Module 3 | Lab 2

We are now onto the second lab of the course. This lab focuses on how to create an SSIS ETL package using Execute SQL Tasks, Sequences Containers, and Precedent Constraints.

■ **Note:** This lab is standalone, and can be completed separately from the previous lab.

SCENARIO

A local company is looking for consultants to help with the creation of a Business Intelligence solution. Currently you are one of few consultants they are interested in. To get a better understanding of your skill level with Microsoft's SSIS, they have asked you to create a prototype ETL solution using an SSIS Project.

This prototype is based on an example subset of Microsoft's AdventureWorks demonstration database. The IT team has provided both a source and destination database for you to work with. They have also provided you with a SQL Script file to be used for the SSIS process within your prototype.

LAB OVERVIEW

In this lab, you will create an SSIS project that will perform ETL processing. You will start by reviewing an existing ETL SQL script and then use this script to configure an SSIS package. Before starting this lab, you need to be familiar with the Module 3 | ETL Processing with SSIS content and have followed the instructions in the Setting up the Lab Environment section at the start of this course.

WHAT YOU'LL NEED

- A personal or virtual computer with the SQL Server 2016 (or 2014 or 2012) installed on it
- Permissions to download files on the computer
- Permissions to create a new folders and files on the computer
- Permissions to create databases in the SQL Server instance
- The following SQL database file: http://msftdbprodsamples.codeplex.com/downloads/get/354847 (This is an external link that opens in a new window.)
- Several Course files: **https://github.com/MicrosoftLearning/Implementing-ETL** (This is an external link that opens in a new window.)

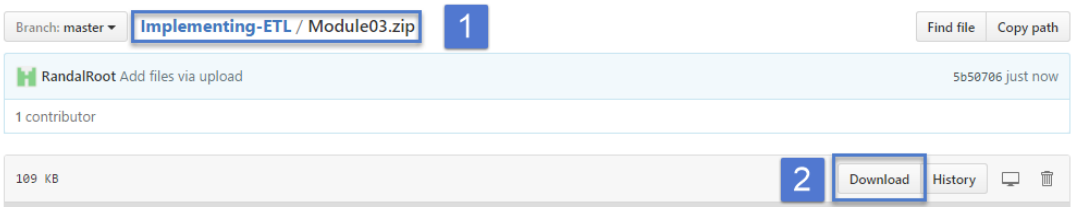
Getting Started

To complete the labs, you must download and extract the lab resources

1. Navigate to **https://github.com/MicrosoftLearning/Implementing-ETL** (This is an external link that opens in a new window.)

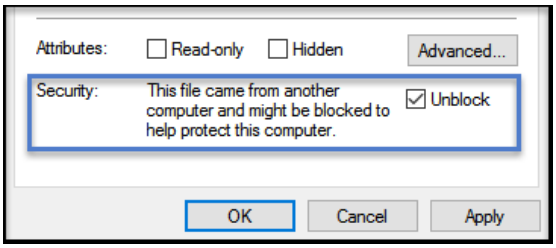
2. Download the course resources in the **Implementing-ETL/Module03.zip** file and save it to your computer.

Lab Figure 3-1 Downloading the Module Files



3. Right-click on **Module03.zip** within the downloaded file, and select **Properties** from the context menu.

4. Some operating systems require the file to be unblocked. In the dialog window, check **Unblock**, then click **OK** to close.



5. Extract the **Module03.zip** file to **C:_ETLwithSSIS**.

6. Navigate to the **C:_ETLwithSSIS** folder and verify that folder contains the **Module03** sub-folder.

■ **Important:** Some lab scripts use absolute file paths, please extract and use the lab resources directly from the C:_ETLwithSSIS path. This lab requires a lot of SQL programming; therefore, the answer code is supplied with this lab in case you need help.

■ **Note:** Each lab in this course is standalone, and can be completed individually.

Exercise 1 | Attaching the Source Database

(If you have already completed this within the previous lab, you can skip to Exercise 2.)

1. Download and **SQL Server AdventureWorks LT 2012 database file** from the product samples page on **Codeplex**. The database file name is AdventureWorksLT2012_Data.mdf. This file can be found at this URL: http://msftdbprodsamples.codeplex.com/downloads/get/354847 (This is an external link that opens in a new window.)

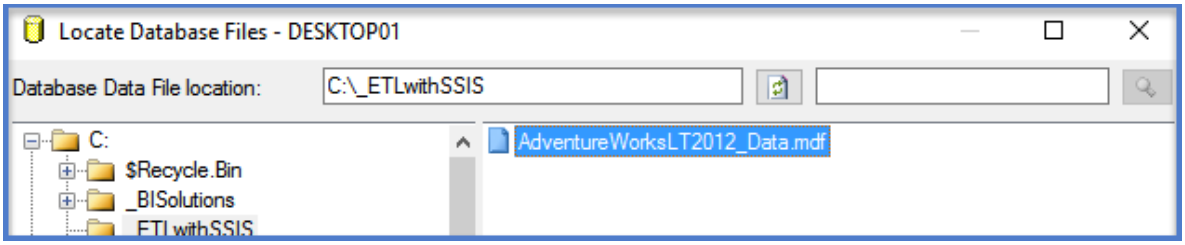
2. Copy the **AdventureWorksLT2012_Data.mdf** file to the **C:_ETLWithSSIS** folder.

3. Start **SQL Server Management Studio**, using the Run as Administrator option, and connect to the Database Engine instance.

4. Right-click **Databases** icon in the **Object Explorer** Tree View window, then click Attach in the Context Menu.

5. In the **Attach Database Dialog** window, click the Add button.

6. Select the **AdventureWorksLT2012_Data.mdf** database file and click OK. If the file is not listed, check the folder to be sure the file is there.

