'* combobox to *'Rigid'*
- Click *'Ok'* button

Now we have a redundant relationship and we need to delete it. It's the red line between the attributes.



- Click the red line

- Press delete key
- Click *‘Ok’* button



- Select the dimension structure tab

Now the warning disappeared because we corrected the attribute relationships.

‘Semester’ and *‘Quarter’* attributes repeat themselves each year, but we already know that we can ‘t allow this in a hierarchy. We need to configure the *‘KeyColumns’* property for both, adding the *‘Calendar Year’* field in the *‘KeyColumns’* property and choosing a correct *‘NameColumn’* value, otherwise we will see an error again when we process the cube.

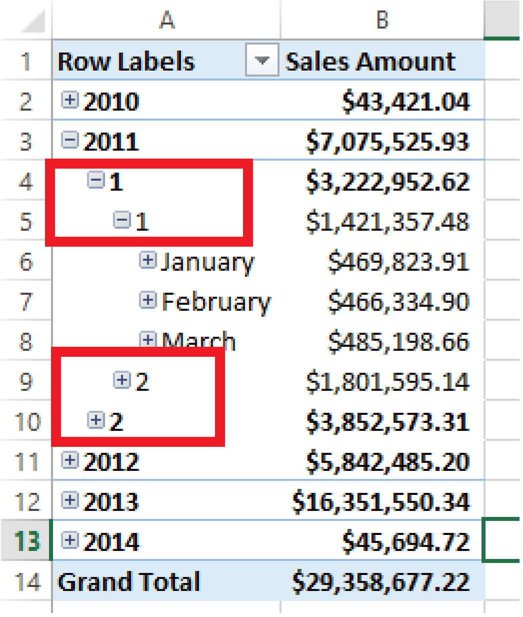
- Select *‘Calendar Semester’* in Attributes area
- Select *‘KeyColumns’* in properties window (open the properties window if not already opened) and click the *‘…’*button
- Select *‘CalendarYear’* field and move it to the right side of the window
- Click *‘Ok’* button
- Change *‘AttributeHierarchyVisible’* property to False
- Select *‘NameColumn’* property and click the *‘…’*button
- Select *‘CalendarSemester’* field as the name column
- Click *‘Ok’* button
- Select *‘Calendar Quarter’* in Attributes area
- Select *‘KeyColumns’* in properties window and click the *‘…’*button
- Select *‘CalendarYear’* and move it to the right side of the window
- Click *‘Ok’* button
- Change *‘AttributeHierarchyVisible’* property to false
- Select *‘NameColumn’* property and click the *‘…’*button
- Select *‘CalendarQuarter’* field as the name column
- Click *‘Ok’* button

3rd Test in Excel

- Click the ‘ *Save All’* button on the toolbar

- Process the cube (the same steps we did before)
- Switch to Excel window
- Uncheck all attributes/hierarchies in *‘PivotTable Fields’* area
- Click *‘Refresh’*-> *‘Refresh All’* on the toolbar
- Select *‘Sales amount’* measure in *‘PivotTable Fields’* area
- Select *‘Full Calendar’* Hierarchy in *‘PivotTable Fields’* area

Notice that the hierarchy is exactly as we asked for, but the names used for semesters and quarters aren‘t good: “1”,”2″ for semester, “1”,”2″,”3″,”4″ for quarter. We need better names for both semesters and quarters.



	A	B
1	Row Labels	Sales Amount
2	⊞2010	$43,421.04
3	⊟2011	$7,075,525.93
4	⊟1	$3,222,952.62
5	⊟1	$1,421,357.48
6	⊞January	$469,823.91
7	⊞February	$466,334.90
8	⊞March	$485,198.66
9	⊞2	$1,801,595.14
10	⊞2	$3,852,573.31
11	⊞2012	$5,842,485.20
12	⊞2013	$16,351,550.34
13	⊞2014	$45,694.72
14	Grand Total	$29,358,677.22

Naming Semester and Quarter

The solution for the naming problem of the ‘CalendarSemester’ and ‘CalendarQuarter’ attributes is to create a named calculation in the data source view. The named calculation is in fact a field that will check the value of the *‘CalendarSemester’* or *‘CalendarQuarter’* field and present the name that should be used for the record.

After we create the named calculations, we can choose these new fields as name column of the attributes, selecting them in *‘NameColumn’* property of the attributes.

This will be the T-SQL for *‘CalendarSemesterName’* named calculation:

```
Case CalendarSemester
        When 1 then '1st Semester'
        When 2 then '2nd Semester'
```

```
End
```

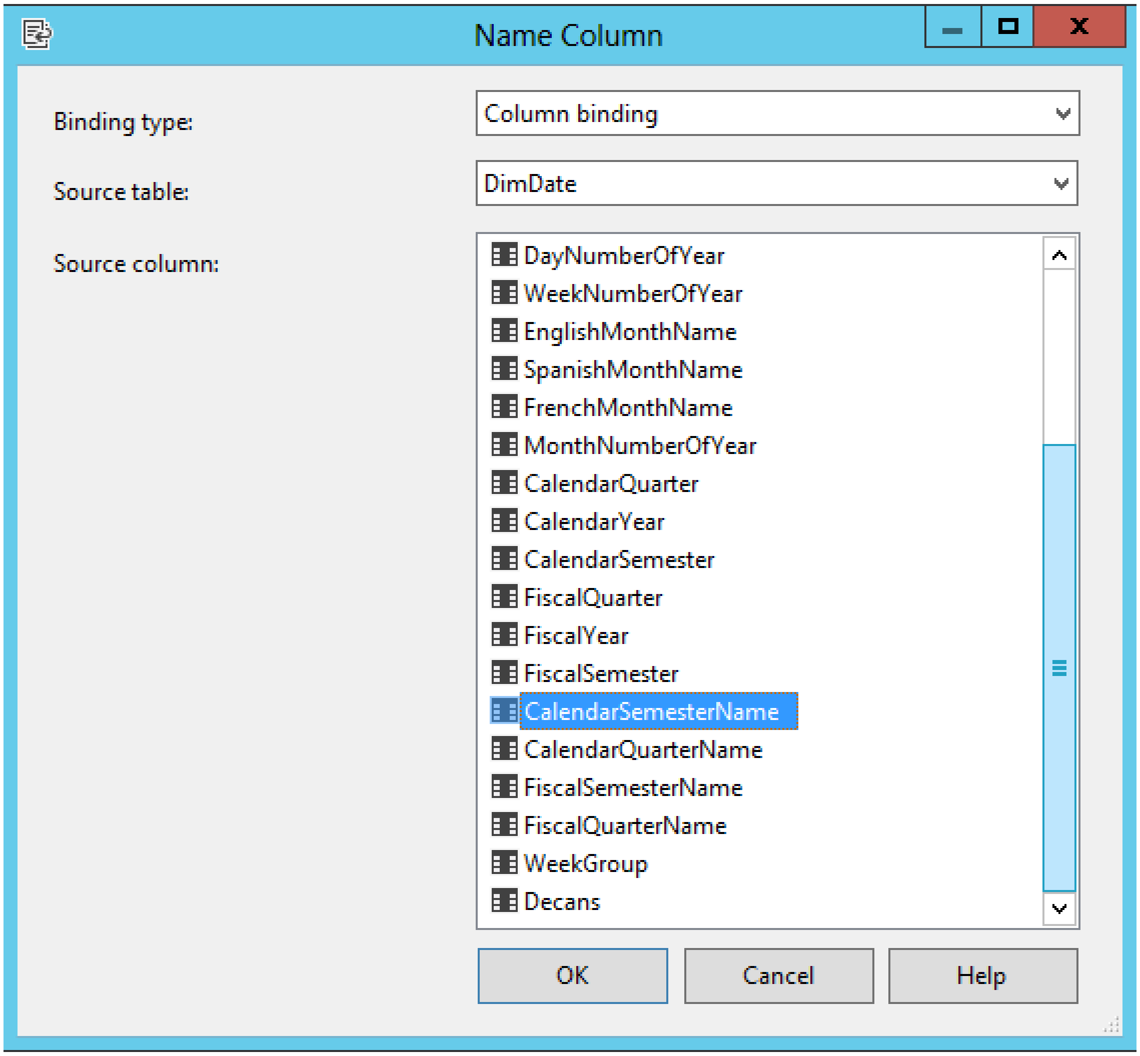
This will be the T-SQL for *‘CalendarQuarterName’* named calculation:

```
Case CalendarQuarter
        When 1 then '1st Quarter'
        When 2 then '2nd Quarter'
        When 3 then '3rd
        When 4 then '4th Quarter'
end
```

You need to repeat the same steps we used to create the *‘Decans’* named calculation to create the new *‘CalendarSemesterName’* named calculation using the first T-SQL instruction above and *‘CalendarQuarterName’* using the second T-SQL instruction above.

