

BDAMD

EXERCISE 2 TUTORIAL

Paulo Oliveira
DEI-ISEP

Start Visual Studio and then start by creating a new project for this tutorial.

What would you like to do?

Open recent

Today



Analysis Services Tutorial.sln

27/11/2023 16:16

C:\Temp\ARMDD\Analysis Services Tutorial

This month



Duplicates Detection Example.sln

17/11/2023 16:13

C:\Temp\ARMDD\Duplicates Detection Example



Data Mart.sln

17/11/2023 12:08

C:\Temp\ARMDD\Data Mart



SSIS ETL Tutorial.sln

17/11/2023 11:43

C:\Temp\PL2\SSIS ETL Tutorial

Older



Create CSV Date Dimension.sln

23/10/2023 11:06

C:\Temp\ARMDD\Create CSV Date Dimension



Create Date Dimension.sln

23/10/2023 10:32

C:\Temp\ARMDD\Create Date Dimension



16/10/2023 14:55

Get started



Clone a repository

Get code from an online repository like GitHub or Azure DevOps



Open a project or solution

Open a local Visual Studio project or .sln file



Open a local folder

Navigate and edit code within any folder



Create a new project

Choose a project template with code scaffolding to get started

Create a new project

Recent project templates

- Integration Services Project
- Analysis Services Multidimensional and Data Mining Project
- Blank Solution

Search for templates (Alt+S)

All languages

All platforms

All project types



Analysis Services Multidimensional and Data Mining Project
An Analysis Services project for creating multidimensional and data mining models.

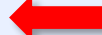


Import from Server (Multidimensional and Data Mining)
Creates a multidimensional and data mining project by extracting the metadata from an existing multidimensional and data mining database on an Analysis Services server.



Integration Services Project (Azure-Enabled)
This project may be used for building high performance data integration and workflow solutions that can also be run/debugged on SSIS Platform-as-a-Service (PaaS) in Azure Data Factory.



Integration Services Project 
This project may be used for building high performance data integration and workflow solutions that can be run on SSIS catalog, including extraction, transformation, and loading (ETL) operations for data warehousing.



Integration Services Import Project Wizard
A wizard that assists you in creating a new Integration Services (SSIS) project that is based on an existing one. Import from a project deployment file (.ispac extension) or from an Integration Services catalog on an instance of SQL Server.



Analysis Services Tabular Project

Next

Select *Integration Services Project* as the template for the project.

Configure your new project

Integration Services Project

Project name

Data Mart

Location

C:\Temp\

Solution name ⓘ

Data Mart

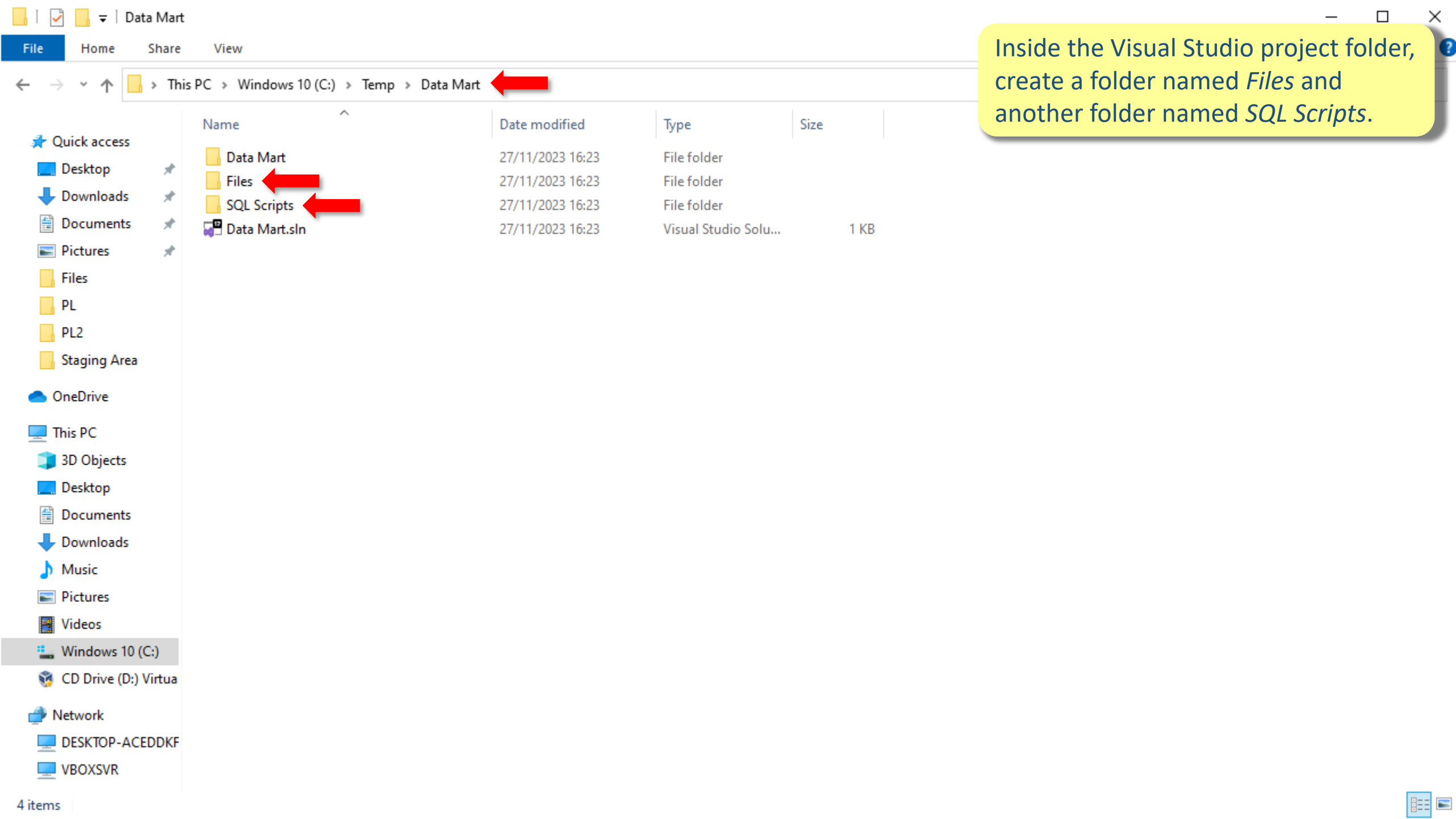
☐ Place solution and project in the same directory

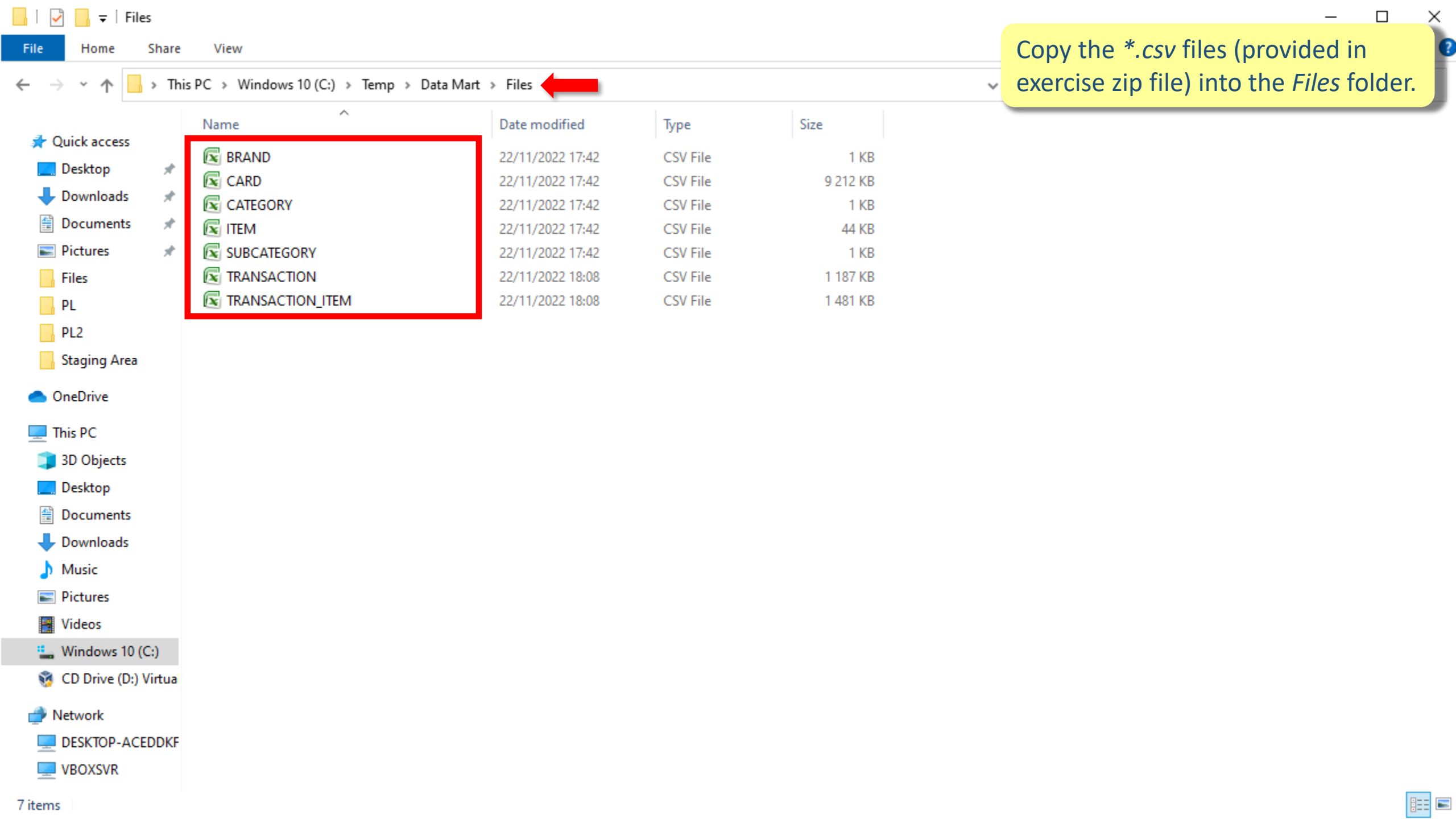
Project will be created in "C:\Temp\Data Mart\Data Mart"

Name the project as *Data Mart* and select a location for it that can be anywhere you want. In this tutorial is used the location *C:\Temp*.

Back

Create





Copy the *.csv files (provided in exercise zip file) into the *Files* folder.

Name	Date modified	Type	Size
BRAND	22/11/2022 17:42	CSV File	1 KB
CARD	22/11/2022 17:42	CSV File	9 212 KB
CATEGORY	22/11/2022 17:42	CSV File	1 KB
ITEM	22/11/2022 17:42	CSV File	44 KB
SUBCATEGORY	22/11/2022 17:42	CSV File	1 KB
TRANSACTION	22/11/2022 18:08	CSV File	1 187 KB
TRANSACTION_ITEM	22/11/2022 18:08	CSV File	1 481 KB

← → ↕ ↑ > This PC > Windows 10 (C:) > Temp > Data Mart > SQL Scripts

Inside the *SQL Scripts* folder create two folders: *Data Mart* and *Staging Area*.

	Name	Date modified	Type	Size
★ Quick access				
Desktop				
Downloads				
Documents				
Pictures				
Files				
PL				
PL2				
Staging Area				
OneDrive				
This PC				
3D Objects				
Desktop				
Documents				
Downloads				
Music				
Pictures				
Videos				
Windows 10 (C:)				
CD Drive (D:) Virtua				
Network				
DESKTOP-ACEDDKF				
VBOXSVR				

Name

Date modified

Type

Size

Data Mart

27/11/2023 16:28

File folder

Staging Area

27/11/2023 16:28

File folder

Staging Area

File Home Share View

← → ↕ ⬆ ⬇ This PC > Windows 10 (C:) > Temp > Data Mart > SQL Scripts > Staging Area

Quick access

- Desktop
- Downloads
- Documents
- Pictures
- Files
- PL
- PL2
- Staging Area

OneDrive

This PC

- 3D Objects
- Desktop
- Documents
- Downloads
- Music
- Pictures
- Videos

Windows 10 (C:)

- CD Drive (D:) Virtua

Network

- DESKTOP-ACEDDKF
- VBOXSVR

7 items

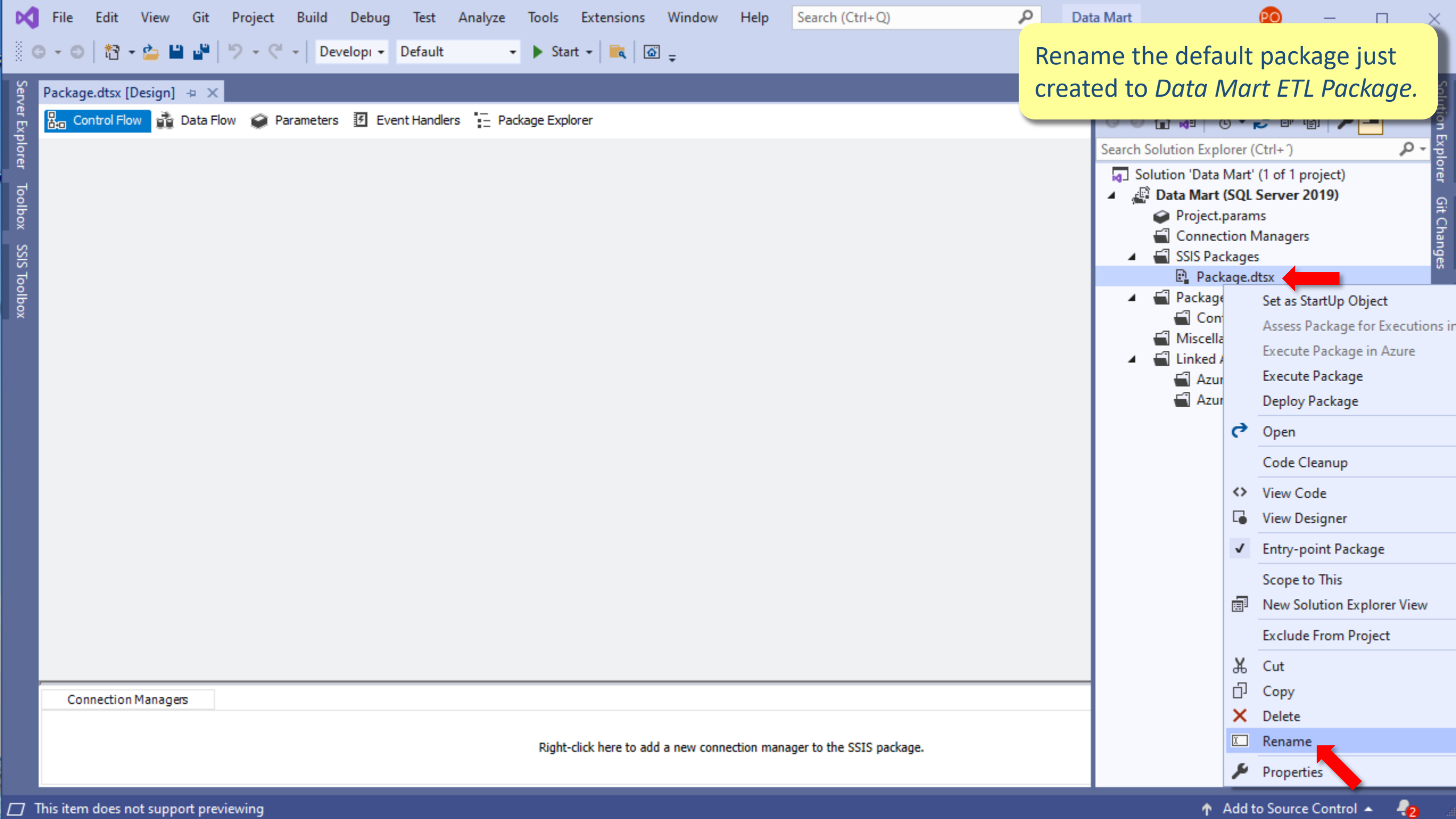
Name	Date modified	Type	Size
CreateStageTableBrand	16/12/2021 17:23	Microsoft SQL Ser...	1 KB
CreateStageTableCard	16/12/2021 17:22	Microsoft SQL Ser...	1 KB
CreateStageTableCategory	16/12/2021 17:22	Microsoft SQL Ser...	1 KB
CreateStageTableItem	16/12/2021 17:21	Microsoft SQL Ser...	1 KB
CreateStageTableSubCategory	16/12/2021 17:21	Microsoft SQL Ser...	1 KB
CreateStageTableTransaction	16/12/2021 17:21	Microsoft SQL Ser...	1 KB
CreateStageTableTransactionItem	16/12/2021 17:20	Microsoft SQL Ser...	1 KB

Copy the *.sql files (provided in the exercise zip file for creating staging area tables) into the *Staging Area* folder.

← → ↕ ↑ > This PC > Windows 10 (C:) > Temp > Data Mart > SQL Scripts > Data Mart

Name	Date modified	Type	Size
CreateDimCard	27/11/2023 16:30	Microsoft SQL Ser...	1 KB
CreateDimDate	14/12/2021 12:18	Microsoft SQL Ser...	1 KB
CreateDimItem	27/11/2023 16:30	Microsoft SQL Ser...	1 KB
CreateFactSales	14/12/2021 12:21	Microsoft SQL Ser...	1 KB

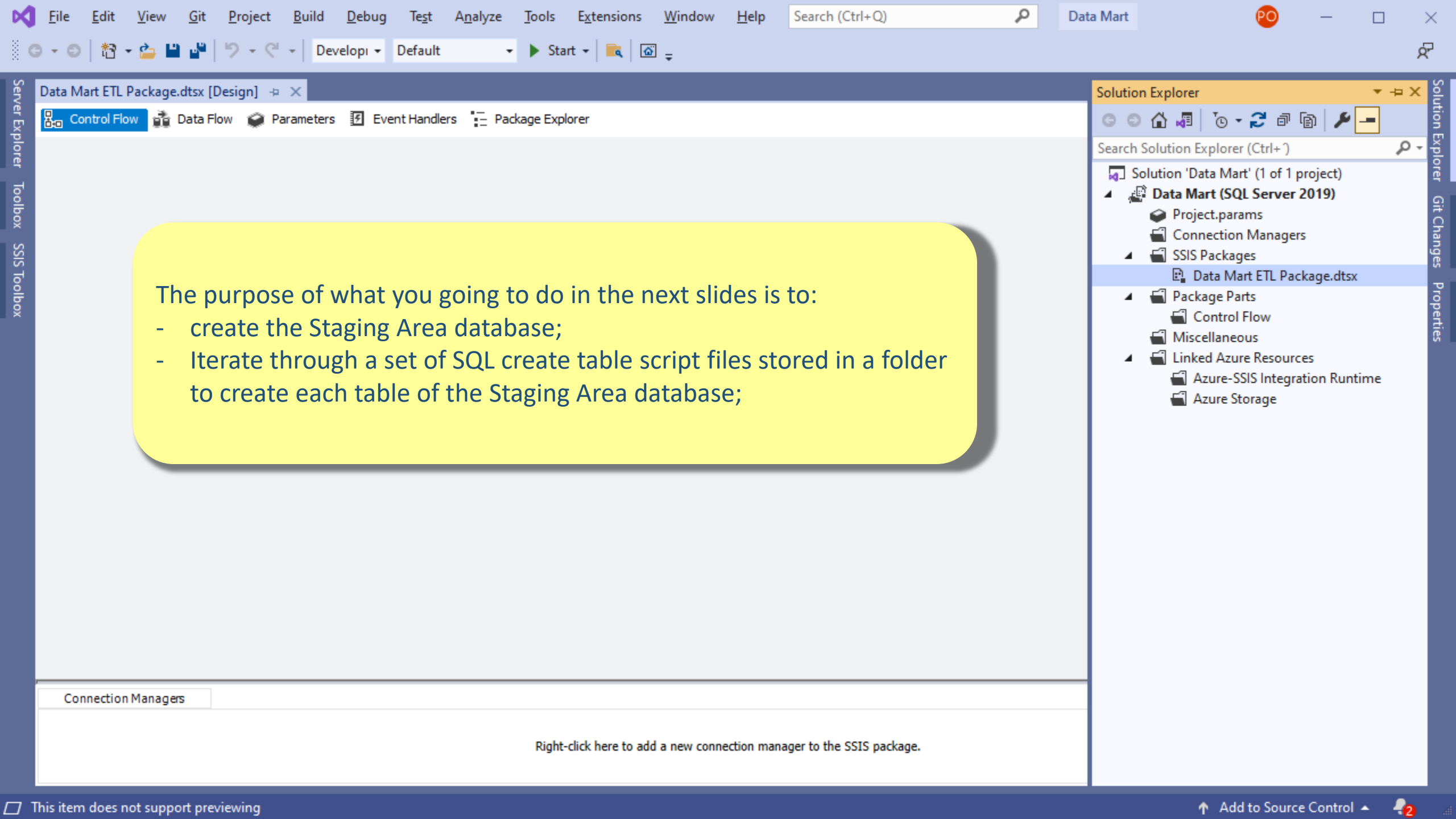
Copy the *.sql files (provided in the exercise zip file for creating data mart tables) into the *Data Mart* folder.



Rename the default package just created to *Data Mart ETL Package*.

- Set as StartUp Object
- Assess Package for Executions in
- Execute Package in Azure
- Execute Package
- Deploy Package
- Open
- Code Cleanup
- View Code
- View Designer
- ✓ Entry-point Package
- Scope to This
- New Solution Explorer View
- Exclude From Project
- Cut
- Copy
- Delete
- ✓ Rename
- Properties

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



The purpose of what you going to do in the next slides is to:

- create the Staging Area database;
- Iterate through a set of SQL create table script files stored in a folder to create each table of the Staging Area database;

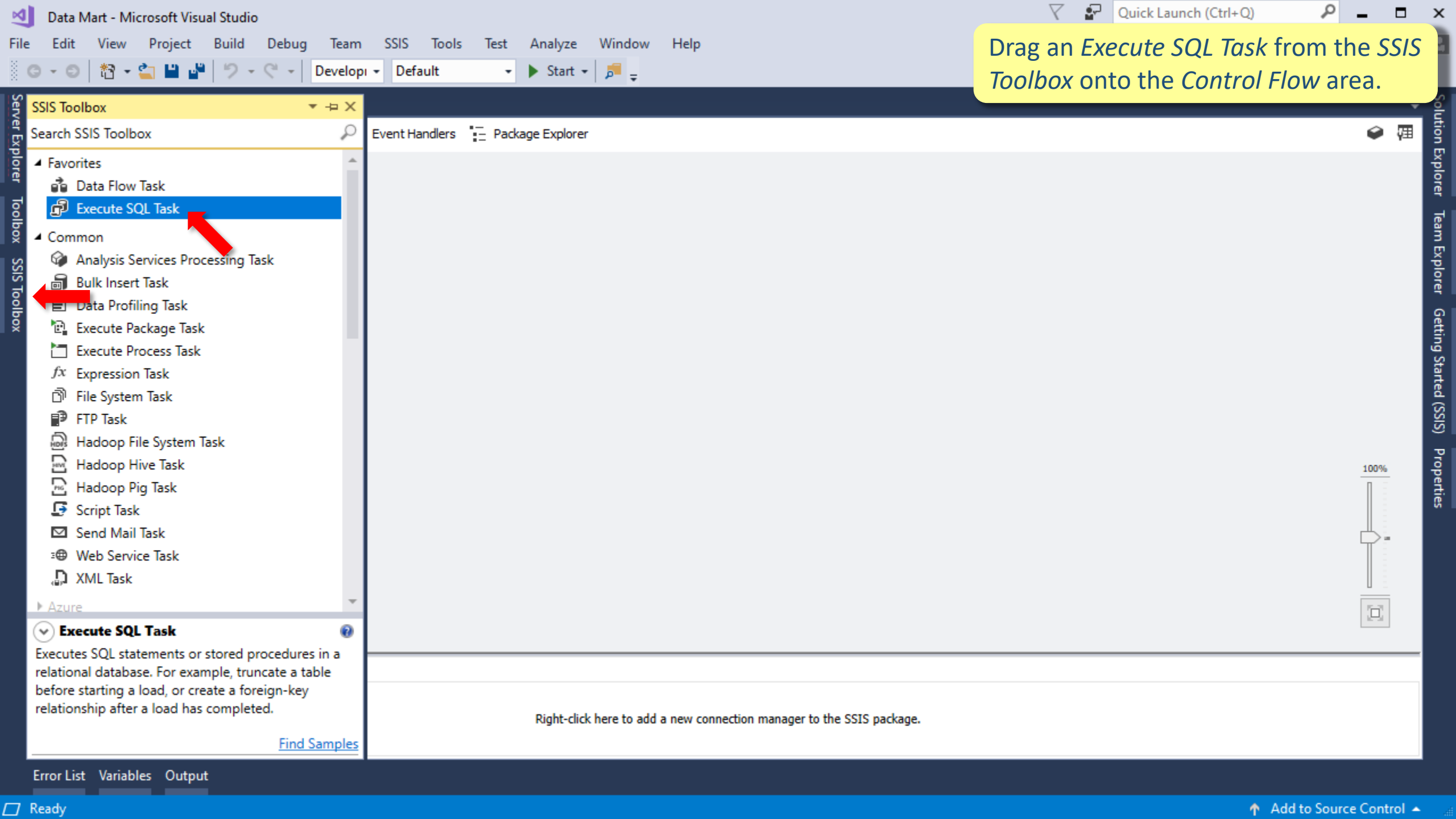
Solution Explorer

Search Solution Explorer (Ctrl+)

- Solution 'Data Mart' (1 of 1 project)
 - Data Mart (SQL Server 2019)
 - Project.params
 - Connection Managers
 - SSIS Packages
 - Data Mart ETL Package.dtsx
 - Package Parts
 - Control Flow
 - Miscellaneous
 - Linked Azure Resources
 - Azure-SSIS Integration Runtime
 - Azure Storage

Connection Managers

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



Drag an *Execute SQL Task* from the *SSIS Toolbox* onto the *Control Flow* area.

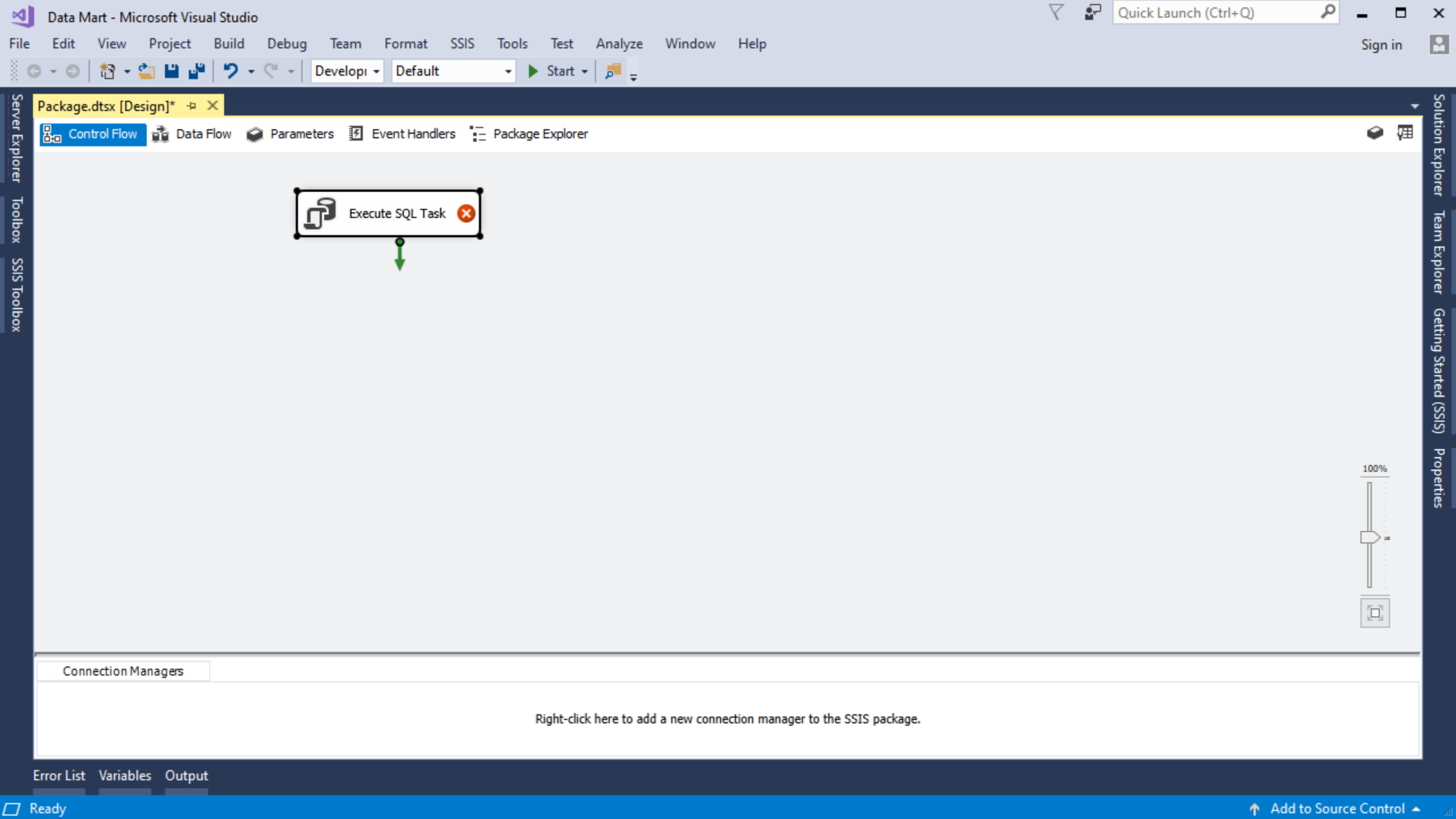
SSIS Toolbox

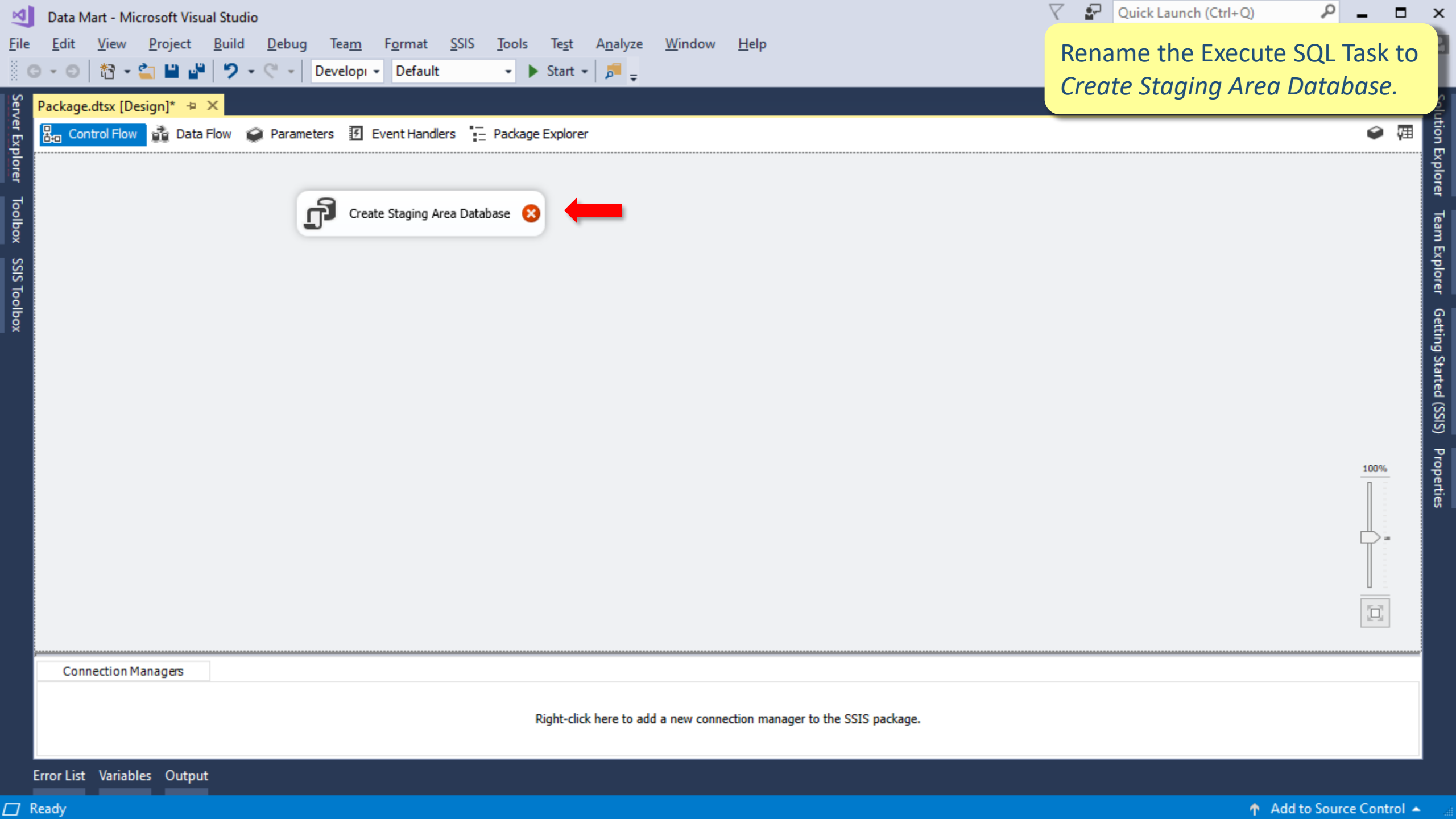
Search SSIS Toolbox

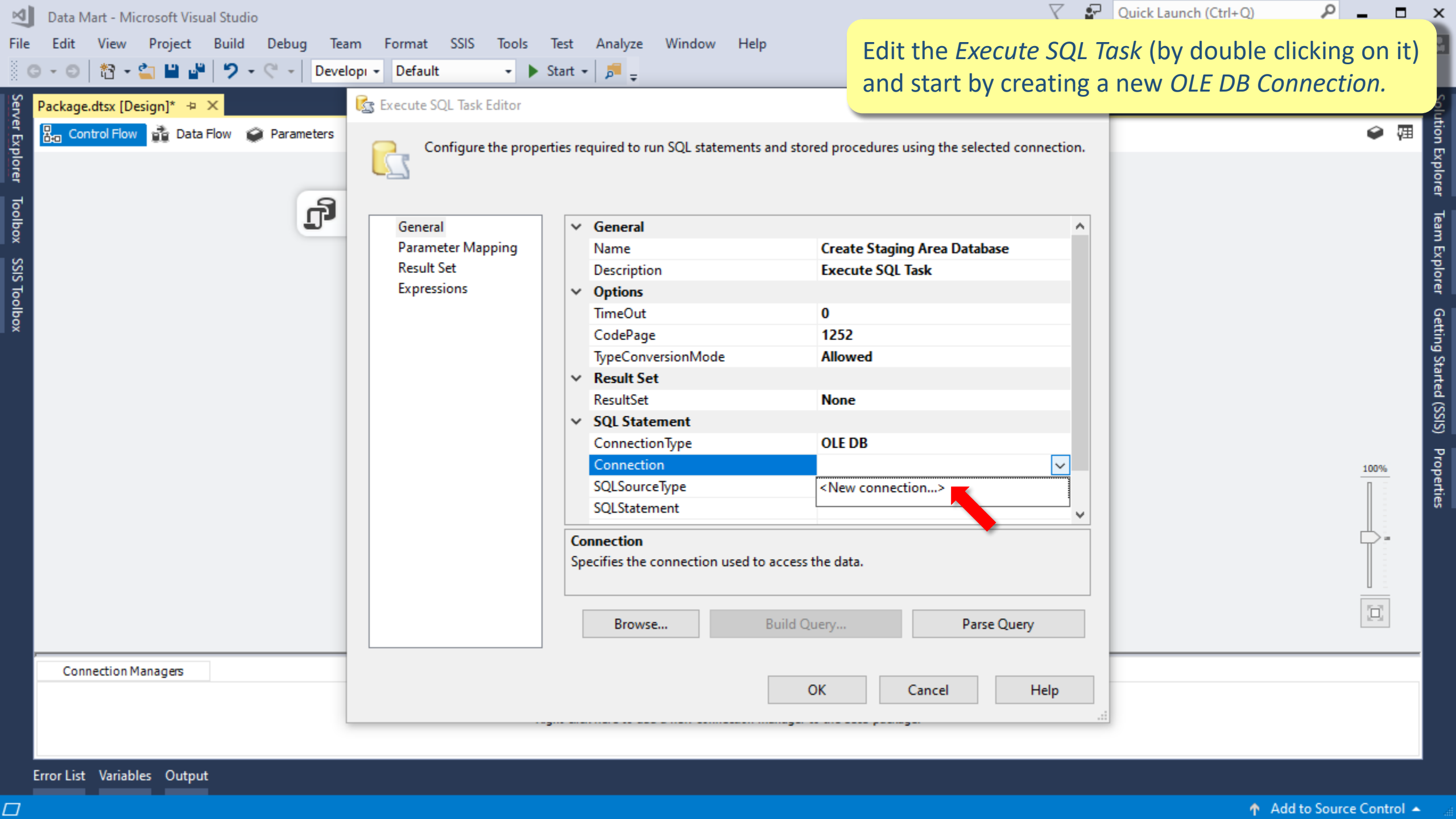
- Favorites
 - Data Flow Task
 - Execute SQL Task
- Common
 - Analysis Services Processing Task
 - Bulk Insert Task
 - Data Profiling Task
 - Execute Package Task
 - Execute Process Task
 - Expression Task
 - File System Task
 - FTP Task
 - Hadoop File System Task
 - Hadoop Hive Task
 - Hadoop Pig Task
 - Script Task
 - Send Mail Task
 - Web Service Task
 - XML Task
- Azure
 - Execute SQL Task
 - Executes SQL statements or stored procedures in a relational database. For example, truncate a table before starting a load, or create a foreign-key relationship after a load has completed.
 - [Find Samples](#)

Event Handlers Package Explorer

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







Edit the *Execute SQL Task* (by double clicking on it) and start by creating a new *OLE DB Connection*.

Execute SQL Task Editor

Configure the properties required to run SQL statements and stored procedures using the selected connection.

General
Parameter Mapping
Result Set
Expressions

General

Name	Create Staging Area Database
Description	Execute SQL Task

Options

TimeOut	0
CodePage	1252
TypeConversionMode	Allowed

Result Set

ResultSet	None
-----------	------

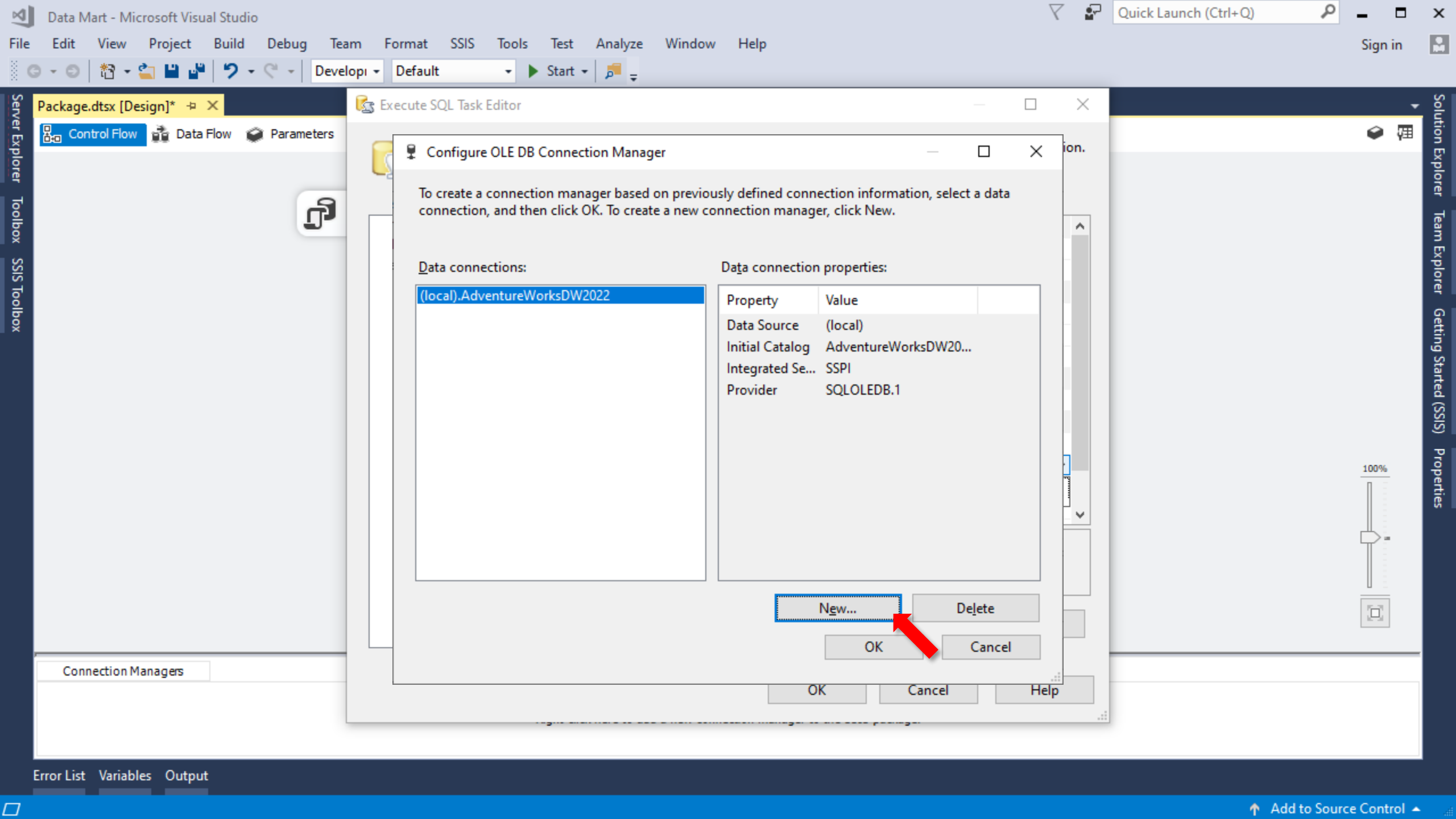
SQL Statement

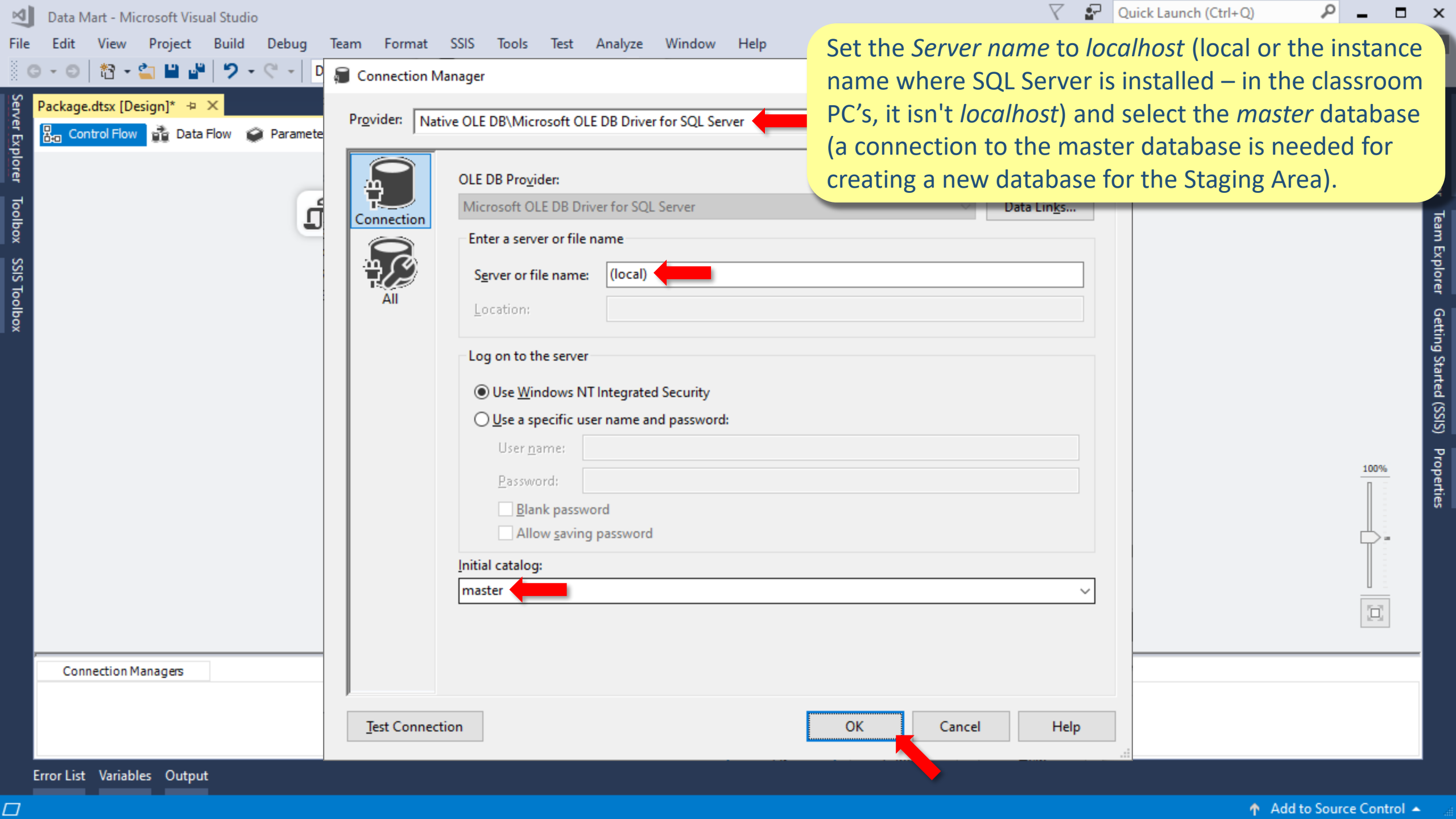
ConnectionType	OLE DB
Connection	<New connection...>
SQLSourceType	
SQLStatement	

Connection
Specifies the connection used to access the data.

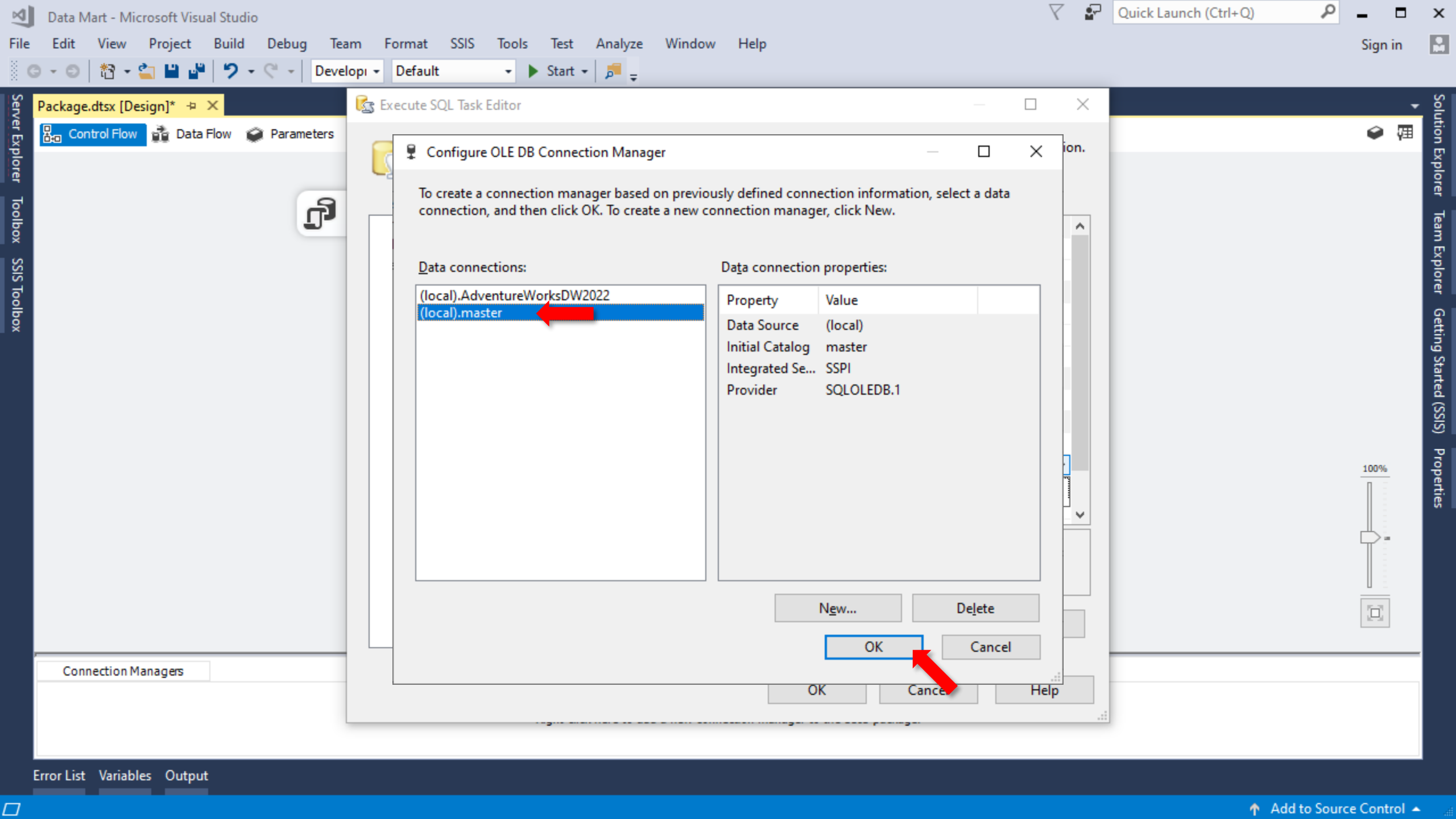
Browse... Build Query... Parse Query

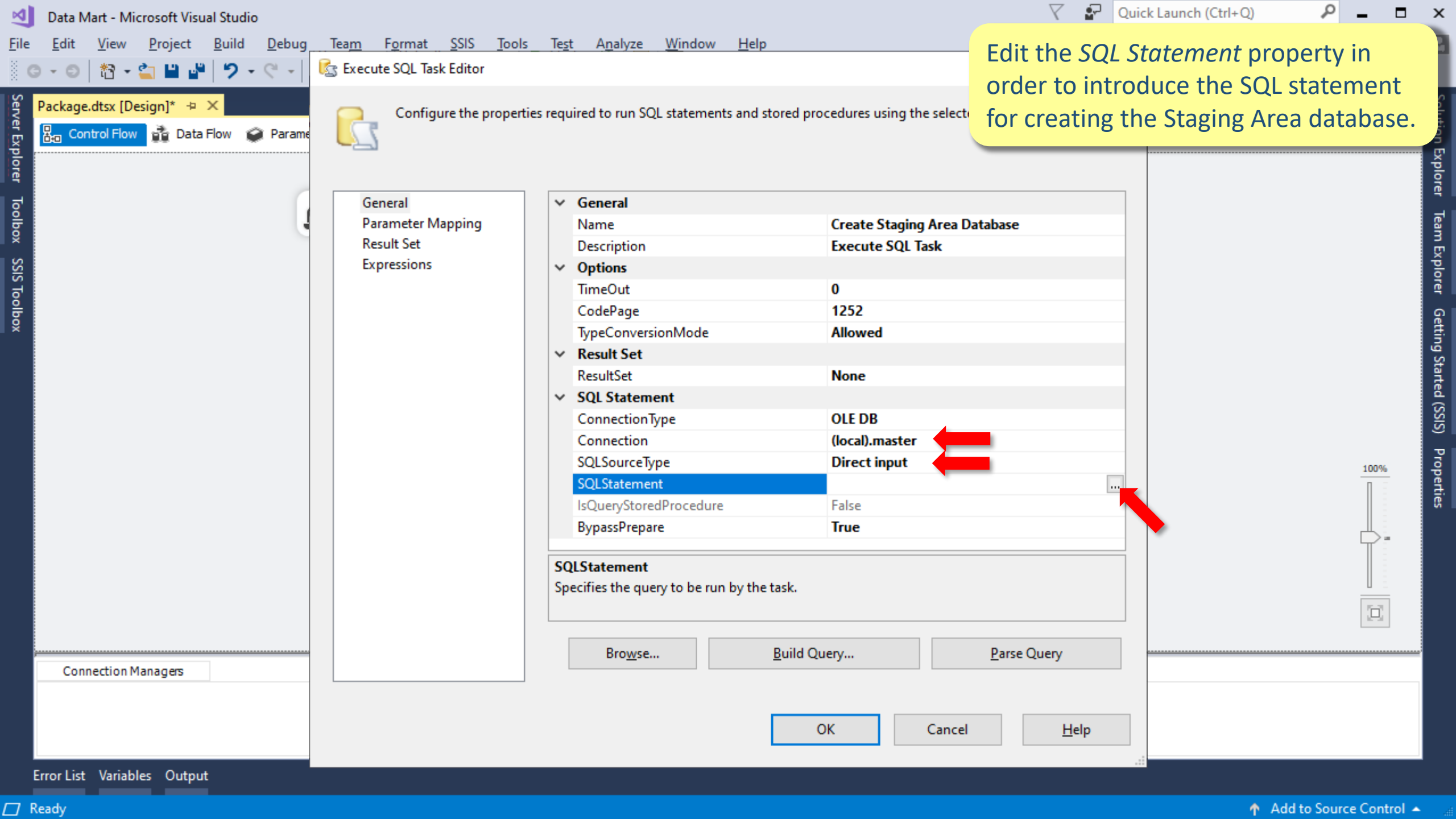
OK Cancel Help





Set the *Server* name to *localhost* (local or the instance name where SQL Server is installed – in the classroom PC's, it isn't *localhost*) and select the *master* database (a connection to the master database is needed for creating a new database for the Staging Area).