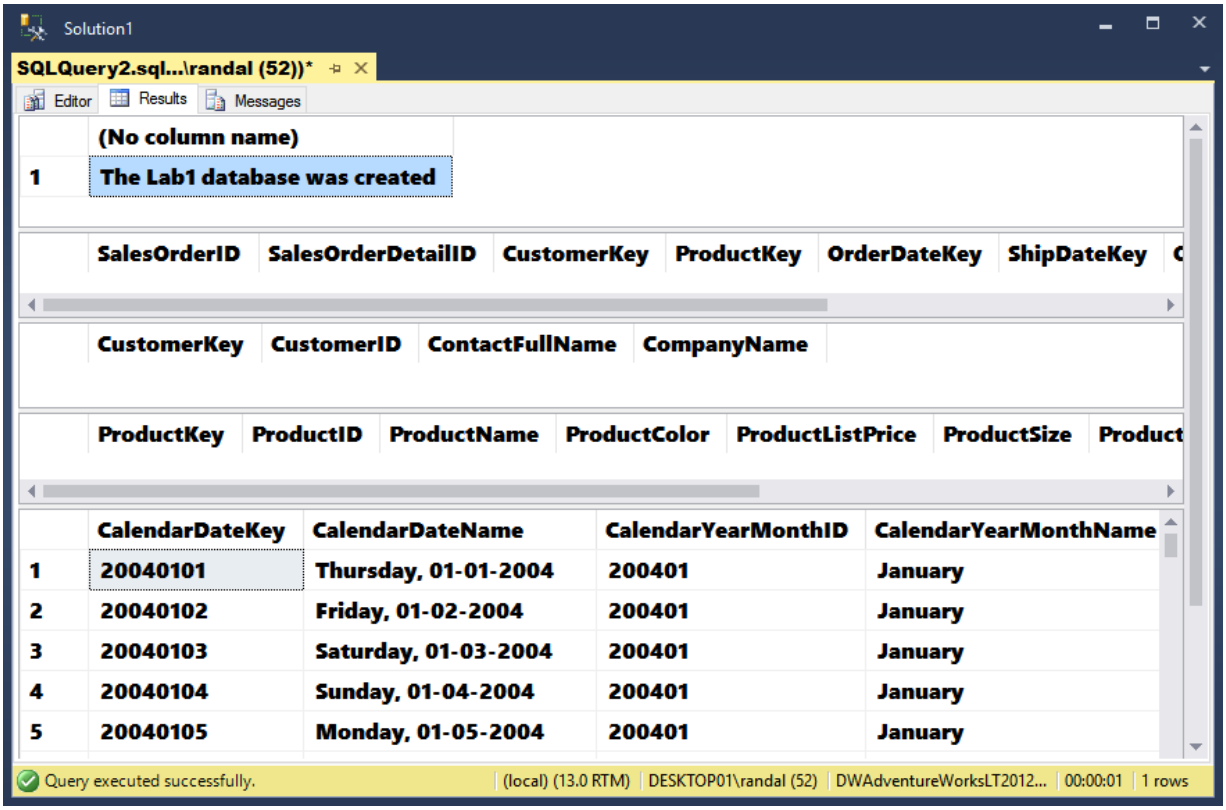


7. In database details, click the **Remove** button to remove the **Log file entry**. The setup program assumes you have a log file, but there is no log file in the sample. A new log file will be created automatically when you attach the database.

8. Click **OK** to attach just the primary database file.

Exercise 2 | Creating the Destination Database

1. Start **SQL Server Management Studio** and connect to the Database Engine instance.
2. Use the **File > Open > File** menu to open the Open File dialog window.
3. Locate and open the **C:_ETLwithSSIS\Module03\Labs\Create the DWAdventureWorksLTLab02 database.sql** file.
4. Use the **[! Execute**] button on the toolbar to run all of the code in the file.
5. Verify that the results of running the script look like the following **image**.



6. Refresh the **Object Explorer** Tree view and verify that the database was created.

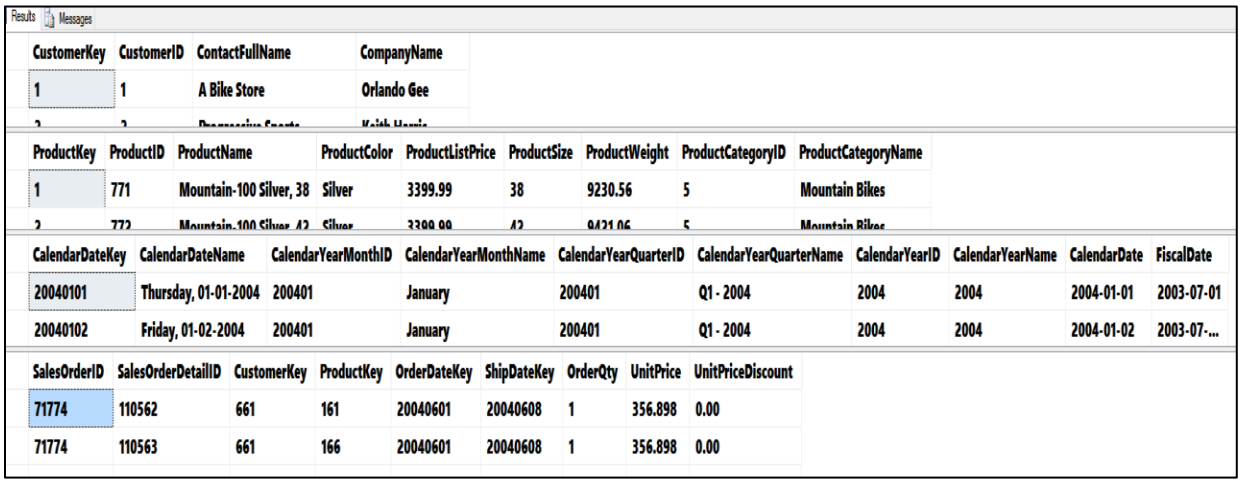
Exercise 3 | Review the Existing ETL SQL Script

1. Start **SQL Server Management Studio** and connect to the Database Engine instance.

2. Use the **File > Open > File** Menu to open the **C:_ETLwithSSIS\Module03\Labs\ETL Code for the DWAdventureWorksLTLab02 database.sql** file.

3. Review the **SQL code**, noting how the stored procedures will be called to fill the dimension and fact tables.

4. Run in entire **script** and verify that all of the tables are filled with data as shown here:



CustomerKey	CustomerID	ContactFullName	CompanyName
1	1	A Bike Store	Orlando Gee

ProductKey	ProductID	ProductName	ProductColor	ProductListPrice	ProductSize	ProductWeight	ProductCategoryID	ProductCategoryName
1	771	Mountain-100 Silver, 38	Silver	3399.99	38	9230.56	5	Mountain Bikes

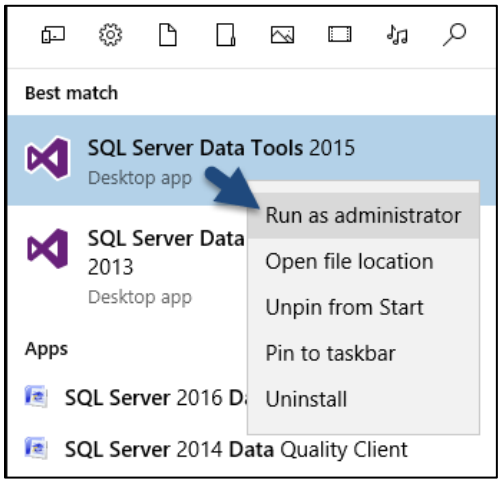
CalendarDateKey	CalendarDateName	CalendarYearMonthID	CalendarYearMonthName	CalendarYearQuarterID	CalendarYearQuarterName	CalendarYearID	CalendarYearName	CalendarDate	FiscalDate
20040101	Thursday, 01-01-2004	200401	January	200401	Q1 - 2004	2004	2004	2004-01-01	2003-07-01
20040102	Friday, 01-02-2004	200401	January	200401	Q1 - 2004	2004	2004	2004-01-02	2003-07-...

