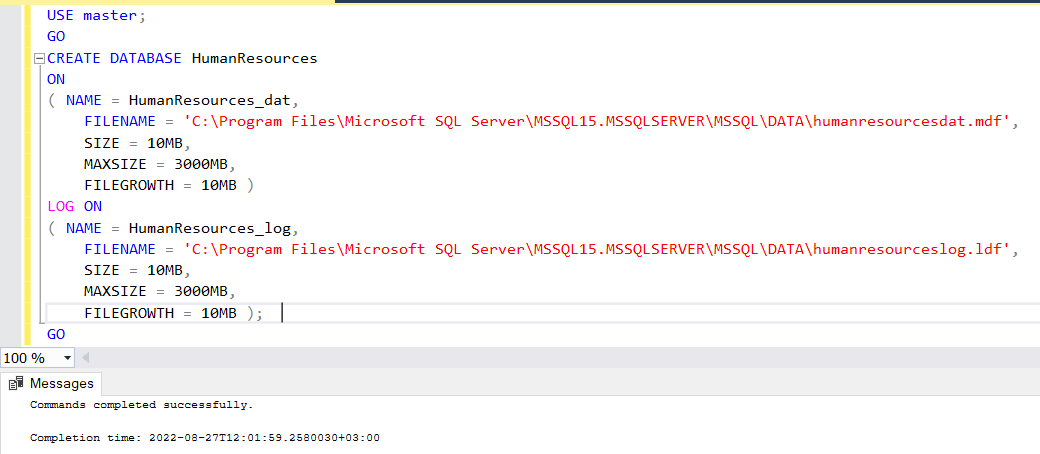
**Anton Slizh’s**

**U3M3.LW.Implementing Data Flow**

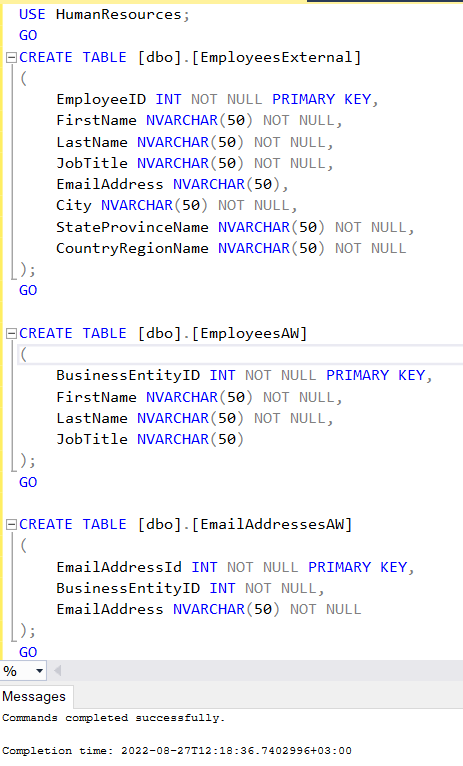
**Task 1**

# 1. Exercise 1: Creating new database for HR department

Creating a database “HumanResources”



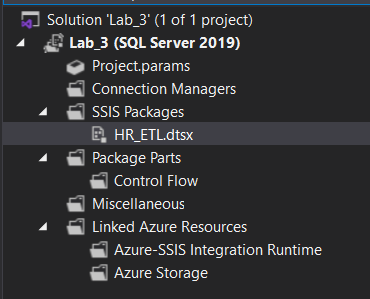
Creating tables in the database

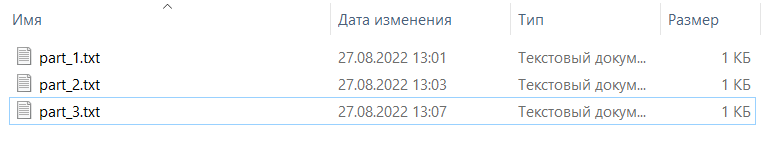


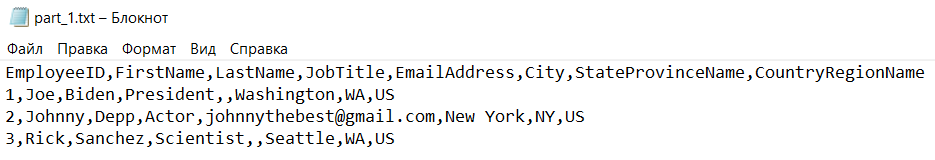
**Task 2**

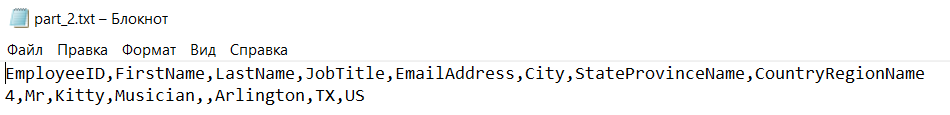
# 2. Exercise 2: Creating ETL solution to import employees from different sources

