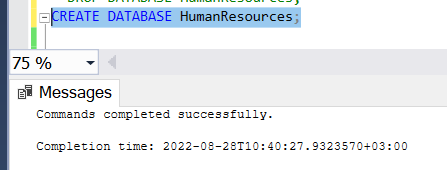
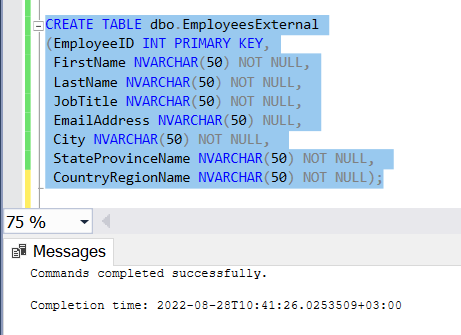
Lab report #3

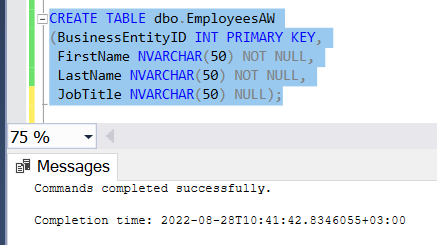
Sadovskaya Veronika

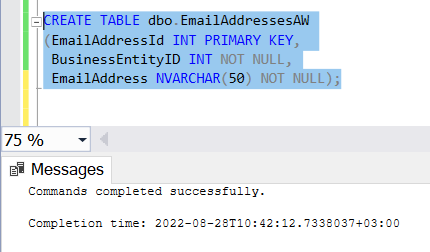
GitHub: <https://github.com/sdveronika/DataMola22>

**Task 1 - Creating new database for HR department**

****

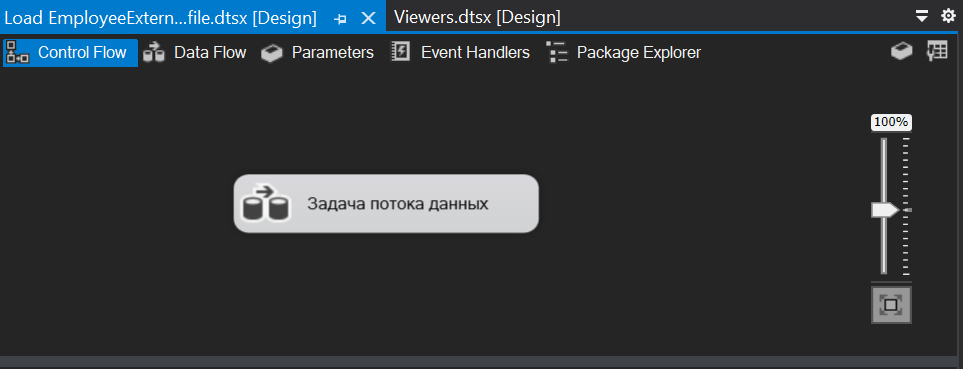
****

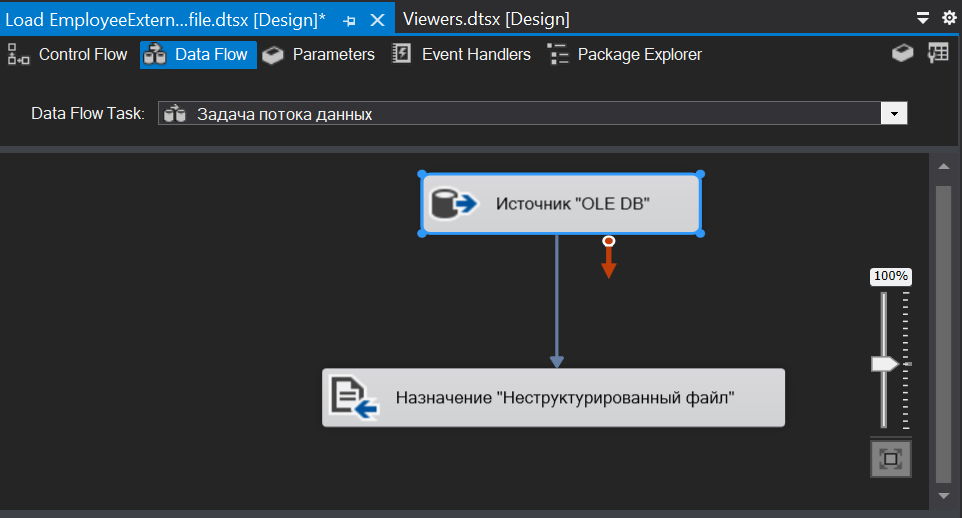
****

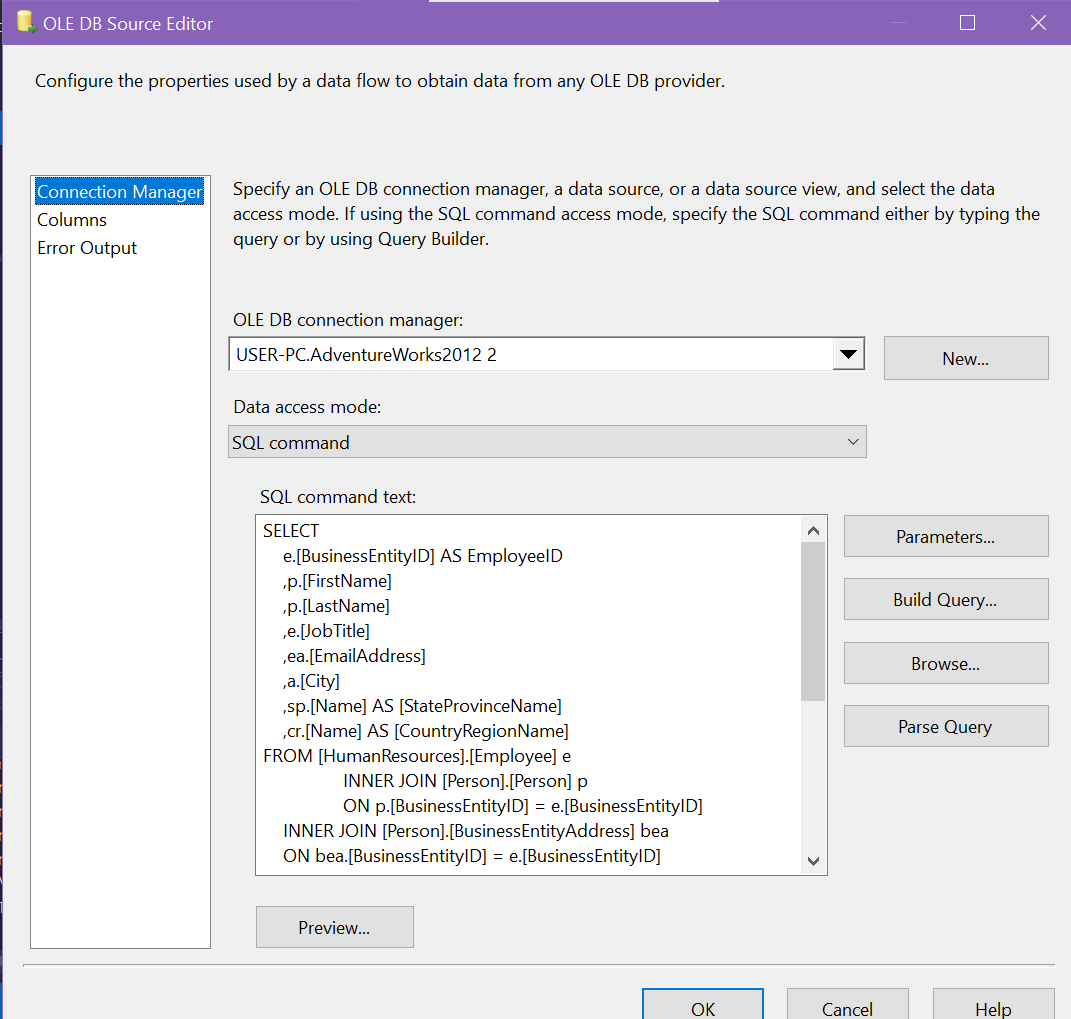
****

**Task 2 - Creating ETL solution to import employees from different sources**

**Note:** To perform the following tasks, we need files that store information for the dbo.HumanResources table. Since no data files were included with the lab, I created a package that loads all the necessary information into a file using tables from the AdwentureWorks2012 database. Next, I manually created test files, into which I copied part of the data from the generated file and pasted it into new ones, after deleting the copied information from the original file, since the primary key constraint was imposed on the EmployeeID column in the dbo.HumanResources table. Thus, the necessary list of files was obtained.







Sql command text to select all necessary employee info to load dbo.HumanResources table:

*SELECT*

*e.[BusinessEntityID] AS EmployeeID*

*,p.[FirstName]*

*,p.[LastName]*

*,e.[JobTitle]*

*,ea.[EmailAddress]*

*,a.[City]*

*,sp.[Name] AS [StateProvinceName]*

*,cr.[Name] AS [CountryRegionName]*

*FROM [HumanResources].[Employee] e*

*INNER JOIN [Person].[Person] p*

*ON p.[BusinessEntityID] = e.[BusinessEntityID]*

*INNER JOIN [Person].[BusinessEntityAddress] bea*

*ON bea.[BusinessEntityID] = e.[BusinessEntityID]*

*INNER JOIN [Person].[Address] a*

*ON a.[AddressID] = bea.[AddressID]*

*INNER JOIN [Person].[StateProvince] sp*

*ON sp.[StateProvinceID] = a.[StateProvinceID]*

*INNER JOIN [Person].[CountryRegion] cr*

*ON cr.[CountryRegionCode] = sp.[CountryRegionCode]*

*LEFT OUTER JOIN [Person].[PersonPhone] pp*

*ON pp.BusinessEntityID = p.[BusinessEntityID]*

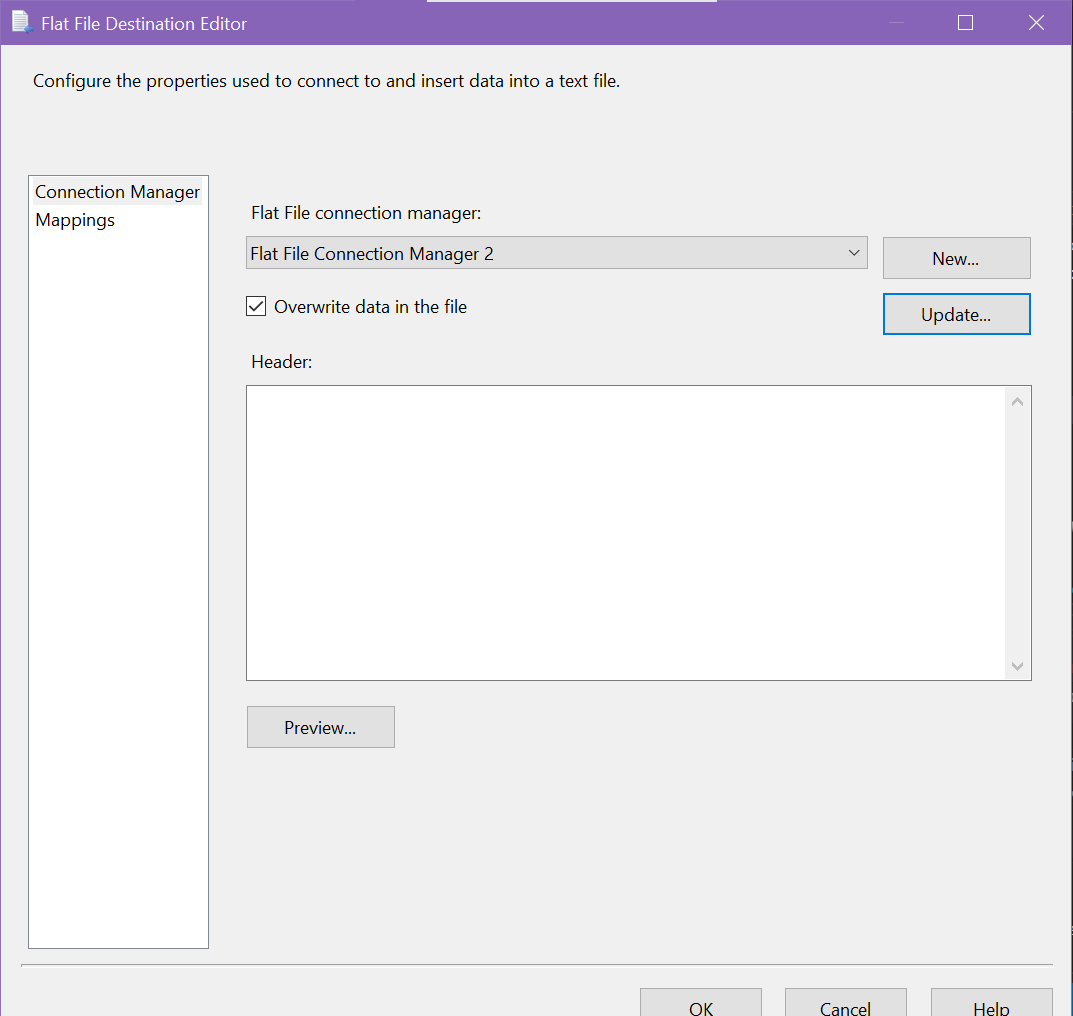
*LEFT OUTER JOIN [Person].[PhoneNumberType] pnt*

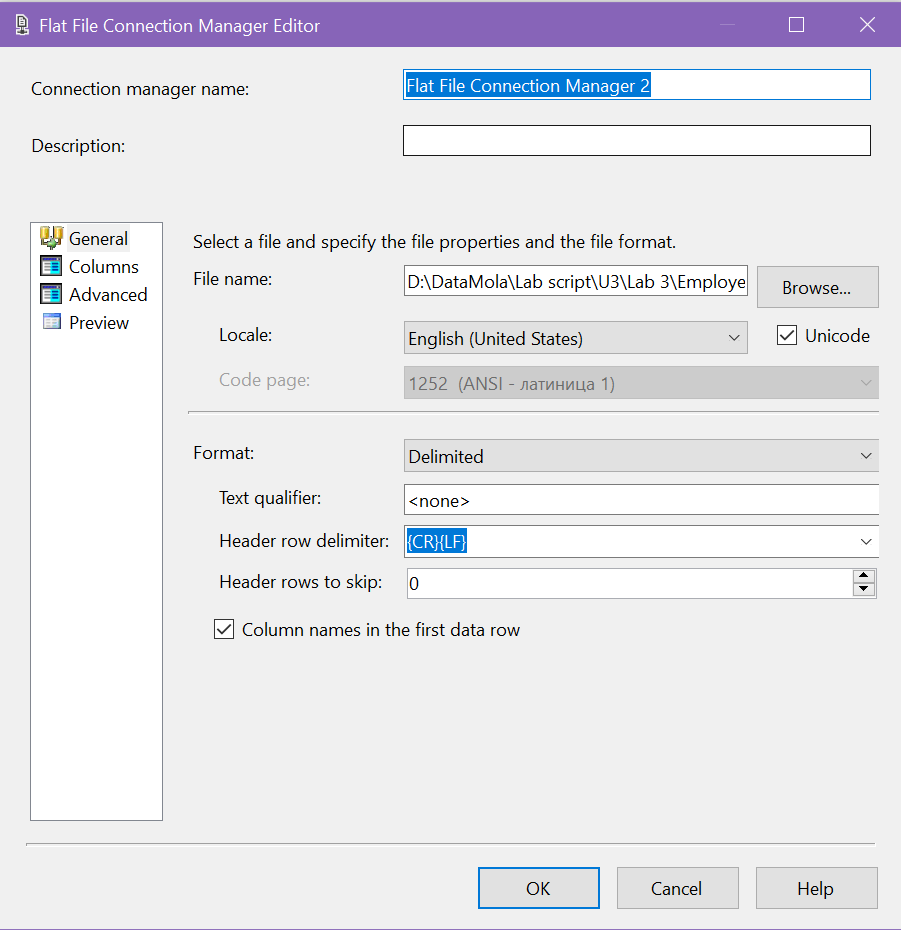
*ON pp.[PhoneNumberTypeID] = pnt.[PhoneNumberTypeID]*

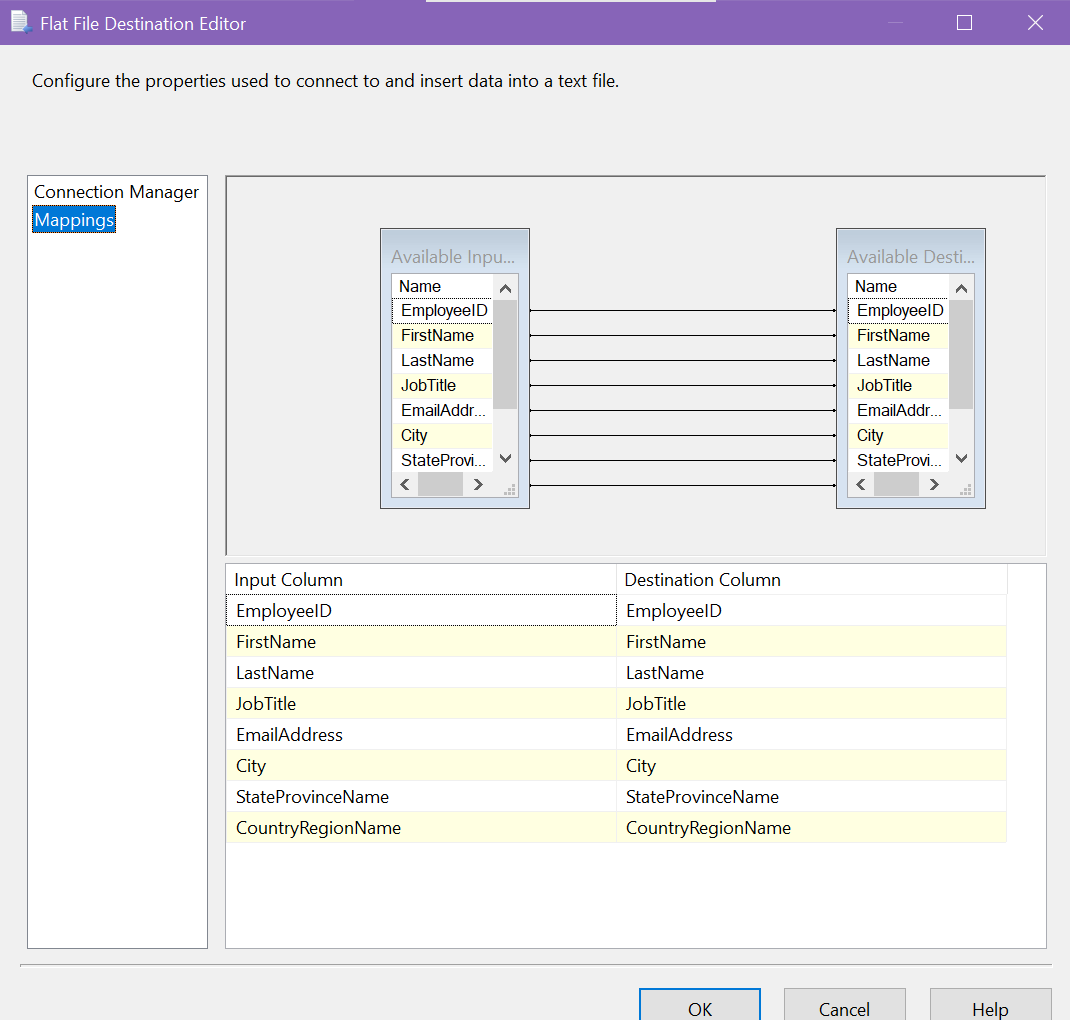
*LEFT OUTER JOIN [Person].[EmailAddress] ea*