SalesOrderID	SalesOrderDetailID	CustomerKey	ProductKey	OrderDateKey	ShipDateKey	OrderQty	UnitPrice	UnitPriceDiscount
71774	110562	661	161	20040601	20040608	1	356.898	0.00
71774	110563	661	166	20040601	20040608	1	356.898	0.00

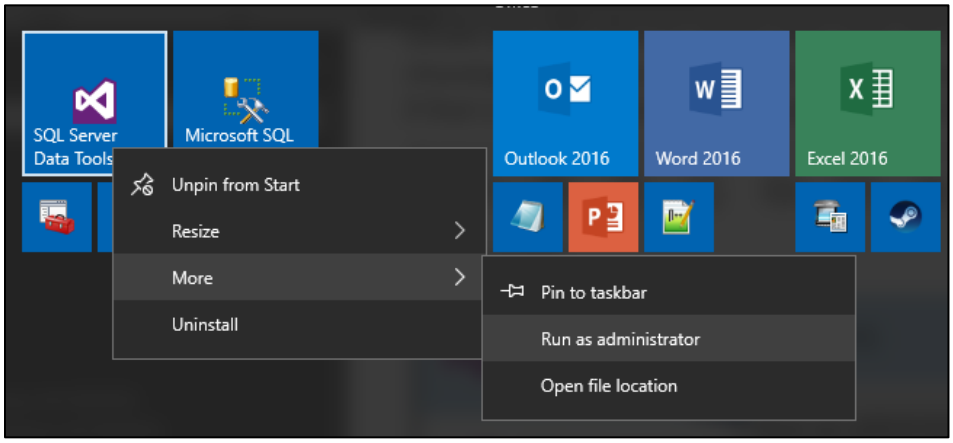
The **Message tab** should indicate the following rows were inserted and selected as shown in this **image**:

Exercise 4 | Create an SSIS Project

1. Using the **Window menu** search feature, locate and open the **SQL Server Data Tools 2015** desktop application, by right-clicking its icon and choosing the **Run as administrator** option.



Or if using the **Window Tile** UI option, choosing the **More > Run as administrator** option.

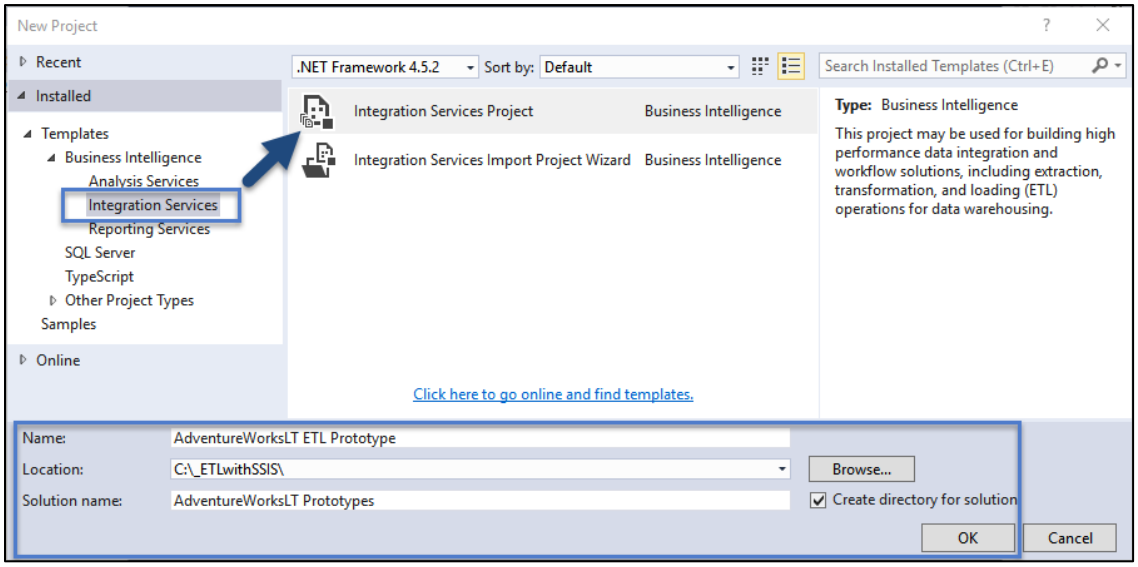


■ **NOTE:** You can use an earlier versions of the SQL Server Data Tools, if that is all you currently have installed, but the images will look a bit different.

2. When **Visual Studio** opens, create a new Integration Services project using **the File > New > Project** to menu item.

3. In the **New Project dialog** box, expand the Business Intelligence node under Installed Templates, and select **Integration Services Project** in the Templates pane.

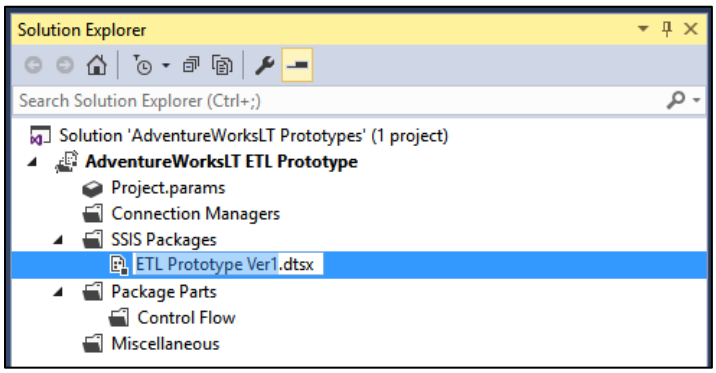
4. In the Name box, change the default name to **AdventureWorksLT ETL Prototype**. Change the default location to the **C:_ETLwithSSIS\Module03** folder. Change the Solution name to **AdventureWorksLT Prototypes**. Then click OK to create the new SSIS project.



By default, an empty package, titled **Package.dtsx**, will be created and added to your project under SSIS Packages.

Exercise 5 | Configure an SSIS Package

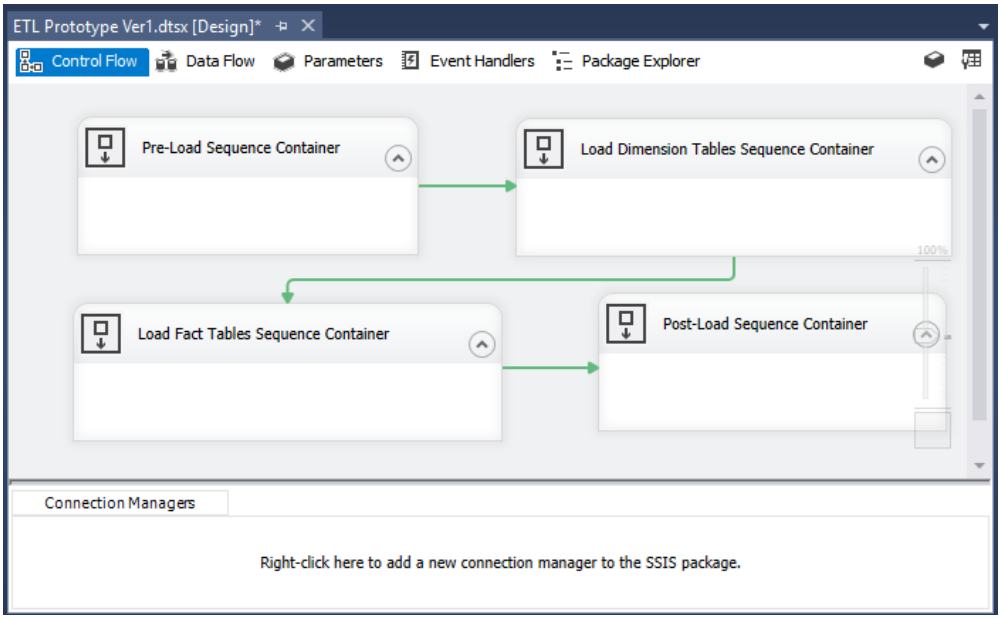
1. In Solution Explorer window, right-click **Package.dtsx**, click Rename, and rename the default package to **ETL Prototype Ver1.dtsx**.