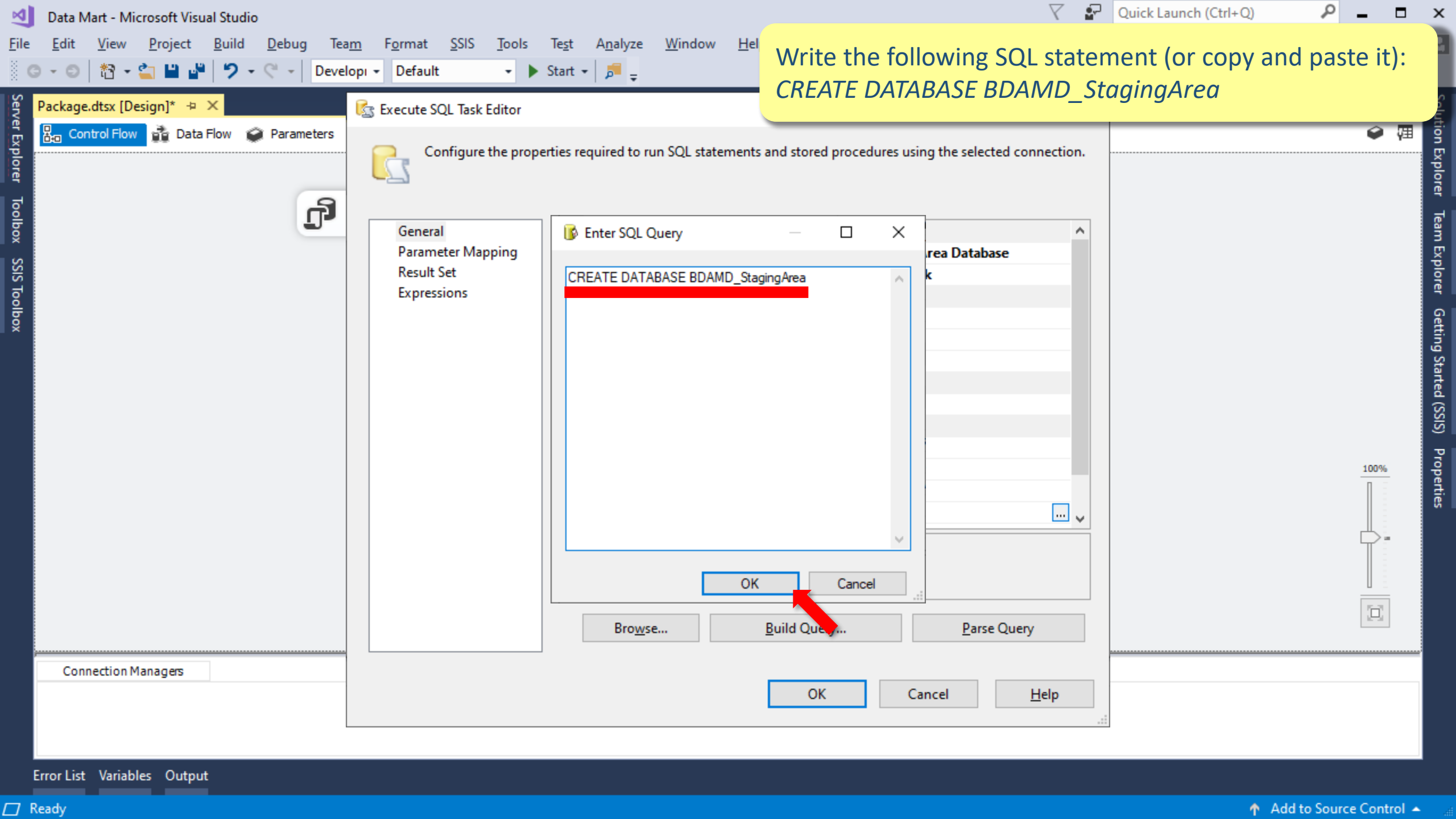




Edit the *SQL Statement* property in order to introduce the SQL statement for creating the Staging Area database.

General	
Name	Create Staging Area Database
Description	Execute SQL Task
Options	
TimeOut	0
CodePage	1252
TypeConversionMode	Allowed
Result Set	
ResultSet	None
SQL Statement	
ConnectionType	OLE DB
Connection	(local).master
SQLSourceType	Direct input
SQLStatement	
IsQueryStoredProcedure	False
BypassPrepare	True

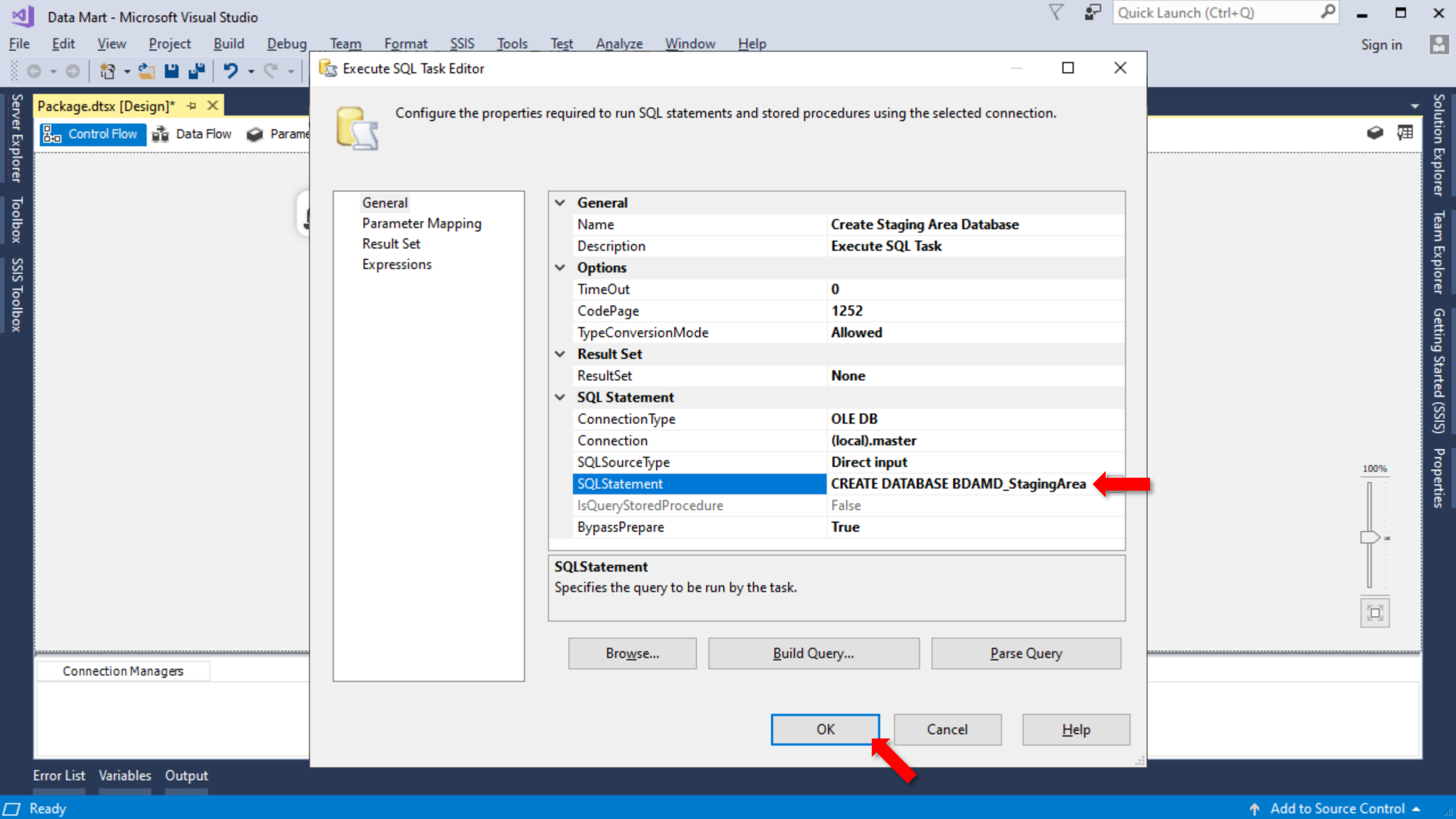
SQLStatement
Specifies the query to be run by the task.

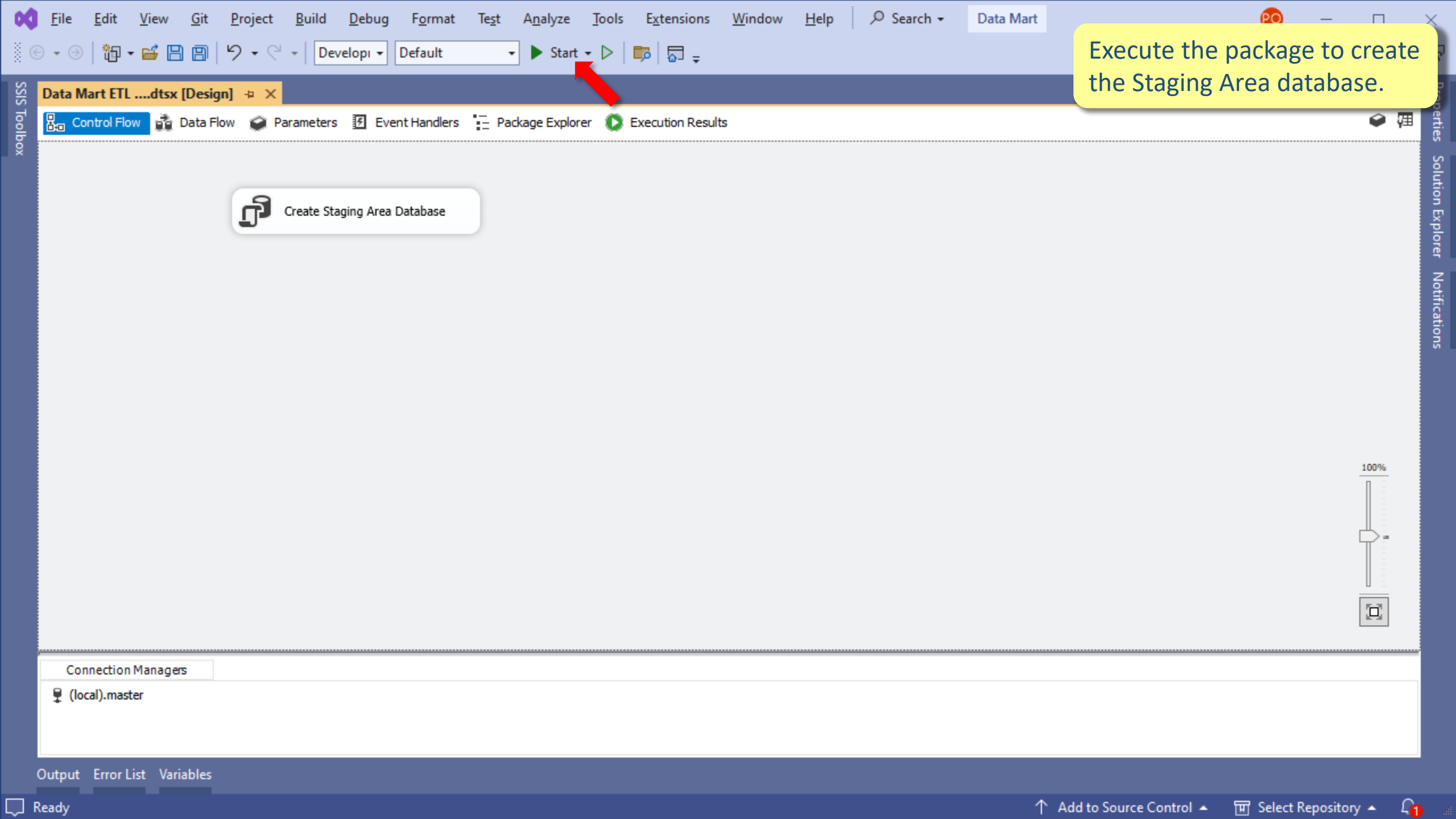


Write the following SQL statement (or copy and paste it):
CREATE DATABASE BDAMD_StagingArea

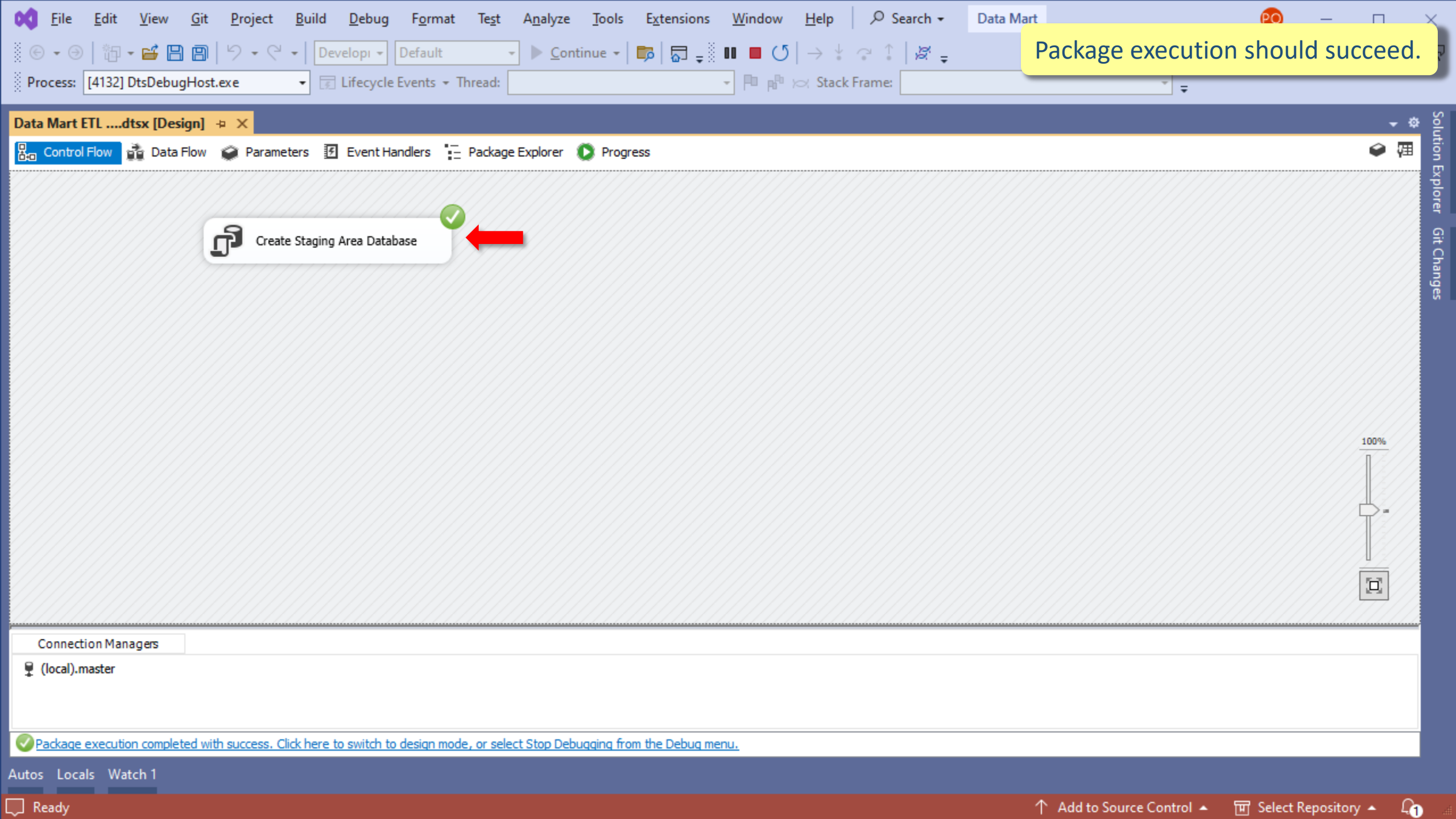
CREATE DATABASE BDAMD_StagingArea

OK





Execute the package to create the Staging Area database.





Start SQL Server Management Studio and connect to the database engine.



Connect to Server

SQL Server

Login | Connection Properties | Always Encrypted | Additional Connection Parameters

Server

Server type: Database Engine

Server name: localhost

Authentication: Windows Authentication

User name: AzureAD\PauloOliveira

Password:

☐ Remember password

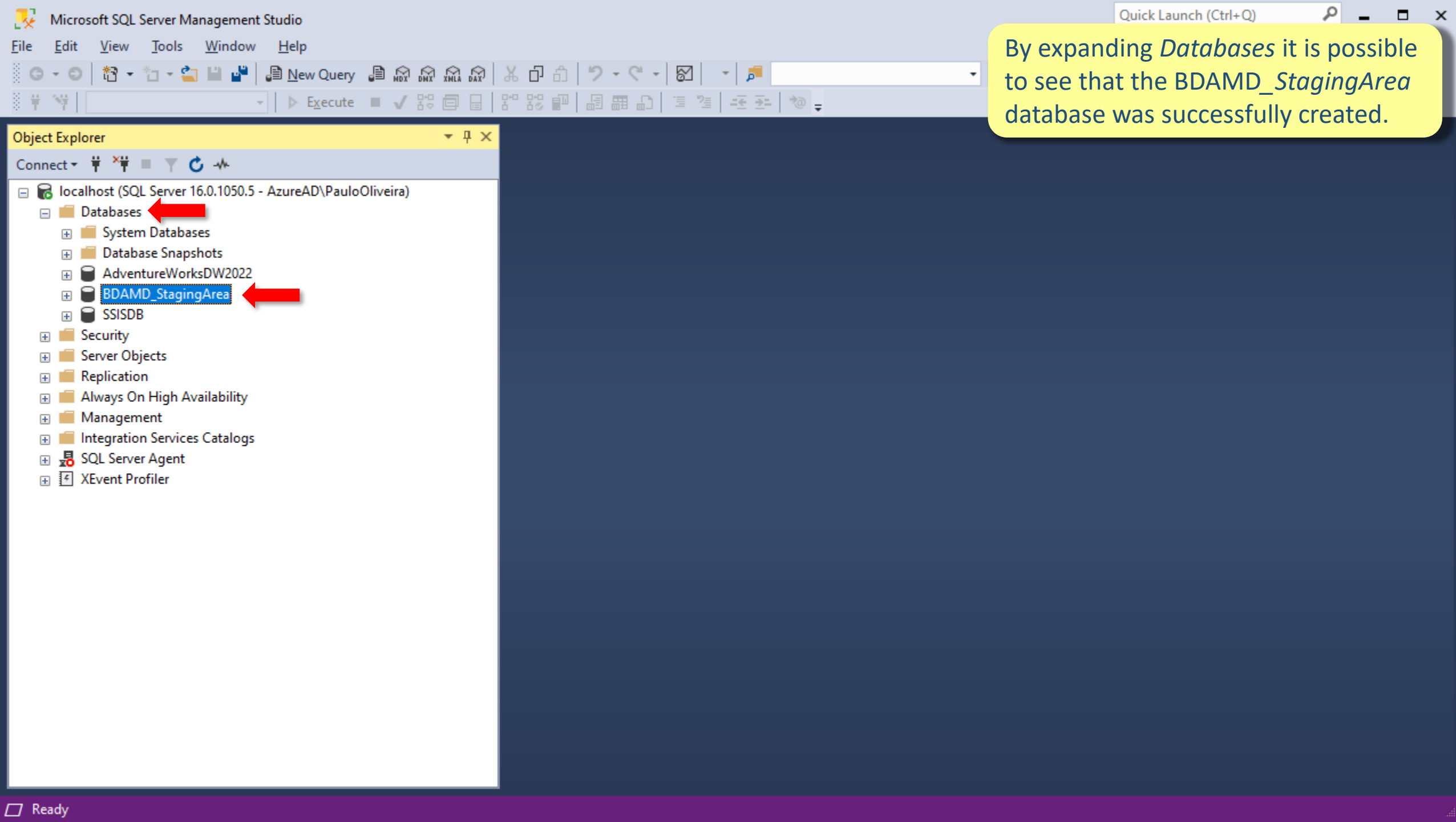
Connection Security

Encryption: Mandatory

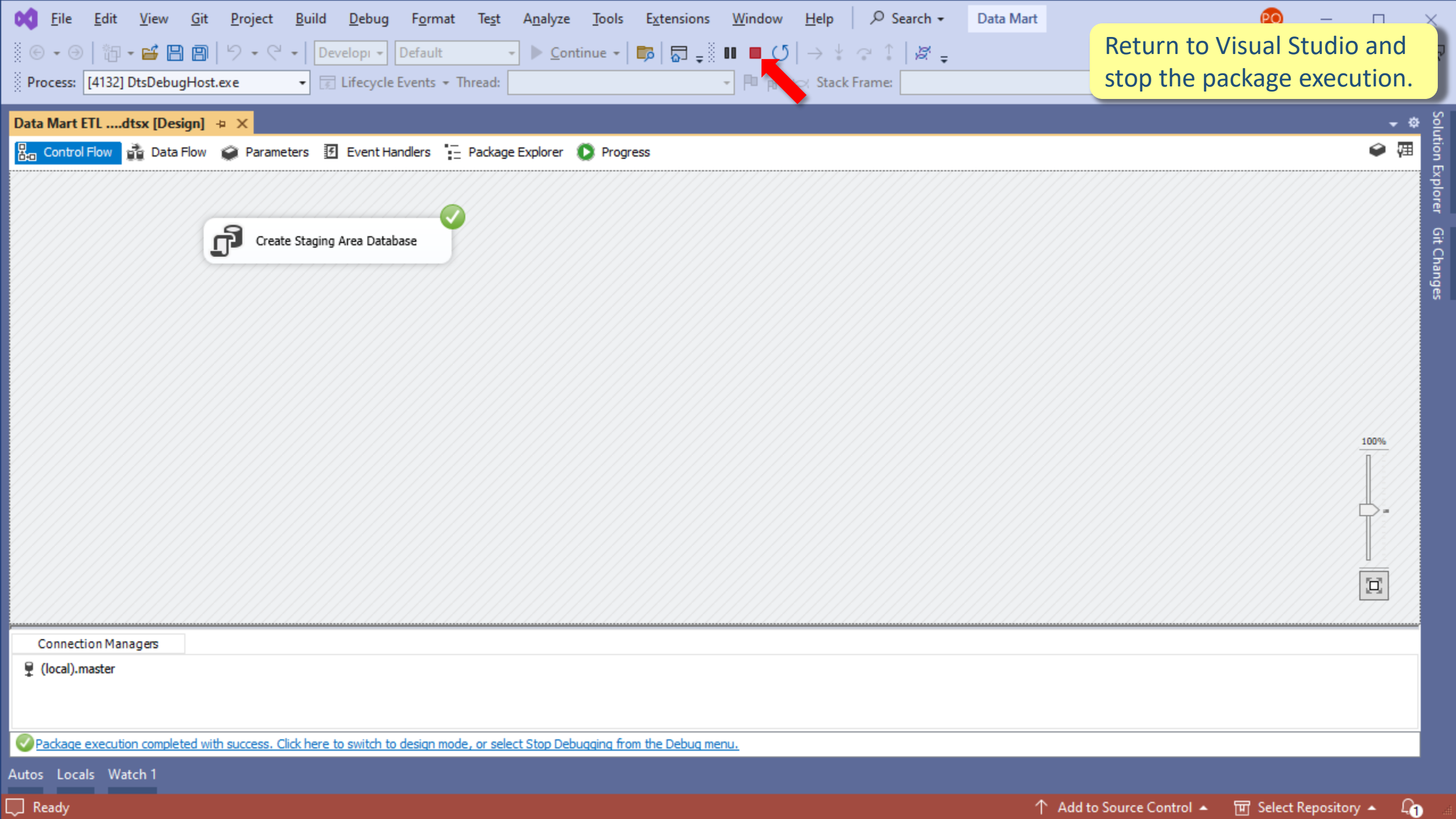
☒ Trust server certificate

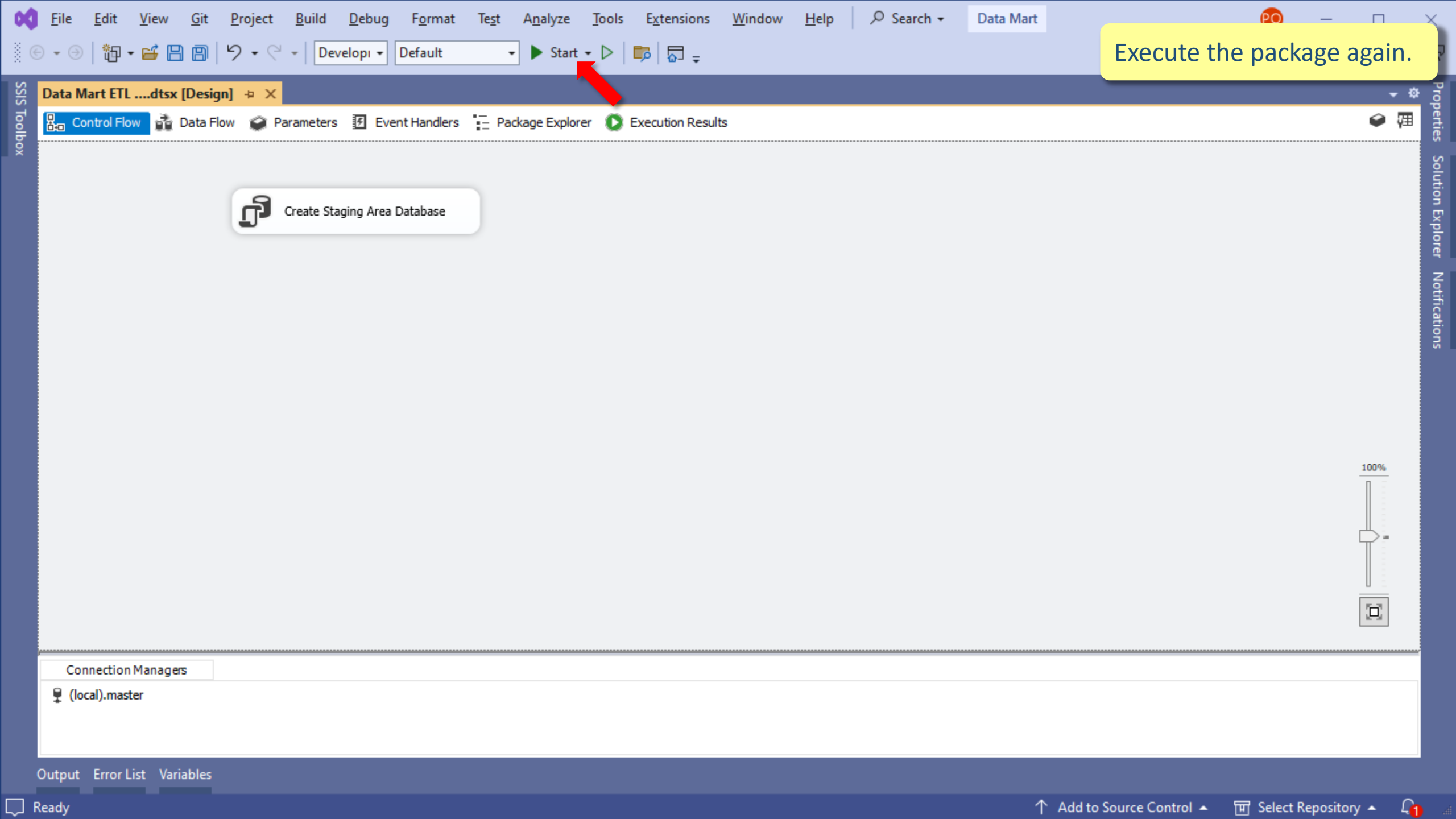
Host name in certificate:

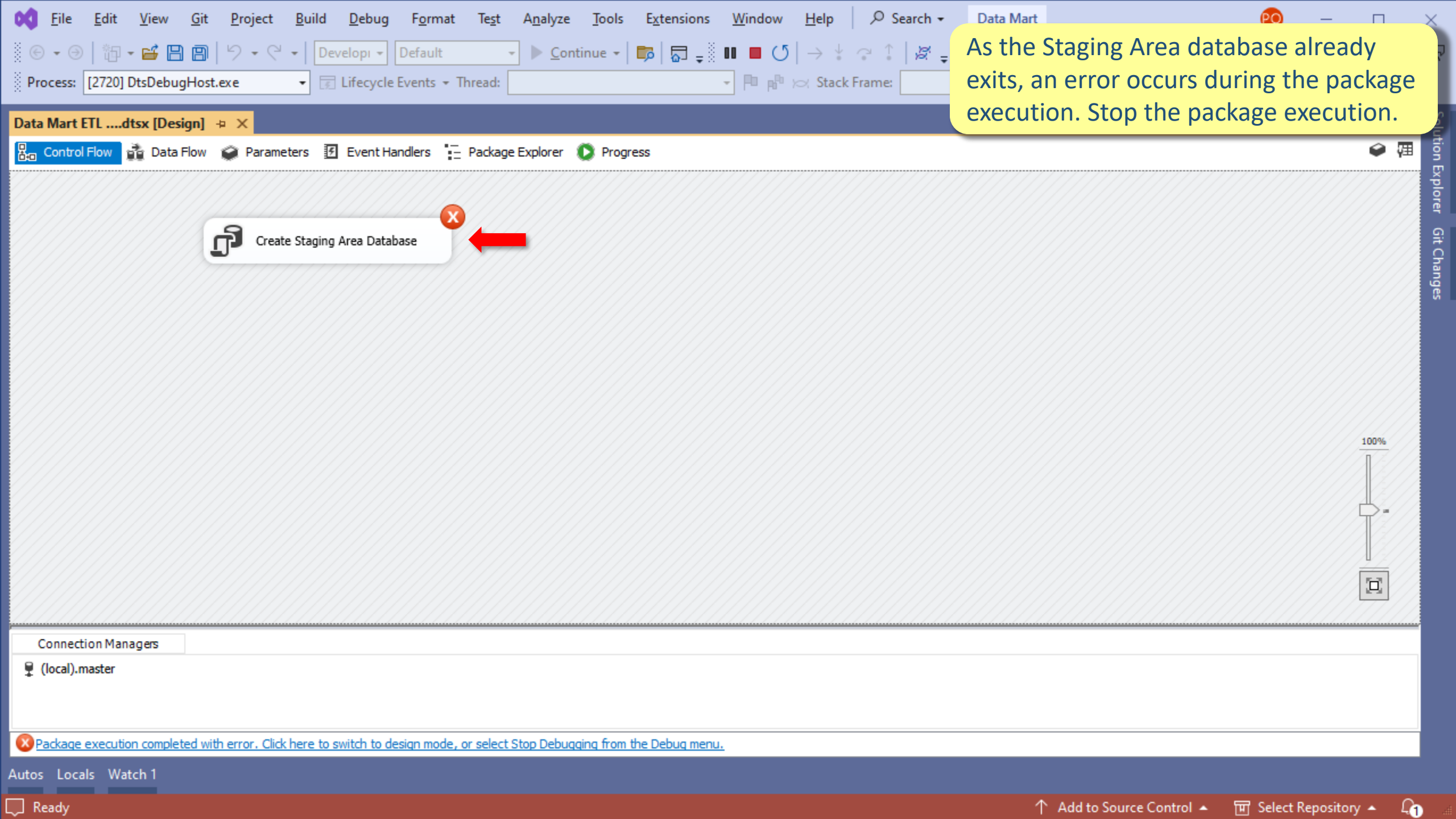
Connect Cancel Help Options <<



By expanding *Databases* it is possible to see that the *BDAMD_StagingArea* database was successfully created.







As the Staging Area database already exists, an error occurs during the package execution. Stop the package execution.

Create Staging Area Database

Connection Managers

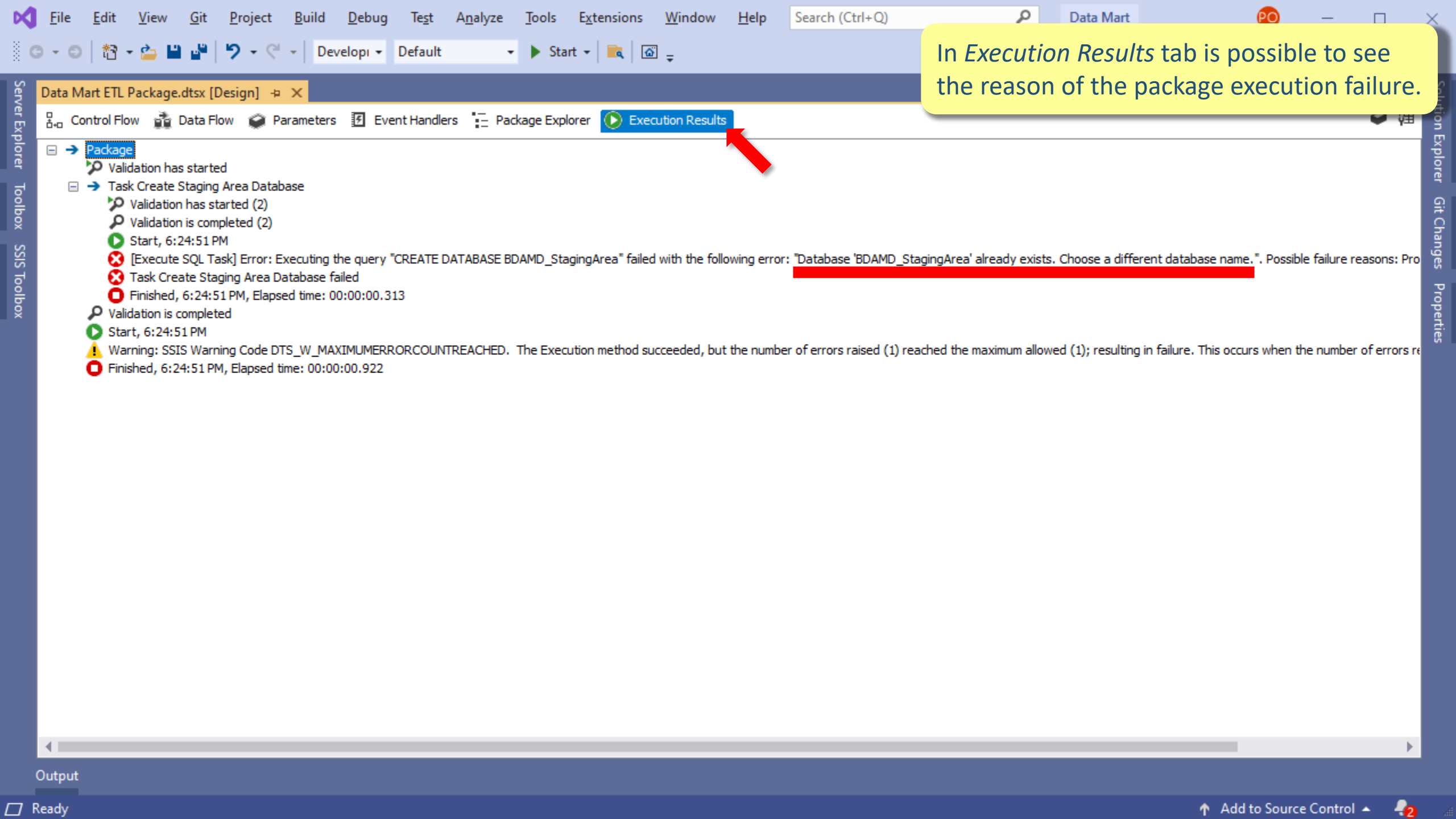
(local).master

Package execution completed with error. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

Autos Locals Watch 1

Ready

Add to Source Control Select Repository



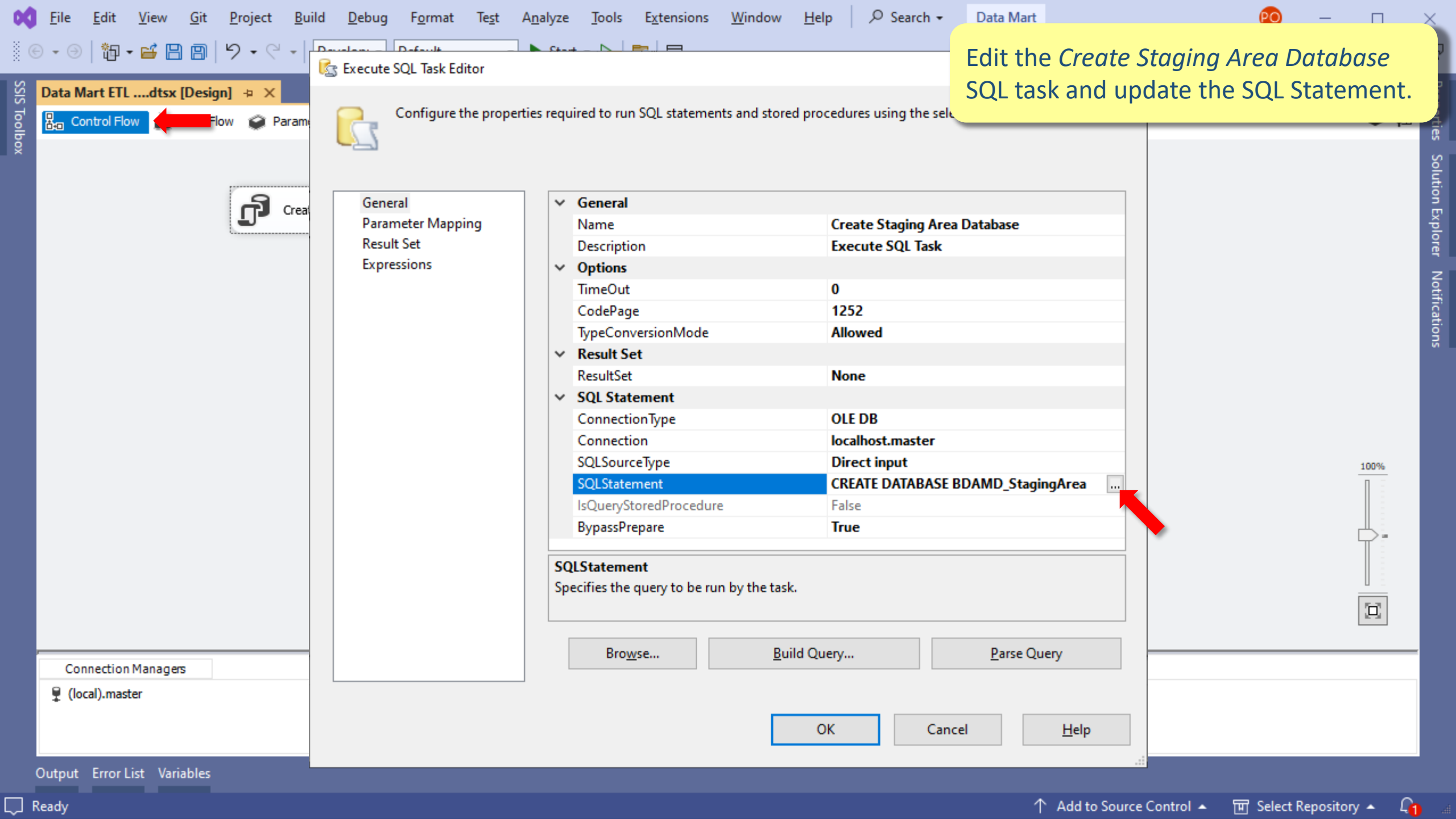
In *Execution Results* tab is possible to see the reason of the package execution failure.

Data Mart ETL Package.dtsx [Design]

Control Flow Data Flow Parameters Event Handlers Package Explorer Execution Results

- Package
 - Validation has started
 - Task Create Staging Area Database
 - Validation has started (2)
 - Validation is completed (2)
 - Start, 6:24:51 PM
 - [Execute SQL Task] Error: Executing the query "CREATE DATABASE BDAMD_StagingArea" failed with the following error: "Database 'BDAMD_StagingArea' already exists. Choose a different database name.". Possible failure reasons: Pro
 - Task Create Staging Area Database failed
 - Finished, 6:24:51 PM, Elapsed time: 00:00:00.313
 - Validation is completed
 - Start, 6:24:51 PM
 - Warning: SSIS Warning Code DTS_W_MAXIMUMERRORCOUNTREACHED. The Execution method succeeded, but the number of errors raised (1) reached the maximum allowed (1); resulting in failure. This occurs when the number of errors re
 - Finished, 6:24:51 PM, Elapsed time: 00:00:00.922

Output



Update the SQL statement to the following in order to create the database only if it doesn't exist:
*IF NOT EXISTS (SELECT name
FROM sys.databases
WHERE name = 'BDAMD_StagingArea')
CREATE DATABASE BDAMD_StagingArea*

Execute SQL Task Editor

Configure the properties required to run SQL statements and store

General

Parameter Mapping
Result Set
Expressions

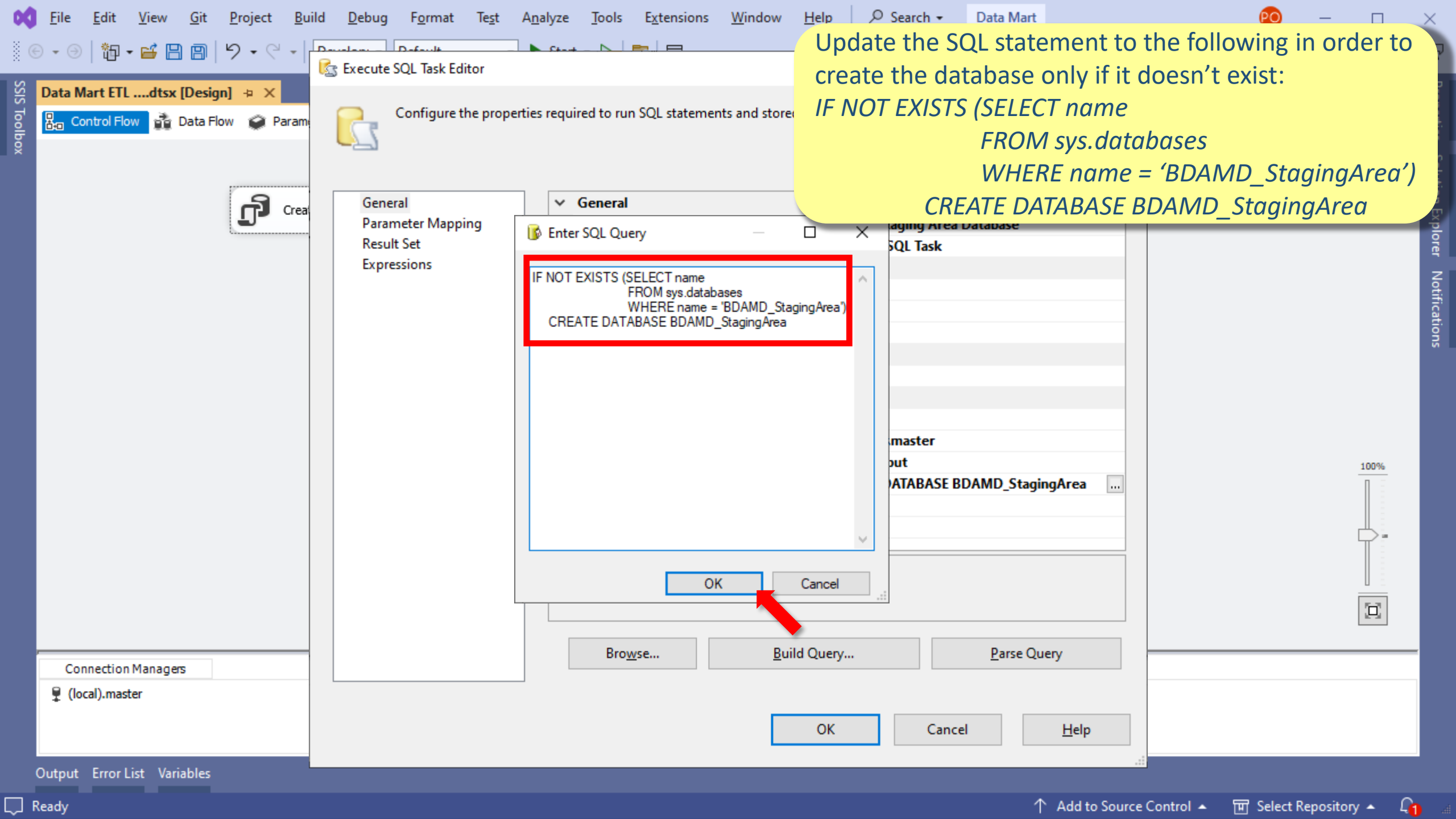
Enter SQL Query

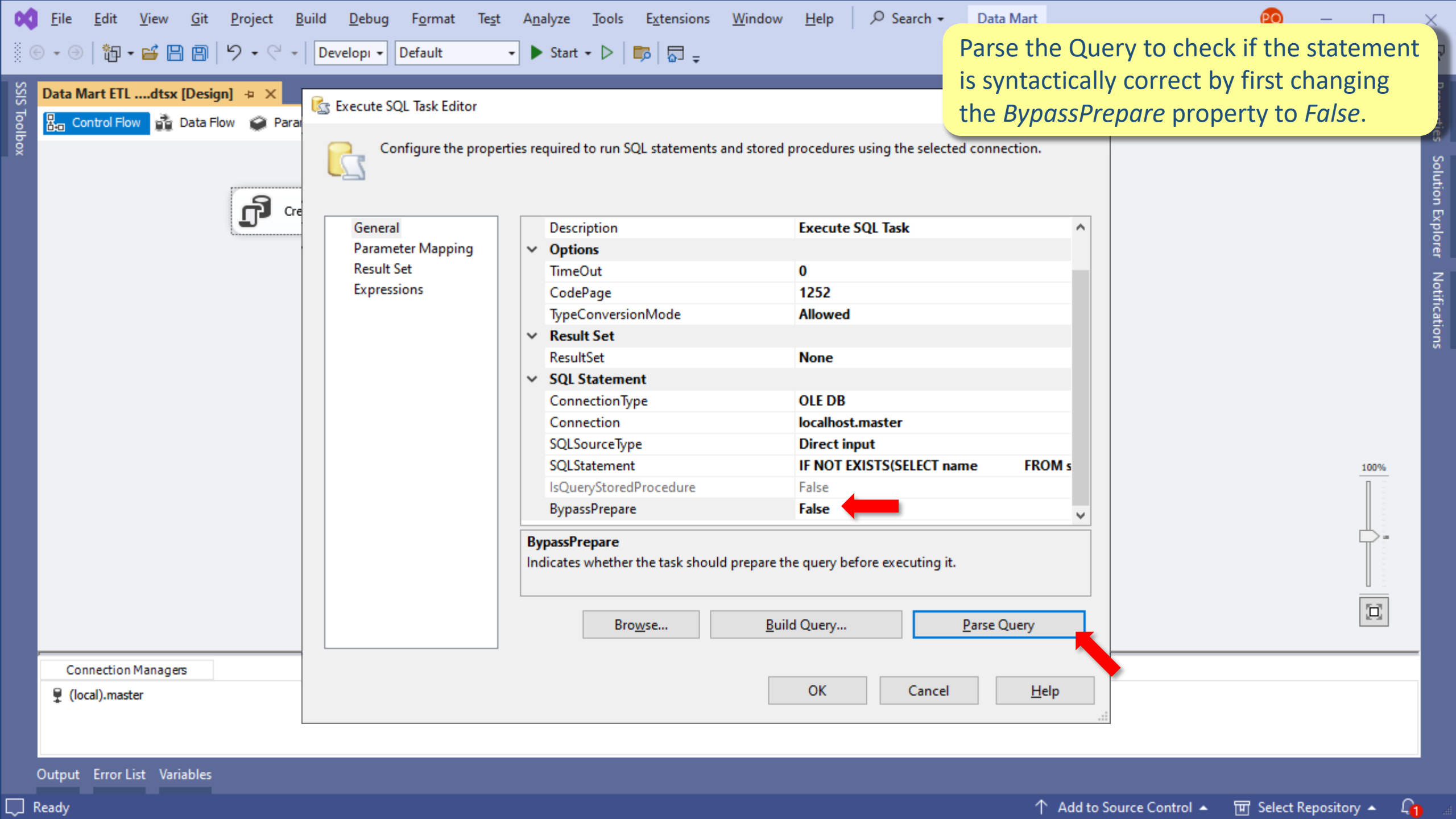
```
IF NOT EXISTS (SELECT name
FROM sys.databases
WHERE name = 'BDAMD_StagingArea')
CREATE DATABASE BDAMD_StagingArea
```

OK Cancel

Browse... Build Query... Parse Query

OK Cancel Help





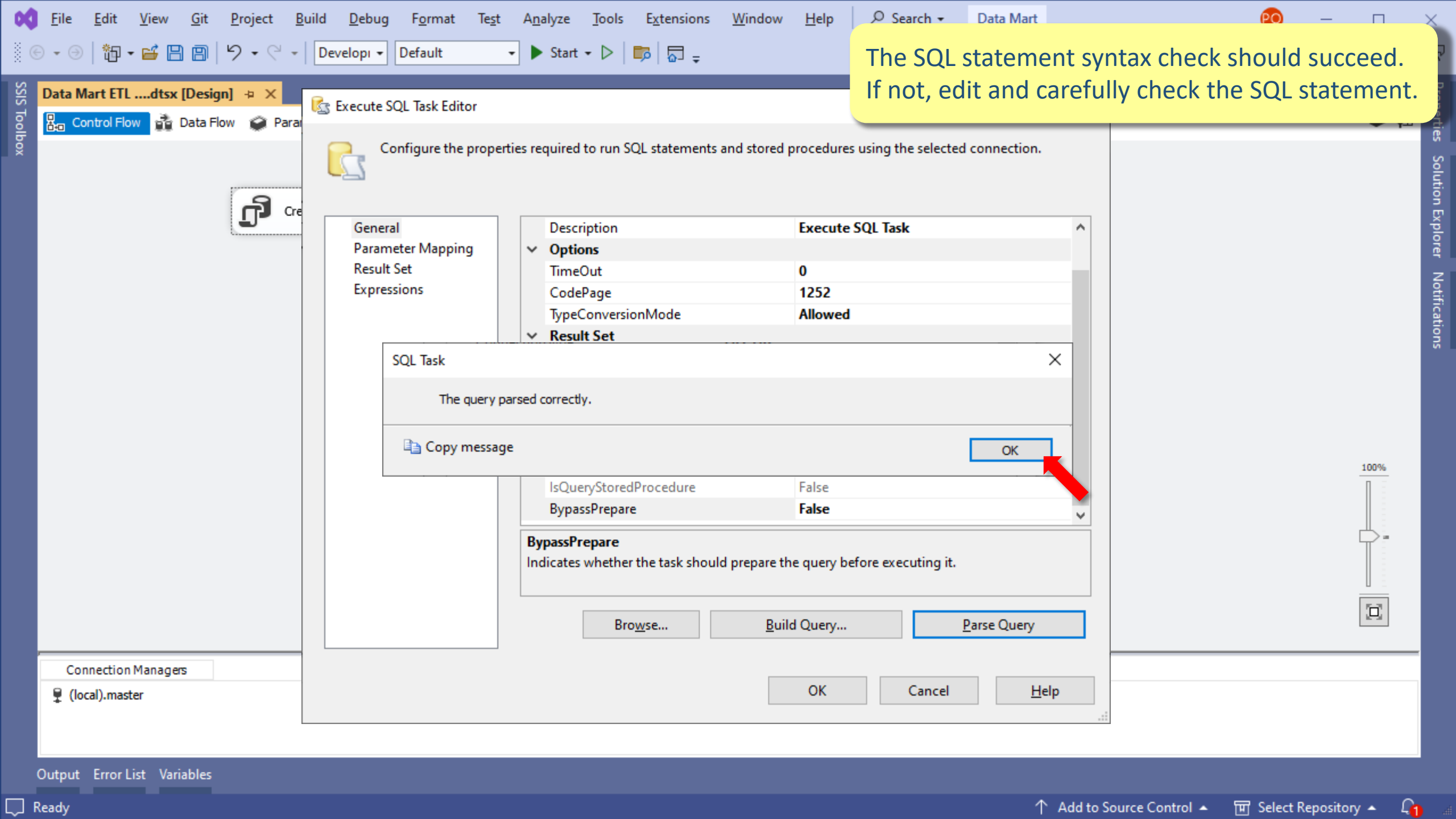
Parse the Query to check if the statement is syntactically correct by first changing the *BypassPrepare* property to *False*.