*ON p.[BusinessEntityID] = ea.[BusinessEntityID]*

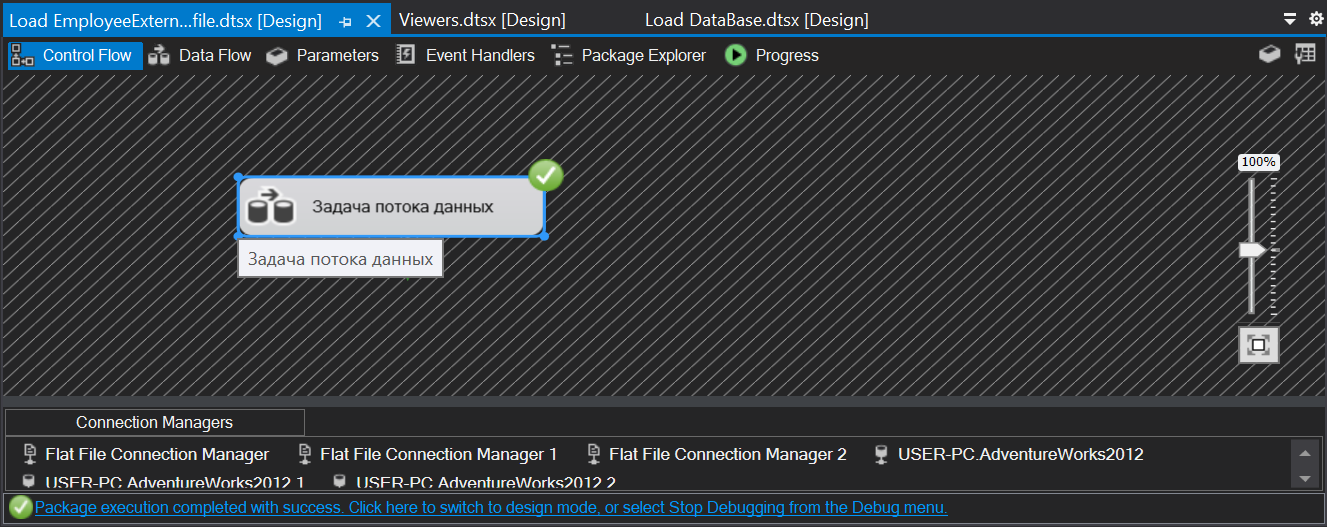
*ORDER BY p.[BusinessEntityID];*



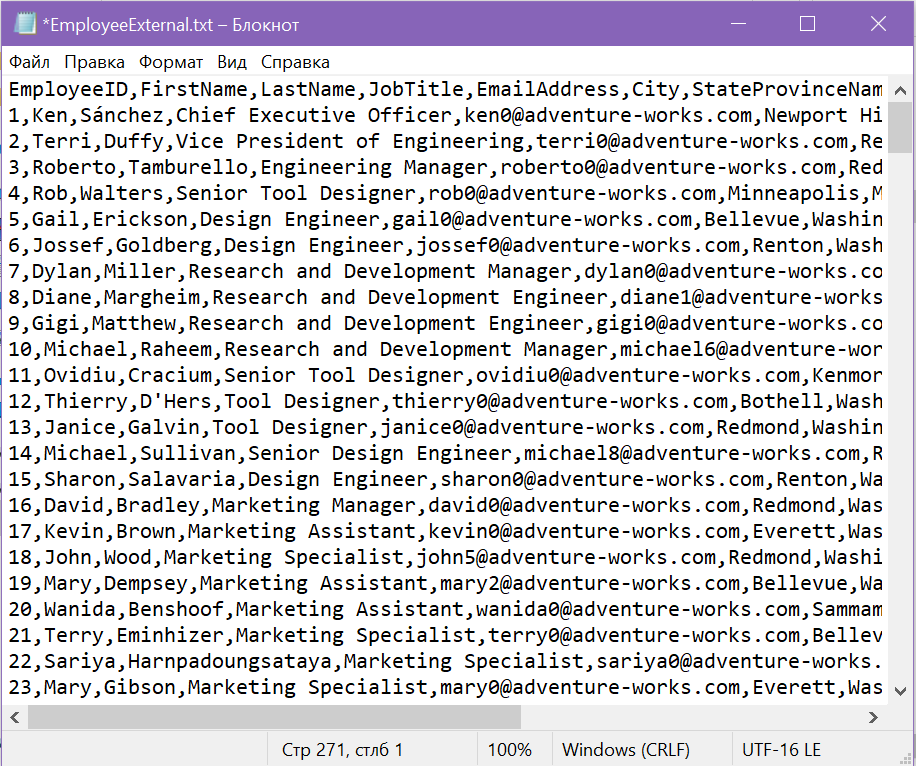




Run package to load EmployeeExternal text file:



Package execution result:

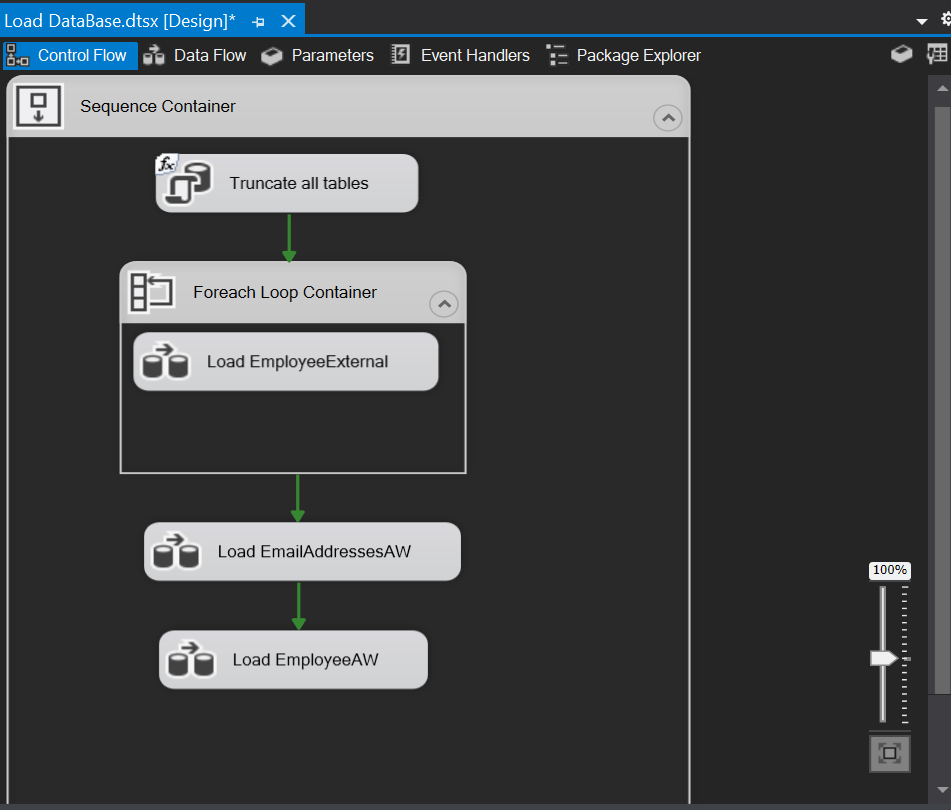


List of manually created files for task 2:

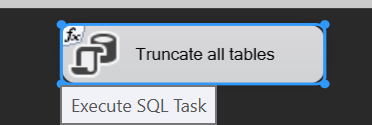


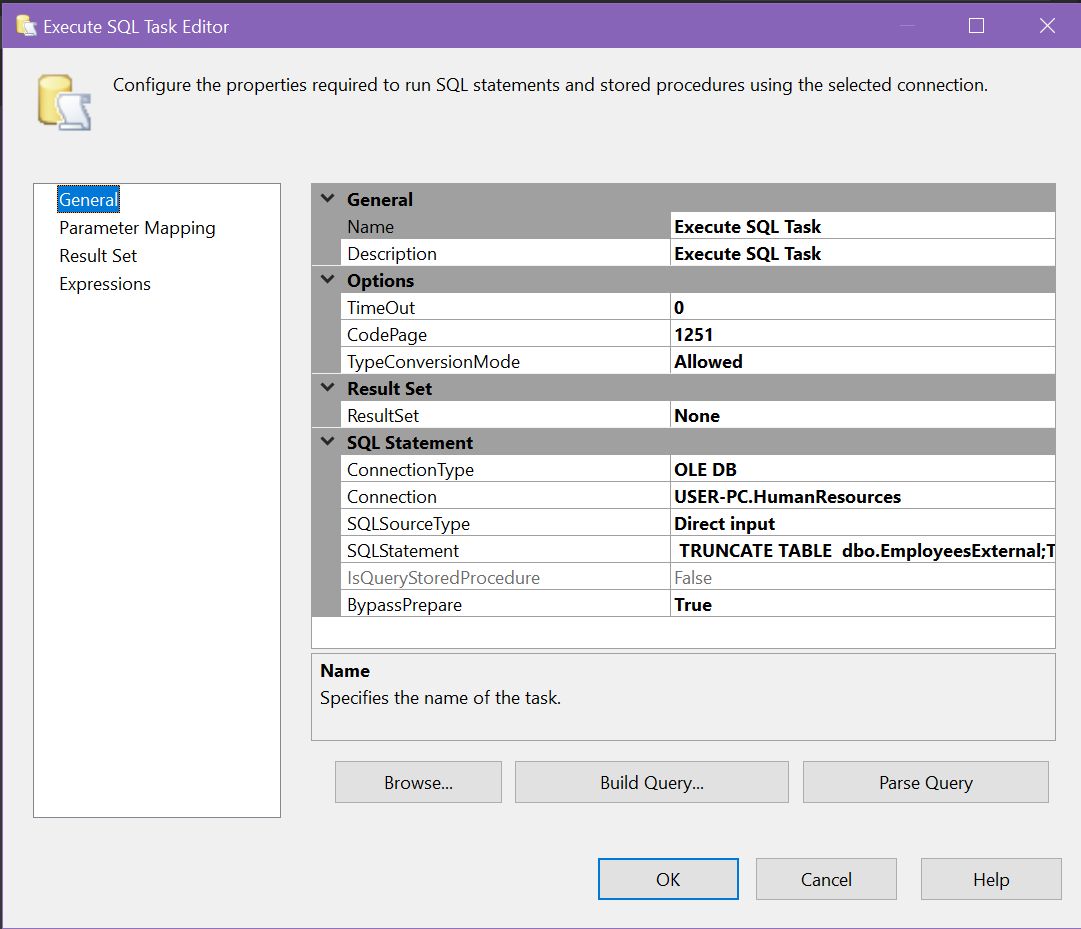
**Completing task 2:**

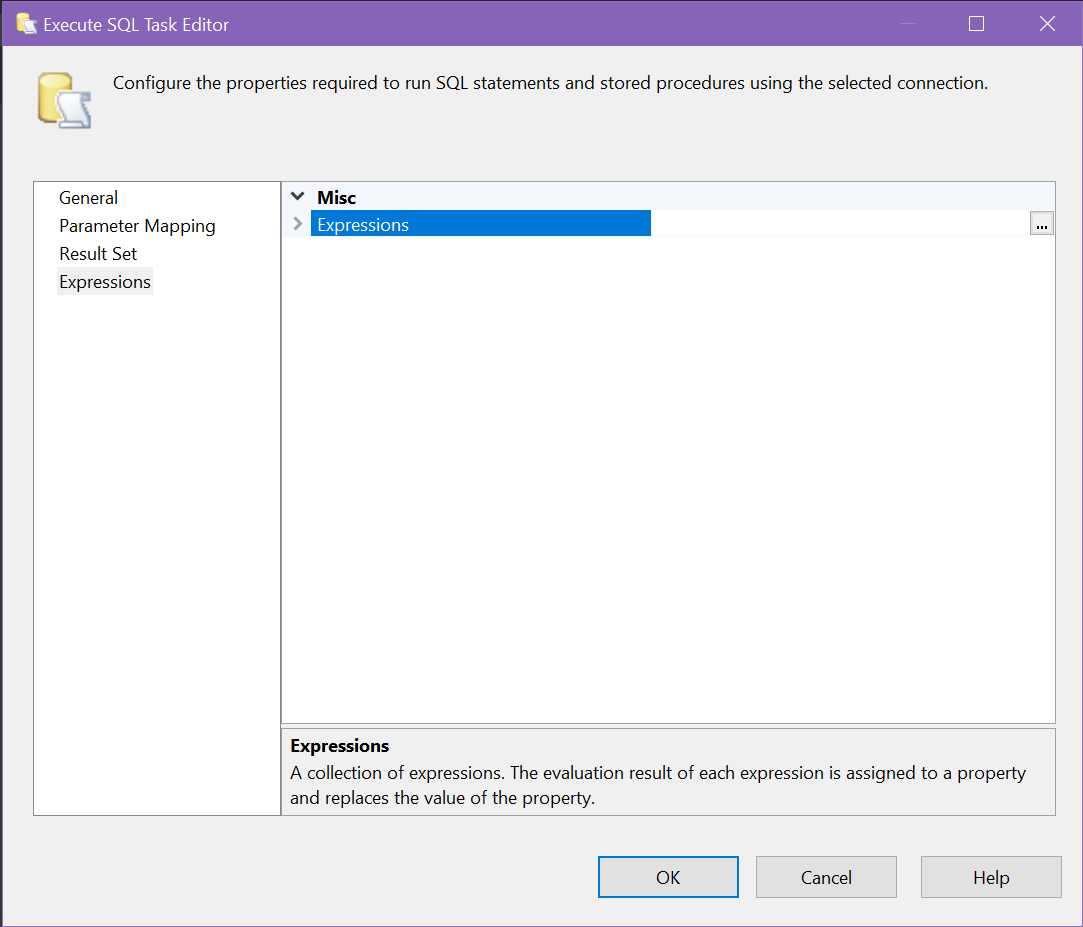
Control flow for load tables:

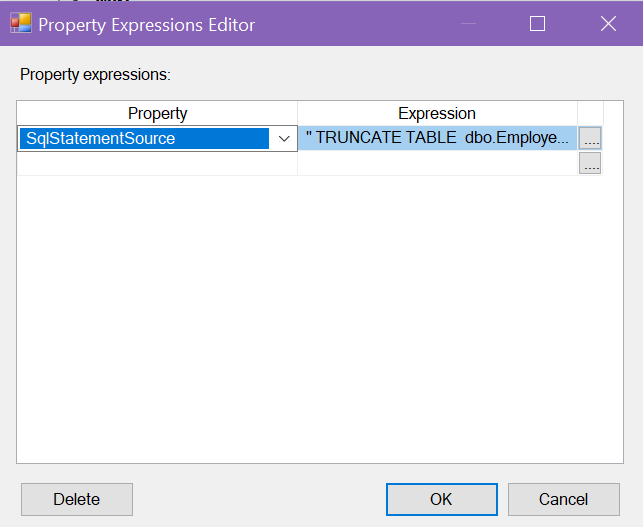
****

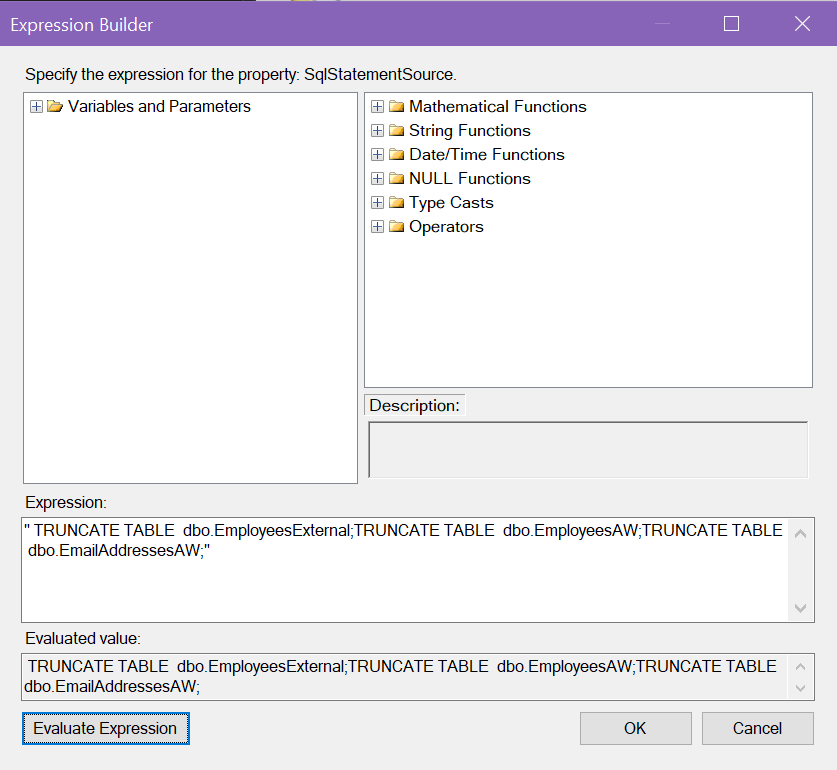
Execute SQL task configuration for truncate all tables before loading data:

****

****

****

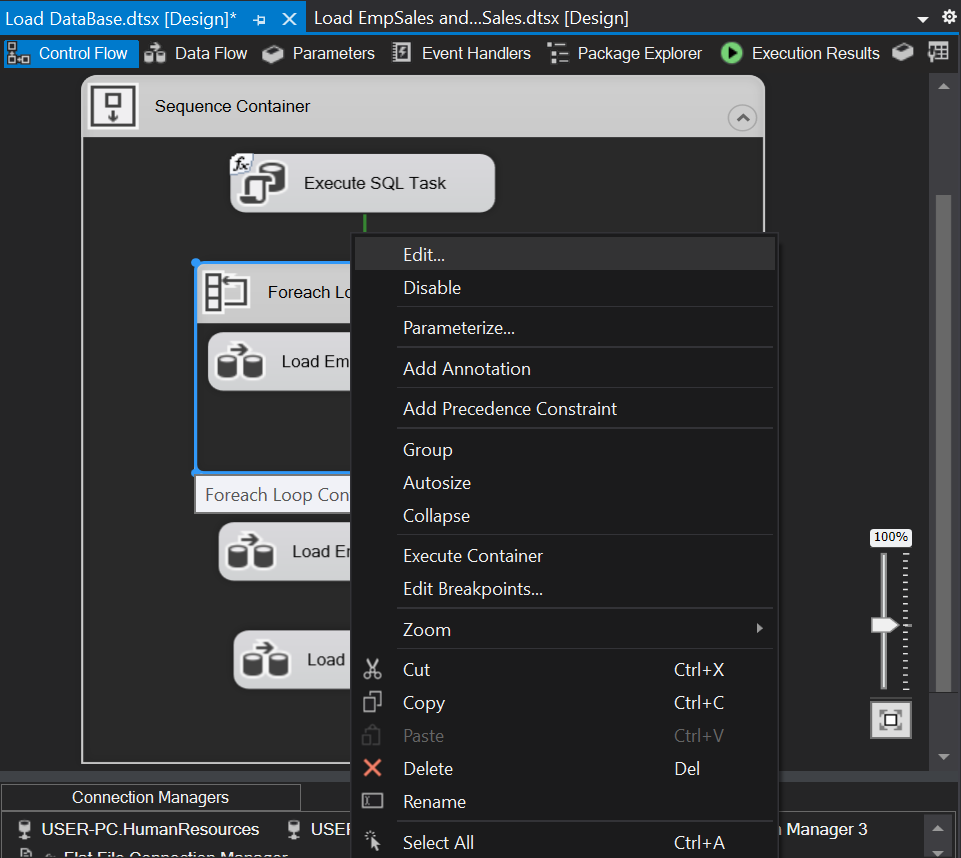
****

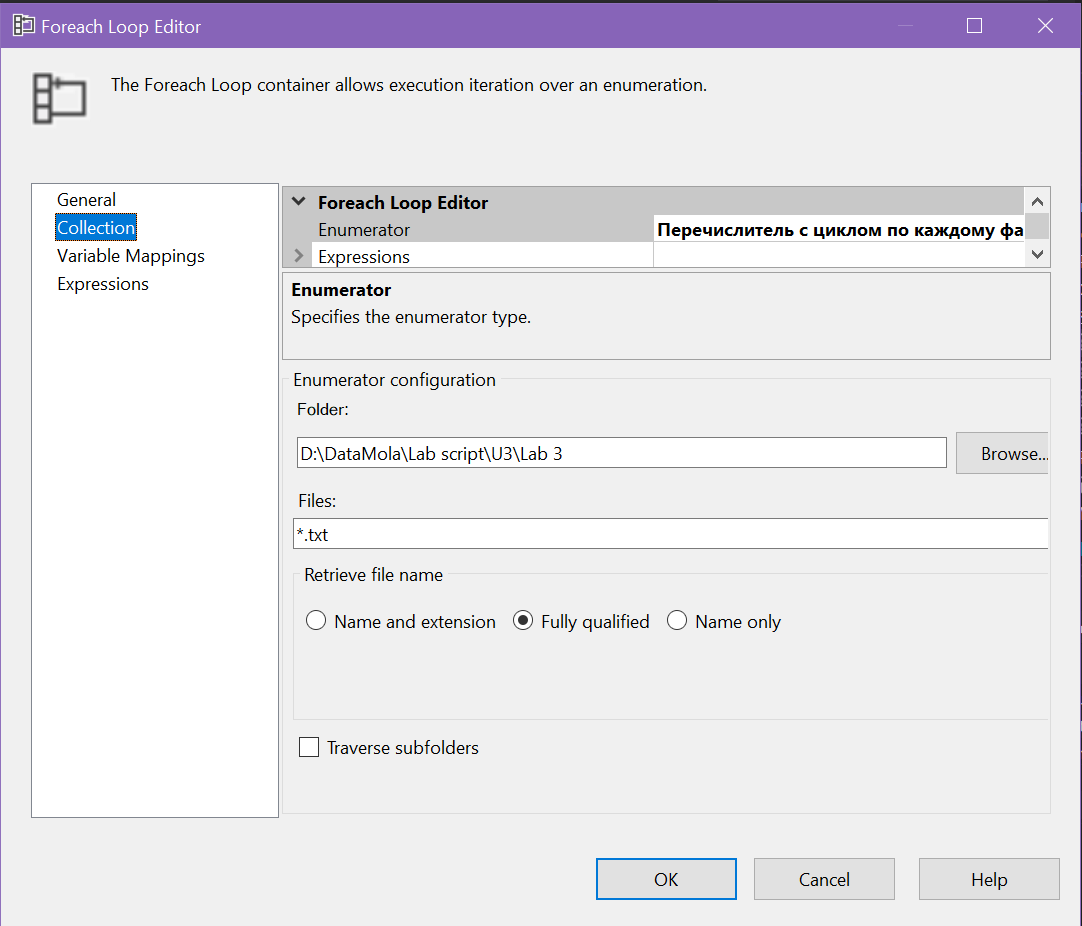
****

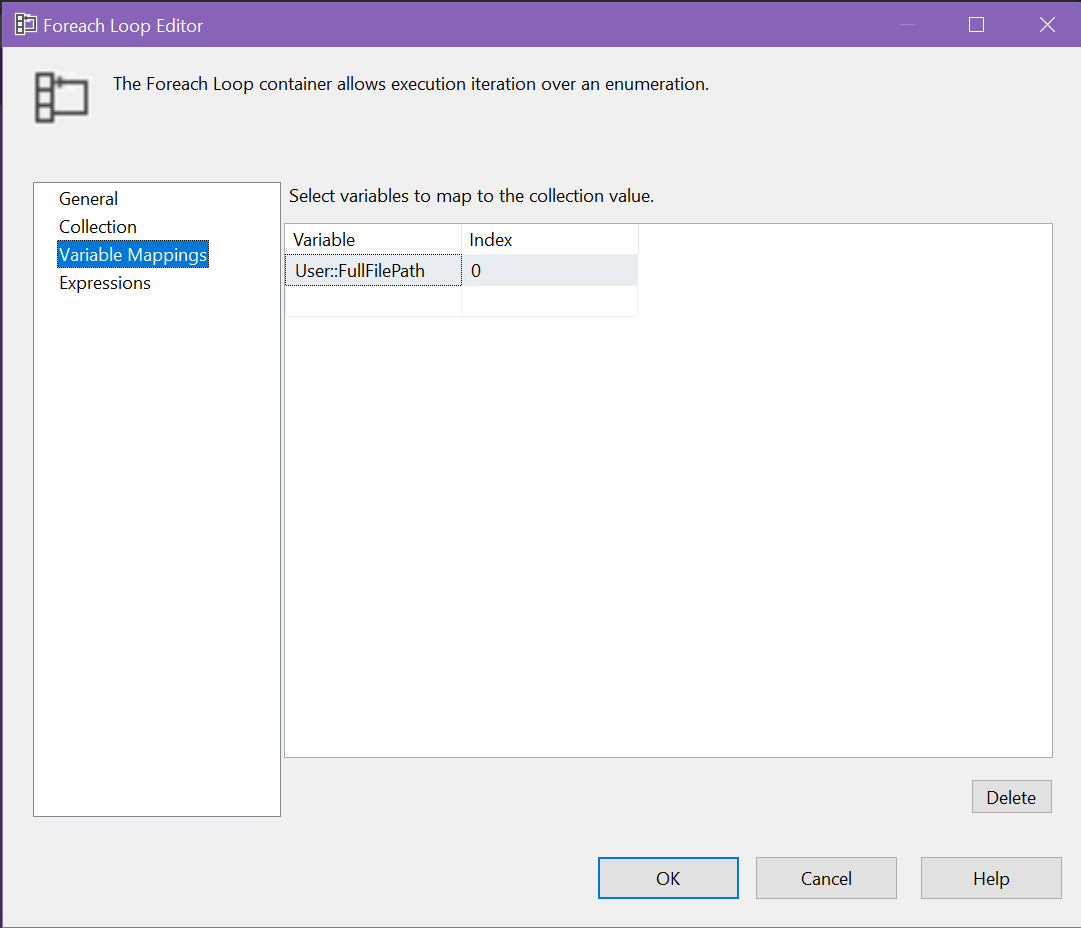
Add FullFilePath variable for using ForEach Loop container:

****

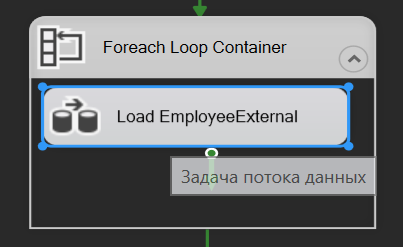
ForEach Loop container configuration to go through all files that store information about employees:

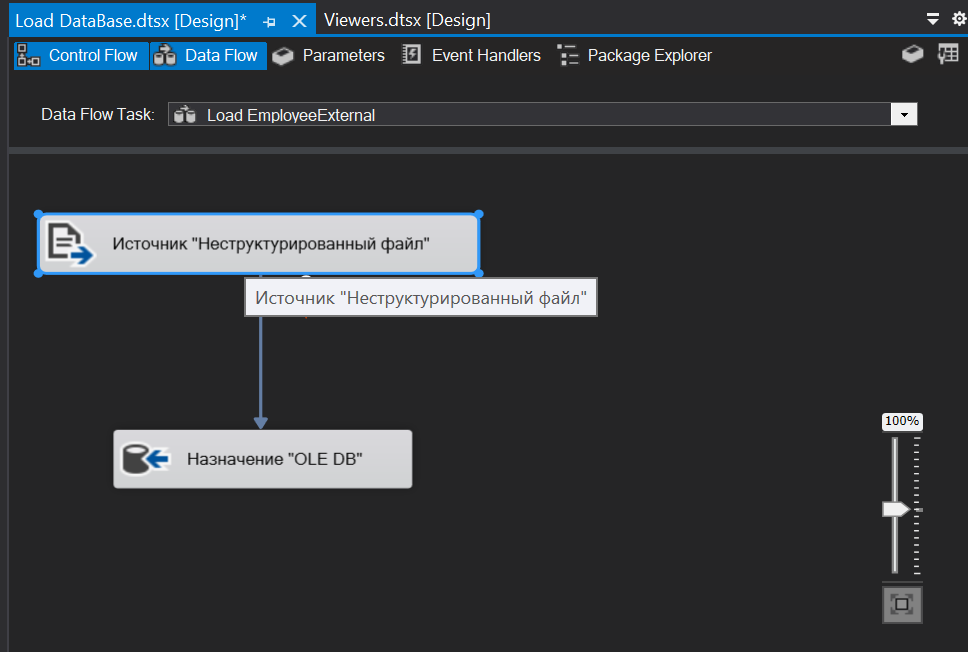
****

****

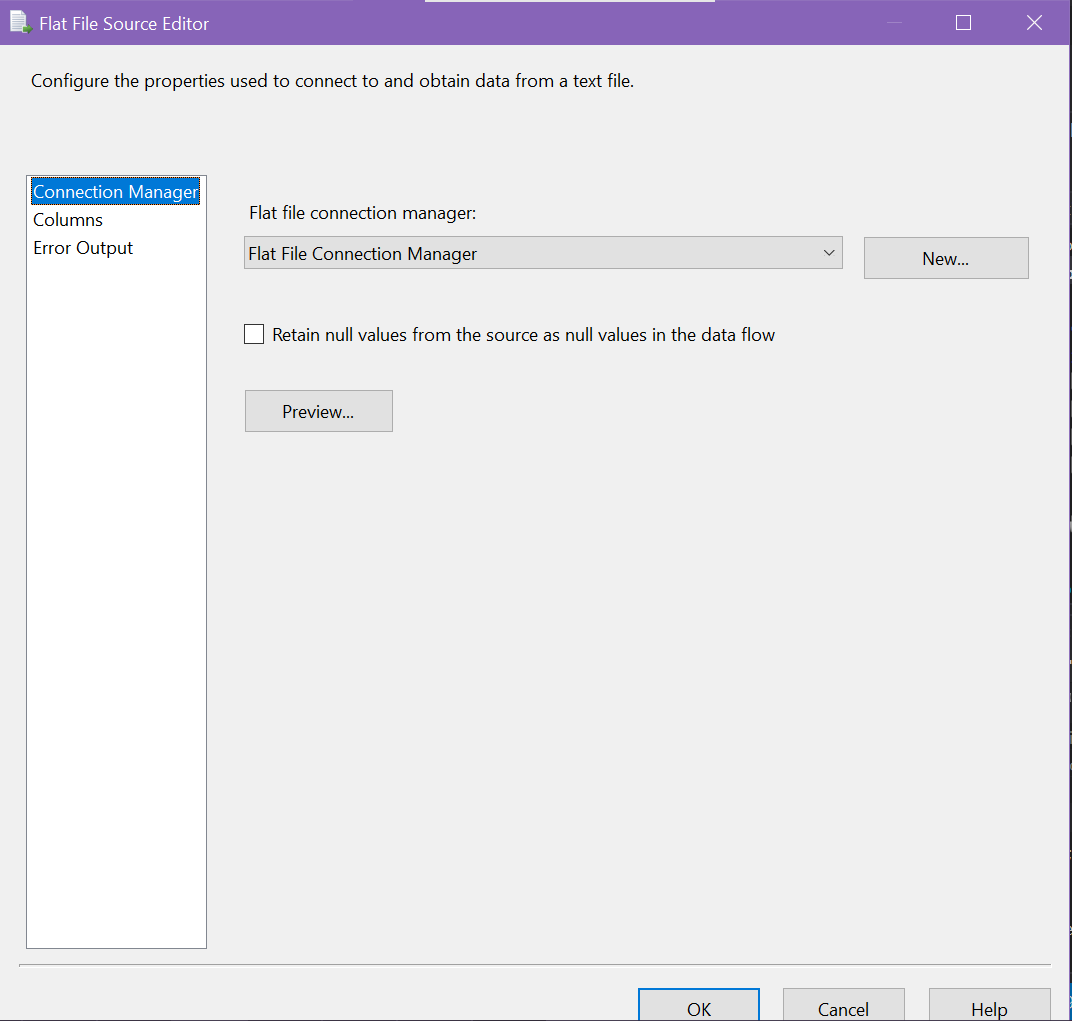
****

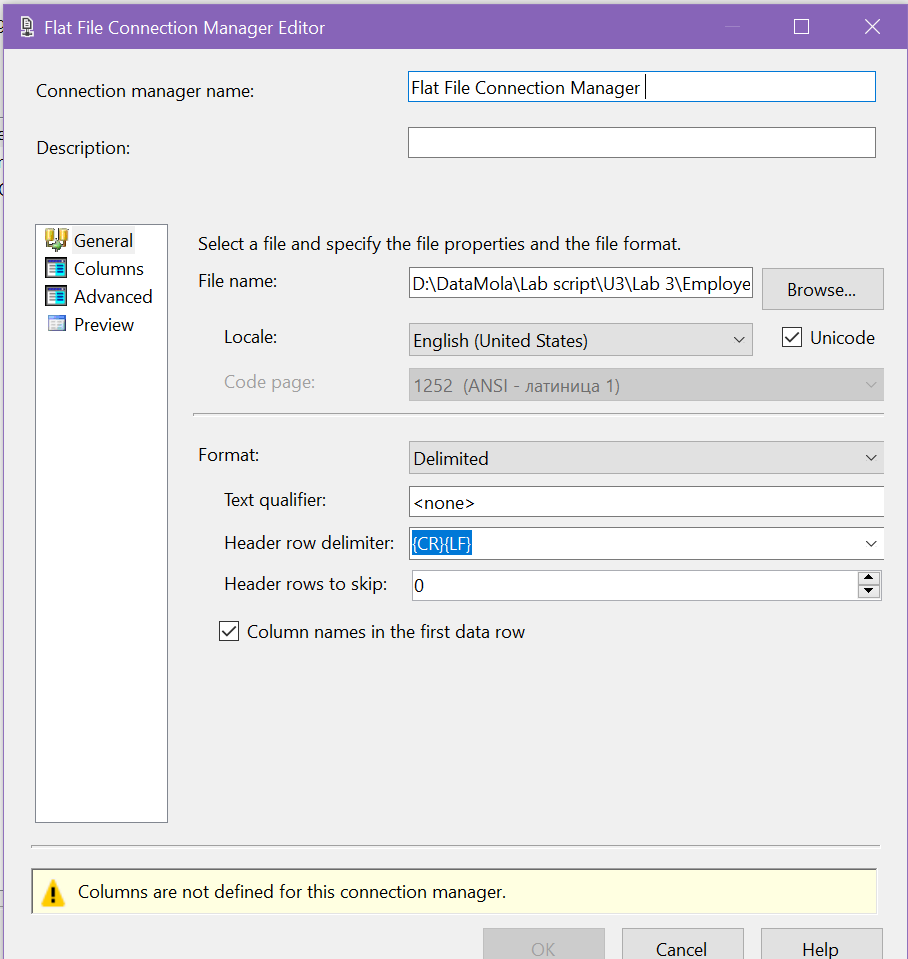
Load EmployeeExternal Data Flow:

****

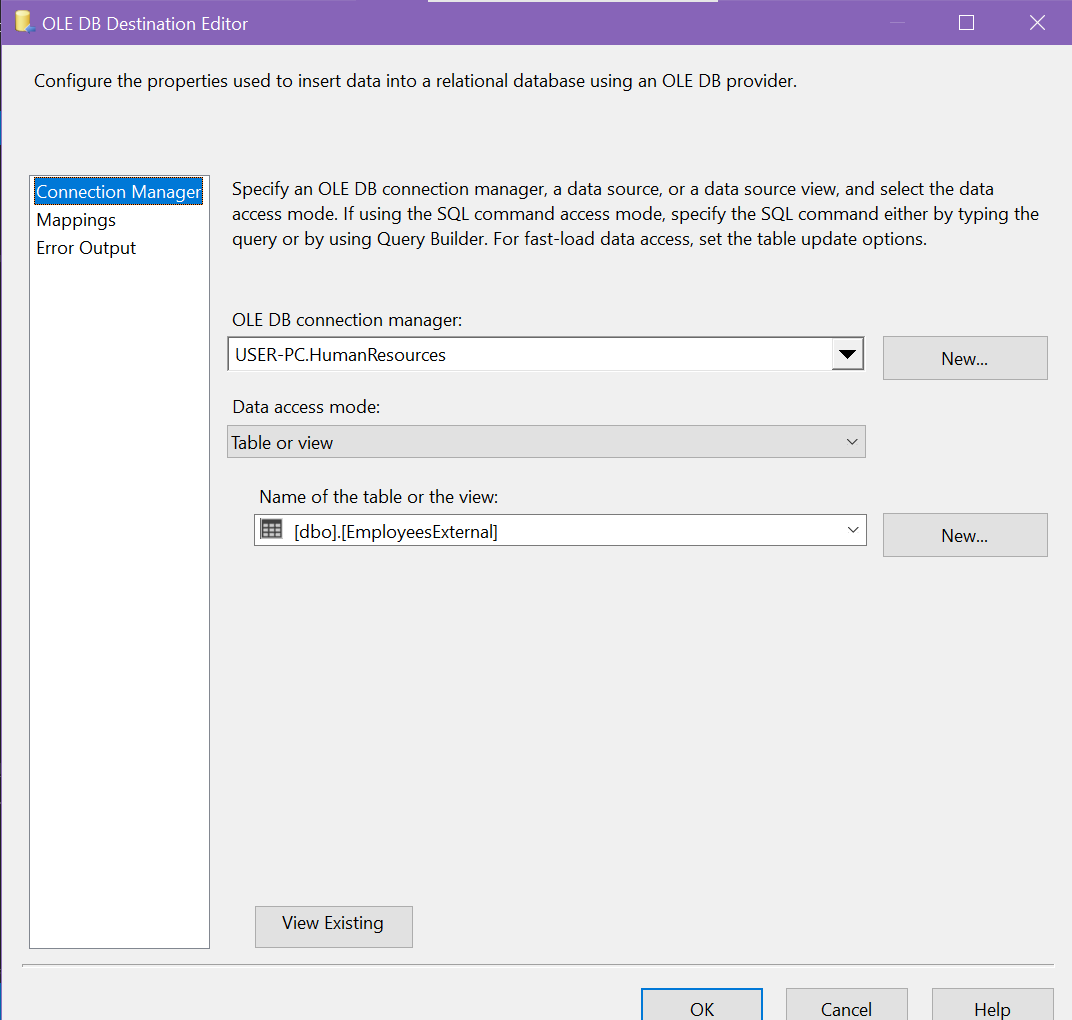
****

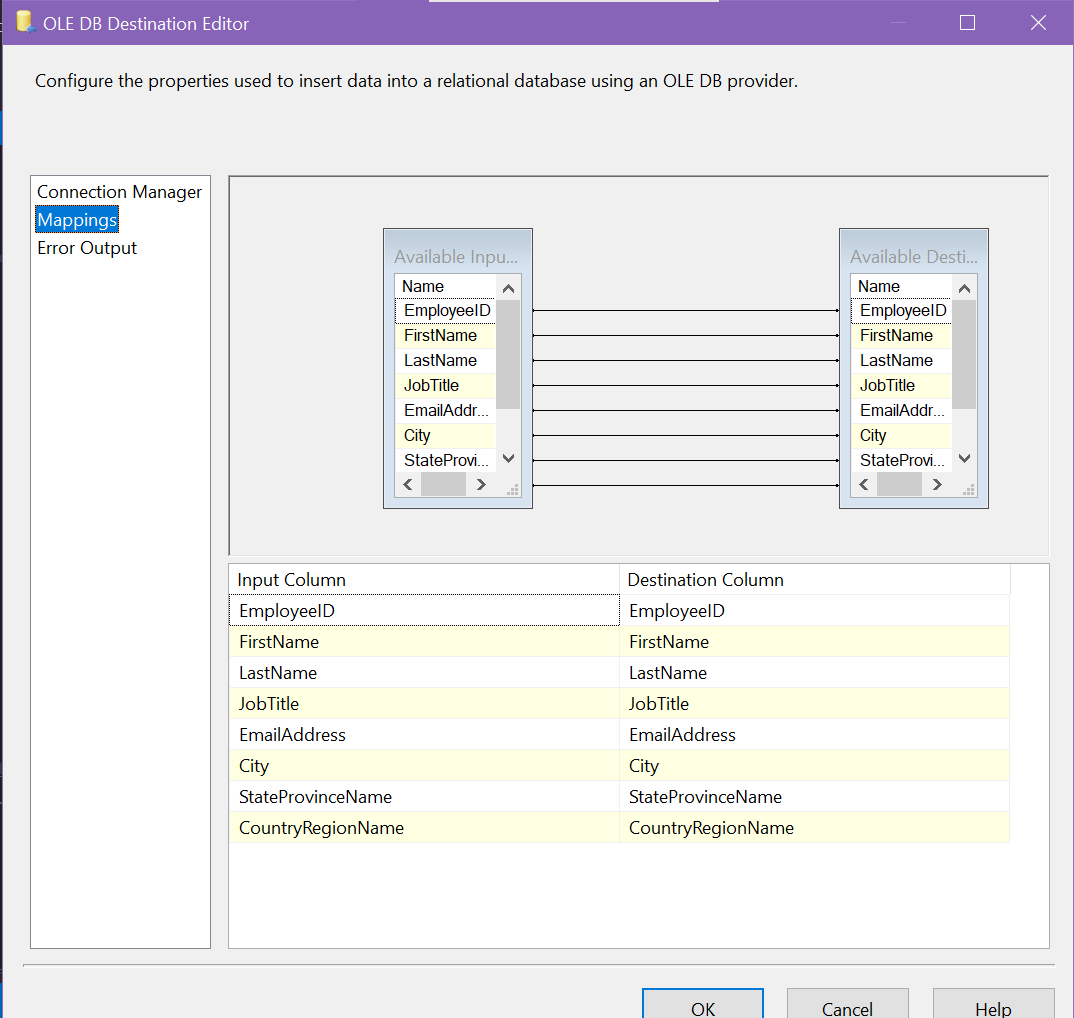
FlatFile Source configuration:

****

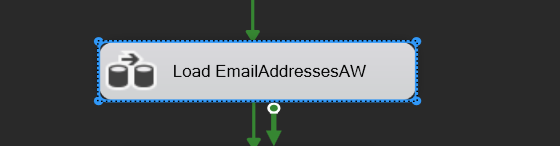
****

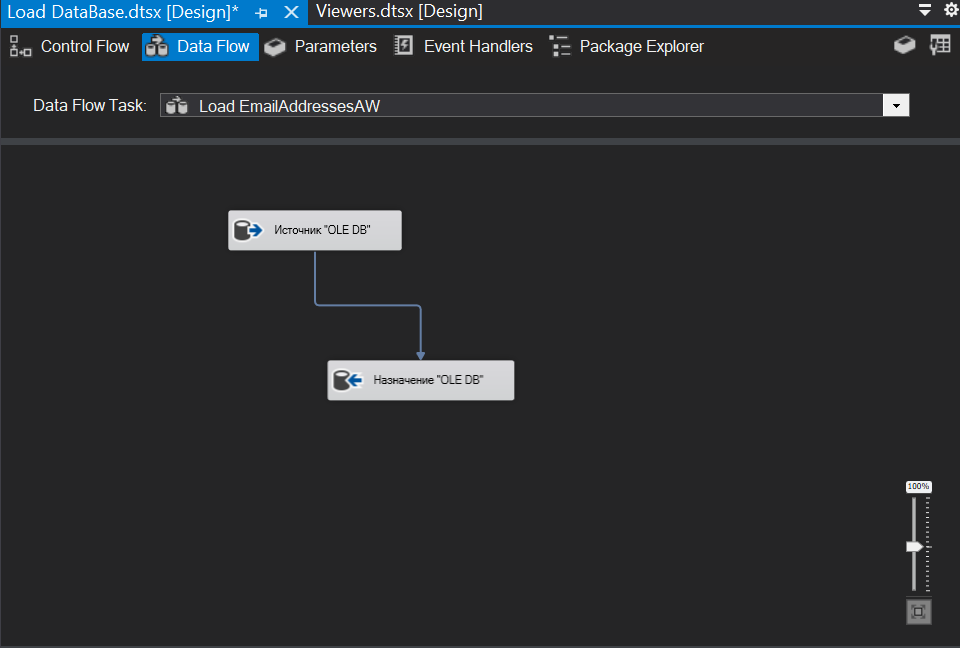
OLE DB destination configuration:

****

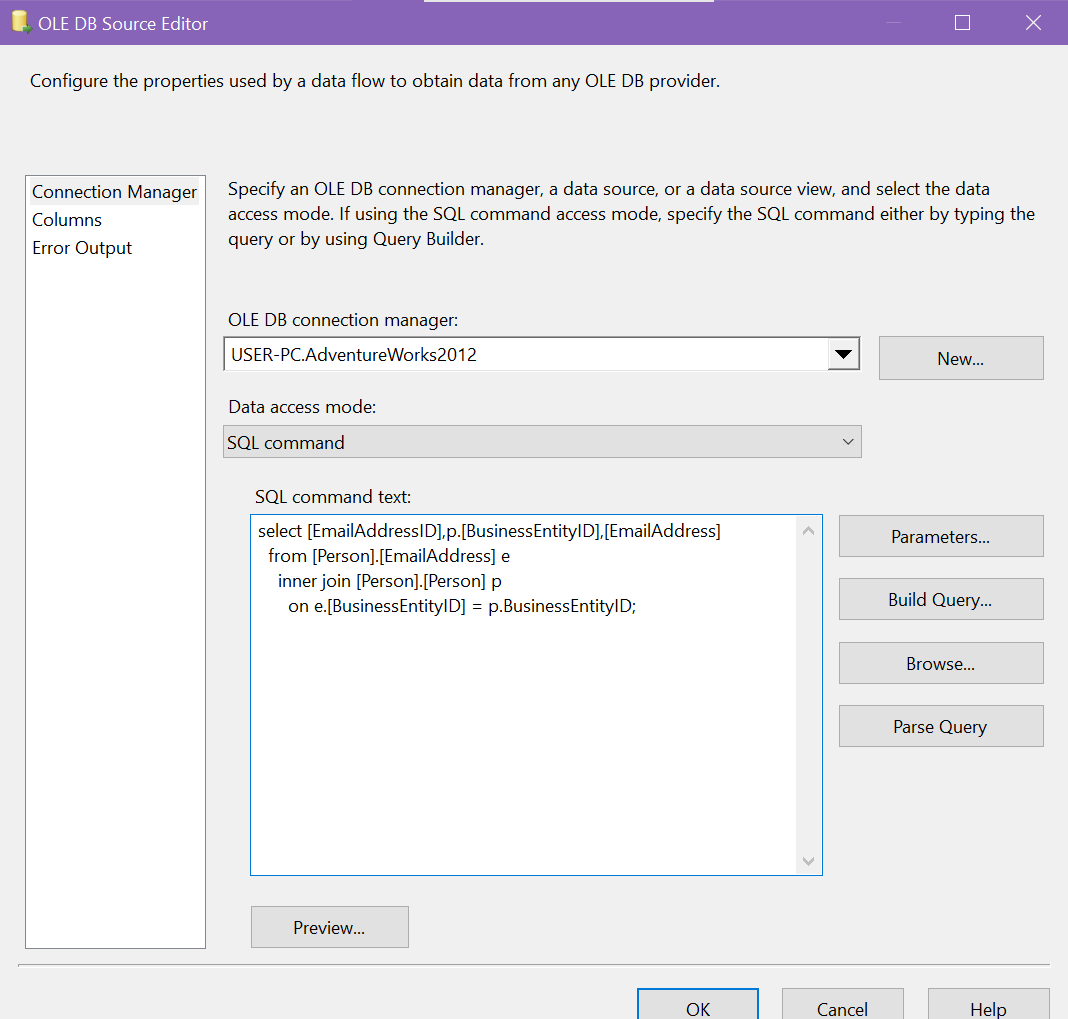
****

Load EmailAddressesAW configuration:

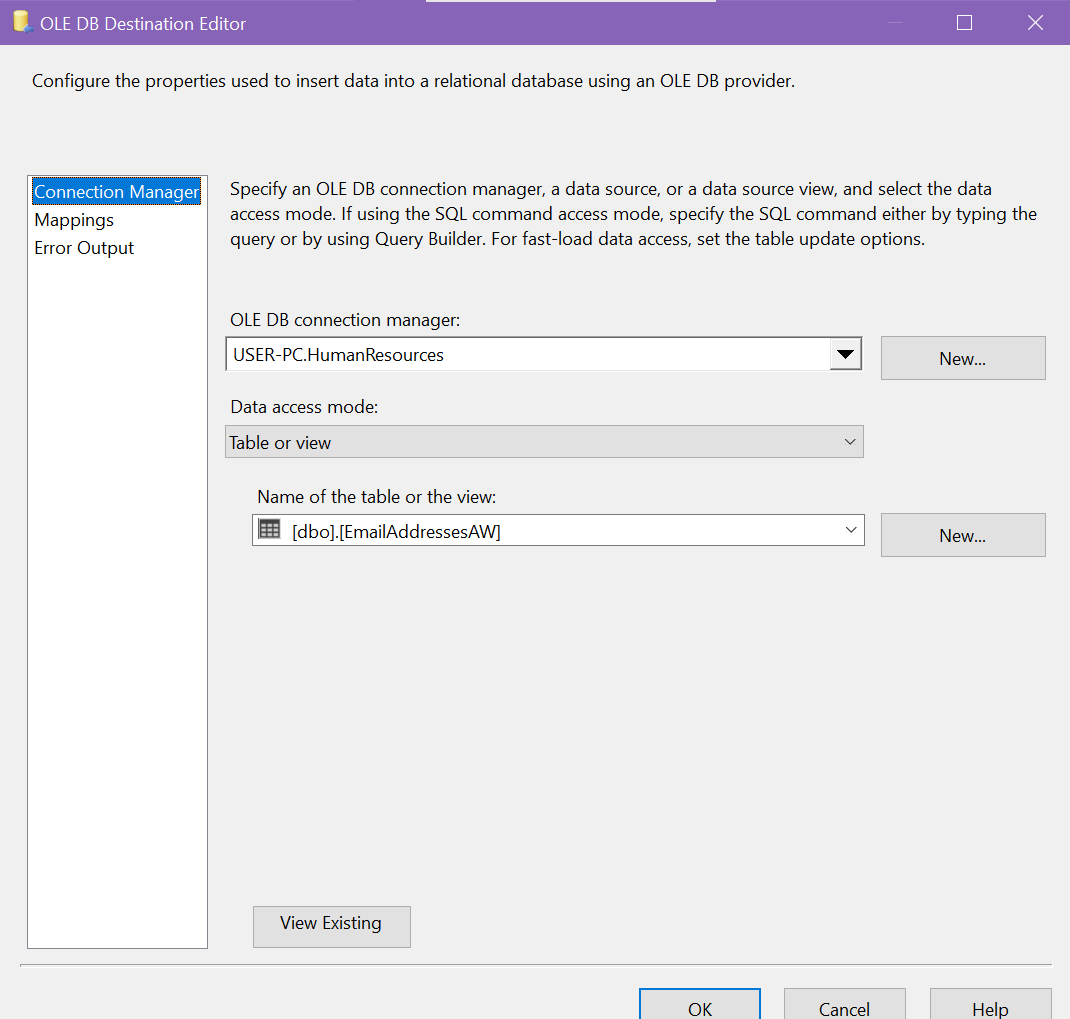
****

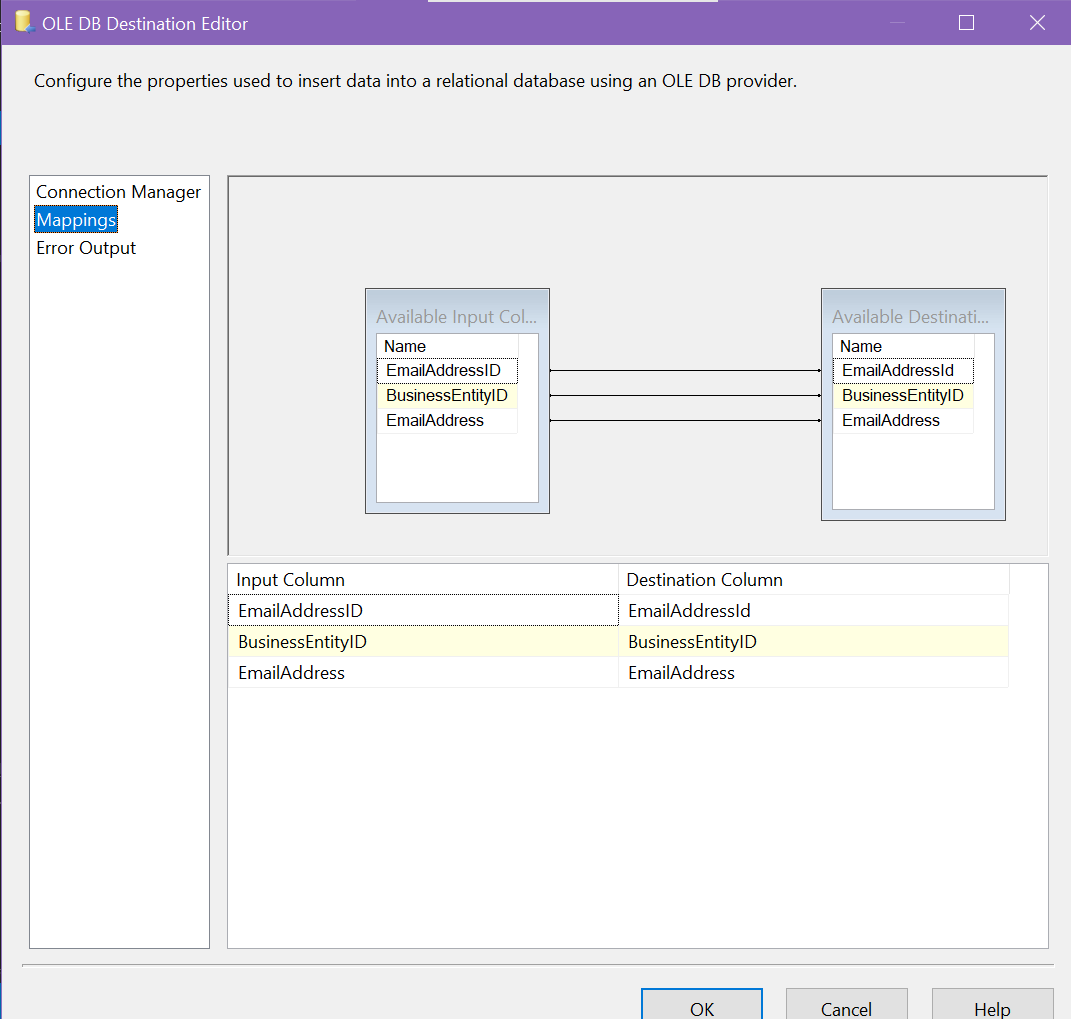
****

OLE DB source configuration:

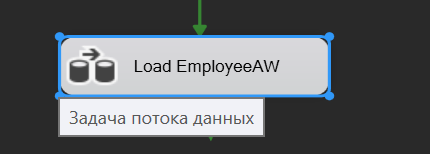
****

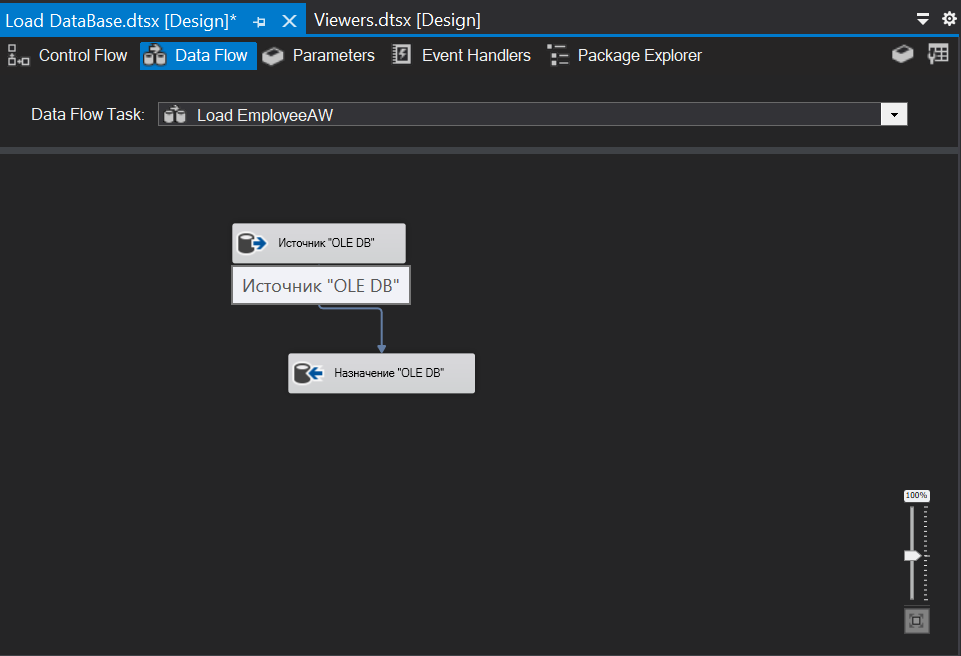
OLE DB destination configuration:

****

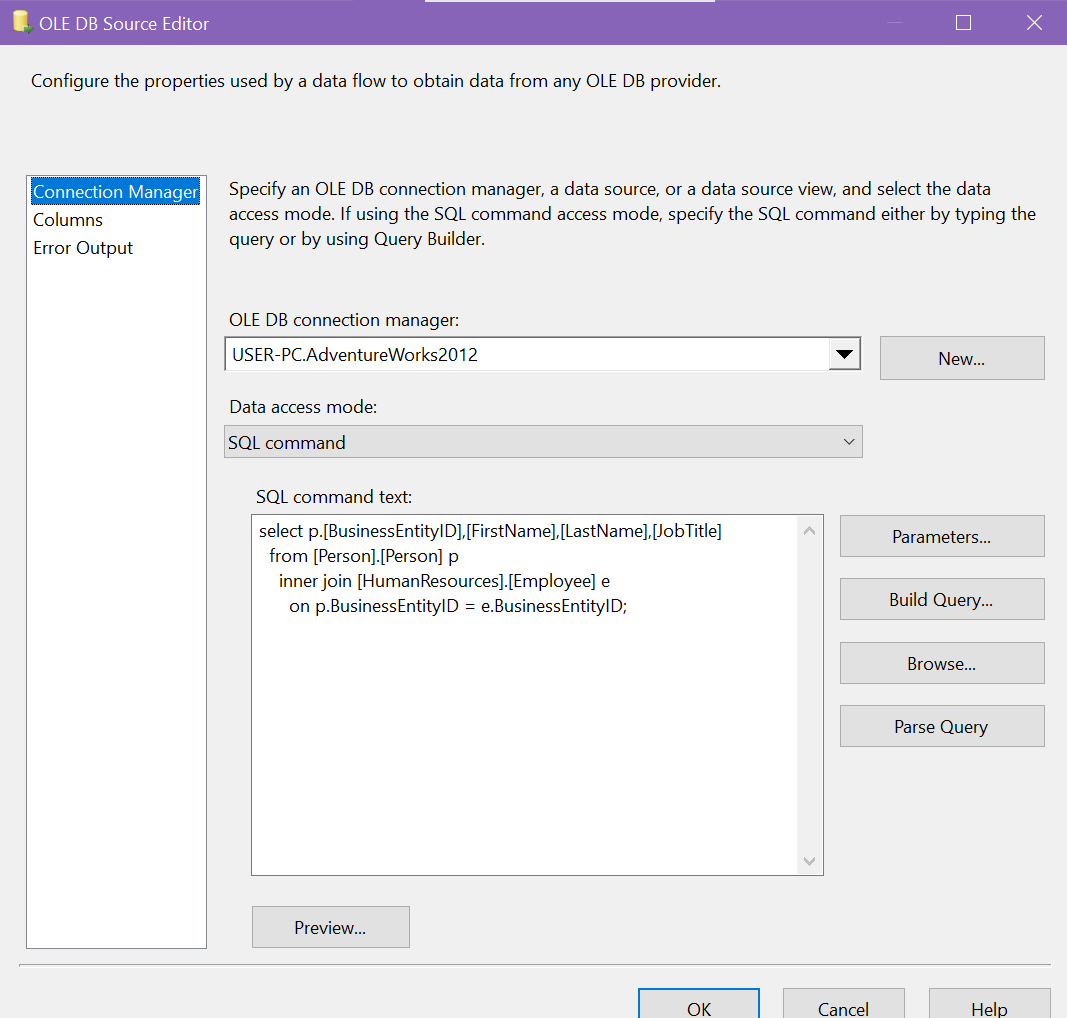
****

Load EmployeeAW data flow:

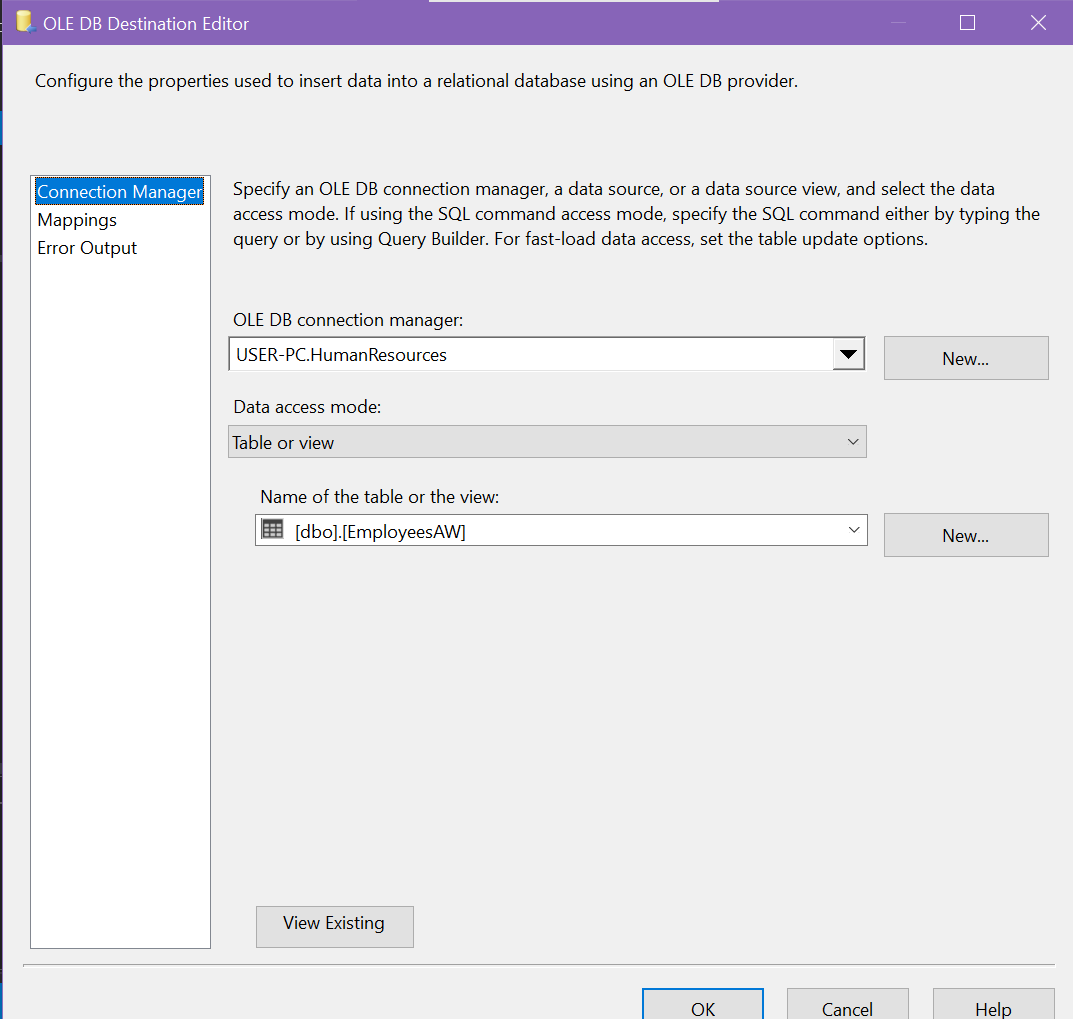
****

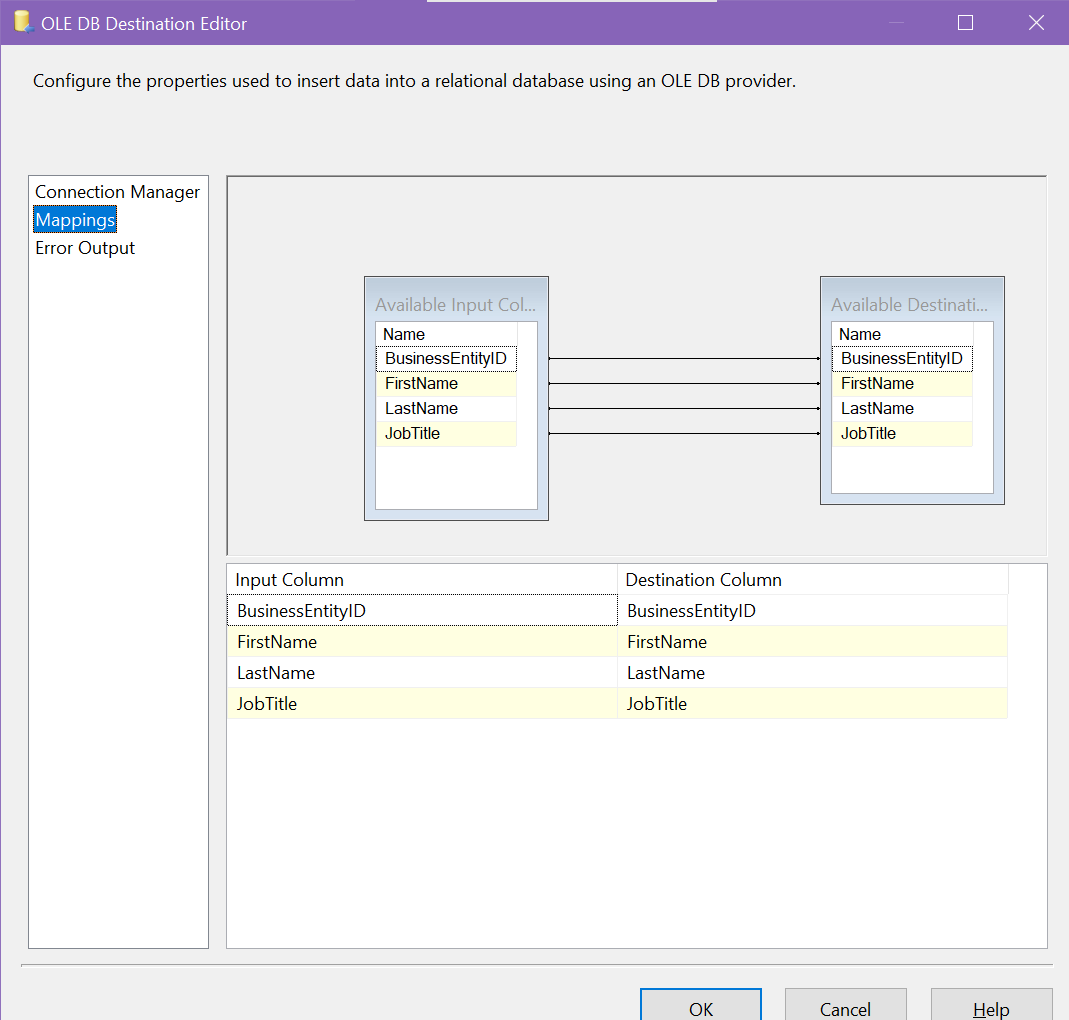
****

OLE DB source configuration:

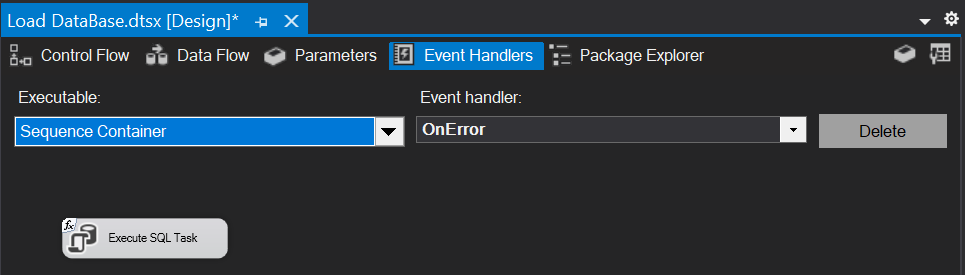
****

OLE DB destination configuration:

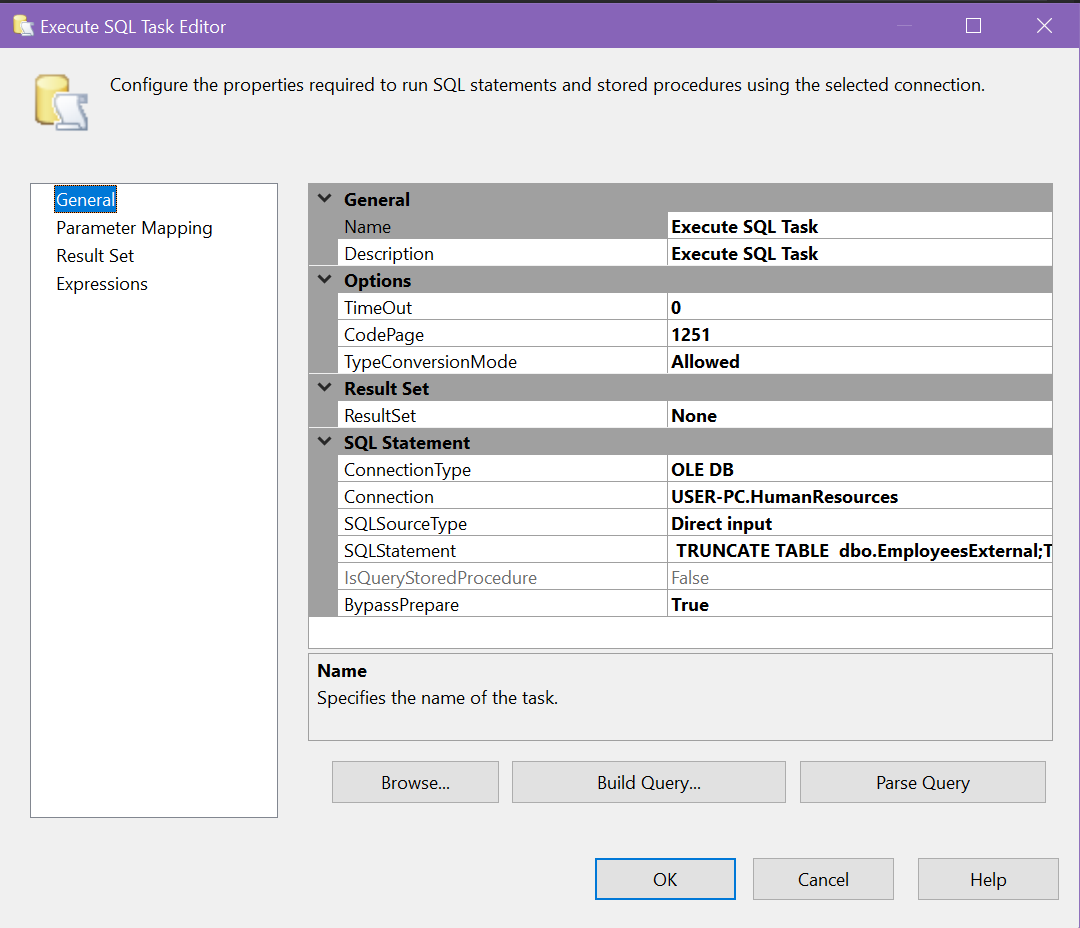
****

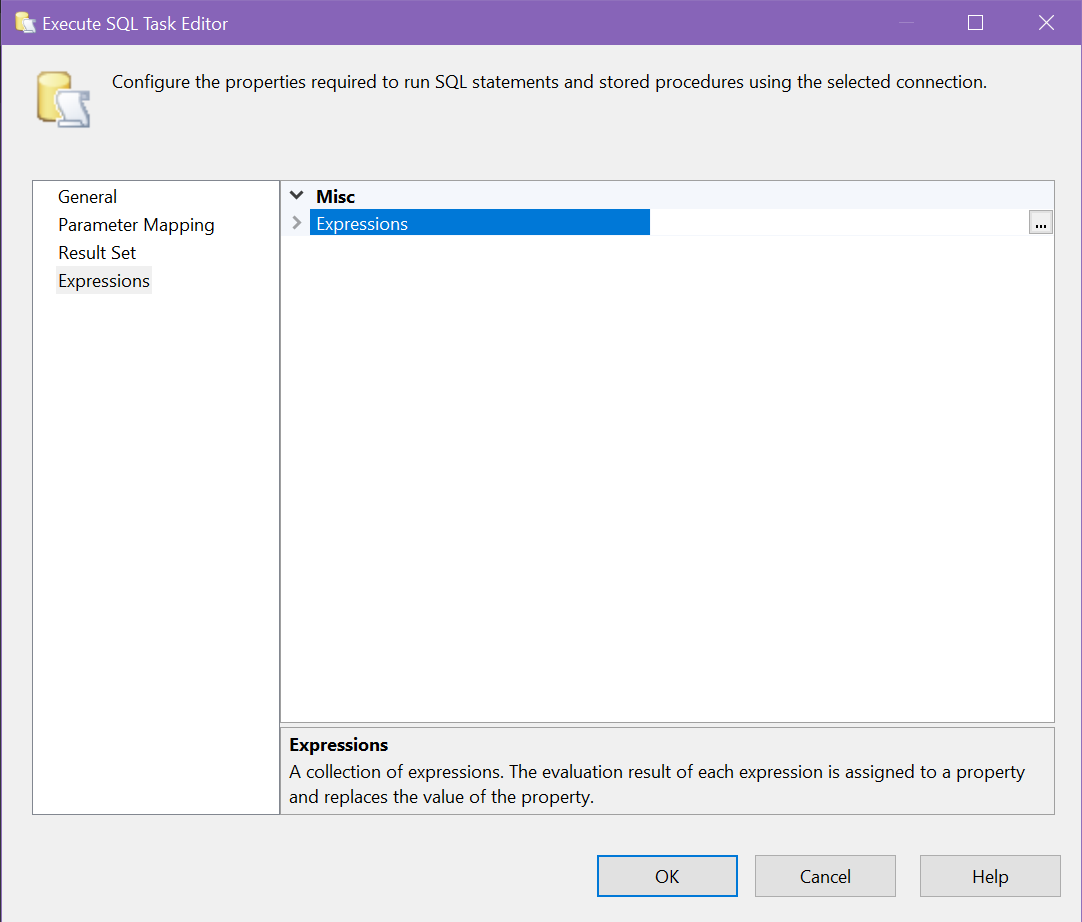
****

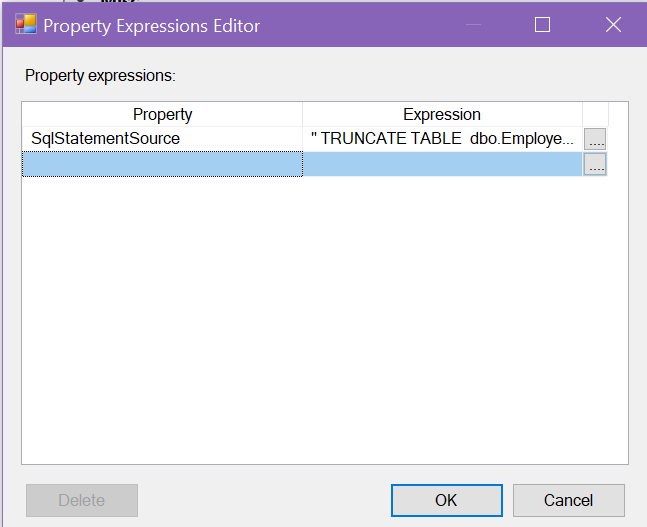
Clearing all tables that have managed to load in case of an error during code execution:

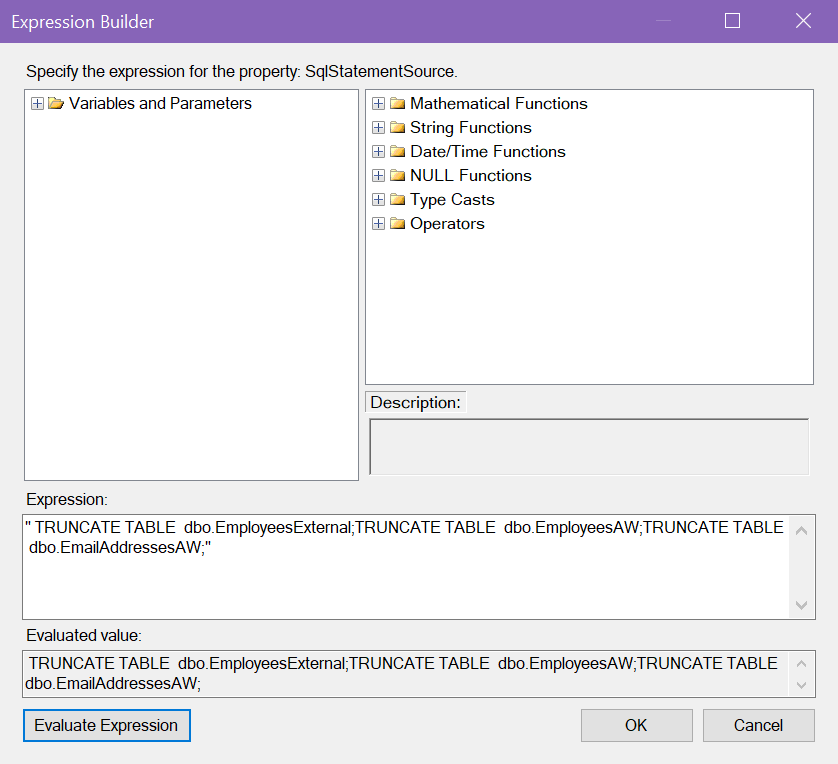
****

Execute SQL Task configuration:

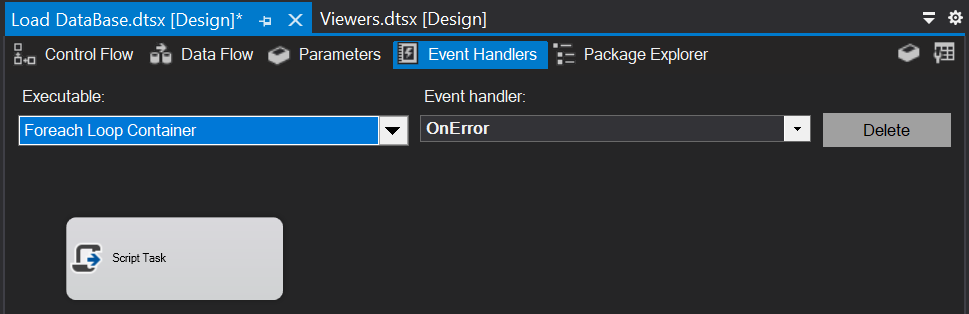
****

****

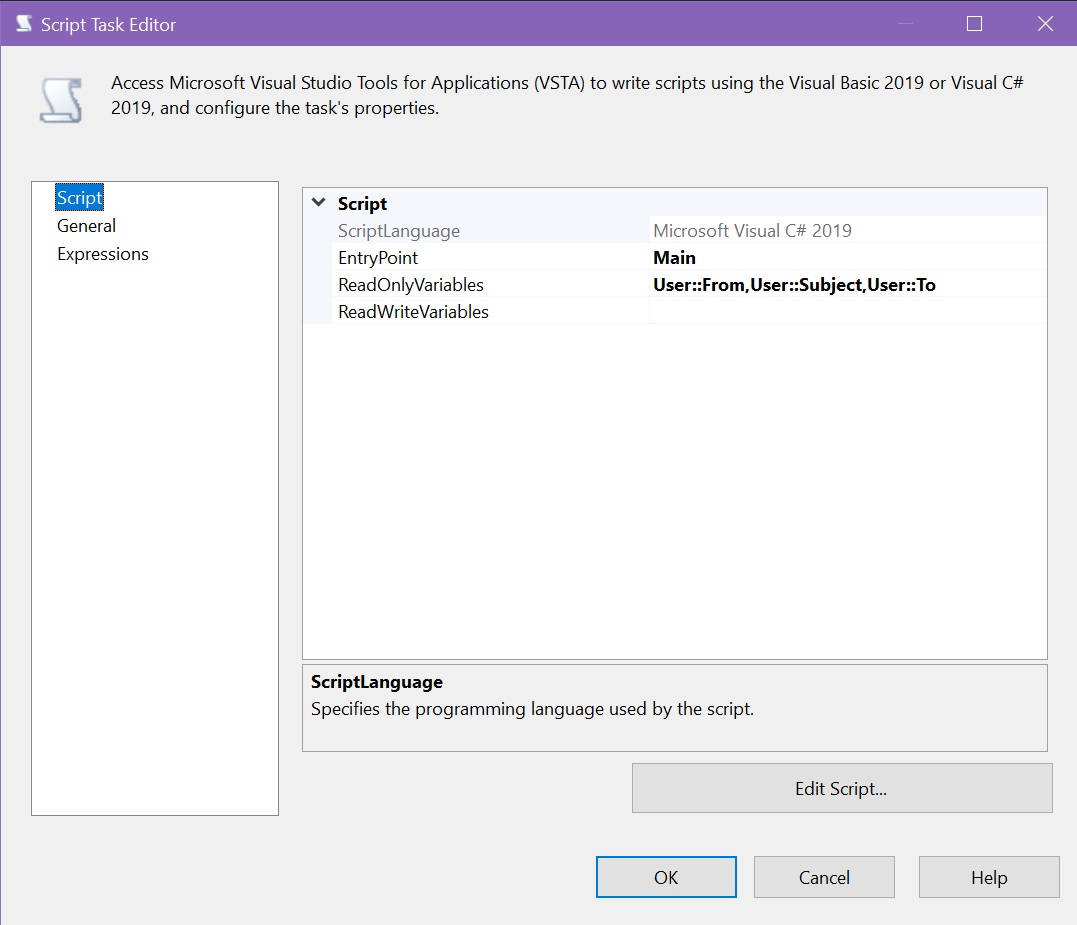
****

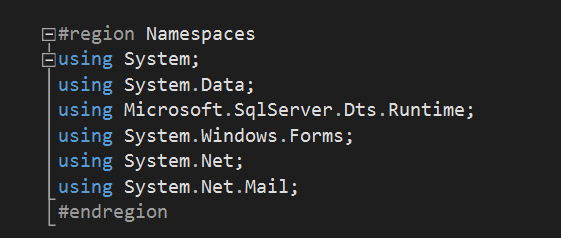
****

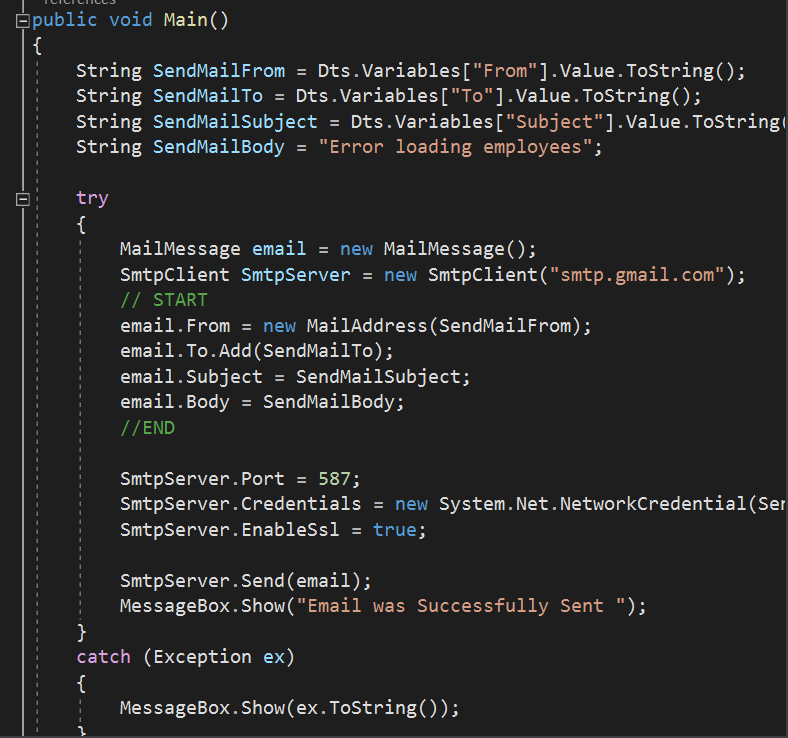
Sending an email in case of an error while loading the EmployeeExternal table:

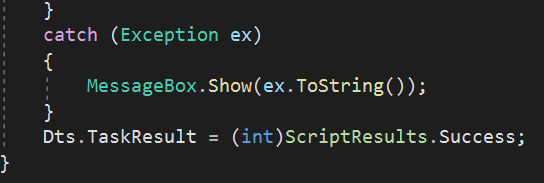
****

Script Task configuration:

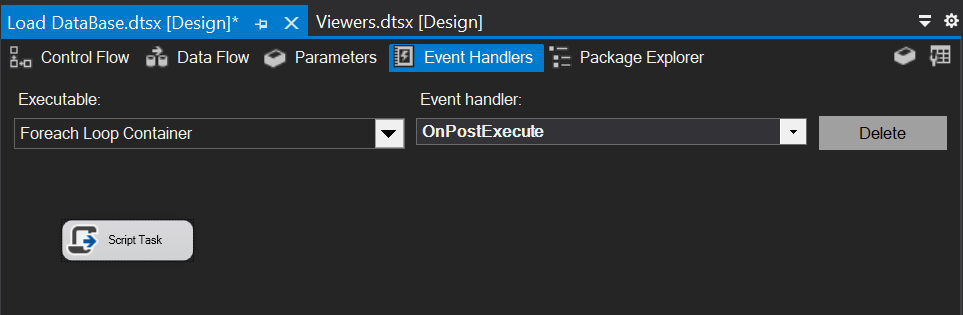
****

****

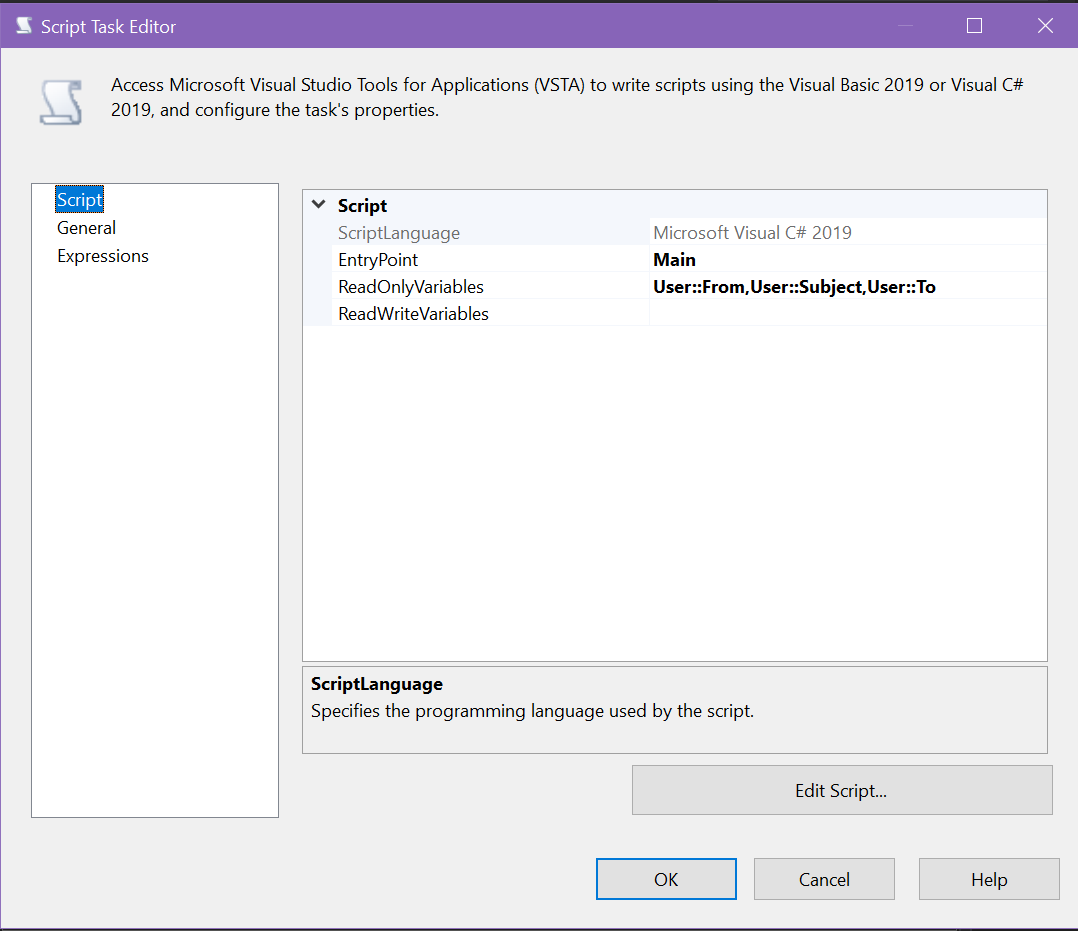
****

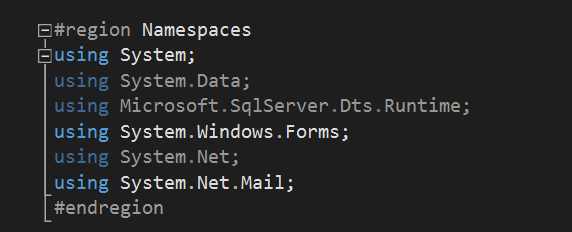
****

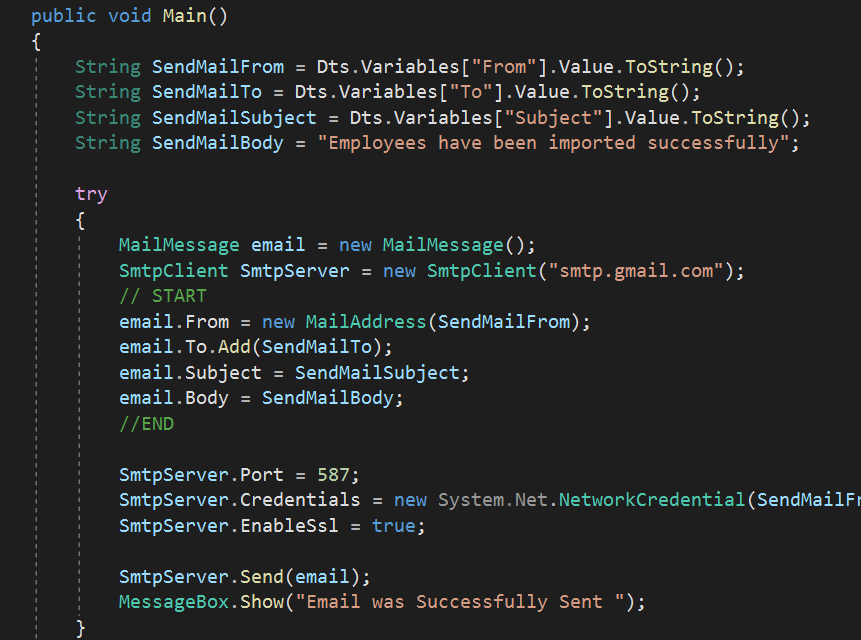
Sending an email in case of successful upload of the EmployeeExternal table:

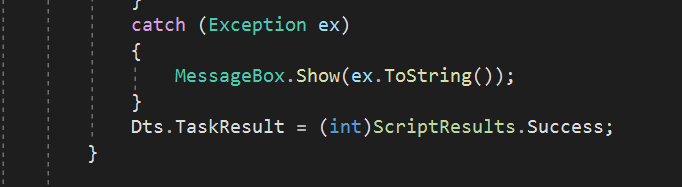
****

Script Task configuration:

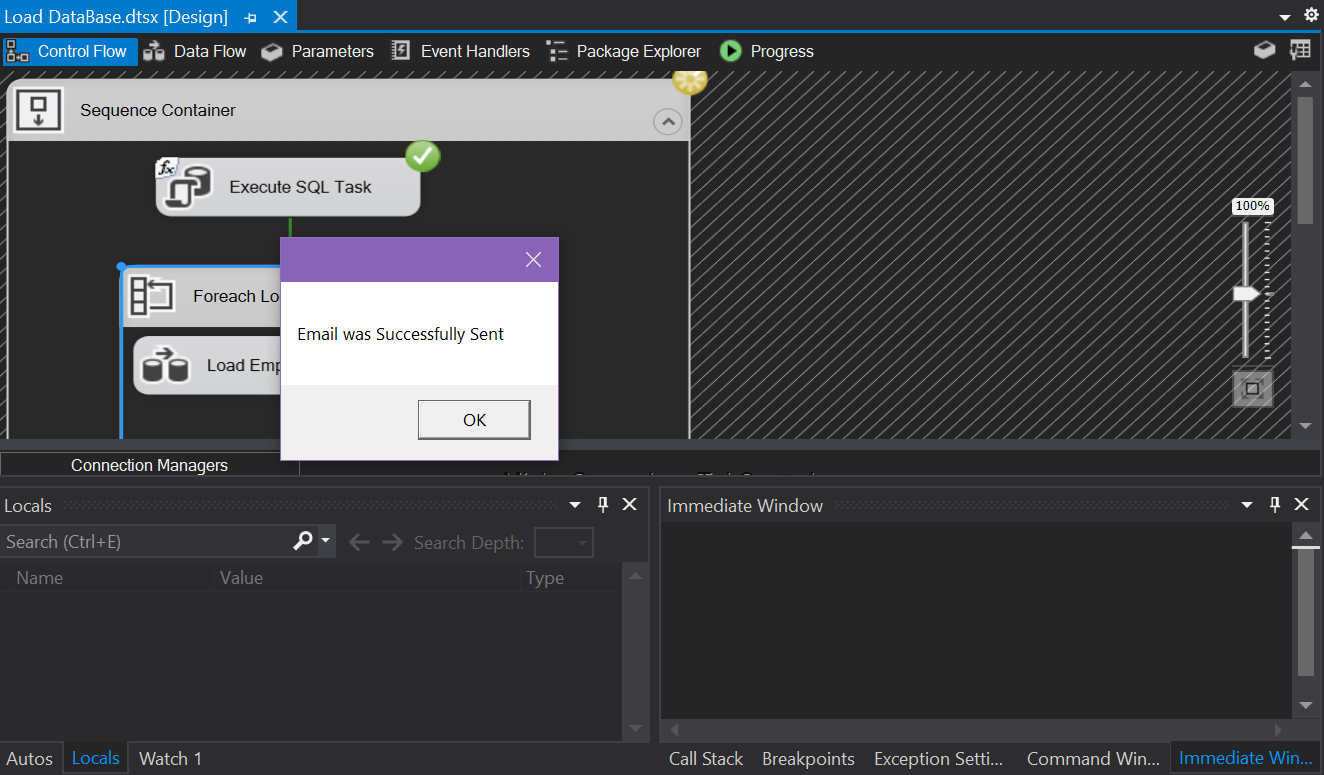
****

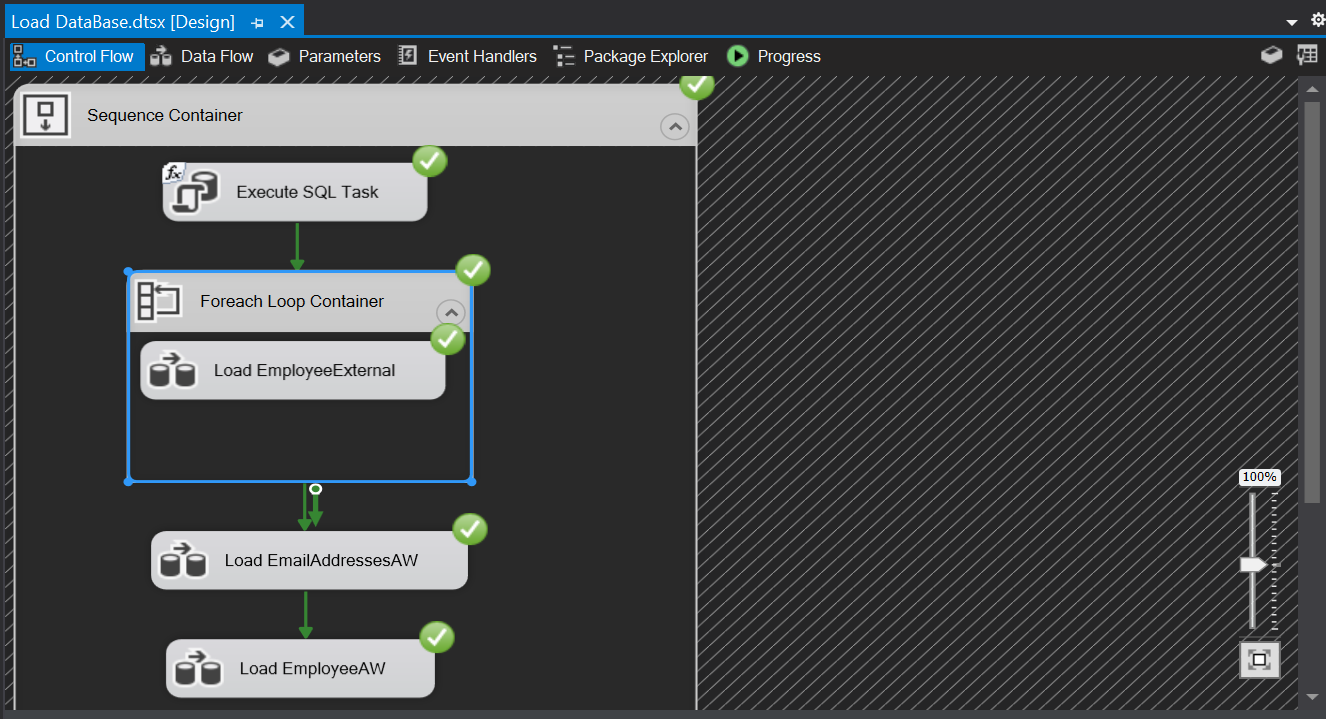
****

****

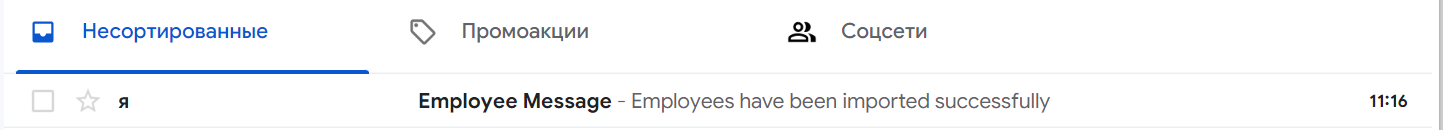
****

Run package:

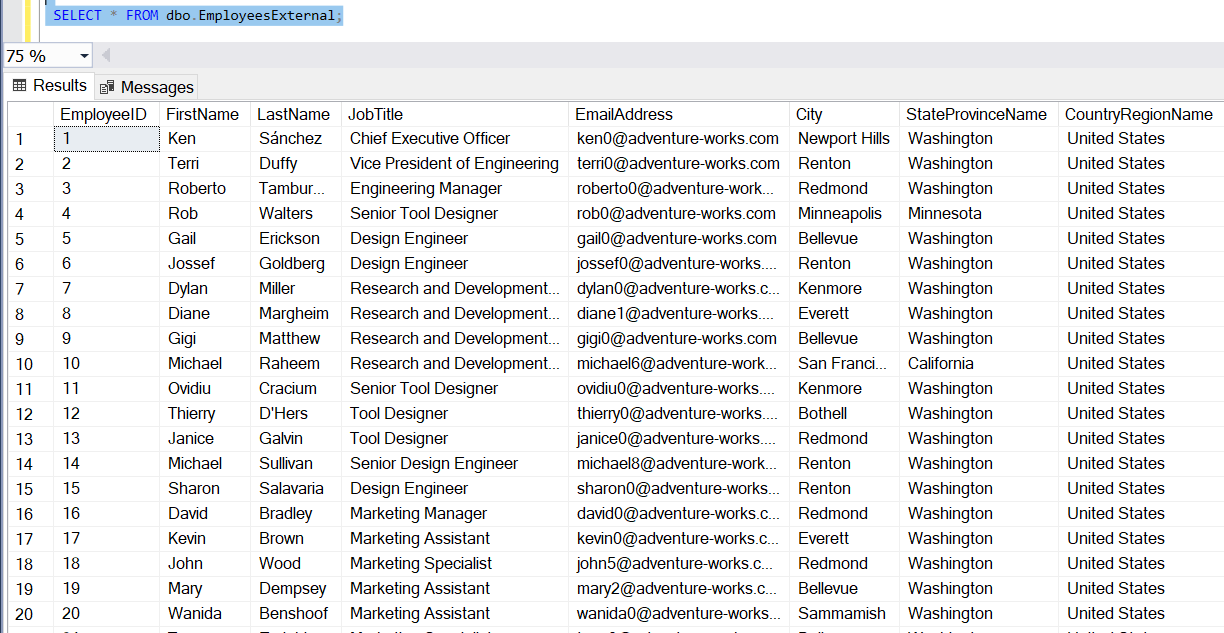
****

****

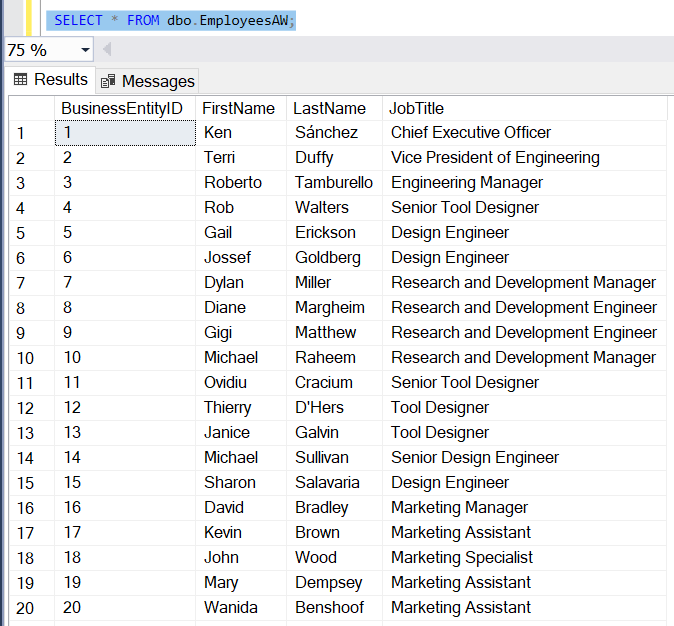
Result of email send:

****

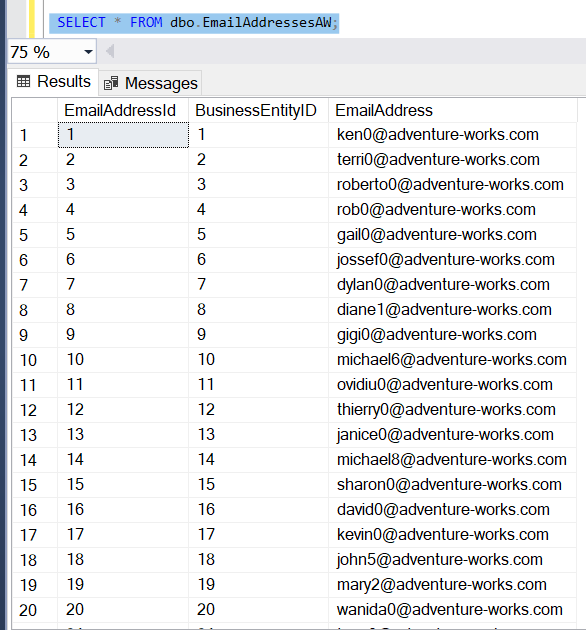
Select from dbo.EmployeeExternal:

****

Select from dbo.EmployeesAW:

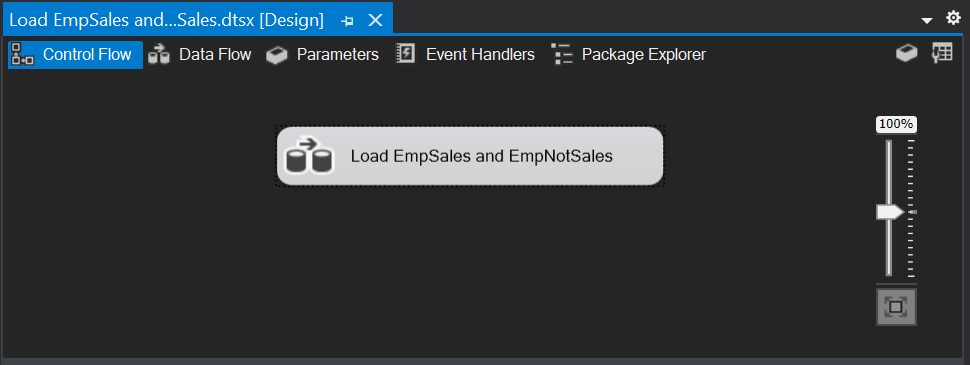
****

Select from dbp.EmailAddressesAW:

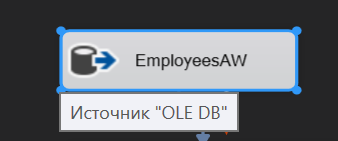
****

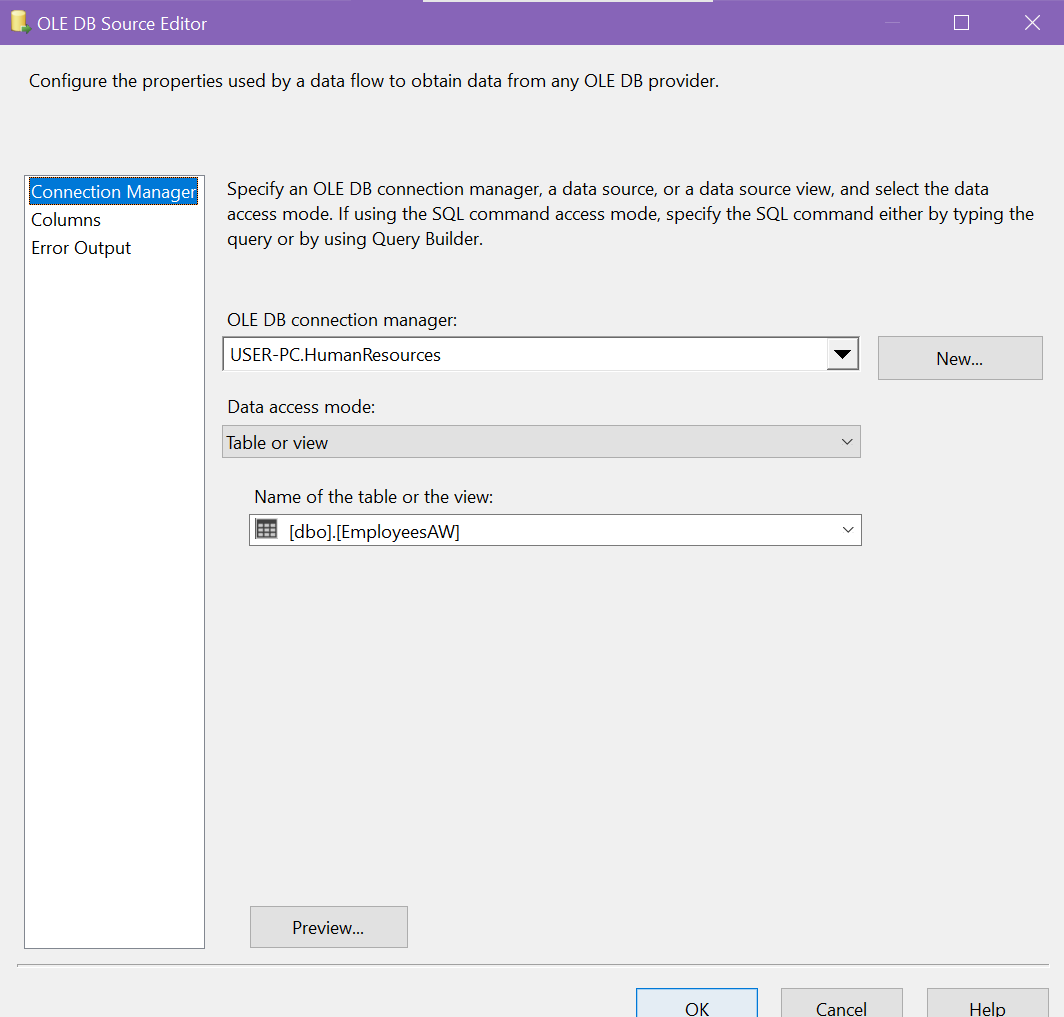
**Task 3 - Creating ETL solution to export the data from HumanResourcesdb**