2. Add four **Sequence Containers** to the packages designer surface, rename each of them using the following list:

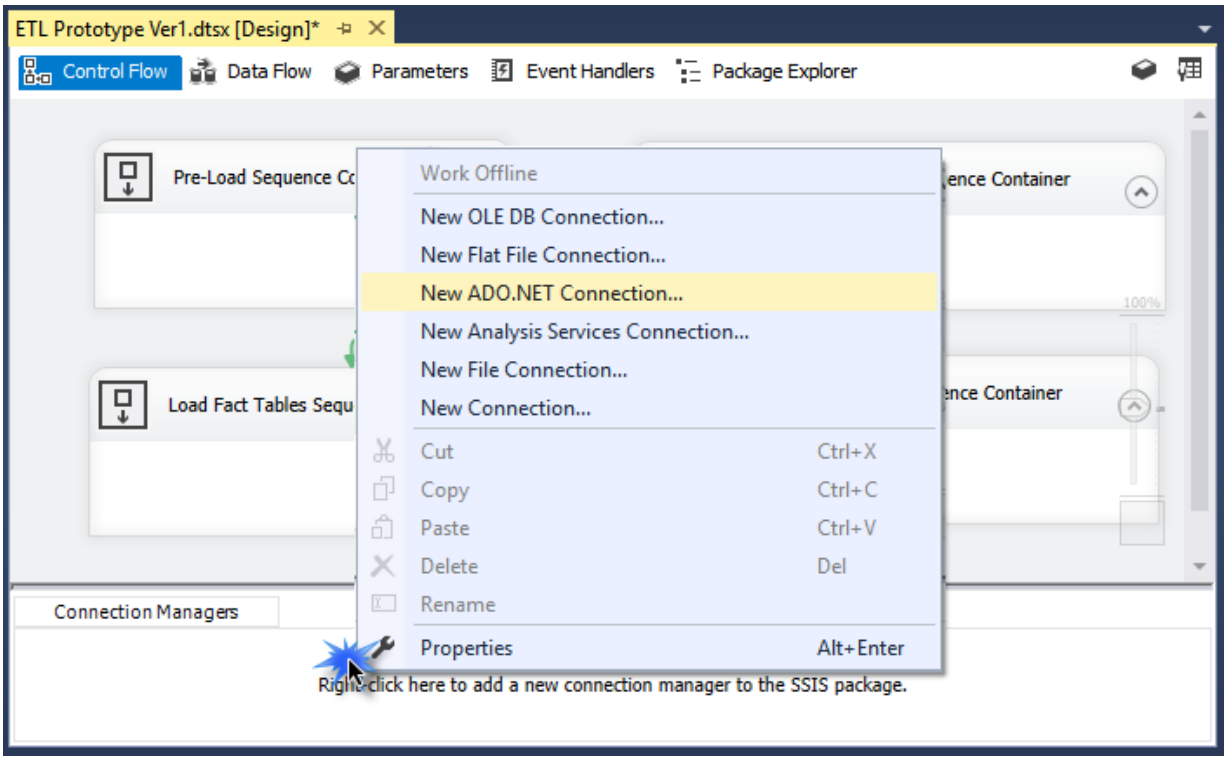
   - **Pre-Load Sequence Container**
   - **Load Dimension Tables Sequence Container**
   - **Load Fact Tables Sequence Container**
   - **Post-Load Sequence Container**

3. Link them together using **Precedent Constraints** in the same order as the named list. Your package should look like the following image when this step is completed.



4. Add a connection to the **DWAdventureWorksLT2012Lab02** database using an ADO.NET Connection.

    a. Right-click anywhere in the **Connection Managers area**, and then click **New ADO.NET Connection**.



    b. In the Configure **ADO.NET Connection Manager dialog** box, click **New**.

    c. For Server name, enter ***localhost***.

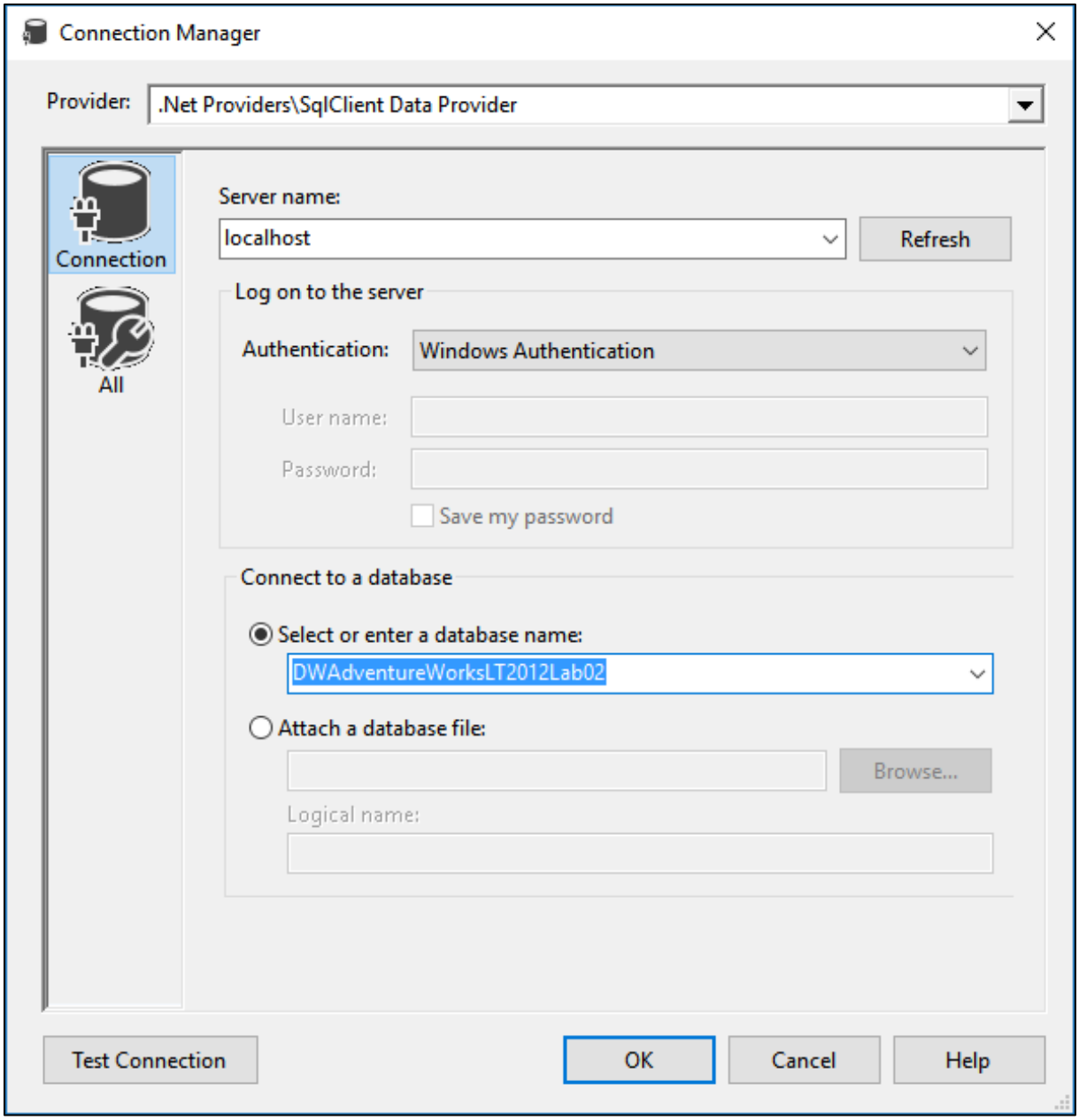
■ **NOTE:** When you specify localhost as the server name, the connection manager connects to the default instance of SQL Server on the local computer. To use a remote instance of SQL Server, replace localhost with the name of the server to which you want to connect. To use a named instance of SQL Server, add a back-slash and the name of the instance (localhost\MyNamedInstance).