Description	Execute SQL Task
Options	
TimeOut	0
CodePage	1252
TypeConversionMode	Allowed
Result Set	
ResultSet	None
SQL Statement	
ConnectionType	OLE DB
Connection	localhost.master
SQLSourceType	Direct input
SQLStatement	IF NOT EXISTS(SELECT name FROM s
IsQueryStoredProcedure	False
BypassPrepare	False

BypassPrepare
Indicates whether the task should prepare the query before executing it.

Browse... Build Query... Parse Query

OK Cancel Help



The SQL statement syntax check should succeed.
If not, edit and carefully check the SQL statement.

SQL Task

The query parsed correctly.

Copy message

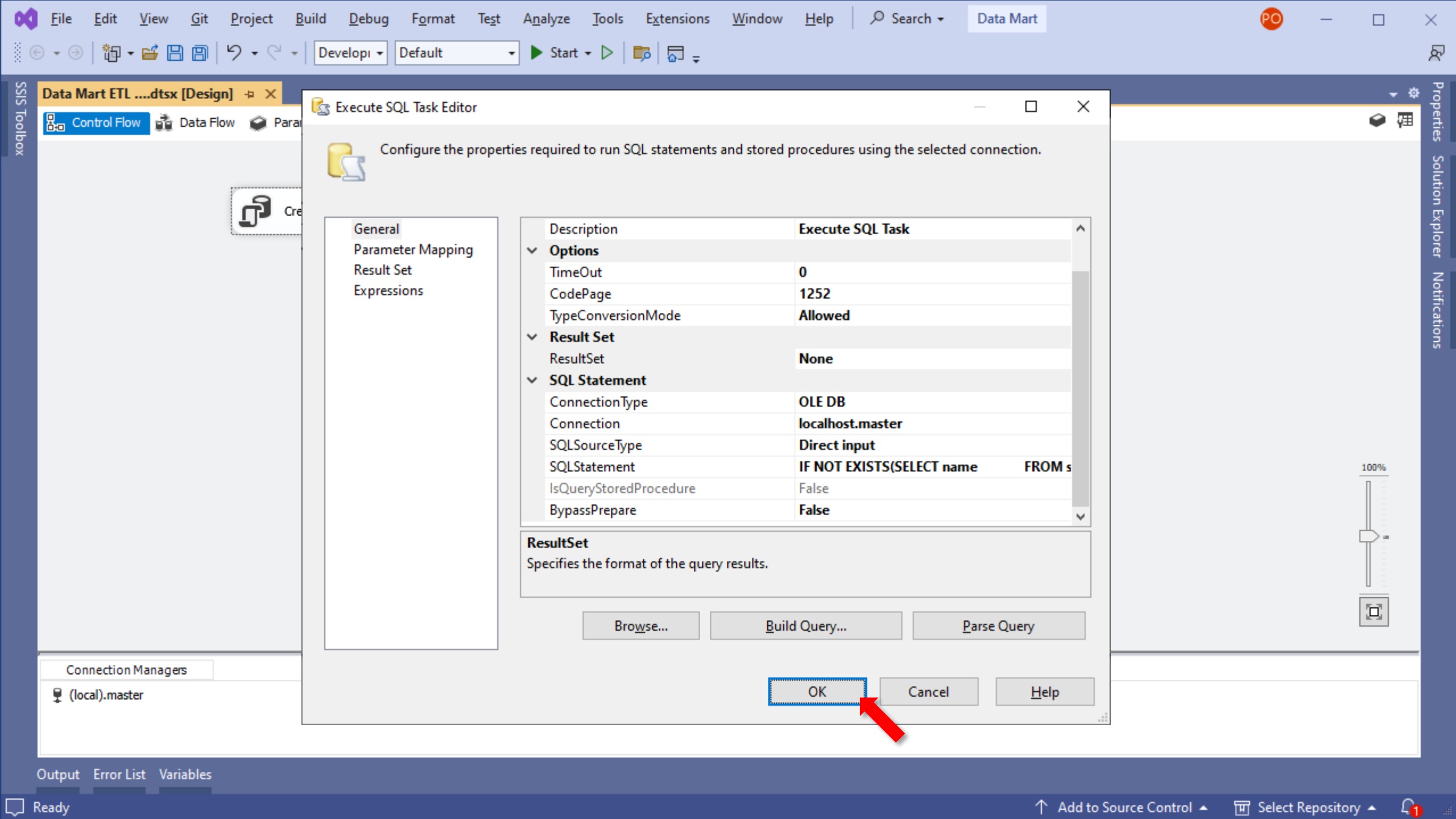
OK

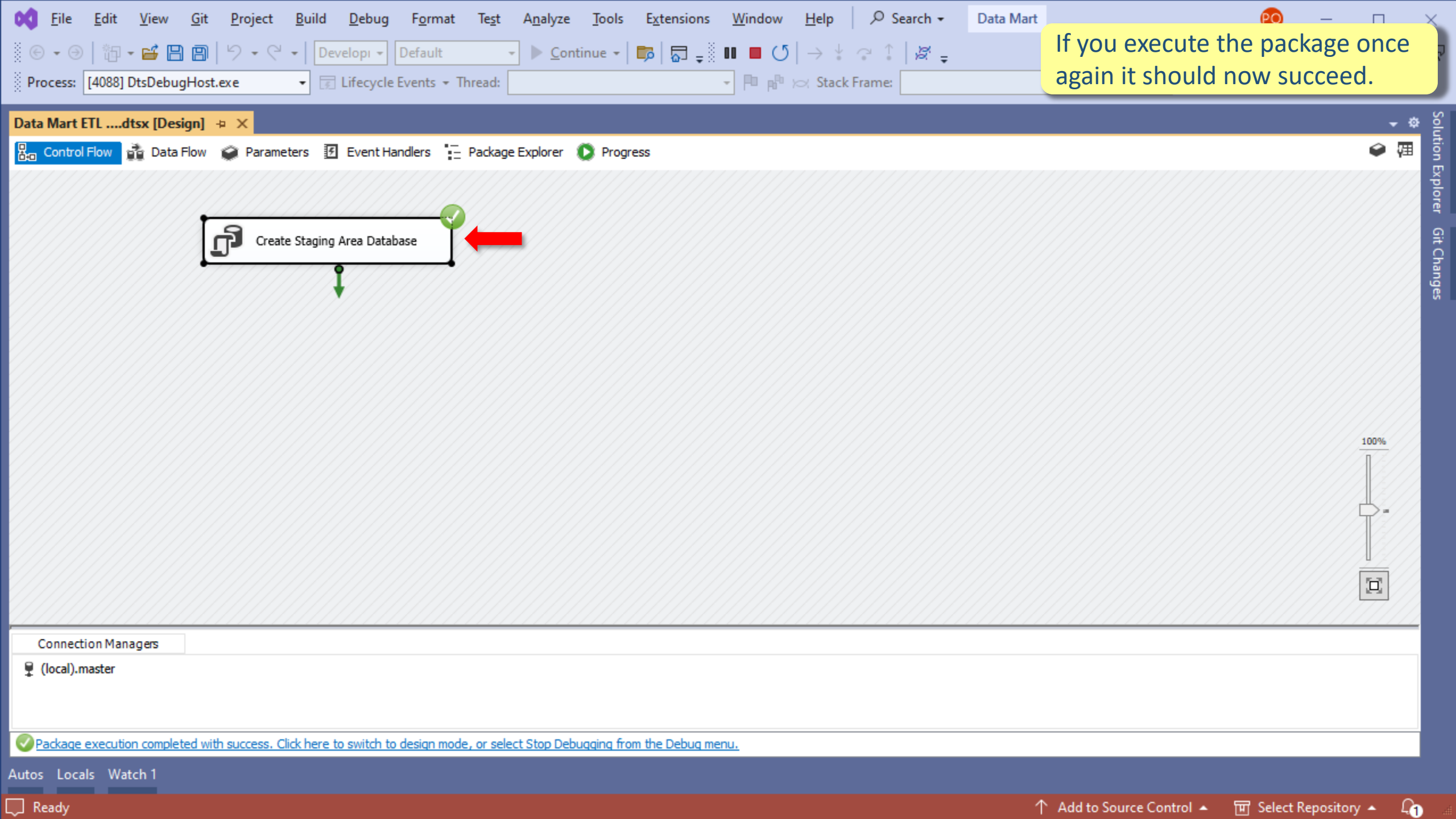
IsQueryStoredProcedure	False
BypassPrepare	False

BypassPrepare
Indicates whether the task should prepare the query before executing it.

Browse... Build Query... Parse Query

OK Cancel Help





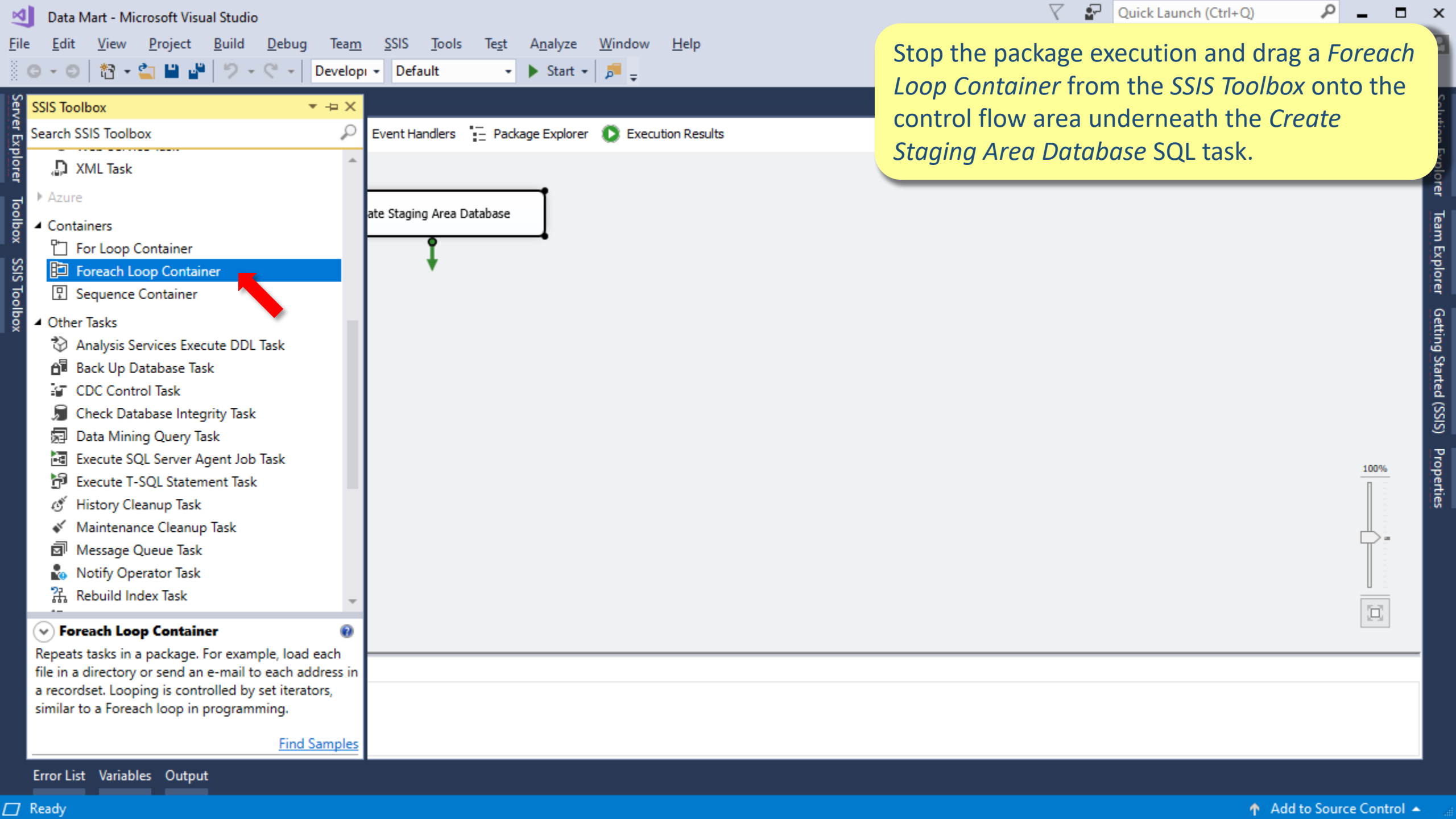
If you execute the package once again it should now succeed.



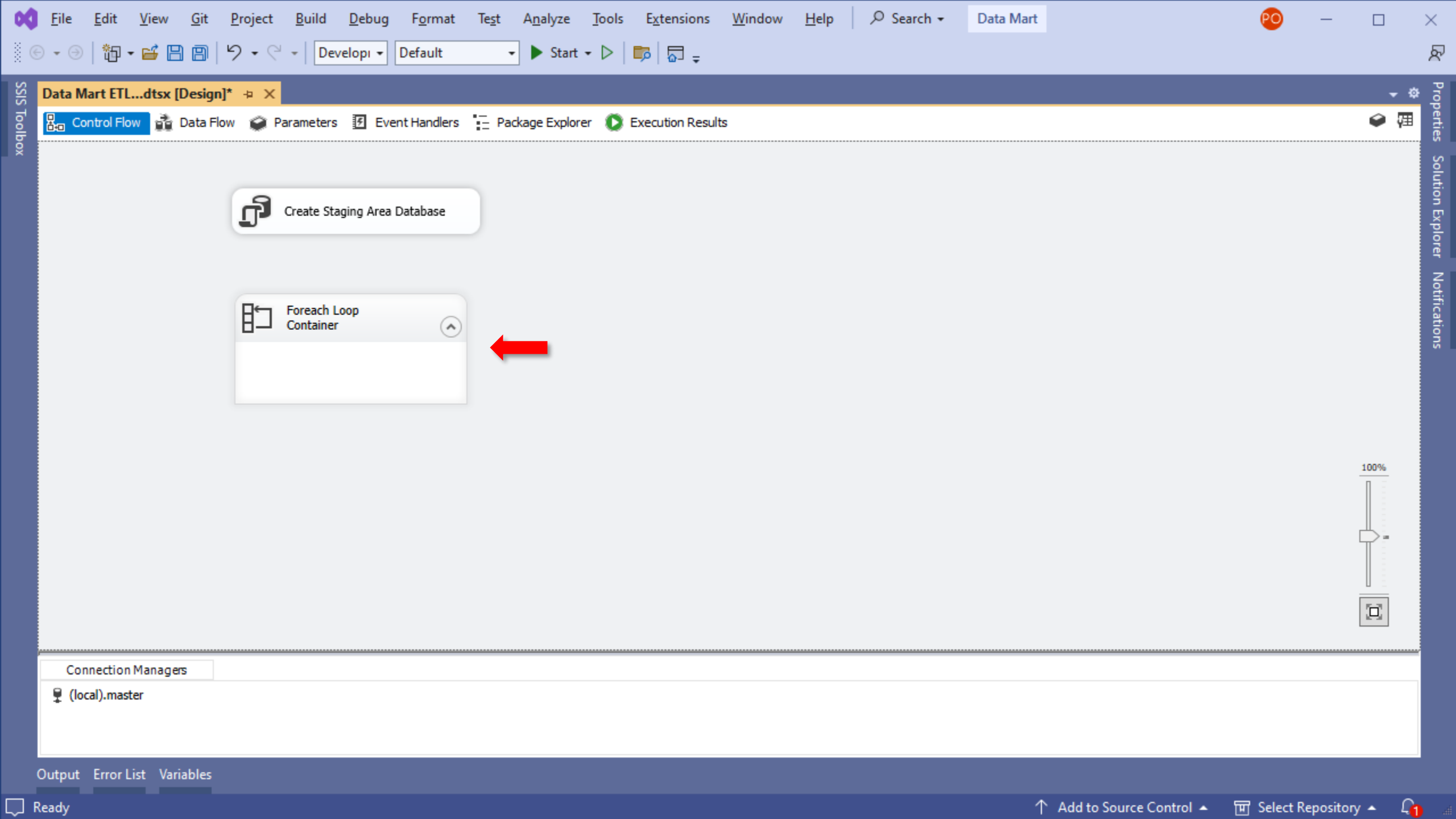
Connection Managers
(local).master

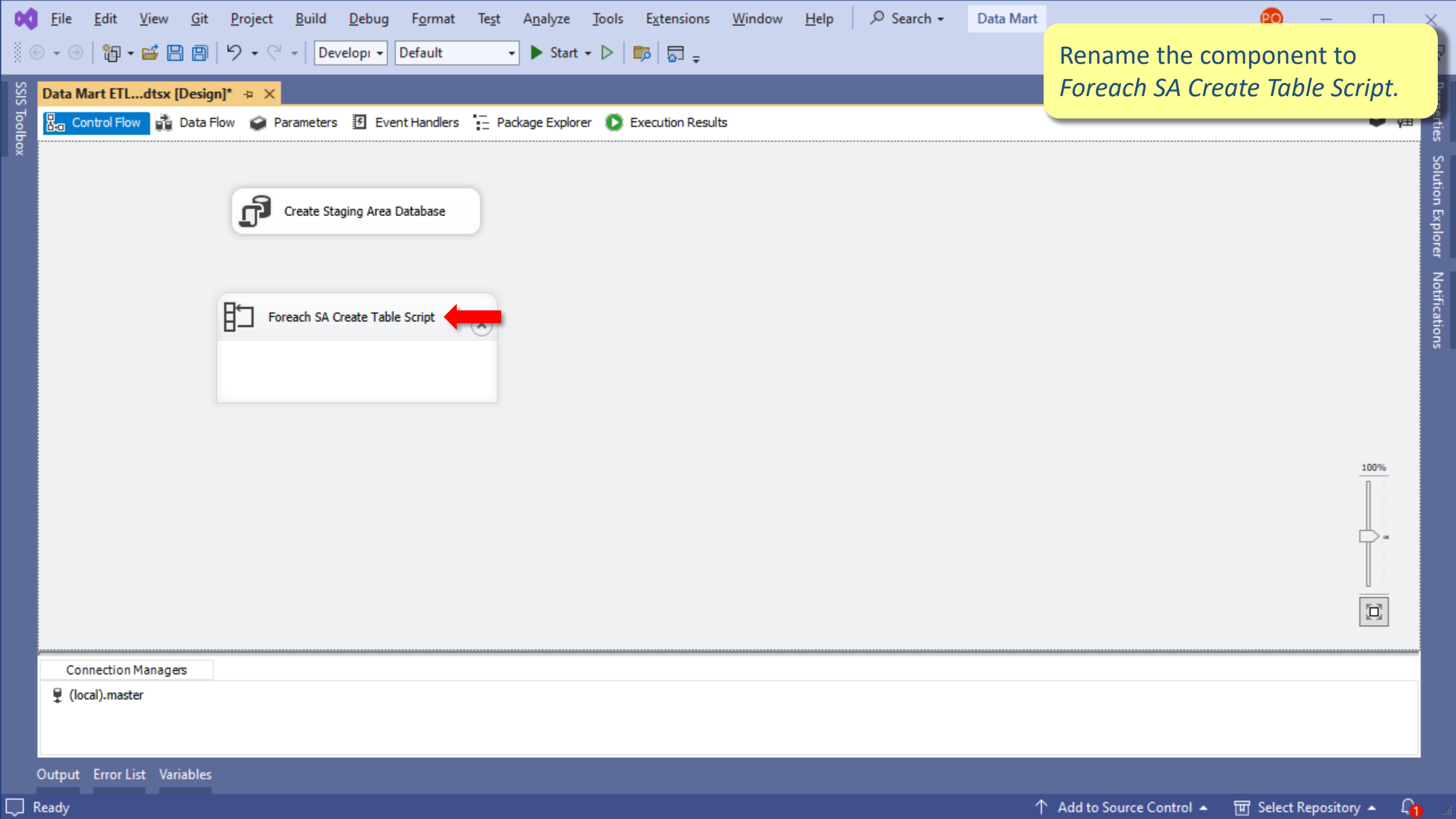
Package execution completed with success. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

Autos Locals Watch 1



Stop the package execution and drag a *Foreach Loop Container* from the *SSIS Toolbox* onto the control flow area underneath the *Create Staging Area Database SQL* task.





Rename the component to
Foreach SA Create Table Script.

Data Mart ETL...dtsx [Design]*

Control Flow Data Flow Parameters Event Handlers Package Explorer Execution Results

Create Staging Area Database

Foreach SA Create Table Script

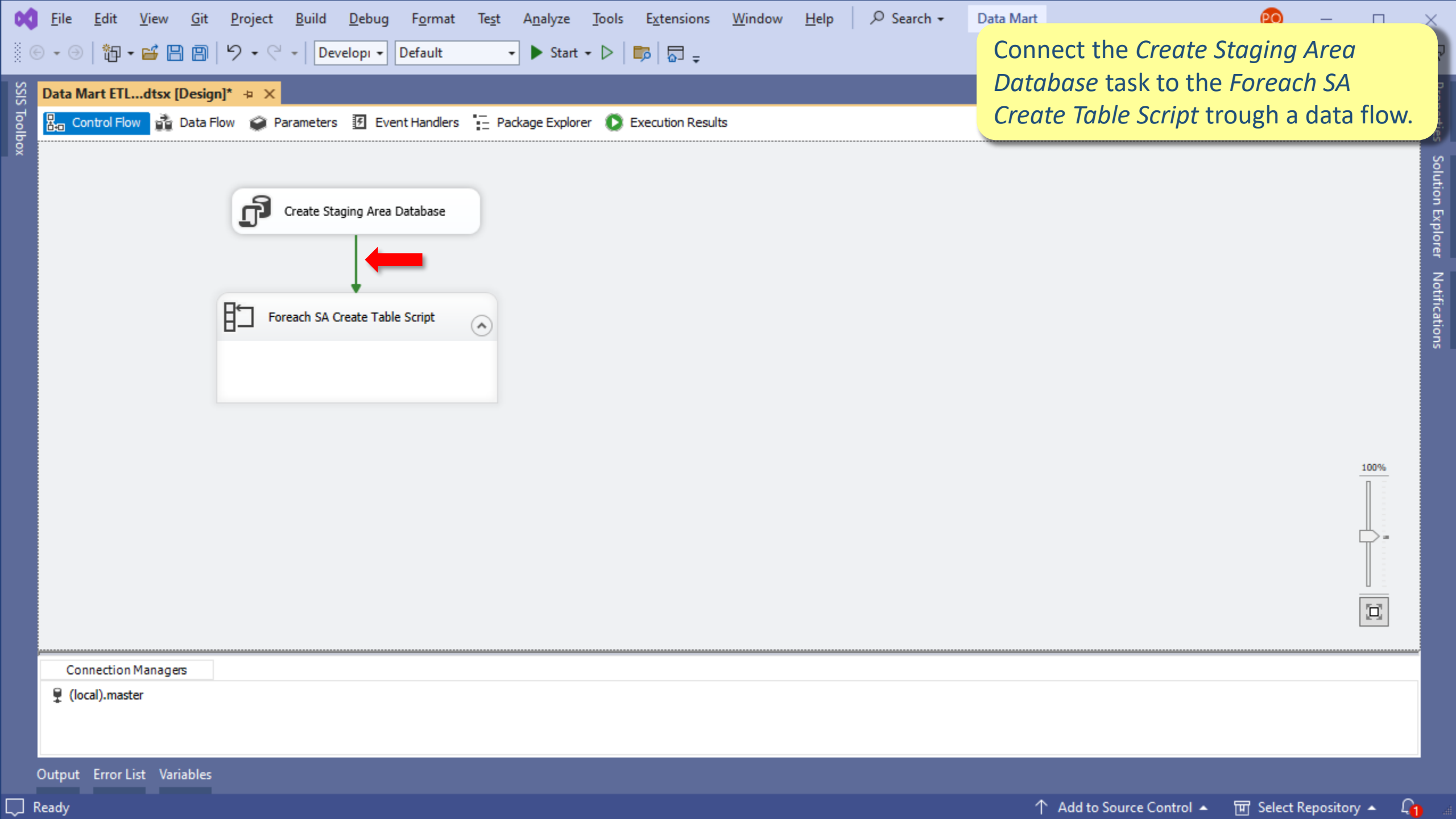
Connection Managers

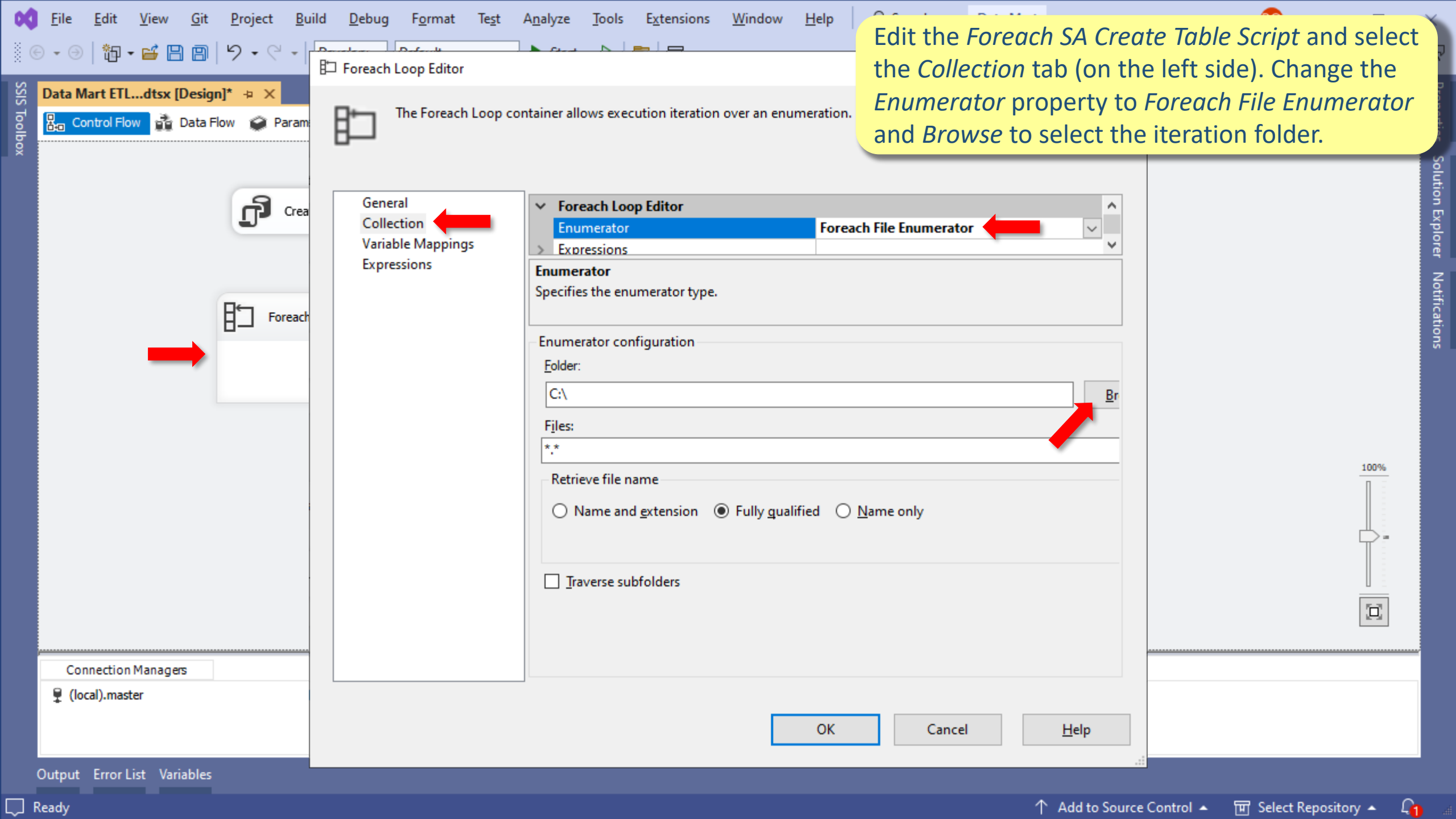
(local).master

Output Error List Variables

Ready

Add to Source Control Select Repository





Edit the *Foreach SA Create Table Script* and select the *Collection* tab (on the left side). Change the *Enumerator* property to *Foreach File Enumerator* and *Browse* to select the iteration folder.

General
Collection
Variable Mappings
Expressions

▼ Foreach Loop Editor
Enumerator
Expressions

Enumerator
Specifies the enumerator type.

Enumerator configuration

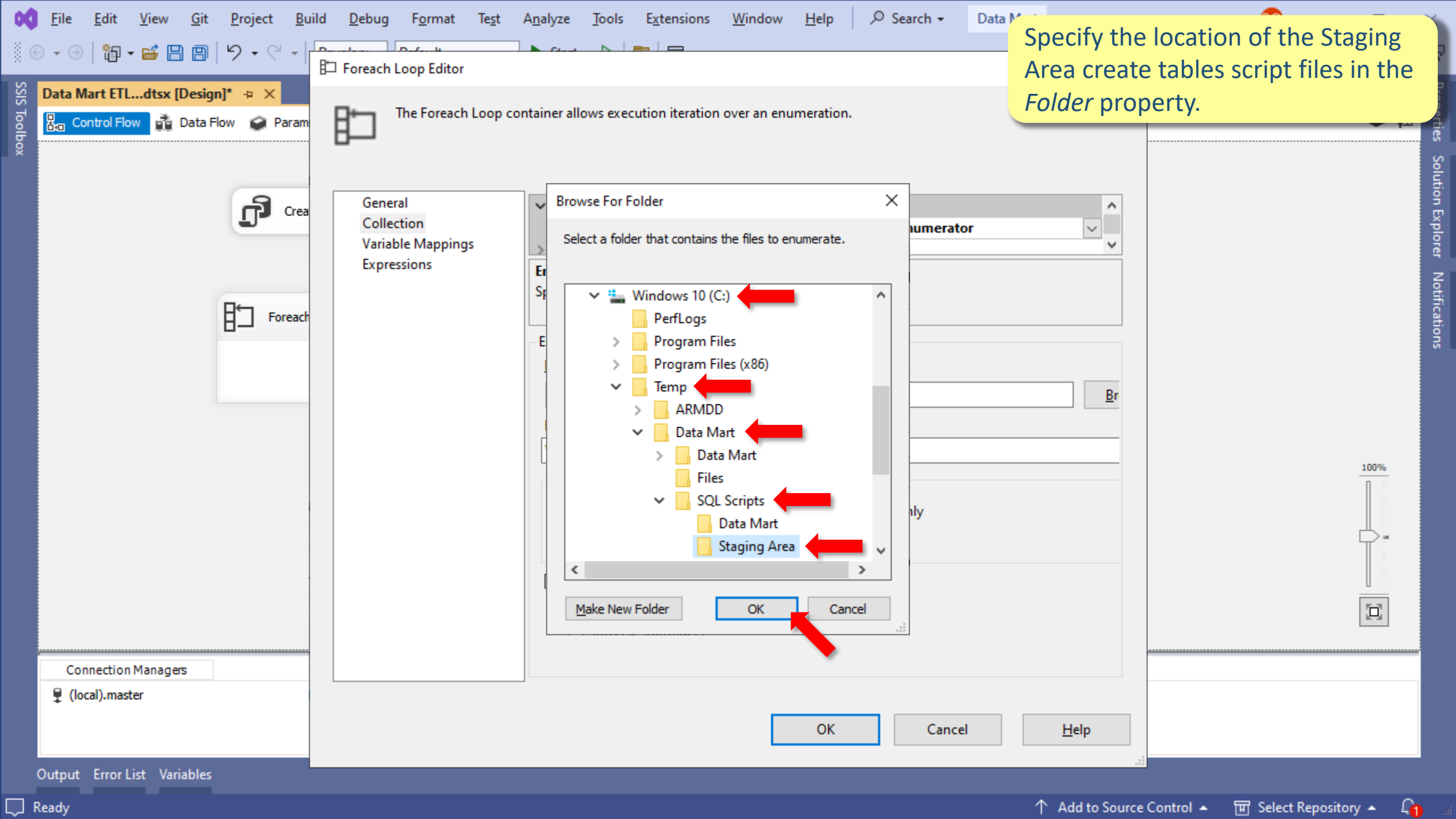
Folder:
C:\

Files:
**

Retrieve file name
☐ Name and extension ☒ Fully qualified ☐ Name only

☐ Traverse subfolders

OK Cancel Help



Specify the location of the Staging Area create tables script files in the *Folder* property.

In the *Files* property it's possible to restrict the iteration just to SQL files by introducing: **.sql*

File Edit View Git Project Build Debug Format Test Analyze Tools Extensions Window Help

SSIS Toolbox

Data Mart ETL...dtsx [Design]*

Control Flow Data Flow Parameters

Properties Solution Explorer Notifications

100%

Ready

Output Error List Variables

Up Add to Source Control Select Repository

Foreach Loop Editor

The Foreach Loop container allows execution iteration over an enumeration.

General
Collection
Variable Mappings
Expressions

Foreach Loop Editor

Enumerator	Foreach File Enumerator
Expressions	

Expressions
A collection of expressions. The evaluation result of each expression is assigned to a property an...

Enumerator configuration

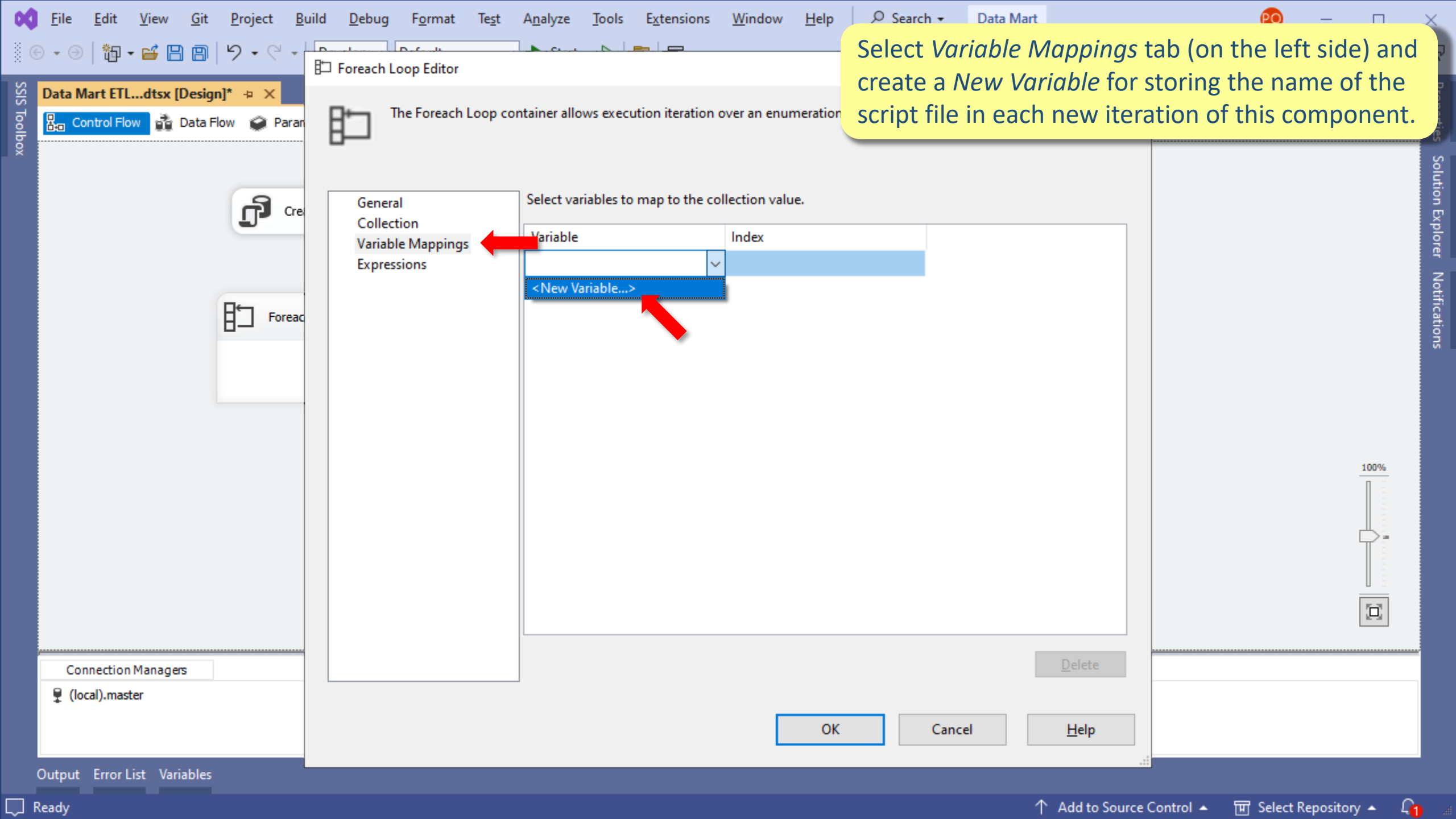
Folder:
C:\Temp\Data Mart\SQL Scripts\Staging Area

Files:
*.sql

Retrieve file name
☐ Name and extension ☒ Fully qualified ☐ Name only

☐ Traverse subfolders

OK Cancel Help



Select *Variable Mappings* tab (on the left side) and create a *New Variable* for storing the name of the script file in each new iteration of this component.

General
Collection
Variable Mappings
Expressions

Select variables to map to the collection value.

Variable	Index
<New Variable...>	

Delete

OK Cancel Help

Set the variable name as *SAScriptFilename*.

Foreach Loop Editor



The Foreach Loop container allows execution iteration over an enumeration.

General

Collection

Variable Mappings

Expressions

Select variables to map to the collection value.

Add Variable

Specify the properties of the new variable.

