Report

Laboratory Work 3

Dmitry Ladutsko

August 27, 2022

1. Exercise 1: Creating new database for HR department

You have to create a new database for HR department. The name of the database must be «HumanResources».The database must contain the following tables:

dbo.EmployeesExternal:

Column name	Type	Null?
EmployeeID (key)	INT	NOT NULL
FirstName	NVARCHAR(50)	NOT NULL
LastName	NVARCHAR(50)	NOT NULL
JobTitle	NVARCHAR(50)	NOT NULL
EmailAddress	NVARCHAR(50)	NULL
City	NVARCHAR(50)	NOT NULL
StateProvinceName	NVARCHAR(50)	NOT NULL
CountryRegionName	NVARCHAR(50)	NOT NULL

dbo.EmployeesAW:

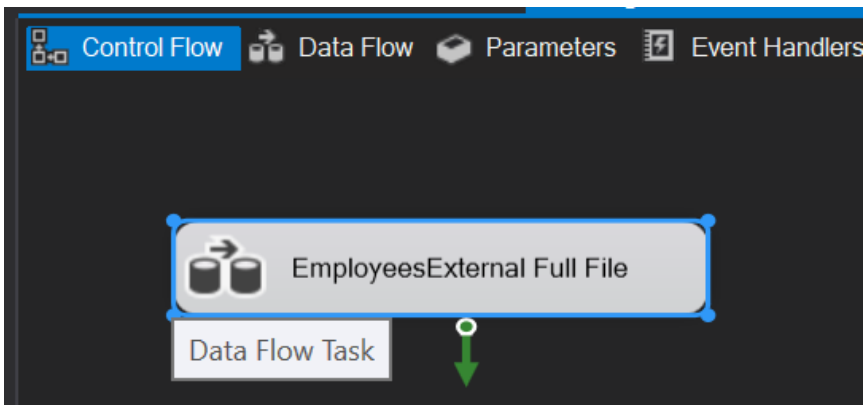
Column name	Type	Null?
BusinessEntityID (key)	INT	NOT NULL
FirstName	NVARCHAR(50)	NOT NULL
LastName	NVARCHAR(50)	NOT NULL
JobTitle	NVARCHAR(50)	NULL

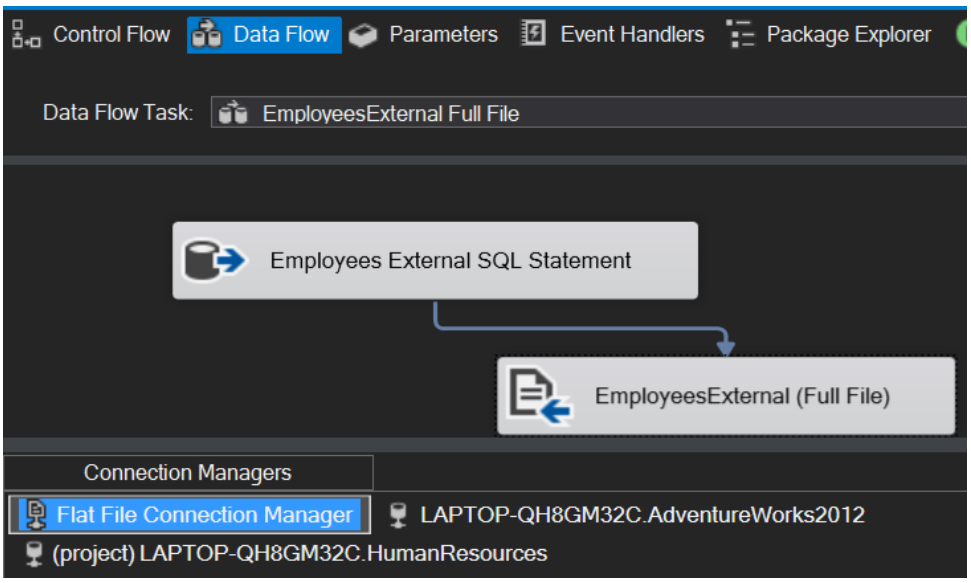
dbo.EmailAddressesAW:

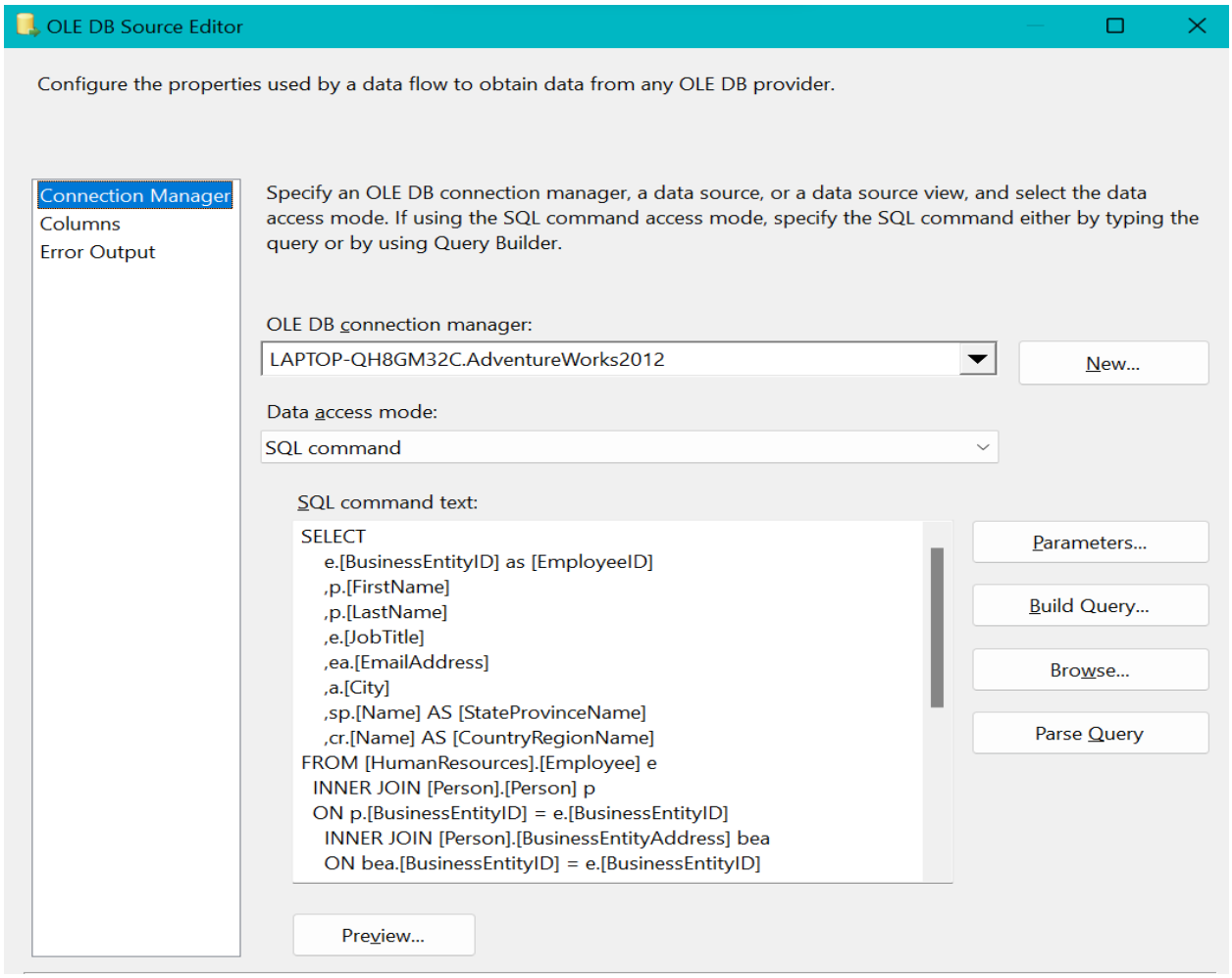
Column name	Type	Null?
EmailAddressId (key)	INT	NOT NULL
BusinessEntityID	INT	NOT NULL
EmailAddress	NVARCHAR(50)	NOT NULL

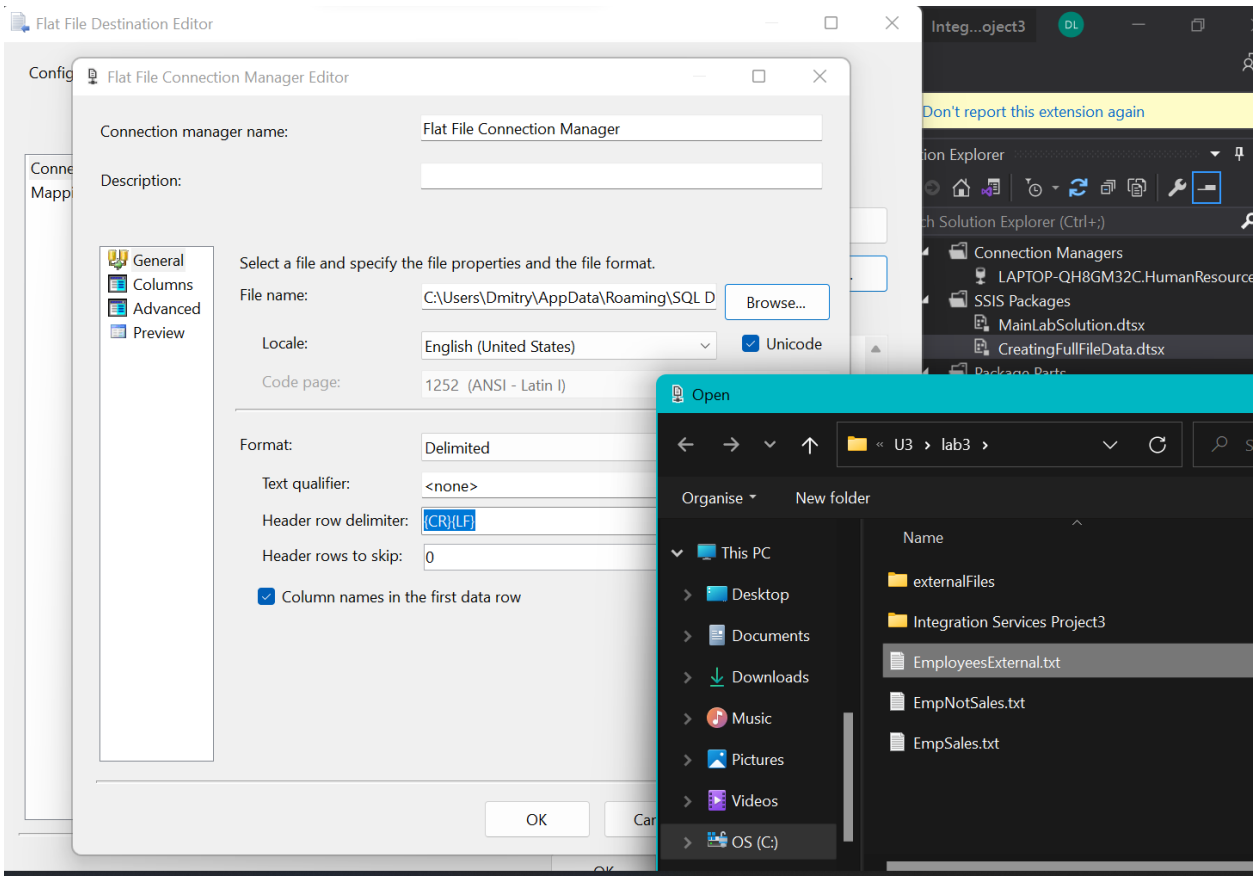
Create sql script to achieve the requirements above.