Load EmpSales and EmpNotSales data flow:

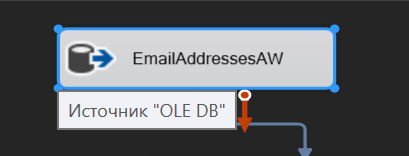
****

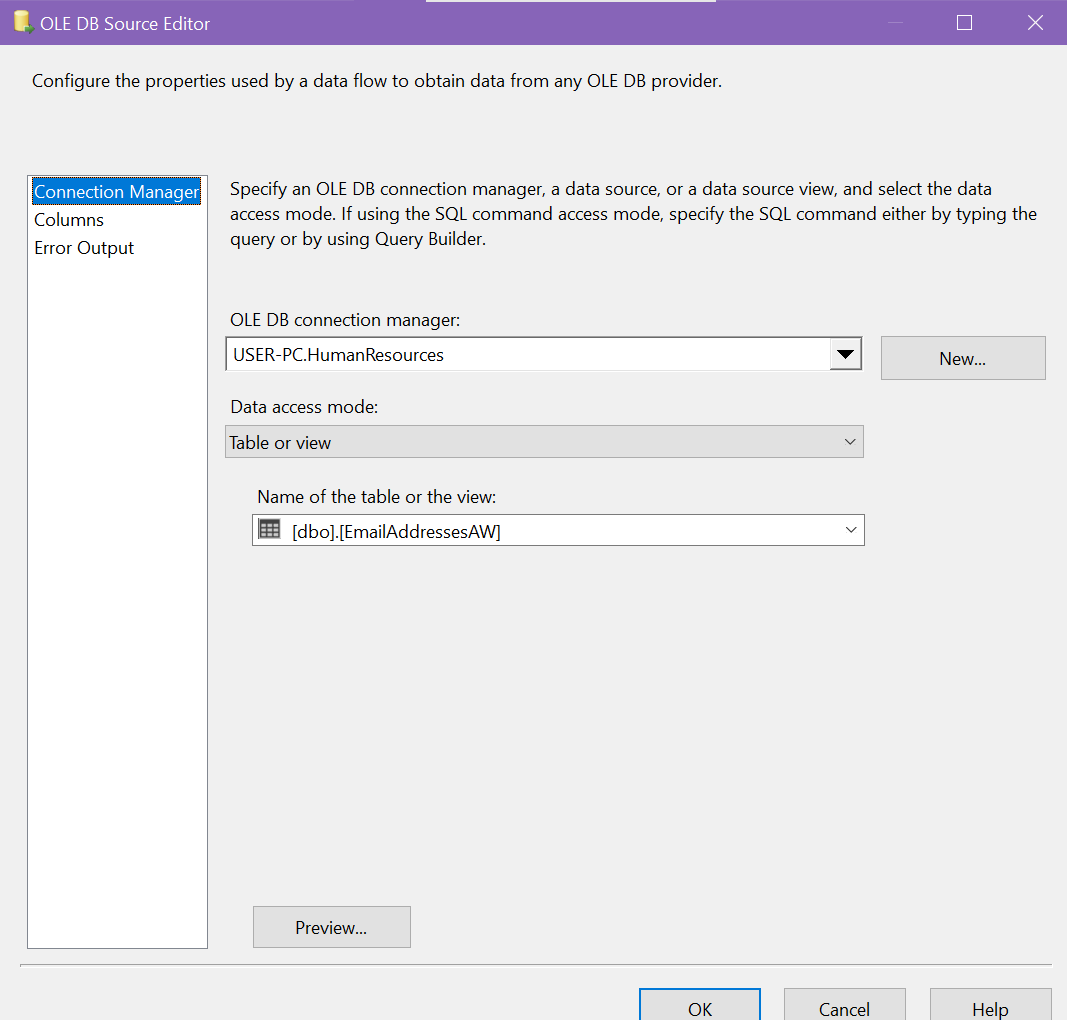
EmployeesAW configuration:

****

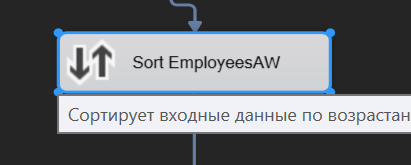
****

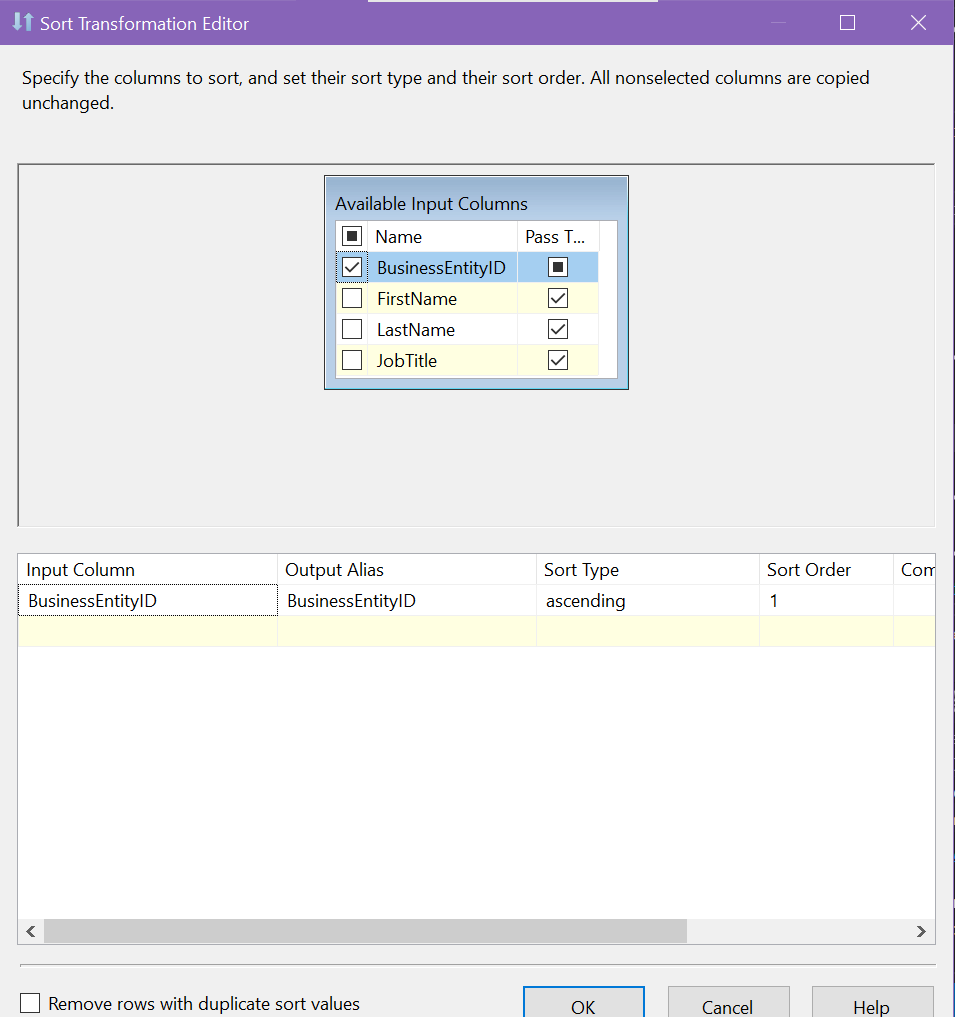
EmailAdressesAW configuration:

****

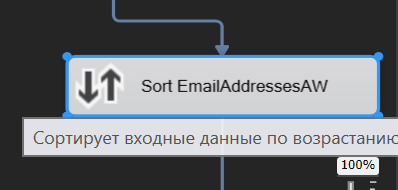
****

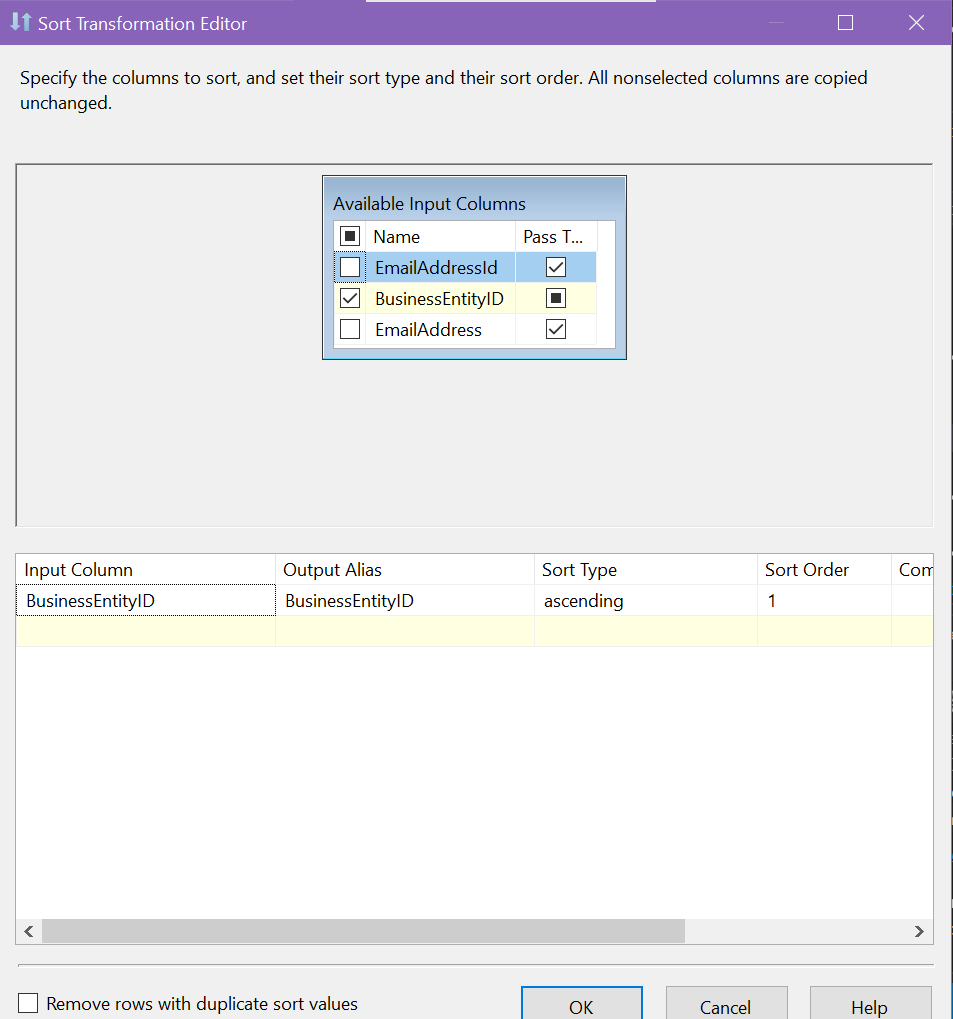
Sort EmployeesAW configuration:

****

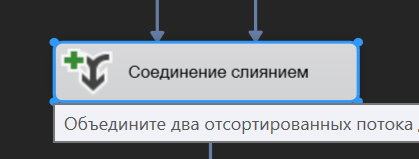
****

Sort EmailAddressesAW configuration:

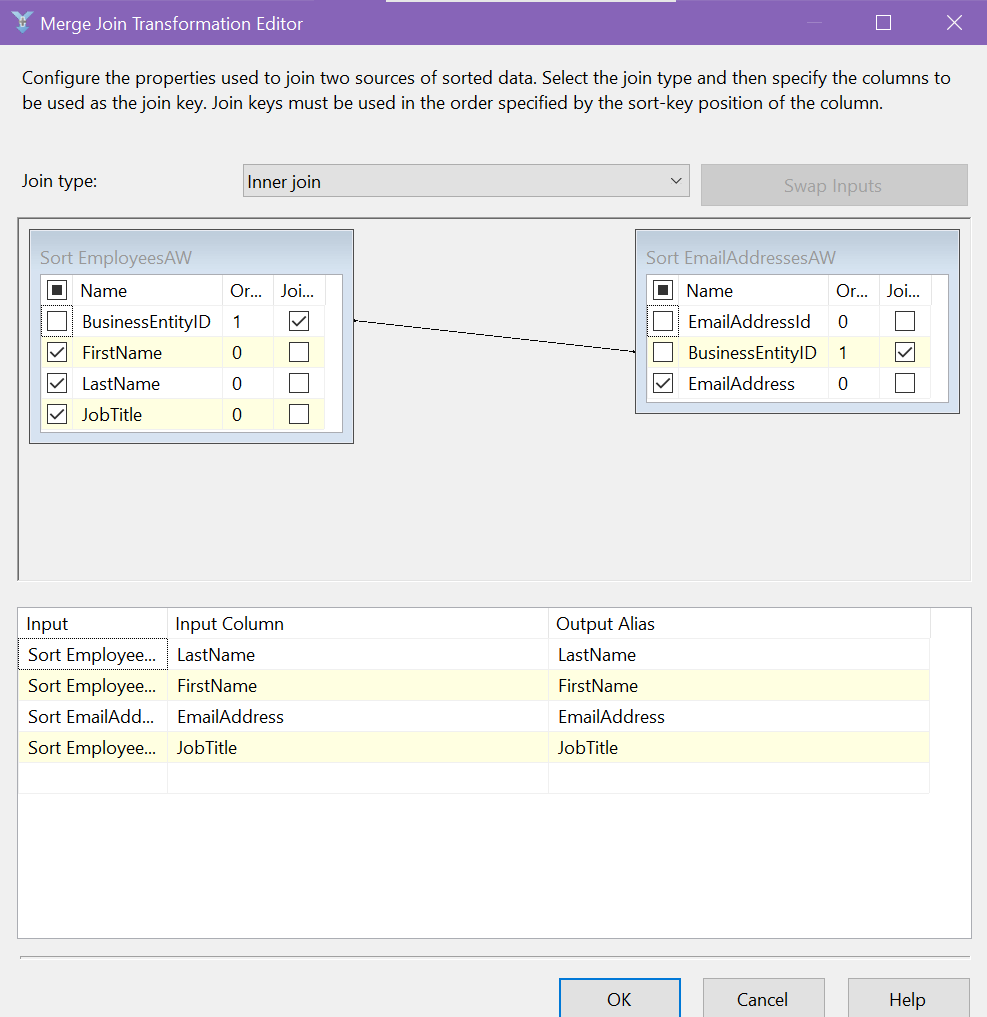
****

****

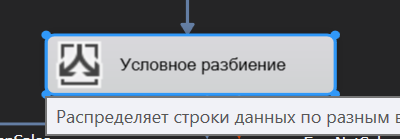
Merge Join configuration:

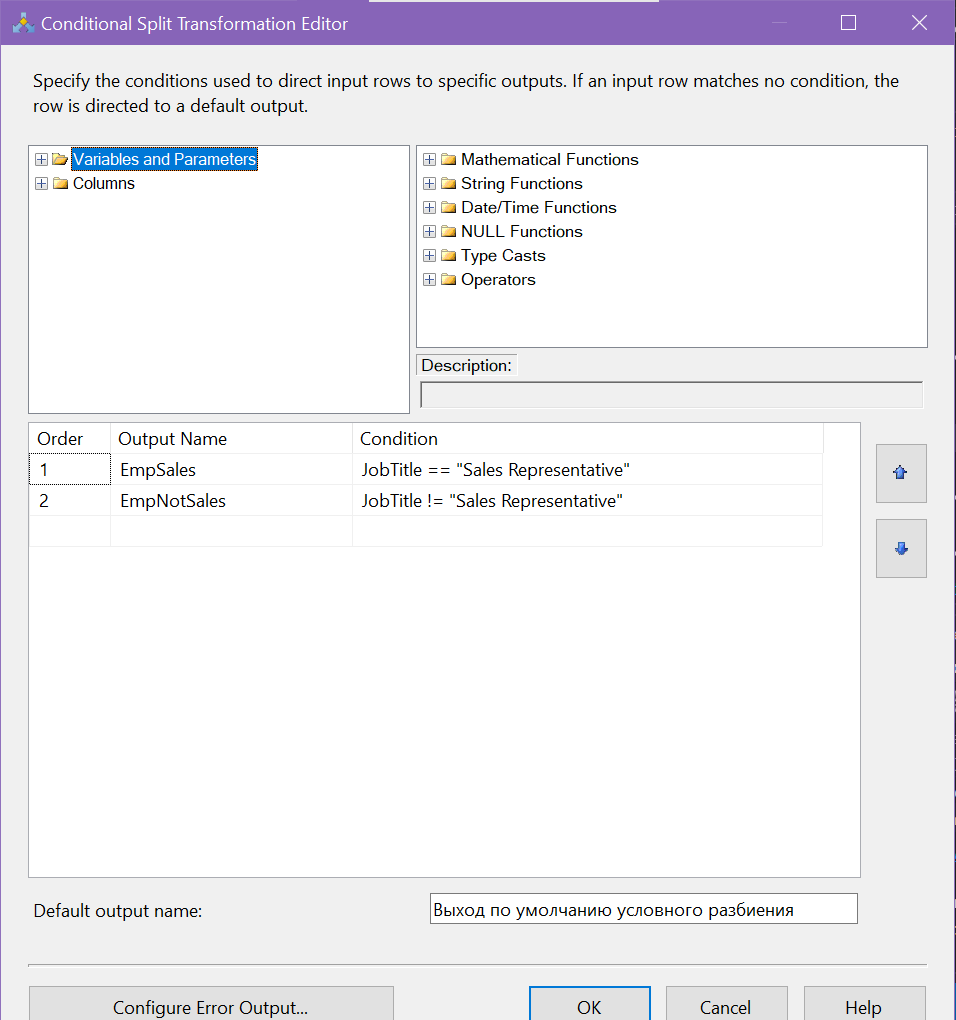
****

Add necessary columns:

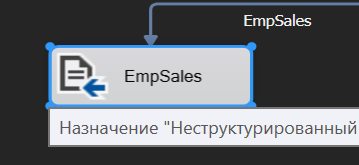
****

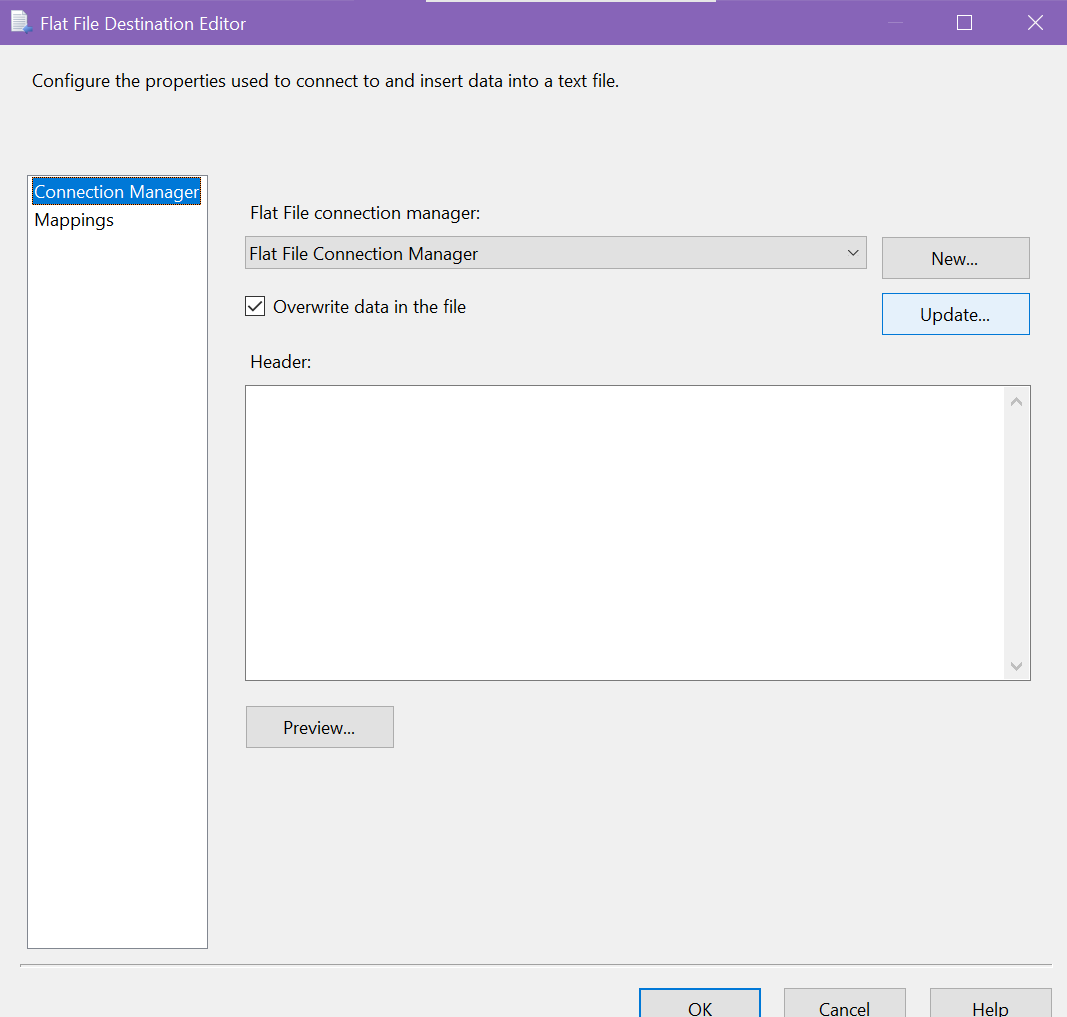
Conditional Split configuration:

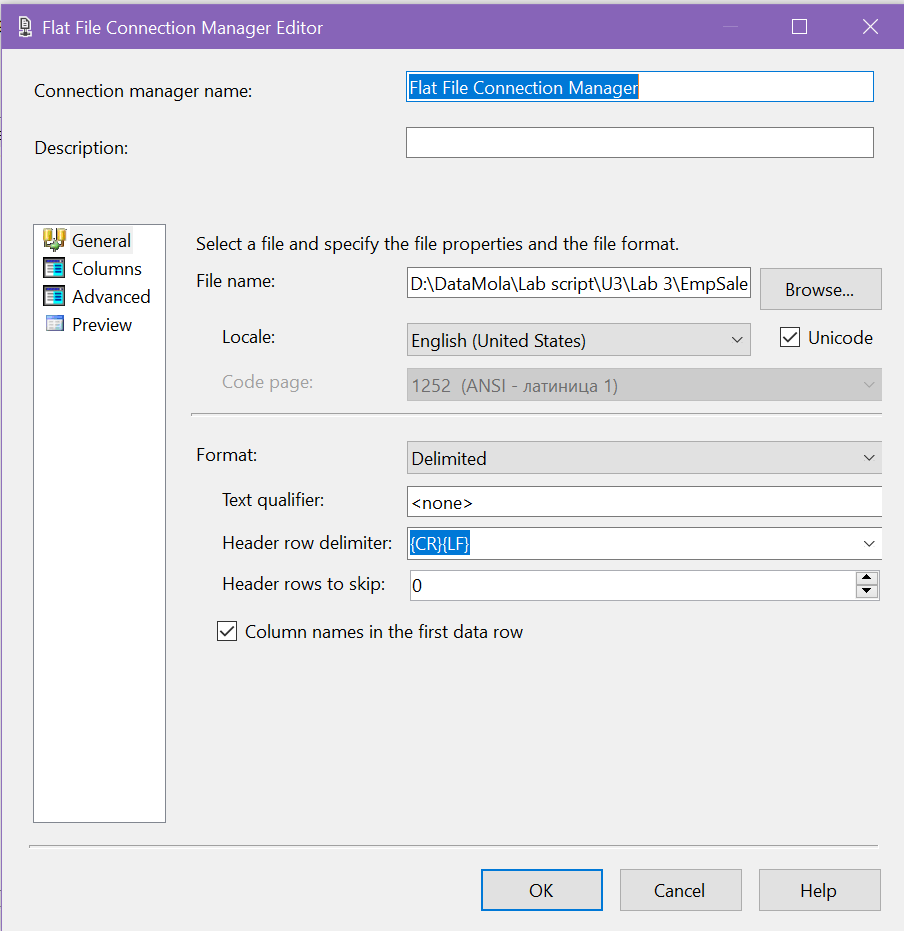
****

****

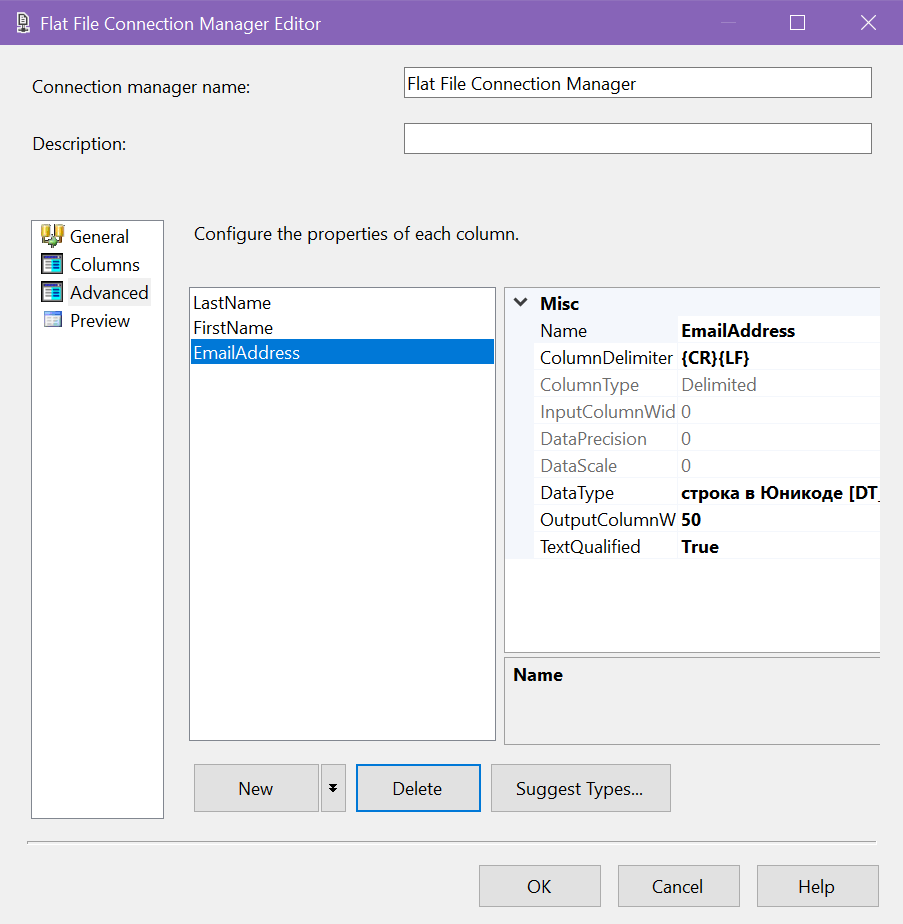
EmpSales configuration:

****

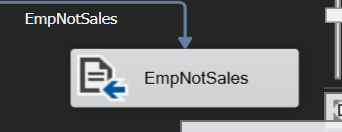
****

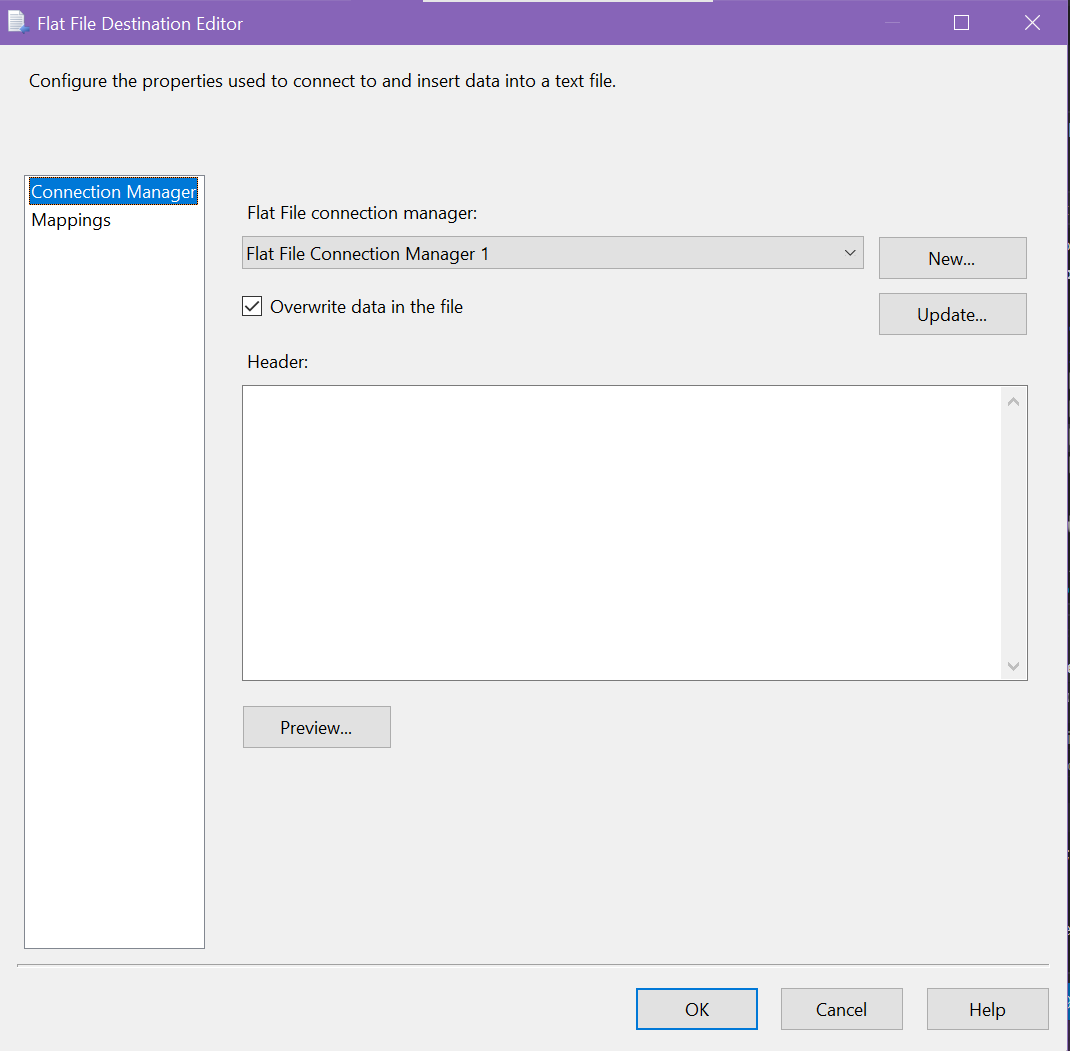
****

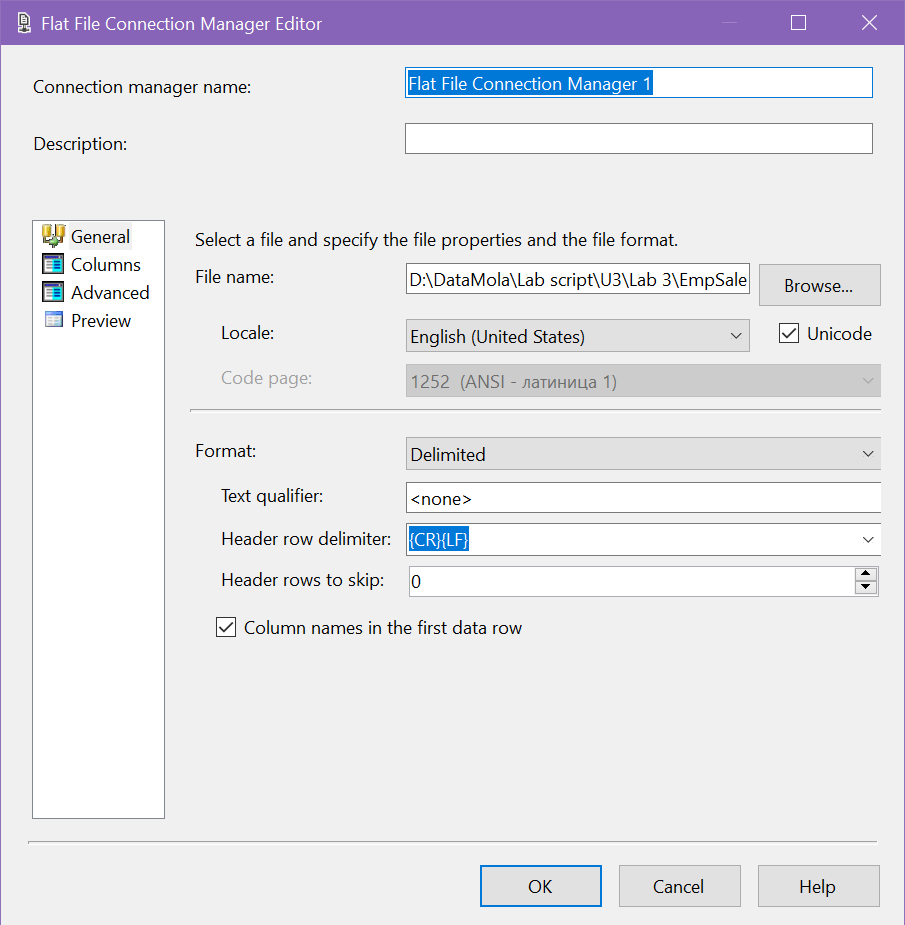
Delete JobTitle column from result set:

****

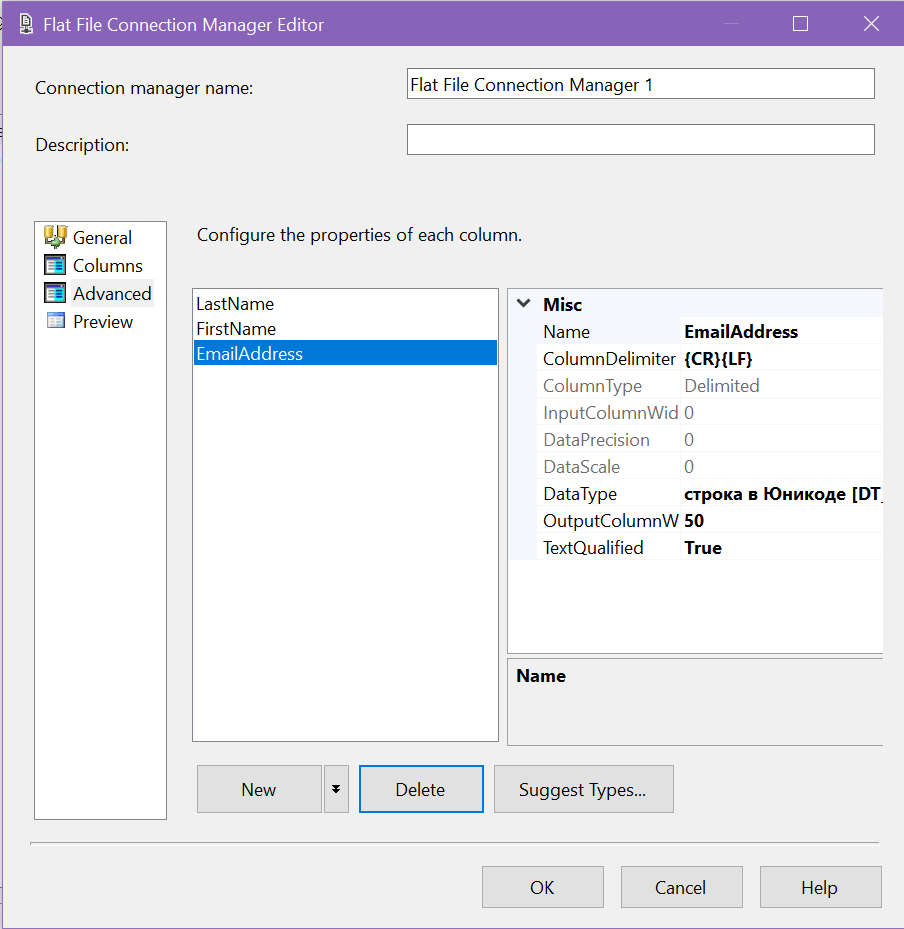
EmpNotSales confiduration:

****

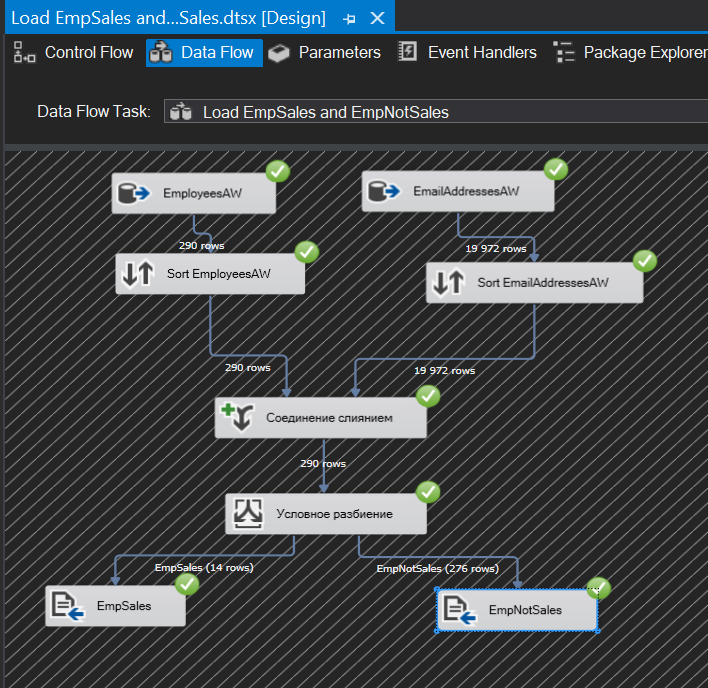
****

****

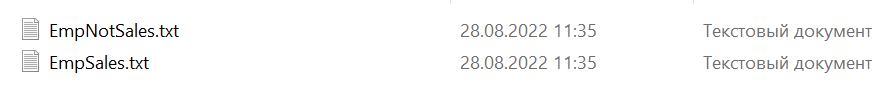
Delete JobTitle column from result set:

****

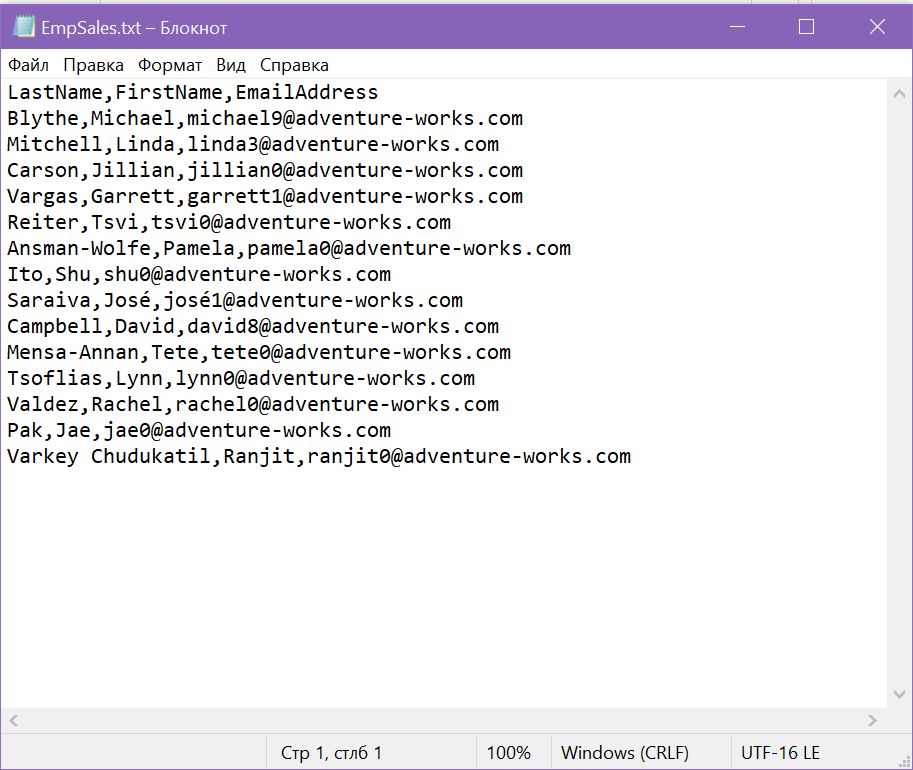
Run package:

****

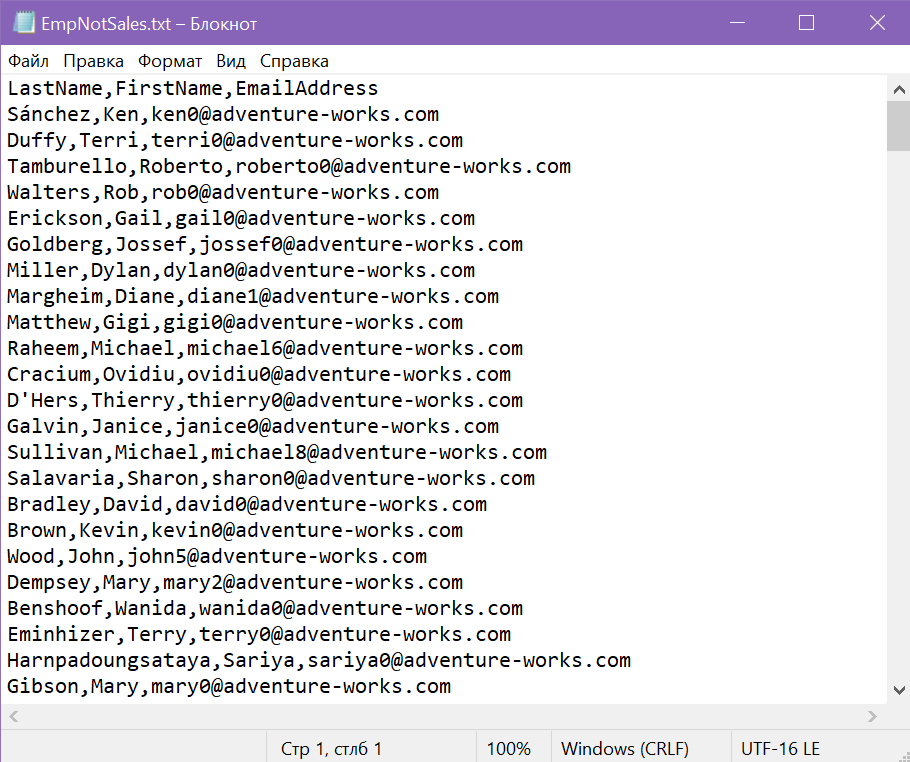
Set of files:

****

Result of loading EmpSales file:

****

Result of loading EmpNotSales file:

****

**Task 4 – Using Data Viewers**

SSIS supports several types of data viewers. The one most commonly used is the grid data viewer, which displays the data in tabular format. However, you can also create data viewers that display histograms, scatter plot charts, or column charts.

For an example of how the viewer works, I'm using a package that loads data from HumanResources database tables into the EmpSales and EmpNotSales files.

I will create a view between Merge Join and Conditional Split to see what data is obtained after Merge Join:

