Lab 1 of 2 Exercise

Microsoft SQL Server Lab Intro to SQL Server Integration Services (SSIS)

READ THE ENTRE DOCUMENT FIRST BEFORE STARTING THE LAB ACTIVITY.

This lab is intended to introduce you to an ETL tool that is often used in data warehouse projects. As part of its SQL Server 2012 suite, Microsoft contains Integration Services which is a greatly enhanced version of their Data Transformation Services (DTS) in previous versions of SQL Server.

SQL Server Integration Services (SSIS) will likely become one of the leading tools in the market for ETL operations because it completes most of the same functions as its competitors – like Informatica PowerMart – but it comes bundled with the database.

Regardless of its place in the market, it operates in a similar manner to these other tools so learning one will definitely aid learning another in the future. The goal of this lab is to give you some preliminary exposure to how these tools operate. This information will also be helpful for the ETL in your data warehouse projects.

SQL Server

- SQL Server 2019 Developer Edition (15.0.2000.5)
- SQL Server Configuration Manager (2019.0150.2000.05)
- SQL Server Management Studio (15.0.18369.0 or v18.8)
- Microsoft Analysis Service Client Tools (15.0.19342.0)

Visual Studio

- Visual Studio Community 2019 (16.9.2)
- SQL Server Data Tools (16.0.62103.10080)
- Microsoft Analysis Services Projects (2.9.16) - Extension
- SQL Server Integration Services Project (3.12) - Extension

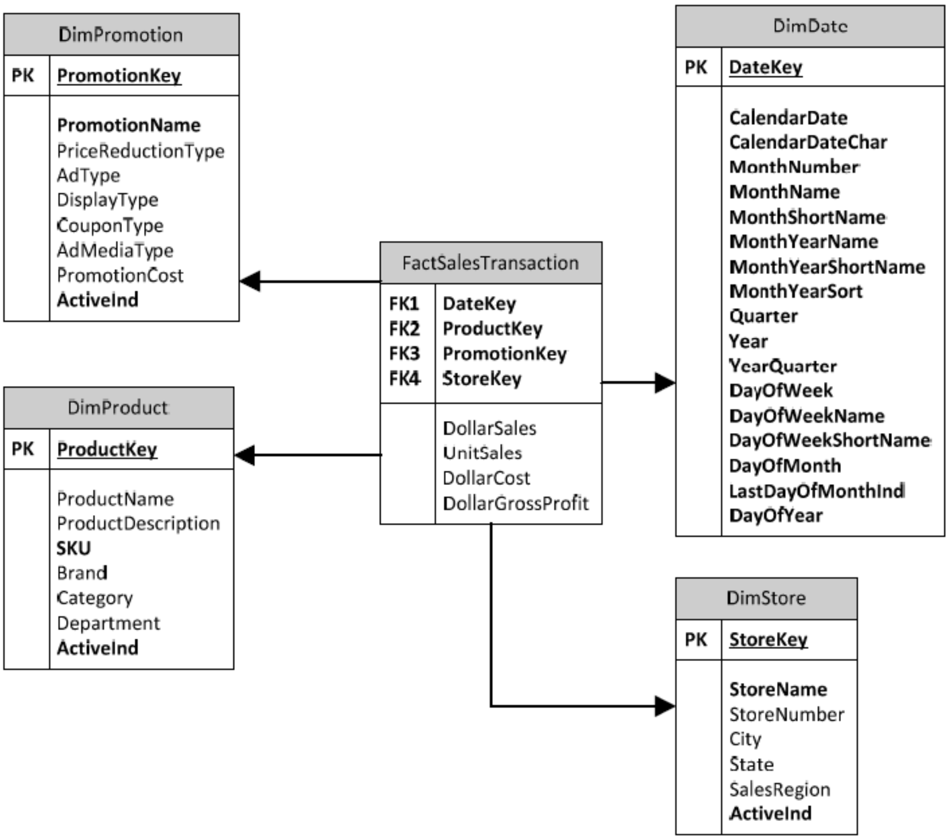
Tip: Using the latest Visual Studio, download the SSDT from the Visual Studio Installer. To install SSIS, open Visual Studio and navigate to Extensions -> Manage Extensions. Search for SQL Server Integration Services Project. Or download it from https://marketplace.visualstudio.com/

Operating System

- Windows 10 (10.0.19042)

Database Schema

We will work with the same example and data model we have used in the lectures – the
retail store transactions. The data model is shown again below.



INSTRUCTIONS:

This laboratory exercise is divided into:

1. Extracting the Lab Files
2. Creating the Database
3. Getting Started – You need to provide screenshots for this section.
4. Load the Dimensions – You need to provide screenshots for this section.
5. Slowly Changing Dimensions– You need to provide screenshots for this section.
6. On your Own– You need to provide screenshots for this section.

Screenshot Here : Should be properly labelled and include the entire screen. Since it is done in PDF, I can just zoom as long as the quality is OK.

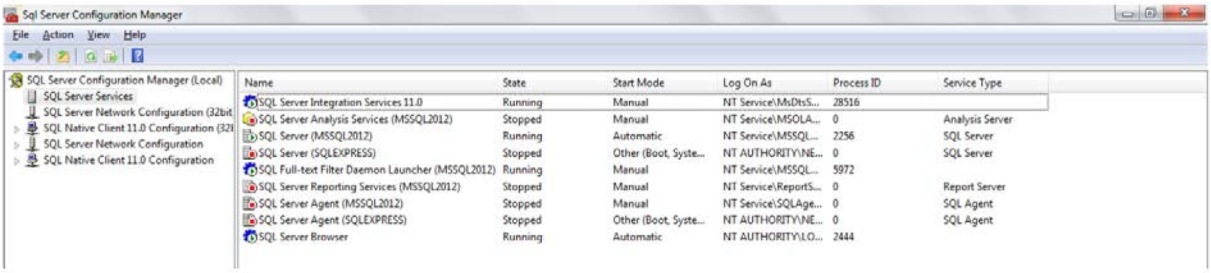
I. Extract the Lab Files

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

II. Create the Database

In this lab, we will begin with an empty database schema and load it using the tools available in SSIS. The starting point will essentially be a blank schema. I have provided scripts for you to create the database model shown above.

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

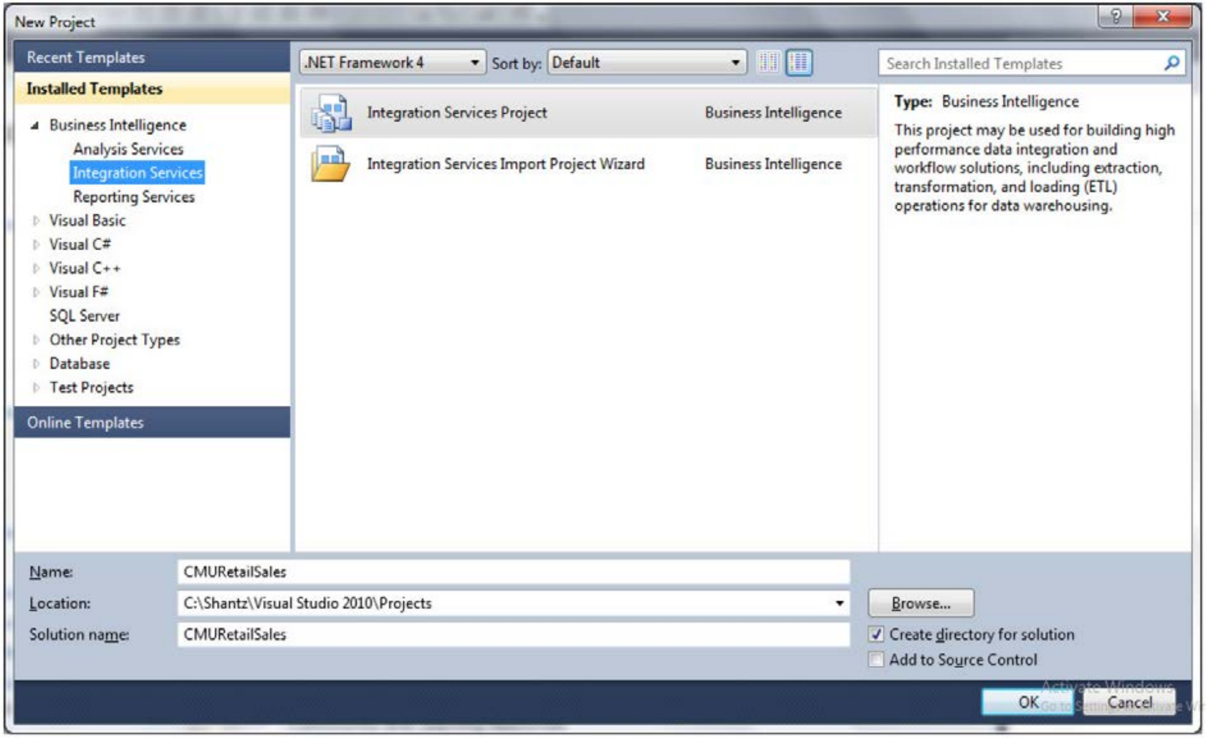
Name	State	Start Mode	Log On As	Process ID	Service Type
SQL Server Integration Services 11.0	Running	Manual	NT Service\MsDtsS...	28516	
SQL Server Analysis Services (MSSQL2012)	Stopped	Manual	NT Service\MSOLA...	0	Analysis Server
SQL Server (MSSQL2012)	Running	Automatic	NT Service\MSSQL...	2256	SQL Server
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...	5972	
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...	2444	

Make sure both the SQL Server (MSSQLSERVER) and SQL Server Integration Services show a state of running. If not, start the services.