Note. We need to make file with Employees called EmployeesExternal to fulfil it with comma – delimited data to later import it to our new created database HumanResources

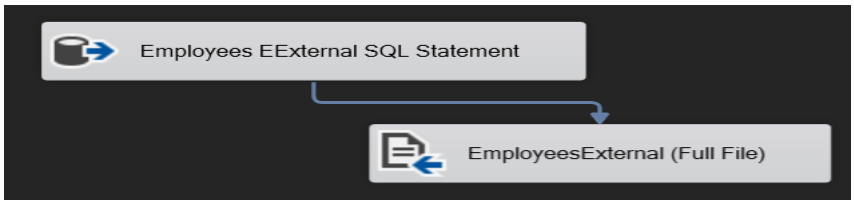


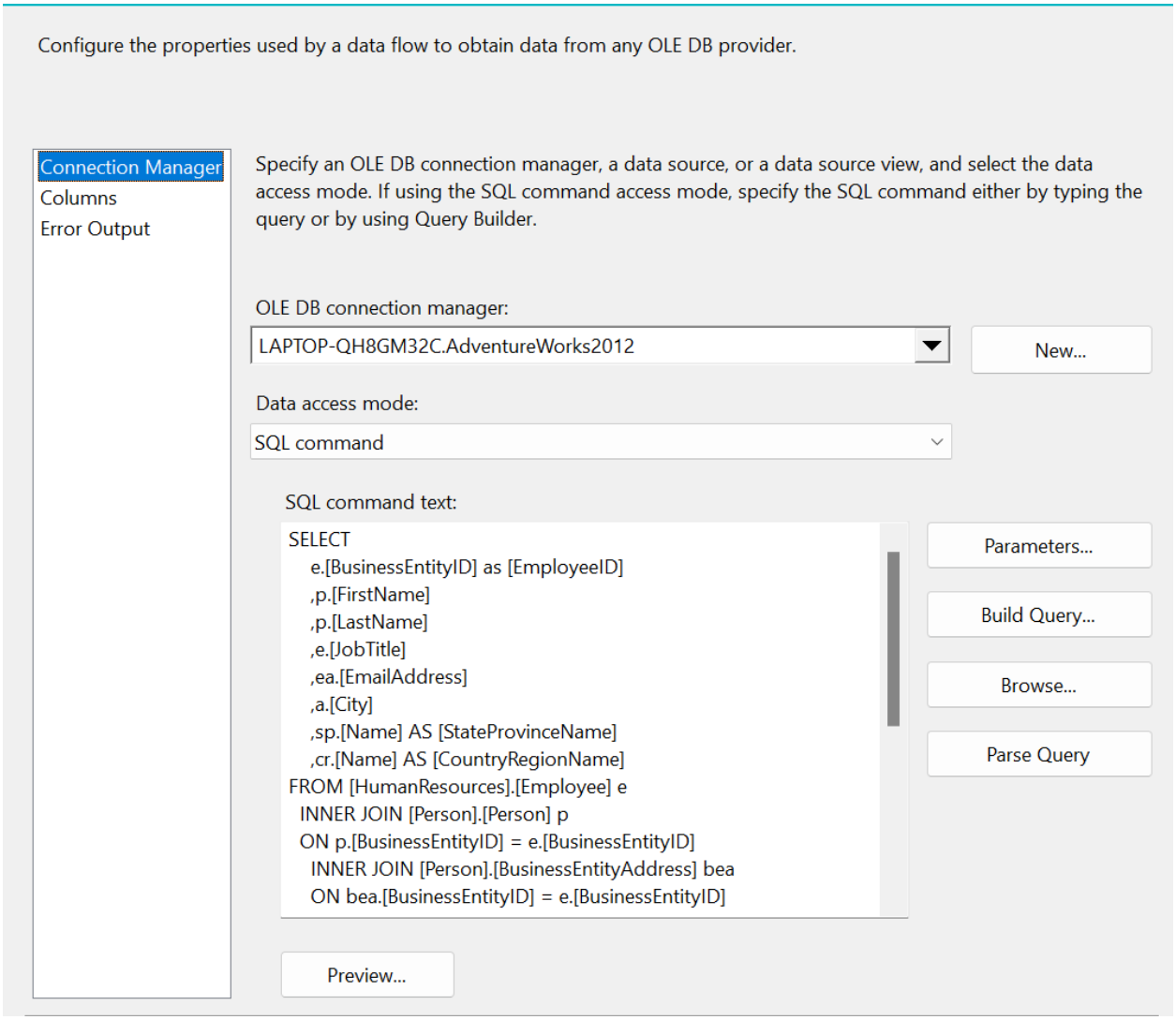

Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Data Flow Task:
EmployeesExternal Full File
Employees External SQL Statement
EmployeesExternal (Full File)
Connection Managers
Flat File Connection Manager
LAPTOP-QH8GM32C.AdventureWorks2012
(project) LAPTOP-QH8GM32C.HumanResources


OLE DB Source Editor
Configure the properties used by a data flow to obtain data from any OLE DB provider.
Connection Manager
Columns
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.
OLE DB connection manager:
LAPTOP-QH8GM32C.AdventureWorks2012
New...
Data access mode:
SQL command
SQL command text:
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]
Parameters...
Build Query...
Browse...
Parse Query
Preview...



Note. All import process will be be executed within one package but I separated a part of this lab in which I need to generate comma delimited file in txt format to later use it showing how I could specify package to import any amount of *.txt files.






```
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] asc;
```

Note. I created database Human Resources with tables listed above. Script stores in file "create database and tables.sql". Used Adventure Works database I joined tables to get following structures as shown above, but with data inside.

dbo.EmployeesExternal:

```
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] asc;
```

dbo.EmployeesAW:

```
select p.[BusinessEntityID],[FirstName],[LastName],[JobTitle]
       from [Person].[Person] p
              inner join [HumanResources].[Employee] e
                     on p.BusinessEntityID = e.BusinessEntityID;
```

dbo.EmailAddressesAW:

```
select [EmailAddressID],p.[BusinessEntityID],[EmailAddress]
       from [Person].[EmailAddress] e
              inner join [Person].[Person] p
                     on e.[BusinessEntityID] = p.BusinessEntityID
```

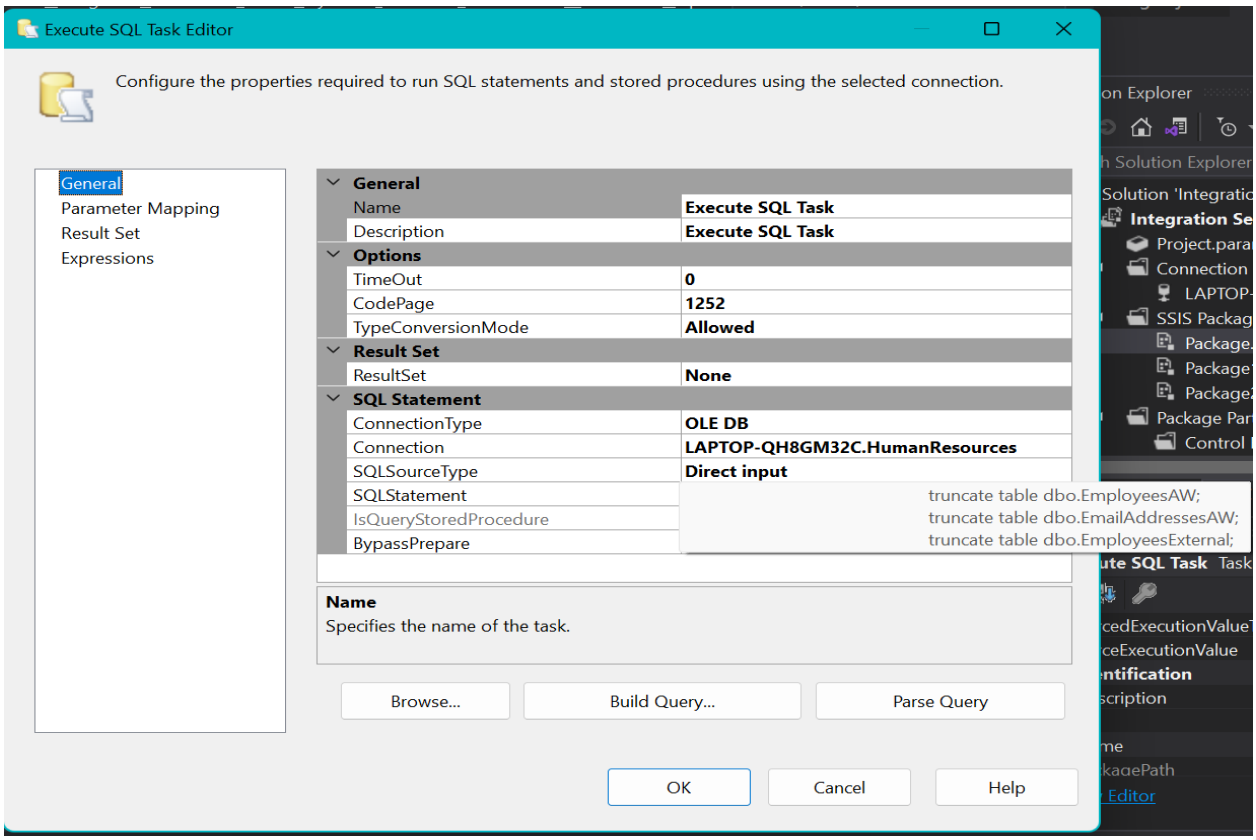
Result: SQL scripts that created needed objects.

2. Exercise 2: Creating ETL solution to import employees from different sources

You were given a few files that must be imported to the new database (see LabFiles). Your task is to create appropriate ETL solution. The following things should be taken into account:

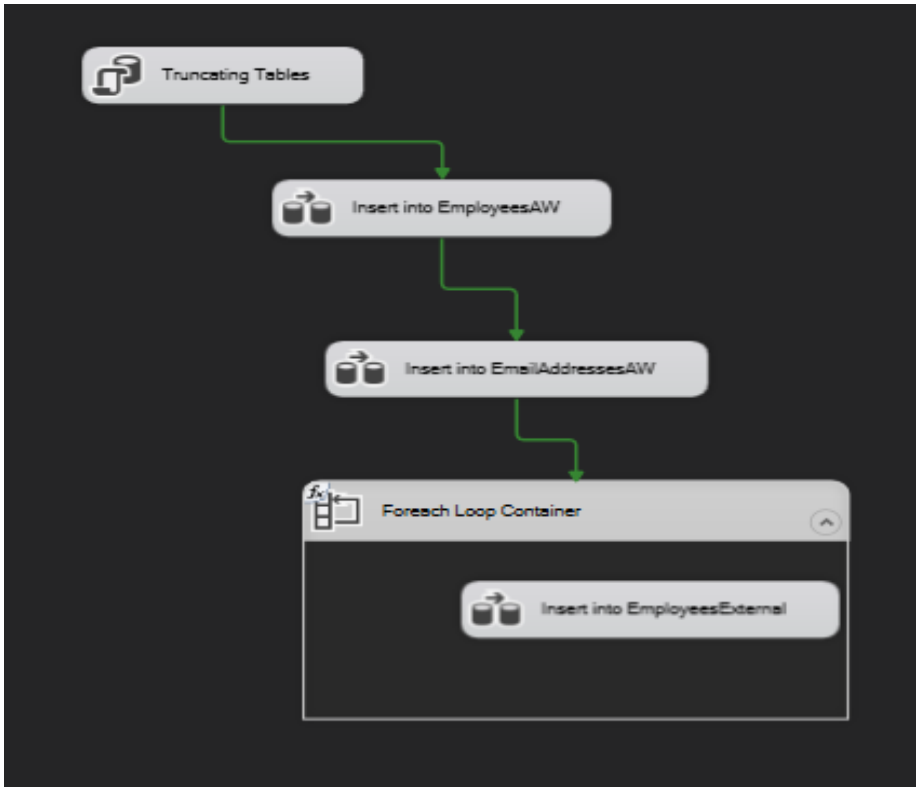
- All import process must be executed within one package
- All files must be imported to the table Employees External and the solution should work if the numbers of files will be increased
- The import process can be executed more than once (the destination tables are truncated every time)
- The table dbo.EmployeesAW must contain all employees from the database Adventure Works (use tables HumanResources.Employee and Person.Person to get needed data)
- The table dbo.EmailAddressesAW must contain all emails of employees from the database Adventure Works (see table Person.EmailAddress)
- In case of any error during the process the tables does not contain partially loaded data (they must be empty)
- If the import process is successful then the package sends an email message with the subject «Employees have been imported successfully», otherwise the subject should be «Error loading employees».