    d. In the "Log on to the server" group, verify that **Use Windows Authentication** is selected.

    e. In the Connect to a database group, in the Select or enter a database name box, type or select **DWAdventureWorksLT2012Lab02**.

    f. Click **Test Connection** to verify that the connection settings you have specified are valid.

    g. Click **OK**.



h. Click **OK**.

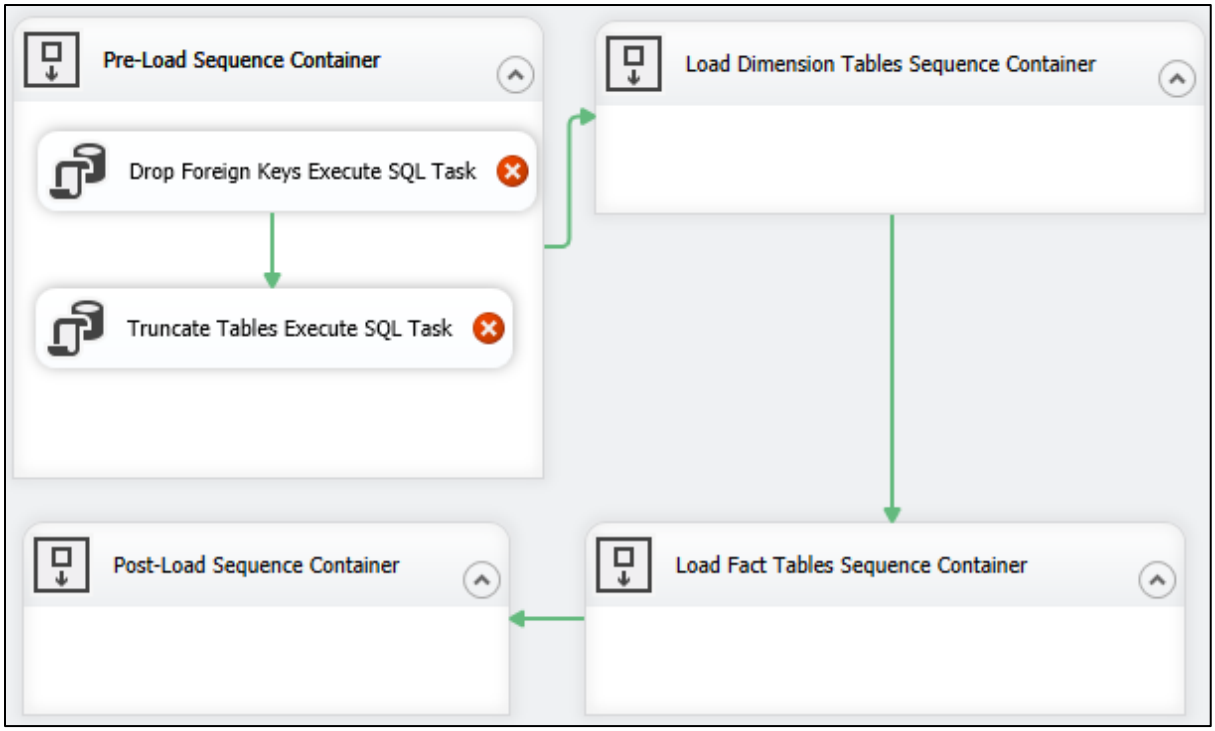
i. In the Data Connections pane of the Configure **ADO.NET Connection Manager dialog** box, verify that localhost. DWAdventureWorksLT2012Lab02 is selected.

j. Click **OK**, again to create the SSIS connection.