2. Connect to SQL Server. To do this, go to Start > Programs > Microsoft SQL Server 2012 > SQL Server Management Studio.
3. Right-click on the Databases folder and choose “New Database…”
4. Type in CMURetailSales for the database name and click OK. This will create your database.
5. Click on the CMURetailSales database on the left and expand it to show the items underneath. Click on the Tables folder. In the upper-right corner of the screen, select “New Query”. You should get a main blank page.
6. First, we will create the dimensions. Open the 01_Create_Dimensions.sql file. One-by-one, execute the commands in the file to create your dimension tables. To execute the queries, copy/paste the query into the SQL window and click the execute (!) button. Notice the syntax for the create table statements, and the identity columns in particular. Also notice the default values for the ActiveInd field in some of the dimensions.
7. Next, we will create the fact table. Open the 01_Create_Fact.sql file. Execute the commands in the file to create your fact table.
8. When the tables are created, right-click on the Tables folder again and choose “Refresh” to make sure all your tables have been successfully created. If you right-click on a table and select “Modify”, you can review the structure of the tables.

III. Getting Started

1. Start the SQL Server Data Tools. Go to Start > Programs > Microsoft SQL Server 2012 > SQL Server Data Tools. If you do not have this option available, then you likely have not installed the correct components of SQL Server.
2. Create a new project. On the File menu, choose New Project. In the New Project dialog box, choose Business Intelligence Projects on the left menu and select the option for Integration Services project. In the bottom boxes, name your project "CMU Retail Sales". Click OK when you are done (see image below). **Screenshot Here**



Type: Business Intelligence

This project may be used for building high performance data integration and workflow solutions, including extraction, transformation, and loading (ETL) operations for data warehousing.

Name: CMURetailSales

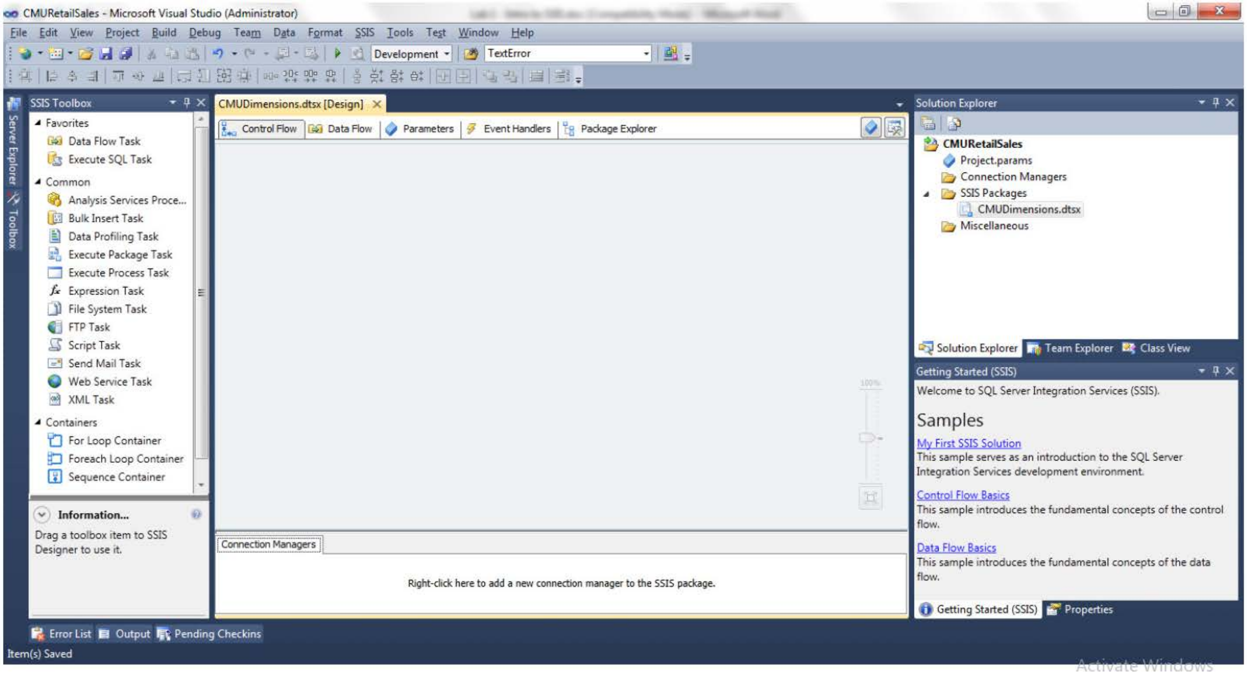
Location: C:\Shantz\Visual Studio 2010\Projects

Solution name: CMURetailSales

☑ Create directory for solution

□ Add to Source Control

3. Rename the package. Right-click on the package.dtsx item on the right of the screen and choose "Rename". Rename the package to be "CMUDimensions.dtsx". Save the package. You are now ready to begin development of your ETL code. **Screenshot Here**



Drag a toolbox item to SSIS Designer to use it.

Right-click here to add a new connection manager to the SSIS package.

Welcome to SQL Server Integration Services (SSIS).

Samples

My First SSIS Solution
This sample serves as an introduction to the SQL Server Integration Services development environment.

Control Flow Basics
This sample introduces the fundamental concepts of the control flow.

Data Flow Basics
This sample introduces the fundamental concepts of the data flow.

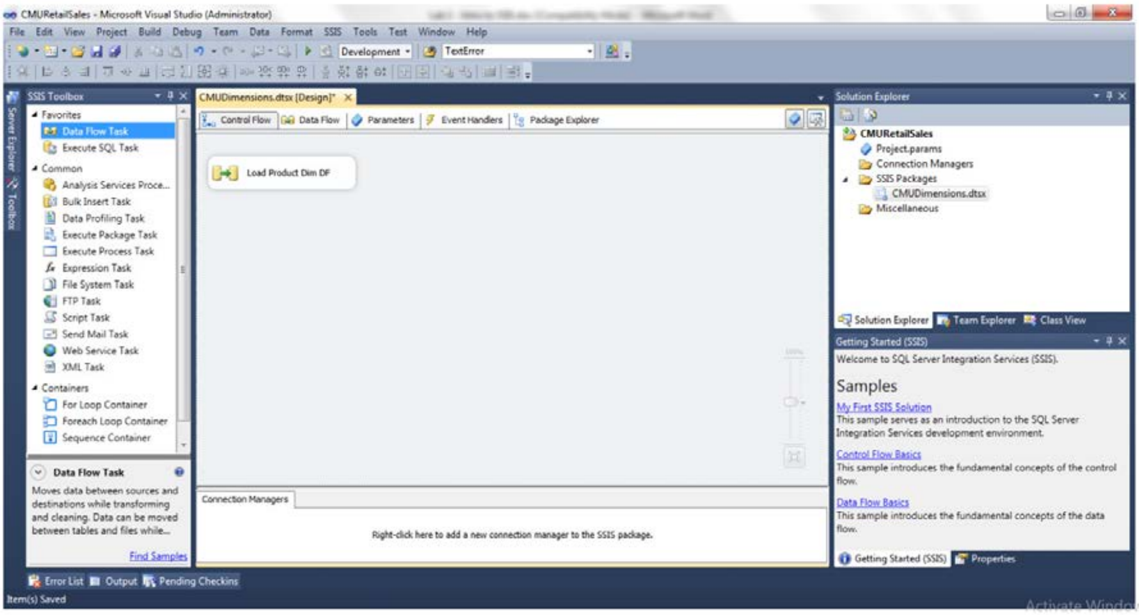
Item(s) Saved

IV. Load the Dimensions

The first step in loading a dimensional model is to load the dimensions. This contains the descriptive attribute information, so it must be loaded first. Once the dimensions are loaded, fact data can be added to the fact tables. The first dimension we will load is the product dimension.