Container:

Data Mart ETL Package

Name:

SAScriptFilename

Namespace:

User

Value type:

String

Value:

☐ Read only

OK

Cancel

OK

Cancel

Help

Delete

Connection Managers

(local).master

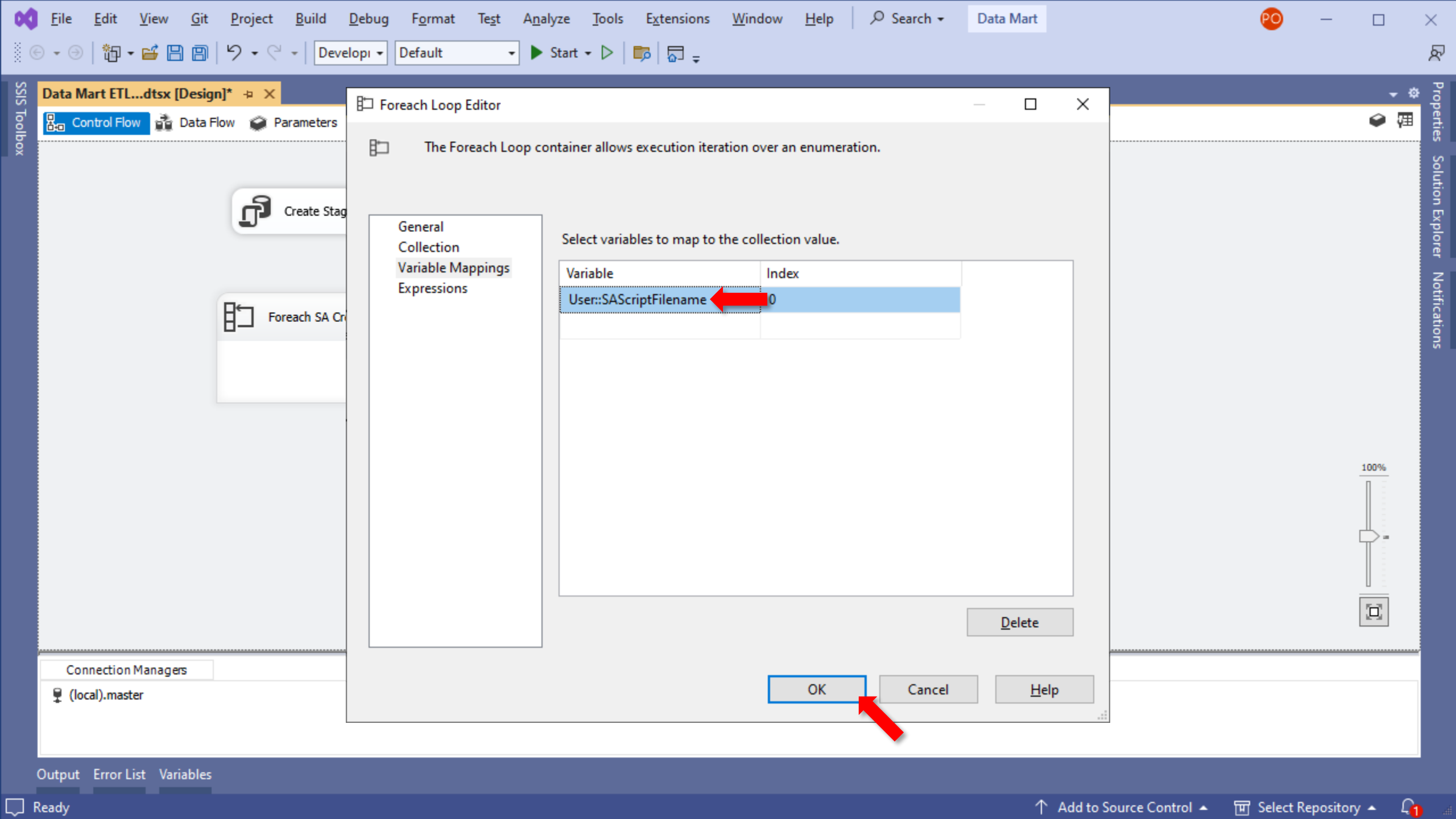
Output Error List Variables

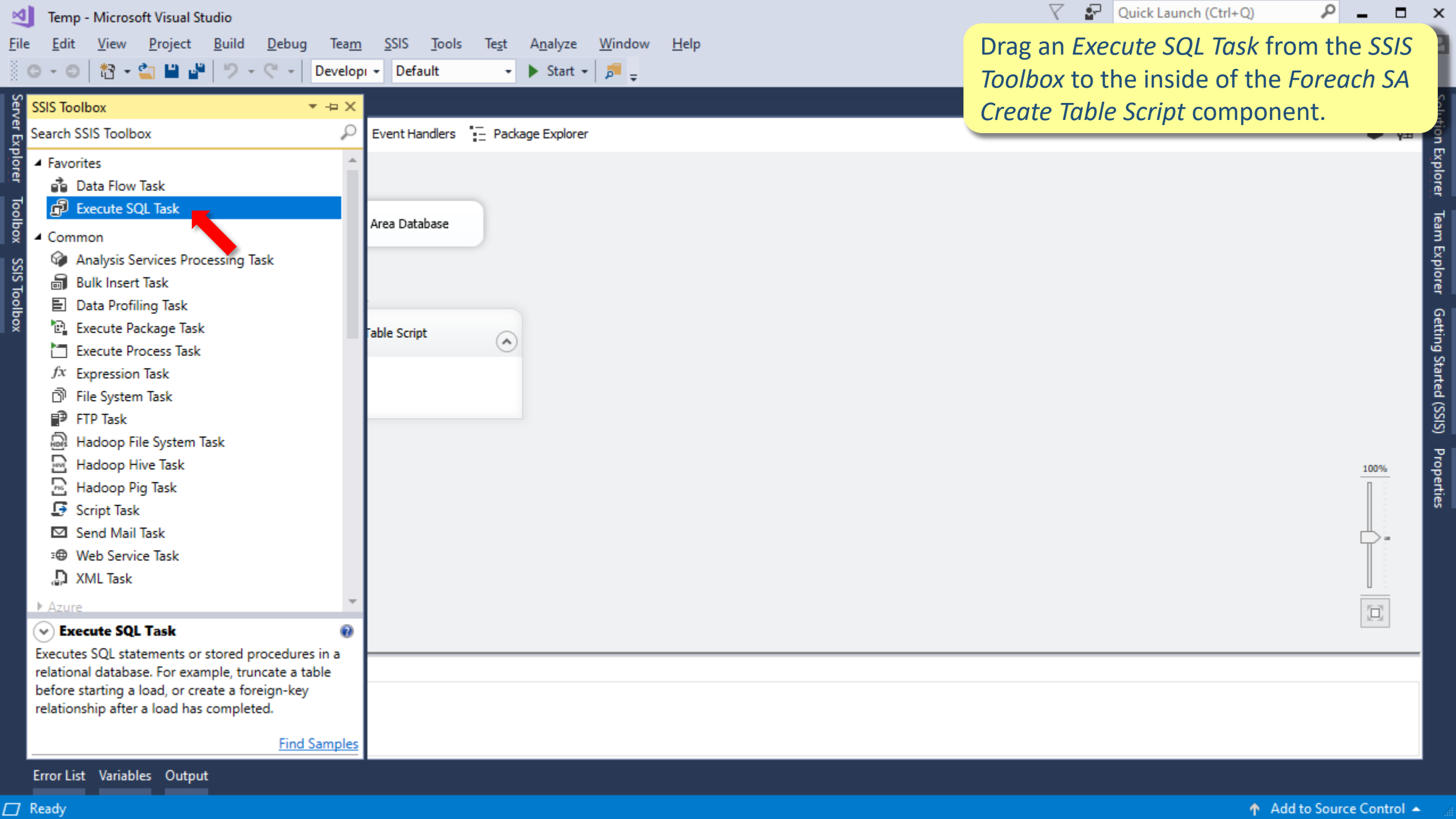
Ready

↑ Add to Source Control

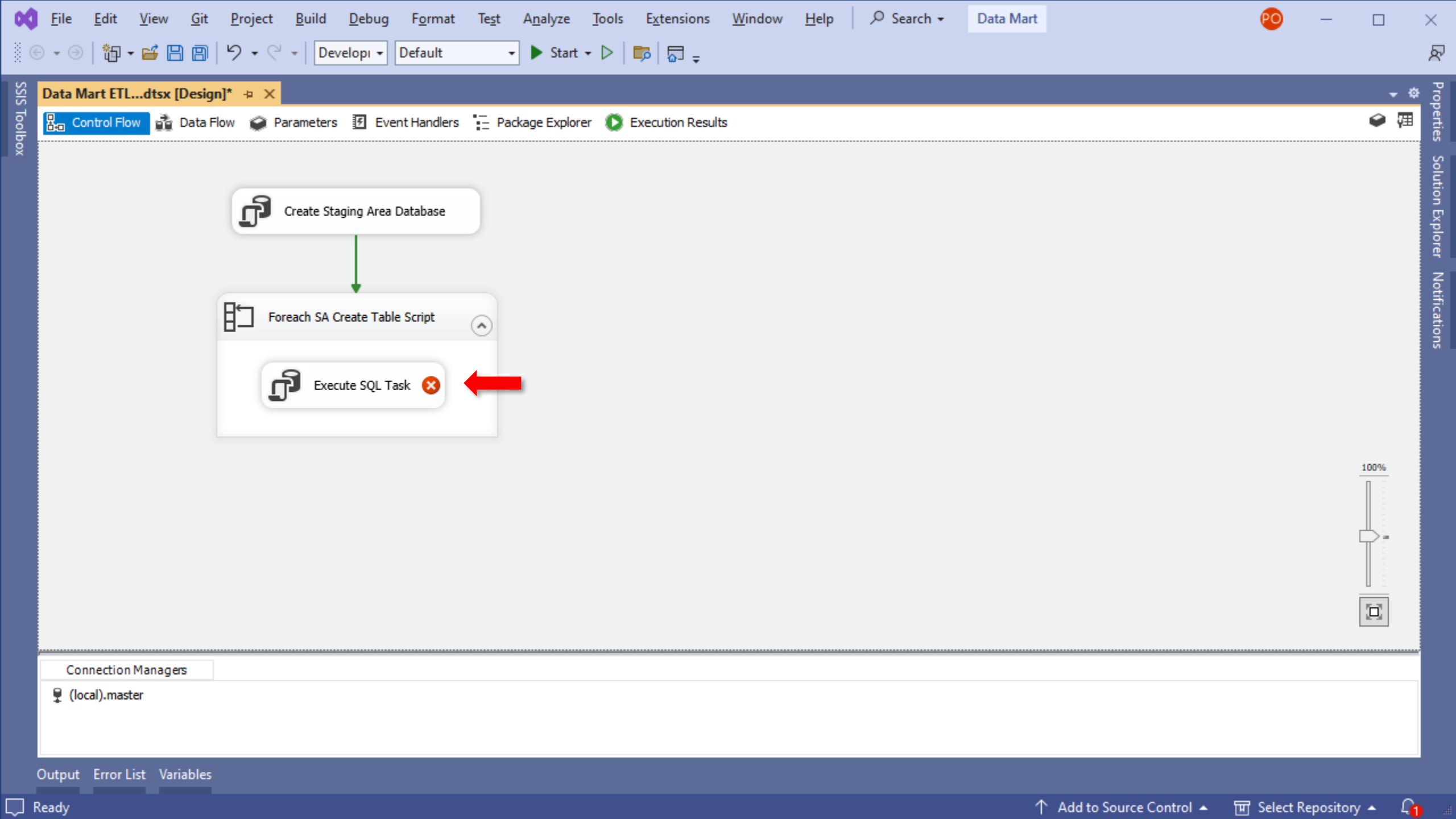
📁 Select Repository

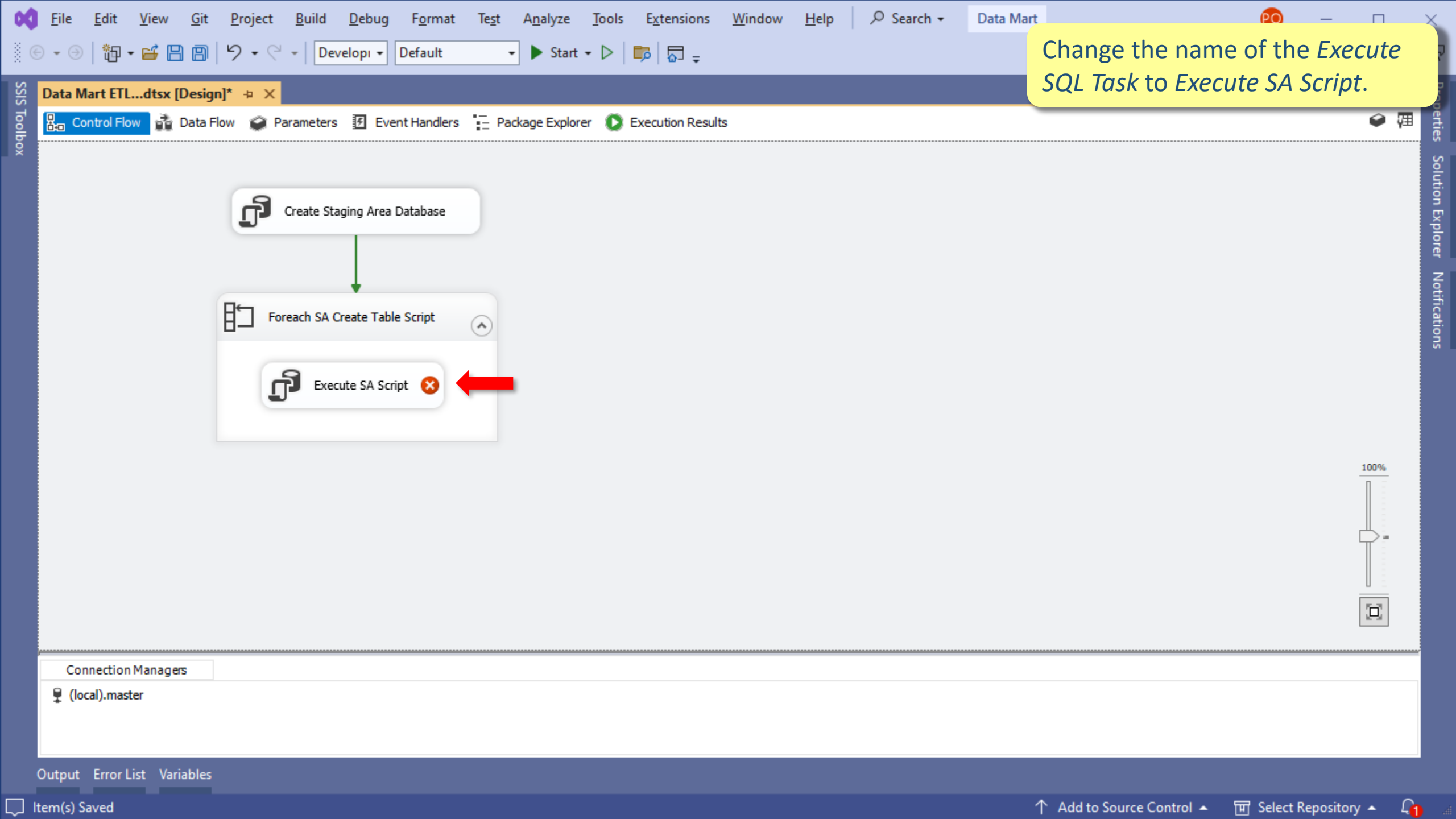
1

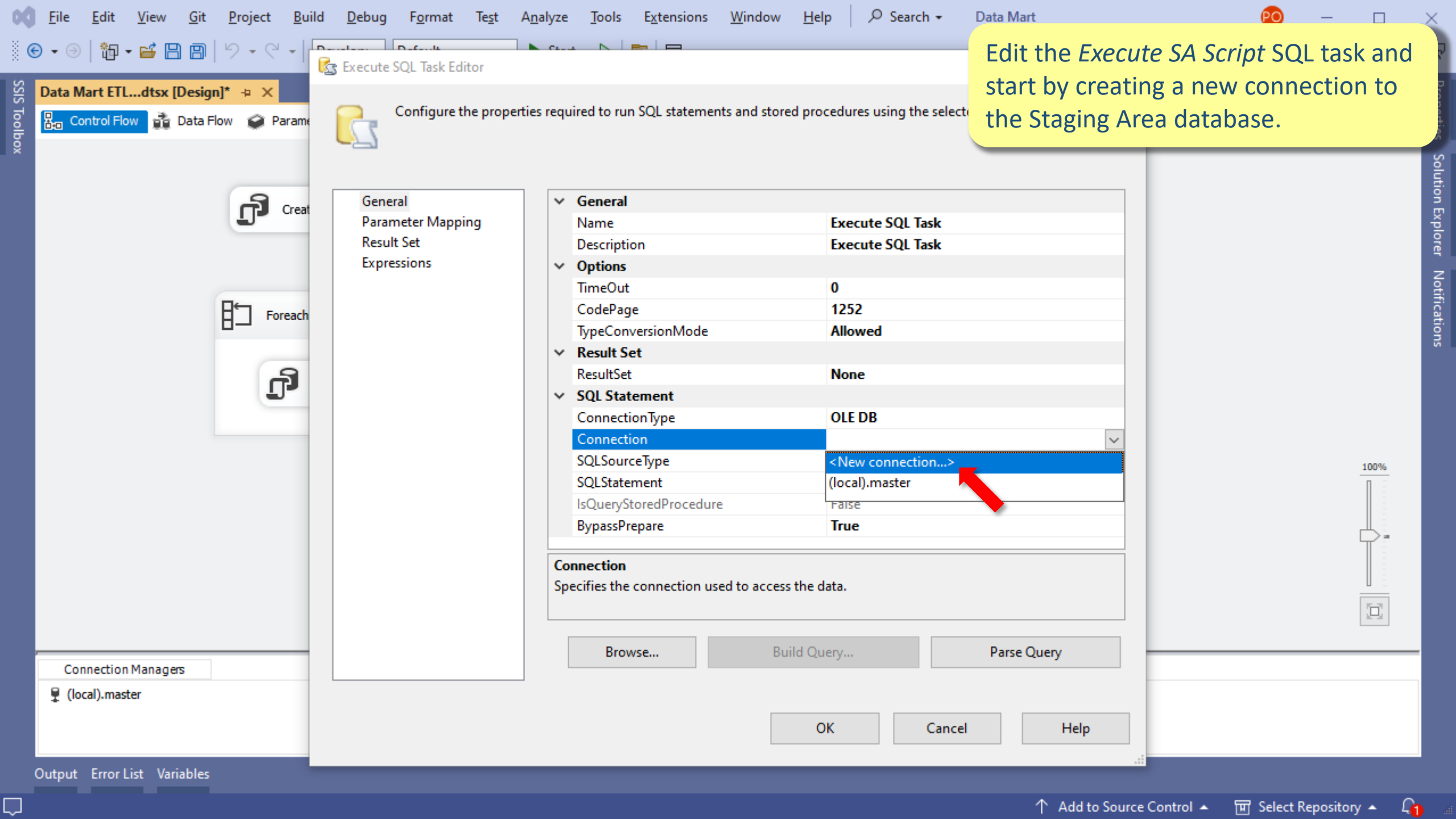




Drag an *Execute SQL Task* from the *SSIS Toolbox* to the inside of the *Foreach SA Create Table Script* component.







Edit the *Execute SA Script* SQL task and start by creating a new connection to the Staging Area database.

- General
- Parameter Mapping
- Result Set
- Expressions

▼ General	
Name	Execute SQL Task
Description	Execute SQL Task
▼ Options	
TimeOut	0
CodePage	1252
TypeConversionMode	Allowed
▼ Result Set	
ResultSet	None
▼ SQL Statement	
ConnectionType	OLE DB
Connection	<New connection...>
SQLSourceType	(local).master
SQLStatement	
IsQueryStoredProcedure	False
BypassPrepare	True

Connection
Specifies the connection used to access the data.

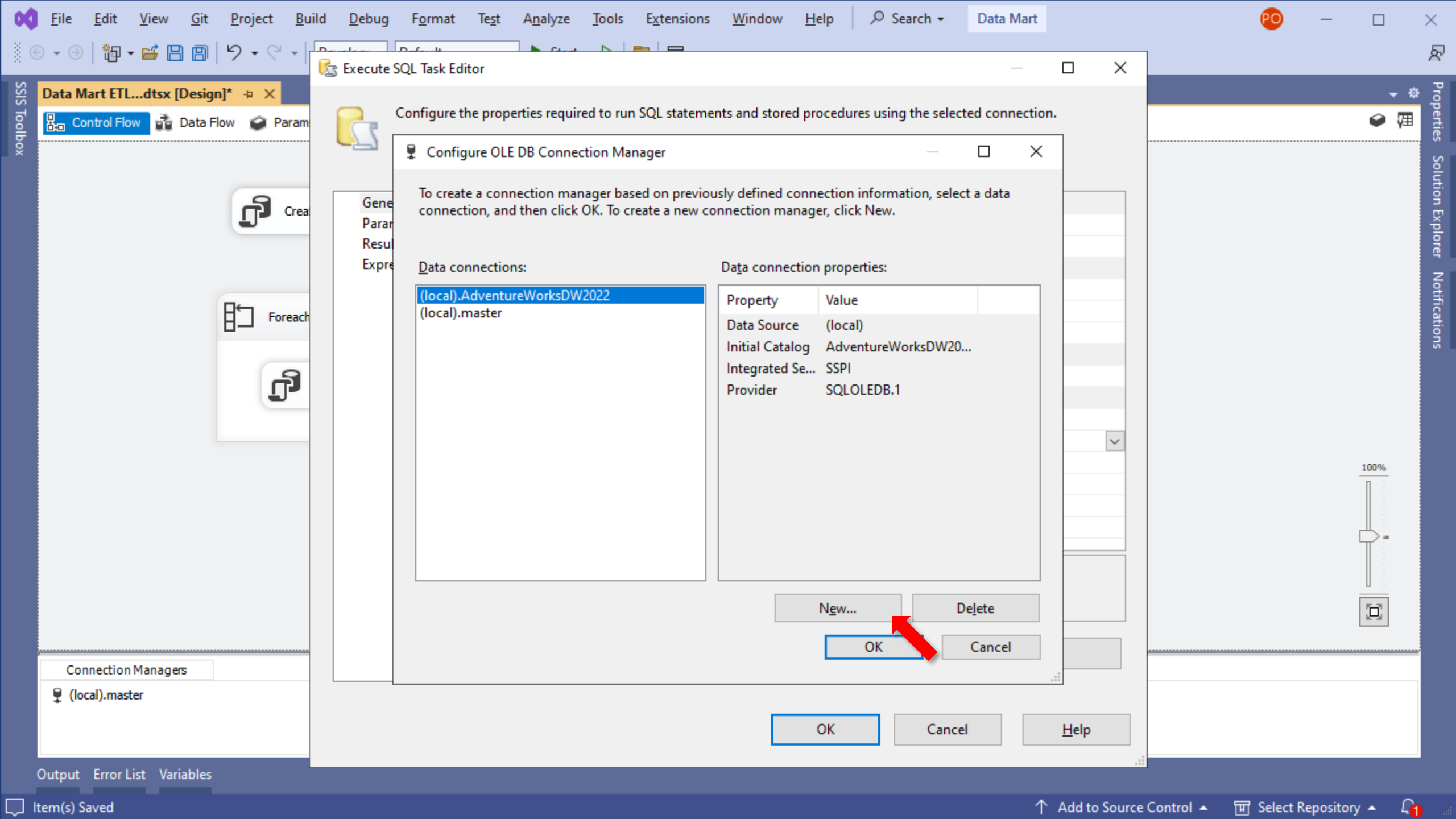
[Browse...](#) [Build Query...](#) [Parse Query](#)

[OK](#) [Cancel](#) [Help](#)

Connection Managers

(local).master

Output Error List Variables



Execute SQL Task Editor

Configure the properties required to run SQL statements and stored procedures using the selected connection.

Configure OLE DB Connection Manager

To create a connection manager based on previously defined connection information, select a data connection, and then click OK. To create a new connection manager, click New.

Data connections:

(local).AdventureWorksDW2022
(local).master

Data connection properties:

Property	Value
Data Source	(local)
Initial Catalog	AdventureWorksDW20...
Integrated Se...	SSPI
Provider	SQLOLEDB.1

New...

Delete

OK

Cancel

OK

Cancel

Help

Connection Managers

(local).master

Output Error List Variables

Item(s) Saved

Add to Source Control

Select Repository

Set the *Server name* to *localhost* (or to the instance name where you installed SQL Server) and select the *BDAMD_StagingArea* database.

Connection Manager

Provider: Native OLE DB\Microsoft OLE DB Driver for SQL Server

OLE DB Provider: Microsoft OLE DB Driver for SQL Server

Enter a server or file name

Server or file name: (local)

Location:

Log on to the server

☒ Use Windows NT Integrated Security

☐ Use a specific user name and password:

User name:

Password:

☐ Blank password

☐ Allow saving password

Initial catalog: BDAMD_StagingArea

Test Connection

OK Cancel Help

Data Mart ETL...dtsx [Design]*

Control Flow Data Flow Parameters

Create

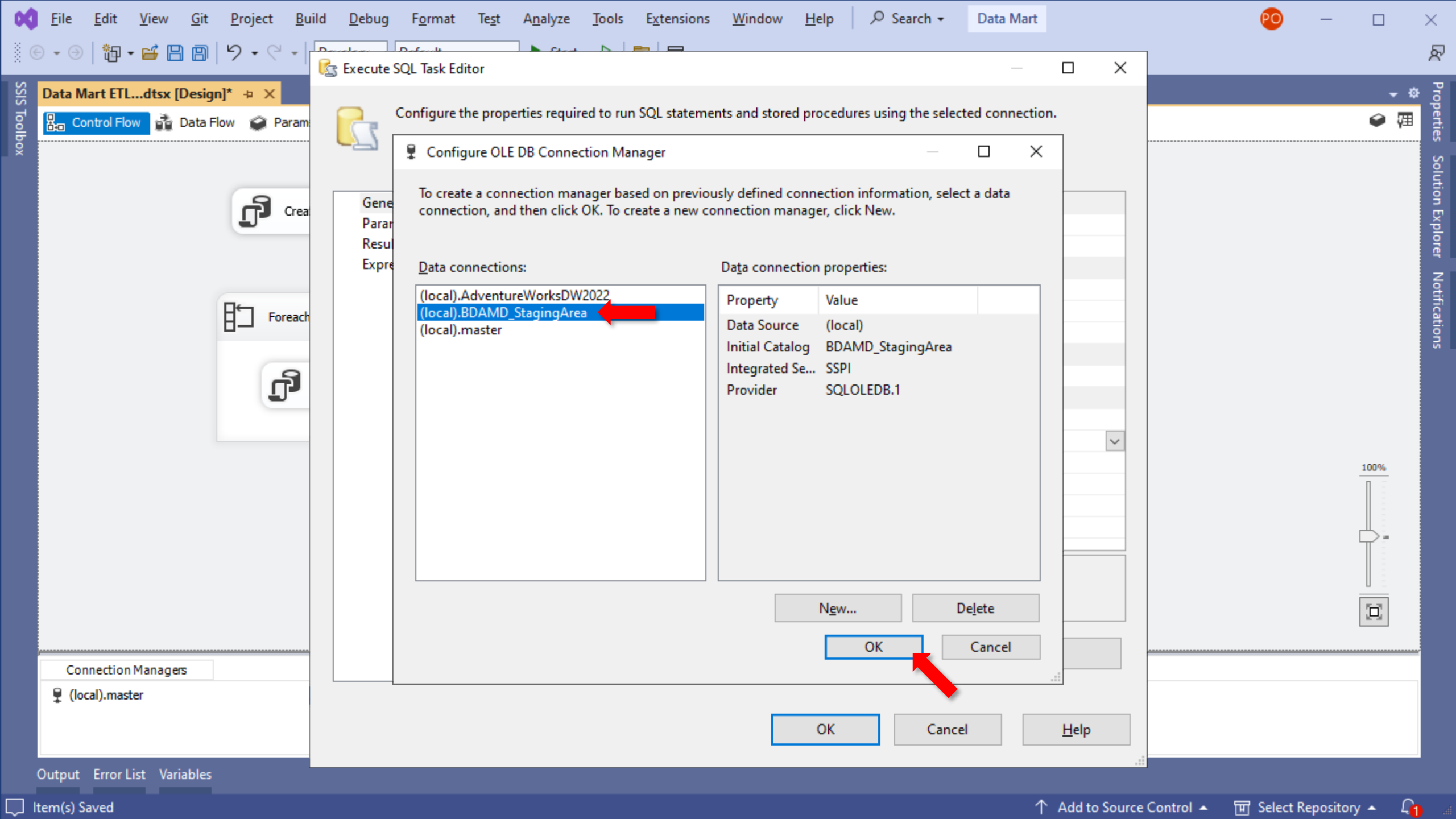
Foreach S

E

Connection Managers

(local).master

Output Error List Variables



Create a *File connection* to the SQL create tables script files of the staging area.

Execute SQL Task Editor

Configure the properties required to run SQL statements and stored procedures using the selected connection.

General

Parameter Mapping

Result Set

Expressions

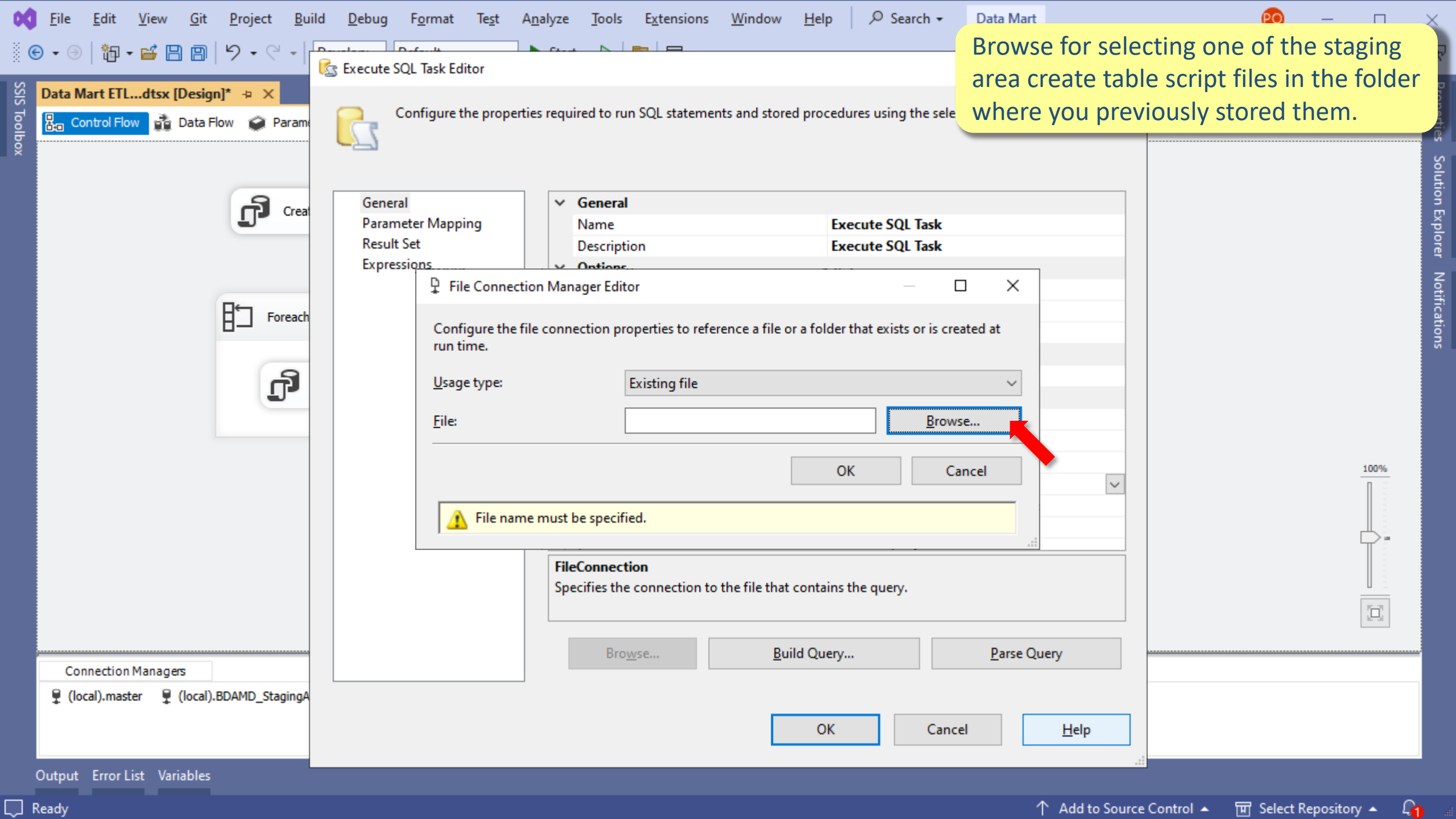
General	
Name	Execute SQL Task
Description	Execute SQL Task
Options	
TimeOut	0
CodePage	1252
TypeConversionMode	Allowed
Result Set	
ResultSet	None
SQL Statement	
ConnectionType	OLE DB
Connection	(local).BDAMD_StagingArea
SQLSourceType	File connection
FileConnection	<New connection...>
IsQueryStoredProcedure	
BypassPrepare	true

FileConnection
Specifies the connection to the file that contains the query.

Browse... Build Query... Parse Query

OK Cancel Help

- Connection Managers
- (local).master
 - (local).BDAMD_StagingArea



Browse for selecting one of the staging area create table script files in the folder where you previously stored them.

File Connection Manager Editor

Configure the file connection properties to reference a file or a folder that exists or is created at run time.

Usage type: Existing file

File: Browse...

OK Cancel

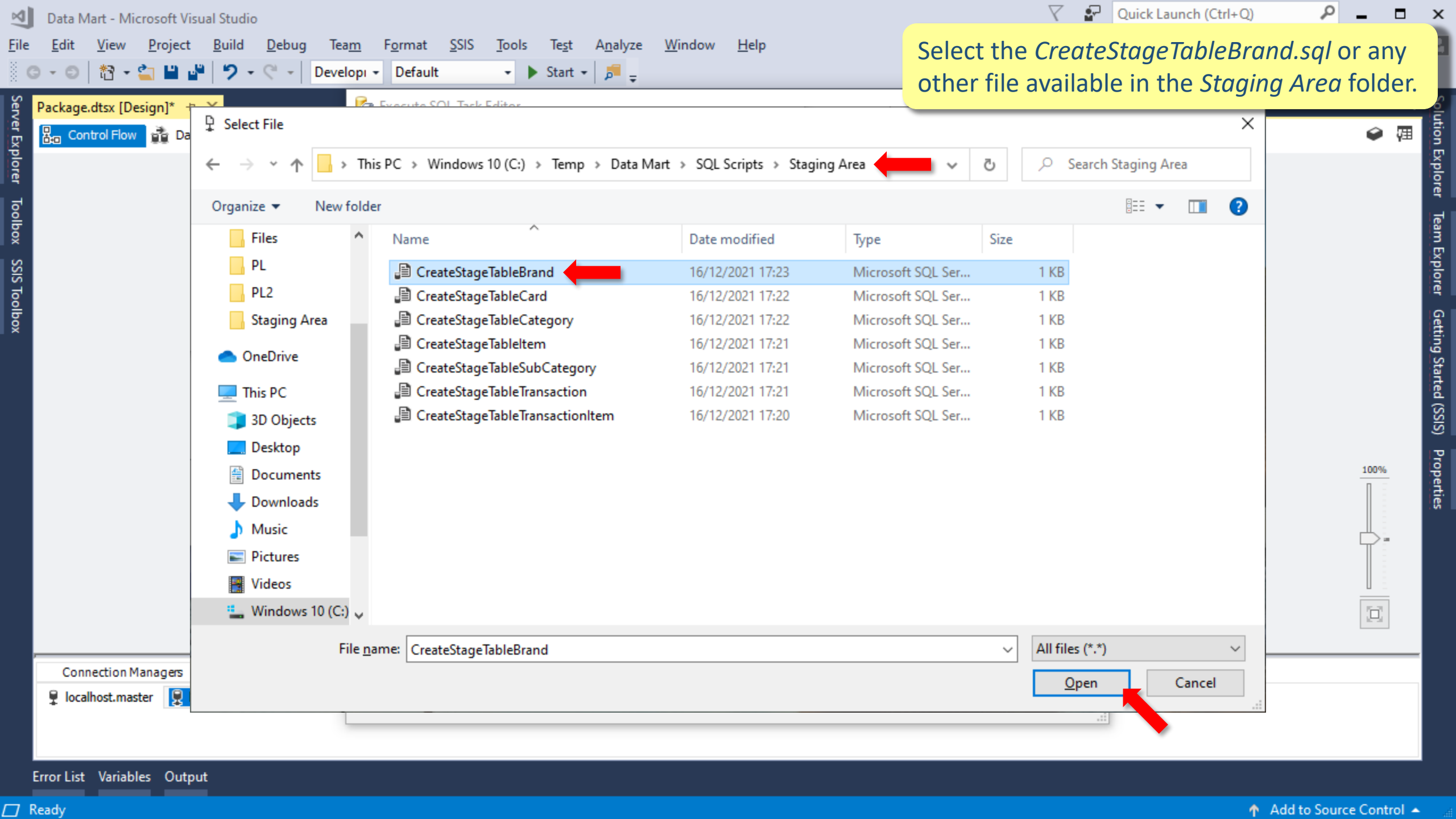
File name must be specified.

FileConnection

Specifies the connection to the file that contains the query.

Browse... Build Query... Parse Query

OK Cancel Help



Select the *CreateStageTableBrand.sql* or any other file available in the *Staging Area* folder.

Select File

← → ↑ ↓ > This PC > Windows 10 (C:) > Temp > Data Mart > SQL Scripts > Staging Area ↻ 🔍 Search Staging Area

Organize ▾ New folder

Name	Date modified	Type	Size
CreateStageTableBrand	16/12/2021 17:23	Microsoft SQL Ser...	1 KB
CreateStageTableCard	16/12/2021 17:22	Microsoft SQL Ser...	1 KB
CreateStageTableCategory	16/12/2021 17:22	Microsoft SQL Ser...	1 KB
CreateStageTableItem	16/12/2021 17:21	Microsoft SQL Ser...	1 KB
CreateStageTableSubCategory	16/12/2021 17:21	Microsoft SQL Ser...	1 KB
CreateStageTableTransaction	16/12/2021 17:21	Microsoft SQL Ser...	1 KB
CreateStageTableTransactionItem	16/12/2021 17:20	Microsoft SQL Ser...	1 KB

File name: CreateStageTableBrand

All files (*.*)

Open Cancel

Select the *CreateStageTableBrand.sql* or any other file available in the *Staging Area* folder.

Execute SQL Task Editor

Configure the properties required to run SQL statements and stored procedures using the selected connection.

General

Parameter Mapping
Result Set
Expressions

General

Name	Execute SQL Task
Description	Execute SQL Task
Options	
TimeOut	0

File Connection Manager Editor

Configure the file connection properties to reference a file or a folder that exists or is created at run time.

Usage type:

Existing file

File:

C:\Temp\Data Mart\SQL Scripts\Staging Area\CreateStageTableBrand.sql

Browse...

OK

Cancel

FileConnection

Specifies the connection to the file that contains the query.

Browse...

Build Query...

Parse Query

OK

Cancel

Help

Connection Managers

(local).master (local).BDAMD_StagingA

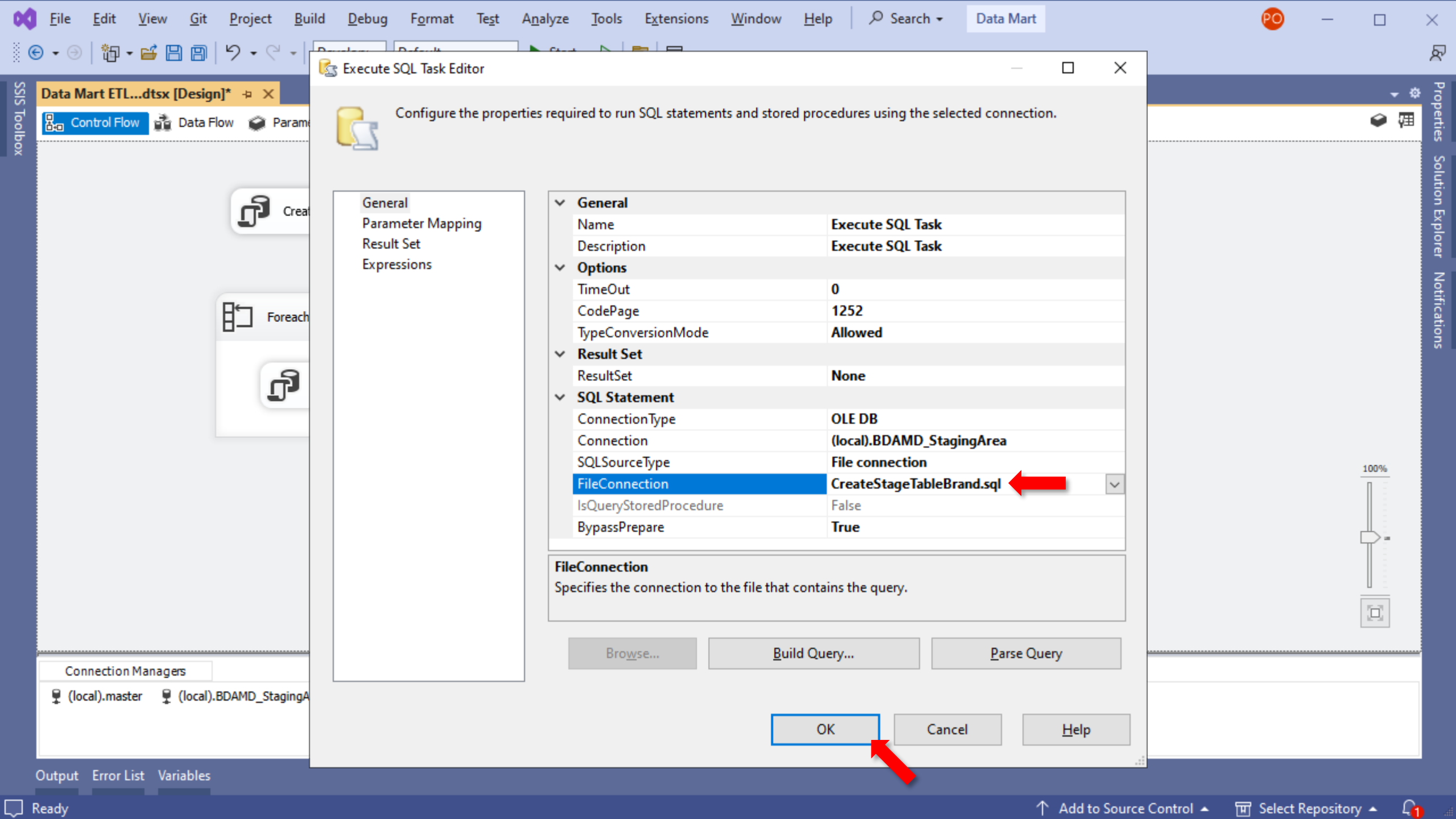
Output Error List Variables

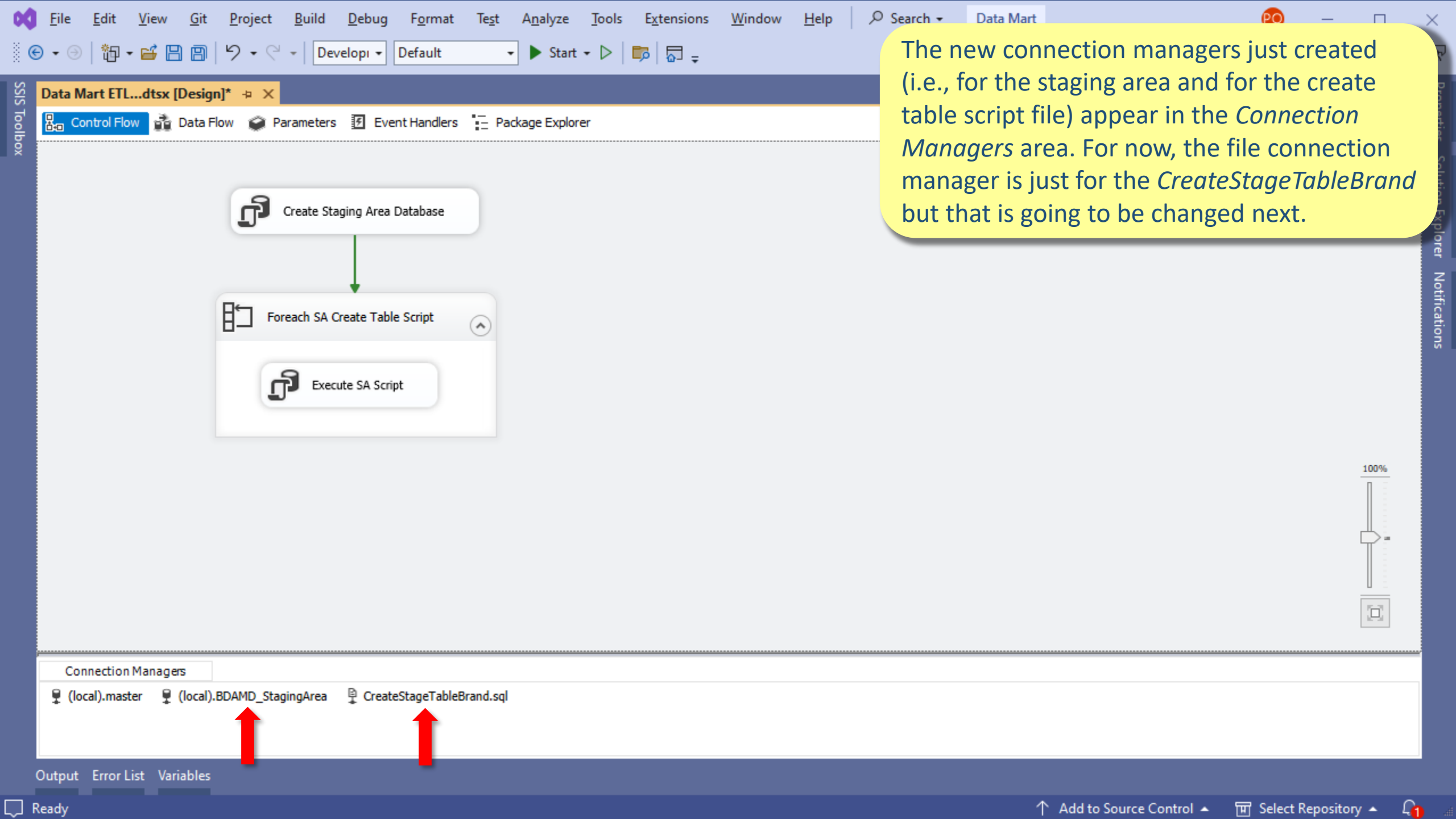
Ready

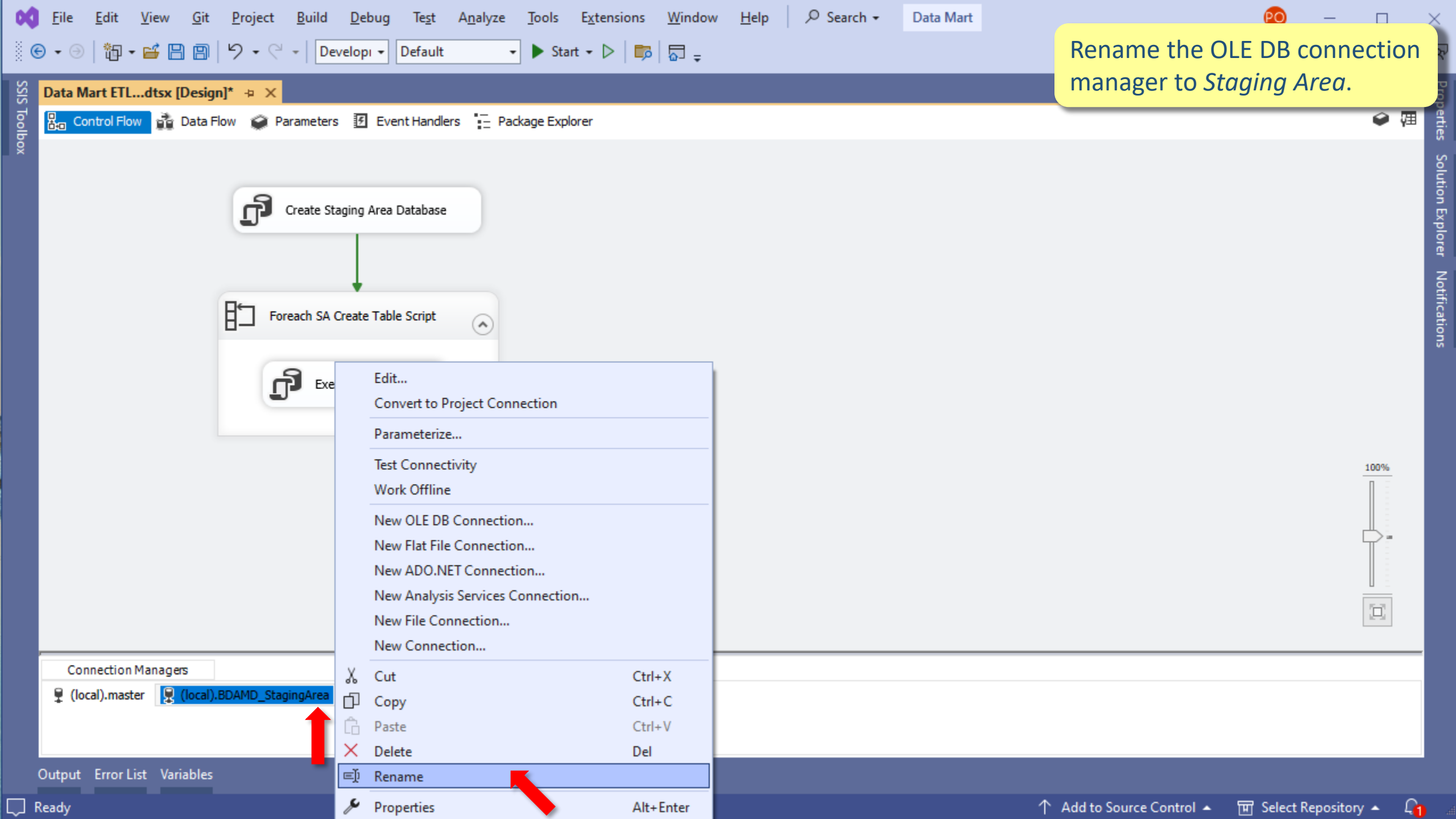
Add to Source Control

Select Repository

1



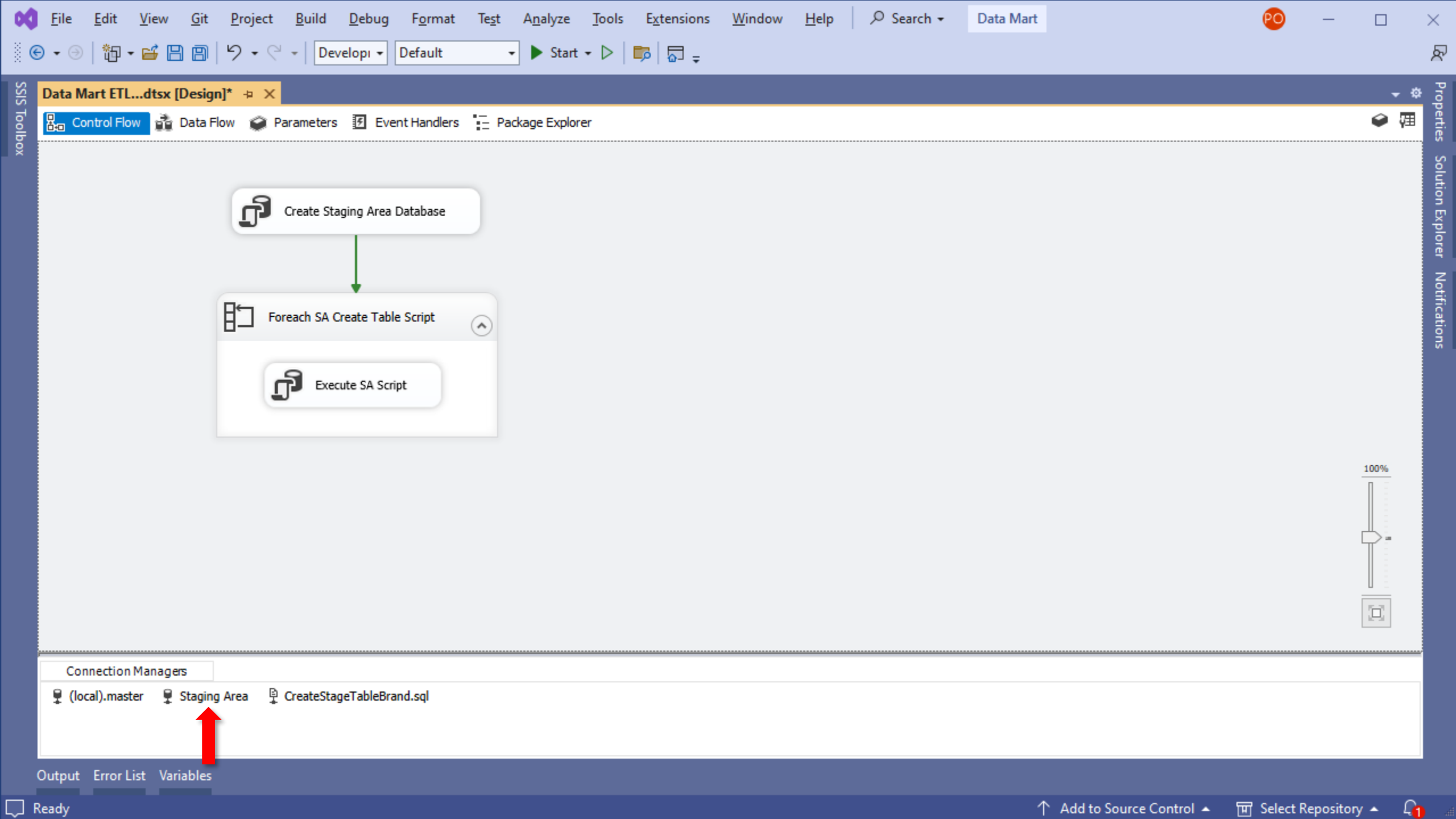


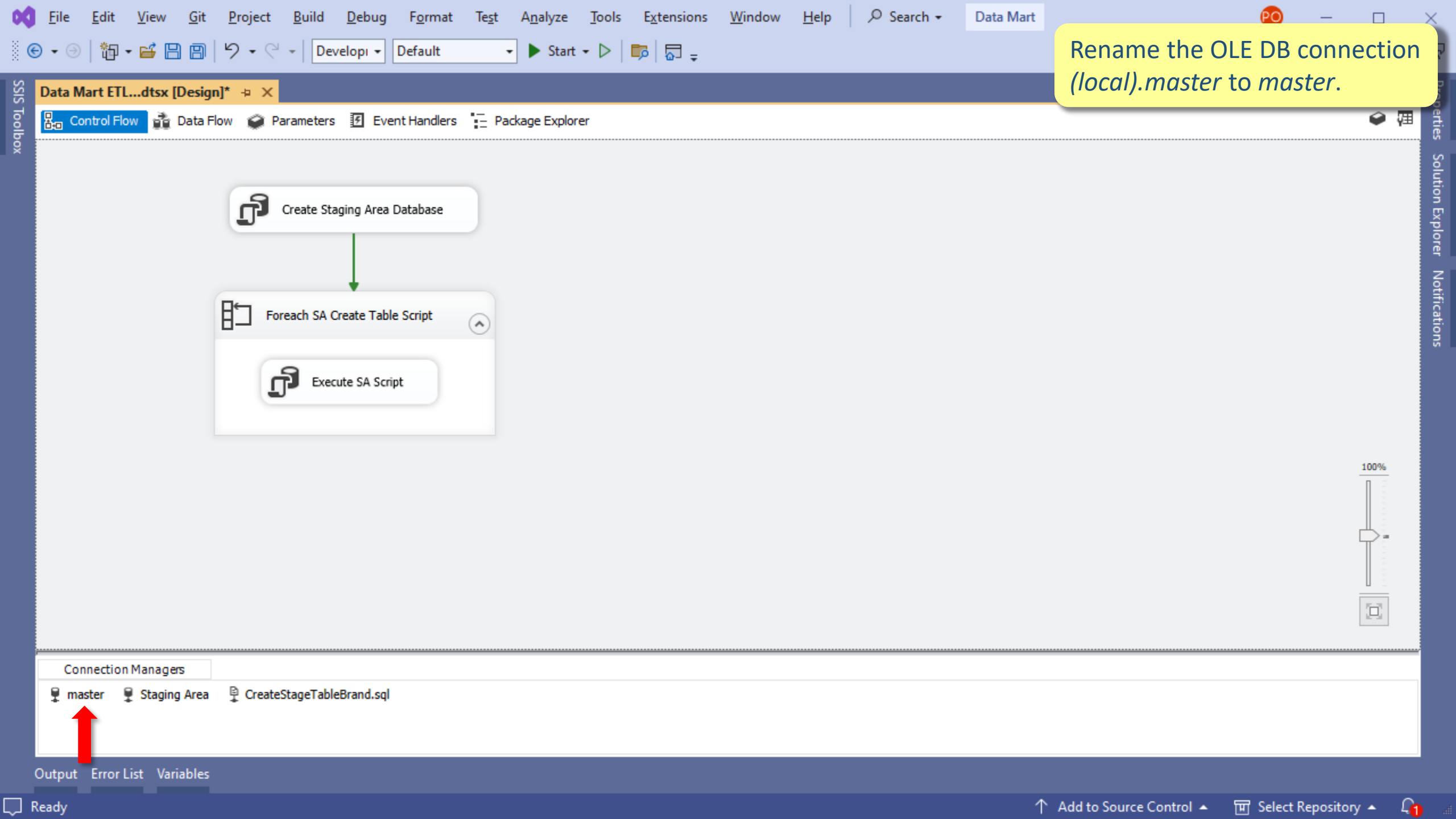


Rename the OLE DB connection manager to *Staging Area*.

- Edit...
- Convert to Project Connection
- Parameterize...
- Test Connectivity
- Work Offline
- New OLE DB Connection...
- New Flat File Connection...
- New ADO.NET Connection...
- New Analysis Services Connection...
- New File Connection...
- New Connection...
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Delete Del
- Rename**
- Properties Alt+Enter







Rename the OLE DB connection *(local).master* to *master*.

Data Mart ETL...dtsx [Design]*

Control Flow Data Flow Parameters Event Handlers Package Explorer

Create Staging Area Database

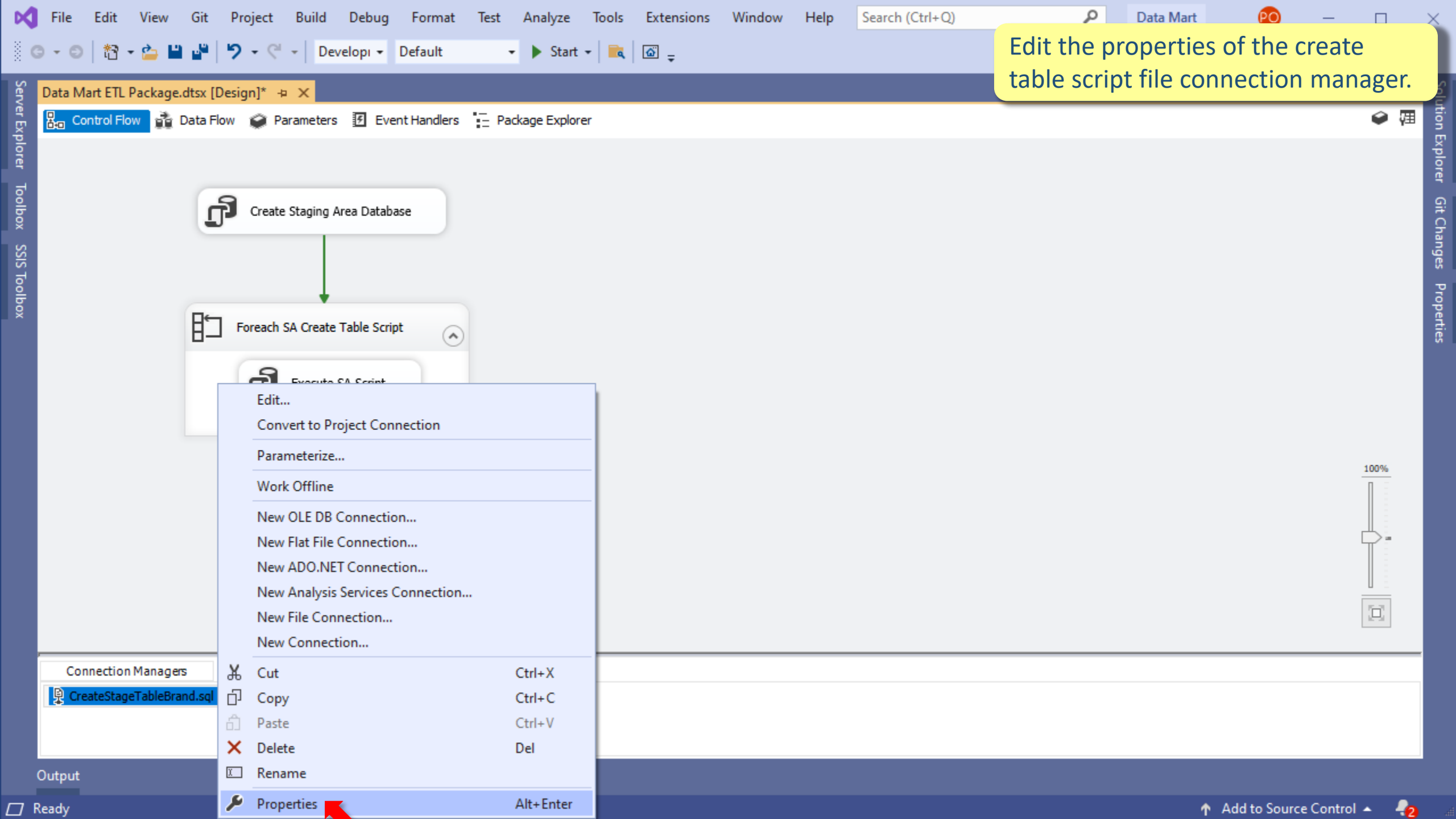
Foreach SA Create Table Script

Execute SA Script

Connection Managers

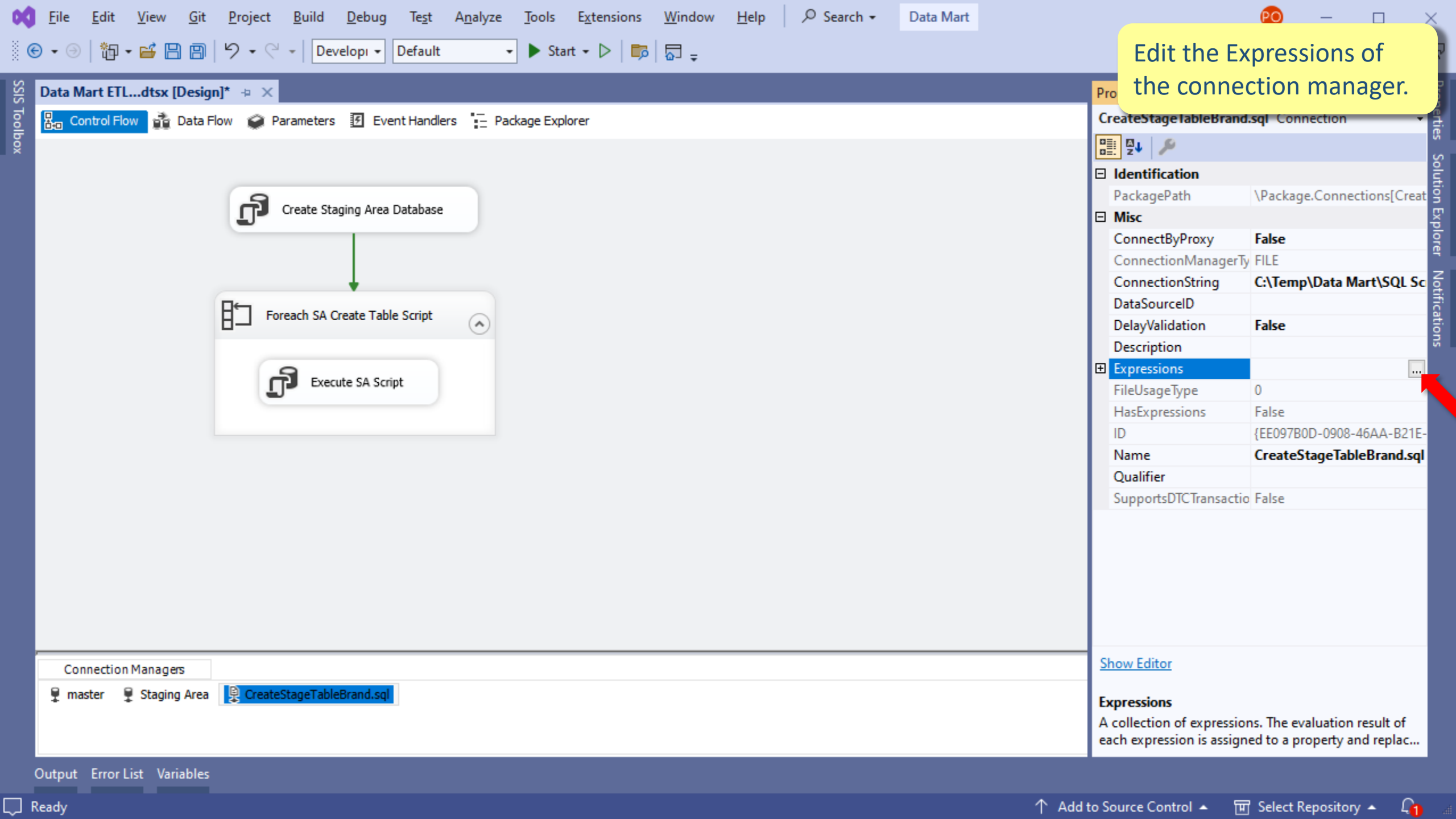
master Staging Area CreateStageTableBrand.sql

Output Error List Variables



Edit the properties of the create table script file connection manager.

- Edit...
- Convert to Project Connection
- Parameterize...
- Work Offline
- New OLE DB Connection...
- New Flat File Connection...
- New ADO.NET Connection...
- New Analysis Services Connection...
- New File Connection...
- New Connection...
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Delete (Del)
- Rename
- Properties (Alt+Enter)**



Edit the Expressions of the connection manager.

Pro

CreateStageTableBrand.sql Connection

Identification

PackagePath	\Package.Connections[Creat
-------------	----------------------------

Misc

ConnectByProxy	False
ConnectionManagerTy	FILE
ConnectionString	C:\Temp\Data Mart\SQL Sc
DataSourceID	
DelayValidation	False
Description	

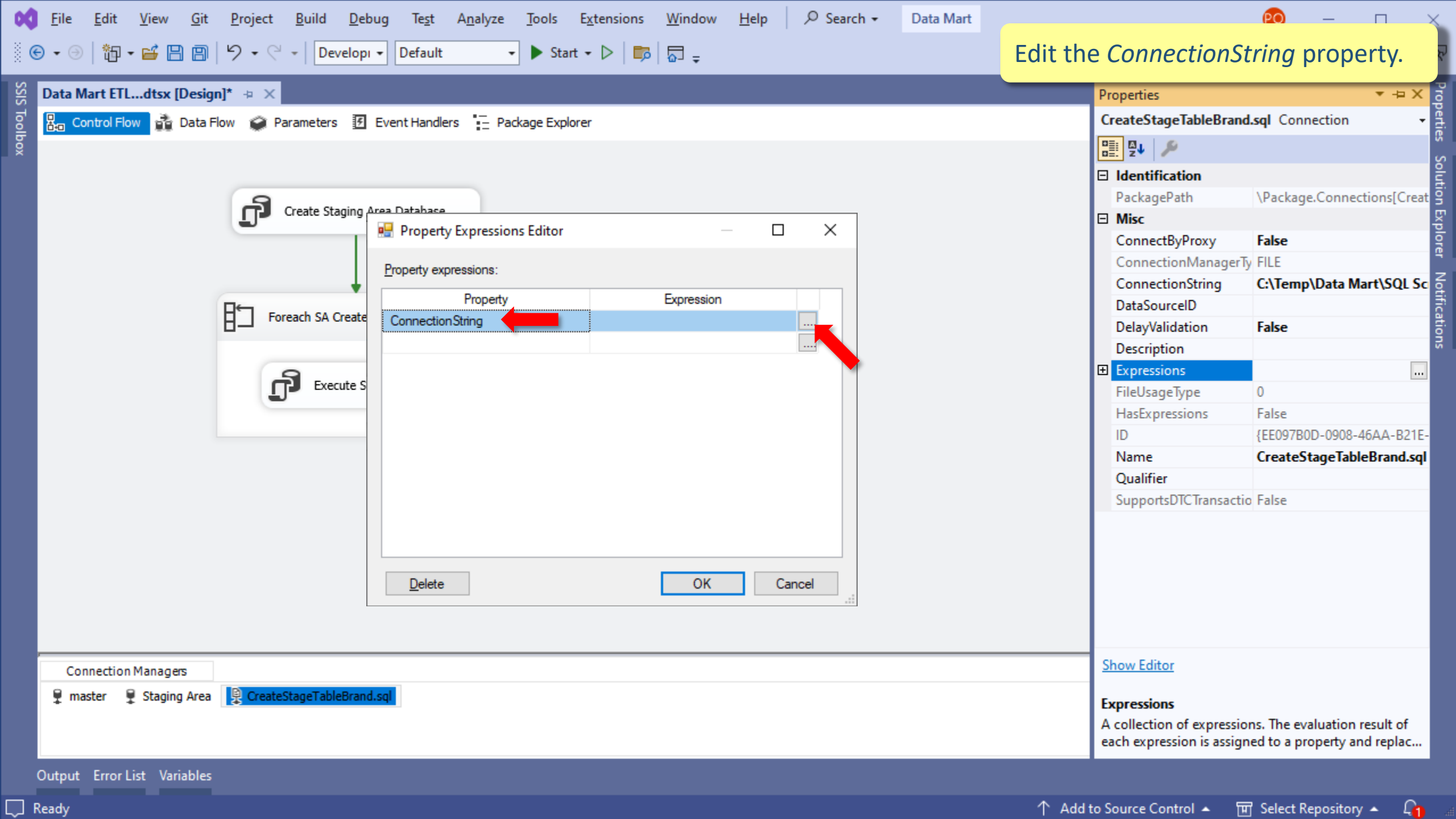
Expressions

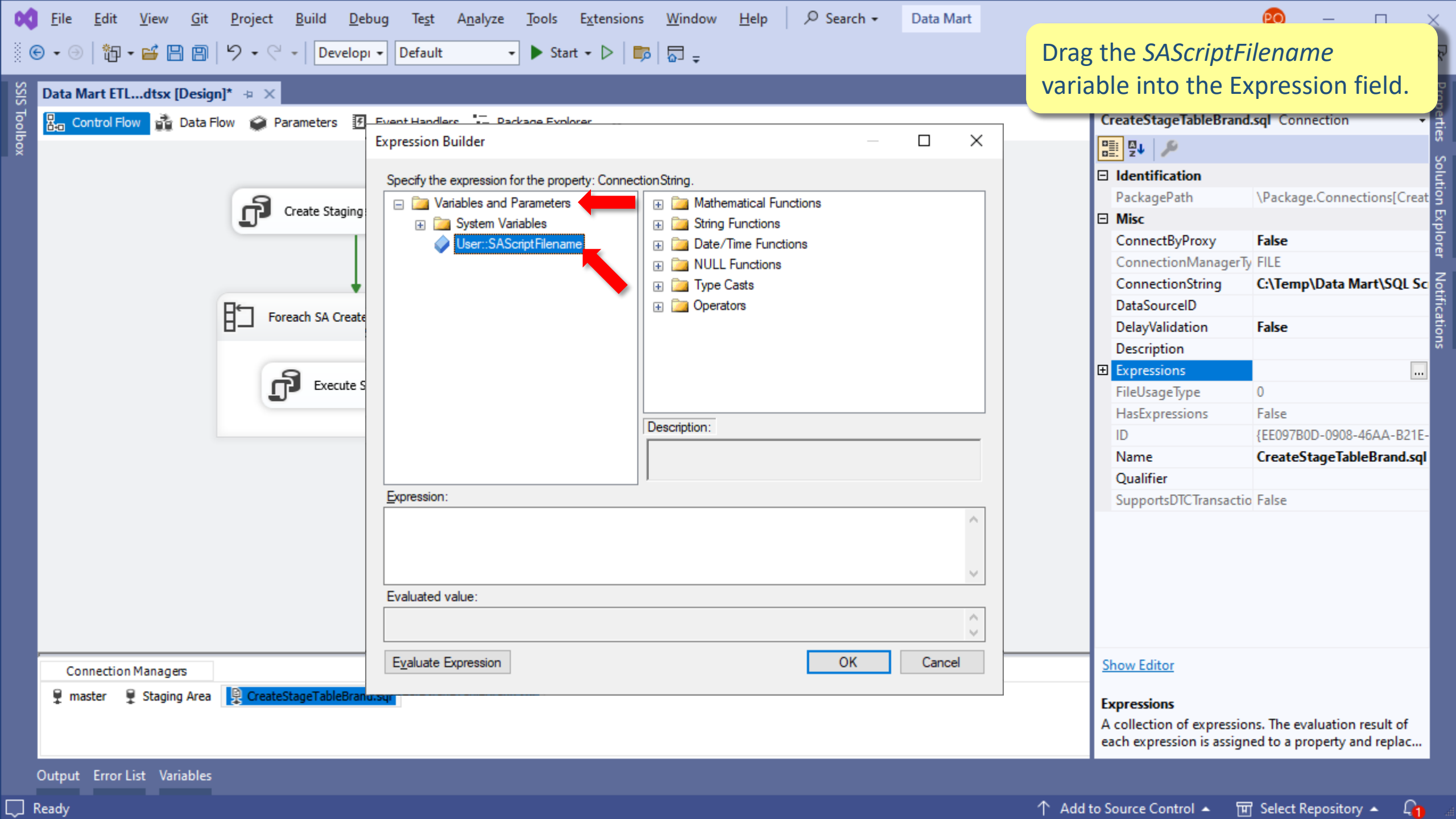
FileUsageType	0
HasExpressions	False
ID	{EE097B0D-0908-46AA-B21E-
Name	CreateStageTableBrand.sql
Qualifier	
SupportsDTCTransactio	False

Show Editor

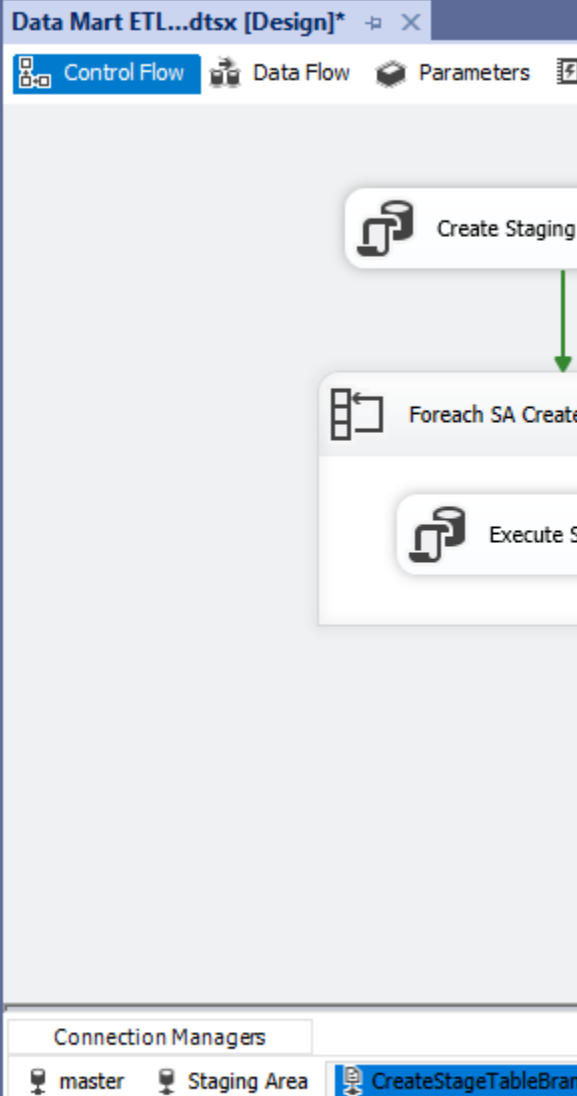
Expressions

A collection of expressions. The evaluation result of each expression is assigned to a property and replac...





Drag the *SAScriptFilename* variable into the Expression field.



Expression Builder

Specify the expression for the property: ConnectionString.

Variables and Parameters

System Variables

User::SAScriptFilename

Mathematical Functions

String Functions

Date/Time Functions

NULL Functions

Type Casts

Operators

Description:

Expression:

Evaluated value:

Evaluate Expression OK Cancel

CreateStageTableBrand.sql Connection

Identification

PackagePath	\Package.Connections[CreateStageTableBrand.sql]
-------------	---

Misc

ConnectByProxy	False
ConnectionManagerType	FILE
ConnectionString	C:\Temp\Data Mart\SQL Scripts\CreateStageTableBrand.sql
DataSourceID	
DelayValidation	False
Description	

Expressions

FileUsageType	0
HasExpressions	False
ID	{EE097B0D-0908-46AA-B21E-...}
Name	CreateStageTableBrand.sql
Qualifier	
SupportsDTCTransaction	False

[Show Editor](#)

Expressions

A collection of expressions. The evaluation result of each expression is assigned to a property and replaces the value of the property.

FileEditViewGitProjectBuildDebugTestAnalyzeToolsExtensionsWindowHelp

SearchData Mart

SSIS Toolbox

Data Mart ETL...dtsx [Design]*

Control FlowData FlowParametersEvent HandlersPackage Explorer

Create Staging

Foreach SA Create

Execute S

Expression Builder

Specify the expression for the property: ConnectionString.

Variables and Parameters

System Variables

User::SAScriptFilename

Mathematical Functions

String Functions

Date/Time Functions

NULL Functions

Type Casts

Operators

Description:

Expression:

@(User::SAScriptFilename)

Evaluated value:

Evaluate ExpressionOKCancel

Connection Managers

masterStaging AreaCreateStageTableBrand.sql

OutputError ListVariables

Ready

Properties

CreateStageTableBrand.sql Connection

Identification

Misc

Expressions

PackagePath

ConnectByProxy

ConnectionManagerTy

ConnectionString

DataSourceID

DelayValidation

Description

FileUsageType

HasExpressions

ID

Name

Qualifier

SupportsDTCTransactio

\\Package.Connections[Creat

False

FILE

C:\Temp\Data Mart\SQL Sc

False

0

False

{EE097B0D-0908-46AA-B21E-

CreateStageTableBrand.sql

False

Show Editor

Expressions

A collection of expressions. The evaluation result of each expression is assigned to a property and replac...

PO

—

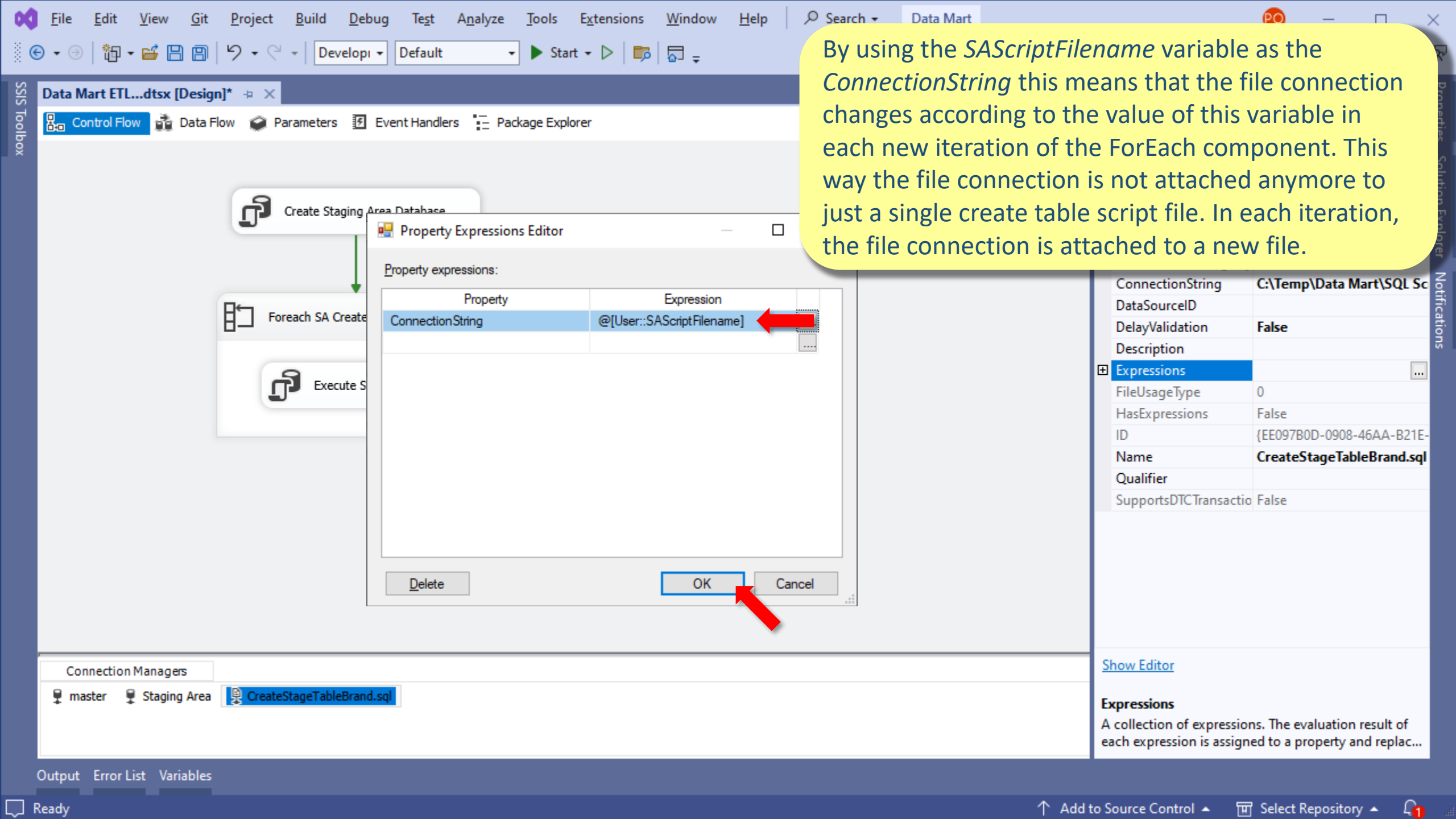
□

×

↑ Add to Source Control

⌵ Select Repository

1



By using the *SAScriptFilename* variable as the *ConnectionString* this means that the file connection changes according to the value of this variable in each new iteration of the ForEach component. This way the file connection is not attached anymore to just a single create table script file. In each iteration, the file connection is attached to a new file.

Property Expressions Editor

Property expressions:

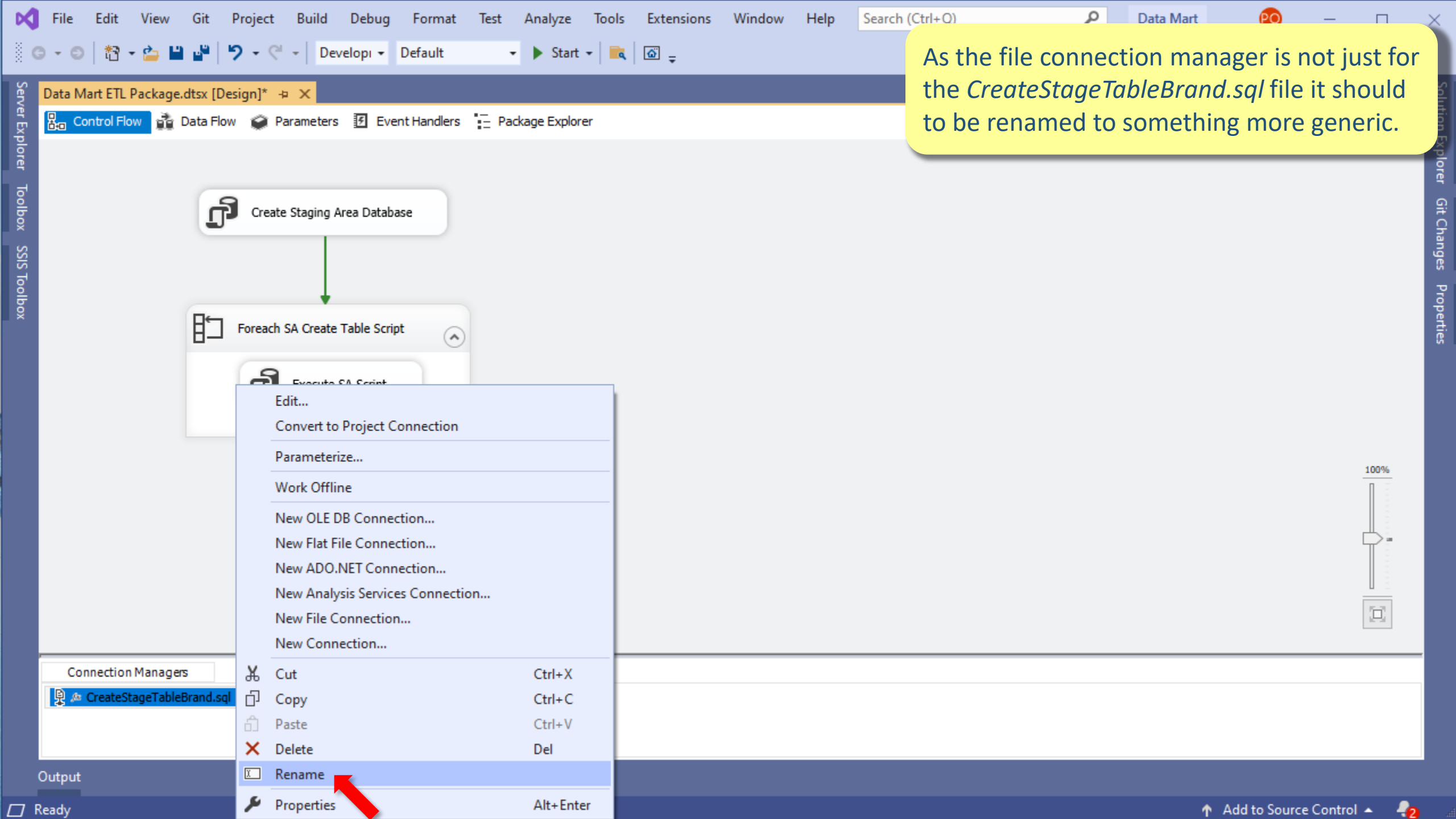
Property	Expression
ConnectionString	@[User::SAScriptFilename]

Delete OK Cancel

ConnectionString	C:\Temp\Data Mart\SQL Sc
DataSourceID	
DelayValidation	False
Description	
Expressions	
FileUsageType	0
HasExpressions	False
ID	{EE097B0D-0908-46AA-B21E-
Name	CreateStageTableBrand.sql
Qualifier	
SupportsDTCTransactio	False

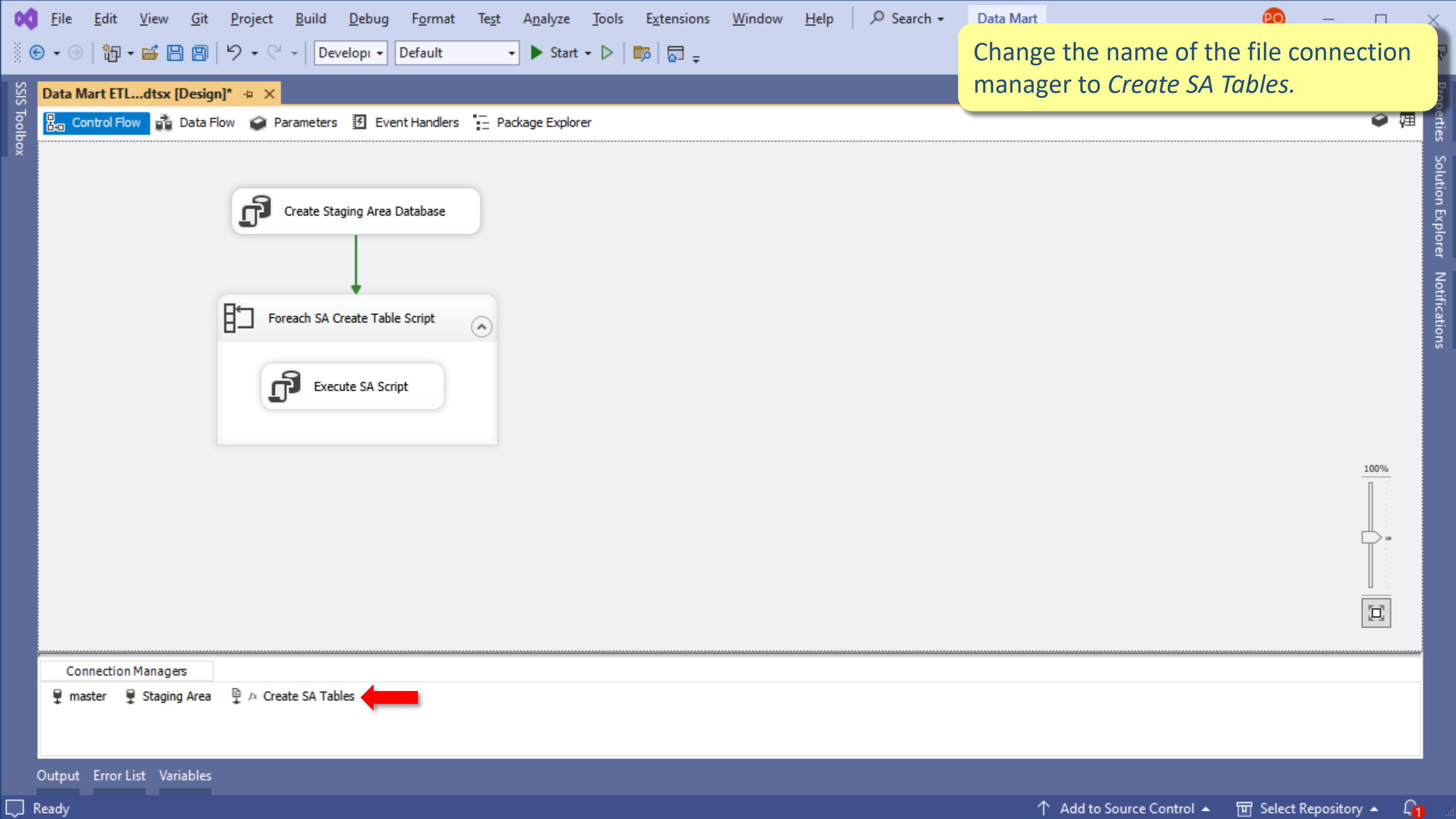
Show Editor

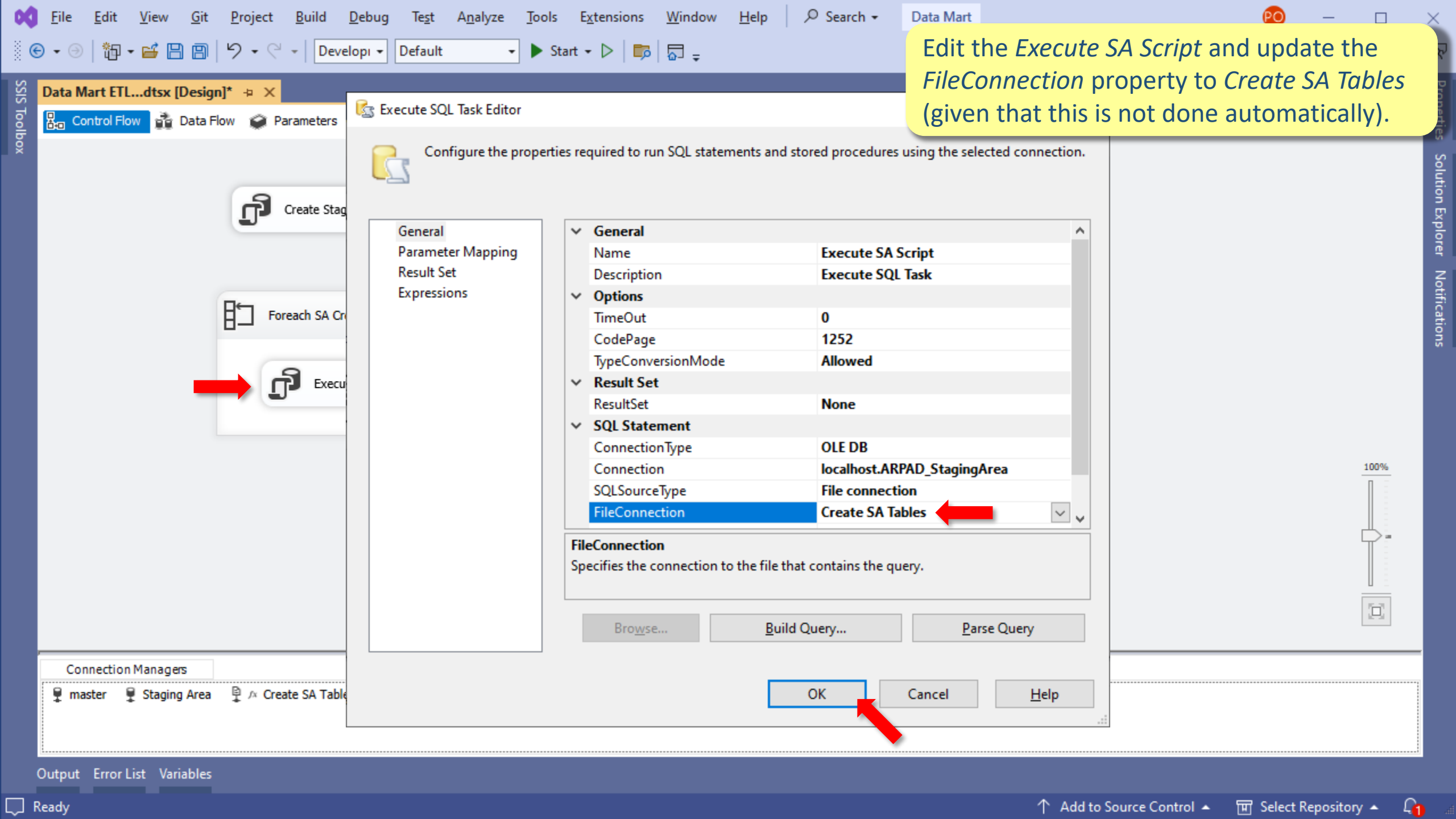
Expressions
A collection of expressions. The evaluation result of each expression is assigned to a property and replac...

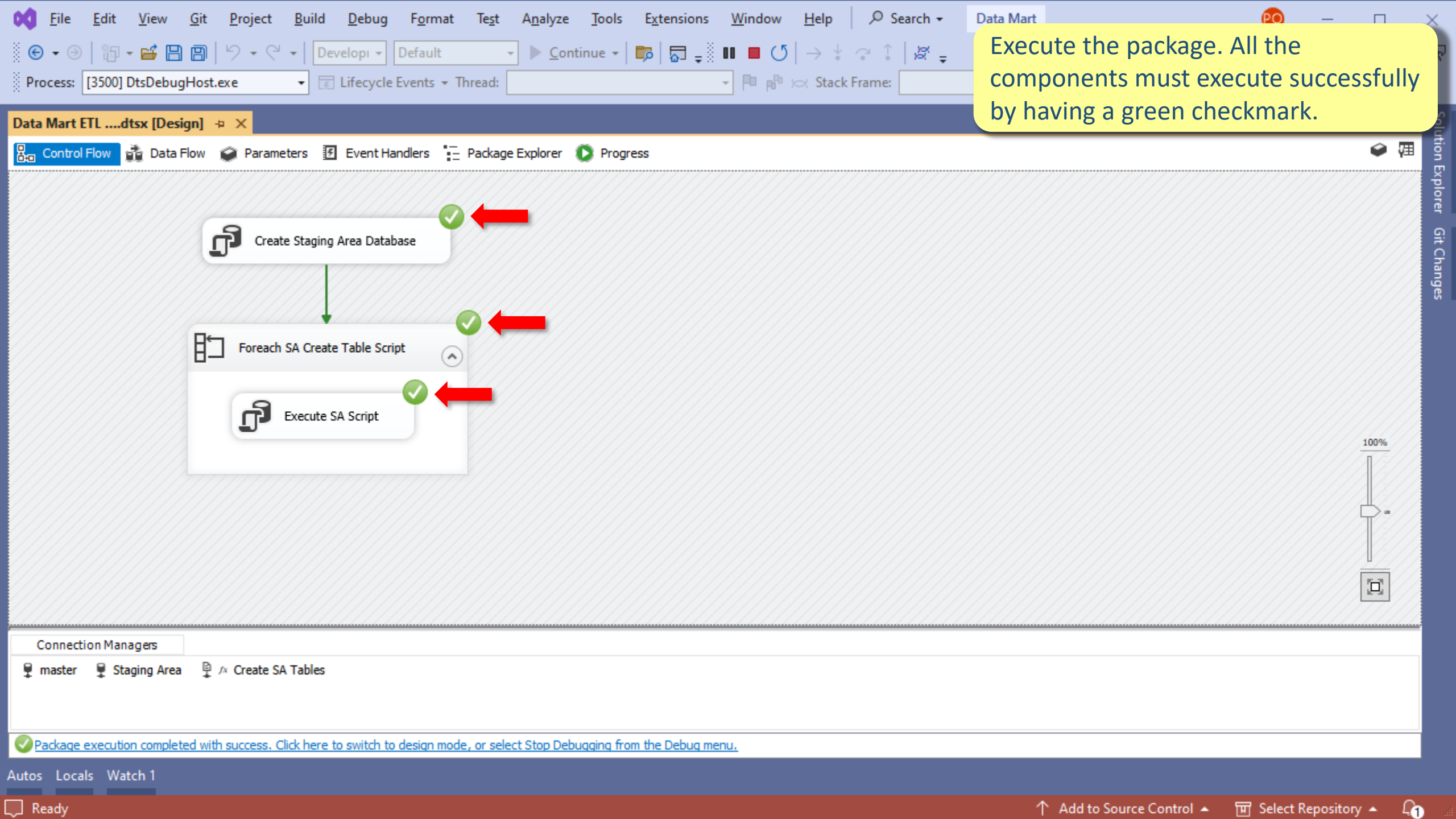


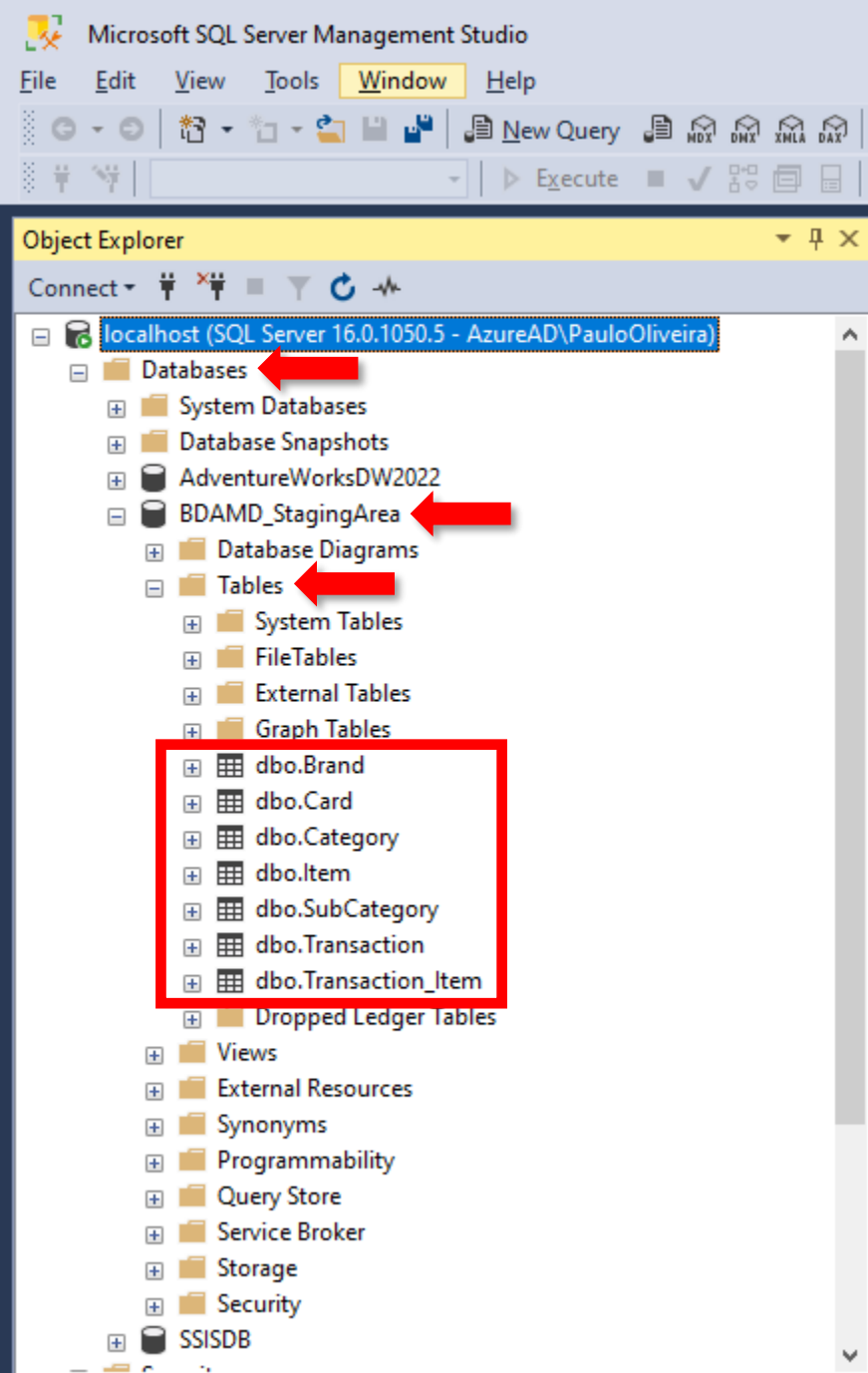
As the file connection manager is not just for the *CreateStageTableBrand.sql* file it should to be renamed to something more generic.

- Edit...
- Convert to Project Connection
- Parameterize...
- Work Offline
- New OLE DB Connection...
- New Flat File Connection...
- New ADO.NET Connection...
- New Analysis Services Connection...
- New File Connection...
- New Connection...
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Delete (Del)
- Rename**
- Properties (Alt+Enter)

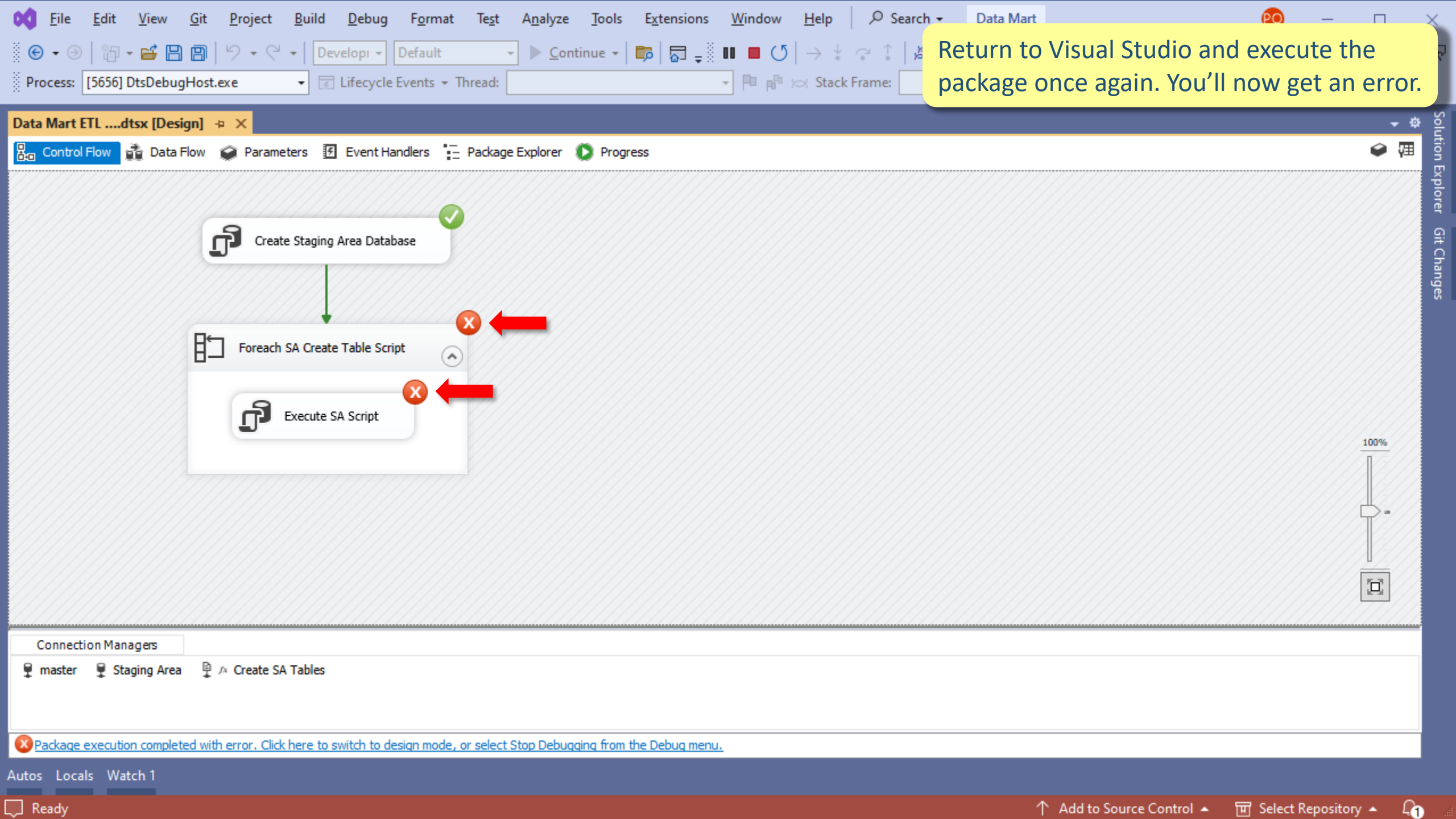




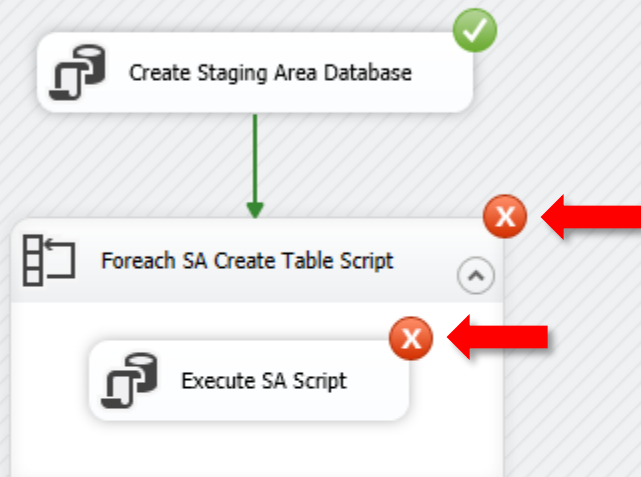




Using SQL Server Management Studio it is possible to check that the Staging Area tables were successfully created. You may have to refresh the database tables.