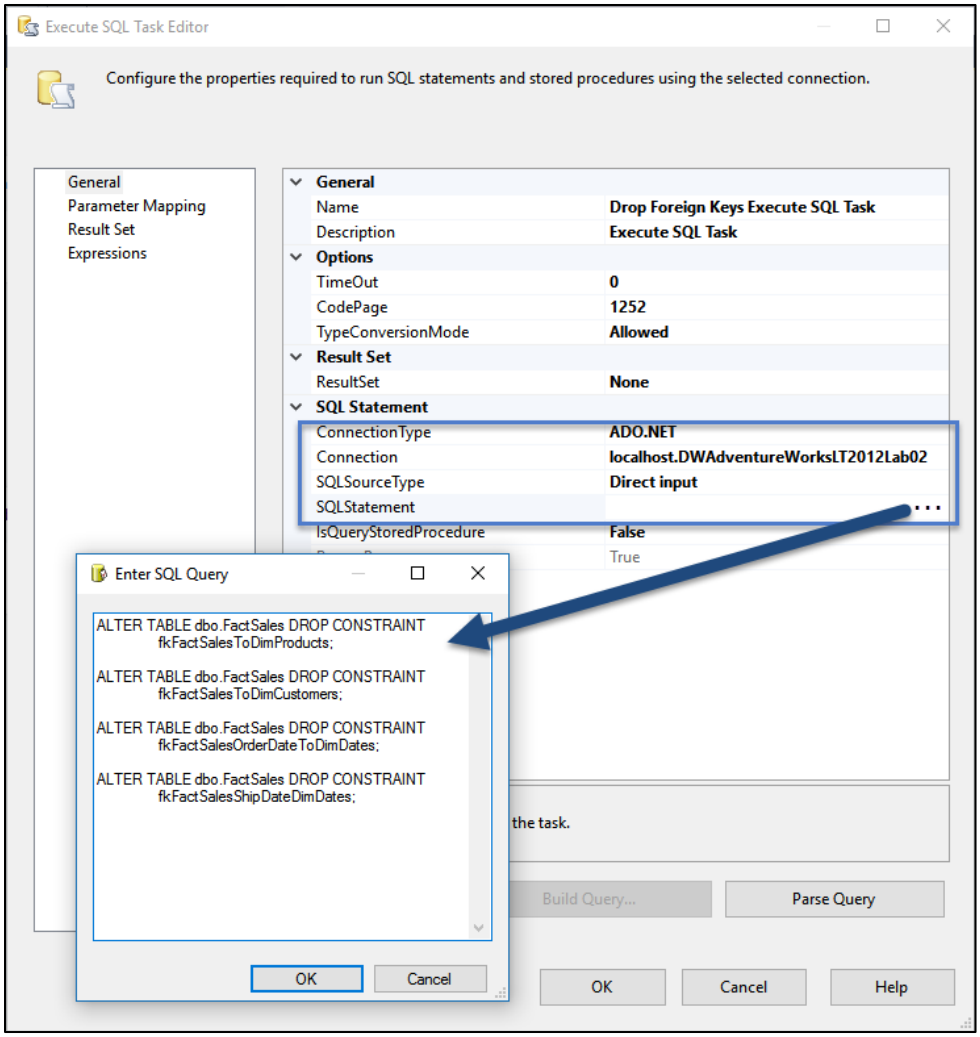
Exercise 6 | Configure the Pre-Load Sequence Container

1. From the **SSIS Toolbox**, drag an Execute SQL Task into the **Pre-Load Sequence Container** on Control Flow design surface.
2. Right-click **Execute SQL Task** and rename it to **Drop Foreign Key Constraints Execute SQL Task**.
3. From the **SSIS Toolbox**, drag another Execute SQL Task into the **Pre-Load Sequence Container** on Control Flow design surface.
4. Right-click **Execute SQL Task** and rename it to **Truncate Tables Execute SQL Task**.
5. Click the **Drop Foreign Key Constraints Execute SQL Task** and drag the green arrow that appears onto the newly added **Truncate Tables Execute SQL Task** to connect the two components.

Completing these steps will make the SSIS Package look *similar* to the following image:



6. Start **SQL Server Management Studio** and connect to the Database Engine instance.
7. Use the **File** > **Open** > **File Menu** to open the **C:_ETLwithSSIS\Module03\Labs\ETL Code for the DWAdventureWorksLTLab02 database.sql** file.
8. Locate the **SQL code** that drops all of the foreign key constraints and copy it.
9. Right-click the **Drop Foreign Key Constraints Execute SQL Task** and choose Edit from its context menu.
10. Set the ***ConnectionType* property** to the ADO.NET, the ***Connection* property** to the SSIS ADO.NET connection you created earlier, and paste the SQL code you copied into the ***SQLStatement* property**.



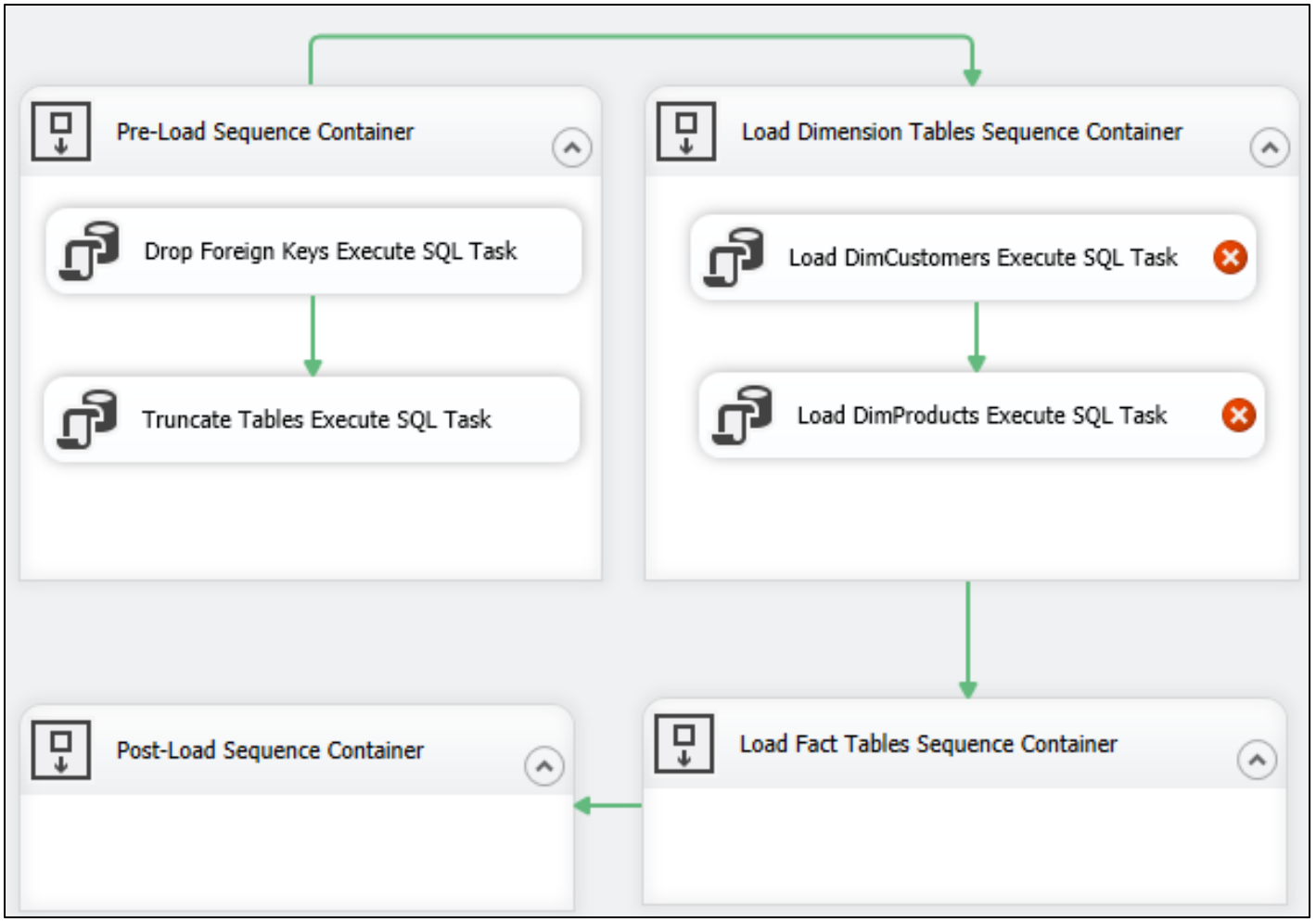
11. Close the **Execute SQL Task Editor**.
12. Locate the **SQL code** that truncates all of the tables and copy it.
13. Right-click the **Truncate Tables Execute SQL Task** and choose Edit from its context menu.
14. Set the ***ConnectionType* property** to the ADO.NET, the ***Connection* property** to the SSIS ADO.NET connection you created earlier, and paste the SQL code you copied into the ***SQLStatement* property**.
15. Right-click the **Pre-Load Sequence Container**, and select **Execute Container** from the context.
16. Verify that all the tasks run successfully or troubleshoot why not. Remember, if you need to change settings in the task, you must first stop the **debugging engine**.