Load the Product Dimension

1. Double-click the Data Flow Task and you will enter a screen to edit the mapping. We will load the product dimension and most other tables using a “Data Flow Task” in SSIS. A data flow task allows you to move data from a source through various transformations and finally to a target table. In the toolbox on your Visual Studio 2005 screen, drag a Data Flow Task onto the main page – I will refer to the main page as the “canvas” for the mapping. Once you have moved the Data Flow Task onto your page, rename it to “Load Product Dim DF”. See the image below.



Moves data between sources and destinations while transforming and cleaning. Data can be moved between tables and files while...

Right-click here to add a new connection manager to the SSIS package.

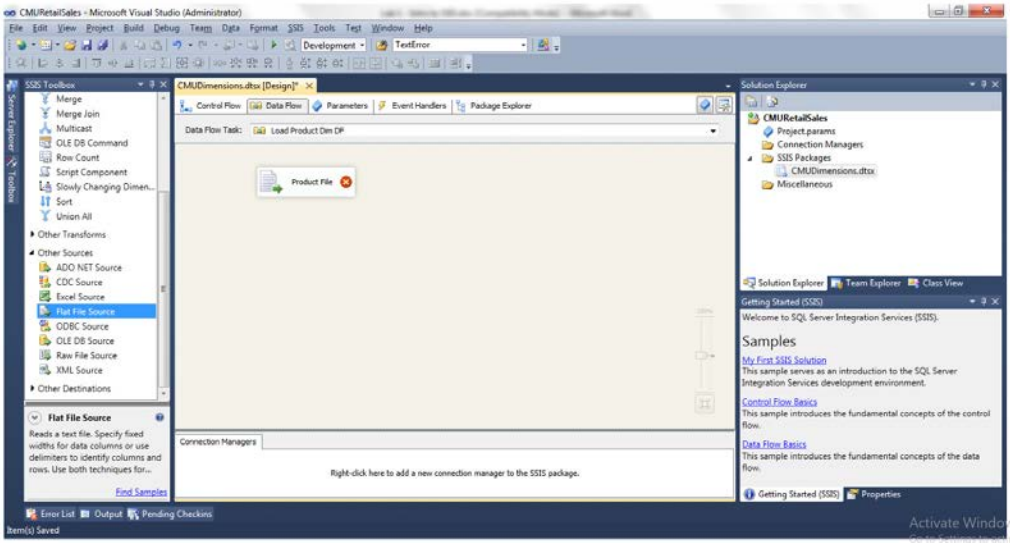
Welcome to SQL Server Integration Services (SSIS).

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 concepts of the data flow.

2. Double-click the Data Flow Task and you will enter a screen to edit the mapping. Notice the highlighting will change to “Data Flow” on the tabs at the top of the screen.
3. From the toolbox, drag a “Flat File Source” control onto the canvas. Rename the control to “Product File”. **Screenshot Here**



Flat File Source

Reads a text file. Specify fixed widths for data columns or use delimiters to identify columns and rows. Use both techniques for...

Right-click here to add a new connection manager to the SSIS package.

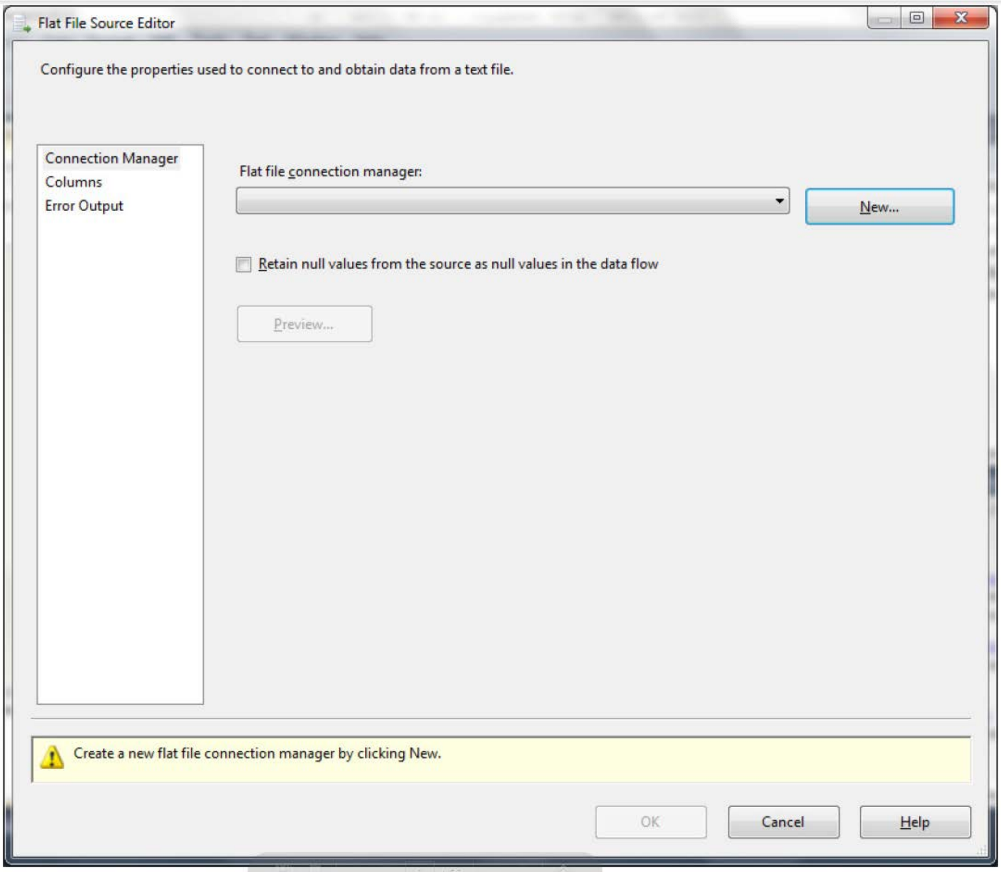
Welcome to SQL Server Integration Services (SSIS).

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 concepts of the data flow.

4. First we need to establish a connection to the source data file. Double-click the Flat File control. In the box that comes up, click the "New…" button next to the Flat File Connection Manager drop down list.



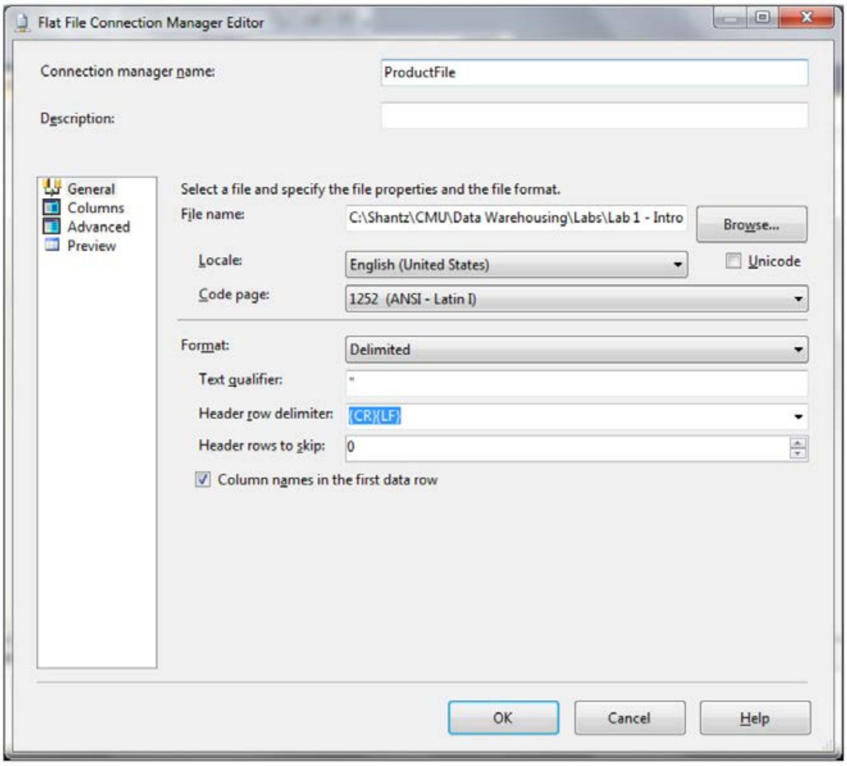
Configure the properties used to connect to and obtain data from a text file.

Flat file connection manager:

□ Retain null values from the source as null values in the data flow

Create a new flat file connection manager by clicking New.

(a) In the resulting screen, click Browse and navigate to the lab folder directory and select the *Data_Product.txt* file.
(b) In the Connection Manager Name box, type ProductFile.
(c) Check the column names in first data row checkbox
(d) Your screen should resemble the following one.



Connection manager name: ProductFile

Description:

Select a file and specify the file properties and the file format.

File name: C:\Shantz\CMU\Data Warehousing\Labs\Lab 1 - Intro

Locale: English (United States)

□ Unicode

Code page: 1252 (ANSI - Latin I)

Format: Delimited

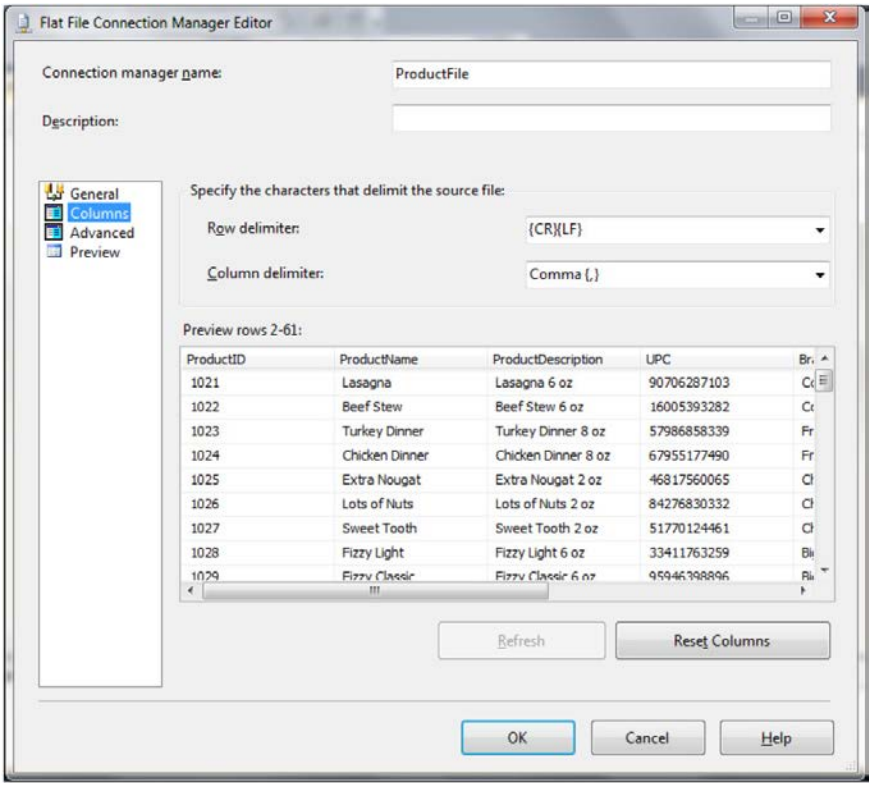
Text qualifier: "

Header row delimiter: {CR}{LF}

Header rows to skip: 0

☑ Column names in the first data row

5. On the left of this screen, click on the columns tab. Make sure the columns show up and the data looks as you expect.



Connection manager name: ProductFile

Description:

Specify the characters that delimit the source file:

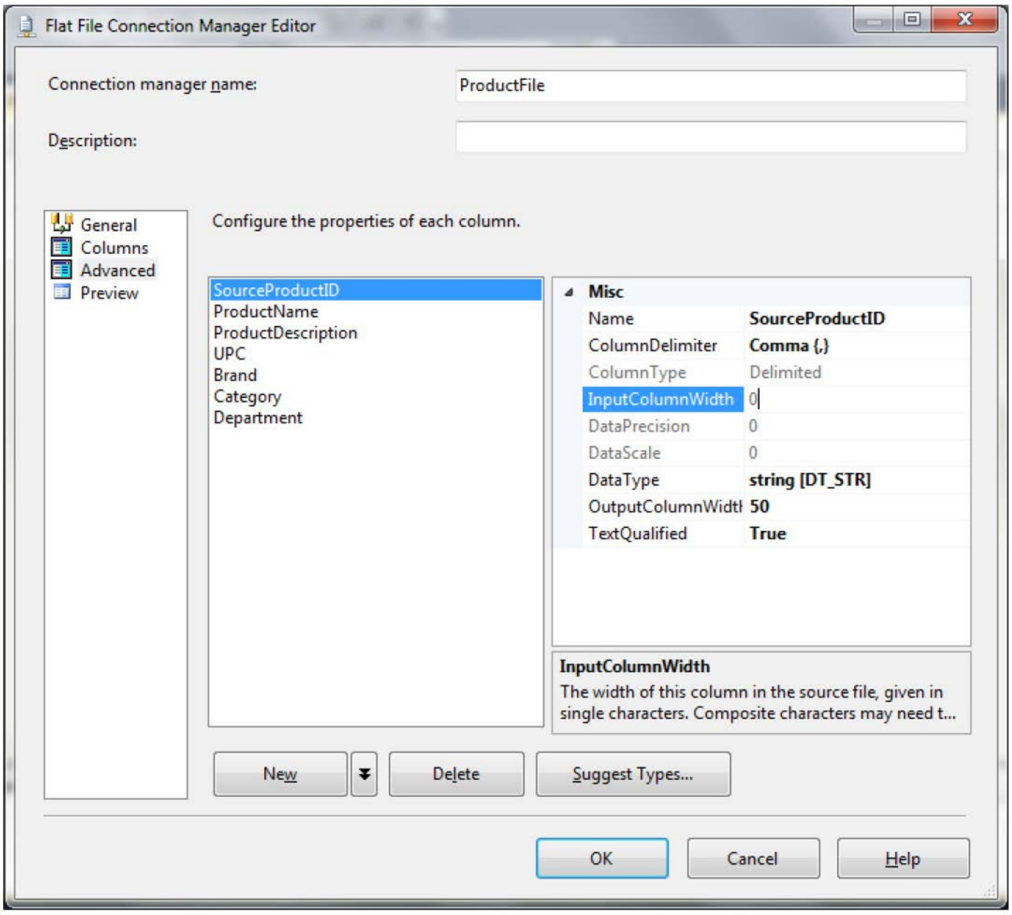
Row delimiter: {CR}{LF}

Column delimiter: Comma {,}

Preview rows 2-61:

ProductID	ProductName	ProductDescription	UPC	Br.
1021	Lasagna	Lasagna 6 oz	90706287103	Cc
1022	Beef Stew	Beef Stew 6 oz	16005393282	Cc
1023	Turkey Dinner	Turkey Dinner 8 oz	57986858339	Fr
1024	Chicken Dinner	Chicken Dinner 8 oz	67955177490	Fr
1025	Extra Nougat	Extra Nougat 2 oz	46817560065	Cl
1026	Lots of Nuts	Lots of Nuts 2 oz	84276830332	Cl
1027	Sweet Tooth	Sweet Tooth 2 oz	51770124461	Cl
1028	Fizzy Light	Fizzy Light 6 oz	33411763259	Bi
1029	Fizzy Classic	Fizzy Classic 6 oz	95946398896	Bi

6. On the left of this screen, click on the Advanced tab. This tab allows you to set the name of fields and their respective data types. Do not modify the data type values but set the column names to the values in the file as shown in the following image.



Connection manager name: ProductFile

Description:

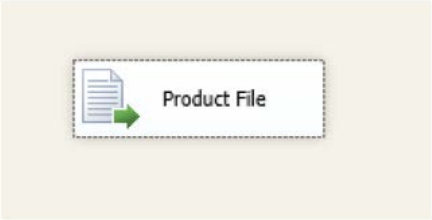
Configure the properties of each column.

SourceProductID
ProductName
ProductDescription
UPC
Brand
Category
Department

Misc	
Name	SourceProductID
ColumnDelimiter	Comma {,}
ColumnType	Delimited
InputColumnWidth	0
DataPrecision	0
DataScale	0
DataType	string [DT_STR]
OutputColumnWidtl	50
TextQualified	True

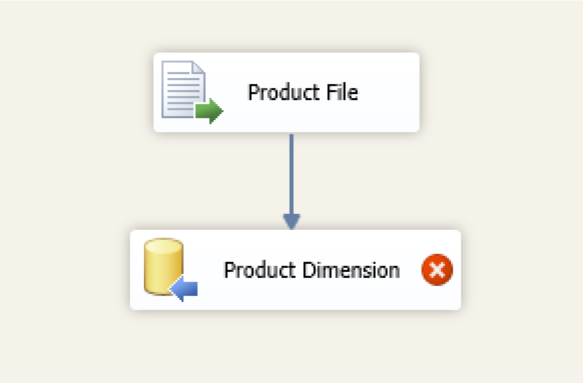
InputColumnWidth
The width of this column in the source file, given in single characters. Composite characters may need t...