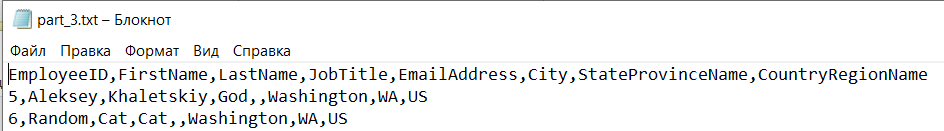
At the beginning create the SSIS project and prepare data files.



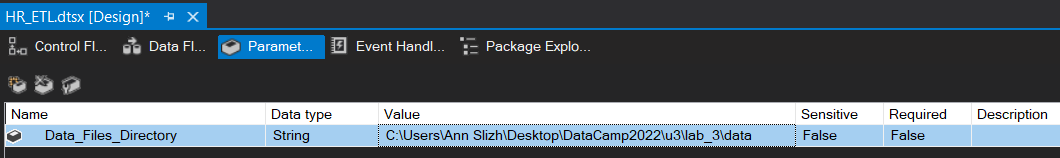








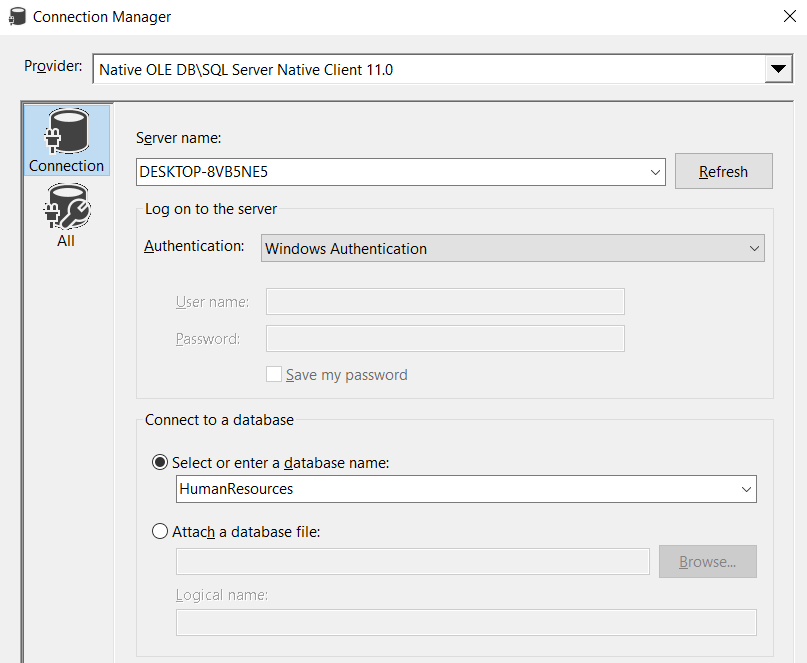
Then create the package parameter which contains the path to directory with data files

