Microsoft SQL Server Lab 2 of 2
Intro to SQL Server Analysis Services (SSAS)

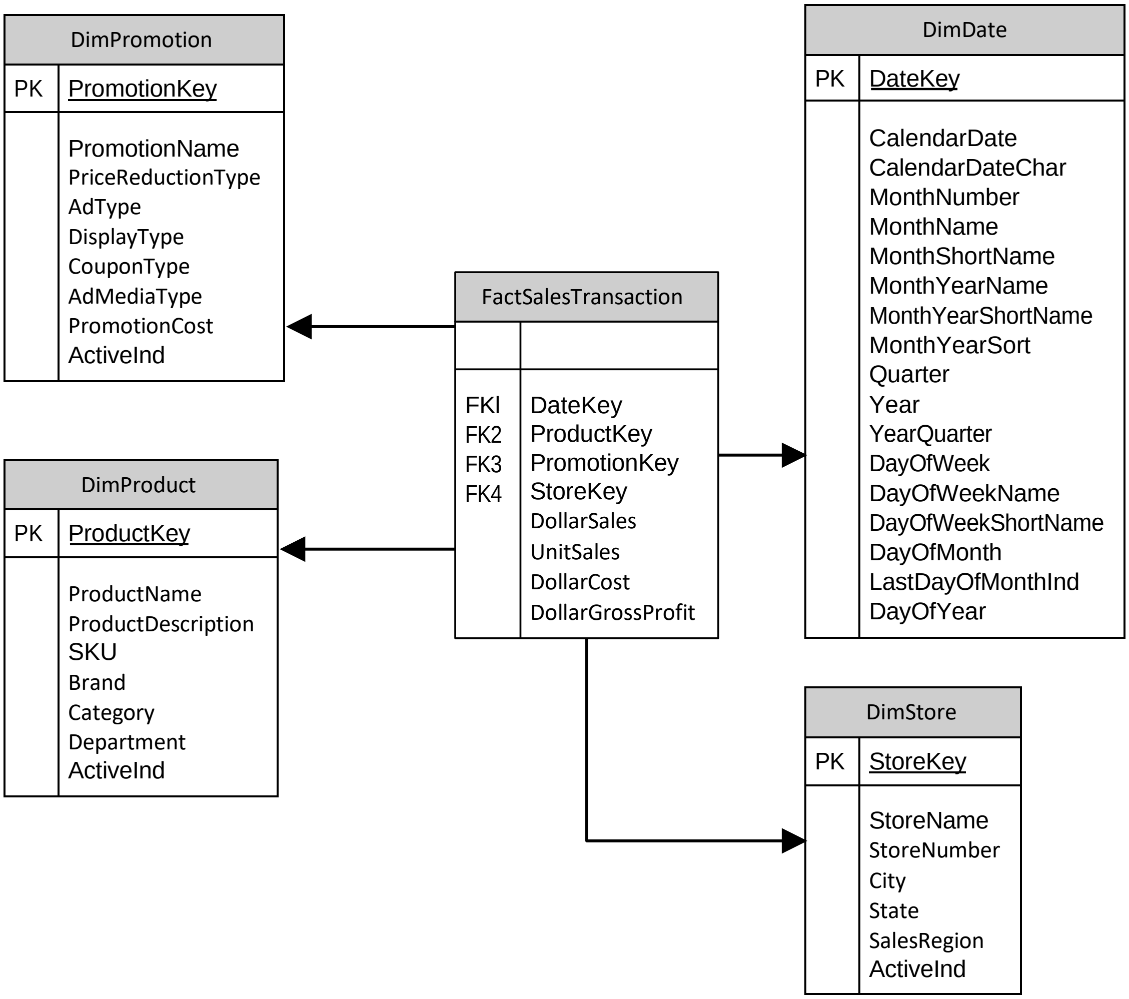
Introduction

This lab is intended to introduce you to another tool that is often used in data warehouse projects. As part of its SQL Server 2012 suite, Microsoft contains Analysis Services which is provides OLAP (On-Line Analytical Processing) capabilities.

As we discussed in class, OLAP provides the ability to perform various operations on multidimensional data "cubes" – operations like drill-up, drill-down, pivot and other "slice-and-dice" methods. The goal of this lab is to show you how Analysis Services OLAP databases are created, and to give you some preliminary exposure to how the cubes operate.

Database Schema

We will continue working with a refined version of the data model we have used so far for retail store transactions. The data model is shown below.



DimPromotion

PK	PromotionKey
	PromotionName PriceReductionType AdType DisplayType CouponType AdMediaType PromotionCost ActiveInd

DimDate

PK	DateKey
	CalendarDate CalendarDateChar MonthNumber MonthName MonthShortName MonthYearName MonthYearShortName MonthYearSort Quarter Year YearQuarter DayOfWeek DayOfWeekName DayOfWeekShortName DayOfMonth LastDayOfMonthInd DayOfYear

FactSalesTransaction

FKI FK2 FK3 FK4	DateKey ProductKey PromotionKey StoreKey DollarSales UnitSales DollarCost DollarGrossProfit

DimProduct

PK	ProductKey
	ProductName ProductDescription SKU Brand Category Department ActiveInd

DimStore

PK	StoreKey
	StoreName StoreNumber City State SalesRegion ActiveInd

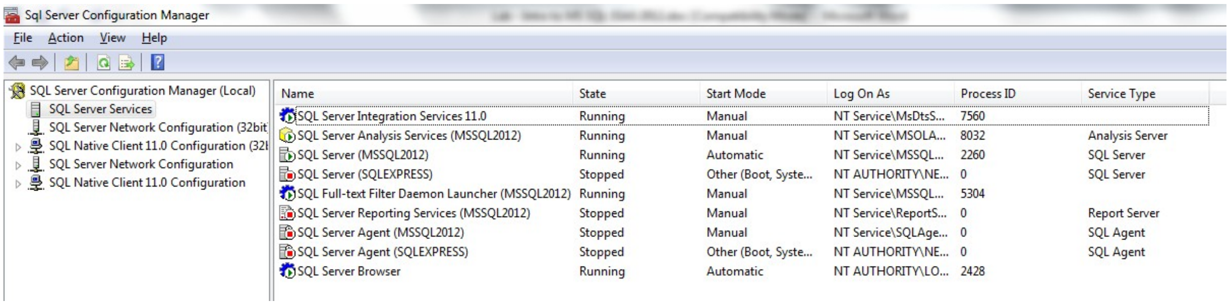
Extract the Lab Files

The files for this lab are contained in a ZIP file called “Intro_SSAS_Data.zip”. Extract the contents of this file into a folder that is located in a convenient place on your machine (e.g., C:\Temp). This folder will be referred to subsequently as the *Lab* folder.

Create the Database

In this lab, we will begin with a star-schema database that has already been loaded for you. You must “restore” this database from a SQL Server backup file before you can create an OLAP cube.

1. Make sure both your SQL Server database instance and SSIS instance are running. To do this, go to Start -> Programs -> Microsoft SQL Server 2012 -> Configuration Tools -> SQL Server Configuration Manager.



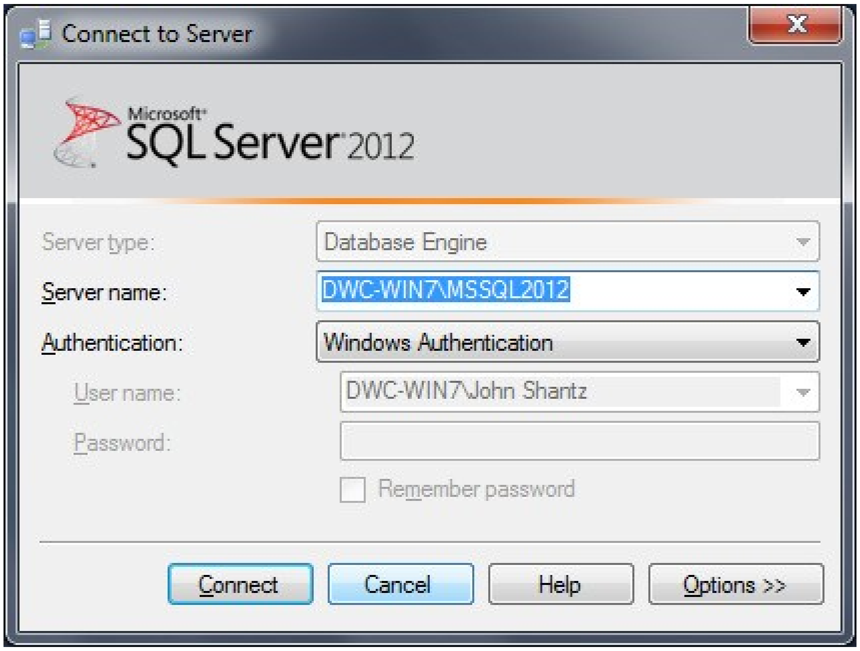
Sql Server Configuration Manager
File Action View Help
SQL Server Configuration Manager (Local)
SQL Server Services
SQL Server Network Configuration (32bit
SQL Native Client 11.0 Configuration (32l
SQL Server Network Configuration
SQL Native Client 11.0 Configuration

Name	State	Start Mode	Log On As	Process ID	Service Type
SQL Server Integration Services 11.0	Running	Manual	NT Service\MsDtsS...	7560	
SQL Server Analysis Services (MSSQL2012)	Running	Manual	NT Service\MSOLA...	8032	Analysis Server
SQL Server (MSSQL2012)	Running	Automatic	NT Service\MSSQL...	2260	SQL Server
SQL Server (SQLEXPRESS)	Stopped	Other (Boot, Syste...	NT AUTHORITY\NE...	0	SQL Server
SQL Full-text Filter Daemon Launcher (MSSQL2012)	Running	Manual	NT Service\MSSQL...	5304	
SQL Server Reporting Services (MSSQL2012)	Stopped	Manual	NT Service\ReportS...	0	Report Server
SQL Server Agent (MSSQL2012)	Stopped	Manual	NT Service\SQLAge...	0	SQL Agent
SQL Server Agent (SQLEXPRESS)	Stopped	Other (Boot, Syste...	NT AUTHORITY\NE...	0	SQL Agent
SQL Server Browser	Running	Automatic	NT AUTHORITY\LO...	2428	

   Make sure the SQL Server (MSSQL2012), SQL Server Analysis Services (MSSQL2012) and SQL Server Browser show a state of running. If not, start the services. Leave this window open as you will need information from here in a later step. **Screenshot Here**

2. Connect to SQL Server Management Studio. To do this, go to Start -> Programs -> Microsoft SQL Server 2012 -> SQL Server Management Studio.

3. Choose a “Database Engine” Server Type and Windows Authentication. For Server Name enter a string with like this [Host Name]\[Database Name] where [Host Name] is your computer name and Database Name is something like MSSQL2012.



Server type: Database Engine

Server name: DWC-WIN7\MSSQL2012

Authentication: Windows Authentication

User name: DWC-WIN7\John Shantz

Password:

□ Remember password

4. Right-click on the Databases folder and choose “Restore Database…”

5. Type in “CMURetailSales” for the database name on the “To database” line.

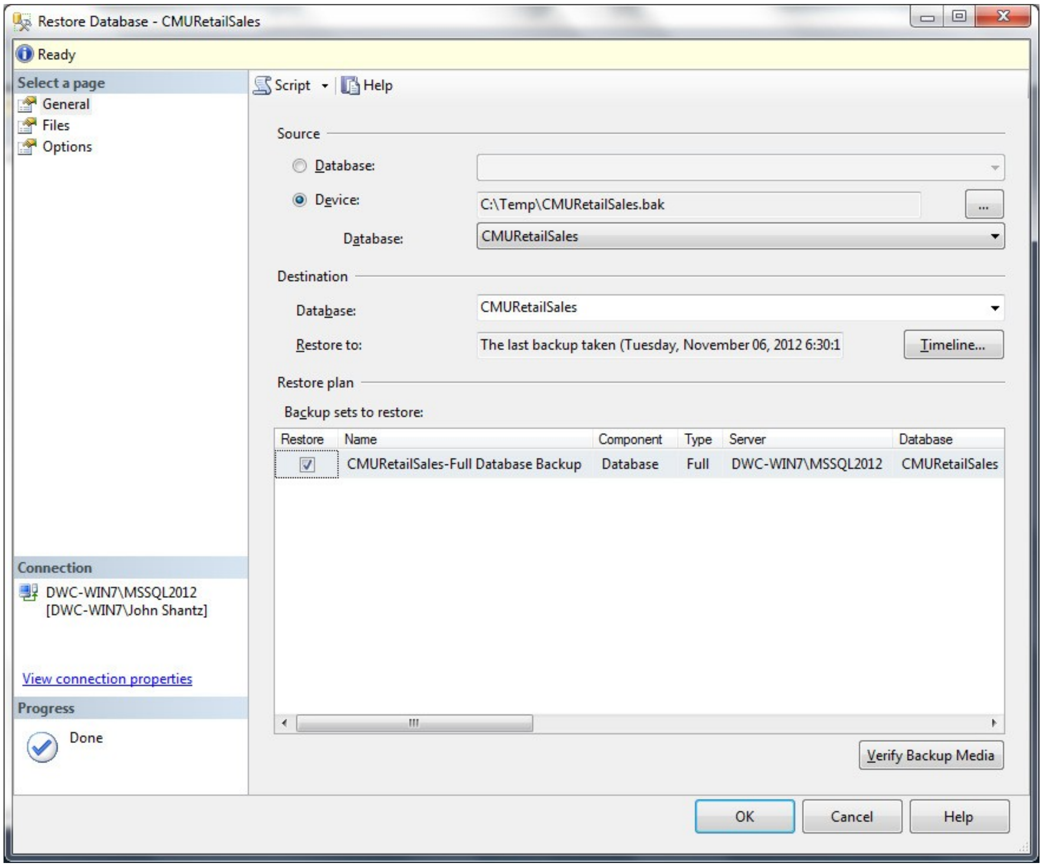
6. In the bottom of the dialog box, select “From device” and then click the button at the right with three dots.

   Tip: Select "Device" radio button under "Source" section.

7. Click Add and then navigate to your Lab folder. Navigate to the file named *CMURetailSales.bak* and select it. Click OK. **Screenshot Here**

8. In the “Specify Backup” window, click OK again.

9. Make sure the checkbox next to the file you just selected is checked. Your screen should now look something like this.



Source

○ Database:

● Device: C:\Temp\CMURetailSales.bak

Database: CMURetailSales

Destination

Database: CMURetailSales

Restore to: The last backup taken (Tuesday, November 06, 2012 6:30:1

Restore plan

Backup sets to restore:

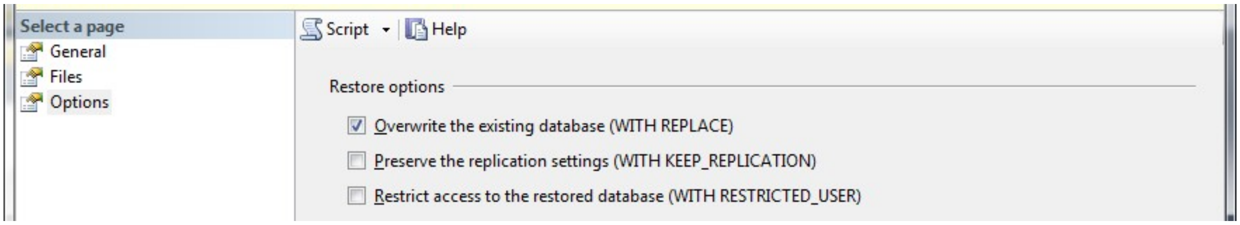
Restore	Name	Component	Type	Server	Database
☑	CMURetailSales-Full Database Backup	Database	Full	DWC-WIN7\MSSQL2012	CMURetailSales

Connection: DWC-WIN7\MSSQL2012 [DWC-WIN7\John Shantz]

Progress: Done

10. Set the database options. Your installation of SQL Server may have default options set that cause errors on the import. To prevent these errors, verify the following two options by clicking on the “Options” tab in the menu at the left.

a. Check the box that says “Overwrite the existing database (WITH REPLACE)”.



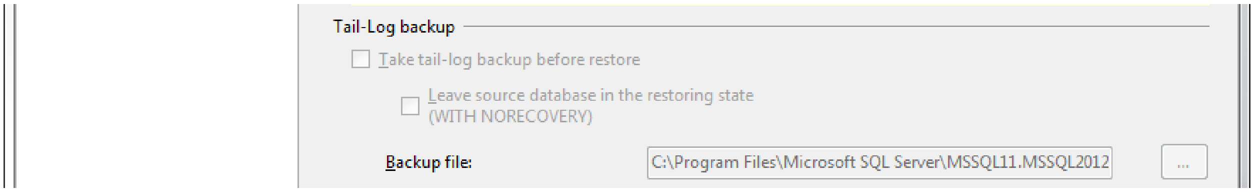
Restore options

☑ Overwrite the existing database (WITH REPLACE)

☐ Preserve the replication settings (WITH KEEP_REPLICATION)

☐ Restrict access to the restored database (WITH RESTRICTED_USER)

b. Uncheck the box that says "Take tail-log backup before restore."