■ **NOTE:** If you keep getting errors, try resetting the database to its normal empty state by running the SQL code in the **C:_ETLwithSSIS\Module03\Labs\Create the DWAdventureWorksLTLab02 database.sql** file. This code re-creates the database and makes it ready for the ETL process.

Exercise 7 | Configure the Load Dimension Tables Sequence Container

1. From the **SSIS Toolbox**, drag an Execute SQL Task into the **Load Dimension Tables Sequence Container** on Control Flow design surface.
2. Right-click Execute SQL Task and rename it to **Load DimCustomers Execute SQL Task**.
3. From the **SSIS Toolbox**, drag another Execute SQL Task into the **Load Dimension Tables Sequence Container** on Control Flow design surface.
4. Right-click Execute SQL Task and rename it to **Load DimProducts Execute SQL Task**.
5. Click the **Drop Foreign Key Constraints Execute SQL Task** and drag the green arrow that appears onto the newly added **Truncate Tables Execute SQL Task** to connect the two components.

Completing these steps will make the SSIS Package look *similar* to this:



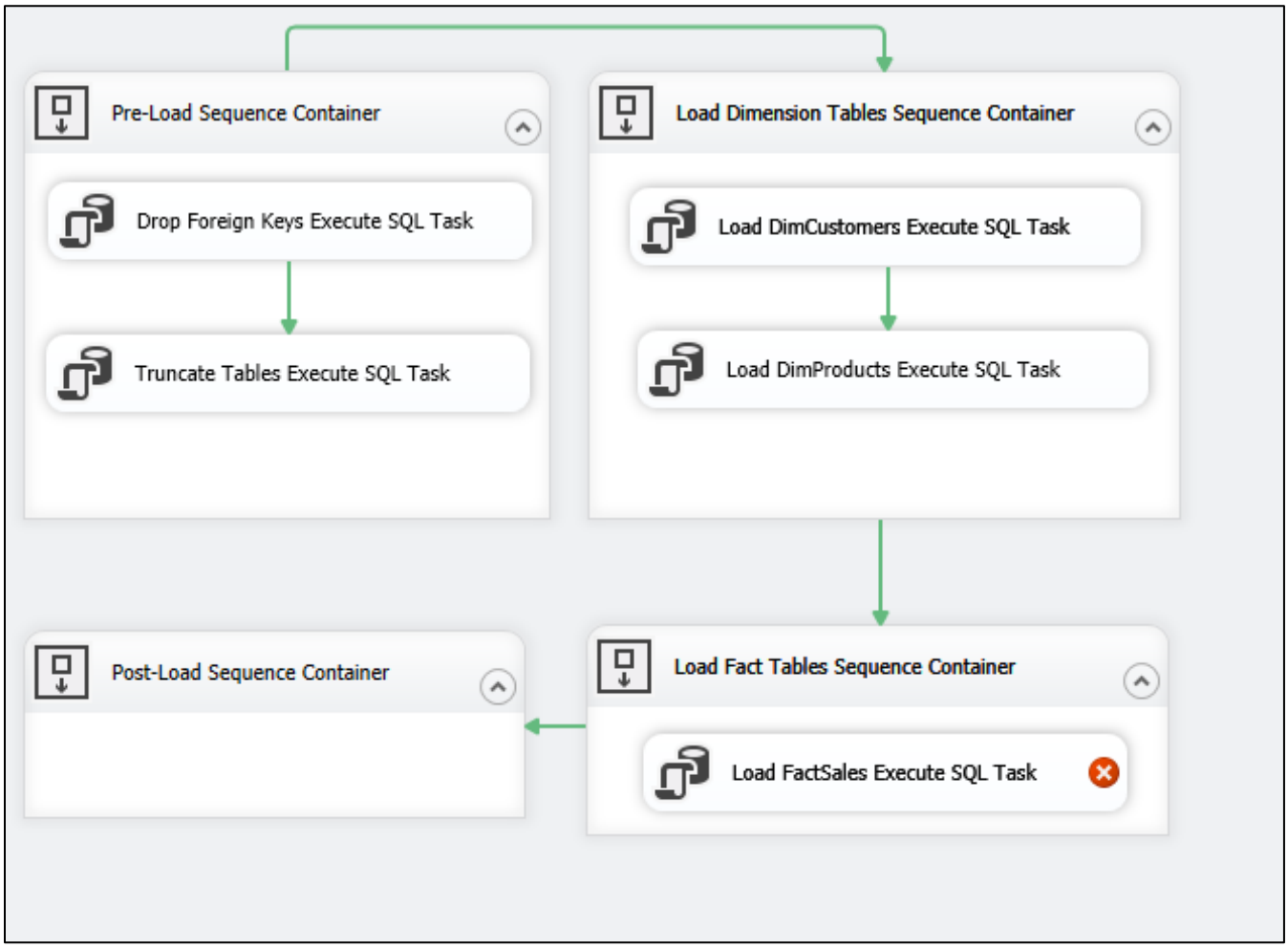
6. Right-click the **Load DimCustomers Execute SQL Task** and choose **Edit** from its context menu.

7. Set the ***ConnectionType*** **property** to the ADO.NET, the ***Connection*** **property** to the SSIS ADO.NET connection you created earlier, the ***IsQueryStoreProcedure*** **property** to True, and the ***SQLStatement*** **property** to **pETLFillDimCustomers**.
8. Close the **Execute SQL Task Editor**.
9. Right-click the **Fill DimProducts Execute SQL Task** and choose **Edit** from its context menu.
10. Set the ***ConnectionType*** **property** to the ADO.NET, the ***Connection*** **property** to the SSIS ADO.NET connection you created earlier, the ***IsQueryStoreProcedure*** **property** to True, and the ***SQLStatement*** **property** to **pETLFillDimProducts**.
11. Close the **Execute SQL Task Editor.**
12. Right-click the **Fill Dimension Tables Sequence Container,** and select **Execute Container** from the context.
13. Verify that all the tasks run successfully or troubleshoot why not. Remember, if you need to change settings in the task, you must first stop the **debugging engine**.

Exercise 8 | Configure the Load Fact Tables Sequence Container

1. From the **SSIS Toolbox**, drag an Execute SQL Task into the **Load Fact Tables Sequence Container** on Control Flow design surface.
2. Right-click Execute SQL Task and rename it to **Load FactSales Execute SQL Task**.