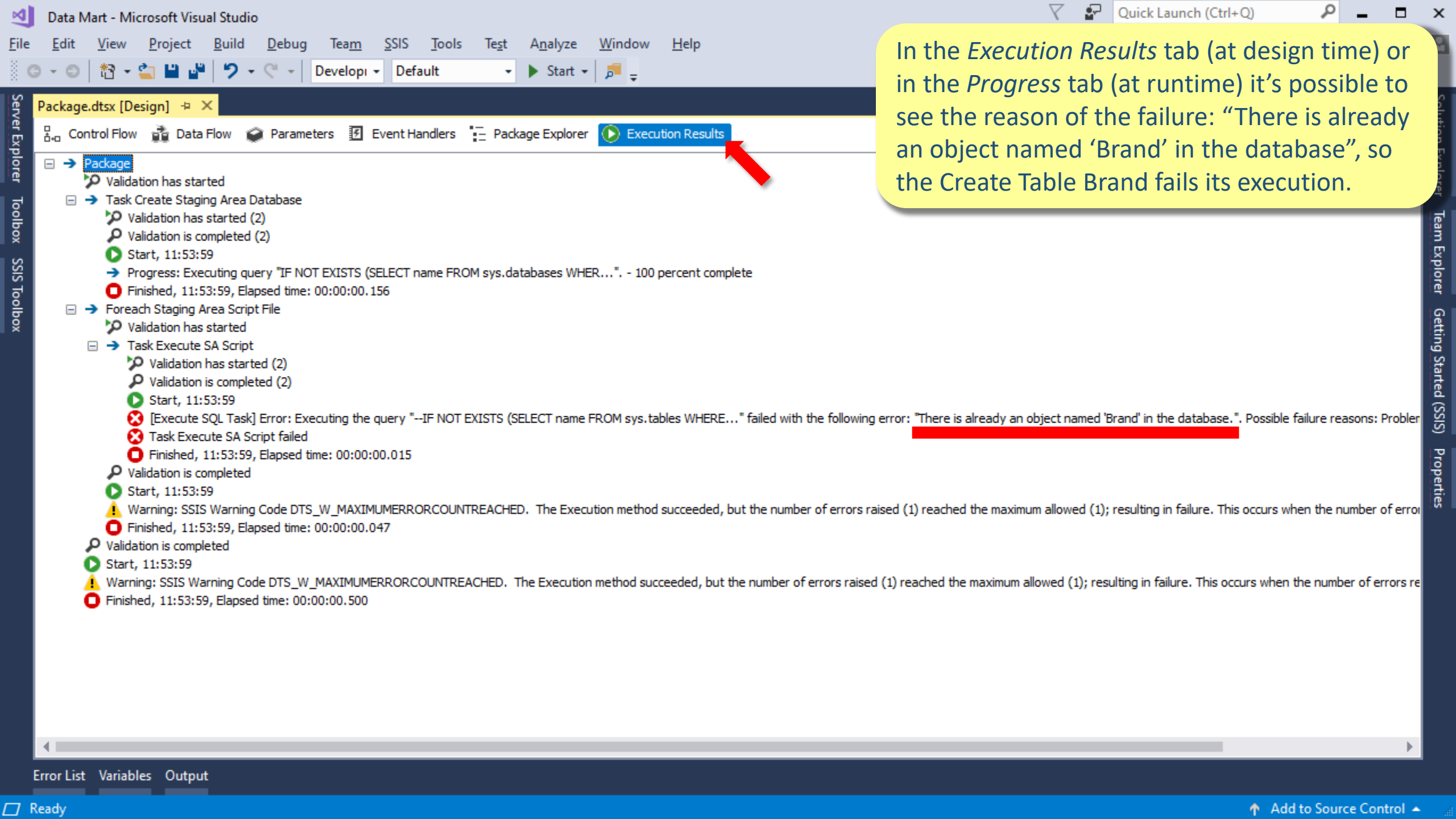


Return to Visual Studio and execute the package once again. You'll now get an error.



Connection Managers		
master	Staging Area	Create SA Tables

Package execution completed with error. Click here to switch to design mode, or select Stop Debugging from the Debug menu.



In the *Execution Results* tab (at design time) or in the *Progress* tab (at runtime) it's possible to see the reason of the failure: "There is already an object named 'Brand' in the database", so the Create Table Brand fails its execution.

Package.dtsx [Design] X

Control Flow Data Flow Parameters Event Handlers Package Explorer Execution Results

Package

- Validation has started
- Task Create Staging Area Database
 - Validation has started (2)
 - Validation is completed (2)
 - Start, 11:53:59
 - Progress: Executing query "IF NOT EXISTS (SELECT name FROM sys.databases WHER..." - 100 percent complete
 - Finished, 11:53:59, Elapsed time: 00:00:00.156
- Foreach Staging Area Script File
 - Validation has started
 - Task Execute SA Script
 - Validation has started (2)
 - Validation is completed (2)
 - Start, 11:53:59
 - [Execute SQL Task] Error: Executing the query "--IF NOT EXISTS (SELECT name FROM sys.tables WHERE..." failed with the following error: "There is already an object named 'Brand' in the database.". Possible failure reasons: Problem with the server, or the database is corrupt and cannot be opened. The database may be corrupt and cannot be opened. The database may be corrupt and cannot be opened.
 - Task Execute SA Script failed
 - Finished, 11:53:59, Elapsed time: 00:00:00.015
 - Validation is completed
 - Start, 11:53:59
 - Warning: SSIS Warning Code DTS_W_MAXIMUMERRORCOUNTREACHED. The Execution method succeeded, but the number of errors raised (1) reached the maximum allowed (1); resulting in failure. This occurs when the number of errors raised is equal to or greater than the value specified in the MaximumErrors property of the TaskHost object.
 - Finished, 11:53:59, Elapsed time: 00:00:00.047
- Validation is completed
- Start, 11:53:59
- Warning: SSIS Warning Code DTS_W_MAXIMUMERRORCOUNTREACHED. The Execution method succeeded, but the number of errors raised (1) reached the maximum allowed (1); resulting in failure. This occurs when the number of errors raised is equal to or greater than the value specified in the MaximumErrors property of the TaskHost object.
- Finished, 11:53:59, Elapsed time: 00:00:00.500

CreateStageTableBrand.sql - localhost.master (AzureAD\PauloOliveira (62))* - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

master Execute

Object Explorer

localhost (SQL Server 16.0.1050.5 - AzureAD)

- Databases
 - System Databases
 - Database Snapshots
 - AdventureWorksDW2022
 - BDAMD_StagingArea
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Brand
 - dbo.Card
 - dbo.Category
 - dbo.Item
 - dbo.SubCategory
 - dbo.Transaction
 - dbo.Transaction_Item
 - Dropped Ledger Tables
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Query Store
 - Service Broker
 - Storage
 - Security

```
IF NOT EXISTS (SELECT name FROM sys.tables WHERE name = 'Brand')
    CREATE TABLE [dbo].[Brand]
    (
        [BrandCode] [char](9),
        [BrandDescription] [nvarchar](50)
    )
ELSE
    TRUNCATE TABLE Brand
```

Results

Commands completed successfully.

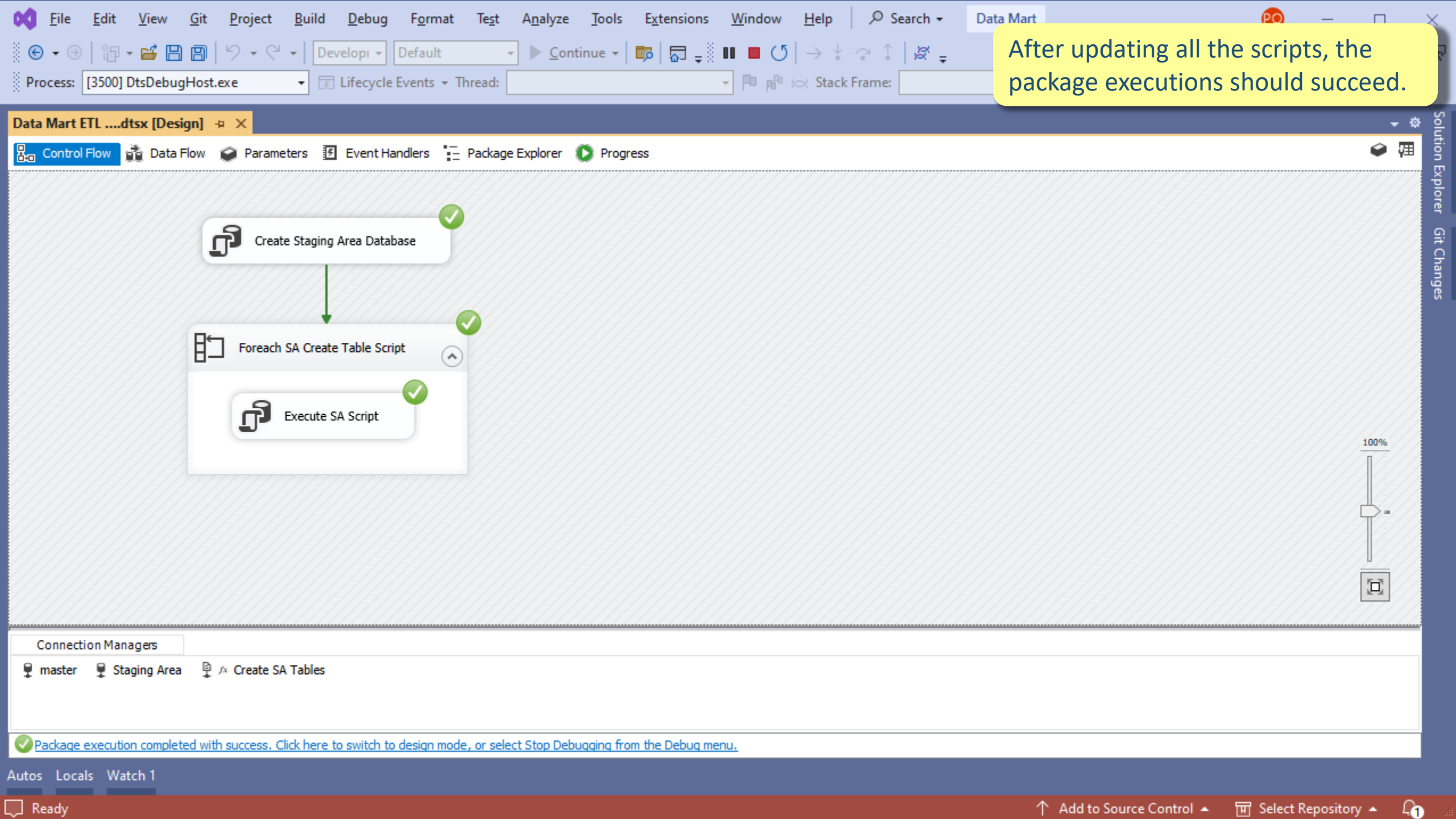
Completion time: 2023-11-27T22:55:23.4517247+00:00

Query executed successfully.

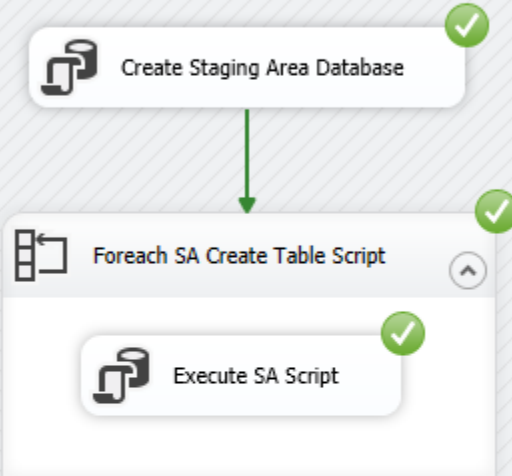
localhost (16.0 RTM) | AzureAD\PauloOliveira ... | master | 00:00:00 | 0 rows

Ready Ln 8 Col 20 Ch 17 INS

You can use SQL Server Management Studio to view and edit the *CreateStageTableBrand.sql*. The script needs to be updated in order to just create the table if it yet doesn't exist. If it already exists, all its data should be deleted/truncated. Update not only this script as presented, but also all the other create table scripts of the staging area. Otherwise, you will get the same error. Every time you change a script you should check its syntax before saving.



After updating all the scripts, the package executions should succeed.



Connection Managers		
master	Staging Area	Create SA Tables

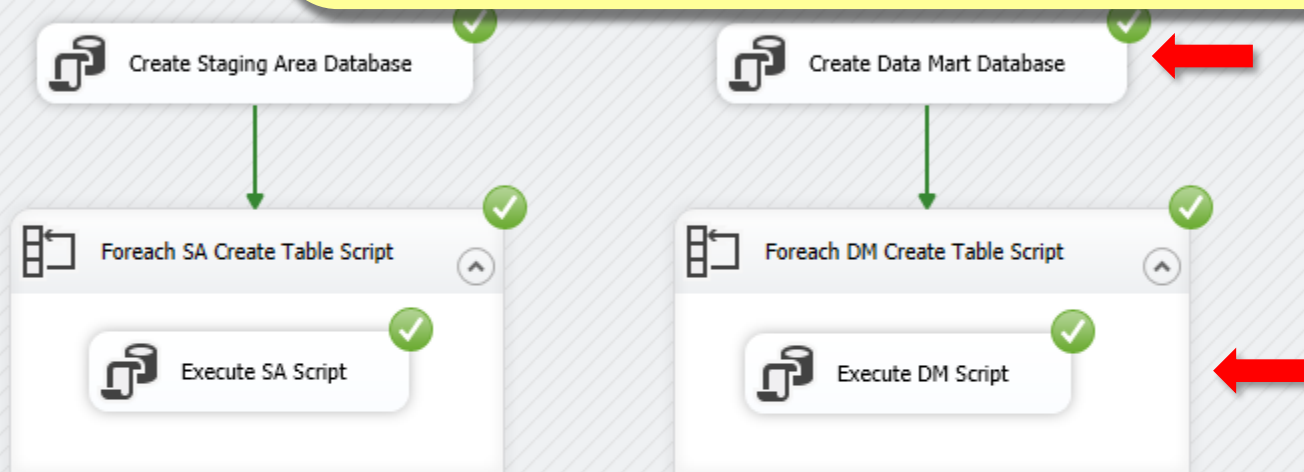
Package execution completed with success. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

Autos Locals Watch 1

The purpose of what you should do next (by yourself) is to:

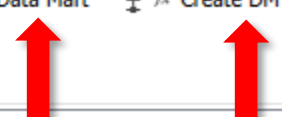
- create the Data Mart database (*BDAMD_DataMart*)
- Iterate through a set of SQL create table script files stored in the Data Mart scripts folder to create each table of the Data Mart database (for that create a *DMScriptFilename* variable)

Basically, you need to repeat the same previous steps, but now for the Data Mart database using the Data Mart script files provided.



Connection Managers

master Staging Area / Create SA Tables Data Mart / Create DM Tables



Package execution completed with success. [Click here to switch to design mode, or select Stop Debugging from the Debug menu.](#)

Autos Locals Watch 1

Ready

↑ Add to Source Control

📁 Select Repository





After the package execution, Data Mart tables should also be created in the Data Mart database. You may need to refresh the databases.

CreateDimCard.sql - localhost.master (AzureAD\PauloOliveira (54))* - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

master Execute

Object Explorer

- localhost (SQL Server 16.0.1050.5 - AzureAD)
 - Databases
 - System Databases
 - Database Snapshots
 - AdventureWorksDW2022
 - BDAMD_DataMart
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.DimCard
 - dbo.DimDate
 - dbo.DimItem
 - dbo.FactSales
 - Dropped Ledger Tables
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Query Store
 - Service Broker
 - Storage
 - Security
 - BDAMD_StagingArea
 - SSISDB
 - Security

CreateDimCard.sql - ...PauloOliveira (54))*

```
IF NOT EXISTS (SELECT name FROM sys.tables WHERE name = 'DimCard')
    CREATE TABLE [dbo].[DimCard](
        [CardKey] [int] IDENTITY(1,1) NOT NULL,
        [CardID] [char](11) NOT NULL,
        [City] [nvarchar](50) NOT NULL,
        [Region] [nvarchar](50) NOT NULL,
        [PostalCode] [nvarchar](10) NOT NULL,
        [CardStartDate] [datetime] NOT NULL,
        [Gender] [nvarchar](6) NOT NULL,
        [DateOfBirth] [datetime] NOT NULL,
        [MaritalStatus] [nvarchar](10) NOT NULL,
        [NumChildren] [tinyint] NULL,
        [YoungestChildren] [tinyint] NULL,
        [EffectiveDate] [datetime] NOT NULL,
        [ExpiredDate] [datetime] NULL
    CONSTRAINT [PK_DimCard] PRIMARY KEY CLUSTERED
    (
        [CardKey] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
    ON [PRIMARY])
```

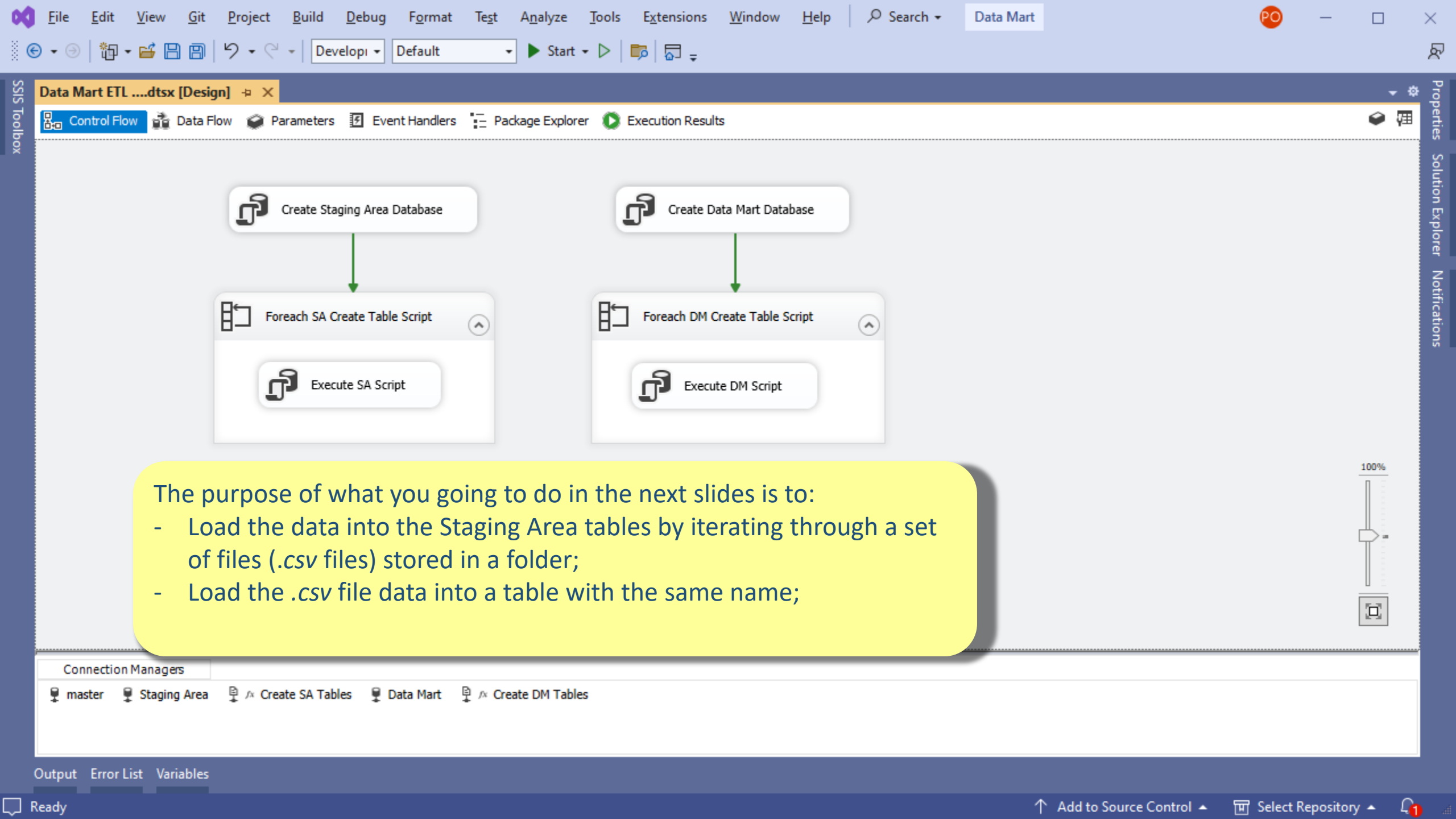
Update the Data Mart create tables script files in order to create the tables only if they don't yet exist. Otherwise, nothing should be done. You don't want to delete all the Data Mart data every time the package runs.

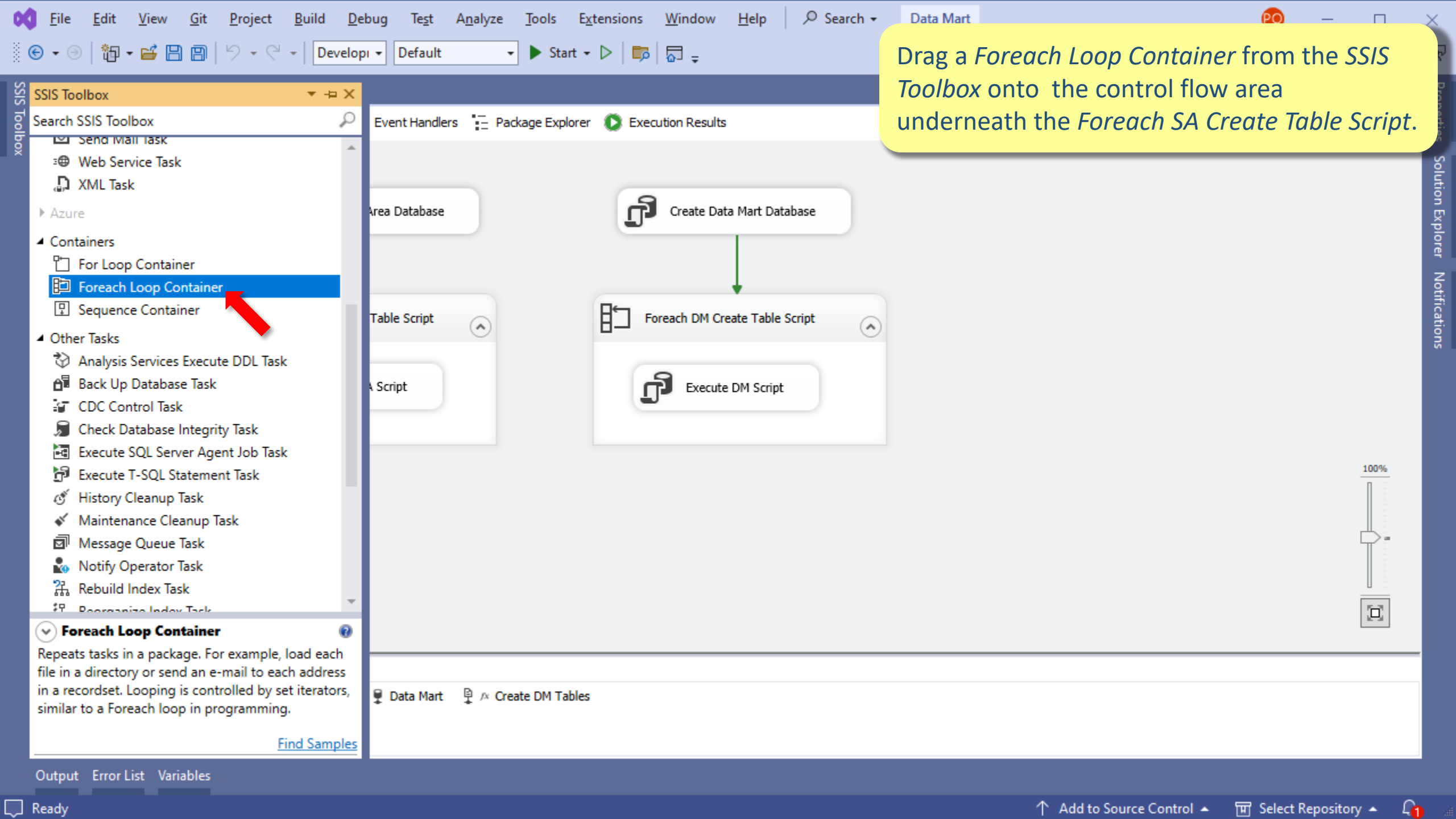
100 %

Connected. (1/1)

localhost (16.0 RTM) | AzureAD\PauloOliveira ... | master | 00:00:00 | 0 rows

Matches:) Ln 1 Col 65 Ch 65 INS





FileEditViewGitProjectBuildDebugFormatTestAnalyzeToolsExtensionsWindowHelp

SearchData Mart

DevelopDefaultStart

SSIS Toolbox

Data Mart ETL...dtsx [Design]*

Control FlowData FlowParametersEvent HandlersPackage ExplorerExecution Results

Create Staging Area Database

Create Data Mart Database

Foreach SA Create Table Script

Execute SA Script

Foreach DM Create Table Script

Execute DM Script

Foreach CSV File

Connection Managers

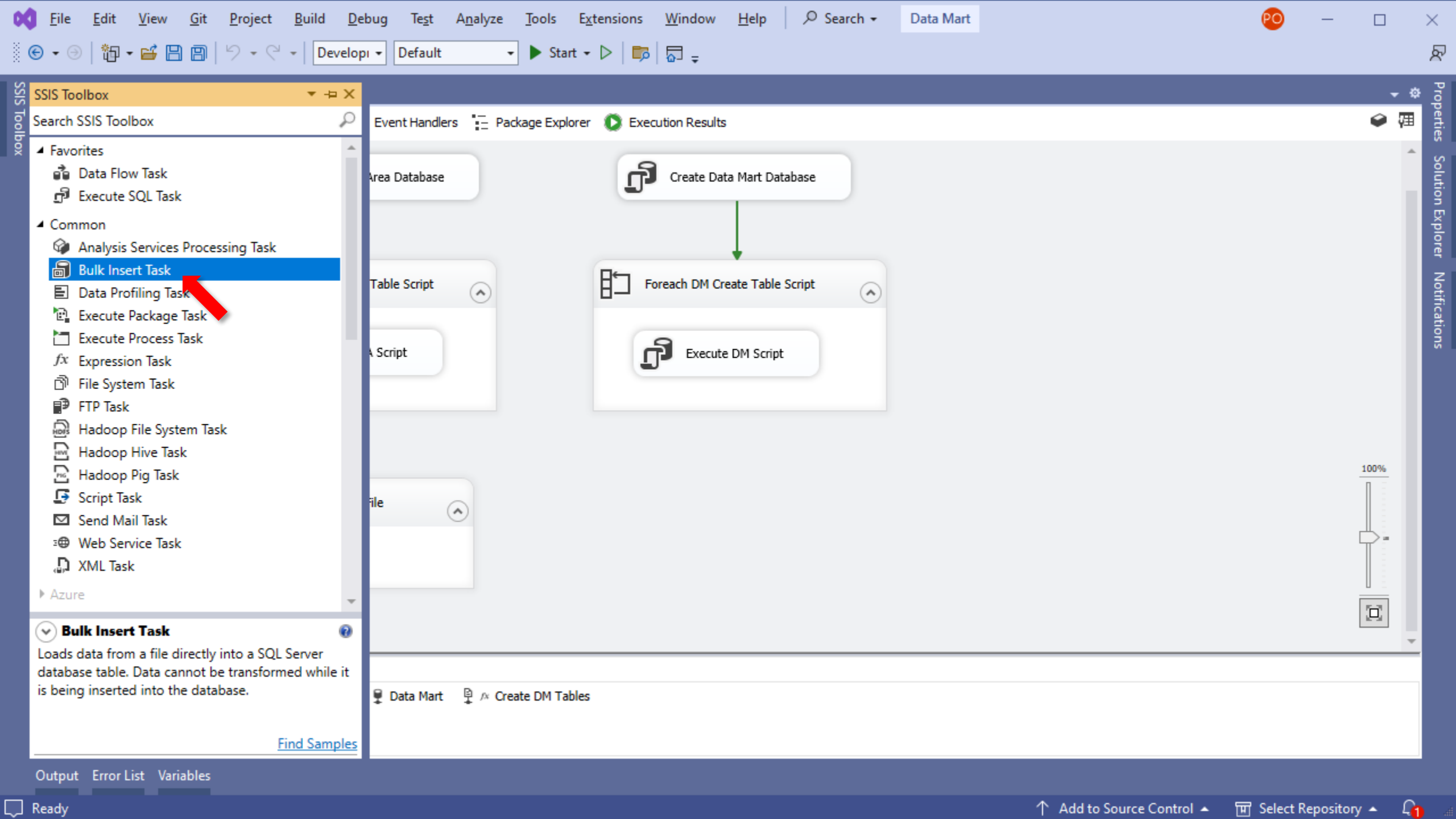
masterStaging AreaCreate SA TablesData MartCreate DM Tables

OutputError ListVariables

Ready

Change the name of the Foreach component to *Foreach CSV File*. Connect this component with the *Foreach SA Create Table Script* through a data flow.

100%



FileEditViewGitProjectBuildDebugFormatTestAnalyzeToolsExtensionsWindowHelp

Search

Data Mart

PO

SSIS Toolbox

Data Mart ETL...dtsx [Design]*

Control FlowData FlowParametersEvent HandlersPackage ExplorerExecution Results

Create Staging Area Database

Foreach SA Create Table Script

Execute SA Script

Foreach CSV File

Bulk Insert Task

Create Data Mart Database

Foreach DM Create Table Script

Execute DM Script

Connection Managers

masterStaging AreaCreate SA TablesData MartCreate DM Tables

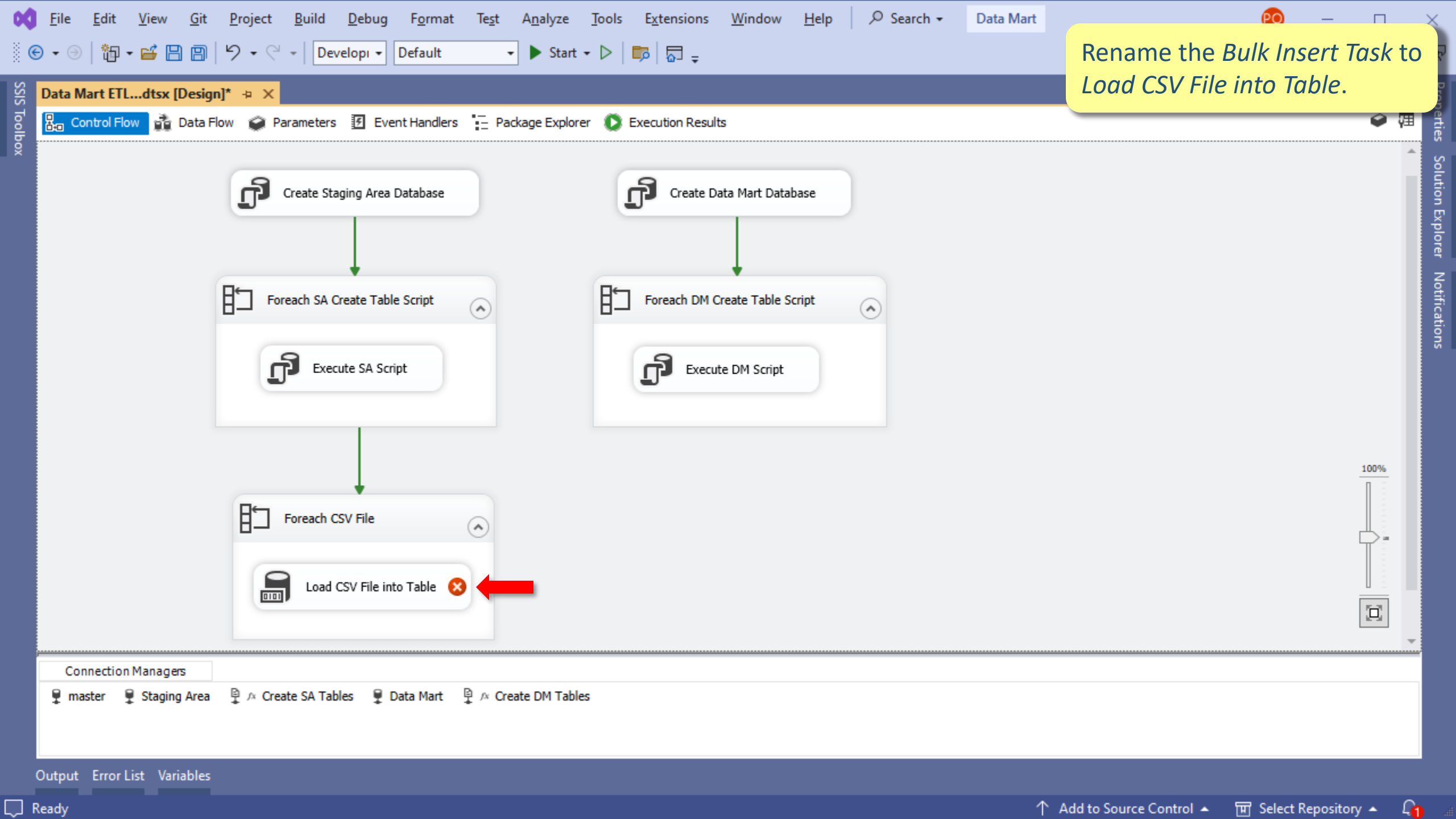
OutputError ListVariables

Ready

PropertiesSolution ExplorerNotifications

PO

1



Rename the *Bulk Insert Task* to *Load CSV File into Table*.

Create Staging Area Database

Create Data Mart Database

Foreach SA Create Table Script

Execute SA Script

Foreach DM Create Table Script

Execute DM Script

Foreach CSV File

Load CSV File into Table



Connection Managers

- master
- Staging Area
- Create SA Tables
- Data Mart
- Create DM Tables

Output Error List Variables

Edit the *Foreach CSV File* and select the *Collection* tab (on the left side). Change the *Enumerator* property to *Foreach File Enumerator* and specify the location of the data files (CSV files) to load in the *Folder* property. In the *Files* property it's possible to restrict the iteration just to CSV files by introducing: *.csv. As you want just to retrieve the filename (without the full path and extension) to be used as the name of the destination table, select the *Name only* option.

The screenshot shows the SQL Server Data Tools (SSDT) interface. The 'Foreach Loop Editor' dialog box is open, and the 'Collection' tab is selected on the left. The 'Enumerator' is set to 'Foreach File Enumerator'. The 'Folder' is 'C:\Temp\Data Mart\Files'. The 'Files' are '*.csv'. The 'Retrieve file name' option is set to 'Name only'. The 'Traverse subfolders' checkbox is unchecked. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.

General
Collection
Variable Mappings
Expressions

Enumerator
Specifies the enumerator type.

Enumerator configuration

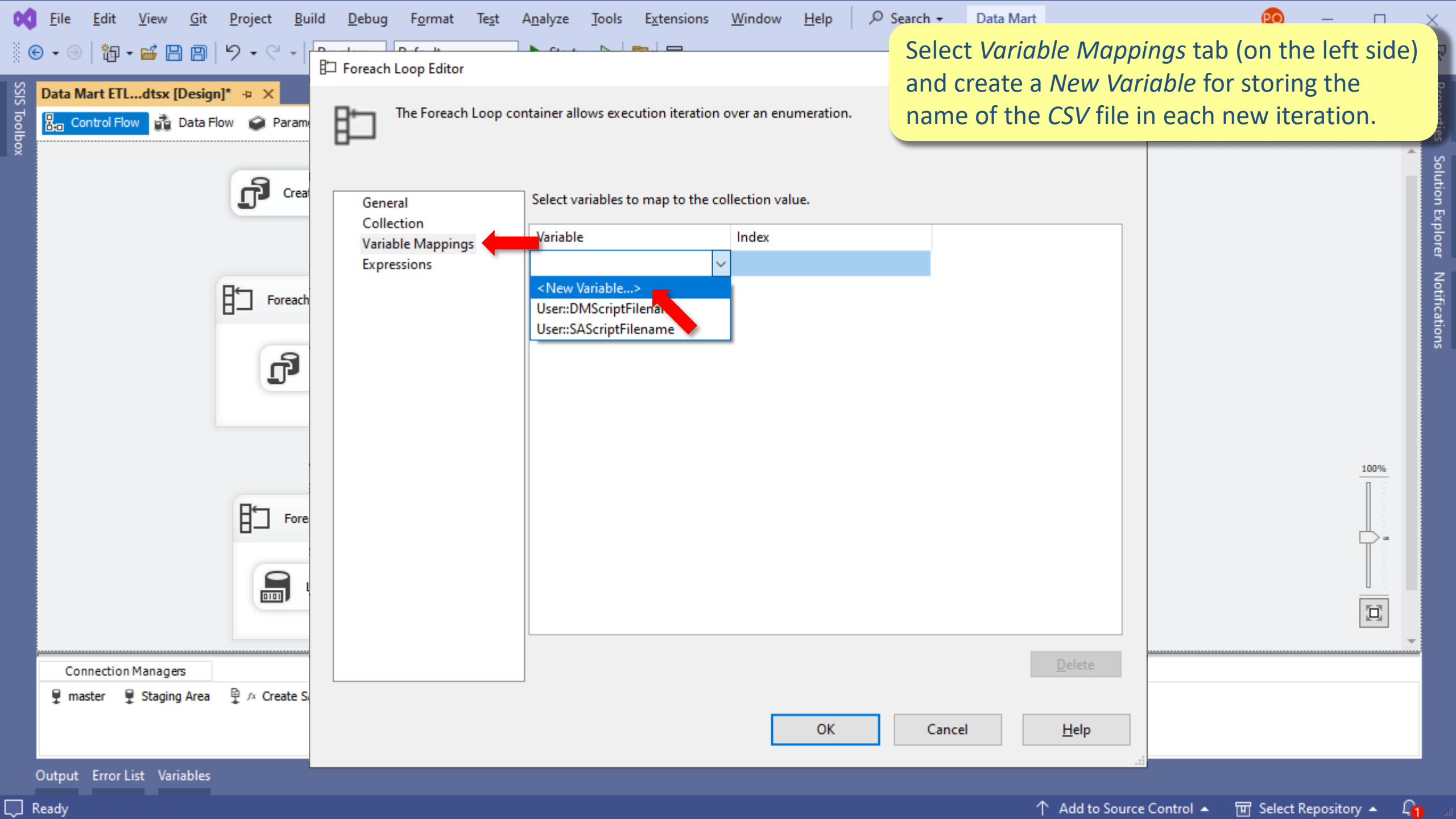
Folder:
C:\Temp\Data Mart\Files

Files:
*.csv

Retrieve file name
☐ Name and extension ☐ Fully qualified ☒ Name only

☐ Traverse subfolders

OK Cancel Help



Select *Variable Mappings* tab (on the left side) and create a *New Variable* for storing the name of the CSV file in each new iteration.

Foreach Loop Editor



The Foreach Loop container allows execution iteration over an enumeration.

- General
- Collection
- Variable Mappings
- Expressions

Select variables to map to the collection value.

Variable	Index
<New Variable...>	
User::DMScriptFilename	
User::SAScriptFilename	

OK

Cancel

Help

Delete

Connection Managers

- master
- Staging Area
- Create S

Output Error List Variables

Ready

Set the variable name as *CSVFilename*.

Foreach Loop Editor



The Foreach Loop container allows execution iteration over an enumeration.

General

Collection

Variable Mappings

Expressions

Select variables to map to the collection value.

Add Variable

Specify the properties of the new variable.

Container:

Data Mart ETL Package

Name:

CSVFilename

Namespace:

User

Value type:

String

Value:

☐ Read only

OK

Cancel

OK

Cancel

Help

Delete

Connection Managers

master

Staging Area

Create S

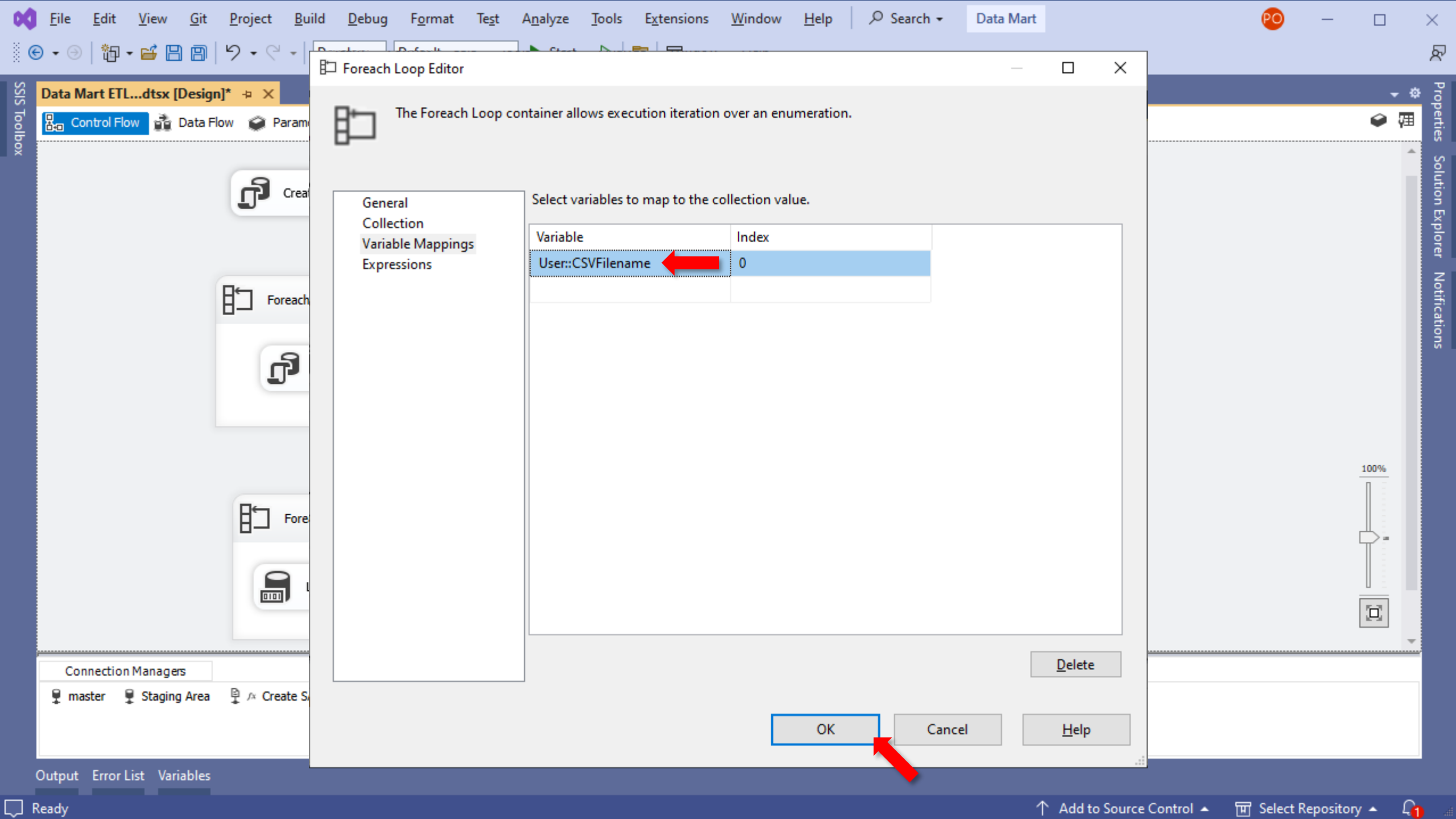
Output Error List Variables

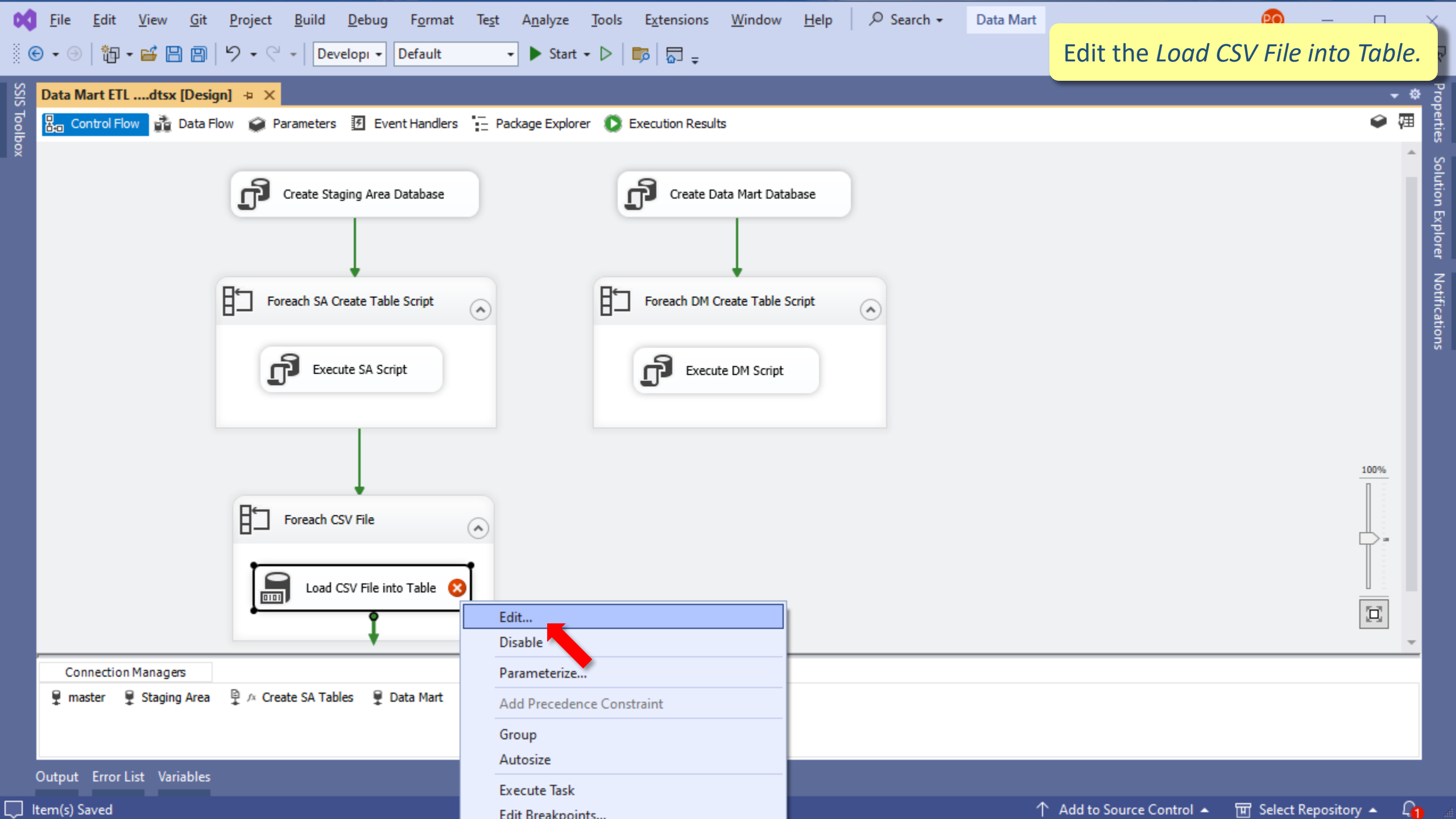
Ready

Add to Source Control

Select Repository

1





In the Connection tab (on the left side) set the *Connection* property to *localhost.StagingArea* connection manager. Set the *DestinationTable* property to the *Brand* table (or any other table you want given that this will be changed briefly). Set the *ColumnDelimiter* property to *Comma*, given that the CSV files are comma delimited. At last, create a new file connection to the CSV files.

Bulk Insert Task Editor

Configure the properties used to import data with this task.

General
Connection
Options
Expressions

Destination
Connection: Staging Area
DestinationTable: [dbo].[Brand]

Format
Format: Specify
RowDelimiter: {CR}{LF}
ColumnDelimiter: Semicolon (;)

Source Connection
File

<New connection...>
Create DM Tables
Create SA Tables

File
Specifies the text file to use as the data source. This file location is relative to the SQL Server Database Engine specified in the connection manager for this task. The text file must be accessible.

Refresh Tables

OK Cancel Help

Browse for selecting one of the CSV files (for instance, *brand.csv*) in the folder where you previously stored them.

Select File

< > > This PC > Windows 10 (C:) > Temp > Data Mart > Files

Organize ▾ New folder

Name	Date modified	Type	Size
BRAND	22/11/2022 17:42	CSV File	1 KB
CARD	22/11/2022 17:42	CSV File	9 212 KB
CATEGORY	22/11/2022 17:42	CSV File	1 KB
ITEM	22/11/2022 17:42	CSV File	44 KB
SUBCATEGORY	22/11/2022 17:42	CSV File	1 KB
TRANSACTION	22/11/2022 18:08	CSV File	1 187 KB
TRANSACTION_ITEM	22/11/2022 18:08	CSV File	1 481 KB

File name: BRAND

All files (*.*)

Open Cancel

OK Cancel Help

Browse for selecting one of the CSV files (for instance, *brand.csv*) in the folder where you previously stored them.

Bulk Insert Task Editor

Configure the properties used to import text files into a SQL Server table. You cannot with this task.

General
Connection
Options
Expressions

Destination Connection

Connection	Staging Area
DestinationTable	[dbo].[Brand]

Format

Format	Specify
--------	---------

File Connection Manager Editor

Configure the file connection properties to reference a file or a folder that exists or is created at run time.

Usage type: Existing file

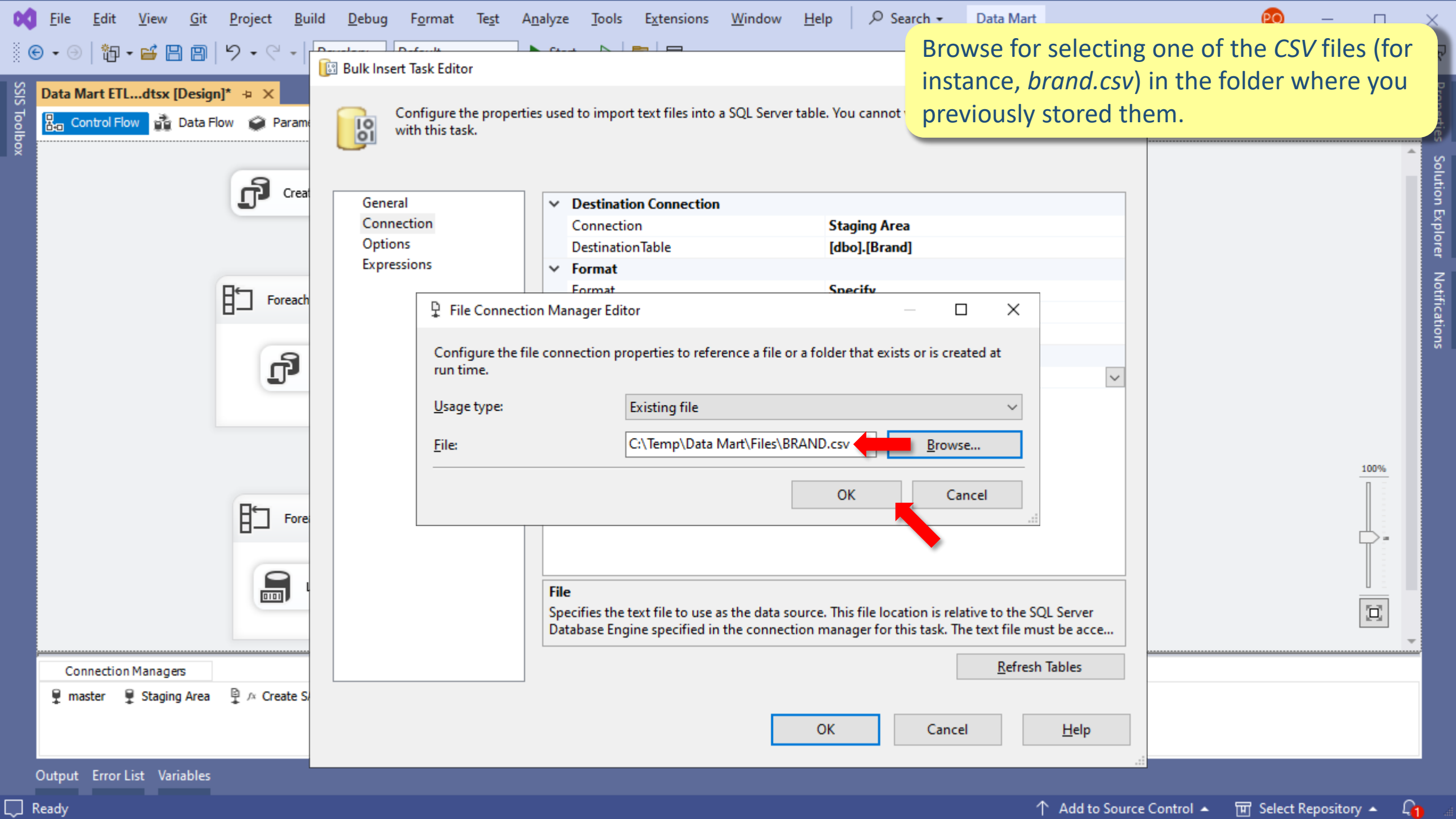
File: C:\Temp\Data Mart\Files\BRAND.csv **Browse...**

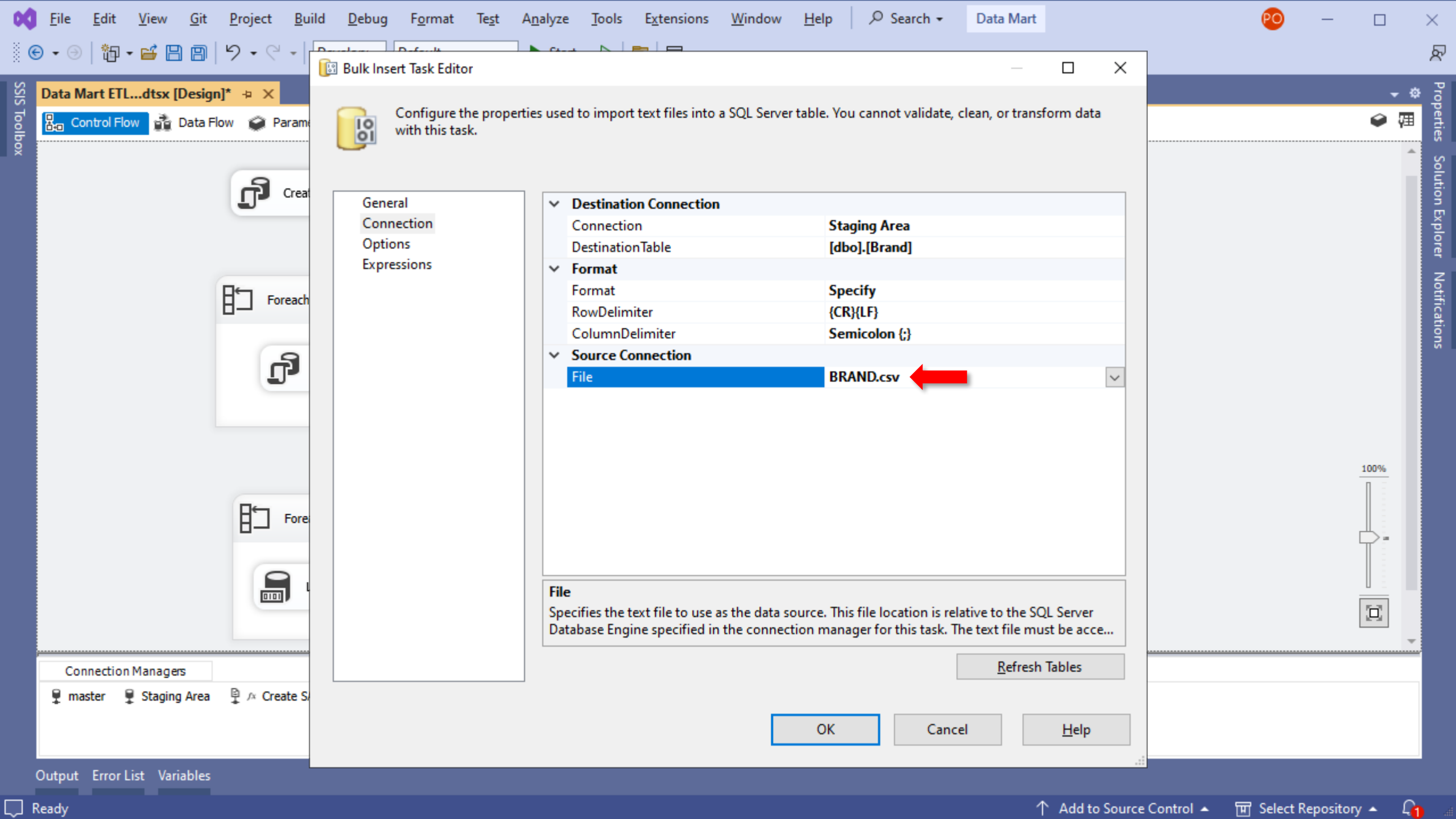
OK Cancel

File
Specifies the text file to use as the data source. This file location is relative to the SQL Server Database Engine specified in the connection manager for this task. The text file must be acce...

Refresh Tables

OK Cancel Help





Bulk Insert Task Editor



Configure the properties used to import text files into a SQL Server table. You cannot validate, clean, or transform data with this task.

General

Connection

Options

Expressions

Destination Connection

Connection

Staging Area

DestinationTable

[dbo].[Brand]

Format

Format

Specify

RowDelimiter

{CR}{LF}

ColumnDelimiter

Semicolon (;)

Source Connection

File

BRAND.csv

File

Specifies the text file to use as the data source. This file location is relative to the SQL Server Database Engine specified in the connection manager for this task. The text file must be acces...

Refresh Tables

OK

Cancel

Help

Connection Managers

master

Staging Area

Create S...

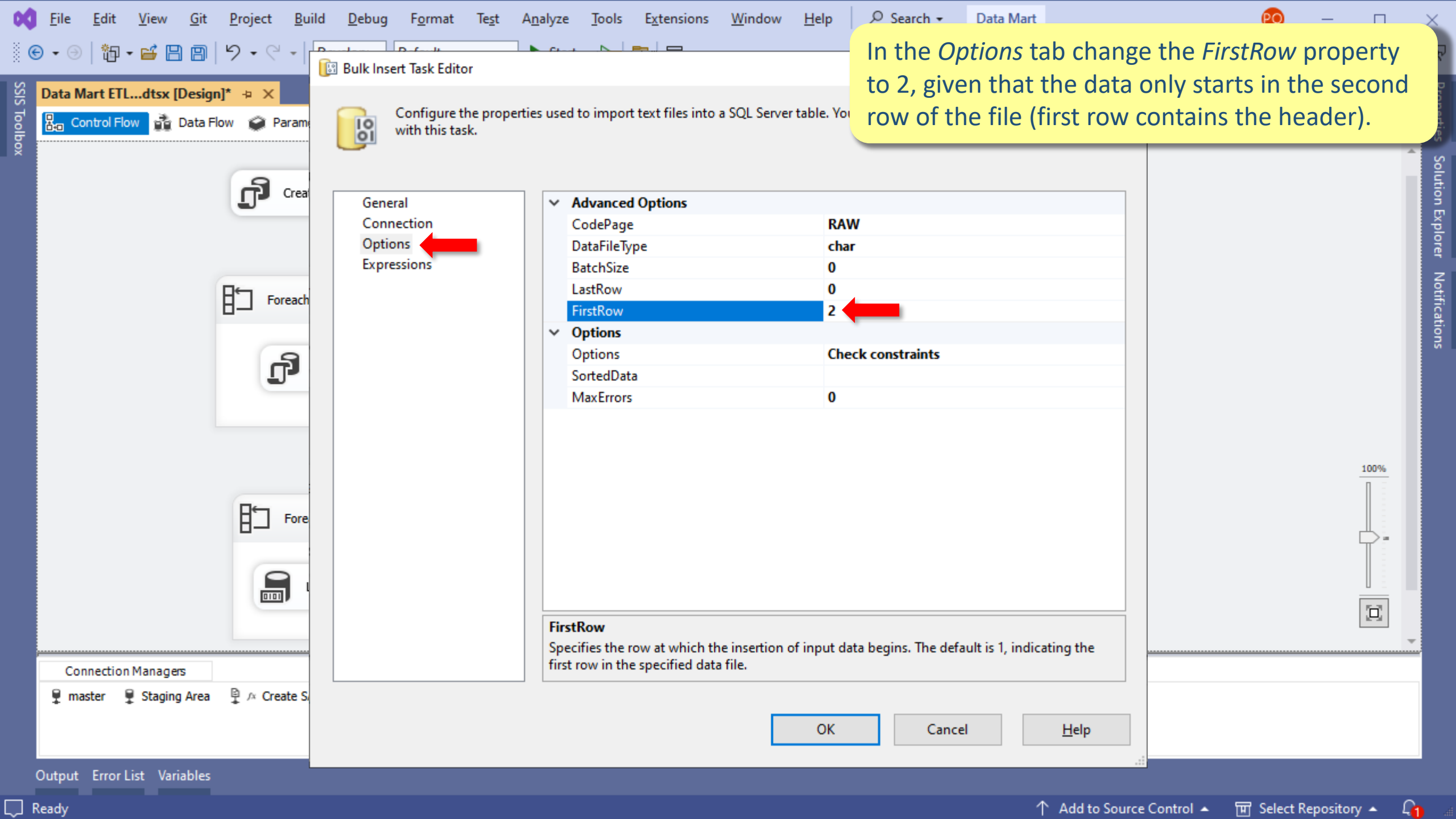
Output Error List Variables

Ready

Add to Source Control

Select Repository

1



In the *Options* tab change the *FirstRow* property to 2, given that the data only starts in the second row of the file (first row contains the header).



Configure the properties used to import text files into a SQL Server table. You can also configure the properties used to import data from a file with this task.

- General
- Connection
- Options
- Expressions

Advanced Options

CodePage	RAW
DataFileType	char
BatchSize	0
LastRow	0
FirstRow	2

Options

Options	Check constraints
SortedData	
MaxErrors	0

FirstRow

Specifies the row at which the insertion of input data begins. The default is 1, indicating the first row in the specified data file.

OK

Cancel

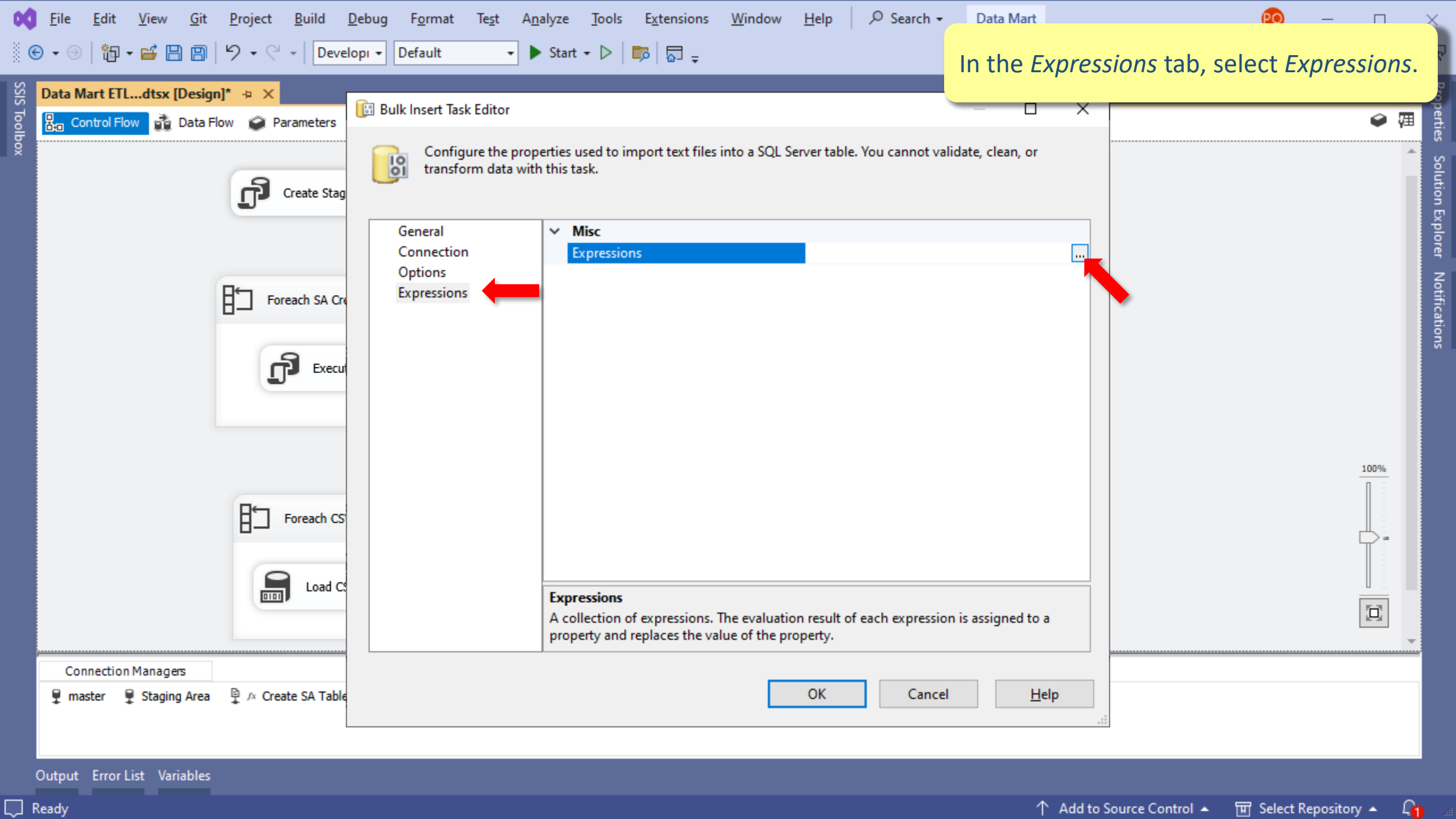
Help

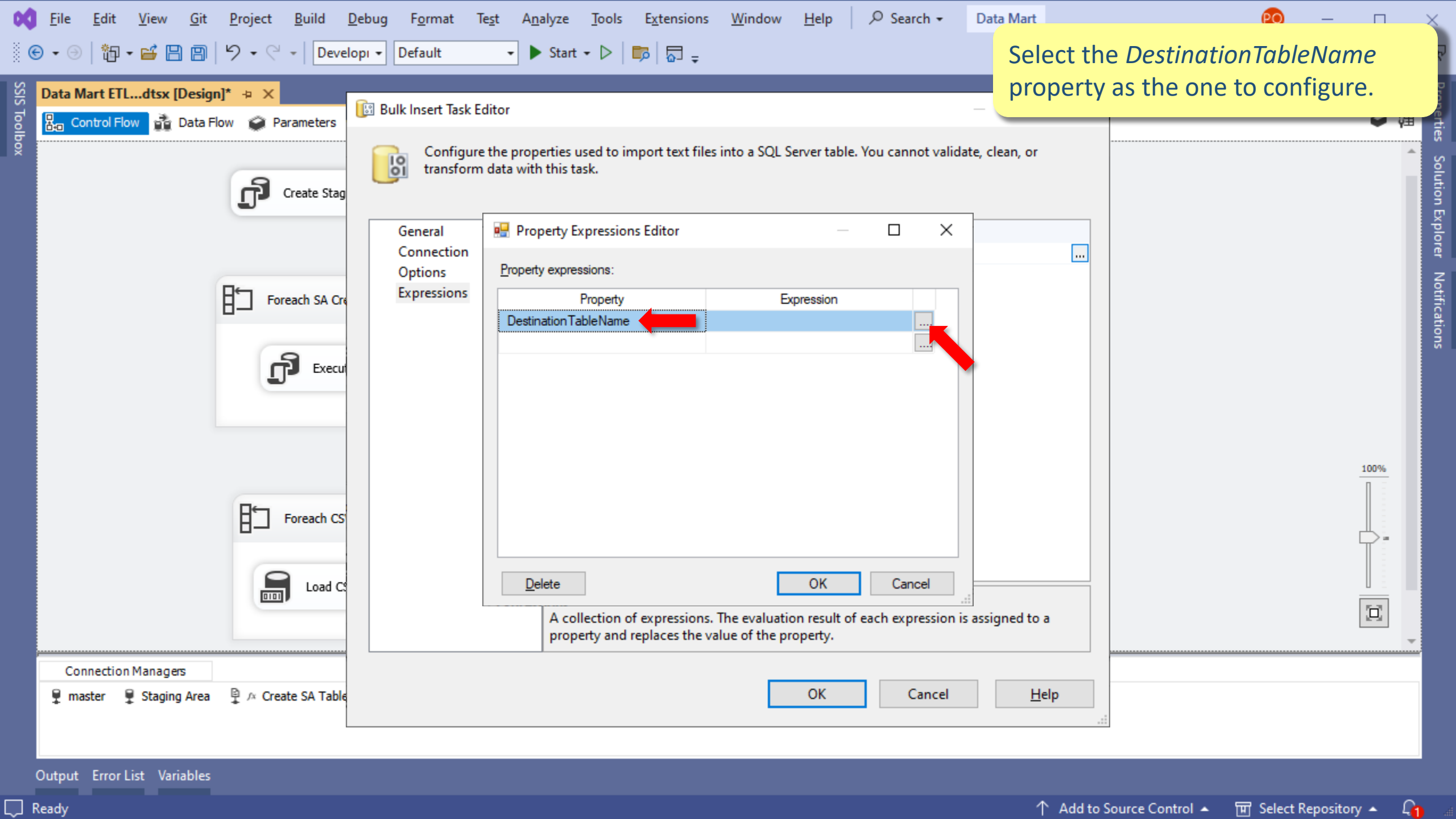
Connection Managers

- master
- Staging Area
- Create S...

Output Error List Variables

Ready





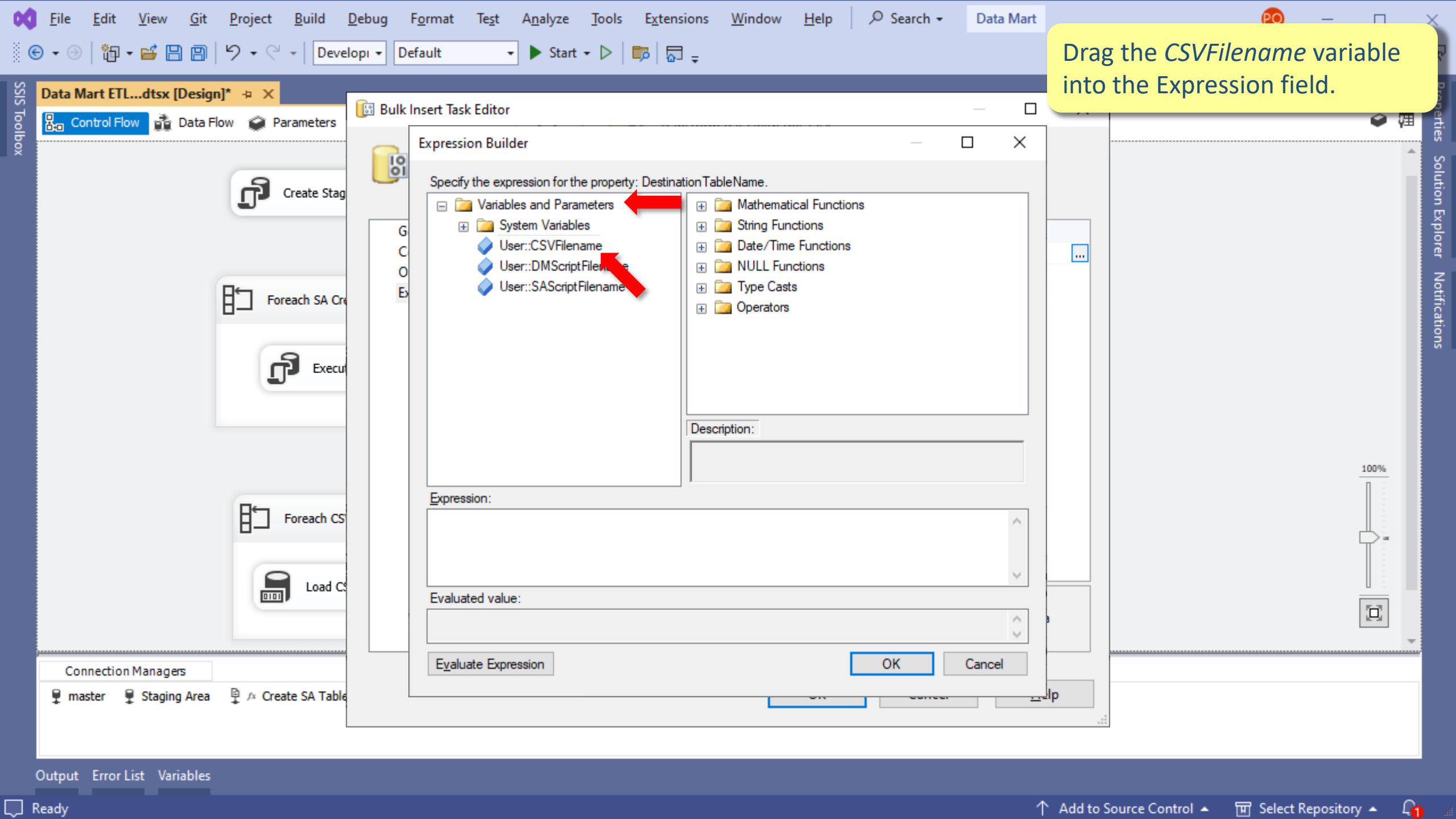
Select the *DestinationTableName* property as the one to configure.

Property Expressions Editor

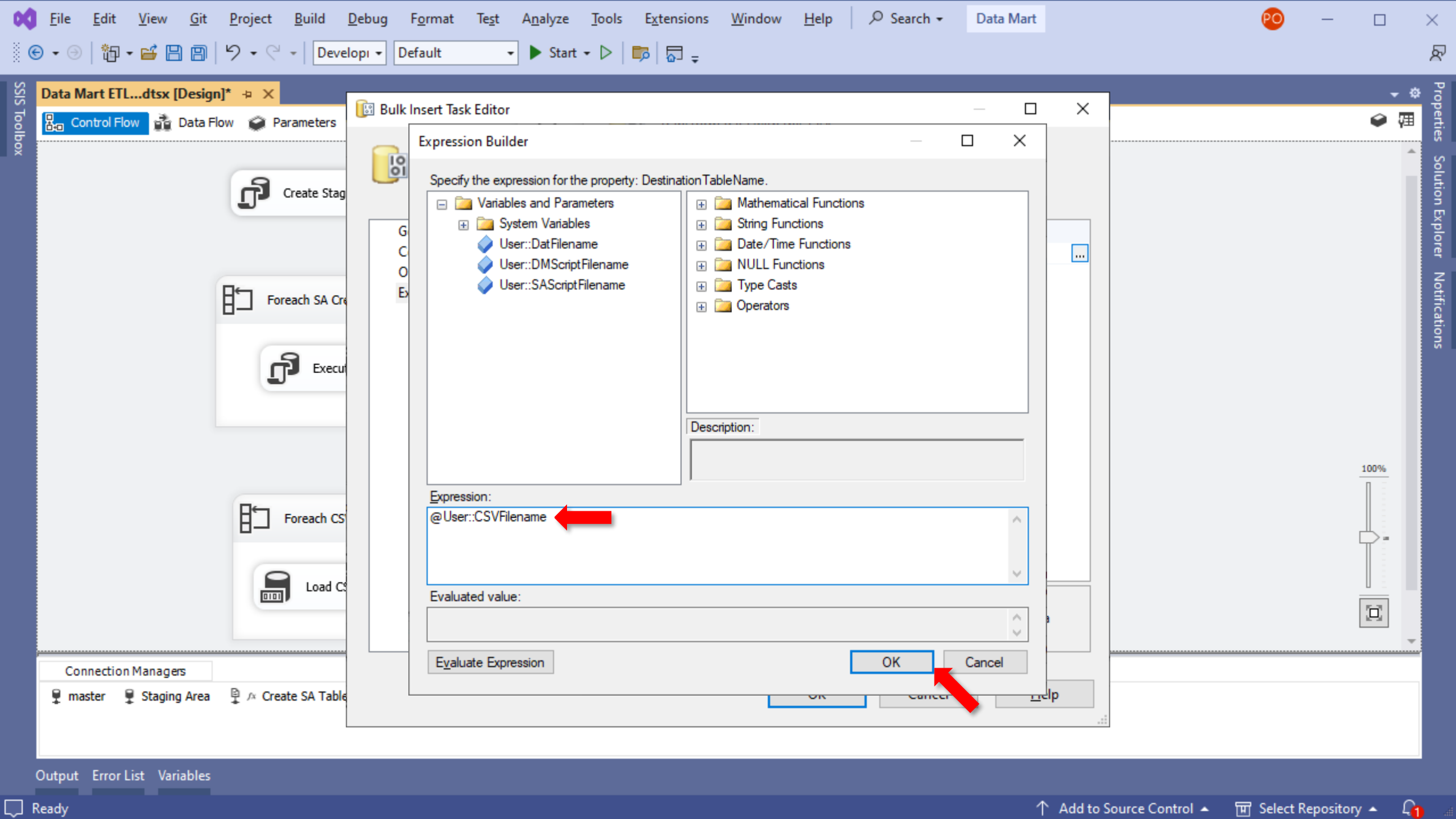
Property expressions:

Property	Expression
DestinationTableName	

A collection of expressions. The evaluation result of each expression is assigned to a property and replaces the value of the property.



Drag the *CSVFilename* variable into the Expression field.



Data Mart ETL...dtsx [Design]*

Control Flow Data Flow Parameters

Create Stage

Foreach SA Cre

Execu

Foreach CS

Load CS

Connection Managers

master Staging Area Create SA Table

Bulk Insert Task Editor



Configure the properties used to import text files into a SQL Server transform data with this task.

General
Connection
Options
Expressions

Property Expressions Editor

Property expressions:

Property	Expression
DestinationTableName	@[User::CSVFilename]

Delete

OK

Cancel

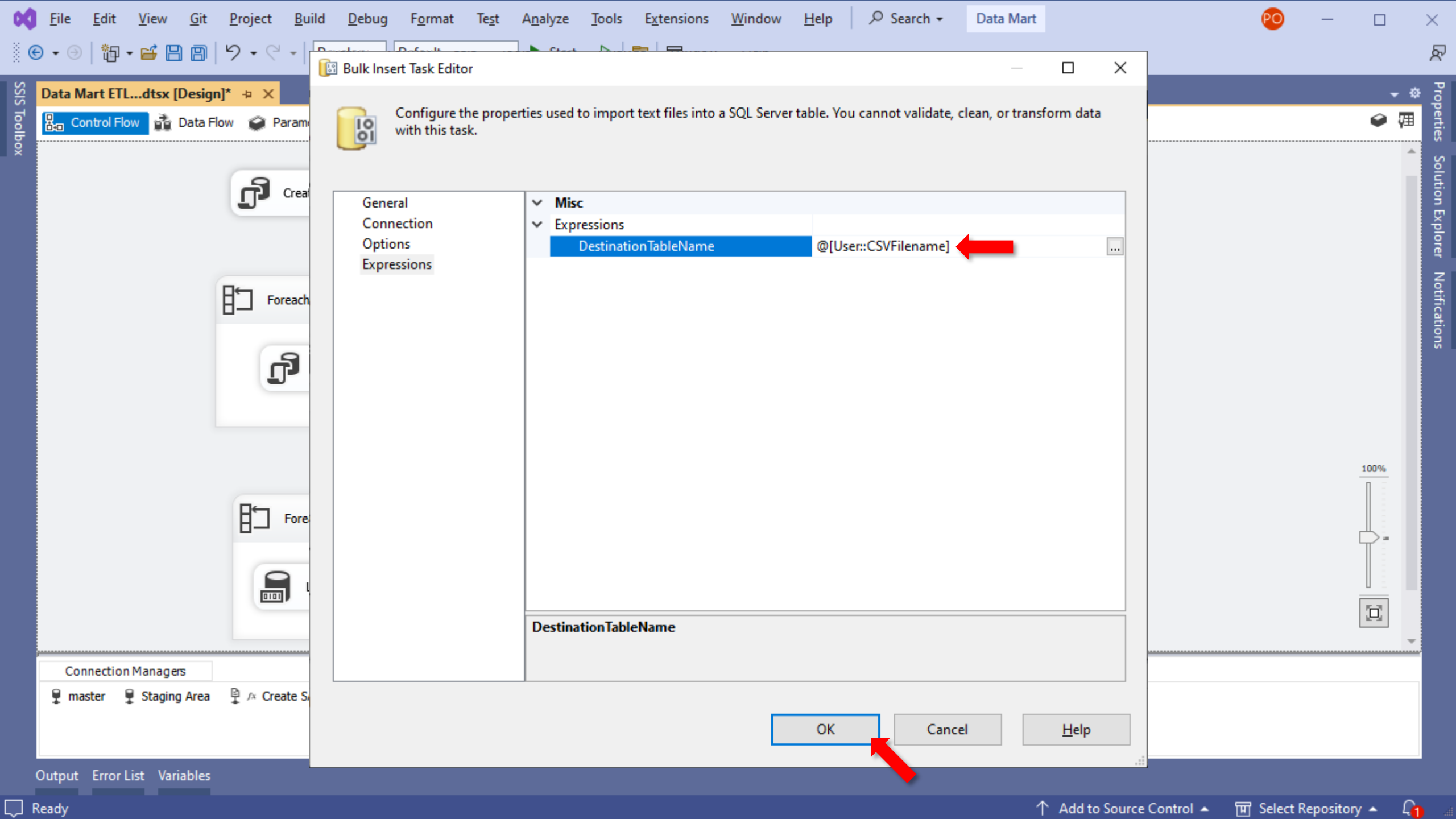
A collection of expressions. The evaluation result of each expression is assigned to a property and replaces the value of the property.

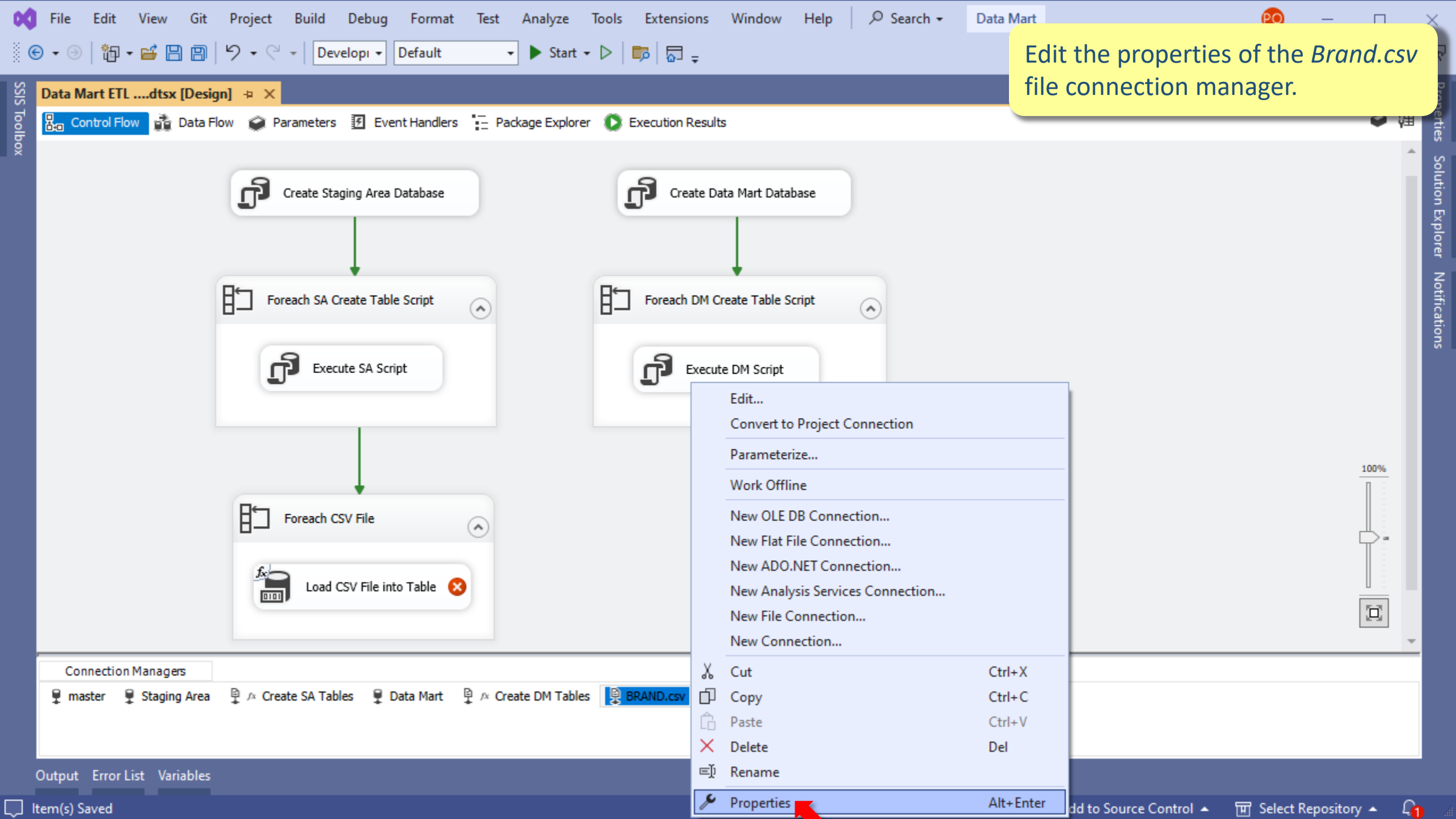
OK

Cancel

Help

By using the *CSVFilename* variable as the *DestinationTableName* this means that the destination table changes according to the value of this variable in each new iteration of the *ForEach* component. This way the destination table is not attached anymore to just a single name of a CSV file. In each iteration, the destination table name is attached to the new name of the csv file.





Edit the properties of the *Brand.csv* file connection manager.

- Edit...
- Convert to Project Connection
- Parameterize...
- Work Offline
- New OLE DB Connection...
- New Flat File Connection...
- New ADO.NET Connection...
- New Analysis Services Connection...
- New File Connection...
- New Connection...
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Delete Del
- Rename
- Properties Alt+Enter**

Connection Managers

master Staging Area Create SA Tables Data Mart Create DM Tables **BRAND.csv**

Output Error List Variables

Item(s) Saved

Add to Source Control Select Repository

FileEditViewGitProjectBuildDebugTestAnalyzeToolsExtensionsWindowHelpSearchData Mart

DevelopiDefaultStart

SSIS Toolbox

Data Mart ETLdtsx [Design]

Control FlowData FlowParametersEvent HandlersPackage ExplorerExecution Results

Create Staging Area Database

↓

Foreach SA Create Table Script

Execute SA Script

↓

Foreach CSV File

Load CSV File into Table

Create Data Mart Database

↓

Foreach DM Create Table Script

Execute DM Script

Connection Managers

masterStaging AreaCreate SA TablesData MartCreate DM TablesBRAND.csv

OutputError ListVariables

Ready

BRAND.csv Connection

Identification

PackagePath\\Package.Connections[BRAND.csv]

Misc

ConnectByProxyFalse

ConnectionManagerTypeFILE

ConnectionStringC:\Temp\Data Mart\Files\BRAND.csv

DataSourceID

DelayValidationFalse

Description

Expressions

FileUsageType0

HasExpressionsFalse

ID{00256E7F-D602-4453-A7D3-...}

NameBRAND.csv

Qualifier

SupportsDTCTransactionFalse

Show Editor

Expressions

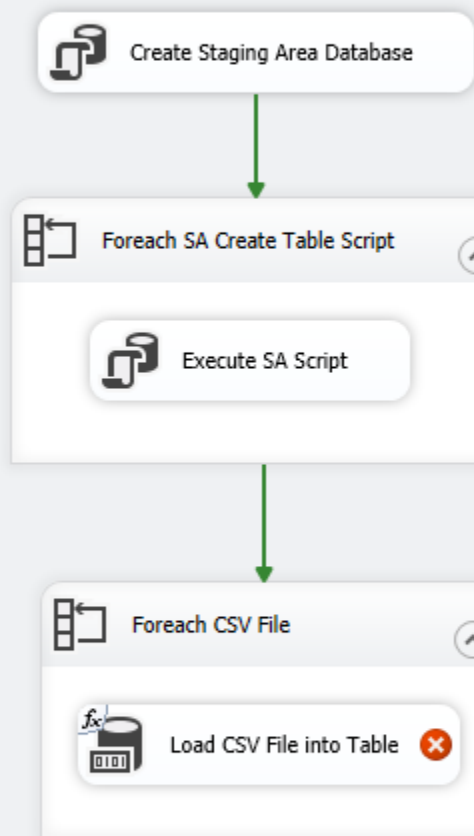
A collection of expressions. The evaluation result of each expression is assigned to a property and replac...

Edit the expressions of the connection manager.

Edit the *ConnectionString* property.

Data Mart ETLdtsx [Design]

Control Flow Data Flow Parameters Event Handlers Package Explorer Execution Results



Property Expressions Editor

Property expressions:

Property	Expression
ConnectionString	

Buttons: Delete, OK, Cancel

Properties

BRAND.csv Connection

Identification

PackagePath: \Package.Connections[BRAND.csv]

Misc

ConnectByProxy: False

ConnectionManagerType: FILE

ConnectionString: C:\Temp\Data Mart\Files\BRAND.csv

DataSourceID:

DelayValidation: False

Description:

Expressions

FileUsageType: 0

HasExpressions: False

ID: {00256E7F-D602-4453-A7D3-...}

Name: BRAND.csv

Qualifier:

SupportsDTCTransaction: False

Show Editor

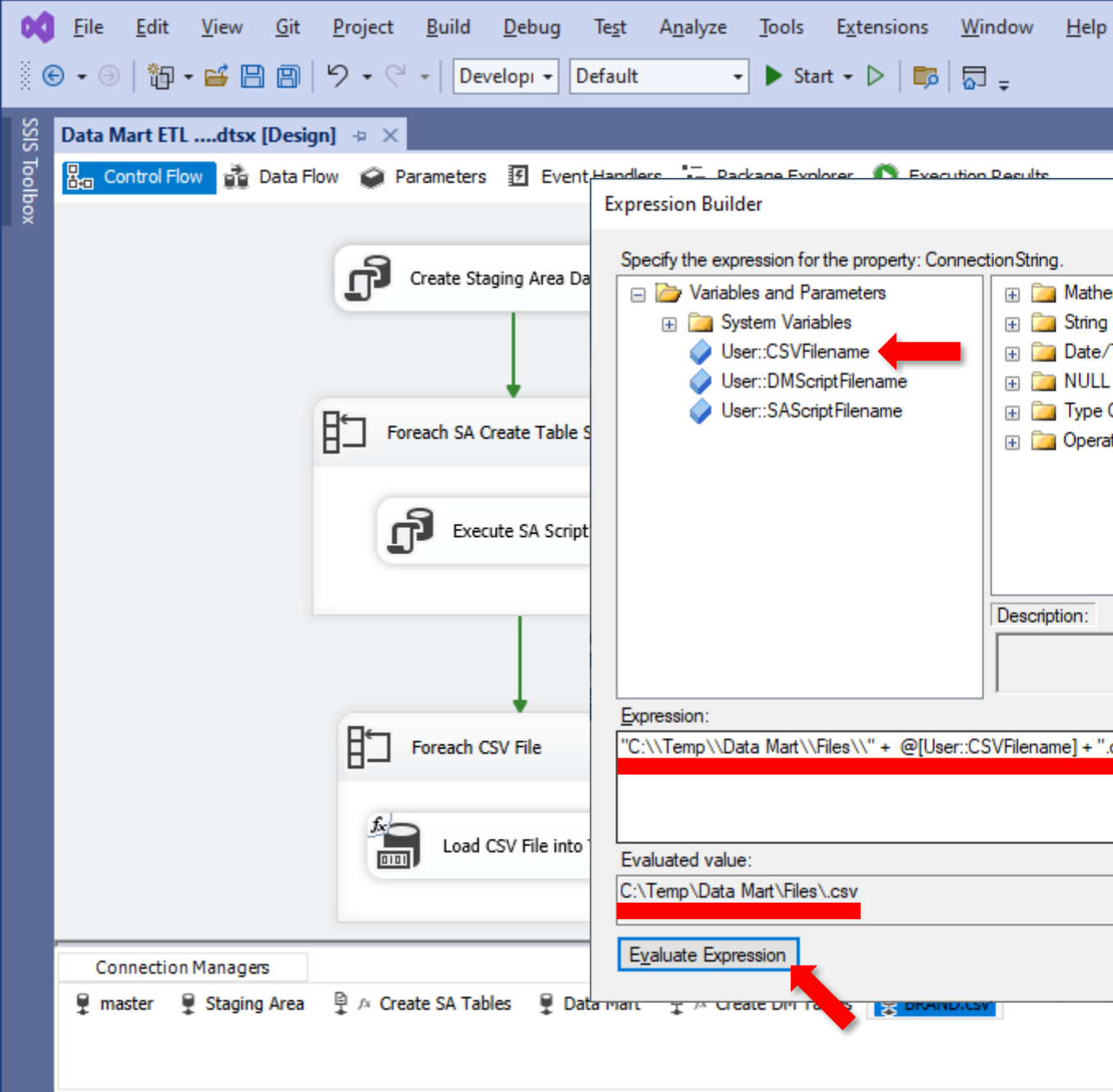
Expressions

A collection of expressions. The evaluation result of each expression is assigned to a property and replaces its value.

Connection Managers

master Staging Area Create SA Tables Data Mart Create DM Tables BRAND.csv

Output Error List Variables



Given that the *CSVFilename* variable only stores the name of the file, the full path and its extension is needed by the connection manager to open the file. In this image, the full path is concatenated with the *CSVFilename* which is concatenated with the extension (*".csv"*). You have to use the path to your *Files* folder. When writing the path, don't forget to use double backslashes (**). You can start by dragging the *CSVFilename* variable into the *Expression* field and then complete the expression with the path (before the variable) and the extension (after the variable). To check if the expression is correct, you can evaluate it. The value of the variable will be empty since it only gets its value at runtime.

[Show Editor](#)

Expressions
A collection of expressions. The evaluation result of each expression is assigned to a property and replac...

FileEditViewGitProjectBuildDebugTestAnalyzeToolsExtensionsWindowHelp

SearchData Mart

SSIS Toolbox

Data Mart ETLdtsx [Design]

Control FlowData FlowParametersEvent HandlersPackage ExplorerExecution Results

Create Staging Area Database

Create Data Mart Database

Foreach SA Create

Execute SA

Foreach CSV Fil

Load CSV F

Property Expressions Editor

Property expressions:

Property	Expression
ConnectionString	"C:\\Temp\\Data Mart\\Files\\" + @[User::CSVFilename] + ".csv"

Delete

OK

Cancel

Connection Managers

masterStaging AreaCreate SA TablesData MartCreate DM TablesBRAND.csv

OutputError ListVariables

Ready

Properties

BRAND.csv Connection

Identification

Misc

Expressions

PackagePath

ConnectionString

DataSourceID

DelayValidation

Description

FileUsageType

HasExpressions

ID

Name

Qualifier

SupportsDTCTransactio

False

FILE

C:\\Temp\\Data Mart\\Files\\B

False

0

False

{00256E7F-D602-4453-A7D3-

BRAND.csv

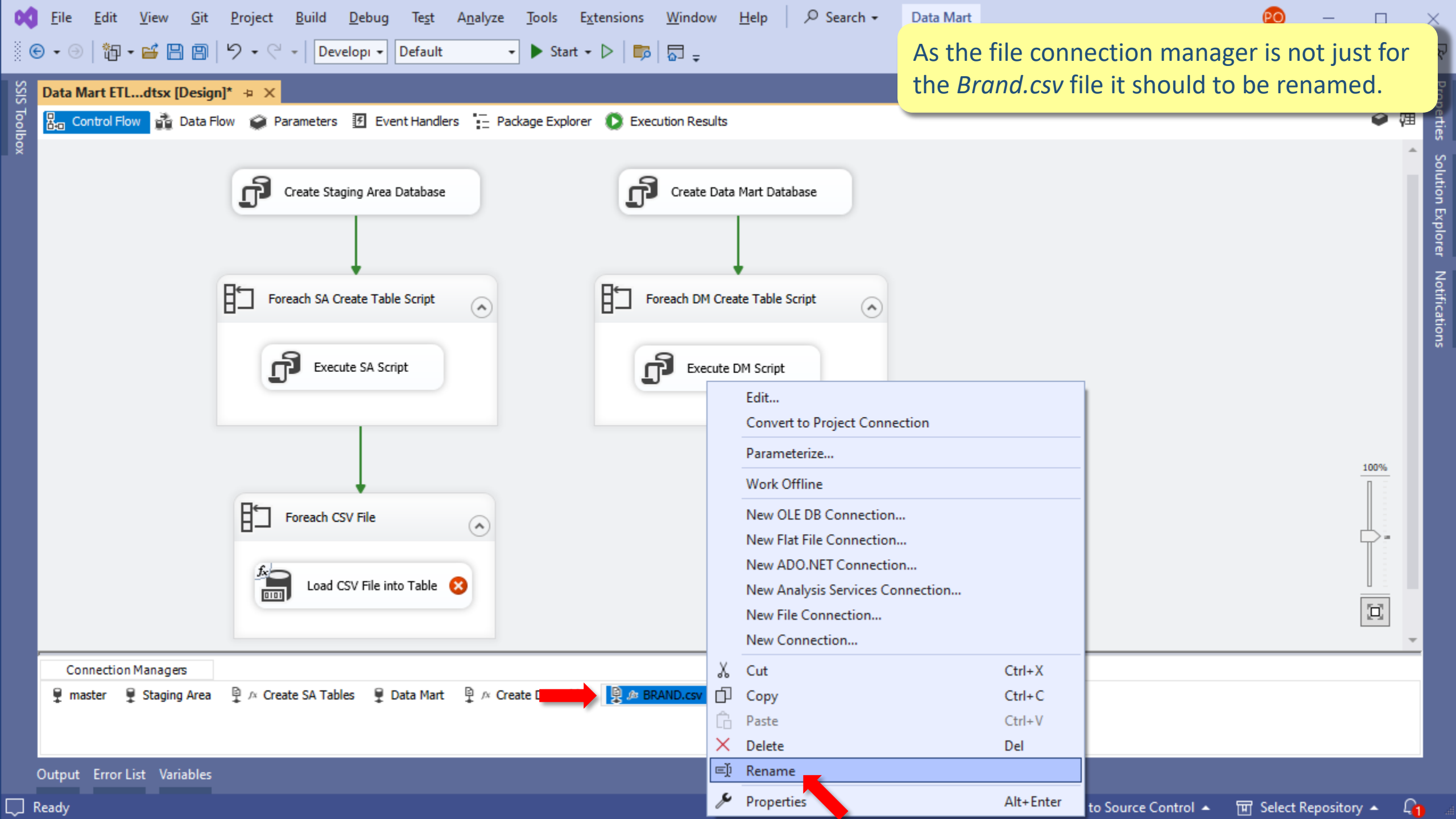
False

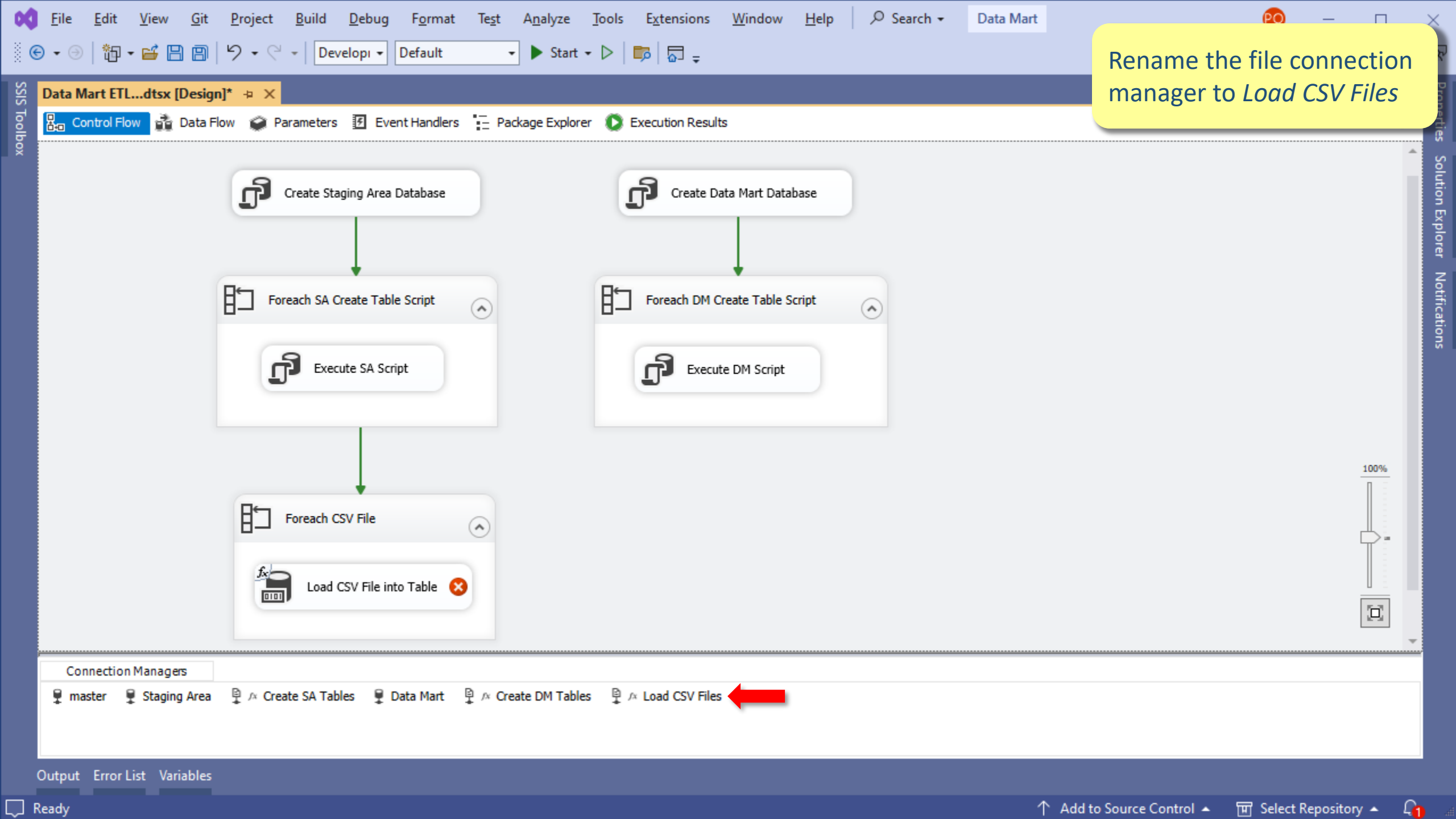
Show Editor

Expressions

A collection of expressions. The evaluation result of each expression is assigned to a property and replac...