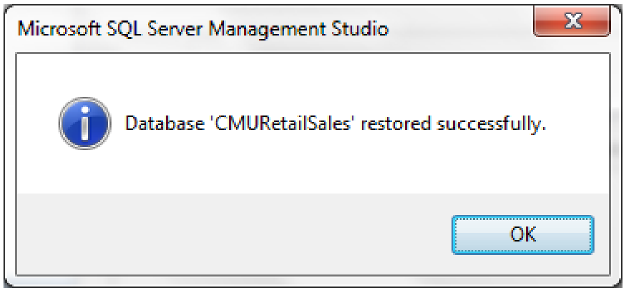
Tail-Log backup

☐ Take tail-log backup before restore

☐ Leave source database in the restoring state (WITH NORECOVERY)

Backup file: C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQL2012

11. Click OK. You should get a dialog box like this. **Screenshot Here**

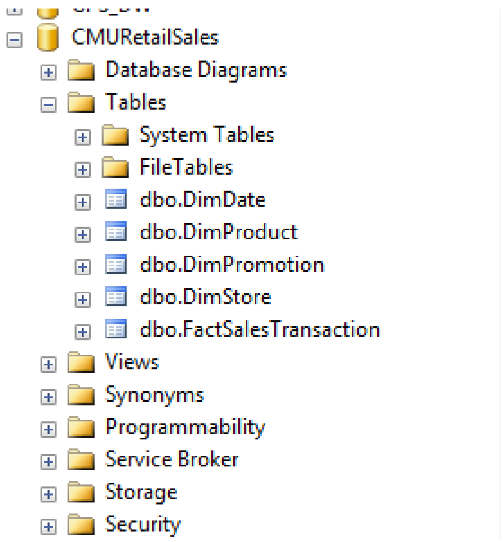


Microsoft SQL Server Management Studio

Database 'CMURetailSales' restored successfully.

Tip: If an error occurs (database in use), disconnect from the SQL Server Management Studio and connect again

12. On the left panel you should now see the "CMURetailSales" database under the databases folder. Expand this database and then expand the Tables folder beneath it. You should see five tables like the picture below. At this point, you are ready to begin working with Analysis Services. **Screenshot Here**



```
CMURetailSales
  Database Diagrams
  Tables
    System Tables
    FileTables
    dbo.DimDate
    dbo.DimProduct
    dbo.DimPromotion
    dbo.DimStore
    dbo.FactSalesTransaction
  Views
  Synonyms
  Programmability
  Service Broker
  Storage
  Security
```

13. Determine your OLAP user name. Now that you have imported the database, you need to make sure the OLAP user can access it. But first we need to determine the OLAP user name for your computer. To do this, go back to the SQL Server Configuration Manager. Look for the user name that is entered in the “Log On As” column of the SQL Server Analysis Services instance. This is the username you need.

Tip: Use SQL Server (MSSQL201X) instead. Latest version of SQL Server do not have Analysis Services in this screen.

NOTE: Newer versions of Windows won't show the SQL Server Configuration Manager in the menu.

If you can't find SQL Server Configuration Manager, try the locations below:

SQL Server 2019:
C:\Windows\SysWOW64\SQLServerManager15.msc

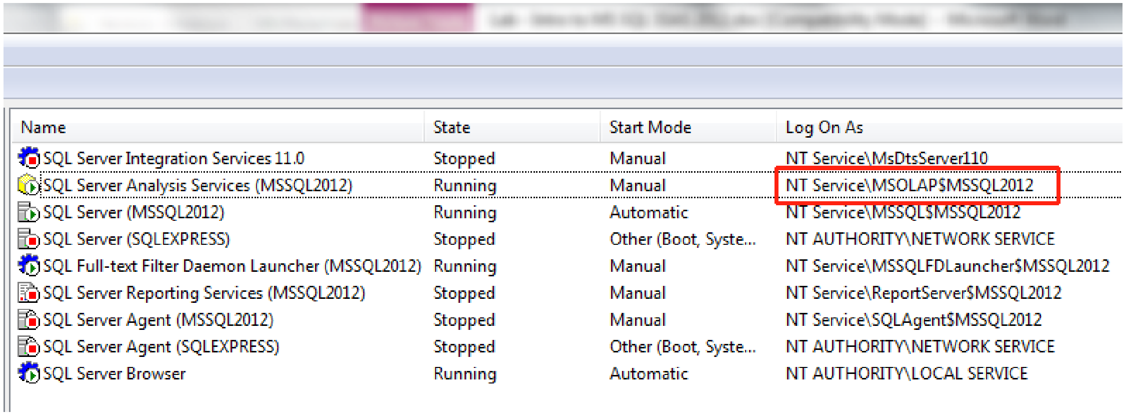
SQL Server 2017:
C:\Windows\SysWOW64\SQLServerManager14.msc

SQL Server 2016:
C:\Windows\SysWOW64\SQLServerManager13.msc

SQL Server 2014:
C:\Windows\SysWOW64\SQLServerManager12.msc

SQL Server 2012:
C:\Windows\SysWOW64\SQLServerManager11.msc

Source:
https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/start-stop-pause-resume-restart-sql-server-services?view=sql-server-ver15#sql-server-configuration-manager

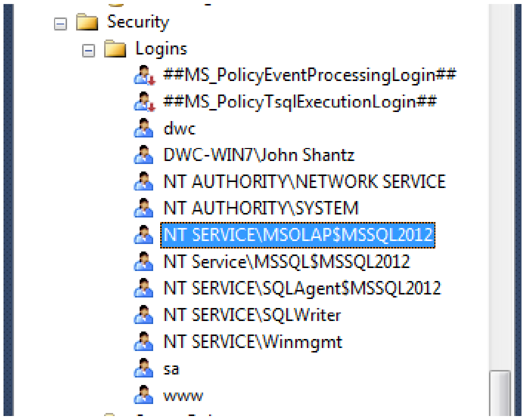


Name	State	Start Mode	Log On As
SQL Server Integration Services 11.0	Stopped	Manual	NT Service\MsDtsServer110
SQL Server Analysis Services (MSSQL2012)	Running	Manual	NT Service\MSOLAP$MSSQL2012
SQL Server (MSSQL2012)	Running	Automatic	NT Service\MSSQL$MSSQL2012
SQL Server (SQLEXPRESS)	Stopped	Other (Boot, Syste...	NT AUTHORITY\NETWORK SERVICE
SQL Full-text Filter Daemon Launcher (MSSQL2012)	Running	Manual	NT Service\MSSQLFDLauncher$MSSQL2012
SQL Server Reporting Services (MSSQL2012)	Stopped	Manual	NT Service\ReportServer$MSSQL2012
SQL Server Agent (MSSQL2012)	Stopped	Manual	NT Service\SQLAgent$MSSQL2012
SQL Server Agent (SQLEXPRESS)	Stopped	Other (Boot, Syste...	NT AUTHORITY\NETWORK SERVICE
SQL Server Browser	Running	Automatic	NT AUTHORITY\LOCAL SERVICE

Double-click the SQL Server Analysis Services instanced, and you can “copy” this username into your local buffer so that you can paste it in the next step.

14. Give the service account access to the database. Go back to SQL Server Management Studio. On the left menu, navigate to the Security tab and expand the Logins folder. Look for your OLAP user. (It will have OLAP in the name.)

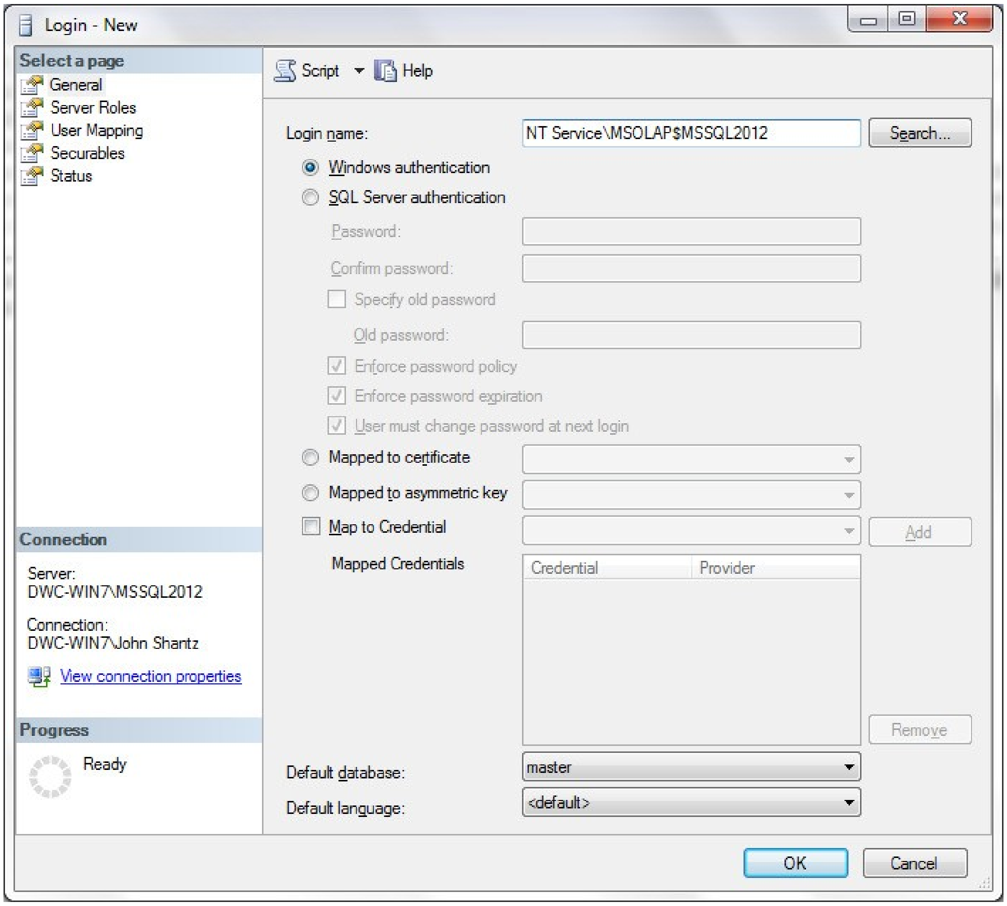
Tip: Use **MSSQL$MSSQL201X** or **DESKTOP-XXXXXXX\SQLEXPRESS** instead.



Security
Logins
##MS_PolicyEventProcessingLogin##
##MS_PolicyTsqlExecutionLogin##
dwc
DWC-WIN7\John Shantz
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
NT SERVICE\MSOLAP$MSSQL2012
NT Service\MSSQL$MSSQL2012
NT SERVICE\SQLAgent$MSSQL2012
NT SERVICE\SQLWriter
NT SERVICE\Winmgmt
sa
www

If you do not have this user, right-click on the “Logins” folder and click “New Login…”

On the next screen under “Login Name” paste the name of the login that you copied in the previous step. Your screen will look like this.



Login name: NT Service\MSOLAP$MSSQL2012

● Windows authentication

○ SQL Server authentication

Password:

Confirm password:

□ Specify old password

Old password:

☑ Enforce password policy

☑ Enforce password expiration

☑ User must change password at next login

○ Mapped to certificate

○ Mapped to asymmetric key

□ Map to Credential

Mapped Credentials

Credential	Provider

Default database: master

Default language: <default>

Connection

Server:
DWC-WIN7\MSSQL2012

Connection:
DWC-WIN7\John Shantz

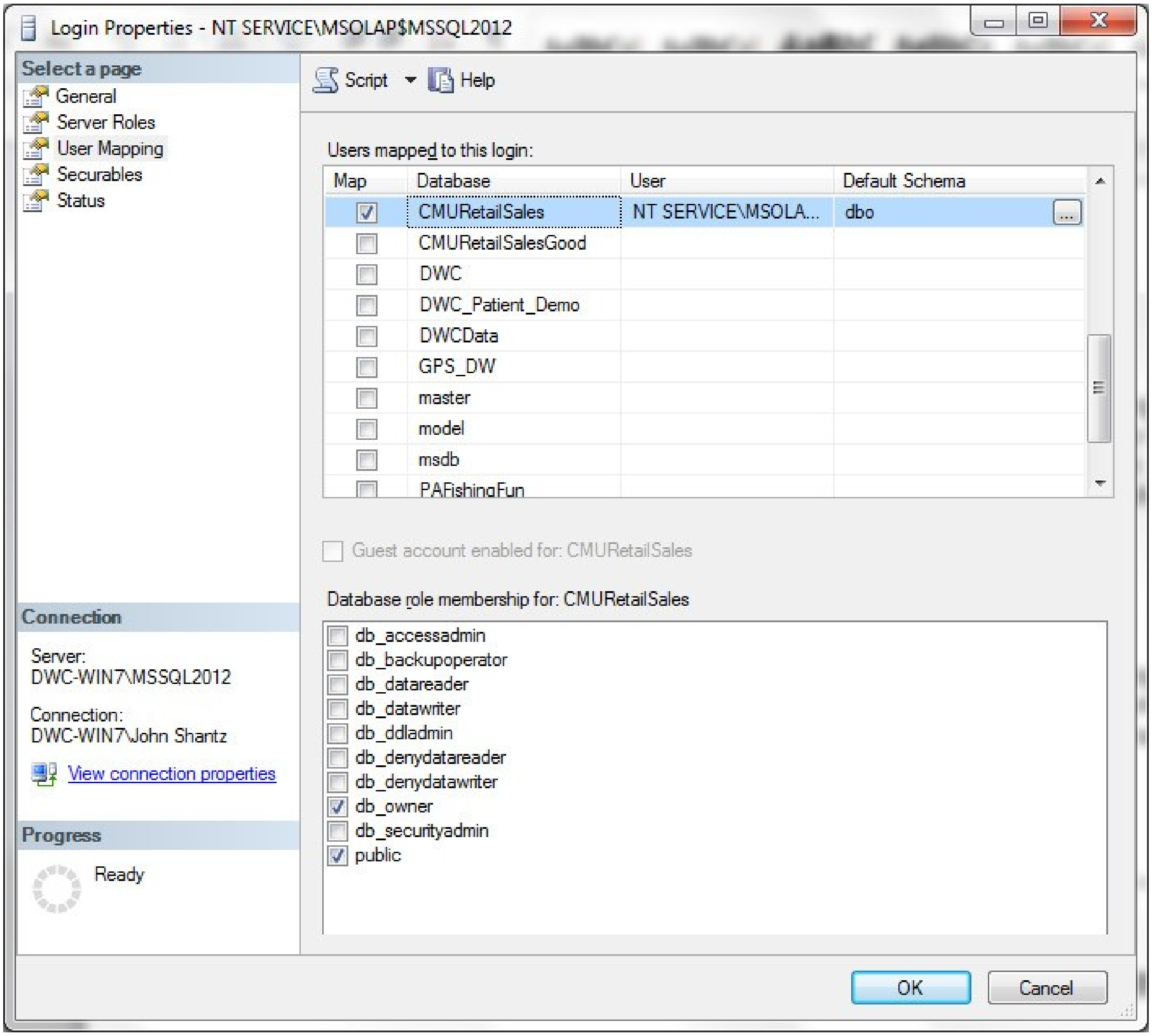
Progress

Ready

Click OK to create a login for the OLAP user.

15. Right-click on the OLAP user and select “Properties”. Go to the User Mapping item on the left menu. This shows the databases this user can access. On the CMU Retail Sales database, make “dbo” the default schema and be sure the “db_owner” role is checked at the bottom of the screen. You window should look something like this: **Screenshot Here**

Tip: Use **MSSQL$MSSQL201X** or **DESKTOP-XXXXXXX\SQLEXPRESS** instead.