Completing these steps will make the SSIS Package look *similar* to this:

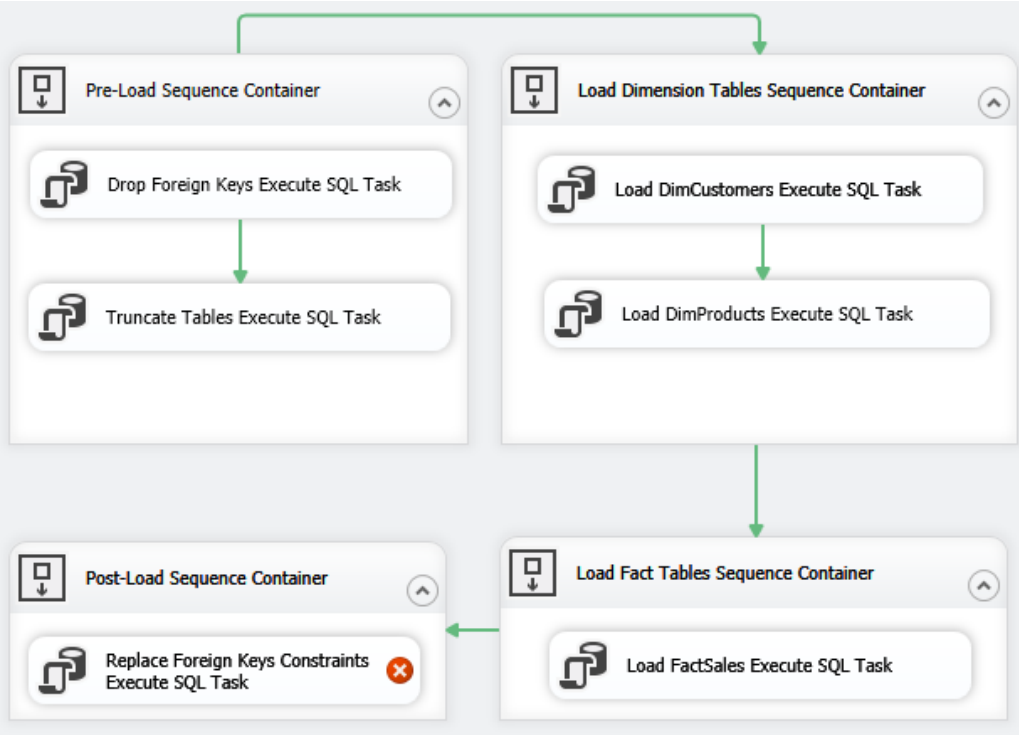


3. Right-click the **Load FactSales Execute SQL Task** and choose **Edit** from its context menu.
4. Set the ***ConnectionType* property** to the ADO.NET, the ***Connection* property** to the SSIS ADO.NET connection you created earlier, the ***IsQueryStoreProcedure* property** to True, and the ***SQLStatement* property** to **pETLFillFactSales**.
5. Close the Execute SQL Task Editor.
6. Right-click the **Load Fact Tables Sequence Container**, and select **Execute Container** from the context.
7. Verify that all the tasks run successfully or troubleshoot why not. Remember, if you need to change settings in the task, you must first stop the **debugging engine**.

Exercise 9 | Configure the Post-Load Sequence Container

1. From the SSIS Toolbox, drag an Execute SQL Task into the **Post-Load Sequence Container** on Control Flow design surface.
2. Right-click Execute SQL Task and rename it to **Replace Foreign Keys Constraints Execute SQL Task**.

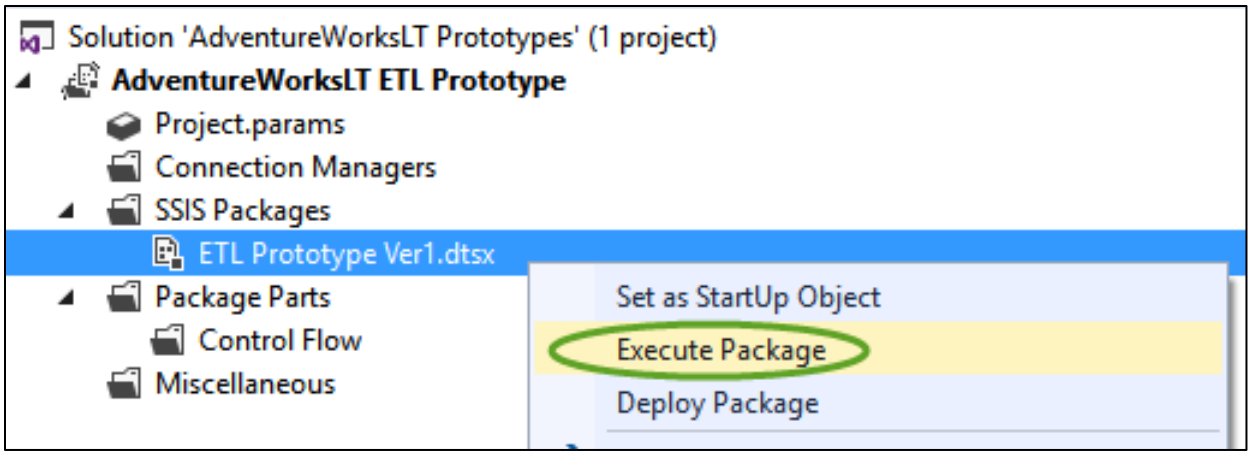
Completing these steps will make the SSIS Package look *similar* to this:



3. Start **SQL Server Management Studio** and connect to the Database Engine instance.
4. Use the **File** > **Open** > **File** Menu to open the **C:_ETLwithSSIS\Module03\Labs\ETL Code for the DWAdventureWorksLTLab02 database.sql** file.
5. Locate the SQL code that re-creates all of the foreign key constraints and copy it.
6. Right-click the **Replace Foreign Key Constraints Execute SQL Task** and choose **Edit** from its context menu.
7. Set the ***ConnectionType* property** to the ADO.NET, the ***Connection* property** to the SSIS ADO.NET connection you created earlier, and paste the SQL code you copied into the ***SQLStatement* property**.
8. Close the **Execute SQL Task Editor**.
9. Right-click the **Post-Load Sequence Container**, and select **Execute Container** from the context.
10. Verify that all the tasks have run successfully or troubleshoot any issues. Remember, if you need to change settings in the task, you must first stop the **debugging engine.**

Exercise 10 | Executing the Entire SSIS Package

1. Right-click the **ETL Prototype Ver1.dtsx** package file and select **Execute Package** from the context menu.



2. Verify that all the tasks have run successfully or troubleshoot any issues. Remember, if you need to change settings in the task, you must first stop the **debugging engine**.

■ **NOTE:** If you keep getting errors, try resetting the database to its normal empty state by running the SQL code in the **C:_ETLwithSSIS\Module03\Labs\Create the DWAdventureWorksLTLab02 database.sql** file. This code re-creates the database and makes it ready for the ETL process.