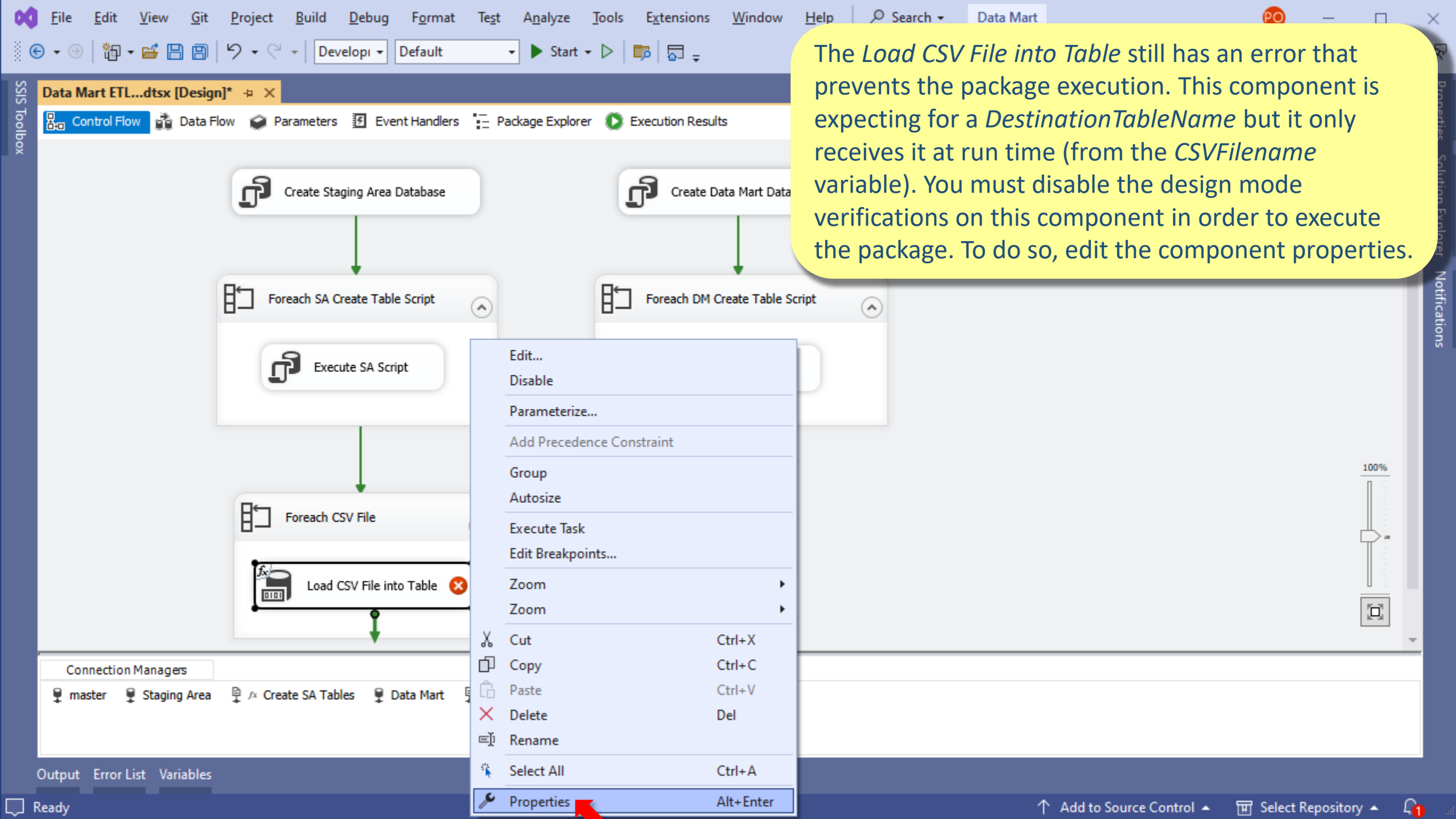
Properties Solution Explorer Notifications





Rename the file connection manager to *Load CSV Files*





The *Load CSV File into Table* still has an error that prevents the package execution. This component is expecting for a *DestinationTableName* but it only receives it at run time (from the *CSVFilename* variable). You must disable the design mode verifications on this component in order to execute the package. To do so, edit the component properties.

- Edit...
- Disable
- Parameterize...
- Add Precedence Constraint
- Group
- Autosize
- Execute Task
- Edit Breakpoints...
- Zoom
- Zoom
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Delete Del
- Rename
- Select All Ctrl+A
- Properties Alt+Enter

FileEditViewGitProjectBuildDebugTestAnalyzeToolsExtensionsWindowHelp

Search

Data Mart

SSIS Toolbox

Data Mart ETL...dtsx [Design]*

Control FlowData FlowParametersEvent HandlersPackage ExplorerExecution Results

Create Staging Area Database

Create Data Mart Database

Foreach SA Create Table Script

Execute SA Script

Foreach DM Create Table Script

Execute DM Script

Foreach CSV File

Load CSV File into Table

Connection Managers

masterStaging AreaCreate SA TablesData MartCreate DM TablesLoad CSV Files

OutputError ListVariables

Ready

Set the *DelayValidation* property of the *Load CSV File into Table* component to *True*.

Load CSV File into Table Task

Execution

DelayValidation	True
Disable	False
DisableEventHandlers	False
FailPackageOnFailure	False
FailParentOnFailure	False
MaximumErrorCount	1

Forced Execution Value

ForcedExecutionValue	0
ForcedExecutionValueType	Int32
ForceExecutionValue	False

Identification

Description	Bulk Insert Task
ID	{C9AFDEDF-5D80-4422-94...
Name	Load CSV File into Table
PackagePath	\Package\Foreach CSV Fil...

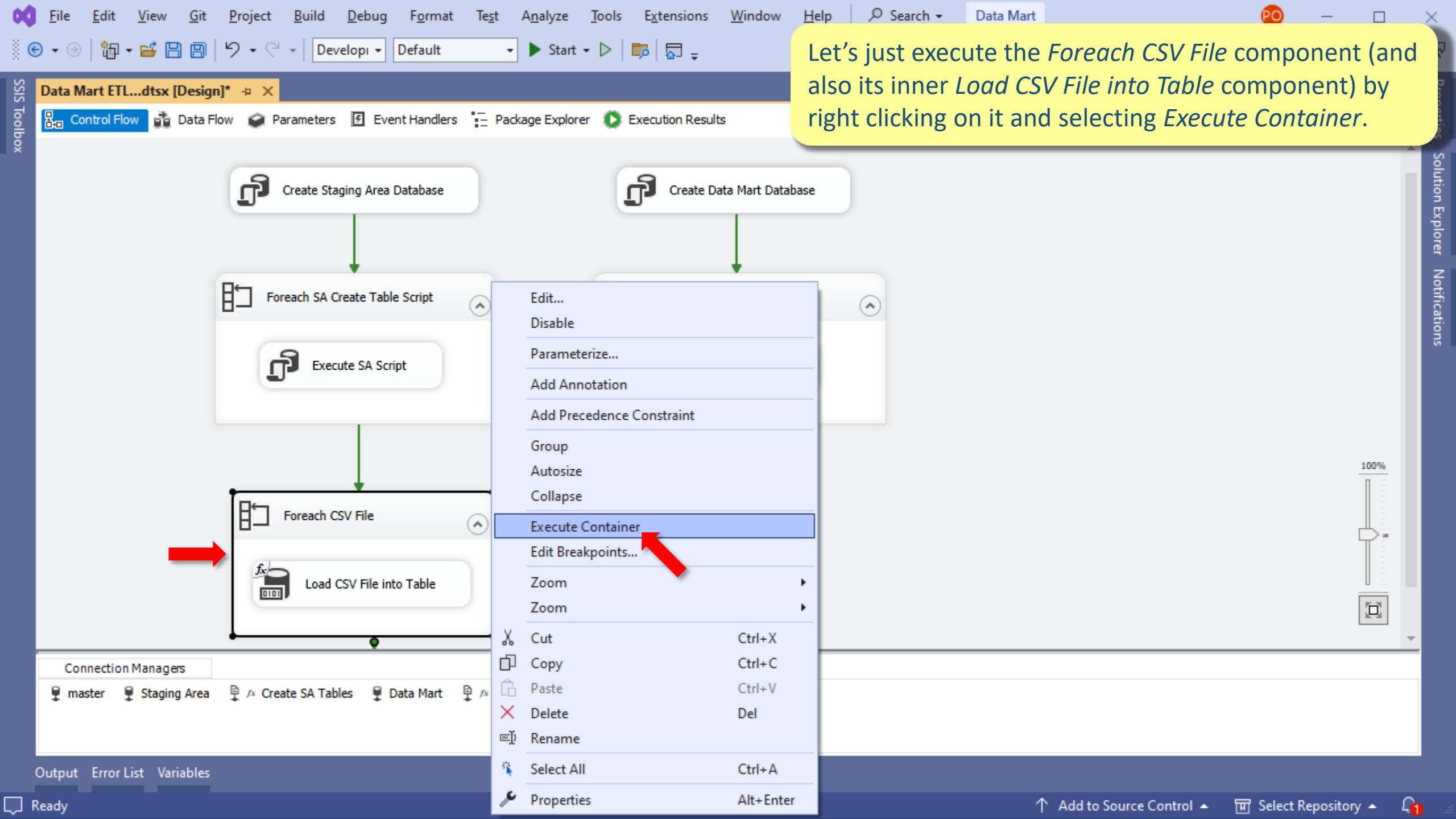
Misc

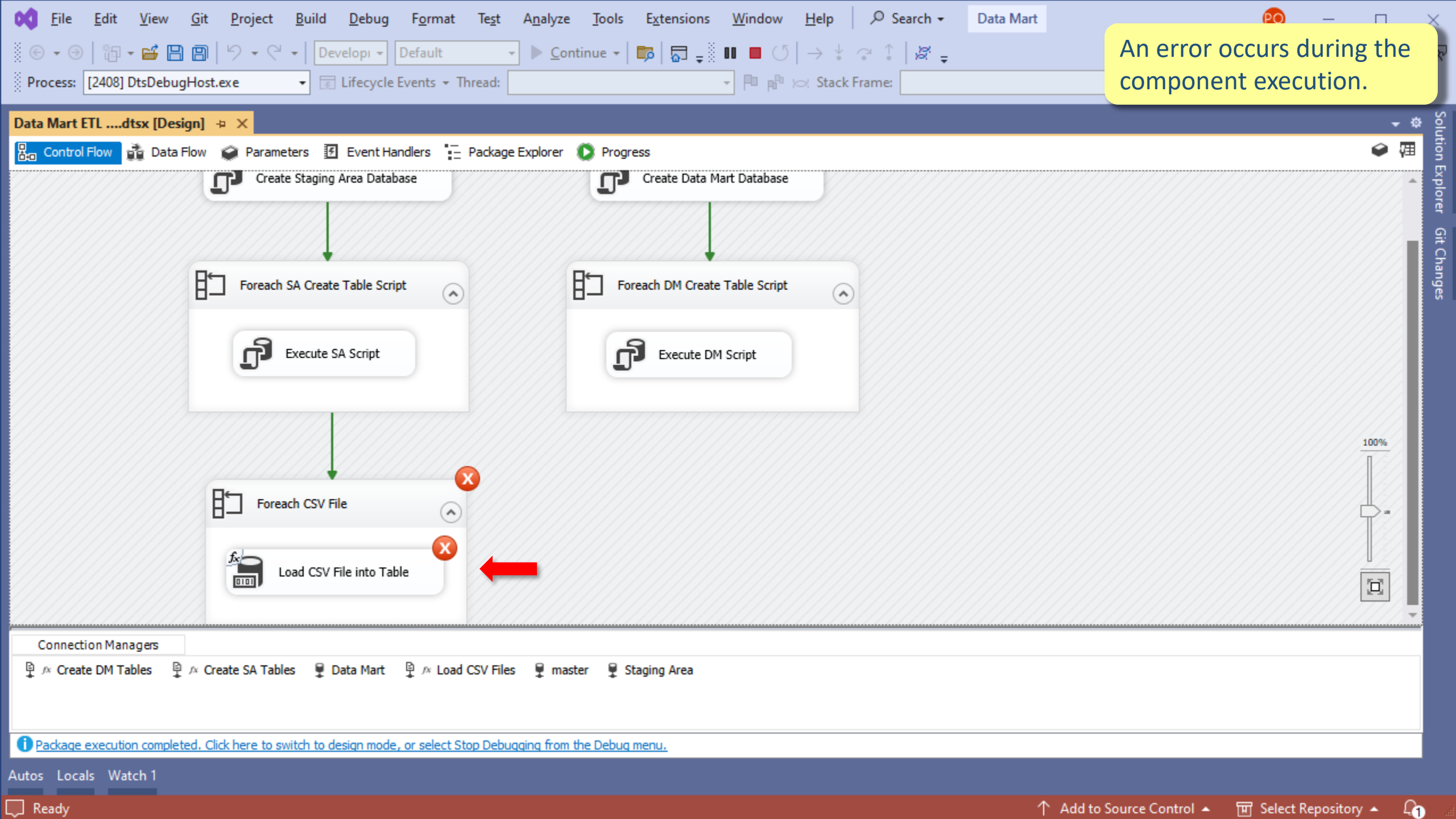
BatchSize	0
CheckConstraints	True
CodePage	RAW
DataFileType	DTSBulkInsert_DataFileType
DestinationConnector	Staging Area

Show Editor

DelayValidation

Indicates whether the validation of the executable is delayed until run time.





An error occurs during the component execution.



Package execution completed. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

FileEditViewGitProjectBuildDebugTestAnalyzeToolsExtensionsWindowHelp

Search (Ctrl+Q)

Data Mart

PO

Process: [9948] DtsDebugHost.exe

Lifecycle Events

Thread:

Stack

Data Mart ETL Package.dtsx [Design]

Control FlowData FlowParametersEvent HandlersPackage ExplorerProgress

Progress: The Bulk Insert task is initiated. - 0 percent complete

Progress: Preparing the following SQL statement: BULK INSERT BRAND FROM 'C:\Temp\BDAMD\Data Mart\Files\BRAND.csv' WITH(CHECK_CONSTRAINTS, CODEPAGE='RAW', DATAFILETYPE='char', FIELDTERMINATOR=';', ROWTERMINATOR=';')

Progress: Executing the following SQL statement: BULK INSERT BRAND FROM 'C:\Temp\BDAMD\Data Mart\Files\BRAND.csv' WITH(CHECK_CONSTRAINTS, CODEPAGE='RAW', DATAFILETYPE='char', FIELDTERMINATOR=';', ROWTERMINATOR=';')

Progress: The Bulk Insert task is completed. - 100 percent complete

Progress: The Bulk Insert task is initiated. - 0 percent complete

Progress: Preparing the following SQL statement: BULK INSERT CARD FROM 'C:\Temp\BDAMD\Data Mart\Files\CARD.csv' WITH(CHECK_CONSTRAINTS, CODEPAGE='RAW', DATAFILETYPE='char', FIELDTERMINATOR=';', ROWTERMINATOR=';')

Progress: Executing the following SQL statement: BULK INSERT CARD FROM 'C:\Temp\BDAMD\Data Mart\Files\CARD.csv' WITH(CHECK_CONSTRAINTS, CODEPAGE='RAW', DATAFILETYPE='char', FIELDTERMINATOR=';', ROWTERMINATOR=';')

Progress: The Bulk Insert task is completed. - 100 percent complete

Progress: The Bulk Insert task is initiated. - 0 percent complete

Progress: Preparing the following SQL statement: BULK INSERT CATEGORY FROM 'C:\Temp\BDAMD\Data Mart\Files\CATEGORY.csv' WITH(CHECK_CONSTRAINTS, CODEPAGE='RAW', DATAFILETYPE='char', FIELDTERMINATOR=';', ROWTERMINATOR=';')

Progress: Executing the following SQL statement: BULK INSERT CATEGORY FROM 'C:\Temp\BDAMD\Data Mart\Files\CATEGORY.csv' WITH(CHECK_CONSTRAINTS, CODEPAGE='RAW', DATAFILETYPE='char', FIELDTERMINATOR=';', ROWTERMINATOR=';')

Progress: The Bulk Insert task is completed. - 100 percent complete

Progress: The Bulk Insert task is initiated. - 0 percent complete

Progress: Preparing the following SQL statement: BULK INSERT ITEM FROM 'C:\Temp\BDAMD\Data Mart\Files\ITEM.csv' WITH(CHECK_CONSTRAINTS, CODEPAGE='RAW', DATAFILETYPE='char', FIELDTERMINATOR=';', ROWTERMINATOR=';')

Progress: Executing the following SQL statement: BULK INSERT ITEM FROM 'C:\Temp\BDAMD\Data Mart\Files\ITEM.csv' WITH(CHECK_CONSTRAINTS, CODEPAGE='RAW', DATAFILETYPE='char', FIELDTERMINATOR=';', ROWTERMINATOR=';')

Progress: The Bulk Insert task is completed. - 100 percent complete

Progress: The Bulk Insert task is initiated. - 0 percent complete

Progress: Preparing the following SQL statement: BULK INSERT SUBCATEGORY FROM 'C:\Temp\BDAMD\Data Mart\Files\SUBCATEGORY.csv' WITH(CHECK_CONSTRAINTS, CODEPAGE='RAW', DATAFILETYPE='char', FIELDTERMINATOR=';', ROWTERMINATOR=';')

Progress: Executing the following SQL statement: BULK INSERT SUBCATEGORY FROM 'C:\Temp\BDAMD\Data Mart\Files\SUBCATEGORY.csv' WITH(CHECK_CONSTRAINTS, CODEPAGE='RAW', DATAFILETYPE='char', FIELDTERMINATOR=';', ROWTERMINATOR=';')

Progress: The Bulk Insert task is completed. - 100 percent complete

Progress: The Bulk Insert task is initiated. - 0 percent complete

Progress: Preparing the following SQL statement: BULK INSERT TRANSACTION FROM 'C:\Temp\BDAMD\Data Mart\Files\TRANSACTION.csv' WITH(CHECK_CONSTRAINTS, CODEPAGE='RAW', DATAFILETYPE='char', FIELDTERMINATOR=';', ROWTERMINATOR=';')

Progress: Executing the following SQL statement: BULK INSERT TRANSACTION FROM 'C:\Temp\BDAMD\Data Mart\Files\TRANSACTION.csv' WITH(CHECK_CONSTRAINTS, CODEPAGE='RAW', DATAFILETYPE='char', FIELDTERMINATOR=';', ROWTERMINATOR=';')

[Bulk Insert Task] Error: An error occurred with the following error message: "Incorrect syntax near the keyword 'TRANSACTION'".

Progress: The Bulk Insert task is completed. - 100 percent complete

Task Load CSV File into Table failed

Stop (6)

Warning: SSIS Warning Code DTS_W_MAXIMUMERRORCOUNTREACHED. The Execution method succeeded, but the number of errors raised (1) reached the maximum allowed (1); resulting in failure. This occurs when the number of errors

Finished, 12:16:27 PM, Elapsed time: 00:00:21.703

Package execution completed. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

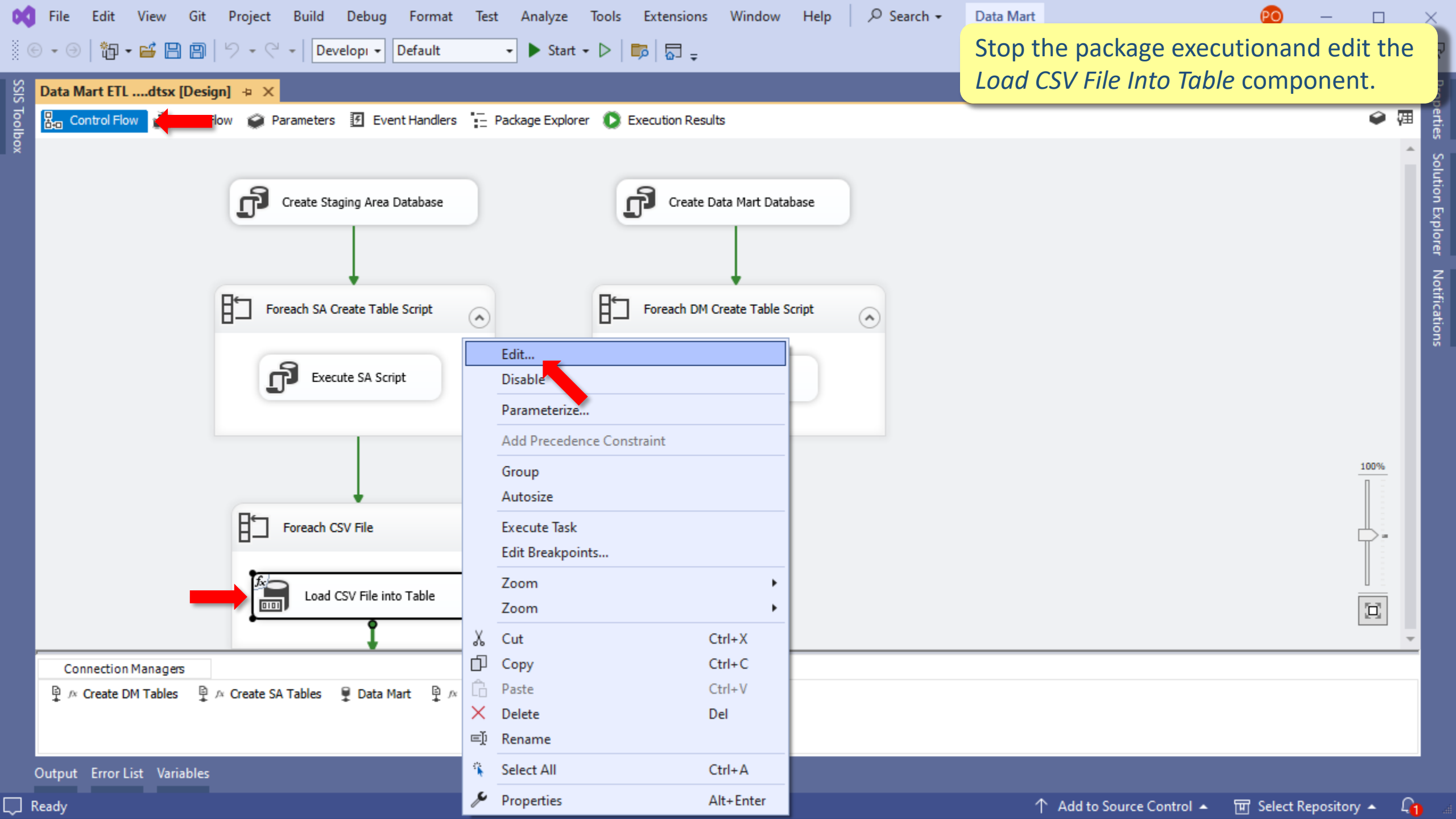
Call StackBreakpointsException SettingsCommand WindowImmediate WindowOutputAutosLocalsWatch 1

Ready

Add to Source Control

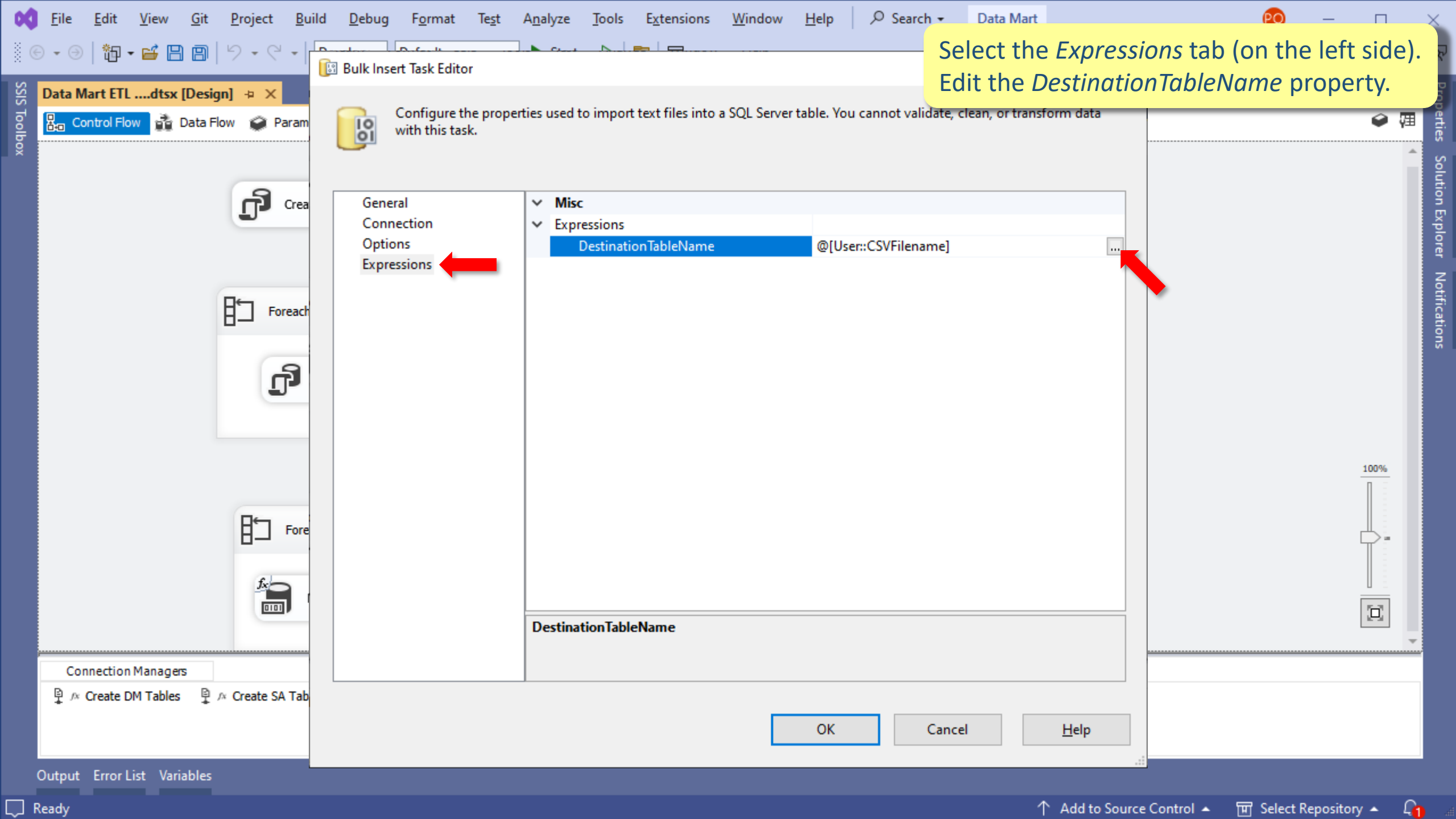
In the *Execution Results* tab or *Progress* tab is possible to see the cause of the failure.

"Transaction" in a SQL reserved word, so it cannot be used as name of a (destination) table.



Stop the package execution and edit the *Load CSV File Into Table* component.

- Edit...
- Disable
- Parameterize...
- Add Precedence Constraint
- Group
- Autosize
- Execute Task
- Edit Breakpoints...
- Zoom
- Zoom
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Delete Del
- Rename
- Select All Ctrl+A
- Properties Alt+Enter



Select the *Expressions* tab (on the left side).
Edit the *DestinationTableName* property.

General
Connection
Options
Expressions

Misc
Expressions
DestinationTableName

@[User::CSVFilename]

DestinationTableName

OK

Cancel

Help

Connection Managers

Create DM Tables

Create SA Tables

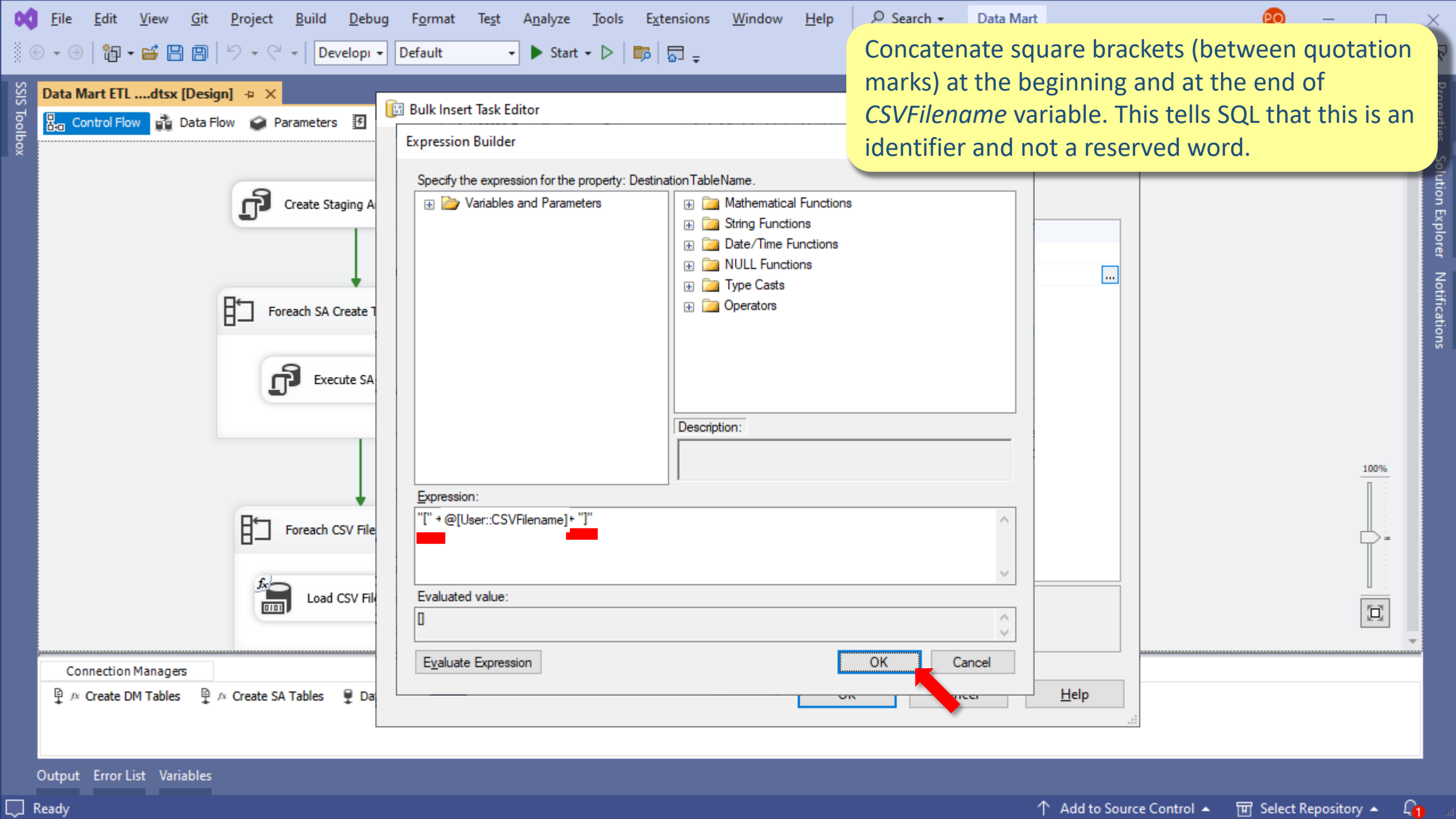
Output
Error List
Variables

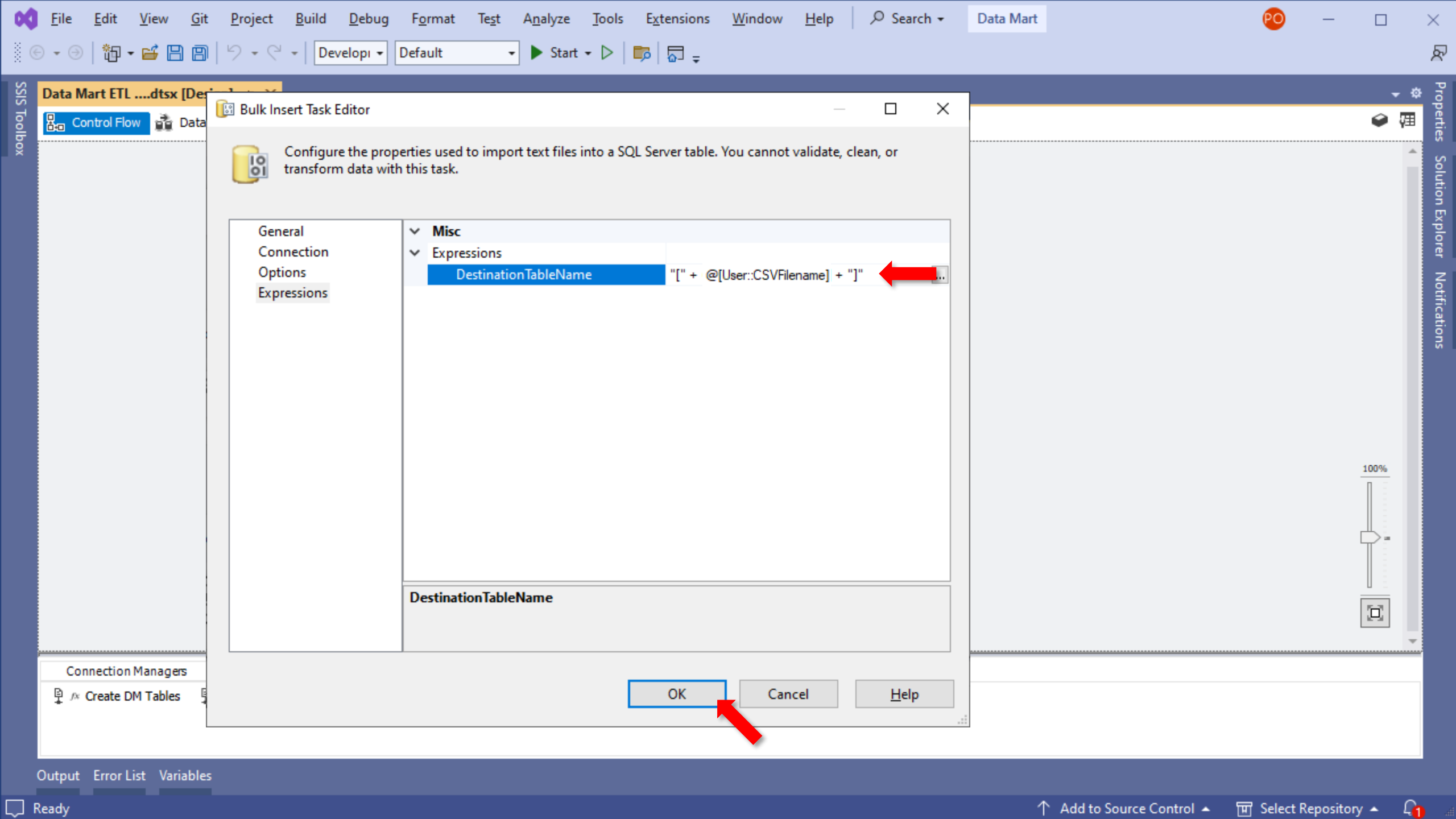
Ready

Add to Source Control

Select Repository

1





FileEditViewGitProjectBuildDebugFormatTestAnalyzeToolsExtensionsWindowHelp

SearchData Mart

DevelopiDefaultStart

SSIS Toolbox

Data Mart ETL...dtsx [Design]*

Control FlowData FlowParametersEvent HandlersPackage ExplorerExecution Results

Create Staging Area Database

Foreach SA Create Table Script

Execute SA Script

Foreach CSV File

Load CSV File into Table

Create Data Mart Database

Foreach DM Create Table Script

Execute DM Script

Connection Managers

Create DM Tables

Create SA Tables

Data Mart

Load CSV Files

master

Staging Area

OutputError ListVariables

Ready

Add to Source ControlSelect Repository

Execute the full package (i.e., from the beginning).

FileEditViewGitProjectBuildDebugFormatTestAnalyzeToolsExtensionsWindowHelp

SearchData Mart

Process: [1936] DtsDebugHost.exe

Lifecycle Events Thread:

Continue

Stack Frame:

Package execution should succeed by having a green checkmark in each component.

Data Mart ETLdtsx [Design]

Control FlowData FlowParametersEvent HandlersPackage ExplorerProgress

Create Staging Area Database

Foreach SA Create Table Script

Execute SA Script

Create Data Mart Database

Foreach DM Create Table Script

Execute DM Script

Foreach CSV File

Load CSV File into Table

Connection Managers

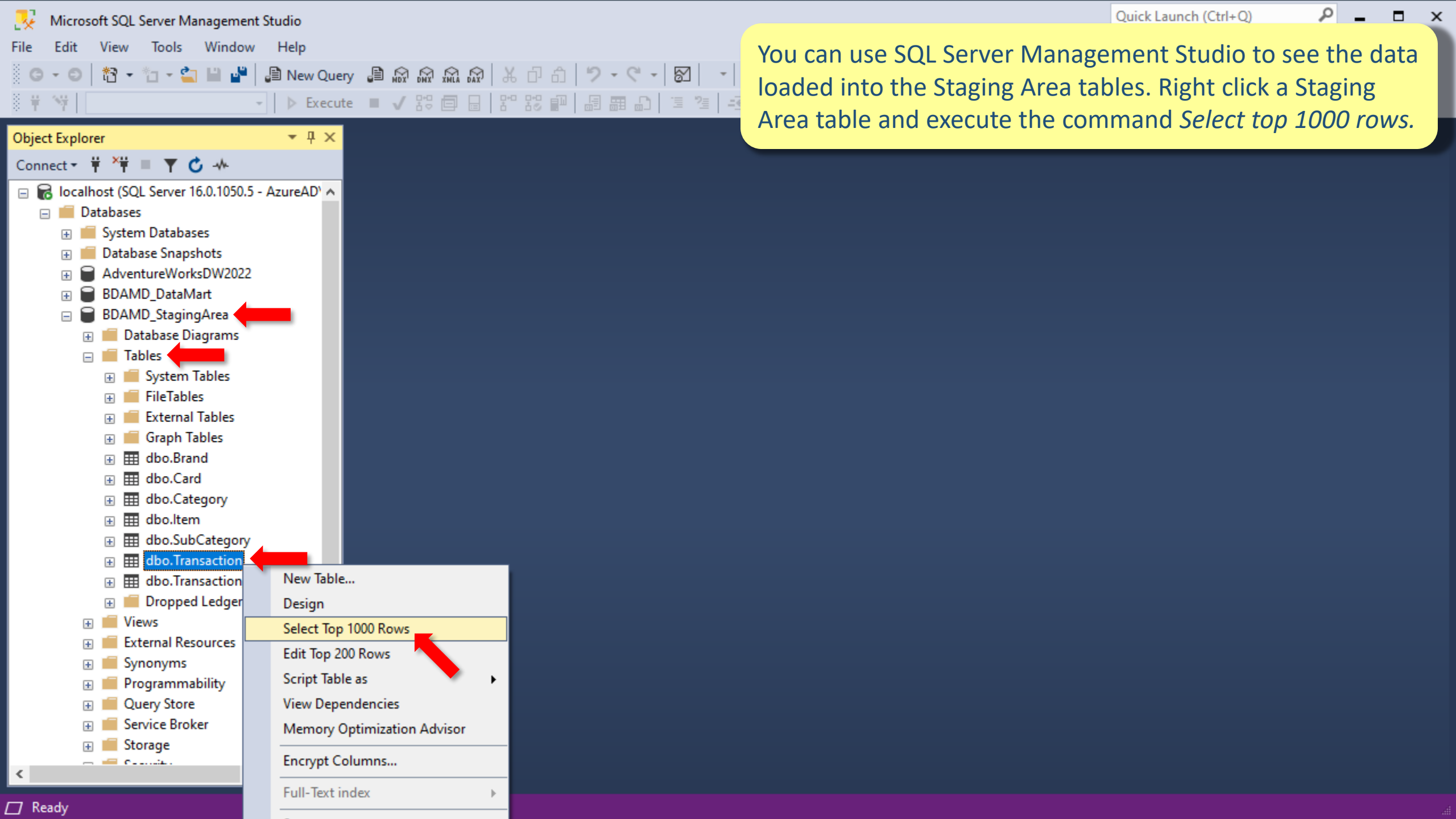
Create DM TablesCreate SA TablesData MartLoad CSV FilesmasterStaging Area

Package execution completed with success. Click here to switch to design mode, or select Stop Debugging from the Debug menu.

AutosLocalsWatch 1

Ready

Add to Source ControlSelect Repository



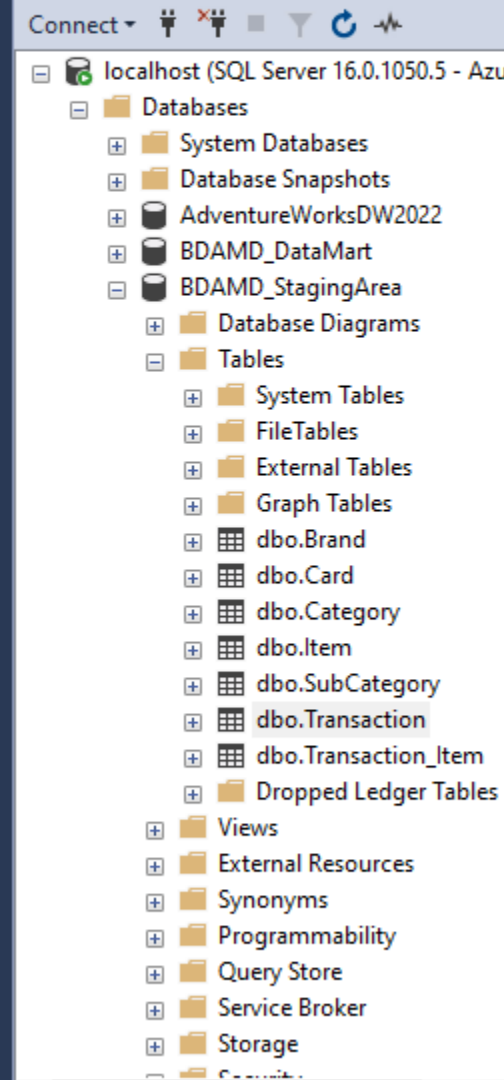
You can use SQL Server Management Studio to see the data loaded into the Staging Area tables. Right click a Staging Area table and execute the command *Select top 1000 rows*.

File Edit View Query Project Tools Window Help



As result you get the first 1000 rows of the table selected.

Object Explorer



SQLQuery1.sql - loc...\PauloOliveira (54))

```
SELECT TOP (1000) [Date]
      ,[TransactionID]
      ,[CardID]
FROM [BDAMD_StagingArea].[dbo].[Transaction]
```

100 %

Results Messages

	Date	TransactionID	CardID
1	2001-11-08 00:00:00.000	T10000208	C0103317864
2	2001-02-02 00:00:00.000	T10000226	C0103318107
3	2001-01-17 00:00:00.000	T10000419	C0103318121
4	2001-10-03 00:00:00.000	T10000065	C0103318289
5	2001-04-15 00:00:00.000	T10000214	C0103318452
6	2001-05-09 00:00:00.000	T10000058	C0103318497
7	2001-02-26 00:00:00.000	T10000387	C0103318558
8	2001-04-03 00:00:00.000	T10000034	C0103318558
9	2001-03-08 00:00:00.000	T10000374	C0103318579
10	2001-07-23 00:00:00.000	T10000284	C0103318579
11	2001-02-18 00:00:00.000	T10000125	C0103318601

Query executed successfully.

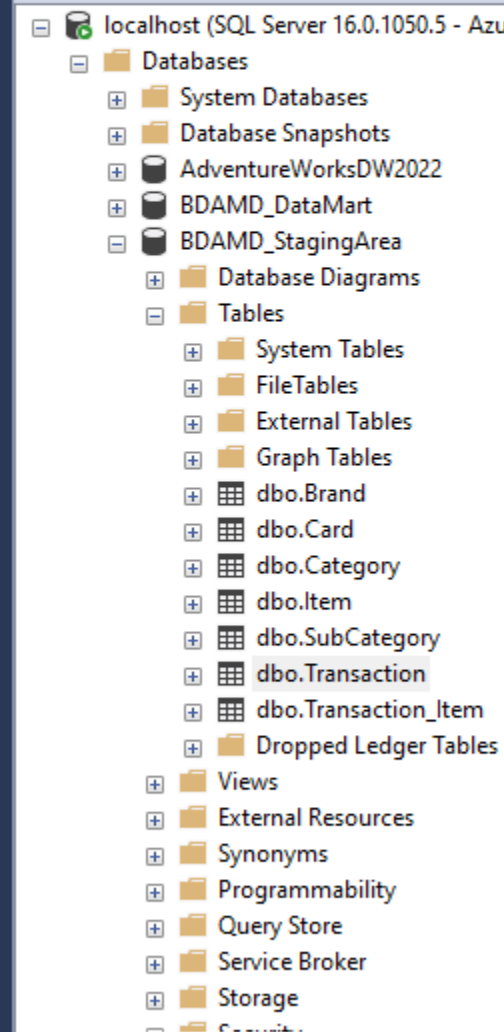
localhost (16.0 RTM) | AzureAD\PauloOliveira ... | BDAMD_StagingArea | 00:00:00 | 1 000 rows

File Edit View Query Project Tools Window Help



Object Explorer

Connect



SQLQuery1.sql - loc...\PauloOliveira (54))*

```
SELECT *  
FROM [BDAMD_StagingArea].[dbo].[Transaction]
```

100 %

Results Messages

	Date	TransactionID	CardID
1	2001-11-08 00:00:00.000	T10000208	C0103317864
2	2001-02-02 00:00:00.000	T10000226	C0103318107
3	2001-01-17 00:00:00.000	T10000419	C0103318121
4	2001-10-03 00:00:00.000	T10000065	C0103318289
5	2001-04-15 00:00:00.000	T10000214	C0103318452
6	2001-05-09 00:00:00.000	T10000058	C0103318497
7	2001-02-26 00:00:00.000	T10000387	C0103318558
8	2001-04-03 00:00:00.000	T10000034	C0103318558
9	2001-03-08 00:00:00.000	T10000374	C0103318579
10	2001-07-23 00:00:00.000	T10000284	C0103318579
11	2001-02-18 00:00:00.000	T10000125	C0103318601

Query executed successfully.

localhost (16.0 RTM) | AzureAD\PauloOliveira ... | BDAMD_StagingArea | 00:00:00 | 28 264 rows