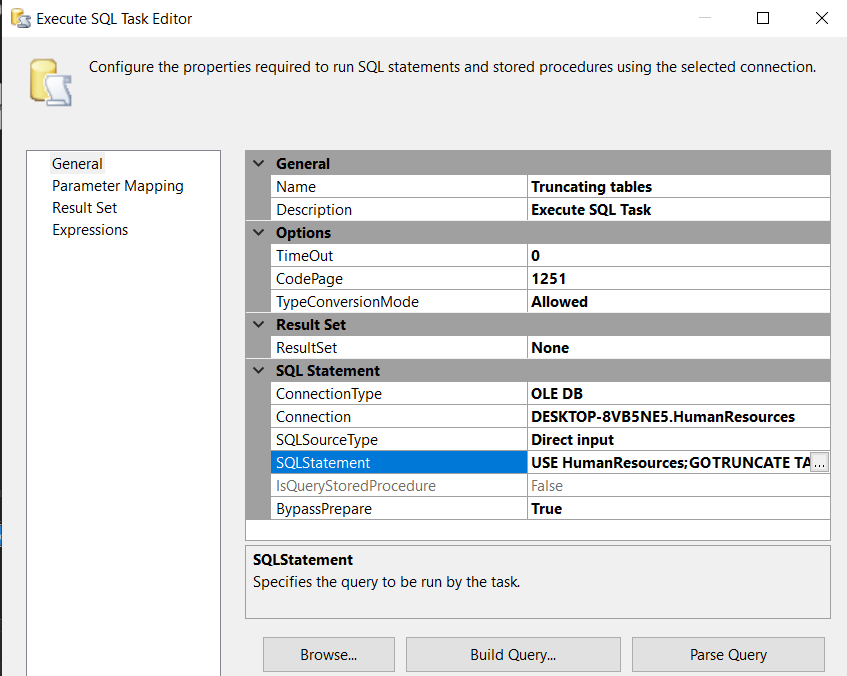


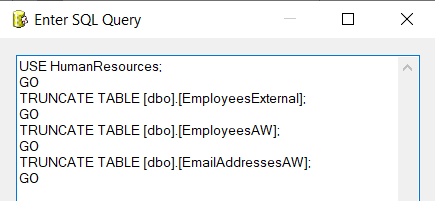
Now we start building ETL pipeline.

1. **Create SQL Task which will truncate all tables**

As we use *HumarResources* database first time, we need create and save new connection

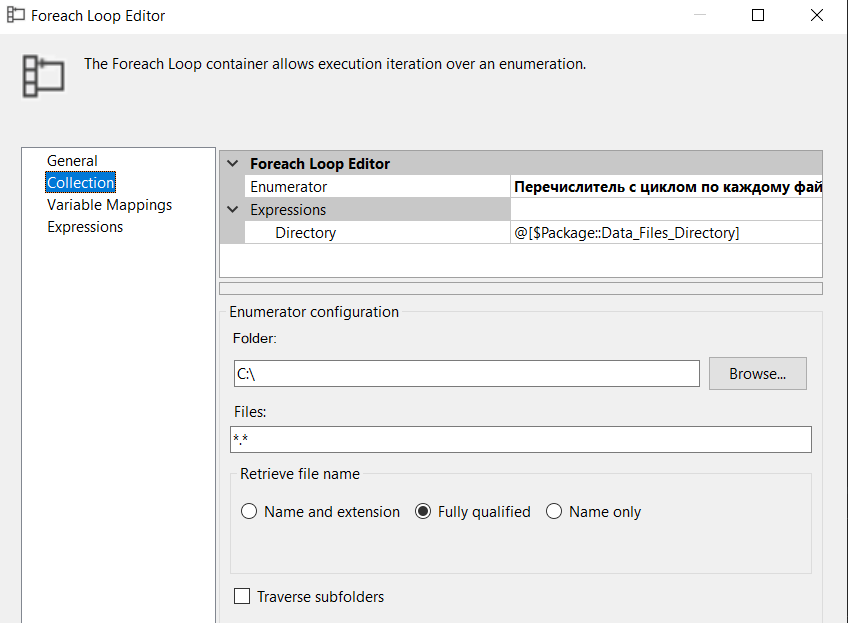


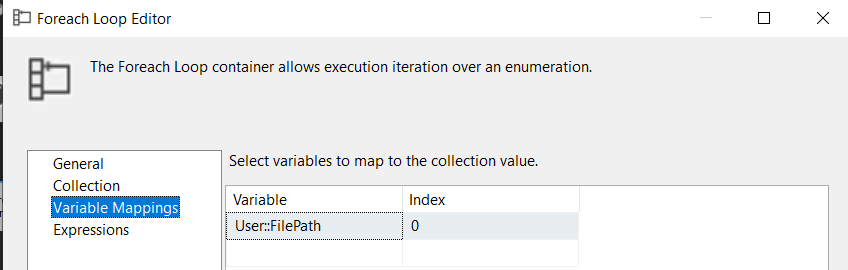




1. **Create load files process. Data should be loaded from Flat Files into *dbo.EmployeesExternal* table in the database**