After the named calculations were created in the ‘Data Source View Editor’ window, you need to configure the name column of the attributes following these steps:

- In *‘Dim Date’* dimension tab, select *‘Calendar Semester’* and configure *‘NameColumn’* property as *‘CalendarSemesterName’*



Name Column

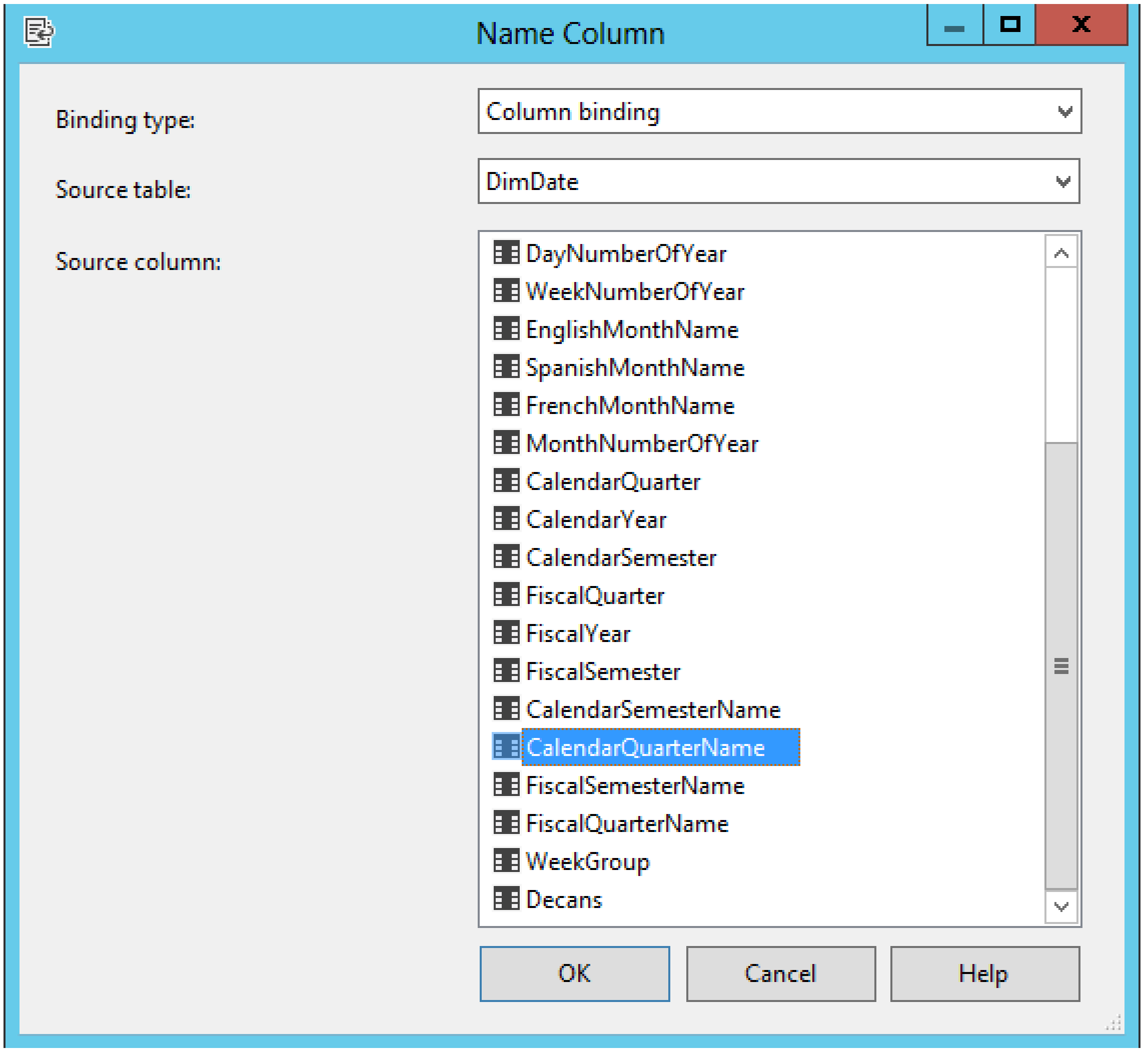
Binding type: Column binding

Source table: DimDate

Source column:

- DayNumberOfYear
- WeekNumberOfYear
- EnglishMonthName
- SpanishMonthName
- FrenchMonthName
- MonthNumberOfYear
- CalendarQuarter
- CalendarYear
- CalendarSemester
- FiscalQuarter
- FiscalYear
- FiscalSemester
- CalendarSemesterName
- CalendarQuarterName
- FiscalSemesterName
- FiscalQuarterName
- WeekGroup
- Decans

- Select *'Calendar Quarter'* and configure *'NameColumn'* property as *'CalendarQuarterName'*



Name Column

Binding type: Column binding

Source table: DimDate

Source column:

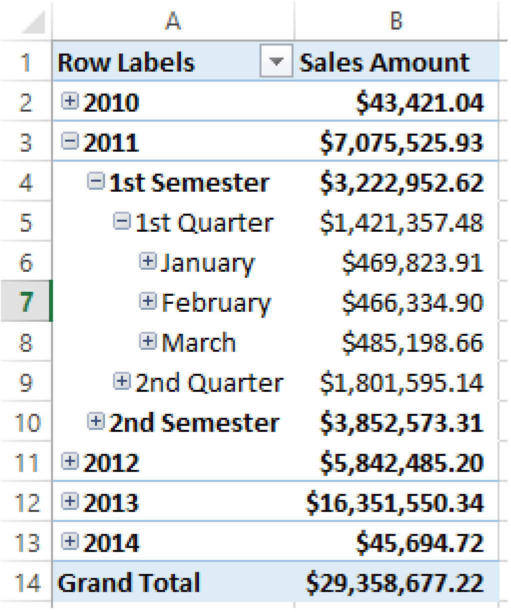
- DayNumberOfYear
- WeekNumberOfYear
- EnglishMonthName
- SpanishMonthName
- FrenchMonthName
- MonthNumberOfYear
- CalendarQuarter
- CalendarYear
- CalendarSemester
- FiscalQuarter
- FiscalYear
- FiscalSemester
- CalendarSemesterName
- CalendarQuarterName
- FiscalSemesterName
- FiscalQuarterName
- WeekGroup
- Decans

- ‘*Save All’* using the toolbar button
- Process the cube using the same steps we did before
- In Excel, select Analyze tab and click ‘ *Refresh All’*

Nothing will change. At this point semesters and quarters in the pivot table are still with the wrong and ugly name. This happens because the relation between attributes are rigid. In this case, we need to process the dimension, not only the cube.

- Select *‘Dim Date’* tab
- Click *‘Process Dimension’* button in the toolbar
- Select *‘Internet Sales.cube’* tab
- Process the cube following the same steps we did before
- Switch to Excel
- Click Refresh->Refresh All button in the ‘*Analyze’* tab

Now the names for semesters and quarters are correct, as you can see in the image below.



Row Labels	Sales Amount
⊞ 2010	$43,421.04
⊟ 2011	$7,075,525.93
⊟ 1st Semester	$3,222,952.62
⊟ 1st Quarter	$1,421,357.48
⊞ January	$469,823.91
⊞ February	$466,334.90
⊞ March	$485,198.66
⊞ 2nd Quarter	$1,801,595.14
⊞ 2nd Semester	$3,852,573.31
⊞ 2012	$5,842,485.20
⊞ 2013	$16,351,550.34
⊞ 2014	$45,694.72
Grand Total	$29,358,677.22

Analyze Semester and Quarter information as Individual attributes

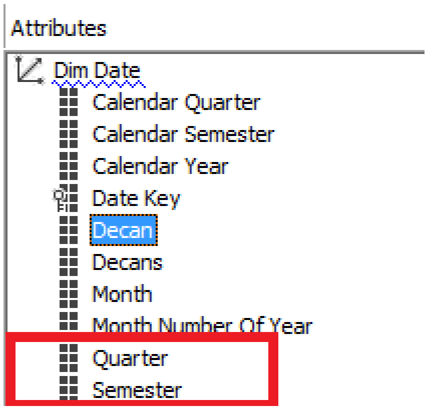
We have already provided *'Month'* and *'Decans'* attribute outside the hierarchies, in the *'Individual Attributes'* folder that we created to improve the analysis that our user is able to do in Excel. Now we need to give to the user the same ability with *'Calendar Semester'* and *'Calendar Quarter'* attributes and the way to do this can be the same: We can add the attributes again and configure the folder for the attributes hierarchy.

However, there is an even better solution: We can create a new hierarchy with *'Calendar Semester'*, *'Calendar Quarter'*, *'Month'* and *'Decans'* attributes. A hierarchy with these attributes and without the *'Calendar Year'* attribute will enable the user to aggregate the sales across the years and will make everything easier for the user, he will be able to drag a single hierarchy to the pivot table and analyze all these attributes across the years.

The only exception is the *'Decans' attribute*: In the hierarchy, each value of the *'Decans'* attribute repeats each month, so the user can analyze this information only by month. Using the hierarchy the user will be able to tell, inside June for example, which decan had the best sales result, but the user isn't able to tell which decan had the best sales result independent of the month. Because of this, although the hierarchy will be very useful, we still need to keep the *'Decan'* attribute available for individual analysis.