7. Click OK. Click OK again. Your product file control should now not have any little error icons on it. **Screenshot Here**

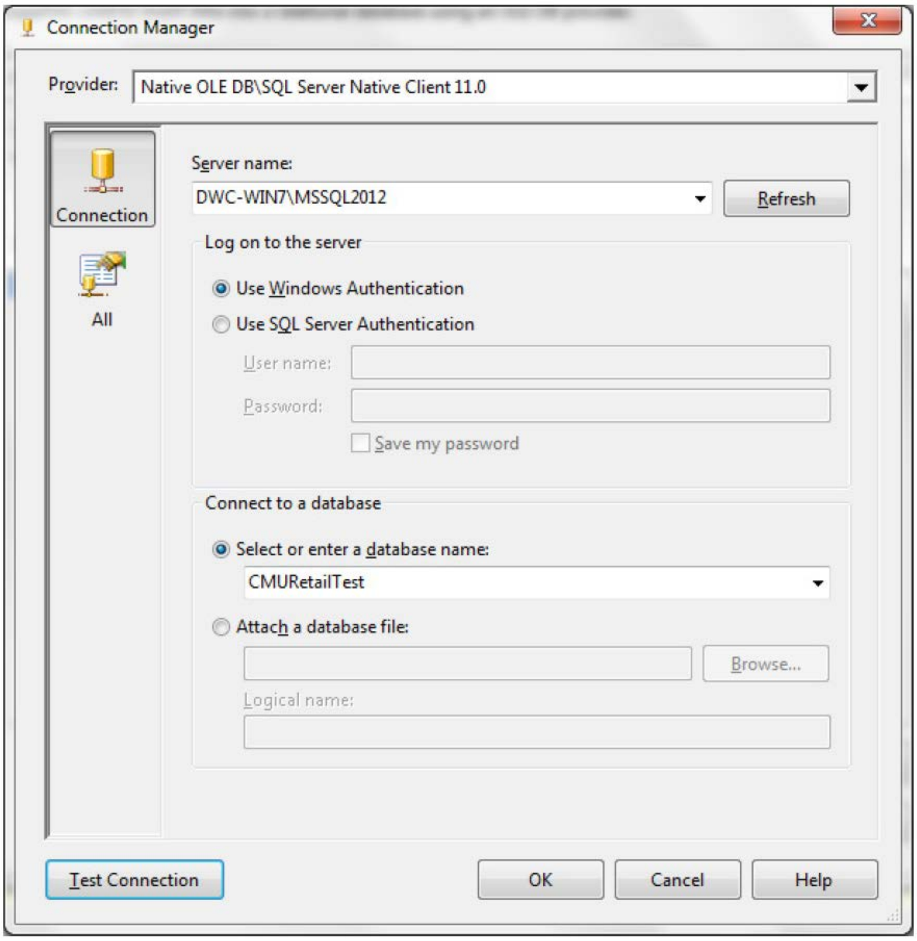


8. Now that we have created the source control, we will create a target control for where we want to move the data to. Since we will be loading a table in a database, we will use an "OLE DB Destination" control. Drag this control from the Toolbox on the left onto the canvas. Rename the control to "Product Dimension".

9. Click on the Product File control and you will see a blue line and a red line. The blue line is for normal data flow. The red line is for records that have errors. Click on the blue line and drag it to the Product Dimension control. Your picture will resemble this one. **Screenshot Here**



10. Now we must link the Product Dimension control to the database table we want to load. Double-click the Product Dimension control. Click the “New…” button next to the OLEDB Connection Manager drop down box. In the resulting box, click “New…” again.
    (a) For the Server Name, type in the name of your SQL Server instance that you used when you installed the software. (This is the same name that you connected to earlier in the SQL Management Studio.)
    (b) Select Use Windows Authentication
    (c) Select the database name CMURetailSales.
    (d) Click Test Connection and the test should succeed.
    (e) Your screen will resemble the following one.


Connection Manager

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

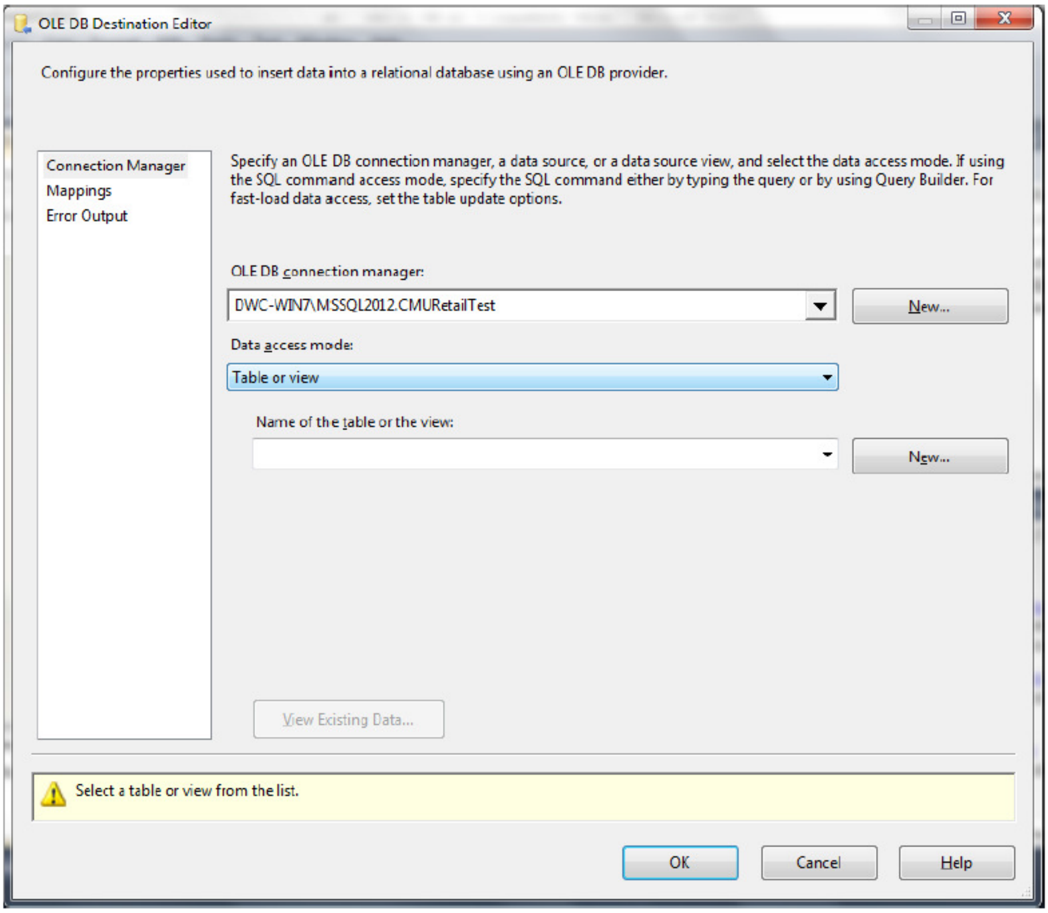
Connect to a database

● Select or enter a database name: CMURetailTest

○ Attach a database file:

Logical name:

11. Click OK and click OK again. You should be back on the OLEDB Destination Editor screen.



OLE DB Destination Editor

Configure the properties used to insert data into a relational database using an OLE DB provider.

Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.

OLE DB connection manager: DWC-WIN7\MSSQL2012.CMURetailTest

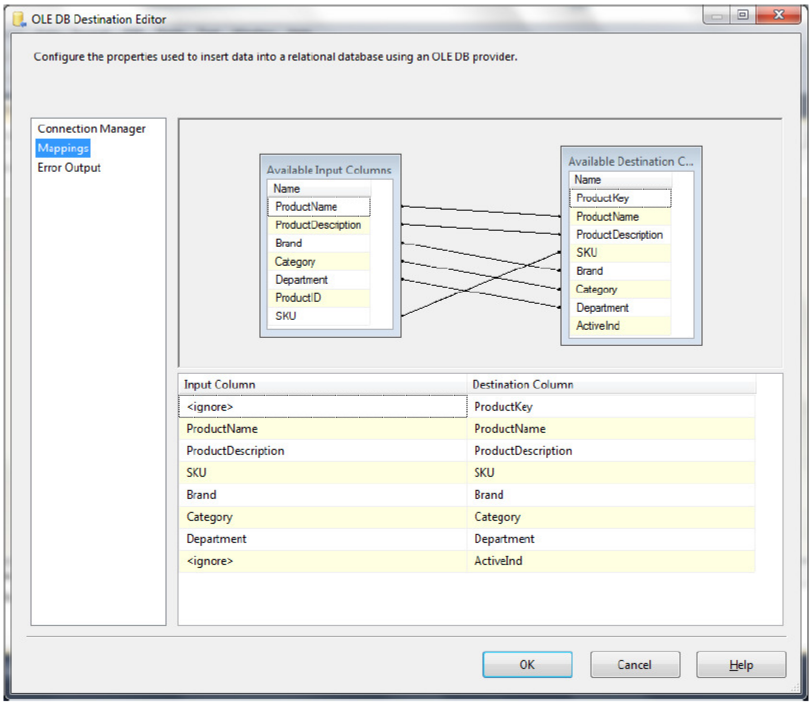
Data access mode: Table or view

Name of the table or the view:

Select a table or view from the list.

12. Click on the “Name of table or view” drop down and select “[dbo].[dim_product]”. This means you are selected the Product Dimension in your database.

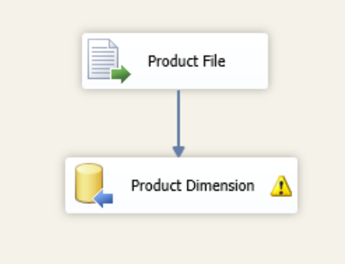
13. At the left, click on the Mappings item. This shows you the mappings that will be performed. Drag and drop columns on the left to the columns on the right until all the columns on the right are mapped EXCEPT ProductKey. Remember that we create our own keys in the data warehouse, so we do not want to load the value provided in the source because we don’t know what it is. (We know that SKU is the source system product key.) Your screen should look as follows:



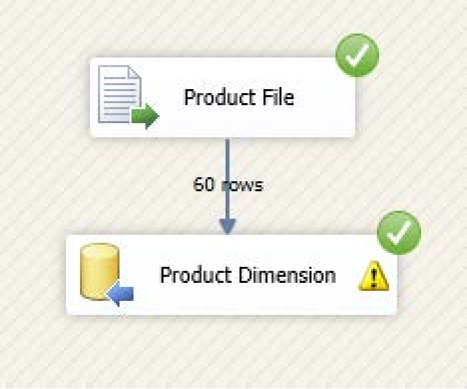
Configure the properties used to insert data into a relational database using an OLE DB provider.

Input Column	Destination Column
<ignore>	ProductKey
ProductName	ProductName
ProductDescription	ProductDescription
SKU	SKU
Brand	Brand
Category	Category
Department	Department
<ignore>	ActiveInd

14. Your canvas will now look as follows. Note that SSIS is giving you a warning that you may have some data truncation because you are loading a field that is 50 characters long into a field that is less than 50 characters. We will deal with this later. **Screenshot Here**



15. Now we are ready to test the mapping. To run the mapping, click the little green arrow at the top of the screen – it looks like the play button on DVD player. The mapping will go through some validations, and it should complete successfully. You will see the controls go yellow while they are running, and then you get a green check box when they complete. You will also see how many rows were loaded. Here is an example picture. **Screenshot Here**



16. Click the “stop” icon at the top of the screen to stop your mapping. Technically, this is a “debug” run of the mapping, so it must be stopped explicitly.
17. We will now verify that the rows were indeed loaded. Go back over to the SQL Enterprise Manager application. Right-click on the DimProduct table and select “Select Top 1000 Rows”. You should see that the 60 rows in the text file were indeed loaded into your table. **Screenshot Here**

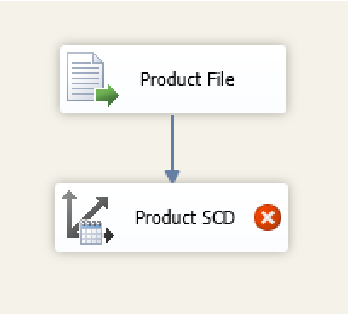
V. Slowly Changing Dimension Problem

1. Unfortunately, we are not done with the product dimension. As business analysts, we are aware that the product dimension may change over time and we need to handle that change. While the mapping is good for loading data one time, it will not work on an ongoing basis. Quickly run through the following steps to see why.
   (a) In the Microsoft Visual Studio click the “play” button to run the mapping again. What happens?
   (b) You can see that the mapping loads 60 records again.
   (c) Now go over to the SQL Enterprise Manager. Open the DimProduct table again. How many records are in the table now?
   (d) You can see that now we have 120 records in the product dimension even though nothing changed. We’ve now got duplicates of everything. This is not good.
2. To recover the mess we’ve created in the previous step, delete all the rows in the product dimension and we’ll start over. Type the following line in your sql window in the SQL Enterprise Manager and execute it. This will delete all records in your Product Dimension.
   a. DELETE FROM DimProduct;

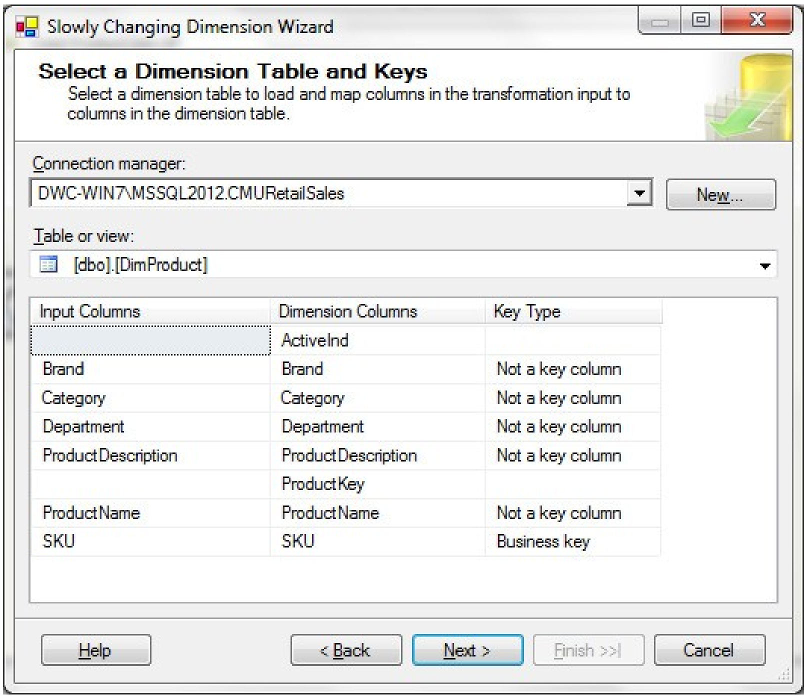
Make the Product Dimension a Slowly Changing Dimension

To prevent the records from loading every time, we will introduce some additional transformations into our mapping.

1. Fortunately for us, SSIS contains a control that solves the exact problem we are facing. It is called a “Slowly Changing Dimension” control.
   a. Drag one of these controls onto your canvas and name it “Product SCD”.
   b. Delete the existing Product Dimension control.
   c. Link the blue line from the Product File control onto the Product SCD control.
   d. Your screen should look as follows. **Screenshot Here**



2. Now we need to set up the SCD control. Double-click this control to begin.
   a. In the resulting box, select [dbo].[DimProduct] in the “Table or View” drop down list
   b. In the column list, make sure all the columns from the source are pointing to their respective targets.
   c. Make all the columns say “Not a Business Key” except for the SKU. Since that is the source key for our purposes, that column should say “Business Key”.
   d. Do not link the ActiveInd column.



Select a Dimension Table and Keys

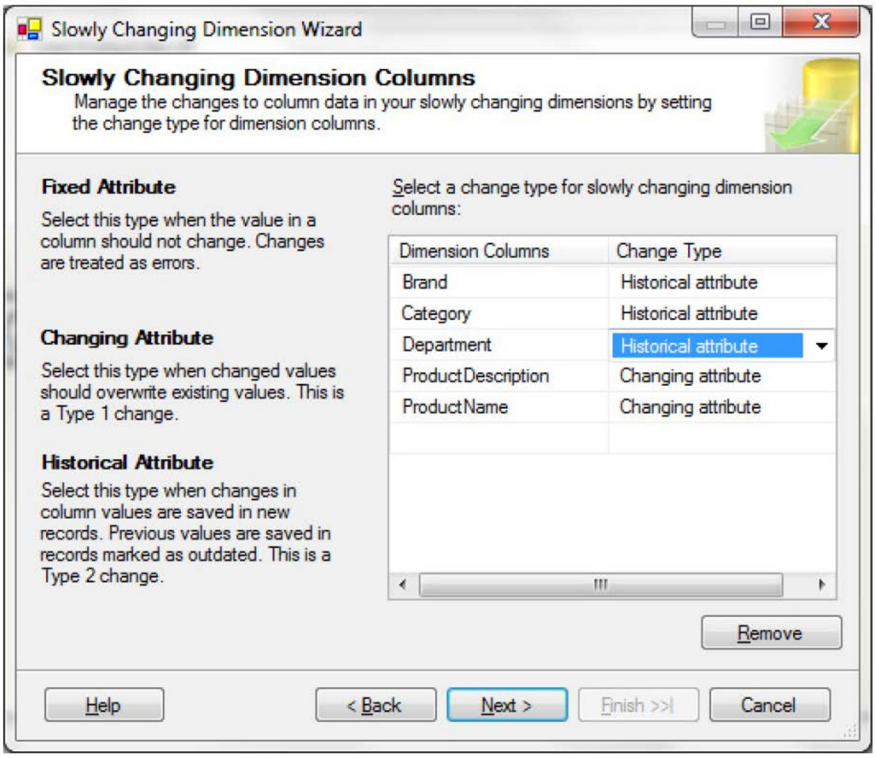
Select a dimension table to load and map columns in the transformation input to columns in the dimension table.

Connection manager: DWC-WIN7\MSSQL2012.CMURetailSales

Table or view: [dbo].[DimProduct]

Input Columns	Dimension Columns	Key Type
	ActiveInd	
Brand	Brand	Not a key column
Category	Category	Not a key column
Department	Department	Not a key column
ProductDescription	ProductDescription	Not a key column
	ProductKey	
ProductName	ProductName	Not a key column
SKU	SKU	Business key

3. Click Next. Now we must decide what update methodology we will use for each of the dimension fields. Let's assume for our purposes that we only need to track history (Type II updates) on a product when the Brand, Category or Department fields change. If the Product Name or Product Description changes, then we assume that is a correction that can be a Type I update. In the SSIS world, a Type I field is a "Changing Attribute" field, and a Type II field is a "Historical Attribute" field. Set up the next window so that Brand, Category and Department are historical attributes, and product name and description are changing attributes.



Slowly Changing Dimension Wizard

Slowly Changing Dimension Columns

Manage the changes to column data in your slowly changing dimensions by setting the change type for dimension columns.

Fixed Attribute

Select this type when the value in a column should not change. Changes are treated as errors.

Changing Attribute

Select this type when changed values should overwrite existing values. This is a Type 1 change.

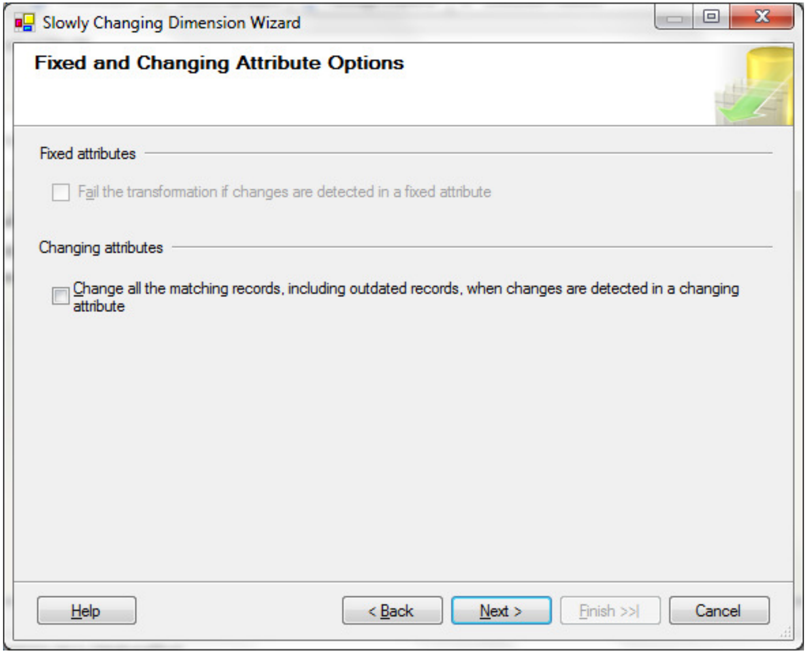
Historical Attribute

Select this type when changes in column values are saved in new records. Previous values are saved in records marked as outdated. This is a Type 2 change.

Select a change type for slowly changing dimension columns:

Dimension Columns	Change Type
Brand	Historical attribute
Category	Historical attribute
Department	Historical attribute
ProductDescription	Changing attribute
ProductName	Changing attribute

4. Click Next. This screen identifies how we want our Type 1 changes to affect records that are not current. No changes are required on this screen.



Slowly Changing Dimension Wizard

Fixed and Changing Attribute Options

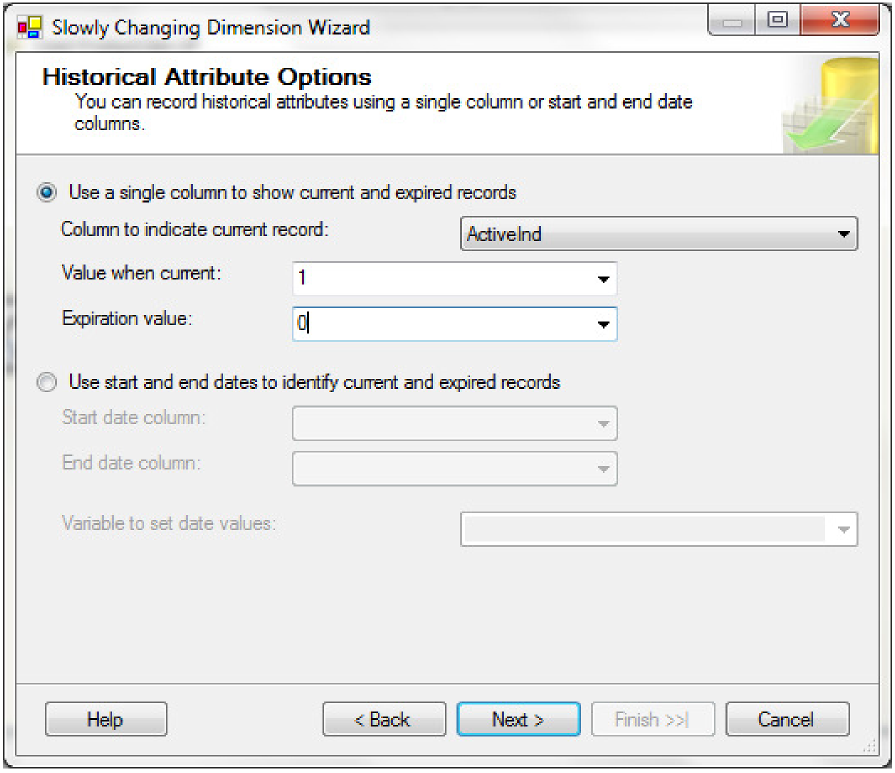
Fixed attributes

□ Fail the transformation if changes are detected in a fixed attribute

Changing attributes

□ Change all the matching records, including outdated records, when changes are detected in a changing attribute

5. On this screen we show SSIS how to distinguish the most current row in the dimension from older rows. In our dimensions that are slowly changing dimensions, we have a field called ActiveInd. This field is 1 when the record is current and 0 otherwise. Set up the next screen using this information. It will look as follows.



Slowly Changing Dimension Wizard

Historical Attribute Options

You can record historical attributes using a single column or start and end date columns.

● Use a single column to show current and expired records

Column to indicate current record: ActiveInd

Value when current: 1

Expiration value: 0

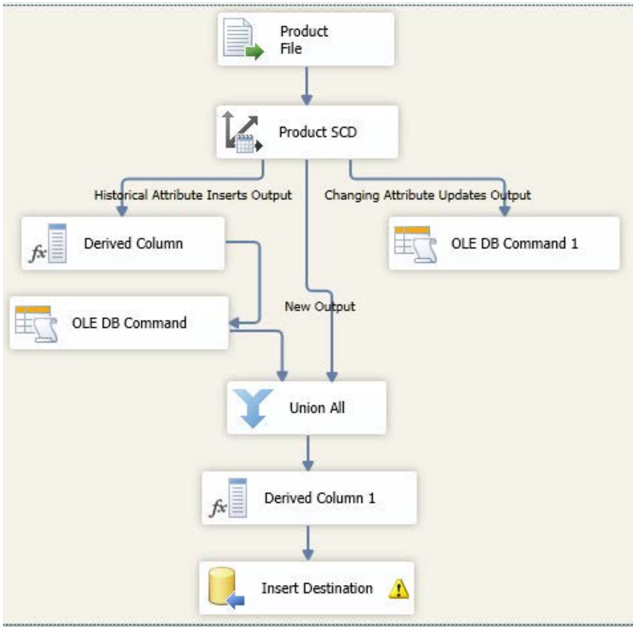
○ Use start and end dates to identify current and expired records

Start date column:

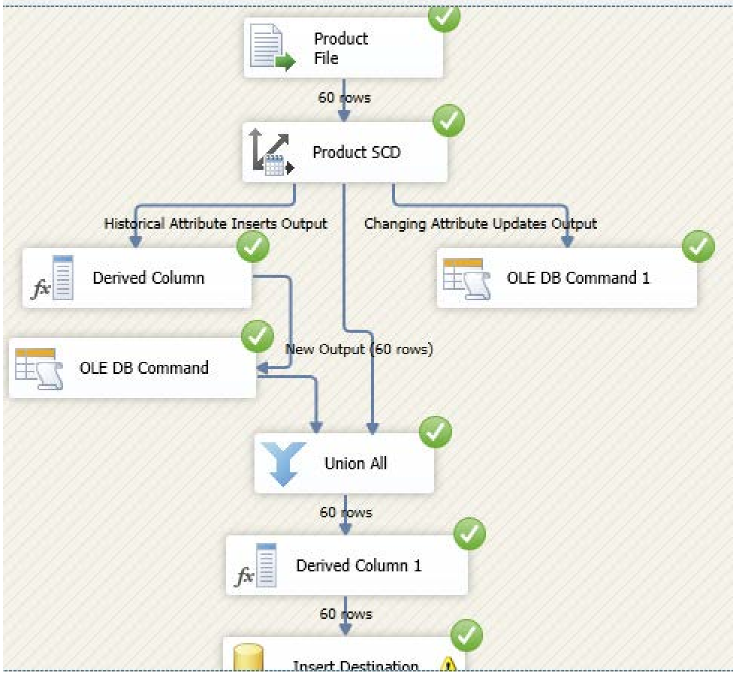
End date column:

Variable to set date values:

6. Click Next. Uncheck the box that says “Enable Inferred Member Support”. We do not need this feature for this example.
7. Click Next and Click Finish. SSIS will produce a number of new controls and transformations that will be something of a mess. There will be three outputs from the SCD Product control, however. There is one for Historical Attribute changes, one for Changing Attribute changes, and one for New Output which is records that have not been seen before. Organize the controls to look something like this. **Screenshot Here**

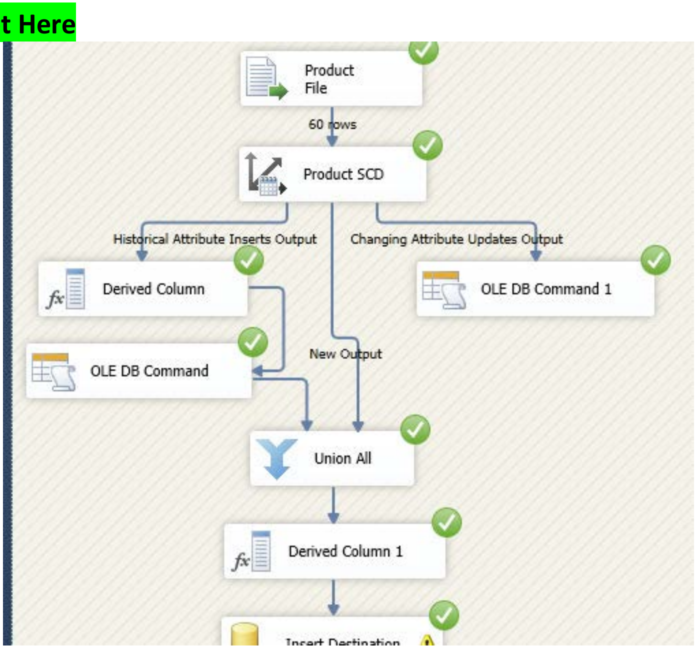


8. Now we are ready to run the mapping again. Click the "play" button at the top of the page to run the mapping. It will show you the number of records at each step, and it will look something like this. Click the stop button when you are done. **Screenshot Here**



9. In SQL Enterprise Manager, open the DimProduct table again. There should be 60 rows. **Screenshot Here**

10. In Visual Studio, run your mapping again. Notice now that it reads 60 rows from the source file, but those rows don't go anywhere. No data is loaded to the destination table. **Screenshot Here**

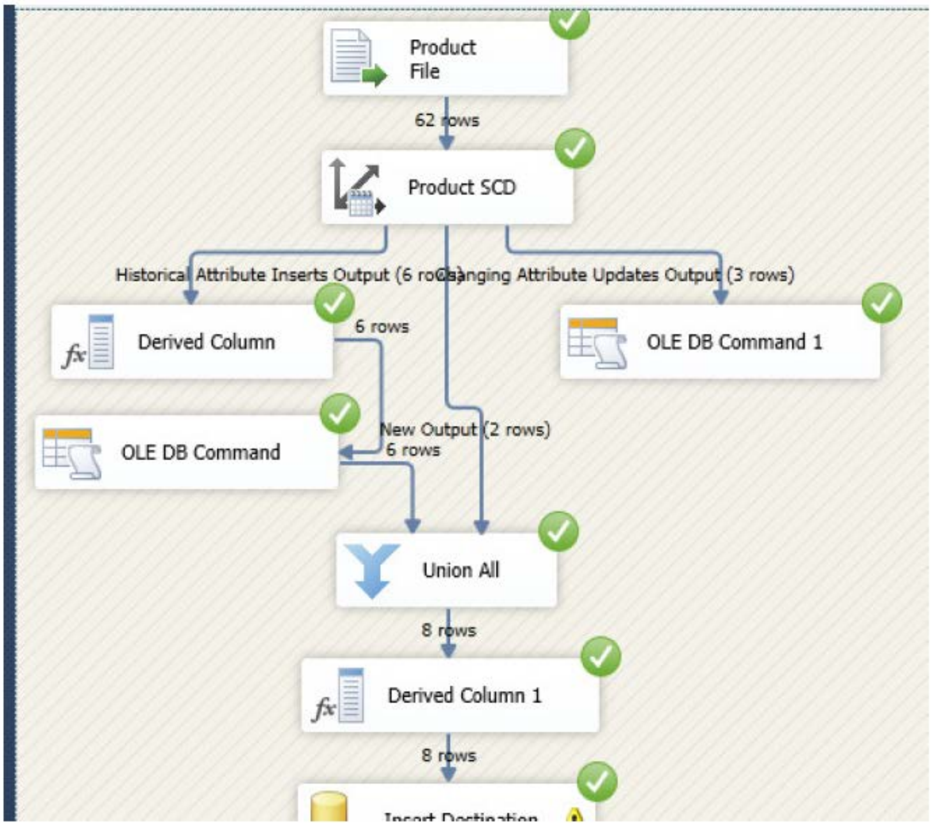


11. Now we will load the file that comes in the second period for the product dimension. There have been several changes to the system during this time:
    a. All paper towel and tissue products were moved to a new category called Paper Products.
    b. A couple new products were added in the "Fizzy" drink line.
    c. Several products underwent name changes.

12. In the lab folder, replace the original product data file with the "Time 2" data file

a. Rename the data_product.txt file to data_product_time1.txt.
b. Rename the data_product_time2.txt file to data_product.txt.

13. In Visual Studio, rerun the product dimension mapping. Notice the various paths and how many records fall on each. **Screenshot Here**



14. In the above picture we see the following:
    a. There are now 62 products. The two new ones flow straight through as inserts to the Product Dimension
    b. Three rows had Type I updates. Those were just updates to the product dimension – on the right side of the mapping.
    c. Six rows had a category change from Supplies to Paper Products. Those were Type II updates. To complete the Type II updates, the mapping first updates the existing six rows to have a ActiveInd = 0 on the left side of the mapping. Then it inserts new rows into DimProduct that contain the new
15. Look at the DimProduct again in SQL Enterprise Manager. Notice that there are now 68 records in the table. Also notice that 6 records have a ActiveInd = 0, which indicates that another SKU in the table has more current information. **Screenshot Here**

This is the end of the detailed portion of this lab. Please continue working on for ideas on practicing further with SSIS.

VI. On Your Own

Now that you have seen the processing for creating and running an SSIS mapping, you can create additional mappings to load the data for the product, promotion, and time dimensions.

Do the following steps on your own. I have provided the source data necessary to fully load your schema if desired. You may just do the things you can.

1. (5 - Easy) Load the promotion dimension **Screenshot Here**
2. (5- Easy) Load the store dimension **Screenshot Here**
3. (10 - Moderate) Load the time dimension – note that you will have to do some SSIS data conversions using the Derived Column control. **Screenshot Here**
4. (15- Difficult) Load the Fact Retail Sales Transaction table. For this mapping, you will need to use Lookup controls that lookup the key values on the dimensions based on the input fact data. You will also need to handle errors when values are not available as well as deal with data type conversions as in the time dimension mapping. **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.