Login Properties - NT SERVICE\MSOLAP$MSSQL2012

Users mapped to this login:

Map	Database	User	Default Schema
☑	CMURetailSales	NT SERVICE\MSOLA...	dbo
☐	CMURetailSalesGood		
☐	DWC		
☐	DWC_Patient_Demo		
☐	DWCData		
☐	GPS_DW		
☐	master		
☐	model		
☐	msdb		
☐	PAFishingFun		

☐ Guest account enabled for: CMURetailSales

Database role membership for: CMURetailSales

☐ db_accessadmin
☐ db_backupoperator
☐ db_datareader
☐ db_datawriter
☐ db_ddladmin
☐ db_denydatareader
☐ db_denydatawriter
☑ db_owner
☐ db_securityadmin
☑ public

Server:
DWC-WIN7\MSSQL2012

Connection:
DWC-WIN7\John Shantz

Ready

16. Click OK. Your database setup is complete.

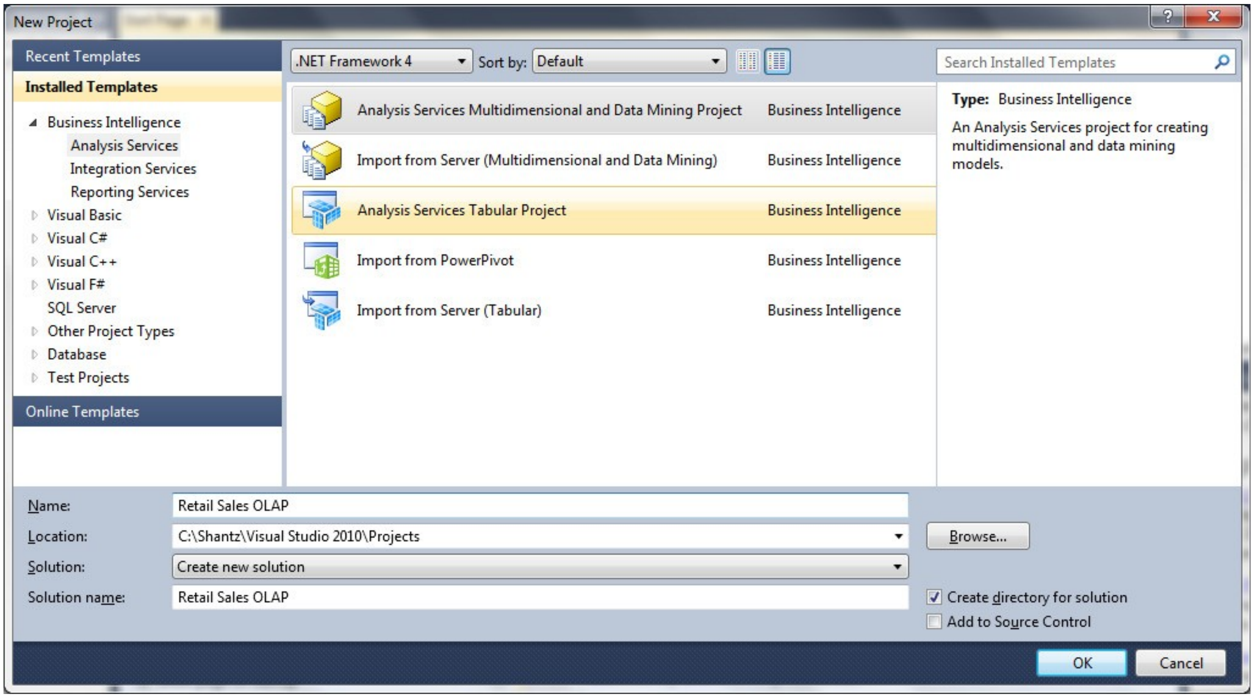
Getting Started With Analysis Services

TIP: To install SSAS, open Visual Studio and navigate to Extensions -> Manage Extensions. Search for Microsoft Analysis Services Projects. Or download it from https://marketplace.visualstudio.com/

1. Start the Business Intelligence Development Studio. Go to Start 7 Programs 7 Microsoft SQL Server 2012 7 SQL Server Data Tools and *run this shortcut as administrator*. If you do not have this option available, then you likely have not installed the correct components of SQL Server.

   Important: Do not just click the above shortcut. You must right-click this value and then left-click on "Run as administrator". This will launch Visual Studio with administrative privileges. Failure to launch the application in this manner will not cause any initial errors, but you will get errors later as you attempt to do other tasks in this lab. This method of accessing the program is required because of new security "features" that were introduced in Microsoft Vista.

2. Create a new project. On the File menu, choose New Project. In the New Project dialog box, choose Business Intelligence 7 Analysis Services on the left menu. Select the option for Analysis Services Multidimensional and Data Mining Project. In the bottom boxes, name your project "Retail Sales OLAP". Click OK when you are done (see image below).



Analysis Services Multidimensional and Data Mining Project	Business Intelligence
Import from Server (Multidimensional and Data Mining)	Business Intelligence
Analysis Services Tabular Project	Business Intelligence
Import from PowerPivot	Business Intelligence
Import from Server (Tabular)	Business Intelligence

Type: Business Intelligence

An Analysis Services project for creating multidimensional and data mining models.

Name: Retail Sales OLAP

Location: C:\Shantz\Visual Studio 2010\Projects

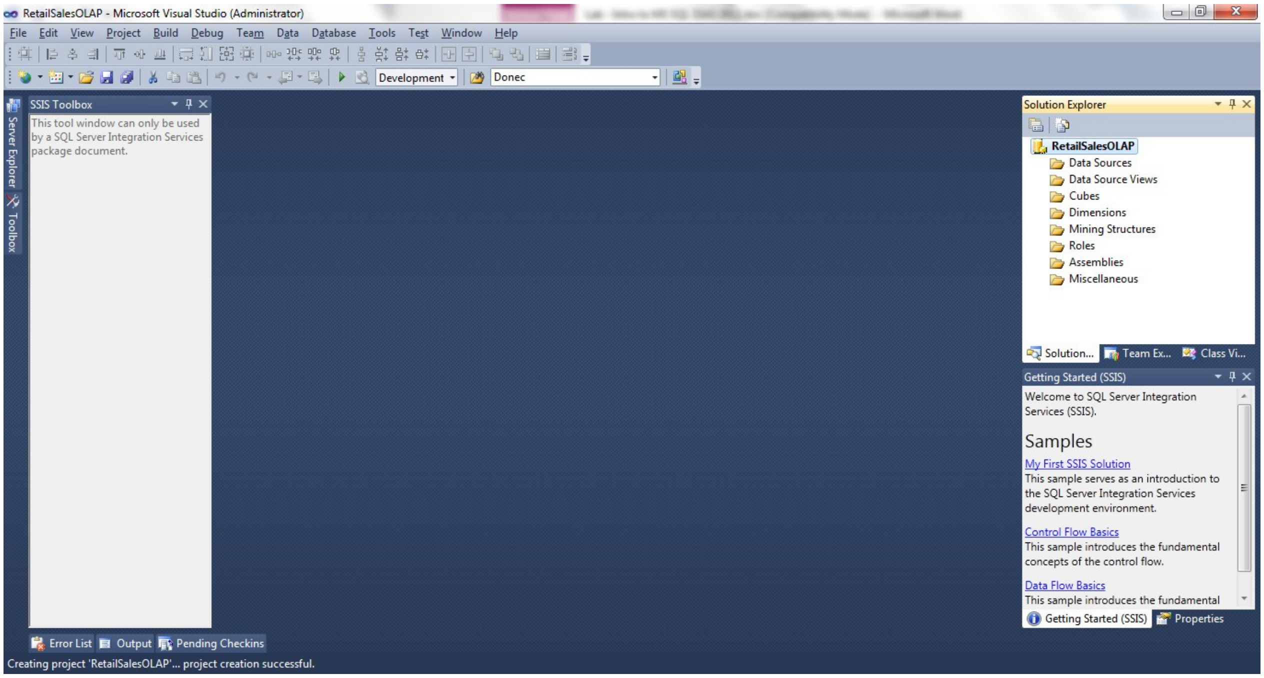
Solution: Create new solution

Solution name: Retail Sales OLAP

☑ Create directory for solution

☐ Add to Source Control

3. After clicking OK, your screen should look like this. Notice the list of folders on the right side in the Solution Explorer.



This tool window can only be used by a SQL Server Integration Services package document.

Welcome to SQL Server Integration Services (SSIS).

This sample serves as an introduction to the SQL Server Integration Services development environment.

This sample introduces the fundamental concepts of the control flow.

This sample introduces the fundamental

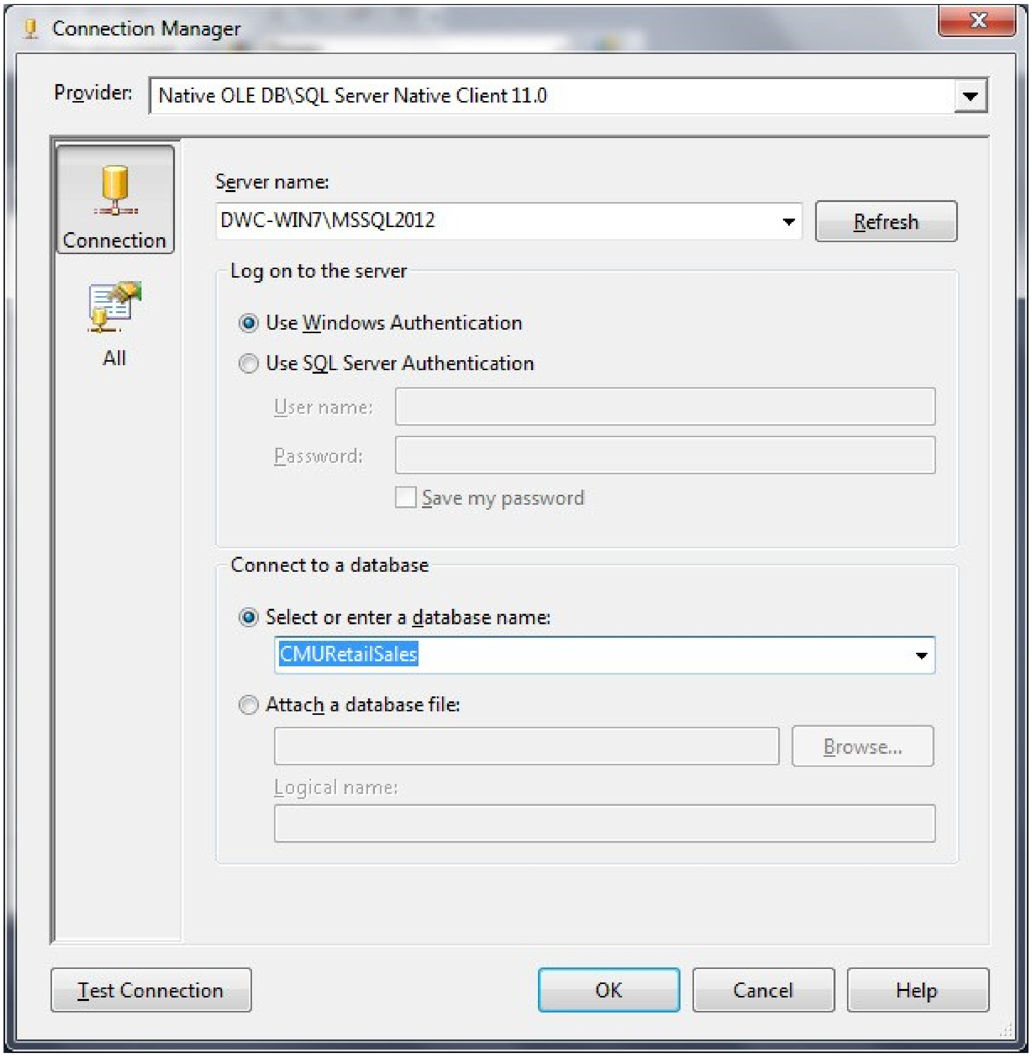
Creating project 'RetailSalesOLAP'... project creation successful.

Create a Data Source

The first step to create your cube is to create an Analysis Services data source. The data source is typically a database or other data source that you will use at the source for your OLAP cube. In our example, the data source is the Retail Sales database we just imported into SQL Server.

1. In the Solution Explorer window on the right, right-click on the Data Sources folder and choose “New Data Source…”
2. In the next dialog box, click “New…” to create a new Data Source.
3. In the Connection Manager dialog box, type localhost into the Server Name box. At the bottom of the window, select the CMU_RETAIL_SSAS database that you just created. Your window will look like the following image. Click OK when you are done.

Tip: Referring to the newly restored database in the earlier setup.



Provider: Native OLE DB\SQL Server Native Client 11.0

Server name: DWC-WIN7\MSSQL2012

Log on to the server

● Use Windows Authentication

○ Use SQL Server Authentication

User name:

Password:

□ Save my password

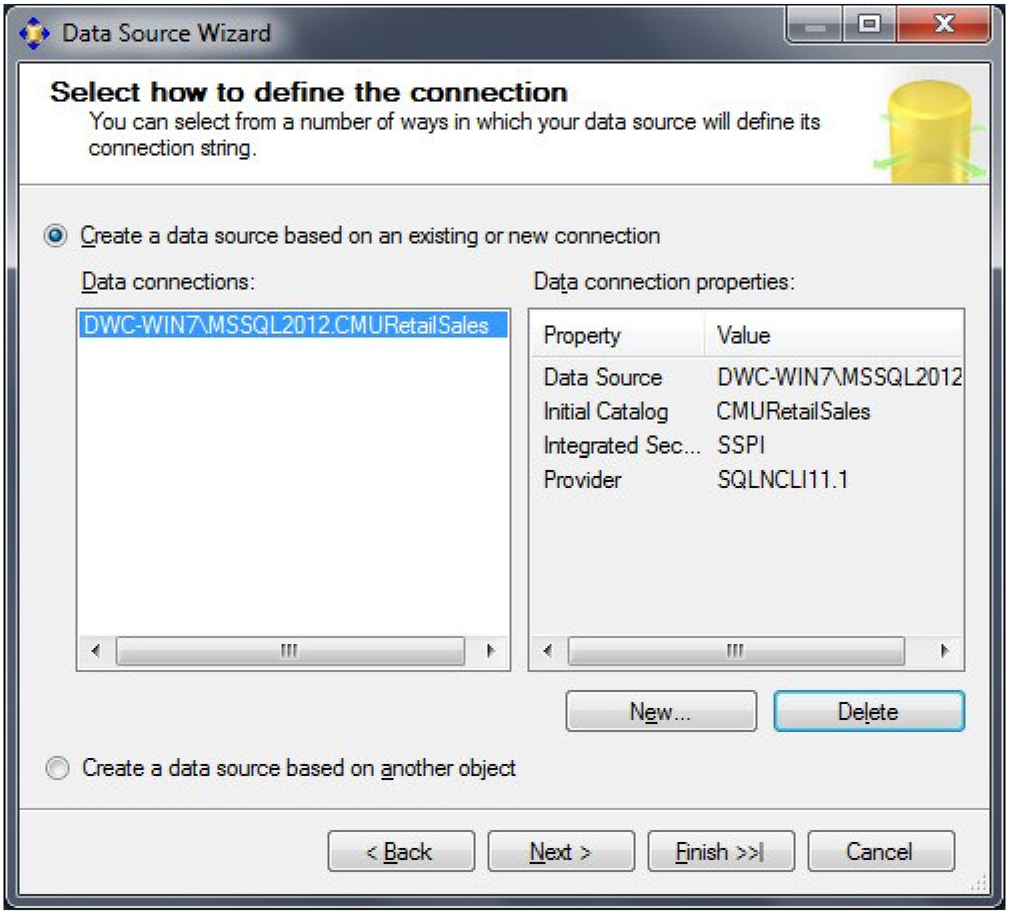
Connect to a database

● Select or enter a database name: CMURetailSales

○ Attach a database file:

Logical name:

4. In the Data Source wizard window that follows, select the Connection you just created and click “Next”.



Data Source Wizard

Select how to define the connection

You can select from a number of ways in which your data source will define its connection string.

● Create a data source based on an existing or new connection

Data connections:

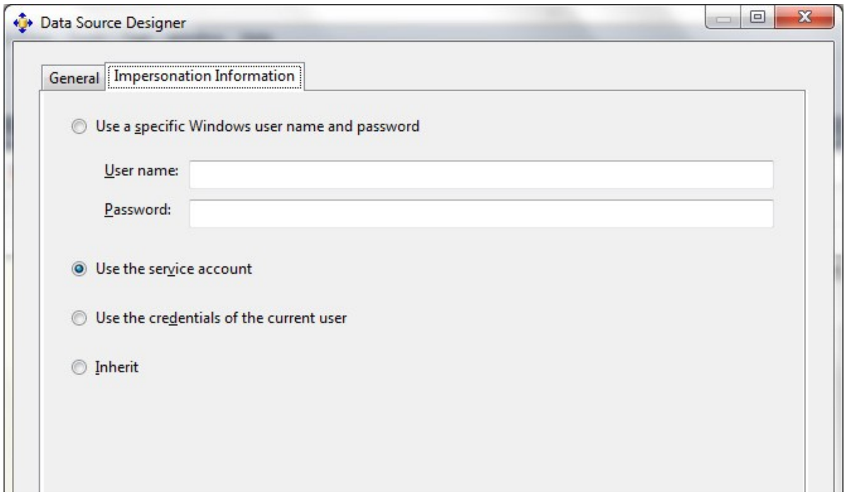
DWC-WIN7\MSSQL2012.CMURetailSales

Data connection properties:

Property	Value
Data Source	DWC-WIN7\MSSQL2012
Initial Catalog	CMURetailSales
Integrated Sec...	SSPI
Provider	SQLNCLI11.1

○ Create a data source based on another object

5. In the next window, choose “Use the service account” and click "Next". Failure to set this item correctly will cause errors later.