Let's execute the following steps to create the new hierarchy and keep *'Decan'* attribute in the *'Individual Attributes'* folder:

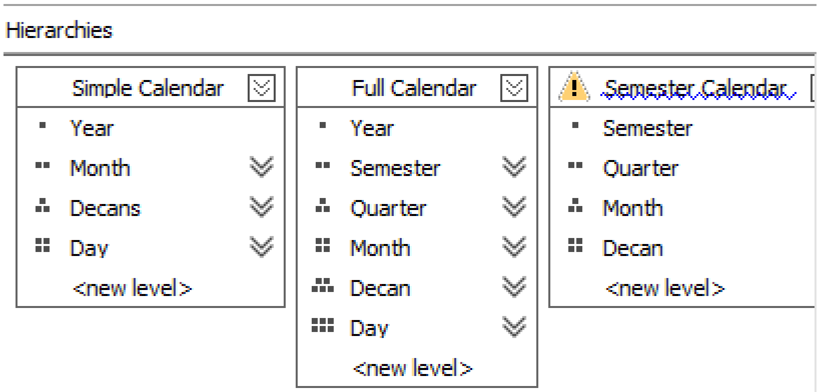
- Drag *'CalendarSemester'* and *'CalendarQuarter'* fields from the *'Data Source View'* area to the attributes area and rename them as *'Semester'* an *'Quarter'*



Attributes

Dim Date
Calendar Quarter
Calendar Semester
Calendar Year
Date Key
Decan
Decans
Month
Month Number Of Year
Quarter
Semester

- Change the *‘NameColumn’* property of *‘Semester’* attribute to *‘CalendarSemesterName’*
- Change *‘AttributeHierarchyVisible’* property of *‘Semester’* attribute to false
- Change the *‘NameColumn’* property of *‘Quarter’* attribute to *‘CalendarQuarterName’*
- Change *‘AttributeHierarchyVisible’* property of *‘Quarter’* attribute to false
- Change *‘AttributeHierarchyVisible’* property of *‘Month’* attribute to false
- Change *‘AttributeHierarchyVisible’* property of *‘Decan’* attribute to false
- Right-click *‘Semester’* attribute and click *‘Start New Hierarchy’* menu item
- Drag *‘Quarter’* attribute to the new hierarchy
- Drag *‘Month’* attribute to the new hierarchy
- Drag *‘Decan’* attribute to the new hierarchy
- Change the new hierarchy name to *‘Semester Calendar’*



Hierarchies

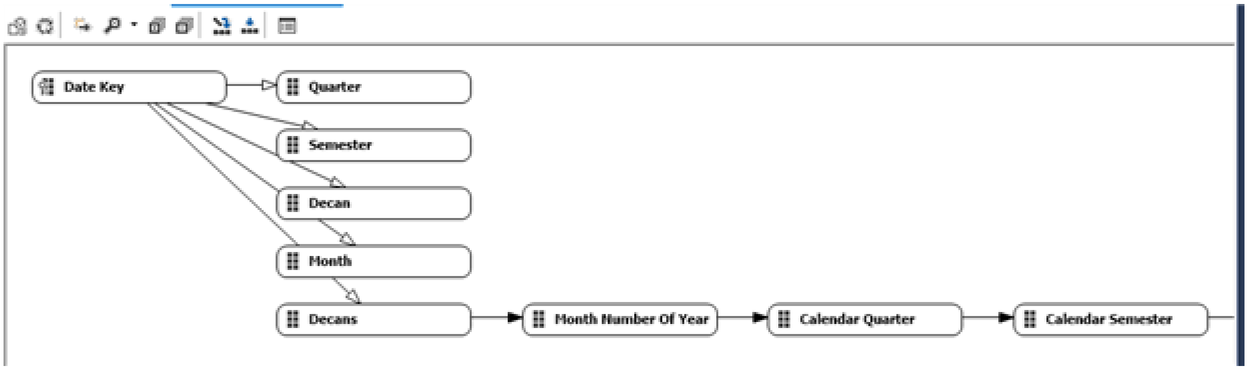
Simple Calendar
Year
Month
Decans
Day
<new level>

Full Calendar
Year
Semester
Quarter
Month
Decan
Day
<new level>

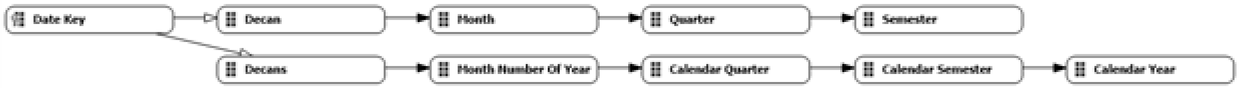
Semester Calendar
Semester
Quarter
Month
Decan
<new level>

Notice that we have problems again with attribute relationships. We need to adjust the attribute relationship again.

Click *‘Attribute Relationship’* tab

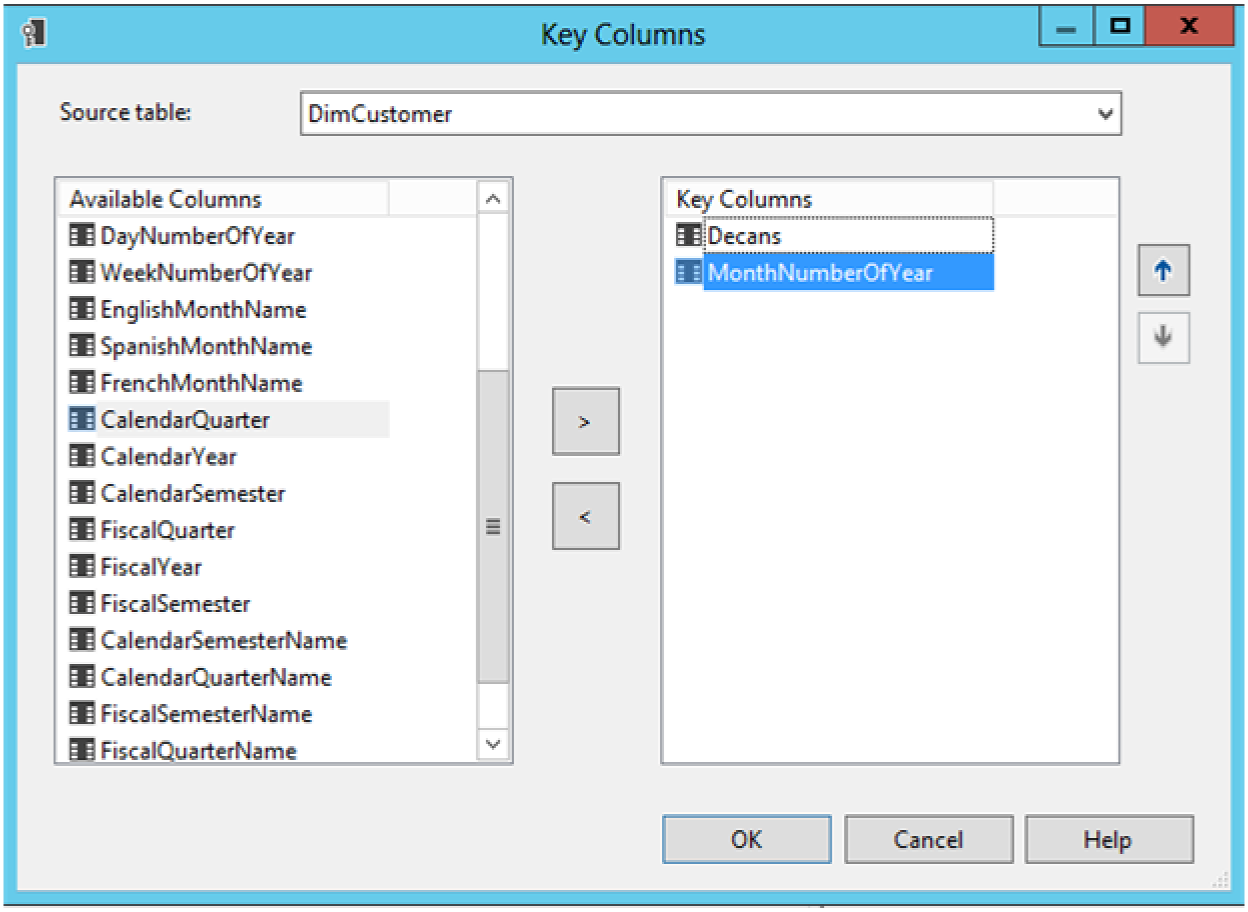


- Right-click *'Decan'* attribute and click *'New Attribute Relationship'*menu item
- Select *'Month'* attribute as *'Related Attribute'* and *'Rigid'* as 'Attribute Type '
- Click *'Ok'* button
- Repeat the same procedure for *'Month'* attribute, creating a relation with *'Quarter'* attribute
- Repeat the same procedure for *'Quarter'* attribute, creating a relation with *'Semester'* attribute



- Click *'Dimension Structure'* Tab

We need to remember that *'Decan'* values will repeat themselves each month and this isn't allowed in the hierarchy, so we need to correct the *'KeyColumns'* property of the *'Decan'* attribute to *'Decans'* + 'MonthNumberofYear' fields and the *'NameColumn'* property to *'Decans'.* We already faced this problem before with *'Month Number of Year'*, *'Calendar Semester'* and *'Calendar Quarter'* attributes.



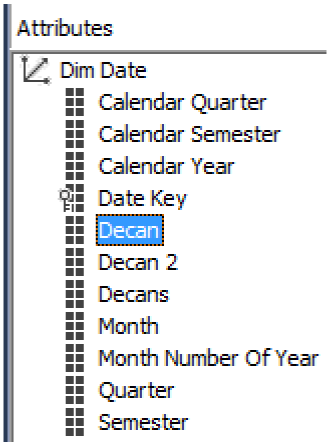
Source table: DimCustomer

Available Columns
- DayNumberOfYear
- WeekNumberOfYear
- EnglishMonthName
- SpanishMonthName
- FrenchMonthName
- CalendarQuarter
- CalendarYear
- CalendarSemester
- FiscalQuarter
- FiscalYear
- FiscalSemester
- CalendarSemesterName
- CalendarQuarterName
- FiscalSemesterName
- FiscalQuarterName

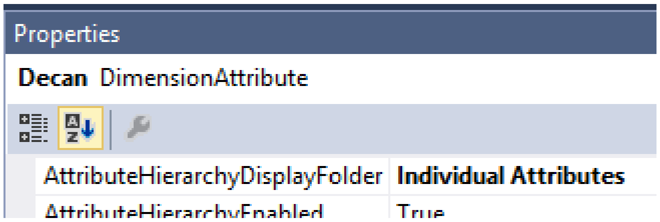
Key Columns
- Decans
- MonthNumberOfYear

Finally, we still need to add the individual analysis by *'Decans'* field. Let 's add *'Decans'* field again:

- Drag *'Decans'* field to the attributes area
- Rename *'Decan'* attribute as *'Decan 2'* (this attribute will not be visible)
- Rename *'Decans 1'* attribute as *'Decan'*


Attributes
- Dim Date
  - Calendar Quarter
  - Calendar Semester
  - Calendar Year
  - Date Key
  - Decan
  - Decan 2
  - Decans
  - Month
  - Month Number Of Year
  - Quarter
  - Semester

- Change *'AttributeHierarchyDisplayFolder'* property of the *'Decan'* attribute to *'Individual Attributes'*



Properties

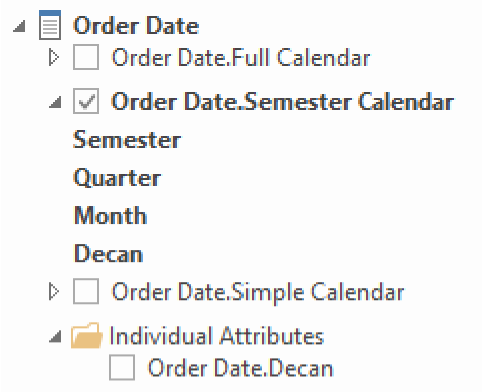
Decan DimensionAttribute

AttributeHierarchyDisplayFolder	**Individual Attributes**
AttributeHierarchyEnabled	True

4th test in Excel

- Process the cube using the same steps we've already used
- Switch to the Excel window
- In the *'Analyze'* tab inside Excel window, click the '*Refresh->Refresh All*' button
- In *'Pivot table'* fields, uncheck the *'Full Calendar'* hierarchy
- Check the *'Semester Calendar'* Hierarchy

Now we completed three calendar hierarchies and the *'Decan'* attribute.



Order Date
- □ Order Date.Full Calendar
- ☑ **Order Date.Semester Calendar**
  - **Semester**
  - **Quarter**
  - **Month**
  - **Decan**
- □ Order Date.Simple Calendar
- Individual Attributes
  - □ Order Date.Decan

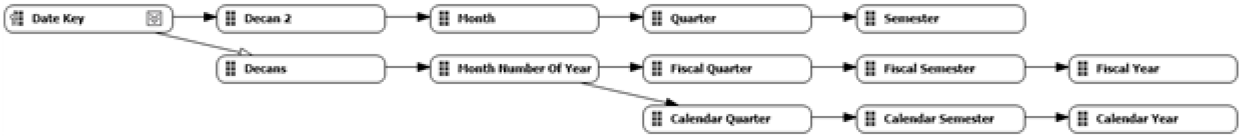
Fiscal Hierarchies

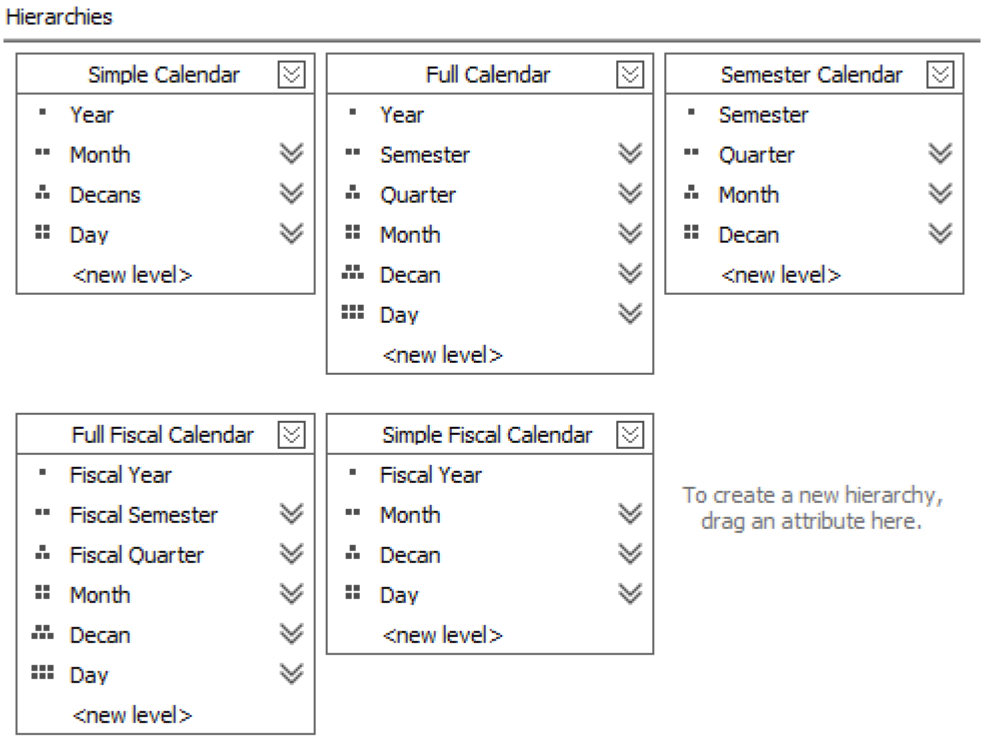
The next step is to create hierarchies for the fiscal calendar. The *'DimDate'* table in *'Data Source View'* area has several fields that allow us to create fiscal calendar hierarchies. We need two hierarchies: *'Simple Fiscal Calendar'* and *'Full Fiscal Calendar'*, following the same pattern that we created with *'Simple Calendar'* and *'Full Calendar'*. We don 't need a *'Fiscal Semester Calendar'* hierarchy, because it would be the same as the *'Semester Calendar'* hierarchy that we already have.

The creation of these hierarchies follows the same steps we used before, it's the same as the calendar hierarchies, so let's summarize the steps:

- We will need *'FiscalYear'*, *'FiscalSemester'* and *'FiscalQuarter'*fields
- We will use the same *'Month Number of Year'*, *'Decans'* and *'Date Key'* attributes, but we will rename them in the hierarchy.

- We will need two new named calculations, *'Fiscal Semester Name'* and *'Fiscal Quarter Name'*. The T-SQL instruction will be the same we used for 'CalendarSemesterName' and 'CalendarQuarterName', only changing the fields.
- We will need to configure the *'KeyColumns'* property and *'NameColumn'* property for *'Fiscal Semester'* and *'Fiscal Quarter'*attributes. *'AttributeHierarchyVisible'* property need to be configure for both plus *'Fiscal Year'*attribute.
- We will need to configure attribute relationship for the new attributes included in the dimension. The *'Fiscal Quarter'* attribute needs to be related with the *'Month Number of Year'* attribute, the *'Fiscal Semester'* attribute with the *'Fiscal Quarter'* attribute and the *'Fiscal Year'* attribute with *'Fiscal Semester'*




Hierarchies

Simple Calendar
Year
Month
Decans
Day
<new level>

Full Calendar
Year
Semester
Quarter
Month
Decan
Day
<new level>

Semester Calendar
Semester
Quarter
Month
Decan
<new level>

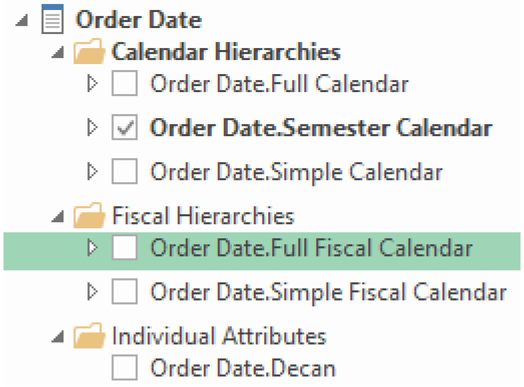
Full Fiscal Calendar
Fiscal Year
Fiscal Semester
Fiscal Quarter
Month
Decan
Day
<new level>

Simple Fiscal Calendar
Fiscal Year
Month
Decan
Day
<new level>

To create a new hierarchy, drag an attribute here.

Now we have five hierarchies. We can organize these hierarchies using *'Display Folder'* property of the hierarchies. Yes, surprise for you: Hierarchies also have properties. You can select a hierarchy and see the hierarchy properties in the *'Properties'* window

A good organization of the folders will be *'Simple Calendar'*, *'Full Calendar'* and *'Semester Calendar'* hierarchies with *'Display Folder'* property configured with the value 'Calendar Hierarchies ' and *'Simple Fiscal Calendar'* and *'Full Fiscal Calendar'* hierarchies with *'Display Folder'* property configured with the value *'Fiscal Calendar Hierarchies'*.



- **Order Date**
  - **Calendar Hierarchies**
    - □ Order Date.Full Calendar
    - ☑ **Order Date.Semester Calendar**
    - □ Order Date.Simple Calendar
  - Fiscal Hierarchies
    - □ Order Date.Full Fiscal Calendar
    - □ Order Date.Simple Fiscal Calendar
  - Individual Attributes
    - □ Order Date.Decan

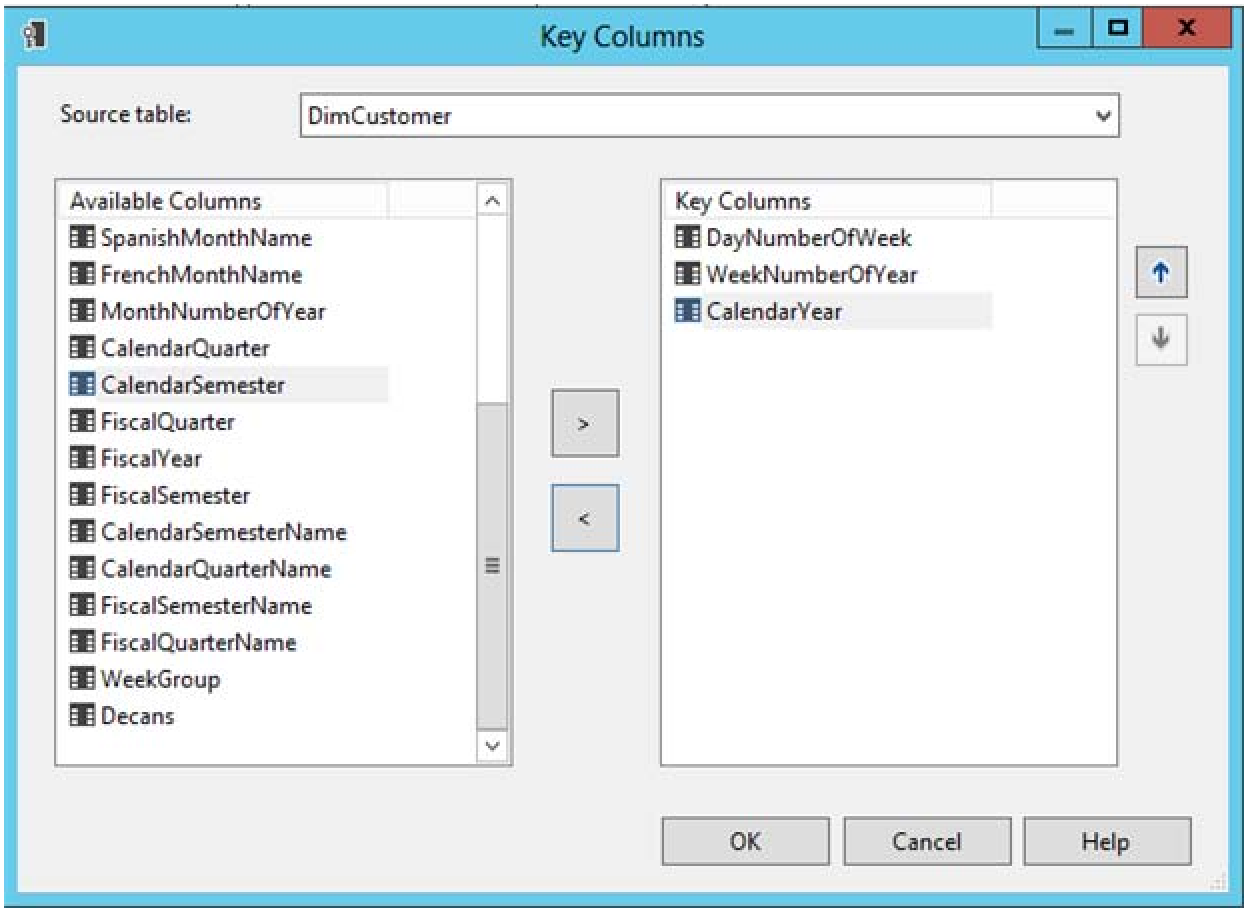
Week Hierarchy

The ‘Date Dim‘ table in AdventureWorks has a field named ‘Week Number of Year‘. Some companies do analysis with the week, so we can create hierarchies using Week. On the other hand, we shouldn ‘t include Semester, Quarter and Month in hierarchies with week, because about ten weeks in the year are part of two different months.

The week hierarchy will be, basically, the *‘Calendar Year’*, *‘Week Number of Year’* and *‘Week Day’* attributes.

Let’s create this new hierarchy using the following steps:

- Drag *‘Week Number of Year’* field to the Attributes Area
- Change *‘AttributeHierarchyVisible’* property of *‘Week Number of Year’* attribute to false
- Change *‘Key Columns’* property of *‘Week Number of Year’* attribute to *‘Week Number of Year’* + *‘Calendar Year’*
- Change the *‘NameColumn’* property of *‘Week Number of Year’* attribute to *‘Week Number of Year’*
- Drag the *‘DayNumberofWeek’* field to the Attributes Area
- Change the *‘NameColumn’* property of *‘Day Number of Week’* attribute to *‘EnglishDayNameofWeek’*
- Change *‘AttributeHierarchyVisible’* property of *‘Day Number of Week’* attribute to False
- Change *‘KeyColumns’* property of *‘Day Number of Week’* attribute to *‘Day Number of Week’* + *‘Week Number of Year’* + *‘Calendar Year’*



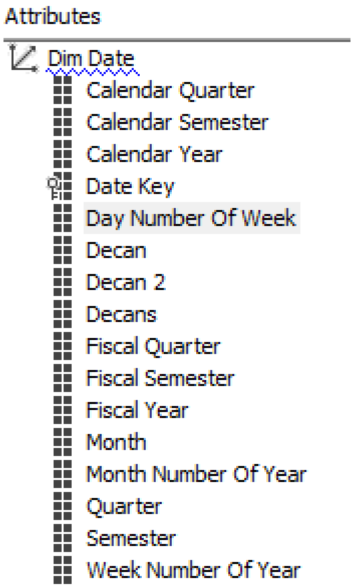
Key Columns

Source table: DimCustomer

Available Columns
SpanishMonthName
FrenchMonthName
MonthNumberOfYear
CalendarQuarter
CalendarSemester
FiscalQuarter
FiscalYear
FiscalSemester
CalendarSemesterName
CalendarQuarterName
FiscalSemesterName
FiscalQuarterName
WeekGroup
Decans

Key Columns
DayNumberOfWeek
WeekNumberOfYear
CalendarYear

- Change *‘Order By’* property of *‘Day Number of Week’* attribute to *‘Key’*

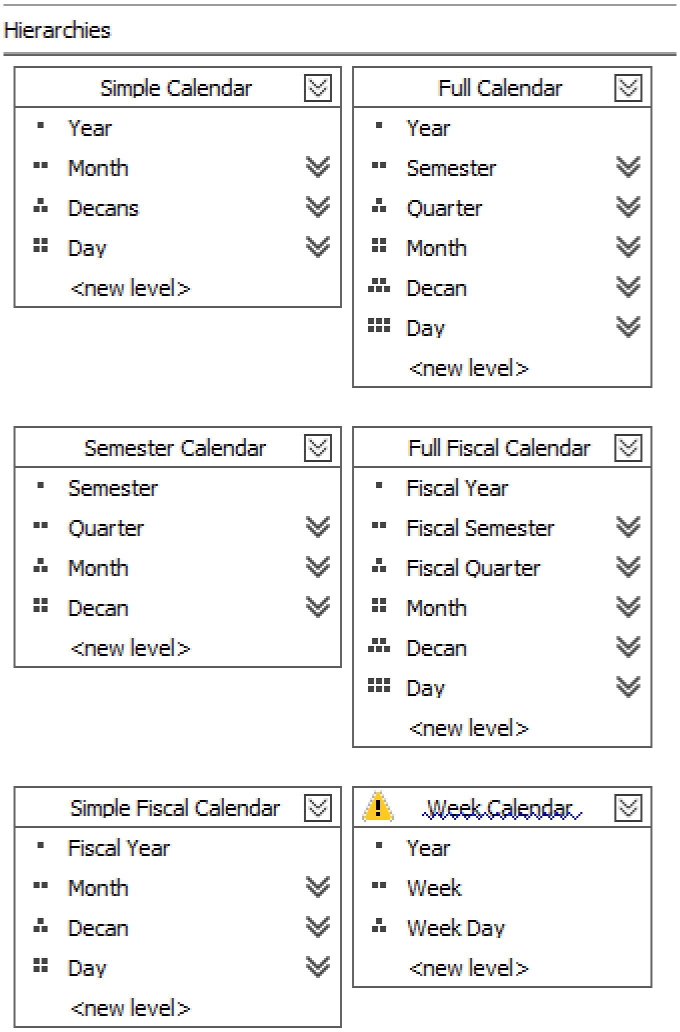


Attributes

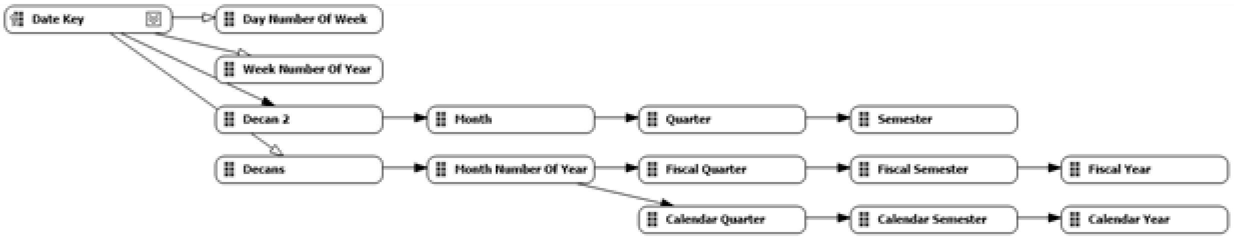
Dim Date
- Calendar Quarter
- Calendar Semester
- Calendar Year
- Date Key
- Day Number Of Week
- Decan
- Decan 2
- Decans
- Fiscal Quarter
- Fiscal Semester
- Fiscal Year
- Month
- Month Number Of Year
- Quarter
- Semester
- Week Number Of Year

- Right-click *‘Calendar Year’* attribute and click *‘Start new Hierarchy’* menu item
- Drag *‘Week Number of Year’* attribute to the new hierarchy
- Drag *‘Day Number of Week’* attribute to the new hierarchy
- Rename the Hierarchy members to *‘Year’*, *‘Week’* and *‘Week Day’*

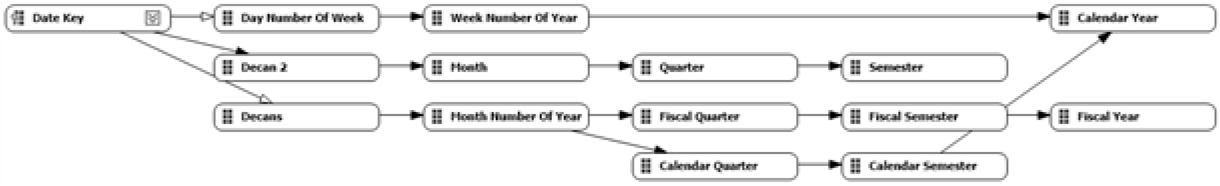
- Right-click the new hierarchy and rename it as *‘Week Calendar’*
- With *‘Week Calendar’* selected, change *‘Display Folder’* property to *‘Week Hierarchies’*



- Select Attribute Relationships tab



- Right-click *‘Day Number of Week’* attribute and create a new attribute Relationship to *‘Week Number of Year’* attribute , type *‘Rigid’*
- Right-click *‘Week Number of Year’* attribute and create a new attribute Relationship to *‘Calendar Year’* attribute, type *‘Rigid’*

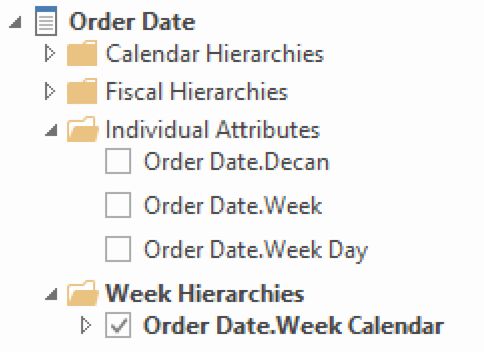


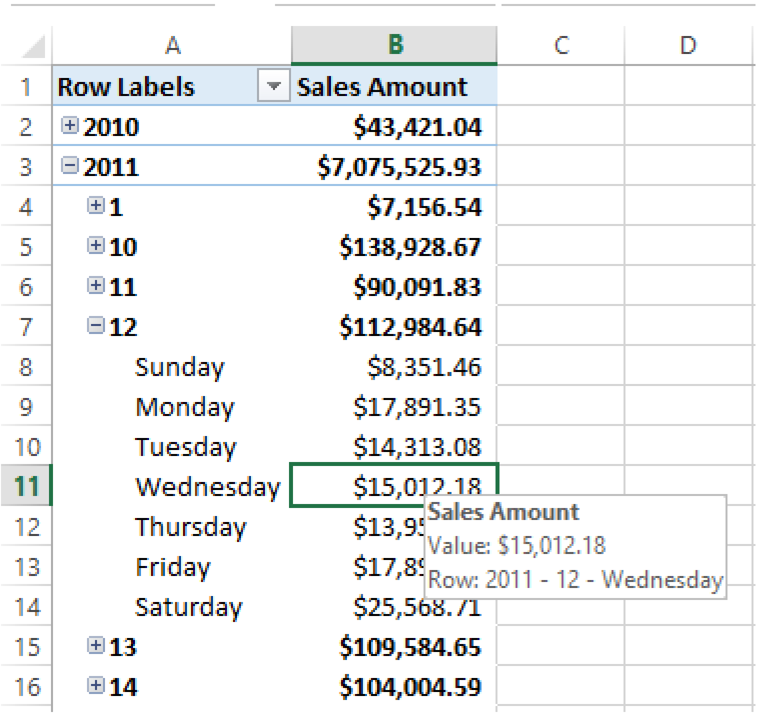
We still need to allow the individual analysis by *'Week Number of Year'* attribute and *'Day Number of* Week' attribute. Let 's do it:

- Drag *'Week Number of Year'* to attributes area
- Rename it as *'Week'*
- Change *'AttributeHierarchyDisplayFolder'* property of 'Week ' to *'Individual Attributes'*
- Drag *'Day Number of Week'* attribute to the attributes area
- Rename it as *'Week Day'*
- Change *'AttributeHierarchyDisplayFolder'* property of *'Week Day*' to *'Individual Attributes'*
- Change *'NameColumn'* property of *'Week Day'* attribute to *'EnglishDayNameofWeek'*
- Change *'OrderBy'* property of *'Week Day'* attribute to *'Key'*

5th test in Excel

- Process the cube using the same steps we already did before
- Switch to Excel
- Click Refresh->Refresh All button in the *'Analyze'* tab
- Check the new hierarchy and attributes in *'Pivot Table Fields'* area





	A	B	C	D
1	Row Labels	Sales Amount		
2	⊞2010	$43,421.04		
3	⊟2011	$7,075,525.93		
4	⊞1	$7,156.54		
5	⊞10	$138,928.67		
6	⊞11	$90,091.83		
7	⊟12	$112,984.64		
8	Sunday	$8,351.46		
9	Monday	$17,891.35		
10	Tuesday	$14,313.08		
11	Wednesday	$15,012.18		
12	Thursday	$13,9		
13	Friday	$17,8		
14	Saturday	$25,568.71		
15	⊞13	$109,584.65		
16	⊞14	$104,004.59		

Sales Amount
Value: $15,012.18
Row: 2011 - 12 - Wednesday

Using Perspectives to control Hierarchy visualization

Perspectives are like views for an entire cube. A perspective determines which parts of a cube a user can view when using the perspective.

For some users the excess of hierarchies can be confusing. We can use perspectives to create different views of the cube, simplifying the user work.

Perspectives are available in Business Intelligence edition and Enterprise Edition, but not in Standard Edition.

In our example, we have plenty of possibilities for creating perspectives of the cube. Let 's try the following combination:

Simple Perspective: This perspective can have the *'Simple Calendar'* hierarchy and *'Semester Calendar'* hierarchy

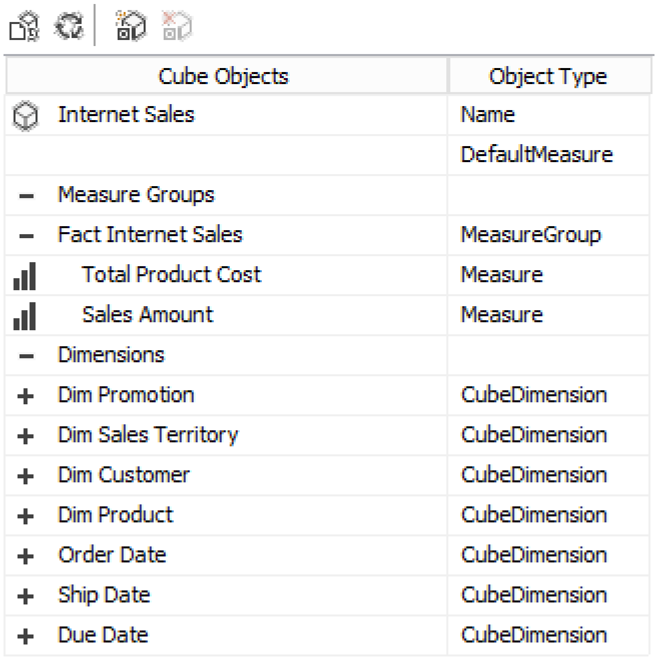
Extended Perspective: The same as the simple perspective plus the *'Full Calendar'* hierarchy

Fiscal Perspective: This perspective will contain the *'Simple Fiscal Calendar'* and *'Full Fiscal Calendar'* hierarchies, *'Semester Calendar'* and *'Week Calendar'* hierarchies

Let 's create the perspectives:

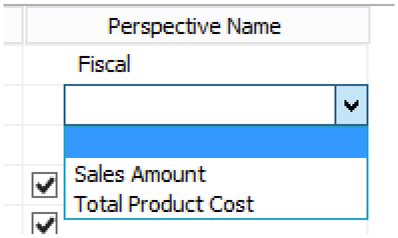
- Switch to Visual Studio window

- Select *‘Internet Sales.cube’* tab
- Select *‘Perpectives’* tab



	Cube Objects	Object Type
	Internet Sales	Name
		DefaultMeasure
−	Measure Groups	
−	Fact Internet Sales	MeasureGroup
	Total Product Cost	Measure
	Sales Amount	Measure
−	Dimensions	
+	Dim Promotion	CubeDimension
+	Dim Sales Territory	CubeDimension
+	Dim Customer	CubeDimension
+	Dim Product	CubeDimension
+	Order Date	CubeDimension
+	Ship Date	CubeDimension
+	Due Date	CubeDimension

- Click *‘New Perspective’* button in the toolbar
- Change the perspective name to *‘Fiscal’*
- In *‘Default Measure’* row, select *‘Sales Amount’* measure



Perspective Name

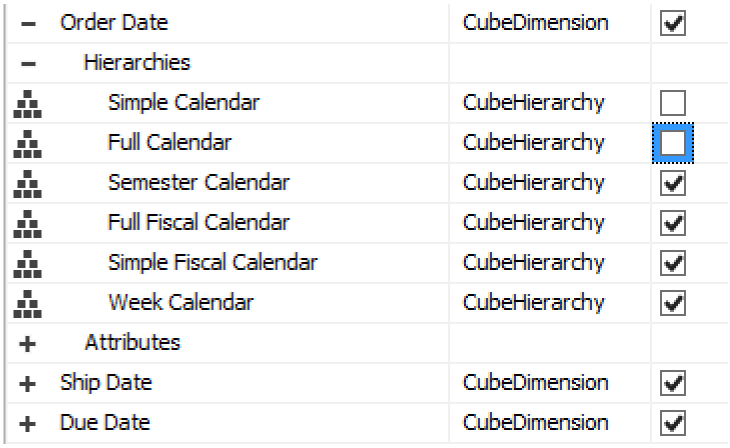
Fiscal

☑ Sales Amount

☑ Total Product Cost

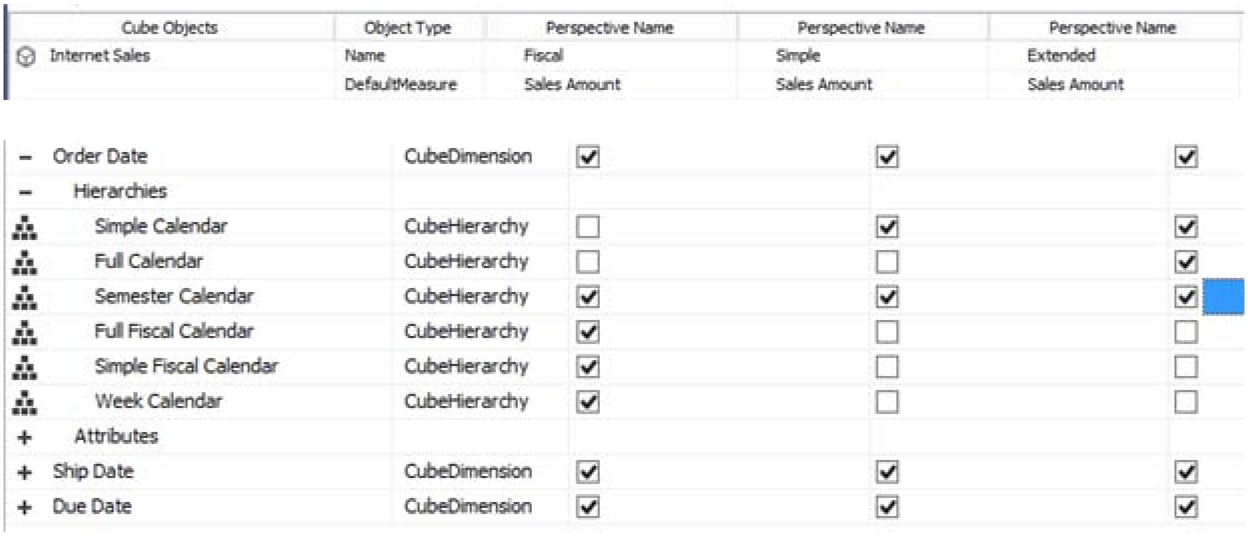
The default measure is the measure used when we do queries against the cube without specifying a measure.

- Expand the ‘order date‘ dimension.
- Uncheck the ‘Simple Calendar’ and ‘Full Calendar’ hierarchies



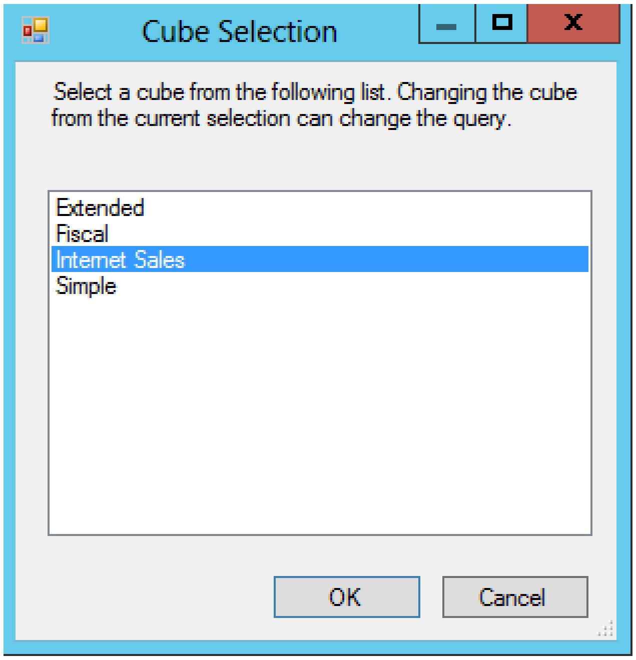
− Order Date	CubeDimension	☑
− Hierarchies		
Simple Calendar	CubeHierarchy	☐
Full Calendar	CubeHierarchy	☐
Semester Calendar	CubeHierarchy	☑
Full Fiscal Calendar	CubeHierarchy	☑
Simple Fiscal Calendar	CubeHierarchy	☑
Week Calendar	CubeHierarchy	☑
+ Attributes		
+ Ship Date	CubeDimension	☑
+ Due Date	CubeDimension	☑

- Repeat this for *‘Ship Date’* and *‘Due Date’* dimensions, unchecking the same hierarchies
- Repeat these steps for *‘Simple’* and *‘Extended’* perspectives, choosing the correct hierarchies for each perspective



Cube Objects	Object Type	Perspective Name	Perspective Name	Perspective Name
Internet Sales	Name	Fiscal	Simple	Extended
	DefaultMeasure	Sales Amount	Sales Amount	Sales Amount
− Order Date	CubeDimension	☑	☑	☑
− Hierarchies				
Simple Calendar	CubeHierarchy	☐	☑	☑
Full Calendar	CubeHierarchy	☐	☐	☑
Semester Calendar	CubeHierarchy	☑	☑	☑
Full Fiscal Calendar	CubeHierarchy	☑	☐	☐
Simple Fiscal Calendar	CubeHierarchy	☑	☐	☐
Week Calendar	CubeHierarchy	☑	☐	☐
+ Attributes				
+ Ship Date	CubeDimension	☑	☑	☑
+ Due Date	CubeDimension	☑	☑	☑

- Select *‘Browse’* tab
- Process the cube
- Click *‘Refresh’* button on the toolbar
- Click *‘..’* button besides the *‘Internet Sales’* cube, above the *‘Metadata’* window

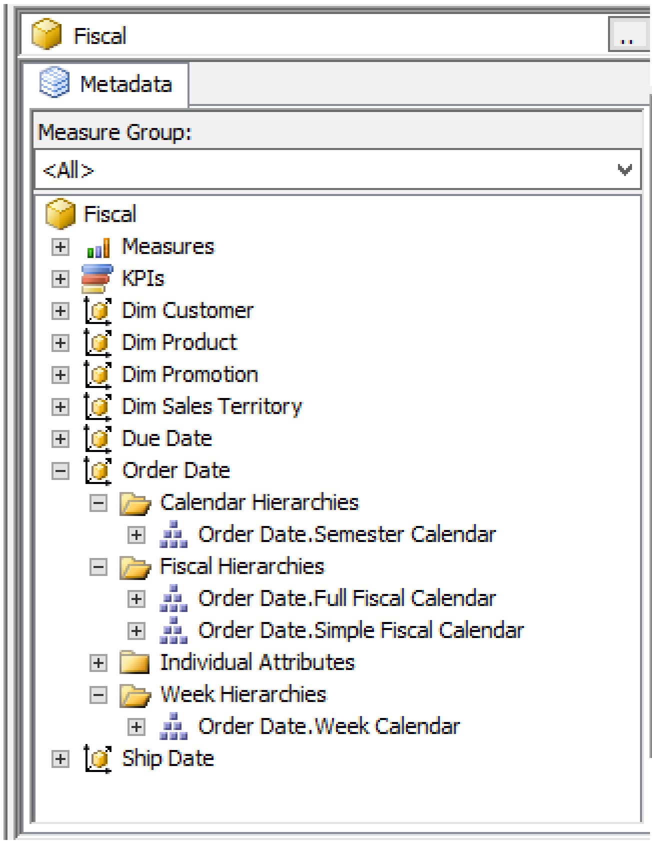


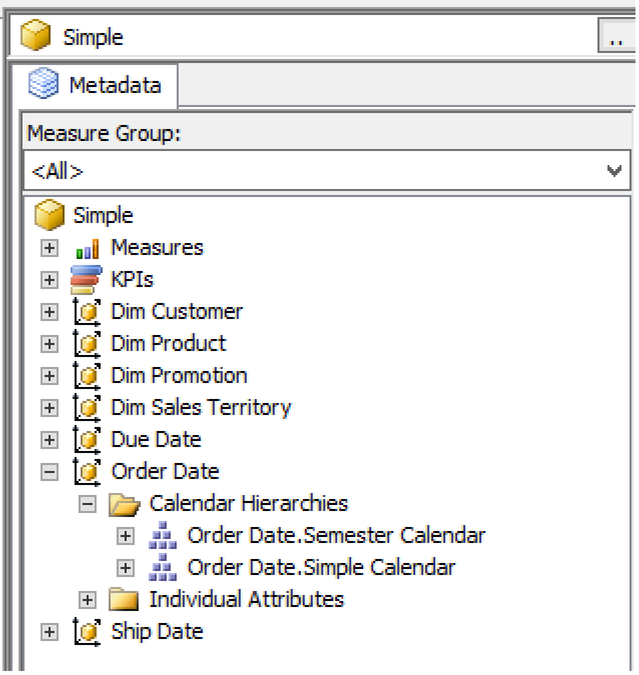
Cube Selection

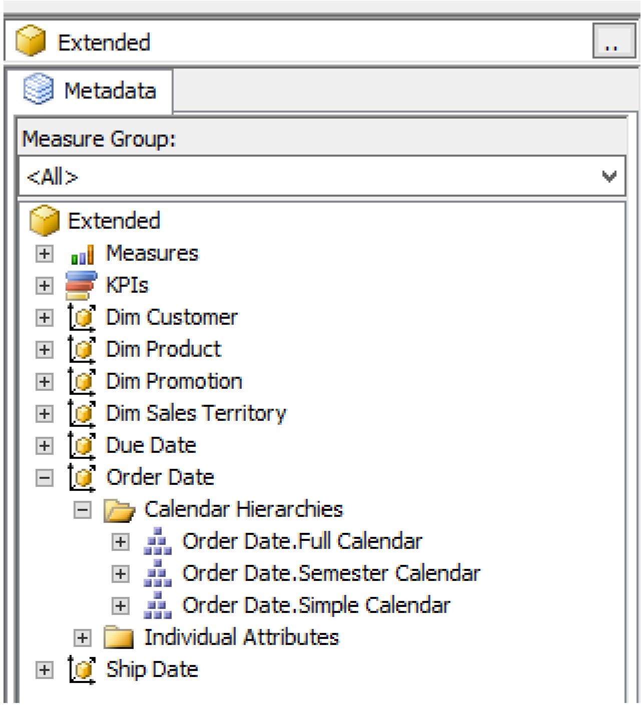
Select a cube from the following list. Changing the cube from the current selection can change the query.

Extended
Fiscal
Internet Sales
Simple

- Each perspective appears like a different cube inside *'Cube Selection'* window. You can try each perspective and identify the visible dimensions and hierarchies.







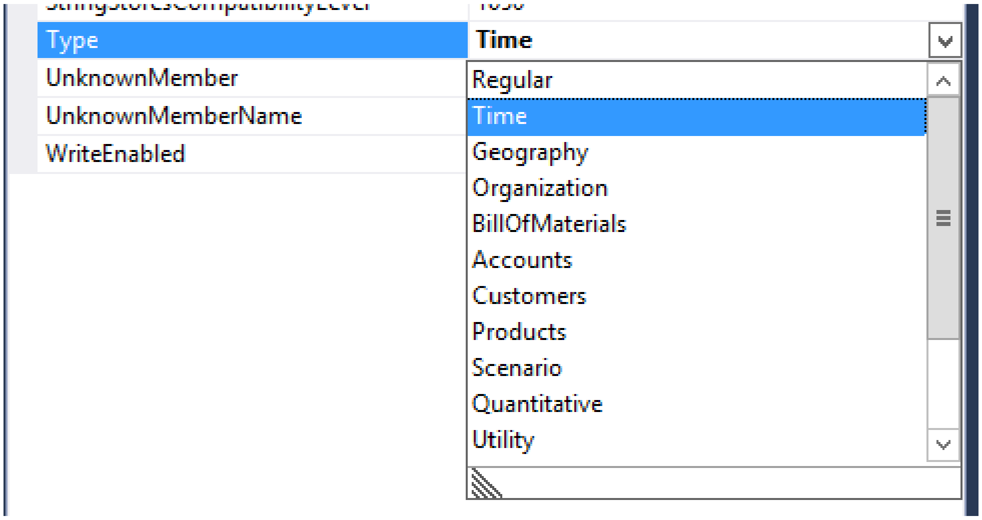
The Dimension Type

Our last, but not least, task is to specify the dimension type.

There are some analysis we can do using MDX queries that needs the dimension type specified. For example, there is a function named *‘ParallelPeriod’* that allow us to analyze periods across the time, for example, the same month across several years.

For this function, *‘ParallelPeriod’*, and several others can work properly, there must be a dimension marked to be of type *‘Time’*.

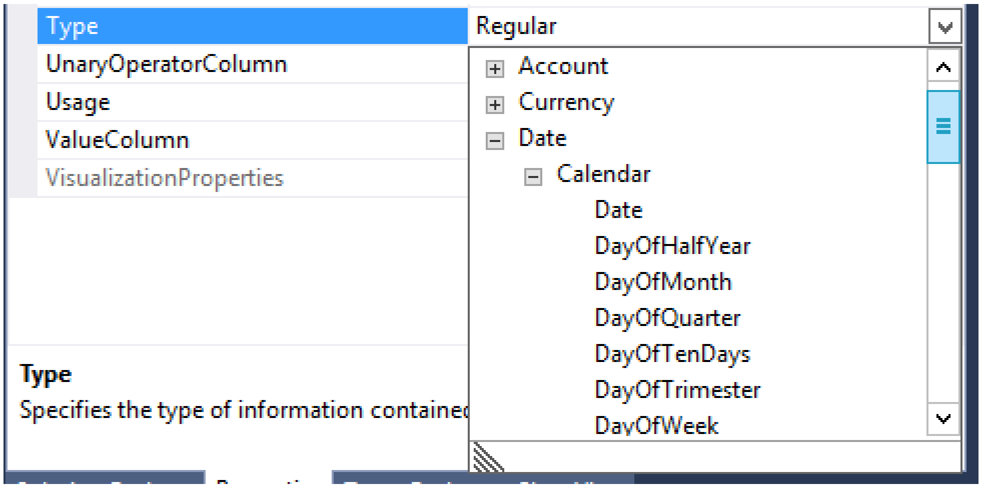
It‘s easy configure this: In properties window, with the *‘DimDate’* selected, change the *‘Type’* property.



Type Time

The attributes in the hierarchy also have a *'Type'* property. We need to identify what each attribute means inside our dimension.

For example, we can select the *'Date Key'* attribute and configure the *'Type'* property as 'Calendar Date ' (Date->Calendar->Date), as you can see in the image below:



Type | Regular

UnaryOperatorColumn

Usage

ValueColumn

VisualizationProperties

Type

Specifies the type of information containe

- Account
- Currency
- Date
  - Calendar
    - Date
    - DayOfHalfYear
    - DayOfMonth
    - DayOfQuarter
    - DayOfTenDays
    - DayOfTrimester
    - DayOfWeek

Below you can see a table with the relation between our attributes and the *'Type'* property values:

Attribute	***'Type'*** **property values**
Monthly Number of Year	Calendar->MonthOfYear
Calendar Quarter	Calendar->QuarterOfYear
Calendar Year	Calendar->Years
Day Number of Week	Calendar->DayOfWeek
Fiscal Quarter	Fiscal->QuarterOfYear
Fiscal Year	Fiscal->Year
Week Number of Year	Calendar->WeekOfYear
Week Day	Calendar->DayOfWeek

What's next

Now that you've seen a bunch of tricks to build a good date dimension for your cubes, we'll see what's next:

- I didn‘t use translations for the dimension. Some companies need to use the cube globally, across countries with different languages. I wrote about translations here
- There are a lot more tricks for dimensions that we didn‘t used for this date dimension. You can see more here.
- You can also implement a dimension like this using the tabular model

> Note: the sample application is in a ZIP file, DateDimension.ZIP, with a link at the head of the article.

□ Downloads

DateDimension.zip (132 kB)

Tags: Analysis #Services, BI, SQL, SQL Server

20108 views Rate ★★★★★

Dennes Torres

Dennes is a software architect who loves SQL Server and software development and has more than 20 years of experience. He created the devSQL PASS Chapter in Rio de Janeiro/Brazil and since 2003 organizes meetings, events and webcasts about SQL Server, including SQL Saturdays. He is an MCT, MCSE in Data Platforms and BI, with more titles in software development. He can be found writing for his blog at http://cidadaocarioca.blogspot.com or snorkeling in Arraial do Cabo, a diving paradise near Rio de Janeiro.

View all articles by Dennes Torres

7 Comments Simple Talk □ Login

□ Recommend □ Share Sort by Newest

Join the discussion…

Meg • 6 months ago

Thanks for your excellent article.

□ | □ • Reply • Share ›

Steve • 7 months ago

Excellent article Dennes! I'm just starting out with SSAS and this has helped me greatly.
I have one question regarding the article, when we created the Full Calendar hierarchy and updated the Attribute Relationships, i notice that we didn't update the Key columns for Calendar Month of Year

to include Calendar Quarter, was this by design or just a missing step in the article?

□ | □ • Reply • Share ›

Meg > Steve • 6 months ago

"year.month" is already unique, we don't have to do "year.quarter.month".

□ | □ • Reply • Share ›

subhraz_gh • 9 months ago

What a wonderful and detailed article. Thanks a lot Dennes for your effort.
Helped me to understand SSAS hierarchy.

□ | □ • Reply • Share ›

Matt B • a year ago

Nice Job!
Awesome article. Very thorough.

□ | □ • Reply • Share ›

Dennes Torres • a year ago

Top rated recent articles in BI

Comparing SSIS Catalog Contents Using DBFit Framework

by Nat Sundar ★★★★★ 0

Generating Plots Automatically From PowerShell and SQL Server Using Gnuplot

by Phil Factor ★★★★★ 0

Statistics in SQL: Kendall’s Tau rank correlation

by Phil Factor ★★★★★ 2

Statistics in SQL: Simple Linear Regressions

by Phil Factor ★★★★★ 0

Scala and Apache Spark in Tandem as a Next-Generation ETL Framework

by Krzysztof Stanaszek ★★★★★ 2

Related articles

ALSO IN ANALYSIS #SERVICES

Creating a Date Dimension in a Tabular Model

As well as its multidimensional model, SQL Server Analysis Services (SSAS) now has a tabular model of database that either runs in-memory or in DirectQuery mode. The in-memory analytics engine allows the users of Excel or Power View to base reports on tabular model objects. Having shown how to handle date-based information using the Multi-dimensional model, Dennes now turns his attention on the in-memory tabular model.… Read more

ALSO IN BI

Relational Algebra and its implications for NoSQL databases

With the rise of NoSQL databases that are exploiting aspects of SQL for querying, and are embracing full transactionality, is there a danger of the data-document model's hierarchical nature causing a fundamental conflict with relational theory? We asked our relational expert, Hugh Bin-Haad to expound a difficult area for database theorists.… Read more

ALSO IN SQL

SQL Server System Functions: The Basics

Every SQL Server Database programmer needs to be familiar with the System Functions. These range from the sublime (such as @@rowcount or @@identity) to the ridiculous (IsNumeric()) Robert Sheldon provides an overview of the most commonly used of them.… Read more

ALSO IN SQL SERVER

The SQL Server 2016 Query Store: Forcing Execution Plans using the Query Store

The SQL Server 2016 Query Store can give you valuable performance insights by providing several new ways of troubleshooting queries, studying their plans, exploring their context settings, and checking their performance metrics. However, it can also directly affect the performance of queries by forcing Execution Plans for specific queries.… Read more

FAQ

Sitemap

Privacy Policy

Write For Us

Contact Us

What do you think of the new Simple Talk? Give us your feedback