Note. So, now we have database consists of needed tables. We need to fulfil this tables with data using SSIS package (not to do this on DB level). Before that we need to truncate all this tables as it said in requirements to this Lab:

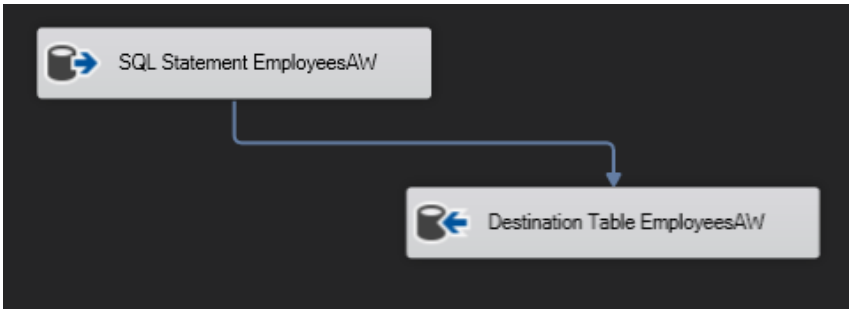



```
truncate table dbo.EmployeesAW;
truncate table dbo.EmailAddressesAW;
truncate table dbo.EmployeesExternal;
```

Result: SSIS solution.



Note. Let's now see how we insert data into tables dbo.EmployeesAW and dbo.EmailAddressesAW.



OLE DB connection manager:

LAPTOP-QH8GM32C.AdventureWorks2012

Data access mode:

SQL command

SQL command text:

```
select p.[BusinessEntityID],[FirstName],[LastName],[JobTitle]
        from [Person].[Person] p
                inner join [HumanResources].[Employee] e
                        on p.BusinessEntityID =
e.BusinessEntityID;
```

OLE DB Destination Editor

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

- Connection Manager
- Mappings
- Error Output

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

OLE DB connection manager:

LAPTOP-QH8GM32C.HumanResources

New...

Data access mode:

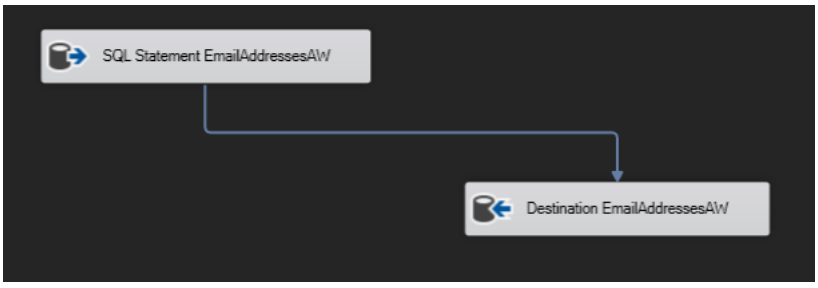
Table or view

Name of the table or the view:

[dbo].[EmployeesAW]

New...

View Existing



OLE DB connection manager:

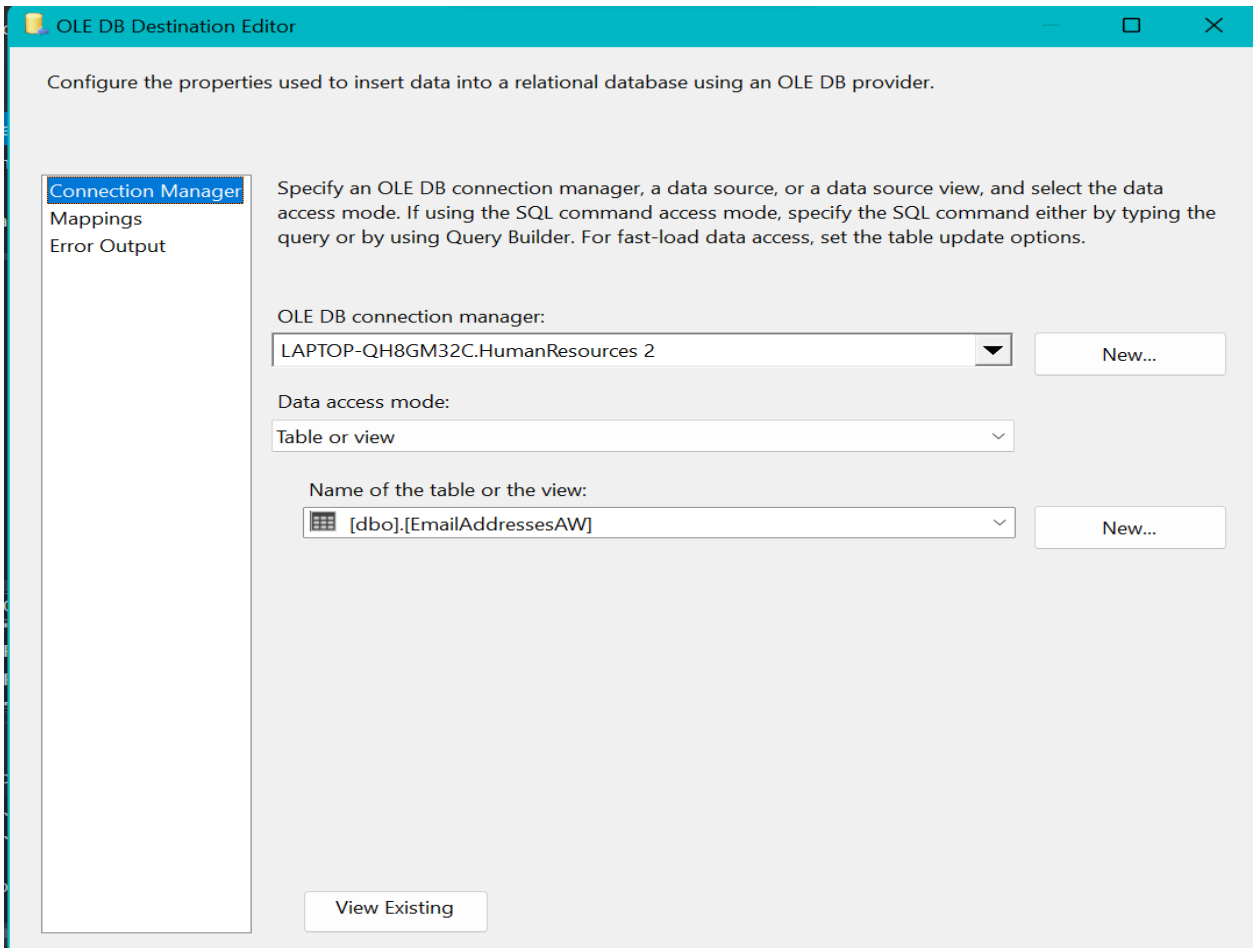
LAPTOP-QH8GM32C.AdventureWorks2012 4

Data access mode:

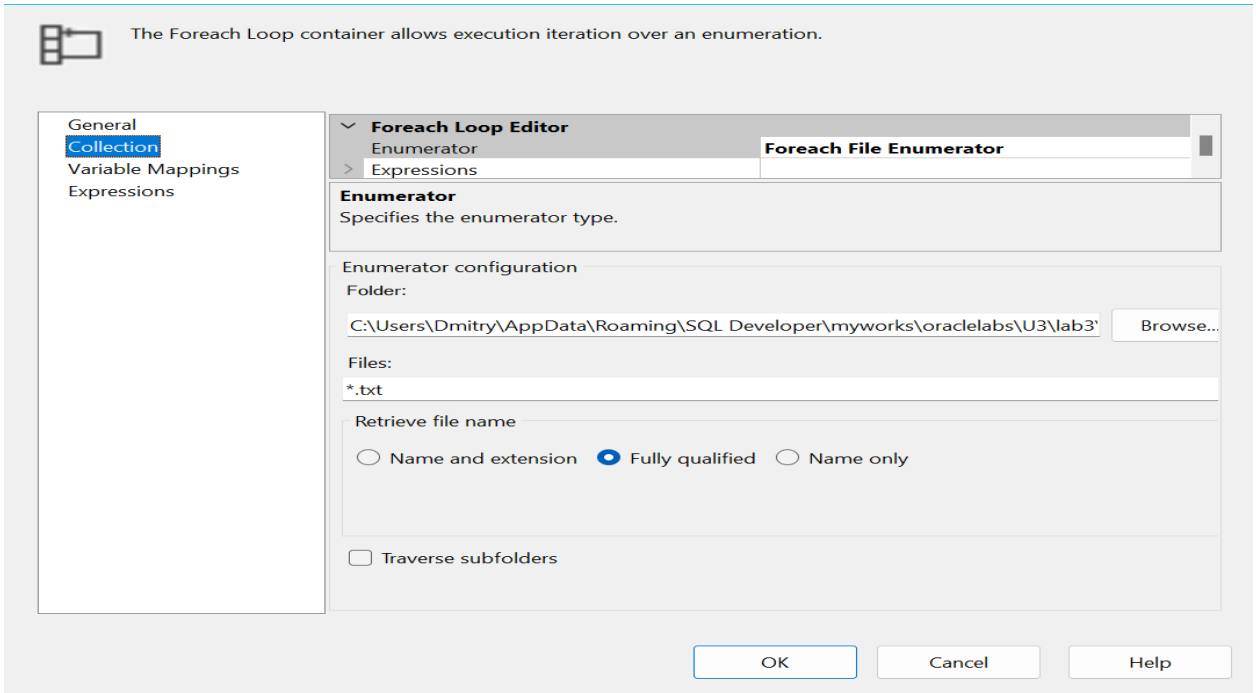
SQL command

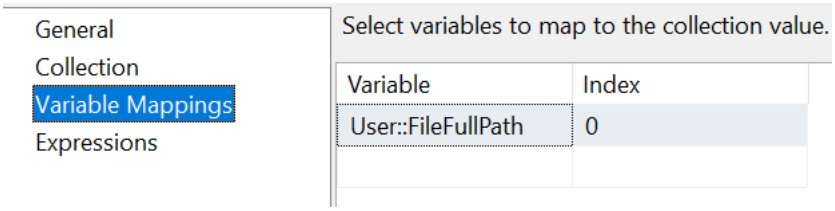
SQL command text:

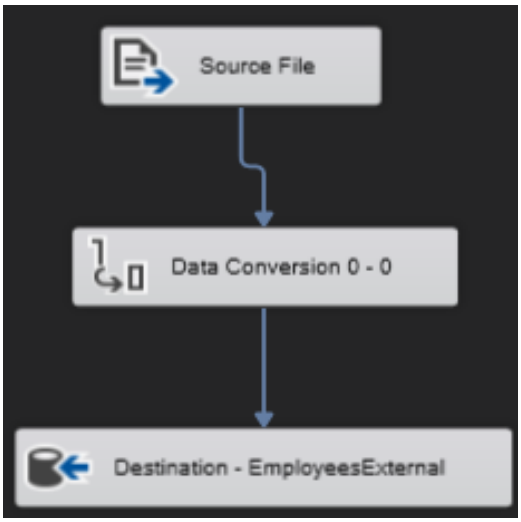
```
select [EmailAddressID],p.[BusinessEntityID],[EmailAddress]
        from [Person].[EmailAddress] e
                inner join [Person].[Person] p
                        on e.[BusinessEntityID] =
p.BusinessEntityID
```



Note. Now we have 2 tables with data. Let's insert data into EmployeesExternal, but using 2 source files to satisfy this requirement. I am going to use For Each Loop container to loop every *.txt file in source folder (so we can use any number of txt files).







Note. We can either try to specify encodings to load Unicode files but I chose to use Data Conversion, because we now can add any *.txt files with any encodings, so this way we can simplify our interaction with files in future.

- All files must be imported to the table Employees External and the solution should work if the numbers of files will be increased

```
select * from dbo.EmailAddressesAW;
select * from dbo.EmployeesExternal;
```

100 %

Results | Messages

	EmployeeID	FirstName	LastName	JobTitle	EmailAddress	City	StateProvinceName	CountryRegionName
269	269	Dan	Bacon	Application Specialist	dan0@adventure-works.com	Issaquah	Washington	United States
270	270	FranÃ§ois	Ajenstat	Database Administrator	franÃ§ois0@adventure-works....	Issaquah	Washington	United States
271	271	Dan	Wilson	Database Administrator	dan1@adventure-works.com	Bellevue	Washington	United States
272	272	Janaina	Bueno	Application Specialist	janaina0@adventure-works.com	Issaquah	Washington	United States
273	273	Brian	Welcker	Vice President of Sales	brian3@adventure-works.com	Issaquah	Washington	United States
274	274	Stephen	Jiang	North American Sales Manager	stephen0@adventure-works.c...	Redmond	Washington	United States
275	275	Michael	Blythe	Sales Representative	michael9@adventure-works.com	Detroit	Michigan	United States
276	276	Linda	Mitchell	Sales Representative	linda3@adventure-works.com	Nevada	Utah	United States
277	277	Jillian	Carson	Sales Representative	jillian0@adventure-works.com	Duluth	Minnesota	United States
278	278	Garrett	Vargas	Sales Representative	garrett1@adventure-works.com	Calgary	Alberta	Canada
279	279	Tsvi	Reiter	Sales Representative	tsvi0@adventure-works.com	Memphis	Tennessee	United States
280	280	Pamela	Ansman-...	Sales Representative	pamela0@adventure-works.com	Portland	Oregon	United States
281	281	Shu	Ito	Sales Representative	shu0@adventure-works.com	San Francisco	California	United States
282	282	JosÃ©	Saraiva	Sales Representative	josÃ©1@adventure-works.com	Ottawa	Ontario	Canada

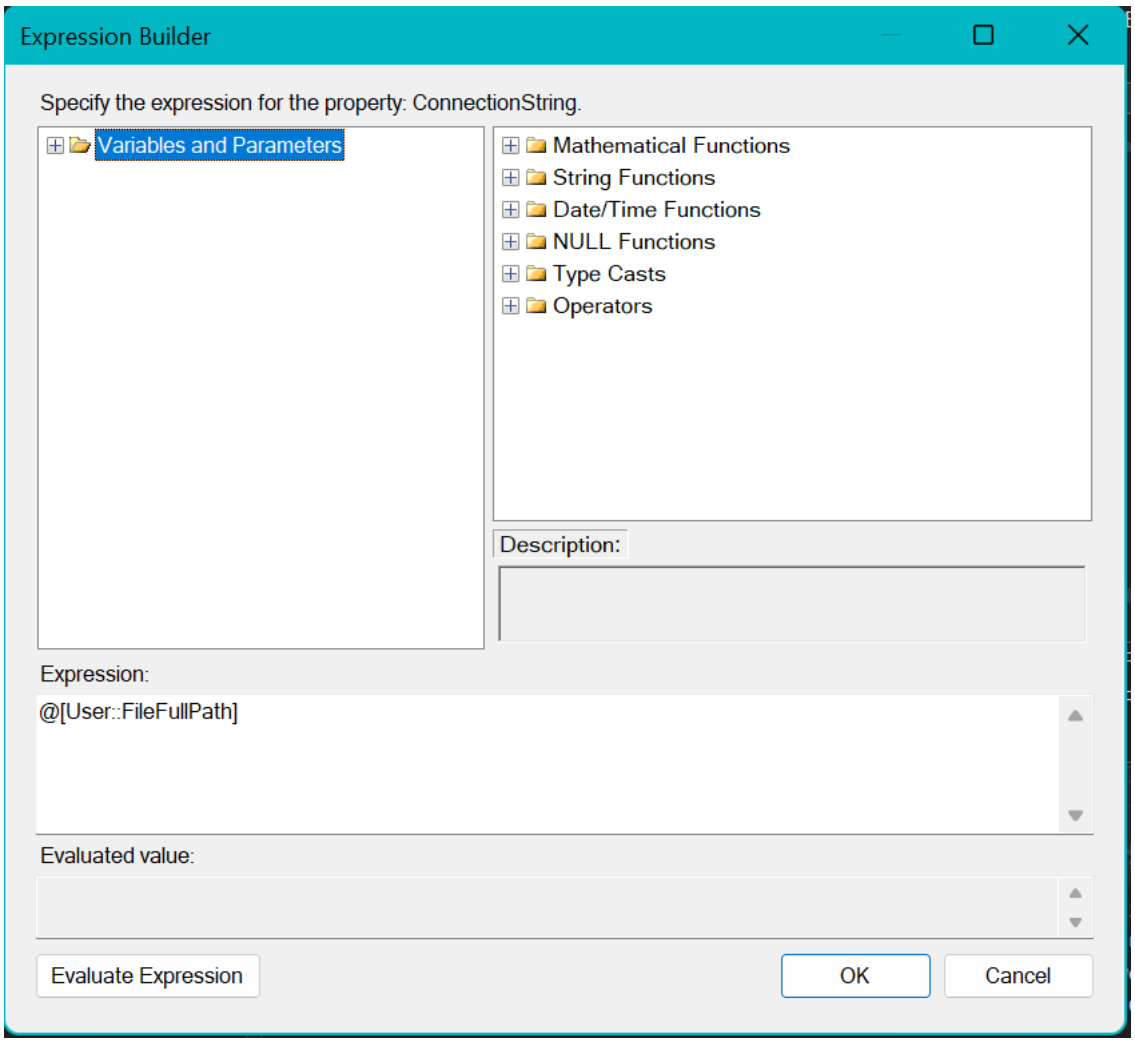
```
select * from dbo.EmployeesAW;
select * from dbo.EmailAddressesAW;
select * from dbo.EmployeesExternal;
```

100 %

Results | Messages

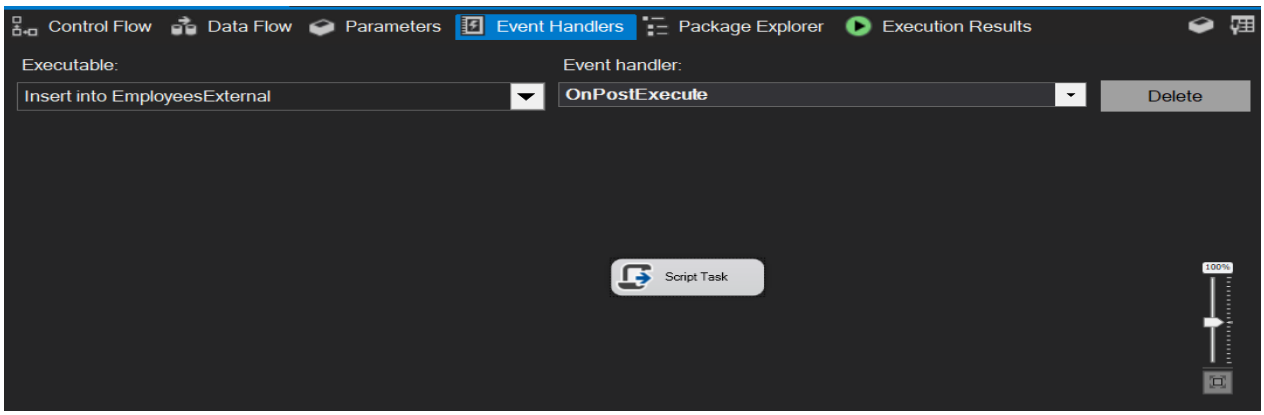
	BusinessEntityID	FirstName	LastName	JobTitle
87	87	Cristian	Petculescu	Production Supervisor - WC10
88	88	Betsy	Stadick	Production Technician - WC10
89	89	Patrick	Wedge	Production Technician - WC10
90	90	Danielle	Tiedt	Production Technician - WC10
91	91	Kimberly	Zimmerm...	Production Technician - WC10
92	92	Tom	Vande V...	Production Technician - WC10

	EmailAddressId	BusinessEntityID	EmailAddress
23	23	23	mary0@adventure-works.com
24	24	24	jill0@adventure-works.com
25	25	25	james1@adventure-works.com
26	26	26	peter0@adventure-works.com
27	27	27	jo0@adventure-works.com
28	28	28	guy1@adventure-works.com



Note. I have also specified an expression in connection used by flat file destination to fulfil it with next file its path by loop.

Note. Now we need to get an email in case of successfully loaded data.



Variables

Name	Scope	Data type	Value
EmailFrom	Package	String	dmitryladutko120...
EmailSubject	Package	String	Employees Loading
EmailTo	Package	String	ladutko98@inbox.ru
FileFullPath	Package	String	C:\Users\Dmitry\A...
MessageBody	Package	String	Employees have b...
Root_directory	Package	String	C:\Users\Dmitry\A...
SMTPPasswo...	Package	String	7373838gsgshsh
SMTPPort	Package	Int32	587
SMTPServer	Package	String	smtp.gmail.com
SMTPUserna...	Package	String	dmitryladutko120...

ScriptMain.cs

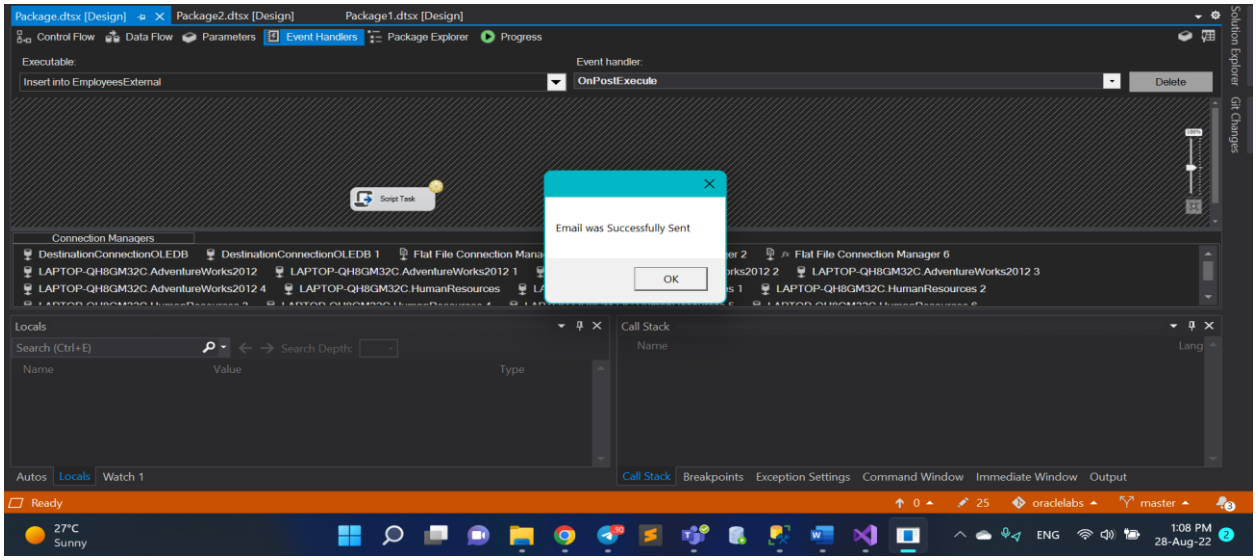
ST_a3472afa195e4a3cbfc2f7c6ba38e | ST_a3472afa195e4a3cbfc2f7c6ba38e | Main()

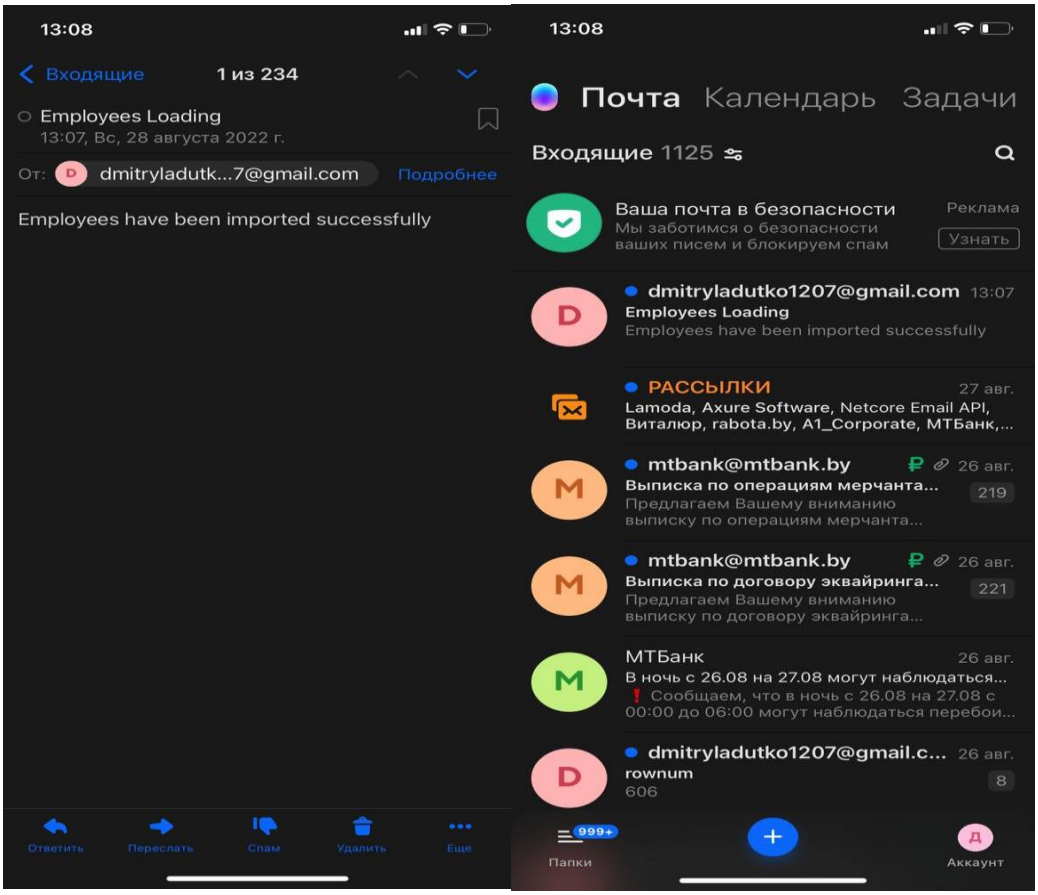
```
0 references
public void Main()
{
    // TODO: Add your code here
    // C# code
    String SendMailFrom = Dts.Variables["EmailFrom"].Value.ToString();
    String SendMailTo = Dts.Variables["EmailTo"].Value.ToString();
    String SendMailSubject = Dts.Variables["EmailSubject"].Value.ToString();
    String SendMailBody = Dts.Variables["MessageBody"].Value.ToString();

    try
    {
        MailMessage email = new MailMessage();
        SmtpClient SmtpServer = new SmtpClient("smtp.gmail.com");
        // START
        email.From = new MailAddress(SendMailFrom);
        email.To.Add(SendMailTo);
        email.Subject = SendMailSubject;
        email.Body = SendMailBody;
        //END

        SmtpServer.Port = 587;
        SmtpServer.Credentials = new System.Net.NetworkCredential(SendMailFrom, "
        SmtpServer.EnableSsl = true;

        SmtpServer.Send(email);
        MessageBox.Show("Email was Successfully Sent ");
    }
```





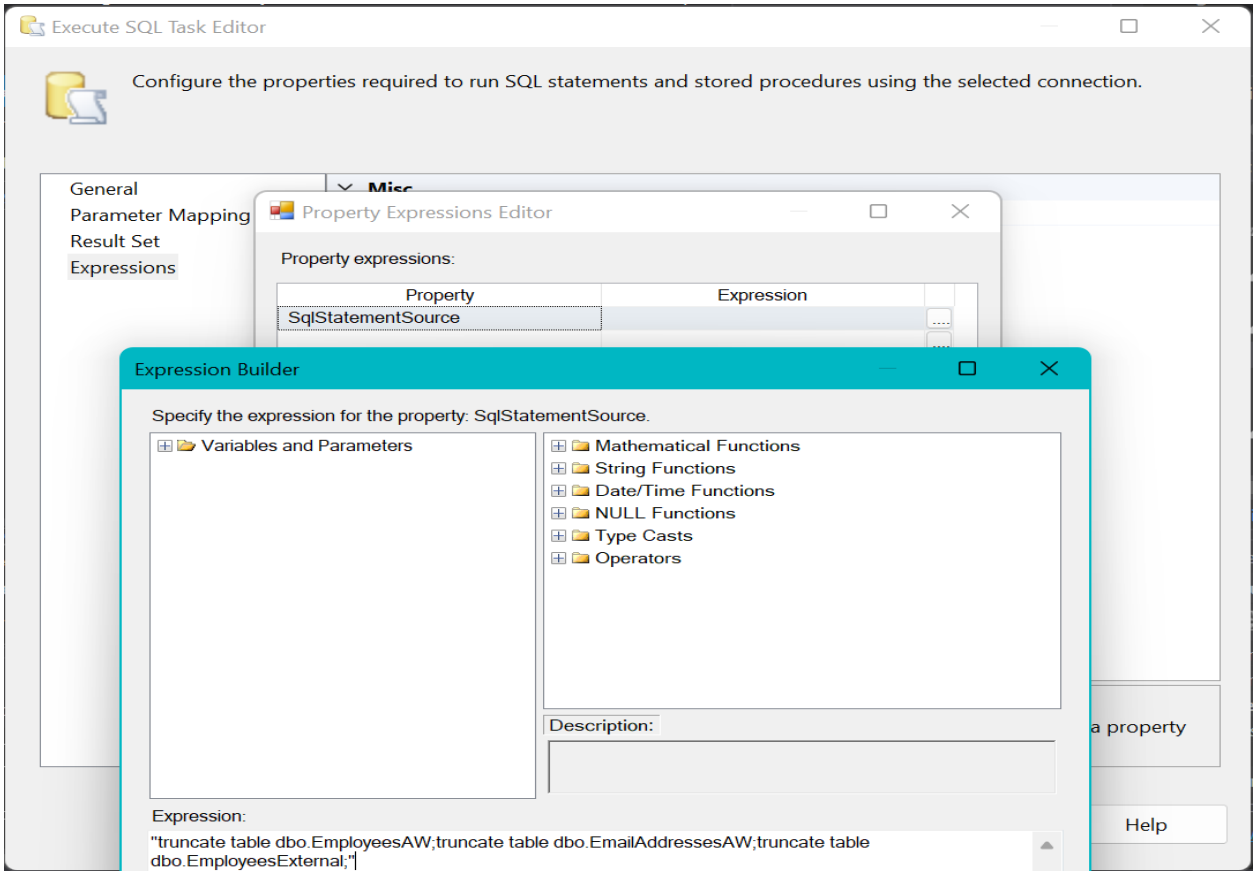
Note. Now we also need to receive a mail in case of errors in loading employees

I added a variable MessageError for it. And used the same script to send an email but changed MessageBody to MessageError variable to send another text, also added one more event on error to send different messages depend on result (as it said in requirements).

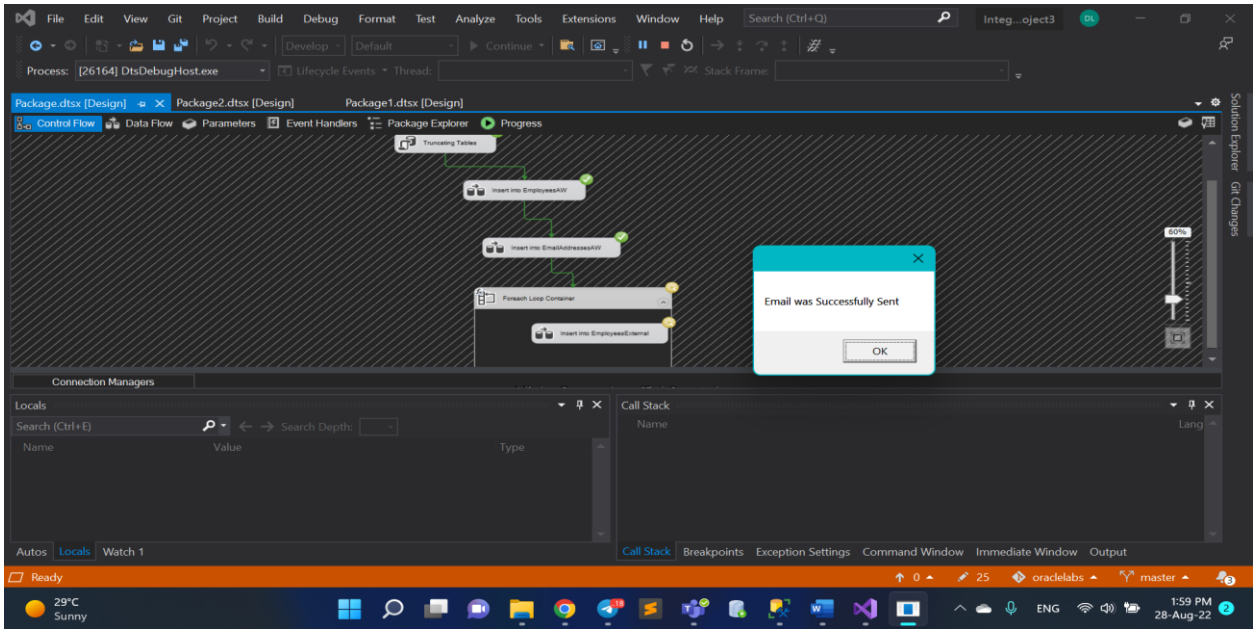
Variables

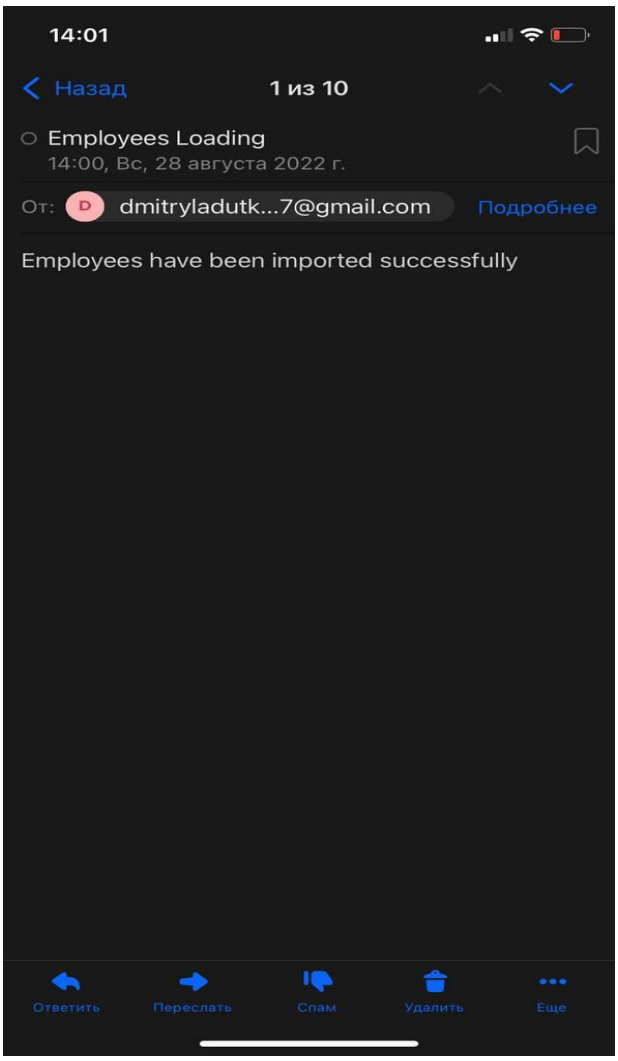
Name	Scope	Data type	Value
EmailFrom	Package	String	dmitryladutko120...
EmailSubject	Package	String	Employees Loading
EmailTo	Package	String	ladutko98@inbox.ru
FileFullPath	Package	String	C:\Users\Dmitry\A...
MessageBody	Package	String	Employees have b...
MessageError	Package	String	Error loading Empl...
Root_directory	Package	String	C:\Users\Dmitry\A...
SMTPPasswo...	Package	String	7373838gsgshsh
SMTPPort	Package	Int32	587
SMTPServer	Package	String	smtp.gmail.com
SMTPUserna...	Package	String	dmitryladutko120...

Note. The last requirement that I need to realize is that if there is (are) some error(-s) we do not have to insert partially data.



Note. As you can see all of the requirements are satisfied! To check it one more time let's execute the whole package once again (pay attention to the time):





3. Exercise 3: Creating ETL solution to export the data from HumanResourcesdb

You should generate two comma separated text files with names EmpSales.txt and EmpNotSales .txt based on employee data (dbo.EmployeesAW and dbo.EmailAddressesAW). The files must have the following structures:

EmpSales.txt:

FirstName, Last Name, EmailAddress

EmpNotSales.txt:

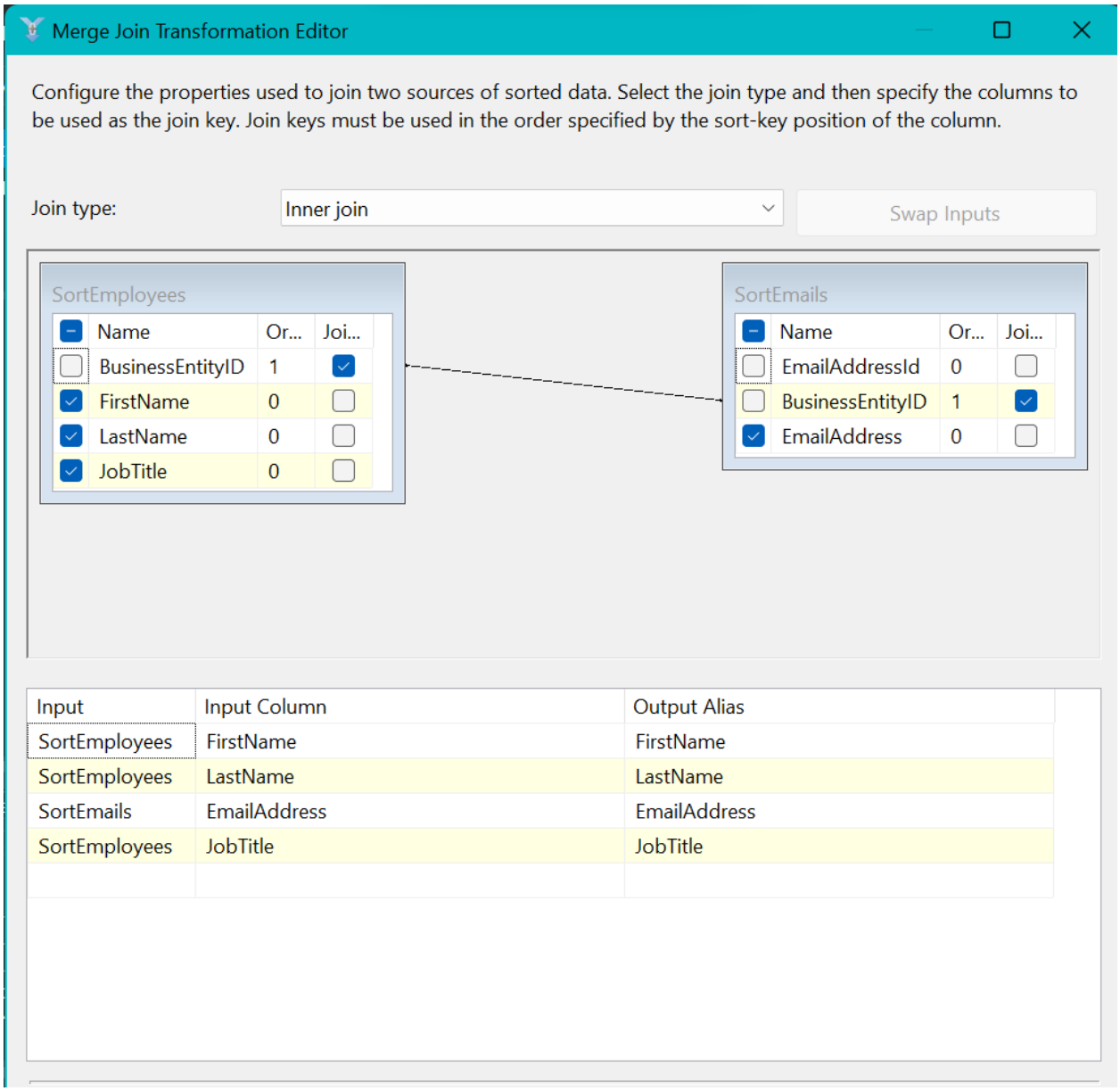
FirstName, LastName, EmailAddress

Employee e-mail addresses are stored separately from employee details, so your data flow must include a lookup to retrieve e-mail addresses based on the

employee ID (using SQL Query to join the data is not allowed). The data in the files must be sorted by the employees' last names and then first names. The first text file should include only employees with JobTitle = “Sales Representative”. The second text file should include all other employees. The files should be generated within one data flow task.

Result: SSIS solution and the output files.

Note. Now we need to create appropriate task to eventually get file output structure that is shown above. I used 2 sources (dbo.EmployeesAW and dbo.EmailAddressesAW) tables, then I used sorting to get BusinessEntityID from both tables so I can join them with merge join



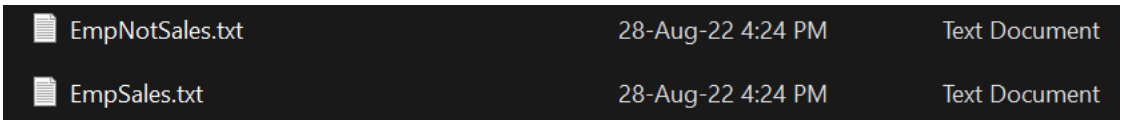
Note. Also we need to split it into 2 files with following structure:

EmpSales.txt:

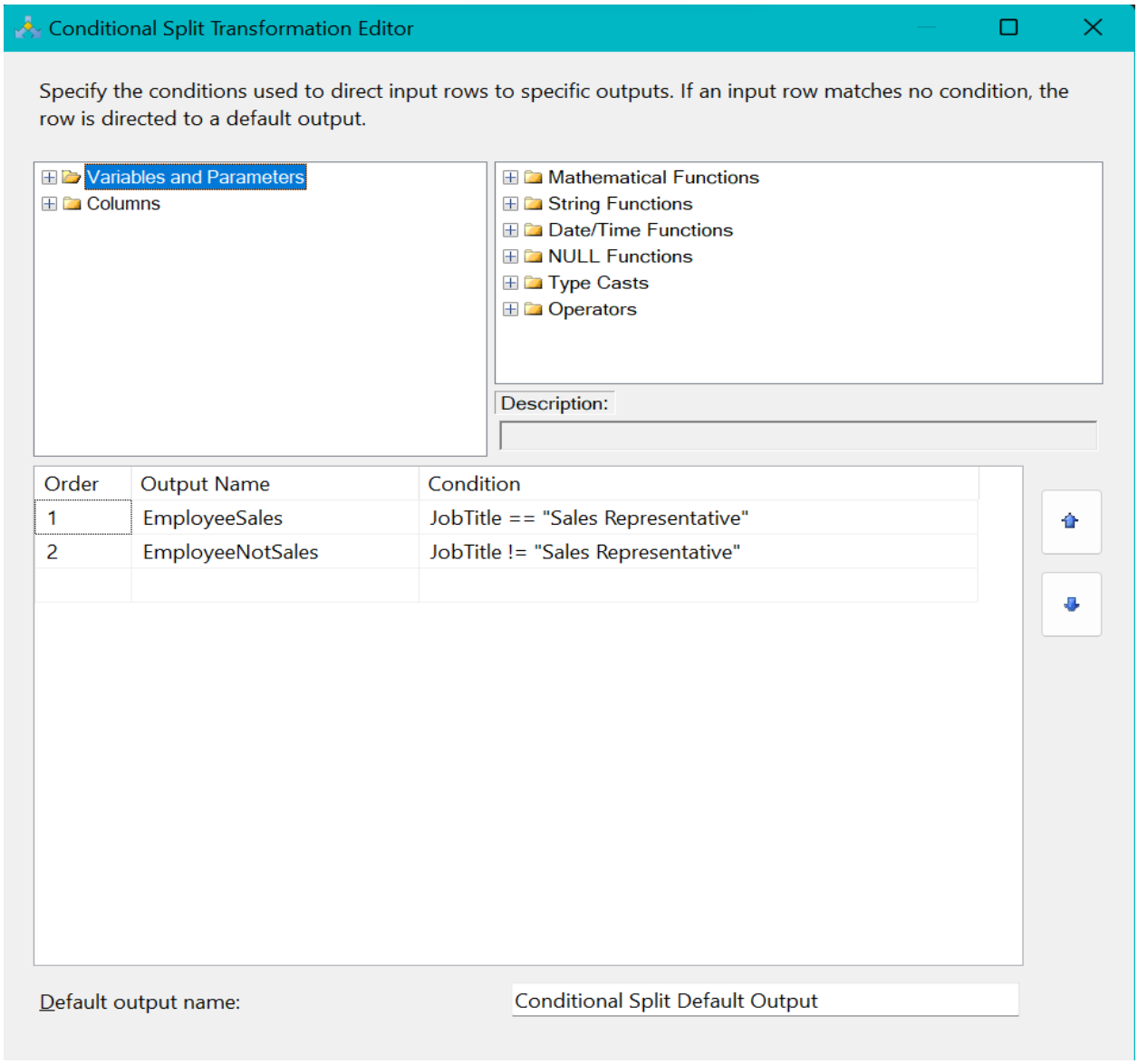
FirstName, Last Name, EmailAddress

EmpNotSales.txt:

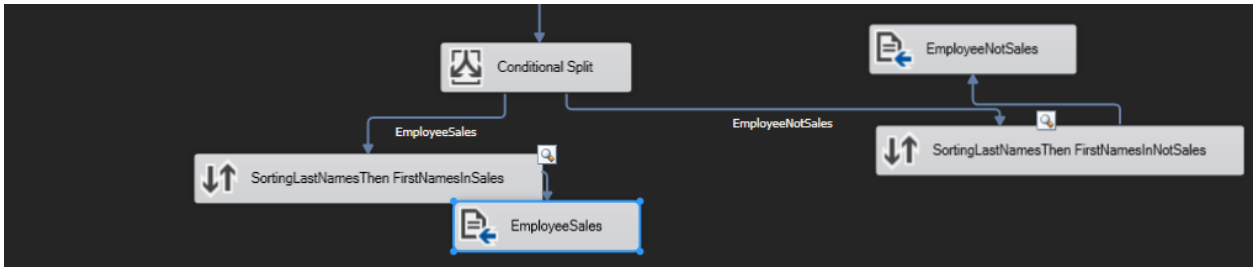
FirstName, LastName, EmailAddress

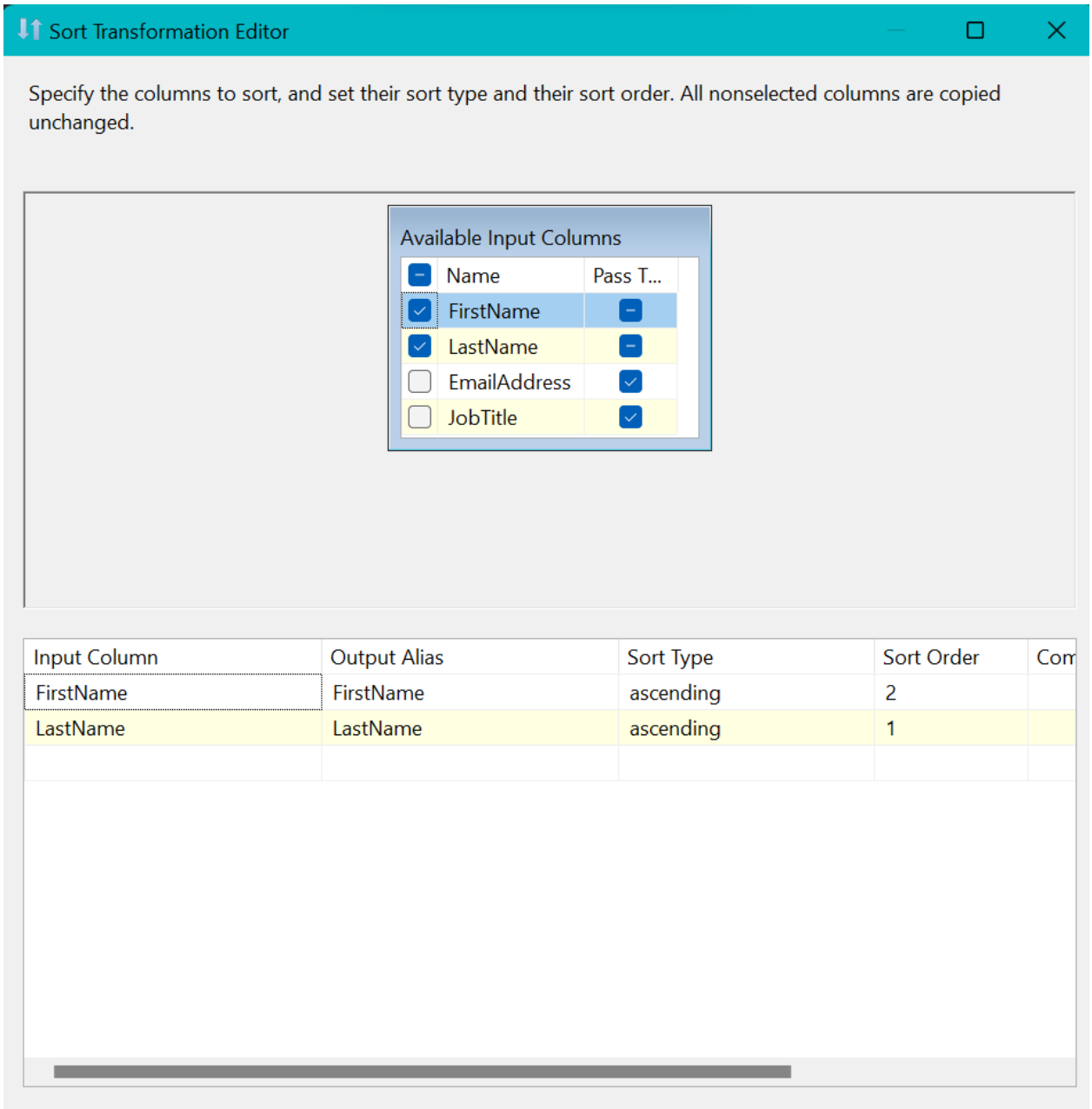


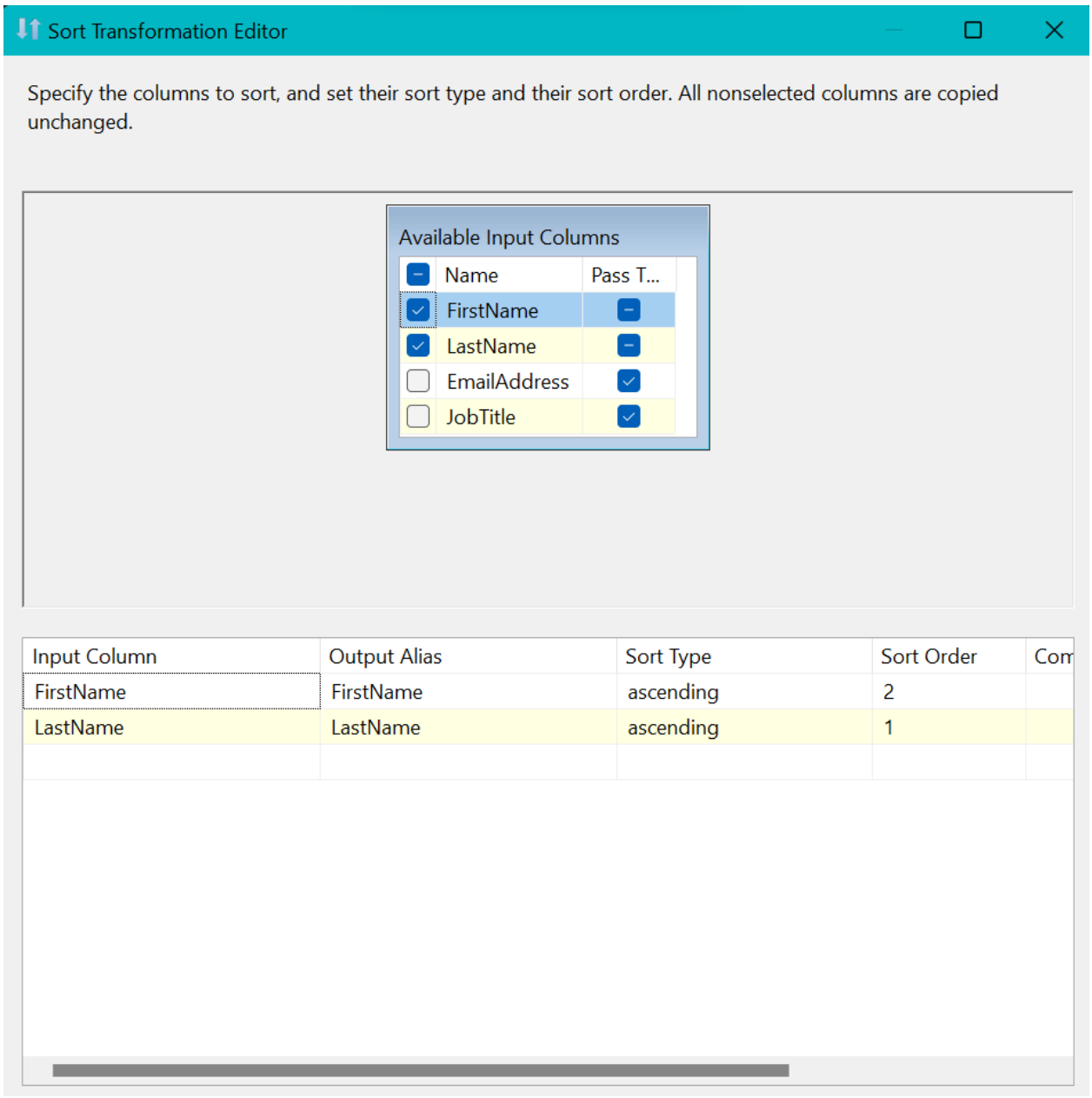
Note. As it said in requirements, we need to divide employees by their JobTitle into 2 groups (Sales And NotSales). At the screenshot below you can see conditions I used to satisfy it.

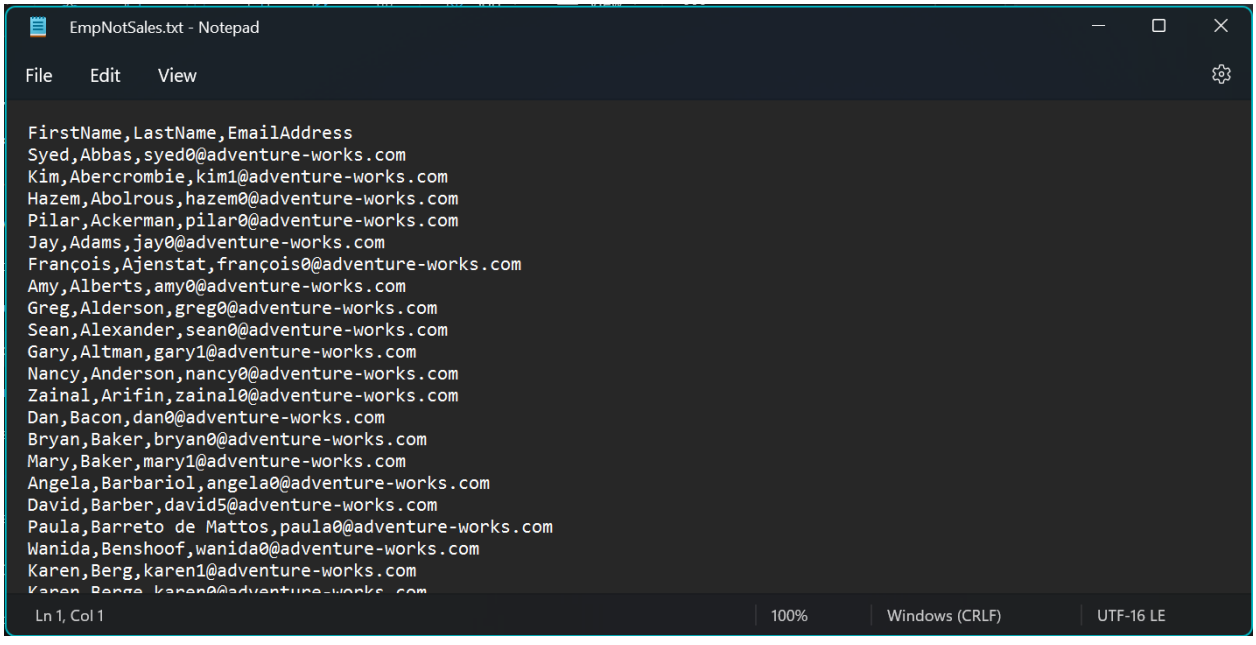


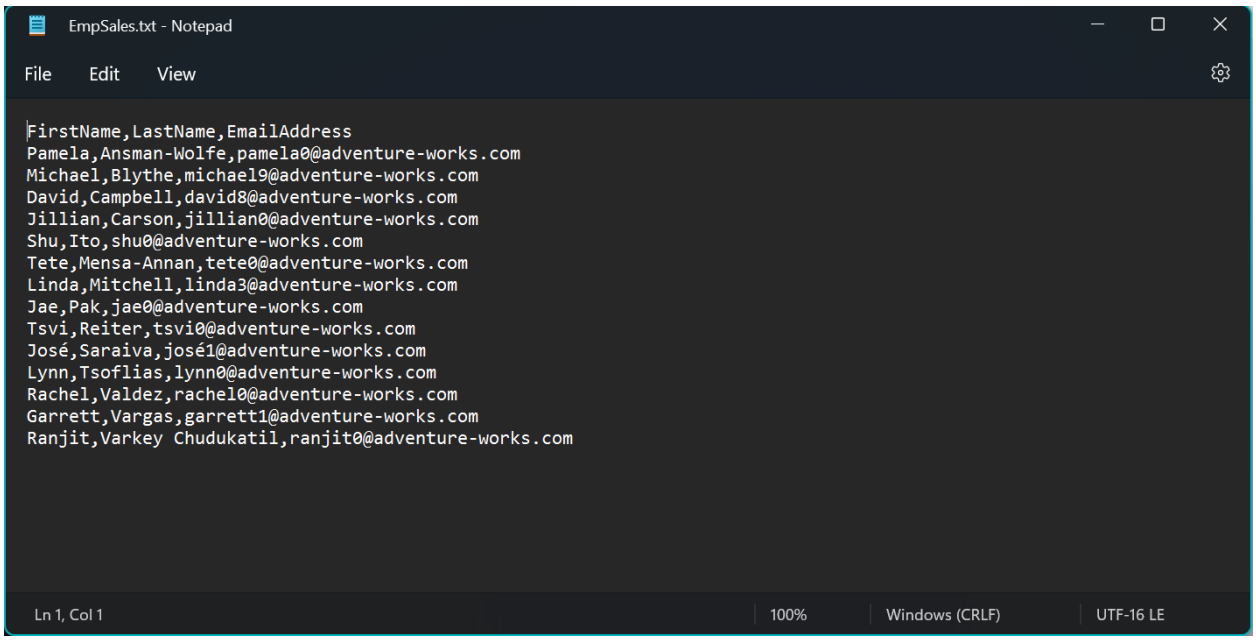
Note. It also were given that I need to sort Employees by their Last Names and then their First Names in both groups. Thus I added sorting for them.

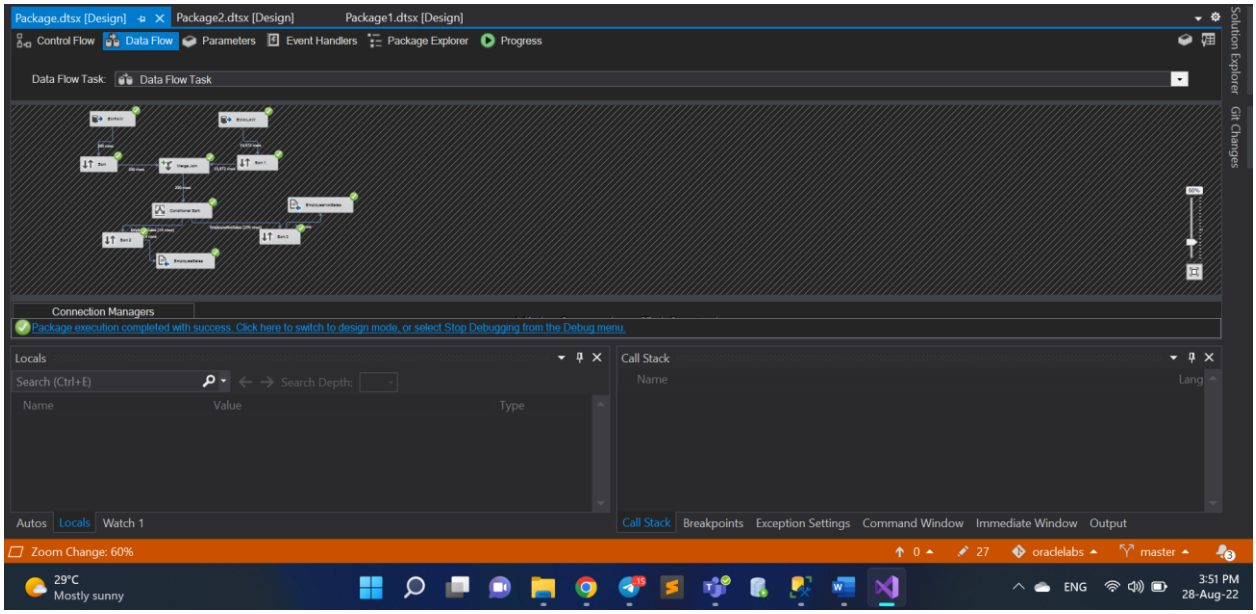





Sort Transformation Editor
Specify the columns to sort, and set their sort type and their sort order. All nonselected columns are copied unchanged.
Available Input Columns
Name
Pass T...
FirstName
LastName
EmailAddress
JobTitle
Input Column
Output Alias
Sort Type
Sort Order
Com
FirstName
FirstName
ascending
2
LastName
LastName
ascending
1


EmpNotSales.txt - Notepad
File
Edit
View
FirstName,LastName,EmailAddress
Syed,Abbas,syed0@adventure-works.com
Kim,Abercrombie,kim1@adventure-works.com
Hazem,Abolrous,hazem0@adventure-works.com
Pilar,Ackerman,pilar0@adventure-works.com
Jay,Adams,jay0@adventure-works.com
François,Ajenstat,françois0@adventure-works.com
Amy,Alberts,amy0@adventure-works.com
Greg,Alderson,greg0@adventure-works.com
Sean,Alexander,sean0@adventure-works.com
Gary,Altman,gary1@adventure-works.com
Nancy,Anderson,nancy0@adventure-works.com
Zainal,Arifin,zainal0@adventure-works.com
Dan,Bacon,dan0@adventure-works.com
Bryan,Baker,bryan0@adventure-works.com
Mary,Baker,mary1@adventure-works.com
Angela,Barbariol,angela0@adventure-works.com
David,Barber,david5@adventure-works.com
Paula,Barreto de Mattos,paula0@adventure-works.com
Wanida,Benshoof,wanida0@adventure-works.com
Karen,Berg,karen1@adventure-works.com
Ln 1, Col 1
100%
Windows (CRLF)
UTF-16 LE





Note. Now please pay attention to destination flat files. Firstly, last names are sorted and then, in group of current character first names are sorted.
All the requirements are satisfied, package executes correctly.

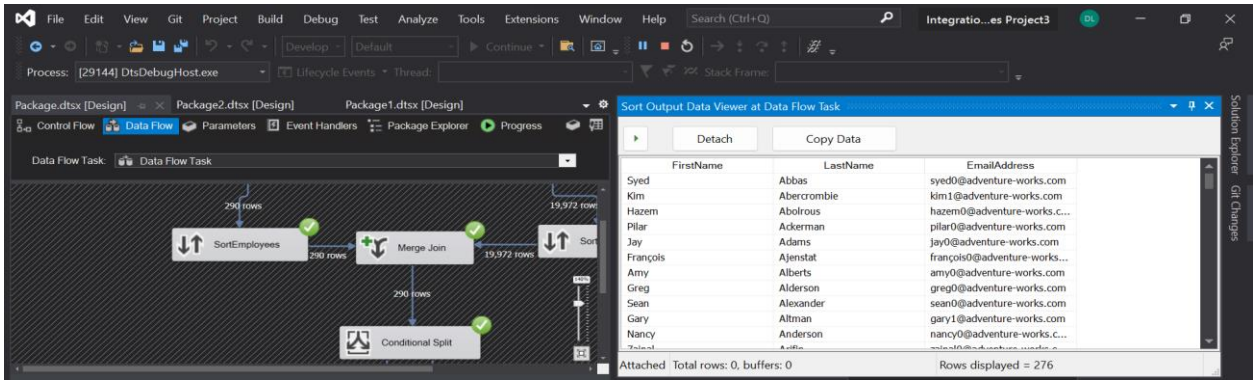
4. Exercise 4: Using Data Viewers

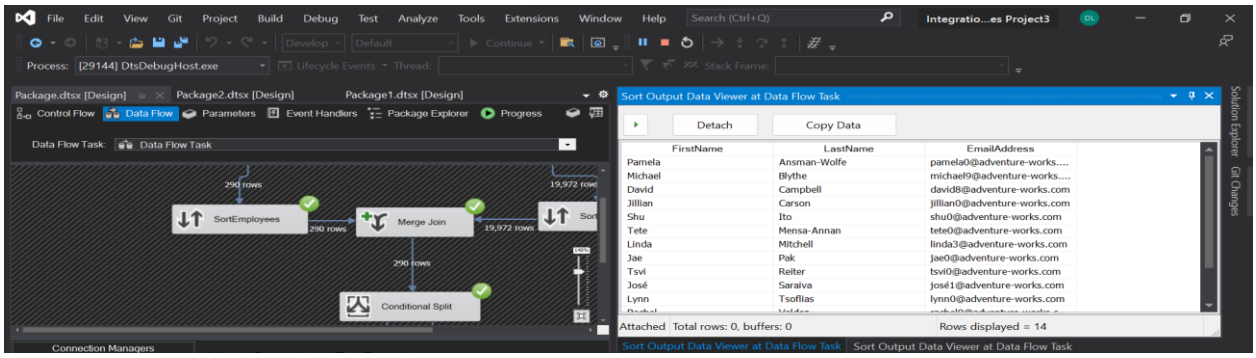
Now that you have created your data flows, you need to verify and test their functionality using data viewers.

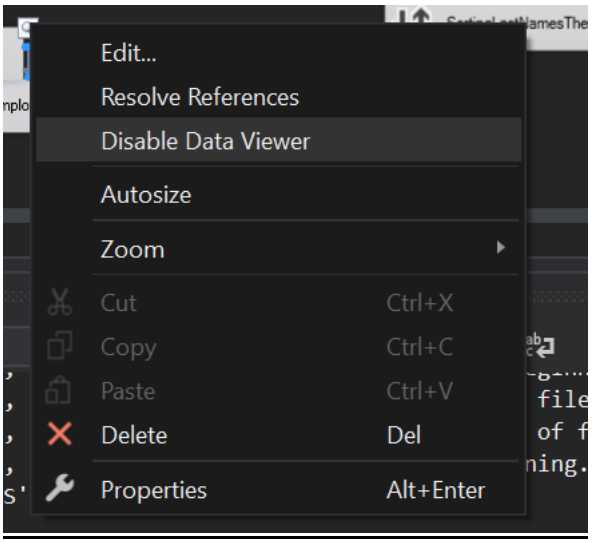
This exercise's main tasks are:

1. Add a grid data viewer.

2. Run the package, and view the data viewer.
3. Remove the data viewer.







Note. In fact we can check files manually after package has been executed but it is much more convenient to use Grid Data Viewer to see the data.

Result: Screenshots of the viewers

Laboratory Work Summary

At this Laboratory Work we more practiced about building package **solutions in SSIS.** Now we know how we could work with different objects to make a good **data flow.** We used lot's of instruments to **satisfy listed requirements**. Practiced more how we can get useful **notifications about data loadings**, used **data converter**s to create **multifunctional loading schemas.**

We also **were introduces** to the **data viewer, merge join and conditional split functionality.**

P.S. The solution is executable from **one package**. I only putted the part with creating external file into another package to logically separate them.