Data Source Designer

○ Use a specific Windows user name and password

User name:

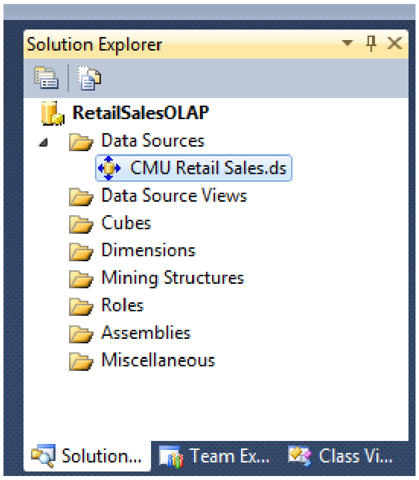
Password:

● Use the service account

○ Use the credentials of the current user

○ Inherit

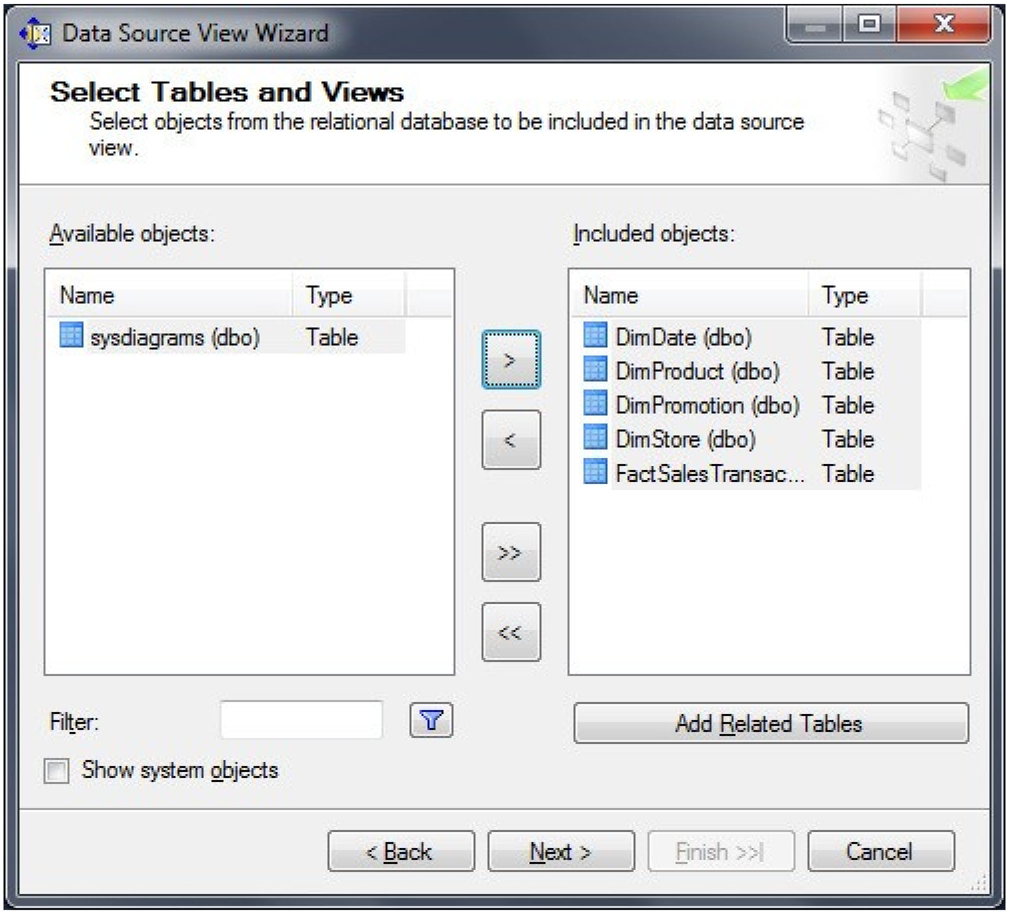
6. On the next screen, leave the default name and click “Finish”. In the right-hand Solution Panel, you should now have a data source listed. It will look like this. **Screenshot Here**



Create a Data Source View

Now that we have created a data source, we must create a data source view. There can be multiple Data Source Views per Data Source. The Data Source view will be equivalent to our star-schema database.

1. In the Solution Explorer window on the right, right-click on the Data Source Views folder and choose "New Data Source View…"
2. In the next dialog box, choose the CMU Retail Sales Data Source you just created and click "Next".
3. In the next window, highlight the one fact table and four dimension tables and click the right arrow in the middle of the box. This will move those tables into the box on the right. After you screen looks like the following one, click "Next".



Data Source View Wizard

Select Tables and Views

Select objects from the relational database to be included in the data source view.

Available objects:

Name	Type
sysdiagrams (dbo)	Table

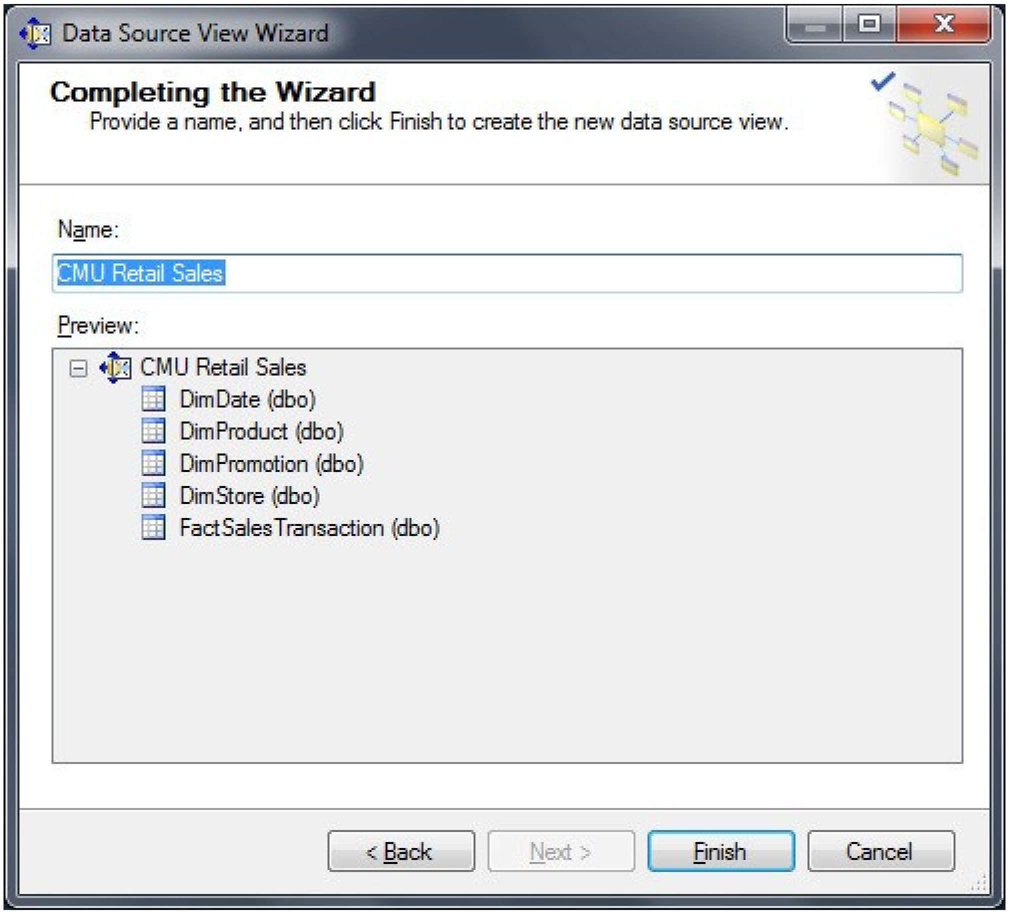
Included objects:

Name	Type
DimDate (dbo)	Table
DimProduct (dbo)	Table
DimPromotion (dbo)	Table
DimStore (dbo)	Table
FactSalesTransac...	Table

Filter:

□ Show system objects

4. On the next screen, leave the default name and click “Finished.”



Data Source View Wizard

Completing the Wizard

Provide a name, and then click Finish to create the new data source view.

Name:

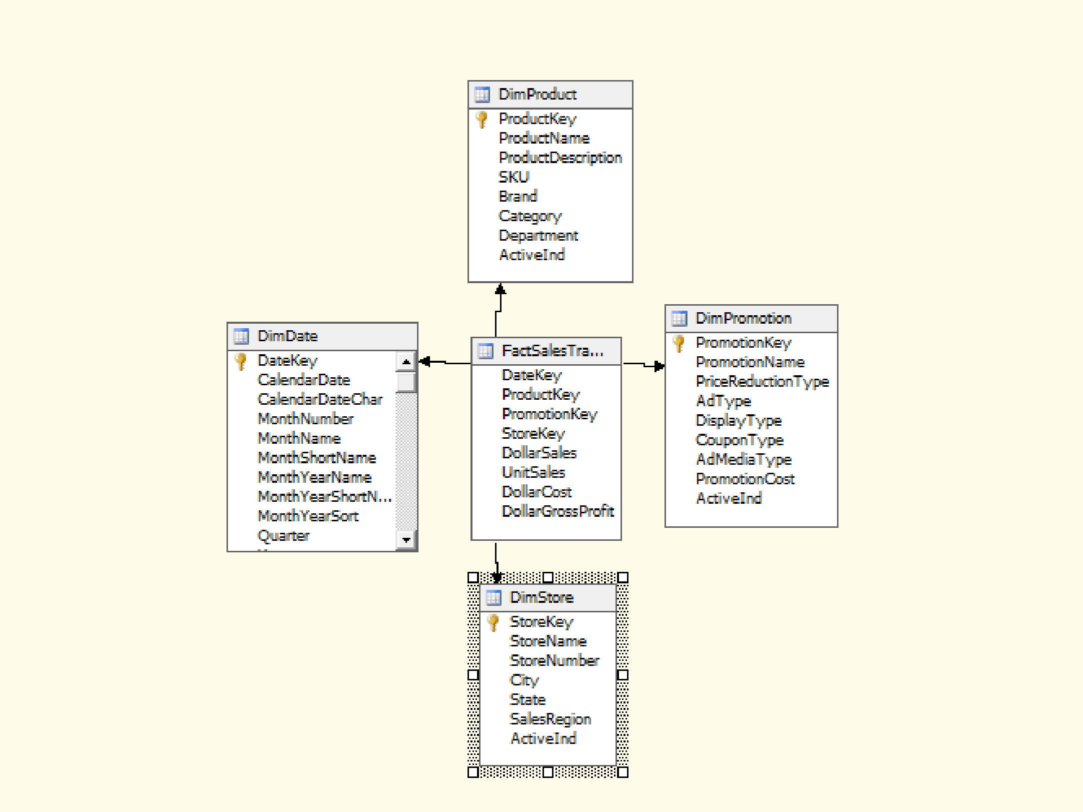
CMU Retail Sales

Preview:

- CMU Retail Sales
  - DimDate (dbo)
  - DimProduct (dbo)
  - DimPromotion (dbo)
  - DimStore (dbo)
  - FactSalesTransaction (dbo)

7. In the right-hand Solution Panel, you should now have a Data Source View listed. The main panel will now resemble our star schema model. Your window should look like this. You can see that Analysis Services read the foreign keys that were defined in the database between the fact tables and dimension tables and it therefore understands the relationships between the tables. **Screenshot Here**

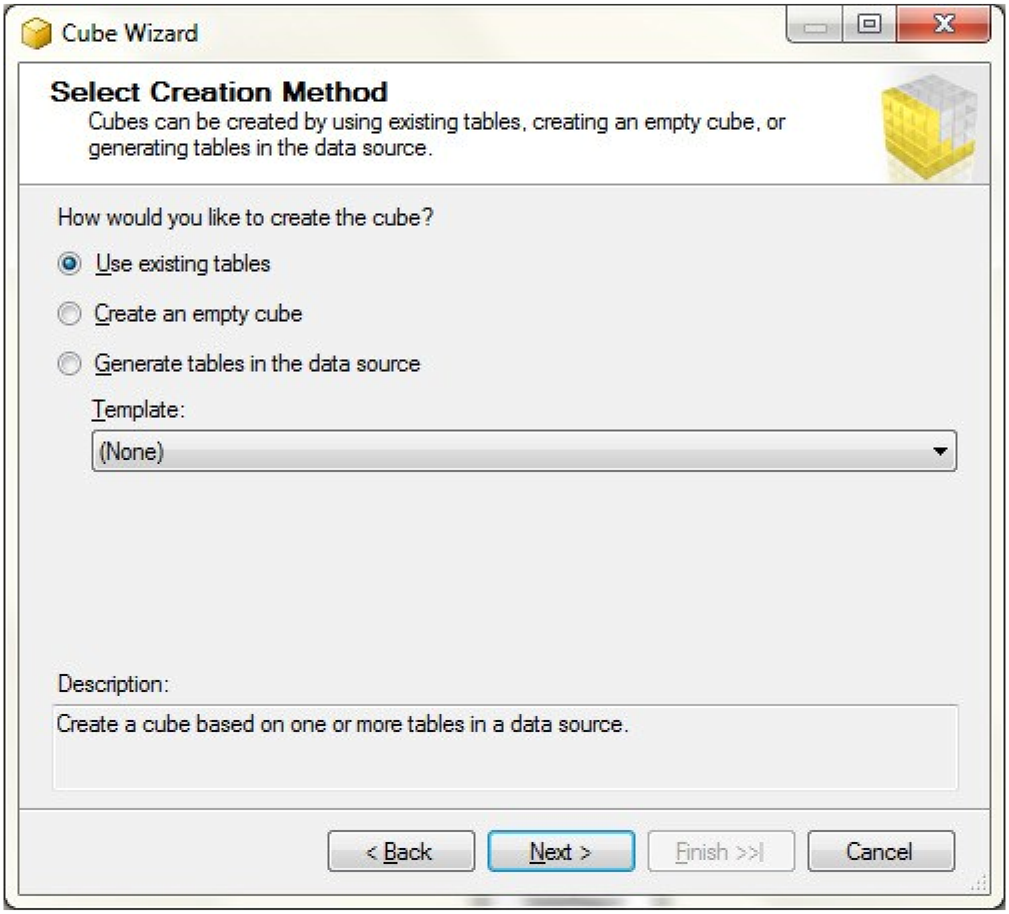
Tip: Need to double click the Data Source Views > CMU Retail Sales.dsv



Create a Cube

Once you have created a Data Source View, you are ready to create a Cube. A Cube is the multidimensional structure that will contain our summarized OLAP data.

1. In the Solution Explorer window on the right, right-click on the Cubes folder and choose “New Cube…”
2. The Cube Wizard will be launched. In the next dialog box, keep the default “Use Existing Tables” and click “Next”. This tells the wizard our star-schema database already exists.



Cube Wizard

Select Creation Method

Cubes can be created by using existing tables, creating an empty cube, or generating tables in the data source.

How would you like to create the cube?

● Use existing tables

○ Create an empty cube

○ Generate tables in the data source

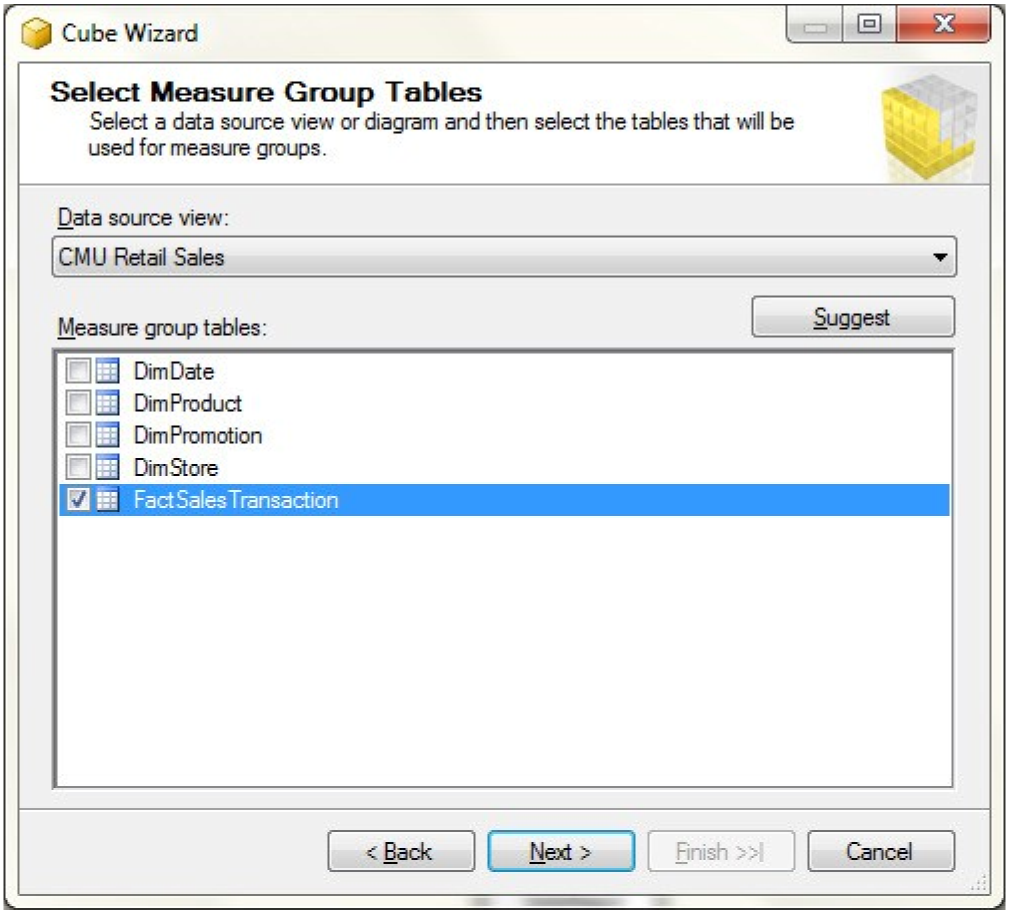
Template:

(None)

Description:

Create a cube based on one or more tables in a data source.

3. In the next window, choose the Data Source View we just created. Check the box next to the *FactSalesTransaction* table and click “Next”. This tells the application that the Fact Sales Transaction table contains our measures.



Cube Wizard

Select Measure Group Tables

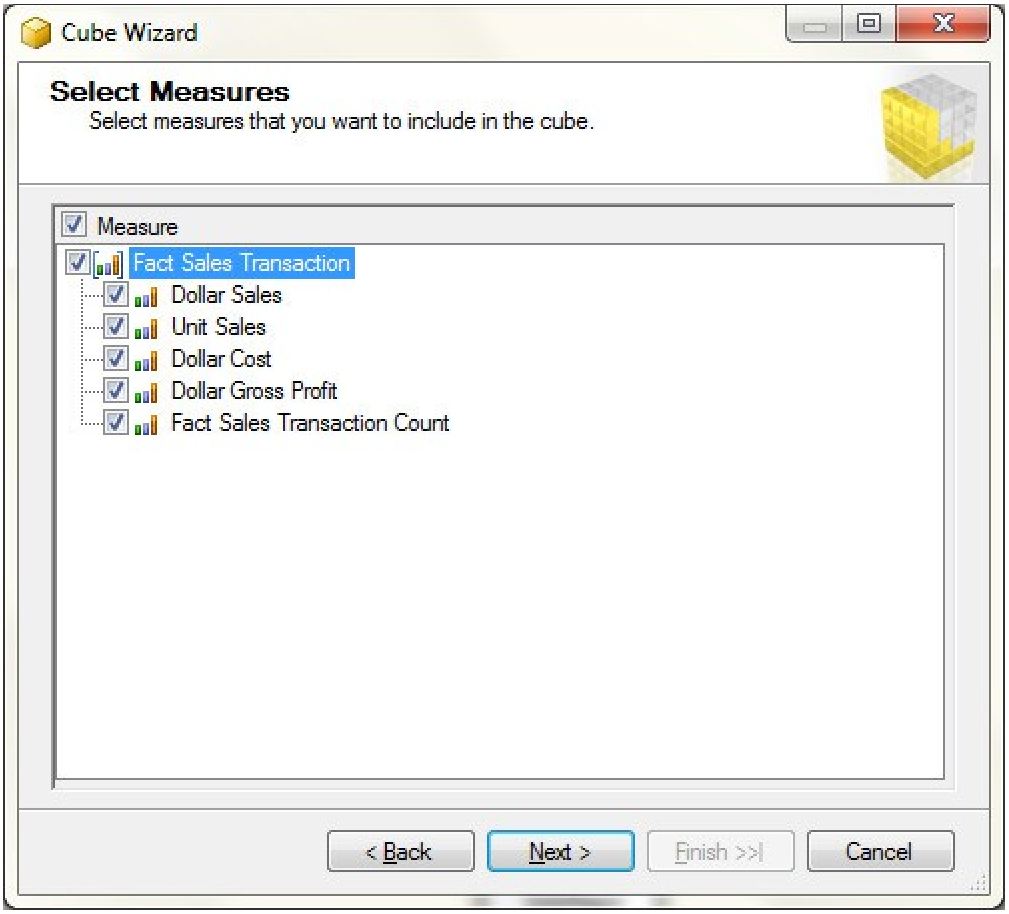
Select a data source view or diagram and then select the tables that will be used for measure groups.

Data source view: CMU Retail Sales

Measure group tables:

- □ DimDate
- □ DimProduct
- □ DimPromotion
- □ DimStore
- ☑ FactSalesTransaction

4. On the next screen, you will see all the fields that are measures in the Fact Sales table. Keep all the fields selected and click “Next”.



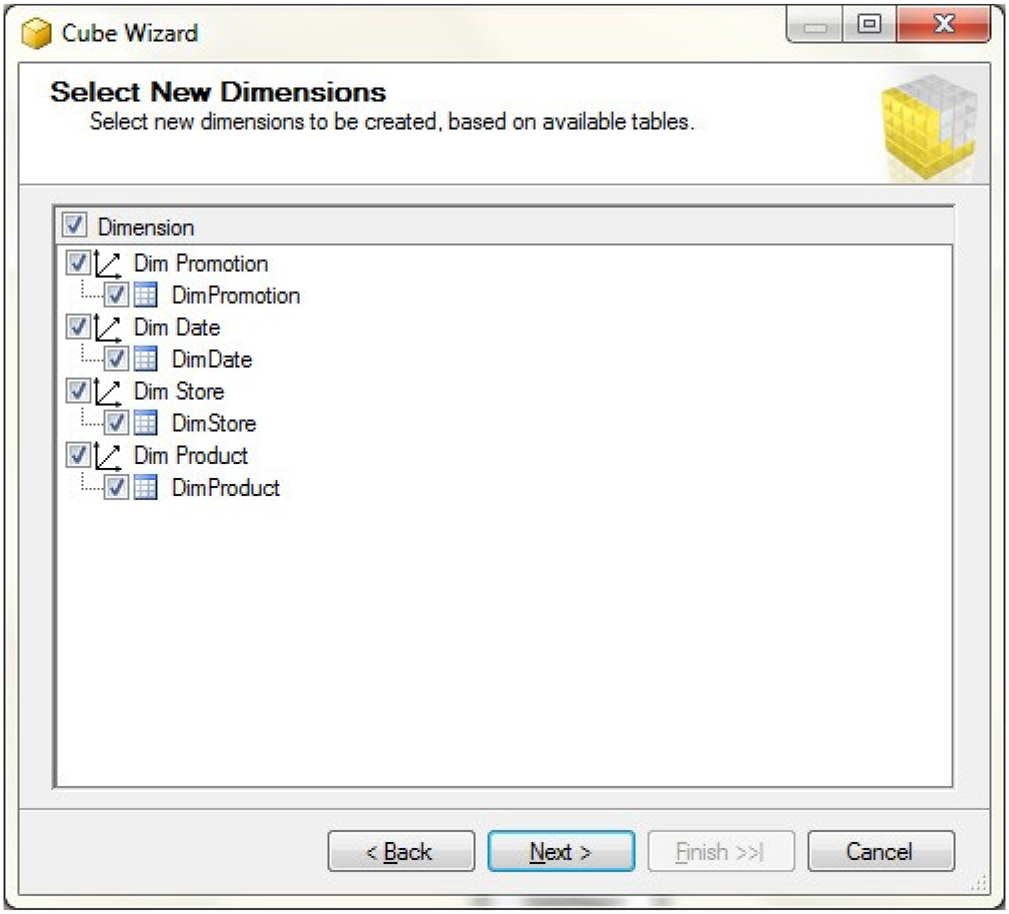
Select Measures

Select measures that you want to include in the cube.

☑ Measure

- ☑ Fact Sales Transaction
  - ☑ Dollar Sales
  - ☑ Unit Sales
  - ☑ Dollar Cost
  - ☑ Dollar Gross Profit
  - ☑ Fact Sales Transaction Count

5. On the next screen, you will select the dimensions that are created. Keep all the dimension boxes checked and click “Next”.



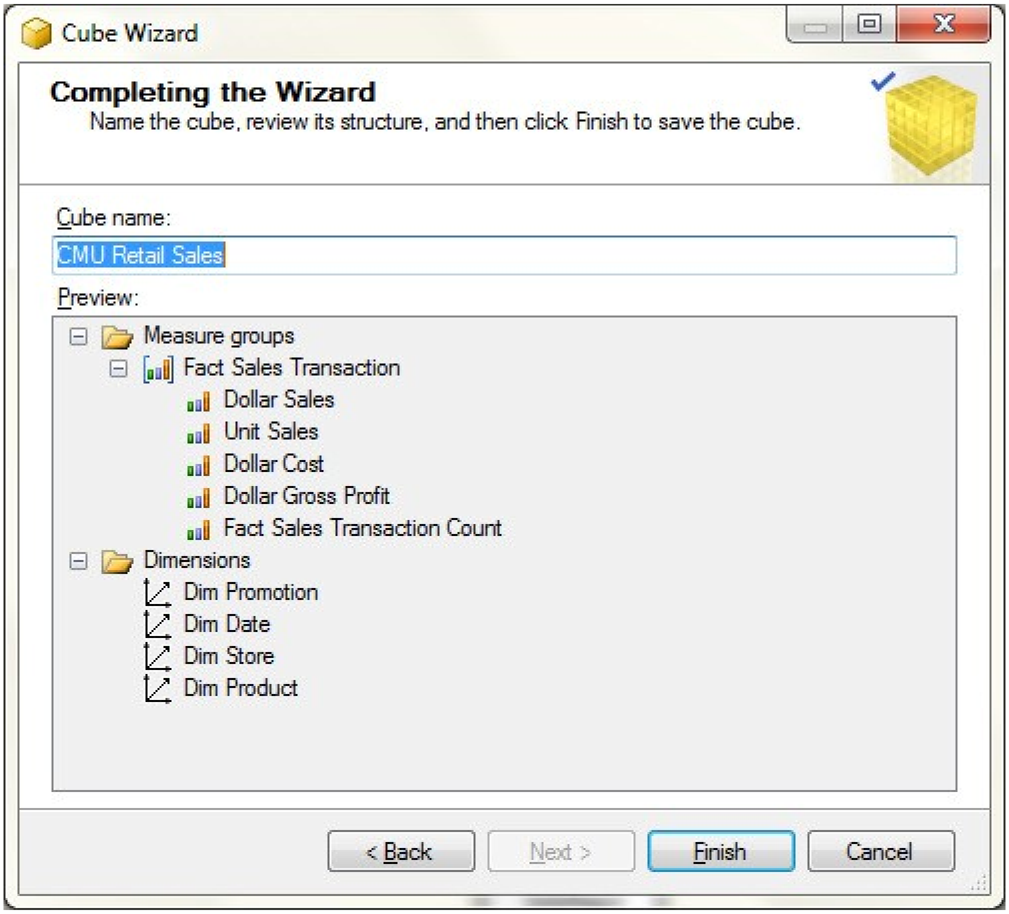
Select New Dimensions

Select new dimensions to be created, based on available tables.

☑ Dimension

- ☑ Dim Promotion
  - ☑ DimPromotion
- ☑ Dim Date
  - ☑ DimDate
- ☑ Dim Store
  - ☑ DimStore
- ☑ Dim Product
  - ☑ DimProduct

6. The next screen is the final screen of the wizard. Select the name you want for the cube and click "Finish".



Completing the Wizard

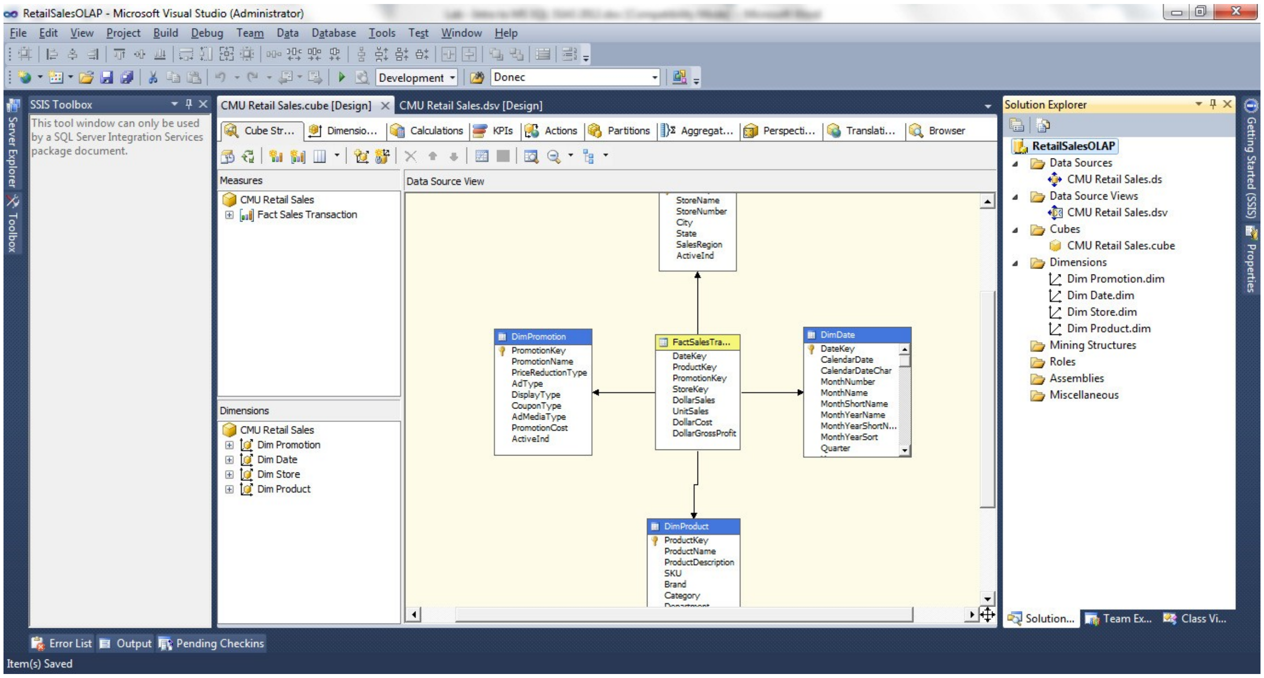
Name the cube, review its structure, and then click Finish to save the cube.

Cube name: CMU Retail Sales

Preview:

- Measure groups
  - Fact Sales Transaction
    - Dollar Sales
    - Unit Sales
    - Dollar Cost
    - Dollar Gross Profit
    - Fact Sales Transaction Count
- Dimensions
  - Dim Promotion
  - Dim Date
  - Dim Store
  - Dim Product

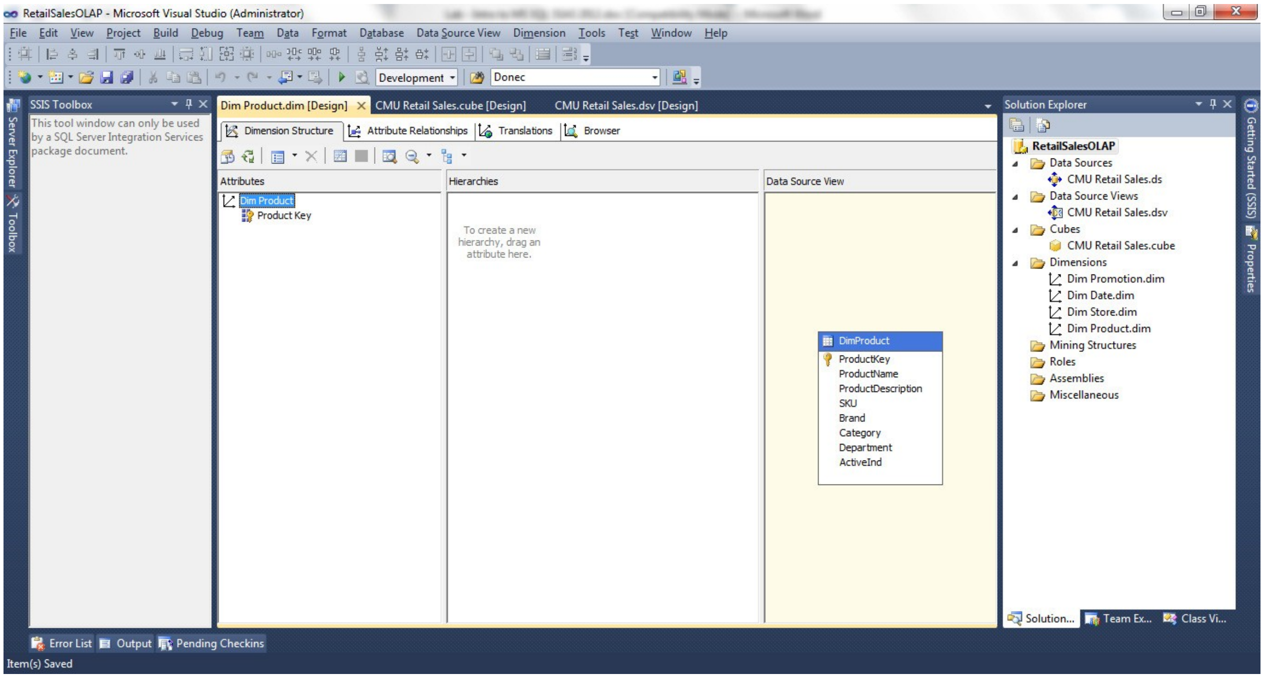
7. In the right-hand Solution Panel, you should now have a Cube and Dimensions listed. The main panel will now resemble our star schema model and there will be a series of tabs at the top of the main canvas. Your window should look like this. **Screenshot Here**



Modify the Basic Product Dimension

Now that you have created a cube, you must create a basic dimensional structure for your cube users to browse. This part of the lab will help you design a very basic cube structure.

1. In the bottom left corner of your screen, expand the Dim Product dimension. Click on the Edit Dim Product link and you will be taken to the Product Dimension edit screen. It should look like this.



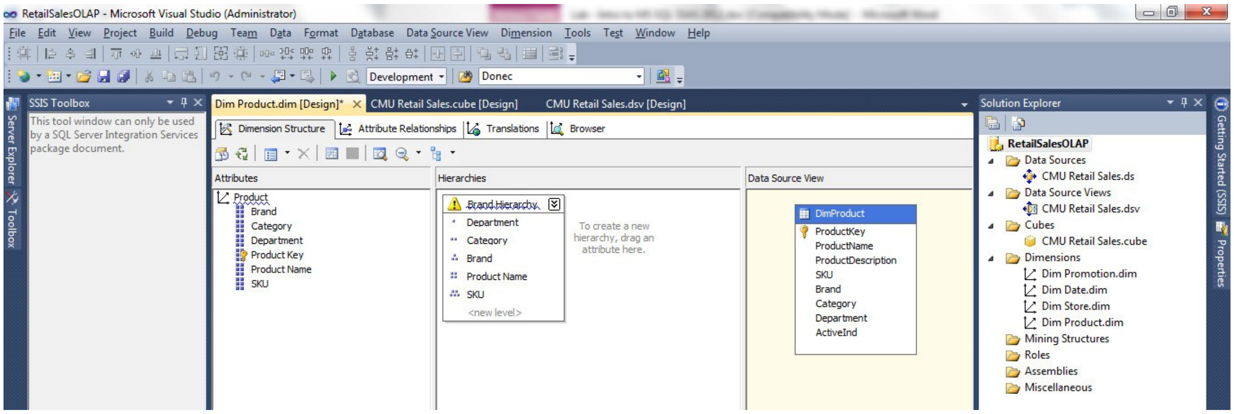
This tool window can only be used by a SQL Server Integration Services package document.

To create a new hierarchy, drag an attribute here.

Item(s) Saved

2. We will now complete a basic configuration of the Product dimension. Complete the following tasks.
    a. On the left side in the Attribute screen, rename the "Dim Product" dimension to be the "Product" dimension.
    b. Using the columns listed in the Data Source View screen, drag and drop the Product Name, SKU, Brand, Category and Department into the Product Dimension. This makes those columns available in the OLAP cube.
    c. Using the columns that are now at the left create a hierarchy named Brand Hierarchy that contains the SKU, Brand, Category and Department. This creates a hierarchy that will be usable by the OLAP cube users to drill down to a lower level of detail. Note that the highest level of the hierarchy is on the top (Department) and it goes down from there. Tip: Include product name

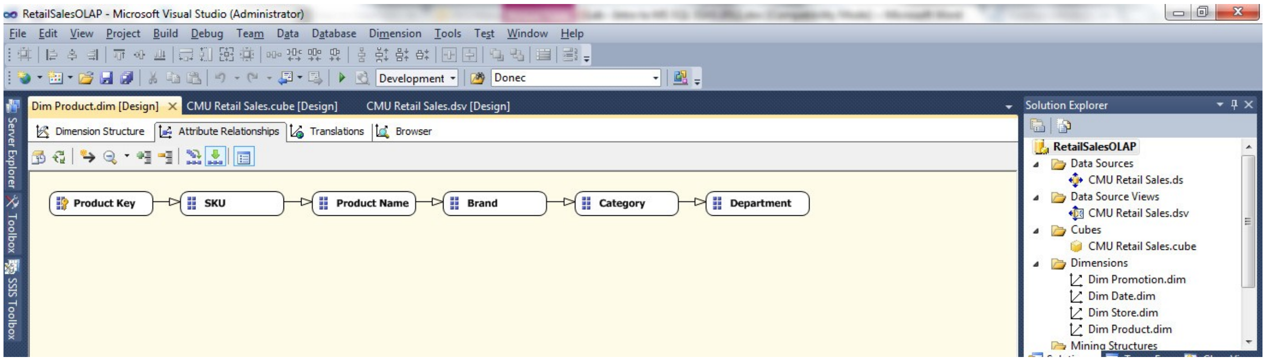
Your screen should now resemble the picture below. **Screenshot Here**



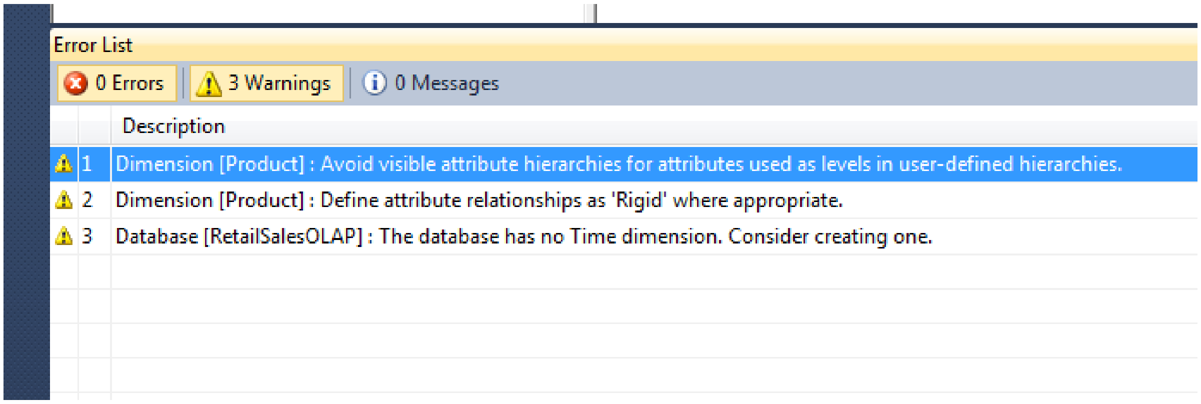
This tool window can only be used by a SQL Server Integration Services package document.

To create a new hierarchy, drag an attribute here.

3. At the top of the current window, click on the Attribute Relationships window. Starting with the Product Key field, create relationships for each member of our hierarchy. This tells the OLAP server that these attributes are related, which will increase query performance later. **Screenshot Here**



4. Under the Build menu at the top of the screen, choose “Build Retail Sales OLAP”. This will test our current project for errors. In the Error List tab at the bottom of the screen you will see that we now have some Warnings that may need to be addressed. **Screenshot Here**



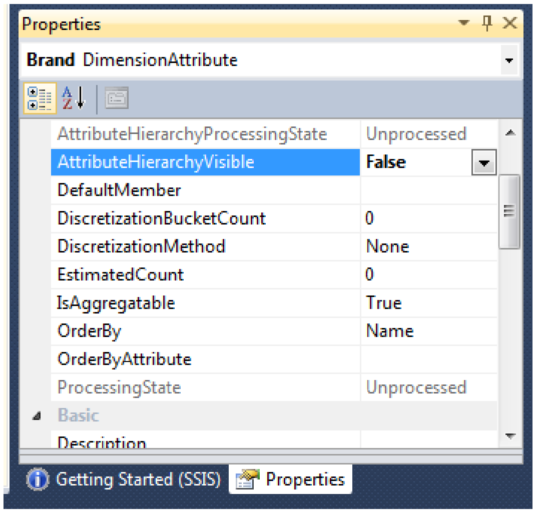
Error List

0 Errors | 3 Warnings | 0 Messages

	Description
1	Dimension [Product] : Avoid visible attribute hierarchies for attributes used as levels in user-defined hierarchies.
2	Dimension [Product] : Define attribute relationships as 'Rigid' where appropriate.
3	Database [RetailSalesOLAP] : The database has no Time dimension. Consider creating one.

5. When working in Visual Studio, it is recommended that you do not ignore errors or warnings until you understand what they mean and what the effect of ignoring them is. At this point I will correct the first two warnings above as follows:

   a. The first warning tells us that we are displaying duplicate hierarchies for some of our Product attributes. To remove this warning, click on each attribute that is visible in the Product dimension on the Dimension Structure tab. In the Properties box for each attribute, set the AttributeHierarchyVisible property to FALSE.

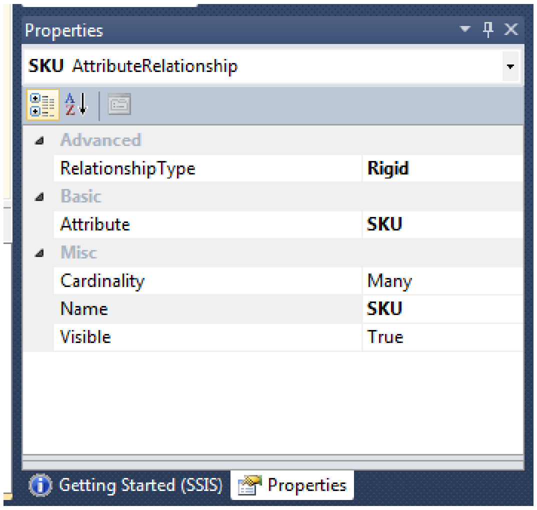


Properties

Brand DimensionAttribute

AttributeHierarchyProcessingState	Unprocessed
AttributeHierarchyVisible	**False**
DefaultMember	
DiscretizationBucketCount	0
DiscretizationMethod	None
EstimatedCount	0
IsAggregatable	True
OrderBy	Name
OrderByAttribute	
ProcessingState	Unprocessed
Basic	
Description	

b. The second warning tells us we have not defined our attribute relationships as strictly as we could have. To remove this warning, go to the Attribute Relationship tab. Right-click on each arrow and go to Relationship Type and set it to “Rigid”.

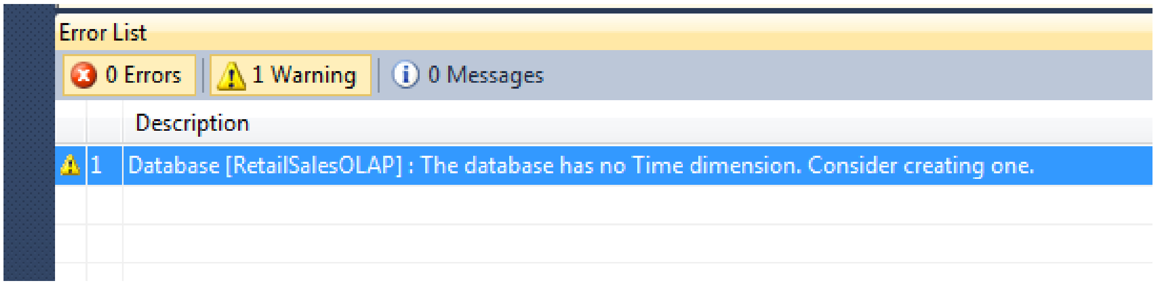


Properties

SKU AttributeRelationship

Advanced	
RelationshipType	**Rigid**
Basic	
Attribute	**SKU**
Misc	
Cardinality	Many
Name	**SKU**
Visible	True

At this point we only have a warning related to the Time dimension. We will correct that in the next section. **Screenshot Here**



Error List

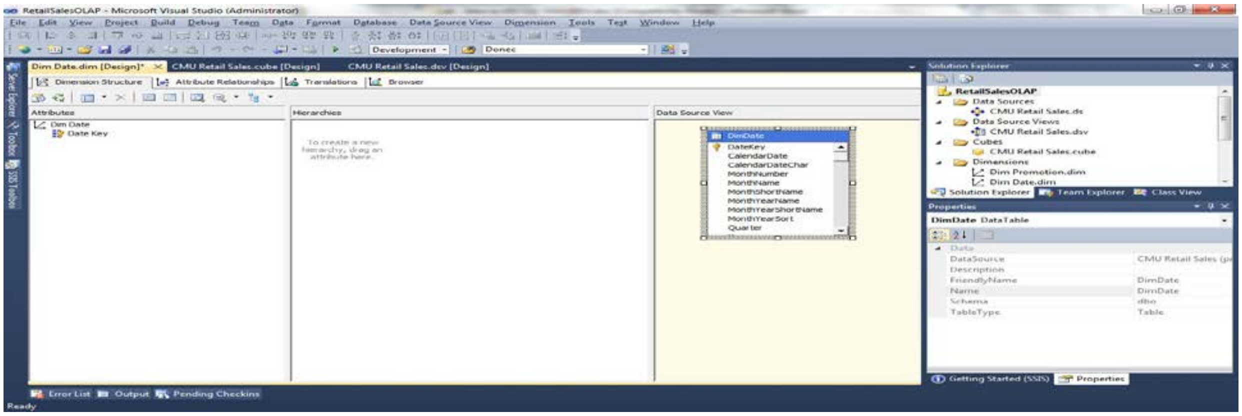
0 Errors | 1 Warning | 0 Messages

		Description
⚠	1	Database [RetailSalesOLAP] : The database has no Time dimension. Consider creating one.

Modify the Basic Time Dimension
Our basic cube construction will continue by modifying the basic properties of our Time Dimension.

1. Go to the cube tab at the top of the page and select the Cube Structure sub-tab.

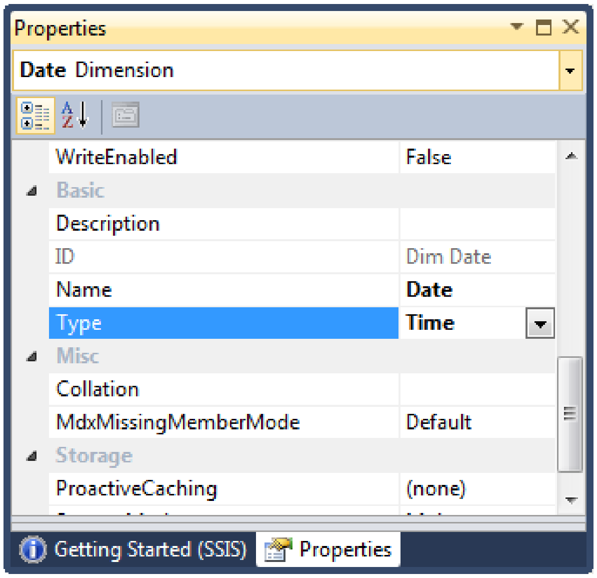
2. In the bottom left corner of your screen, expand the Dim Date dimension. Click on the Edit Dim Date link and you will be taken to the Date Dimension edit screen. It should look like this.



DimDate DataTable

Data	
DataSource	CMU Retail Sales (p
Description	
FriendlyName	DimDate
Name	DimDate
Schema	dbo
TableType	Table

3. We will now complete a basic configuration of the Date dimension. Complete the following tasks.
   a. On the left side in the Attribute screen, rename the "Dim Date" dimension to be the "Date" dimension.
   b. In the Properties dialog box for the Date Dimension, set the Type property to be "Time". This tells the OLAP server that this is our Time dimension

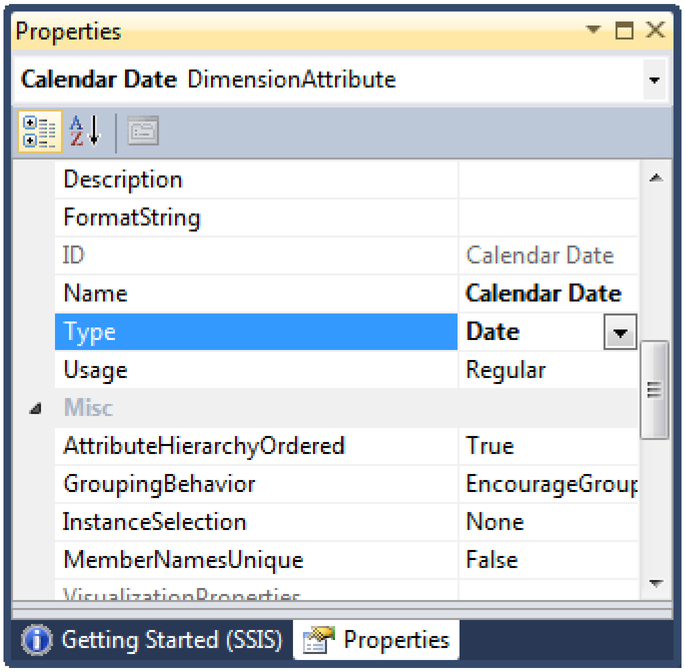


Date Dimension

WriteEnabled	False
Basic	
Description	
ID	Dim Date
Name	Date
Type	Time
Misc	
Collation	
MdxMissingMemberMode	Default
Storage	
ProactiveCaching	(none)

c. Using the columns listed in the Data Source View screen, drag and drop all available columns in the dimension table into the Time Dimension. This makes those columns available in the OLAP cube.

d. In the Properties dialog box for the Calendar Date attribute, set the Type property to be “Date”. This tells the OLAP server that this is our Date attribute.

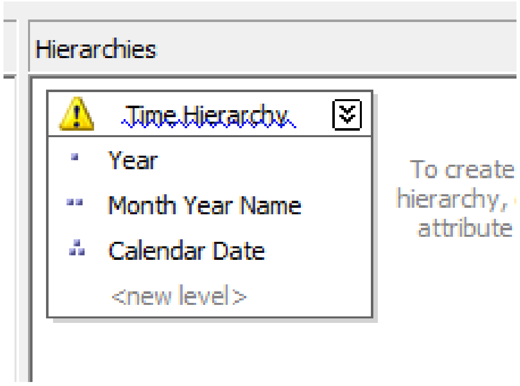


Properties

Calendar Date DimensionAttribute

Description	
FormatString	
ID	Calendar Date
Name	**Calendar Date**
Type	**Date**
Usage	Regular
Misc	
AttributeHierarchyOrdered	True
GroupingBehavior	EncourageGroup
InstanceSelection	None
MemberNamesUnique	False
VisualizationProperties	

e. Using the columns that are now at the left create a hierarchy named Time Hierarchy that contains the Year, Month Year Name and Calendar Date. This creates a hierarchy that will be usable by the OLAP cube users to drill down to a lower level of detail.



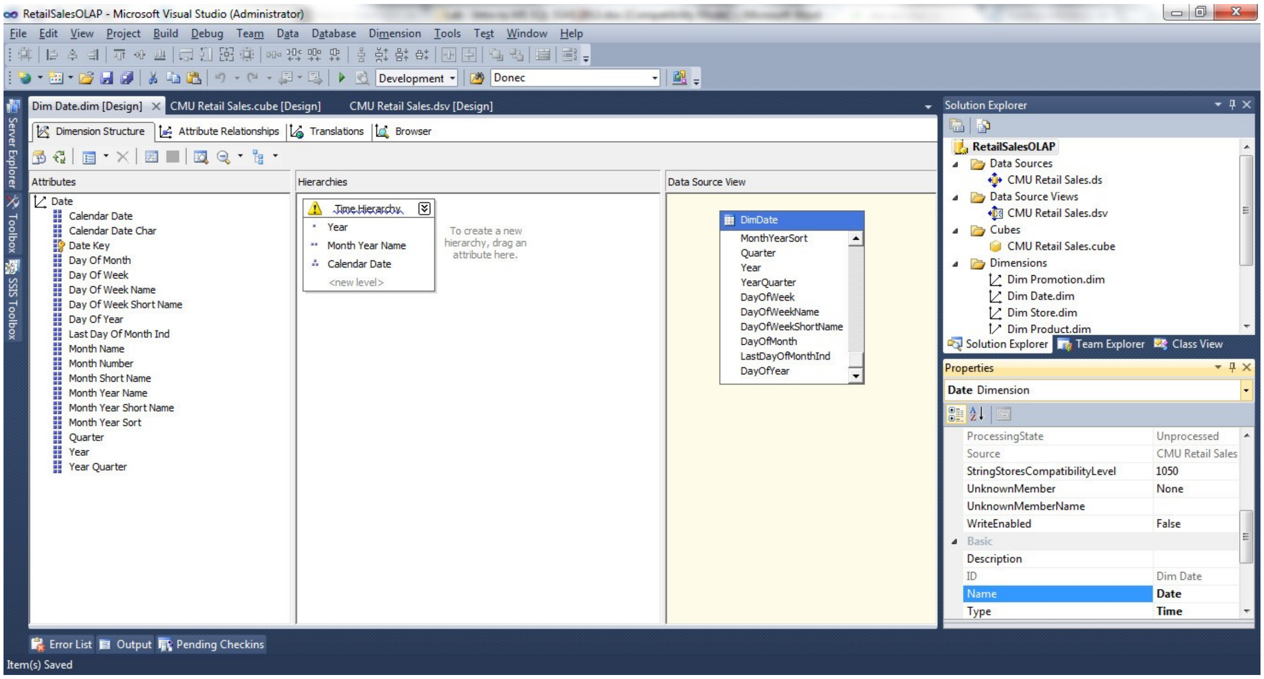
Hierarchies

Time Hierarchy

- Year
- Month Year Name
- Calendar Date
- <new level>

f. To avoid the Attribute Hierarchy warning we received on the Product dimension, click on each attribute that is visible in the hierarchy you just created and set the AttributeHierarchyVisible property to FALSE in the Properties dialog box. **Screenshot Here**

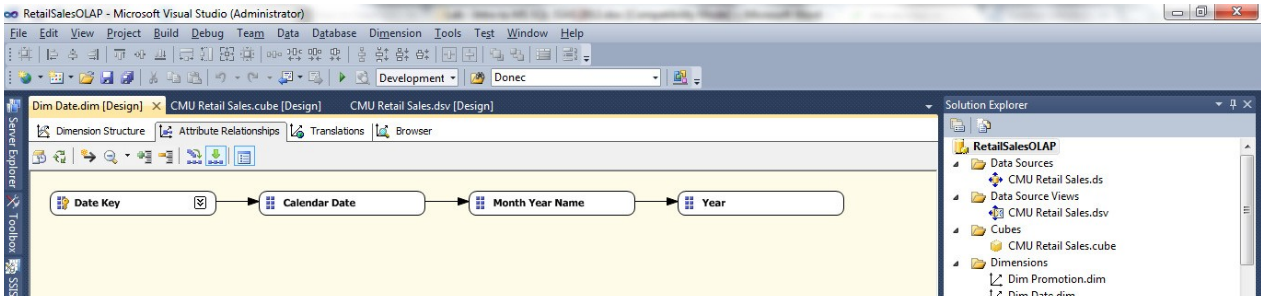
Your screen should now resemble the picture below.



Date Dimension

ProcessingState	Unprocessed
Source	CMU Retail Sales
StringStoresCompatibilityLevel	1050
UnknownMember	None
UnknownMemberName	
WriteEnabled	False
Basic	
Description	
ID	Dim Date
Name	Date
Type	Time

4. At the top of the current window, click on the Attribute Relationships window. Set up your attribute relationships so that they look like the picture below. Be sure to designate the relationships again as “Rigid”. **Screenshot Here**

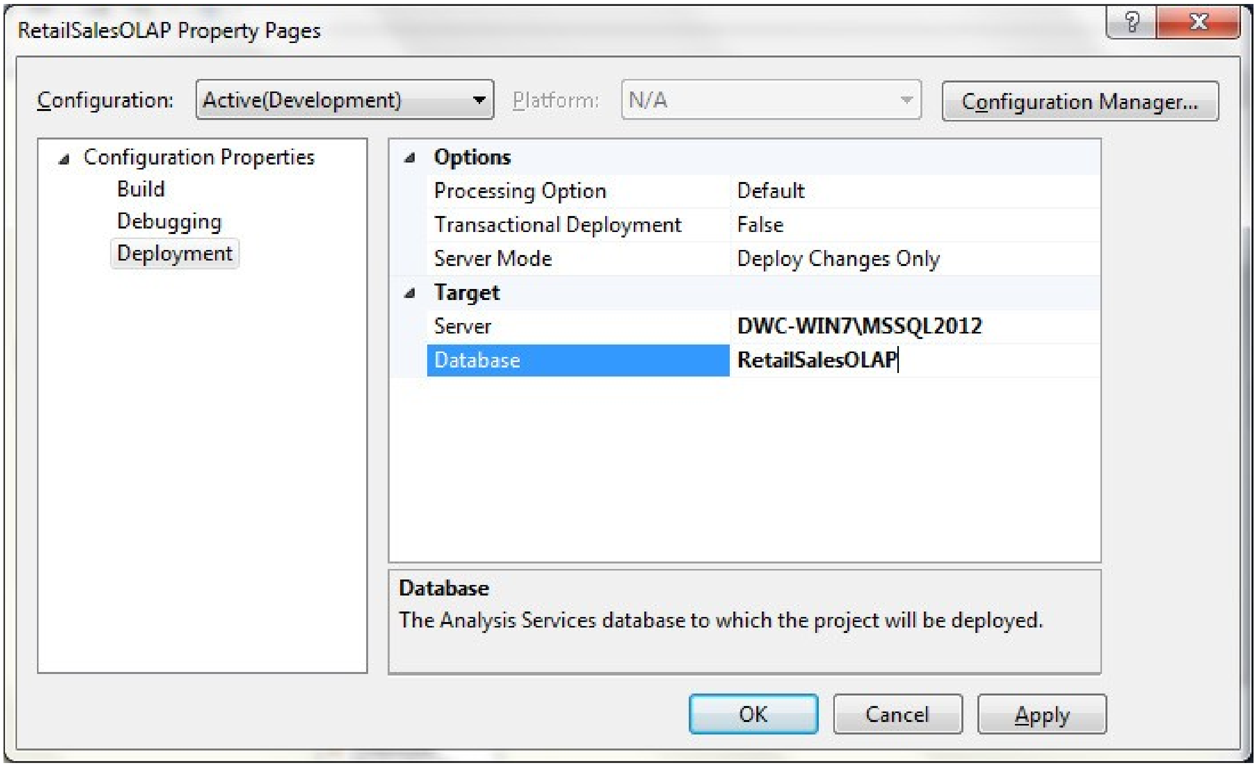


5. Rebuild your project again. Now you should have no warnings or errors. **Screenshot Here**

Deploy the Cube
Now that you have completed the design of a basic cube, you must actually deploy it to a SQL Server database before it can be viewed. When you deploy the cube, it will run the necessary queries against that data source and store the results in the OLAP database.

1. Go to the cube tab at the top of the page and select the Cube Structure tab.

2. First we must tell the cube the server it should be deployed to. In the Solution Explorer at the top right of your screen, right-click the Retail Sales OLAP project and select Properties. On the left of the resulting screen, select Deployment. Change the Server property to point to the SQL Server database name for your computer. This will generally be the same name you have been using. After your window looks like the one below click "OK".



Options
Processing Option: Default
Transactional Deployment: False
Server Mode: Deploy Changes Only
Target
Server: DWC-WIN7\MSSQL2012
Database: RetailSalesOLAP

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

3. At the top of your cube window, click the green right-arrow that looks like a Play button. This will build and deploy your cube. You will see the cube building in the bottom right window.
TIP: If an error occurs (project could not be deployed to <SQL server name> server because of the following connectivity problems: A connection cannot be made to redirector. Ensure that 'SQL Browser' service is running), go to **SQL Server Configuration Manager** and enable **SQL Server Browser**.

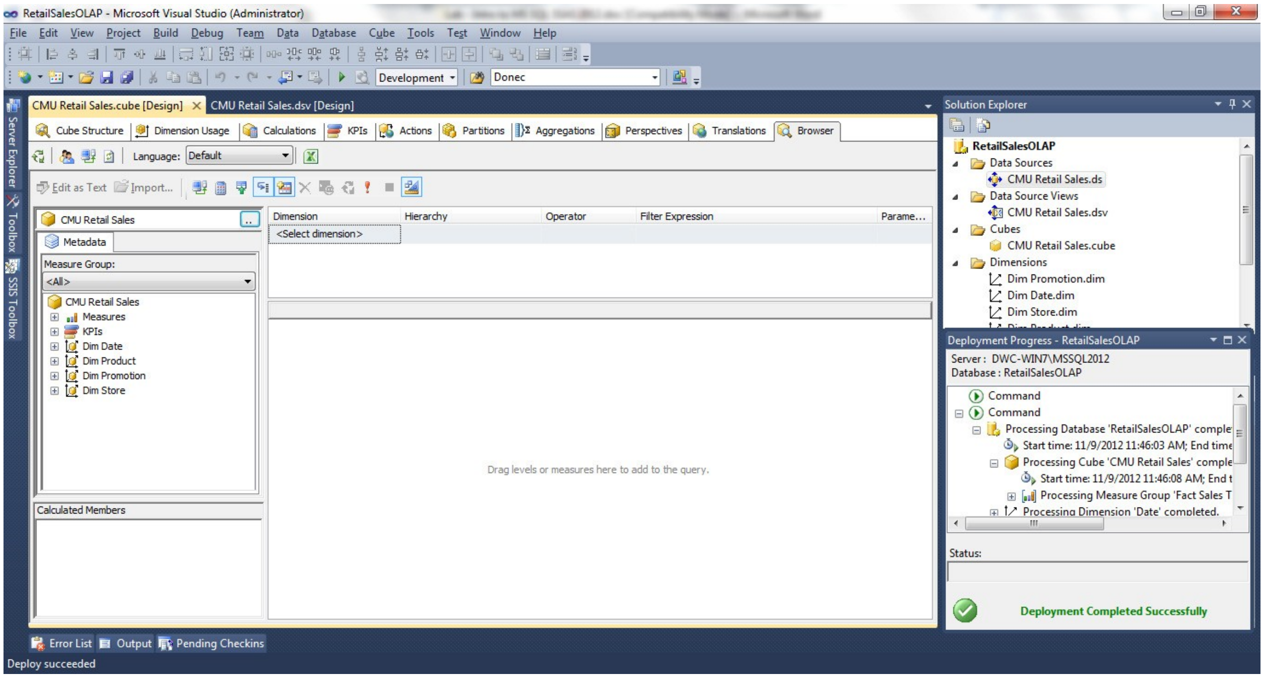
Source:
https://docs.microsoft.com/en-us/dynamics-nav/how-to--start-sql-browser-service#:~:text=To%20start%20SQL%20Server%20Browser%20Service&text=Right%2Dclick%20SQL%20Server%20Browser,again%2C%20and%20then%20choose%20Start.

4. It is very typical to get security or permissions errors at this point. One common error says something like this:

   Login failed for user 'NT SERVICE\MSOLAP$MSSQL2012'.; 28000; Cannot open database "CMURetailSales" requested by the login. The login failed.

   This occurs because the Service Account user that we set up the OLAP engine to use does not have access to the database. If this happens, go back to the step where we set up the database and be sure the OLAP user has access.

5. If the build is successful, your screen will look like this. **Screenshot Here**



Deployment Progress - RetailSalesOLAP

Server : DWC-WIN7\MSSQL2012
Database : RetailSalesOLAP

```
Command
Command
  Processing Database 'RetailSalesOLAP' comple
    Start time: 11/9/2012 11:46:03 AM; End time
    Processing Cube 'CMU Retail Sales' comple
      Start time: 11/9/2012 11:46:08 AM; End t
      Processing Measure Group 'Fact Sales T
    Processing Dimension 'Date' completed.
```

Status:

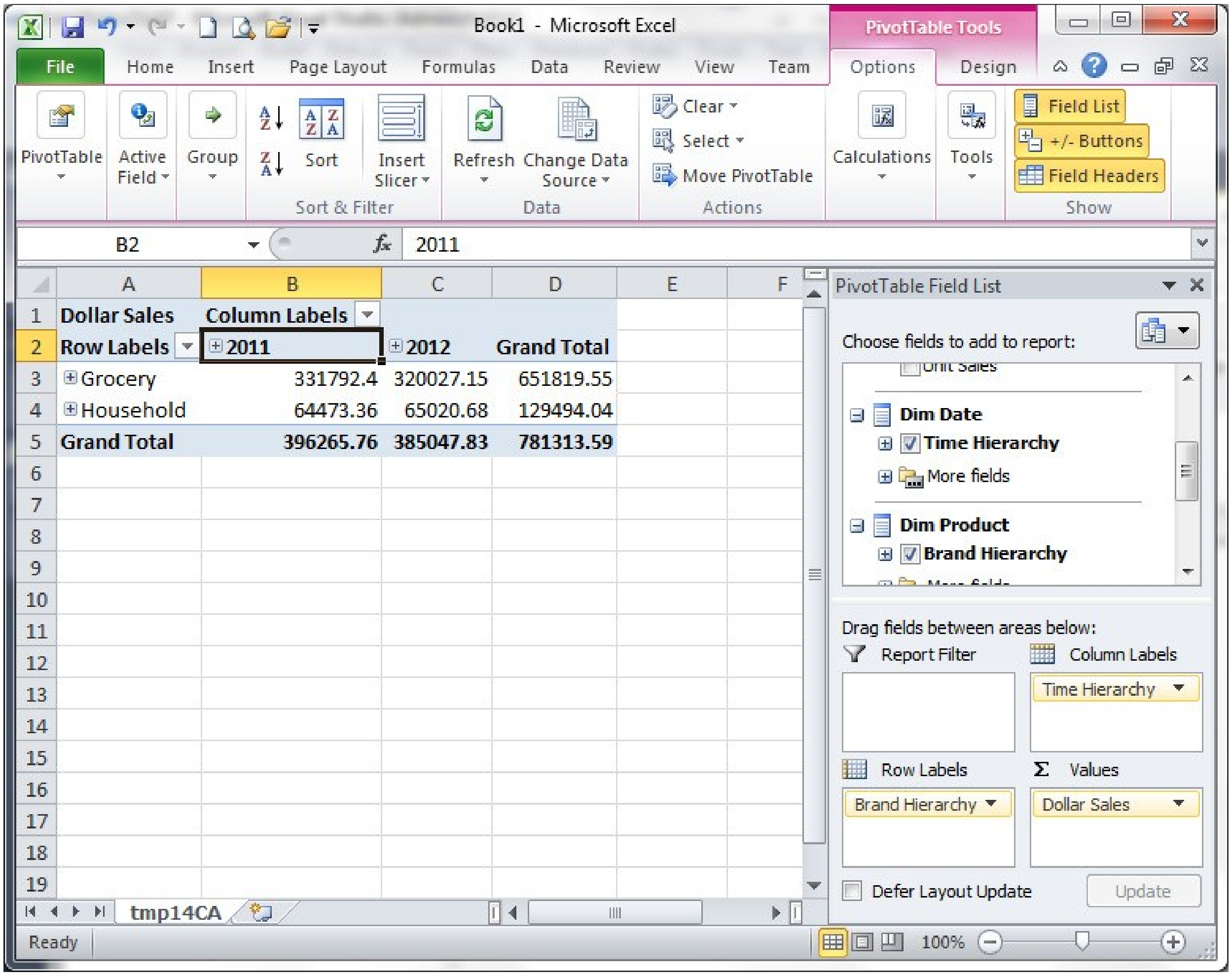
Deployment Completed Successfully

Deploy succeeded

Browse the Cube

At this point, it is possible to begin browsing the multidimensional cube you have created. Within Visual Studio, the browse feature allows you to manipulate the cube as a business user would. Notice that you are already on the Browse tab and are ready to browse the cube.

Notice the little Excel icon in the top left corner. You can click this button to view the cube in Excel. Using excel, drag and drop dimension values into the rows and columns and measure values into the main body. Try to create a basic report of Dollar Sales by Year and Department like in the picture below. At this point, you can “slice-and-dice” the data in any way that makes sense.



	A	B	C	D
1	Dollar Sales	Column Labels		
2	Row Labels	⊞2011	⊞2012	Grand Total
3	⊞Grocery	331792.4	320027.15	651819.55
4	⊞Household	64473.36	65020.68	129494.04
5	Grand Total	396265.76	385047.83	781313.59

Column Labels: Time Hierarchy

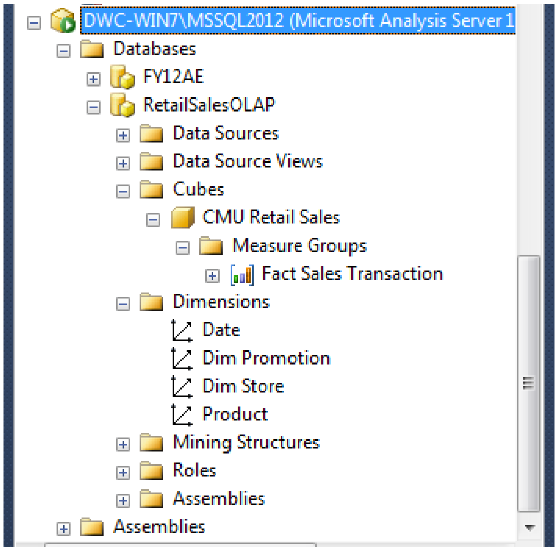
Row Labels: Brand Hierarchy

Values: Dollar Sales

View OLAP Database

When you created and deployed the cube, you actually created an OLAP database in SQL Server. We can see this in the SQL Server management studio. Once the database is created and deployed, you no longer need Visual Studio to do any further design.

1. Switch back to the Microsoft SQL Server Management Studio. (If you closed it previously, then restart it.)
2. At the top left, choose "Connect 7 Analysis Services…" Type in localhost or your computer name and click "Connect".
3. In your Object Explorer at the left, you will now see the Retail Sales OLAP listed under the Analysis Services databases. If you expand the folders, it will look something like this.



- DWC-WIN7\MSSQL2012 (Microsoft Analysis Server 1
  - Databases
    - FY12AE
    - RetailSalesOLAP
      - Data Sources
      - Data Source Views
      - Cubes
        - CMU Retail Sales
          - Measure Groups
            - Fact Sales Transaction
      - Dimensions
        - Date
        - Dim Promotion
        - Dim Store
        - Product
      - Mining Structures
      - Roles
      - Assemblies
  - Assemblies

4. This view allows you to see facts and dimensions in the database, but you cannot do full manipulation of the database from within SQL Server. Development tasks must be done in back in Visual Studio. **Screenshot Here**

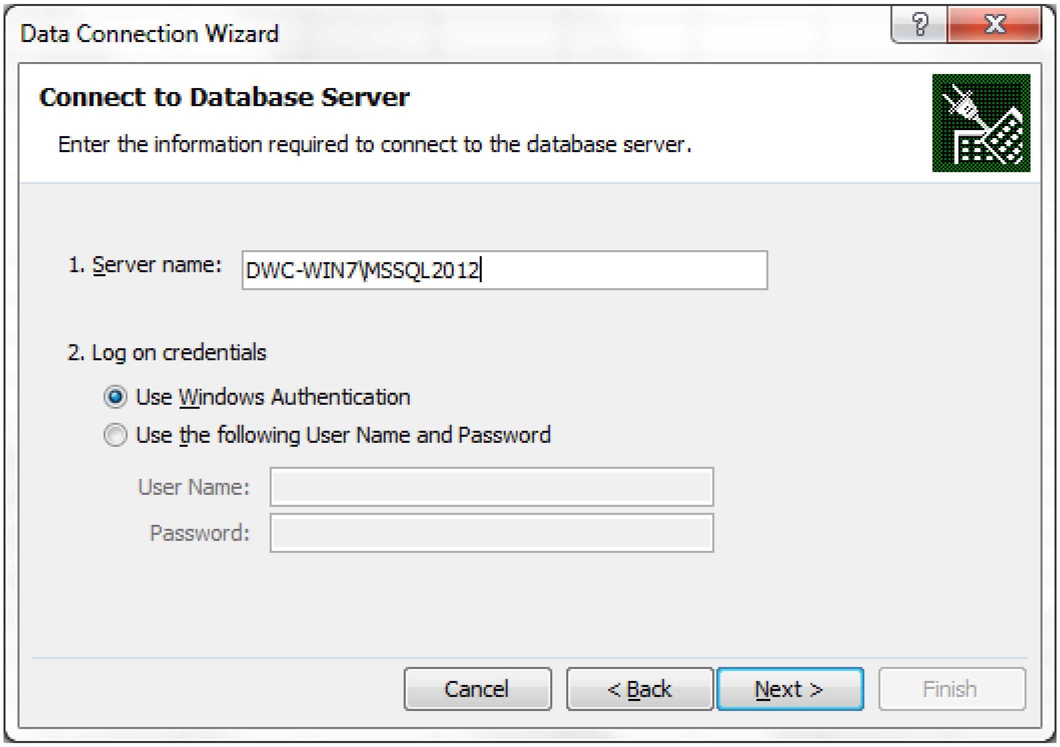
   One item you can do here is refresh the cube. To do this, right-click on the cube you want to refresh and choose “Process”. Follow this short wizard to refresh the OLAP database from the source. This would be required every time the source data changes.

Connect to OLAP Cube with MS Excel

Now that you have created the OLAP cube, you really have only created the “back-end” database. Almost every OLAP application will have some type of front-end tool that can be used to access the data. The “Browse” feature in Visual Studio is one such took, but most business users will not have this software.

For Microsoft OLAP, Microsoft Excel can be used as the front-end tool. This short example shows you how you can use Excel to access the power of OLAP.

1. Start Microsoft Excel. (Note, for this Lab I am demonstrating Microsoft Excel 2010. The same tasks can be done in earlier versions of Excel, but it is accomplished slightly differently.
2. Go to the “Data” ribbon and choose “From Other Sources 7 Analysis Services”.
3. In the “Connect to Database Server” dialog box, type localhost for the server name and click “Next”.



Data Connection Wizard

Connect to Database Server

Enter the information required to connect to the database server.

1. Server name: DWC-WIN7\MSSQL2012

2. Log on credentials

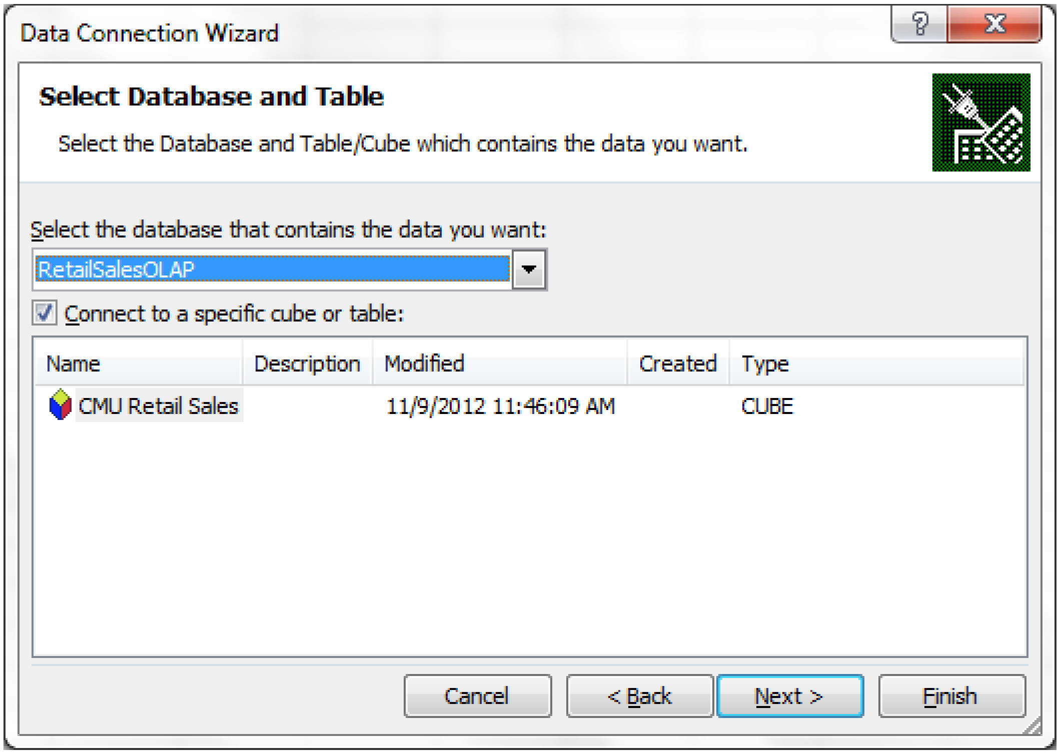
● Use Windows Authentication

○ Use the following User Name and Password

User Name:

Password:

4. Choose the Retail Sales OLAP database and click “Next”.



Data Connection Wizard

Select Database and Table

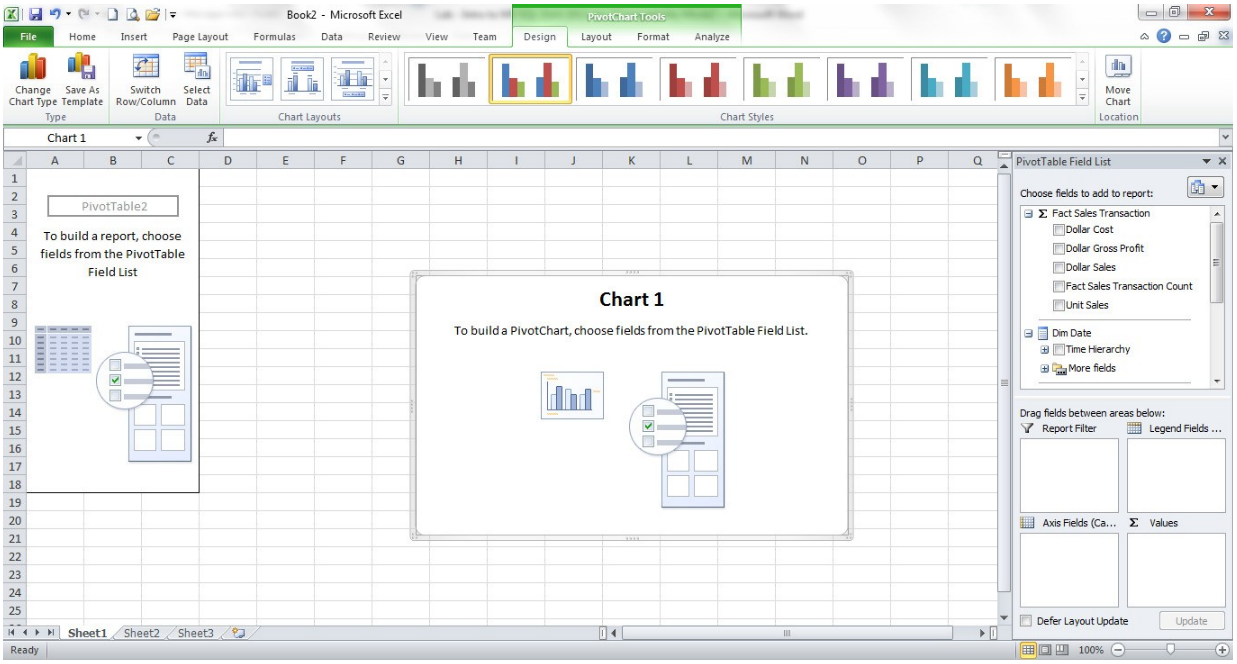
Select the Database and Table/Cube which contains the data you want.

Select the database that contains the data you want: RetailSalesOLAP

☑ Connect to a specific cube or table:

Name	Description	Modified	Created	Type
CMU Retail Sales		11/9/2012 11:46:09 AM		CUBE

5. In the next dialog box, click “Finish”.
6. In the next box, select “PivotChart and PivotTable Report” and click OK. Your screen should now look something like this. **Screenshot Here**



PivotTable2

To build a report, choose fields from the PivotTable Field List

Chart 1

To build a PivotChart, choose fields from the PivotTable Field List.

PivotTable Field List

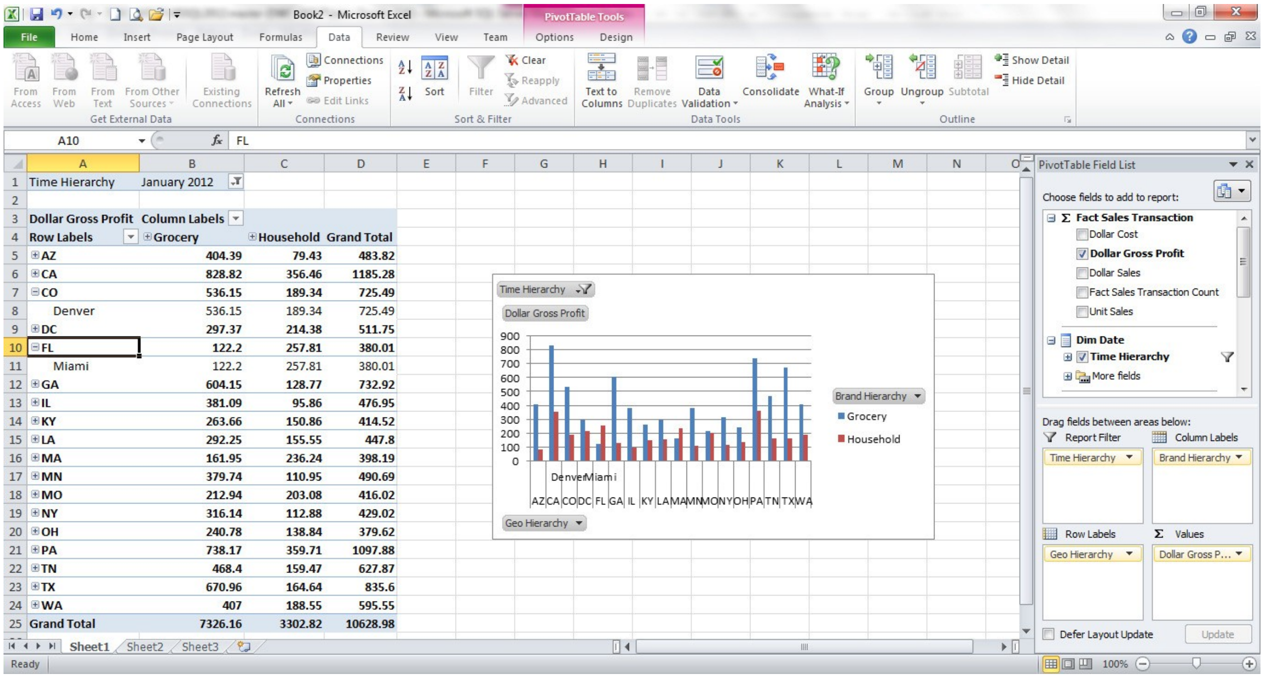
Choose fields to add to report:

- Fact Sales Transaction
  - □ Dollar Cost
  - □ Dollar Gross Profit
  - □ Dollar Sales
  - □ Fact Sales Transaction Count
  - □ Unit Sales
- Dim Date
  - □ Time Hierarchy
  - More fields

Drag fields between areas below:

□ Defer Layout Update

7. With a few clicks in the right panel, it is not easy to produce an OLAP report showing the total Gross Profit by State/City for each department for January 2012. The report would look something like this. **Screenshot Here**



	A	B	C	D
1	Time Hierarchy	January 2012		
2				
3	Dollar Gross Profit	Column Labels		
4	Row Labels	Grocery	Household	Grand Total
5	AZ	404.39	79.43	483.82
6	CA	828.82	356.46	1185.28
7	CO	536.15	189.34	725.49
8	Denver	536.15	189.34	725.49
9	DC	297.37	214.38	511.75
10	FL	122.2	257.81	380.01
11	Miami	122.2	257.81	380.01
12	GA	604.15	128.77	732.92
13	IL	381.09	95.86	476.95
14	KY	263.66	150.86	414.52
15	LA	292.25	155.55	447.8
16	MA	161.95	236.24	398.19
17	MN	379.74	110.95	490.69
18	MO	212.94	203.08	416.02
19	NY	316.14	112.88	429.02
20	OH	240.78	138.84	379.62
21	PA	738.17	359.71	1097.88
22	TN	468.4	159.47	627.87
23	TX	670.96	164.64	835.6
24	WA	407	188.55	595.55
25	Grand Total	7326.16	3302.82	10628.98

PivotTable Field List

Choose fields to add to report:

Σ Fact Sales Transaction
- □ Dollar Cost
- ☑ Dollar Gross Profit
- □ Dollar Sales
- □ Fact Sales Transaction Count
- □ Unit Sales

Dim Date
- ☑ Time Hierarchy
- More fields

Drag fields between areas below:

Report Filter: Time Hierarchy

Column Labels: Brand Hierarchy

Row Labels: Geo Hierarchy

Σ Values: Dollar Gross P...

□ Defer Layout Update   Update

8. At this point the quest for information is now in the hands of the business user. They can very easily manipulate the cube to identify trends and other data realities. This is the primary purpose of OLAP cubes and data warehousing in general – to provide quick, reliable data access to business users to help to run the business better.

On Your Own

Now that you have seen the processing for creating and running an SSAS cube, you can experiment further with the tool. You can continue on your own with the following steps if you would like more practice

- (Easy) Add all relevant attributes to the Promotion Dimension. **Screenshot Here**
- (Moderate) Add geographic attributes to the Store Dimension. Create a hierarchy that allows you to drill down from Region 7 State 7 Zip Code. **Screenshot Here**
- (Difficult) Experiment with the various measures. Try to create a custom Measure that calculates Gross Margin correctly for all levels of the hierarchies. **Screenshot Here**

Credits to the following:
Bob Brichacek (CMU, 2018) – Source of Lecture Notes and Activities
David Ybanez (MSIT Candidate) – For Review and Adjustments based on the Compatibility based on latest versions.