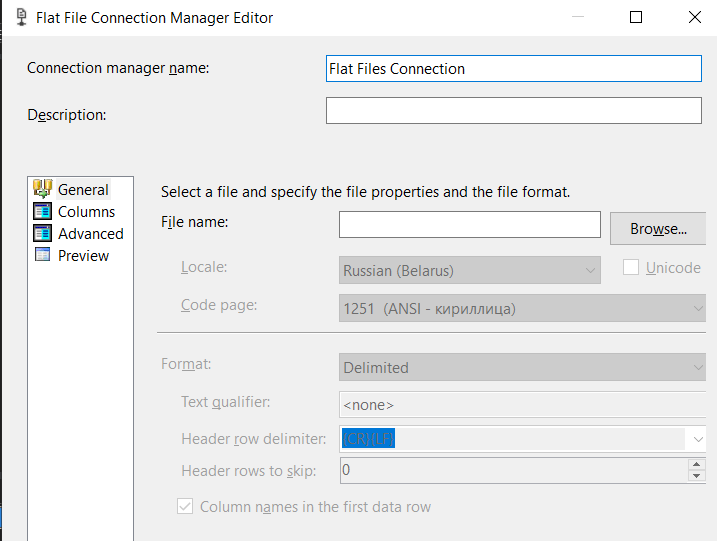
We will use Foreach Loop Container to load all files from directory which was specified in package parameters.



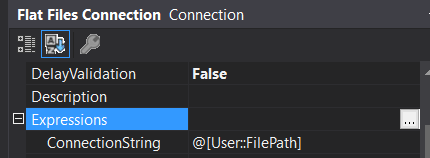


Then create a Data Flow Task which will load every file in definite loop iteration into database.

Create Flat File Connection.

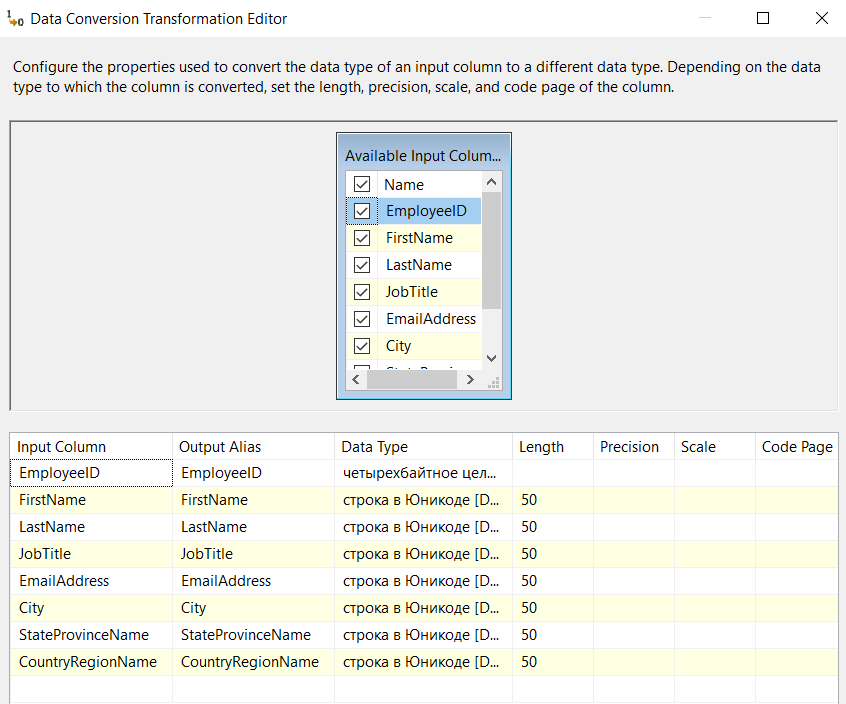


We didn’t define the file name, because we need to load undefined number of files. Instead of file name we should specify the Property Expression for our connection. We need to specify the variable which contains the file path from loop iteration.

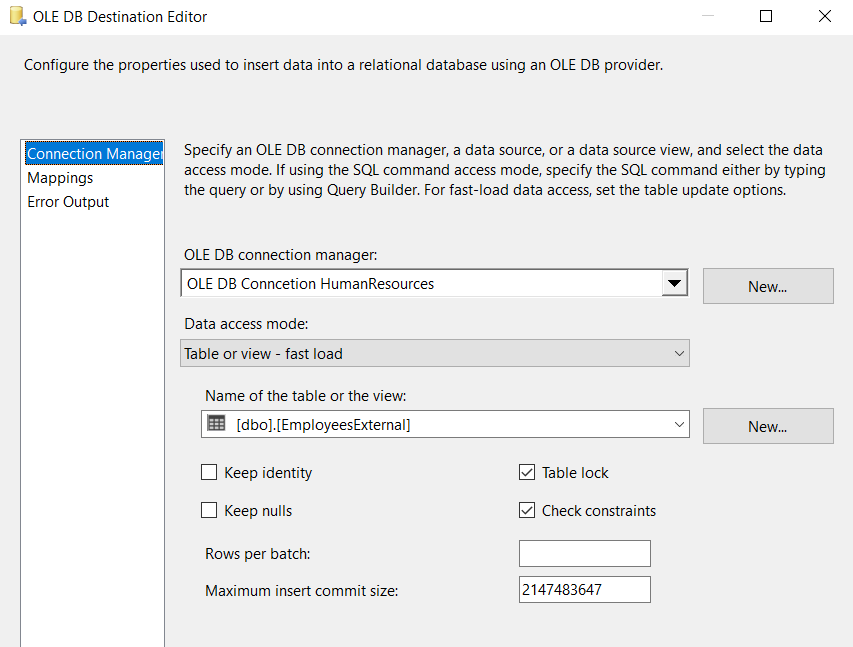


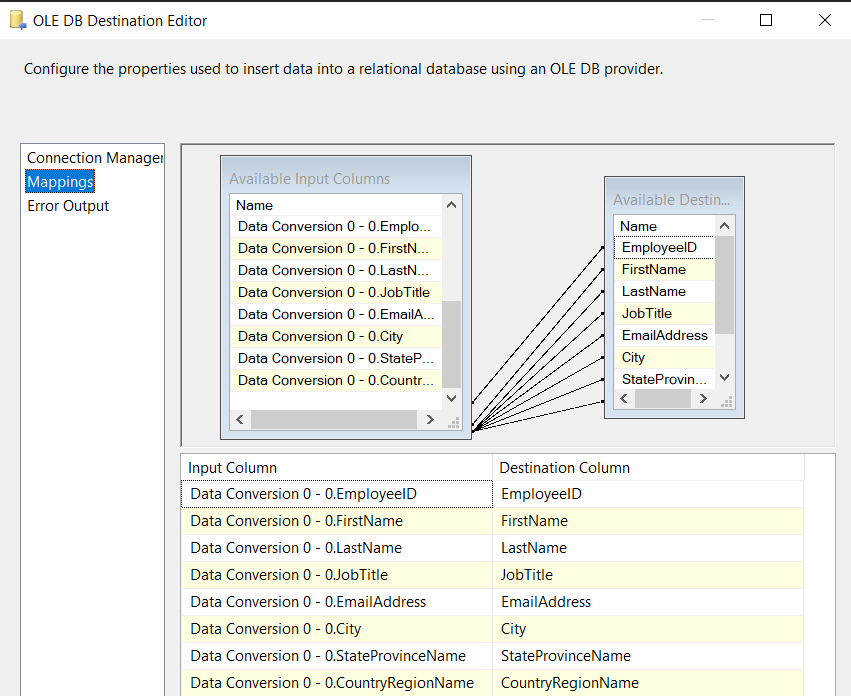
So, this way using foreach loop we can load undefined number of files from the specified directory.

Create Data Conversion

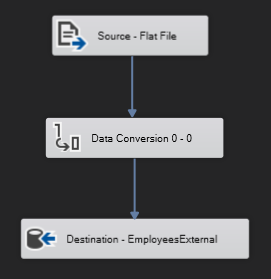


Create Destination OLE DB

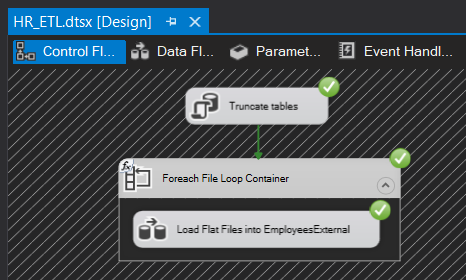




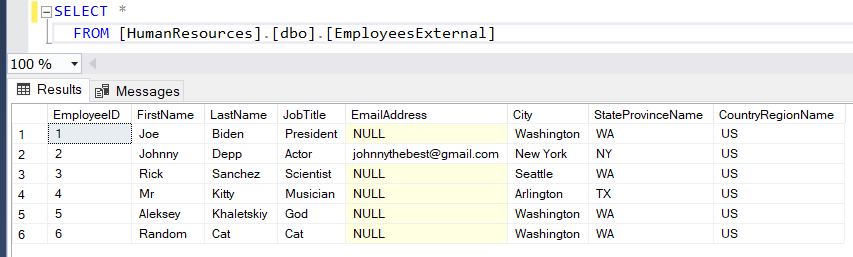
Prepared Data Flow Task:



Prepared part of ETL pipeline

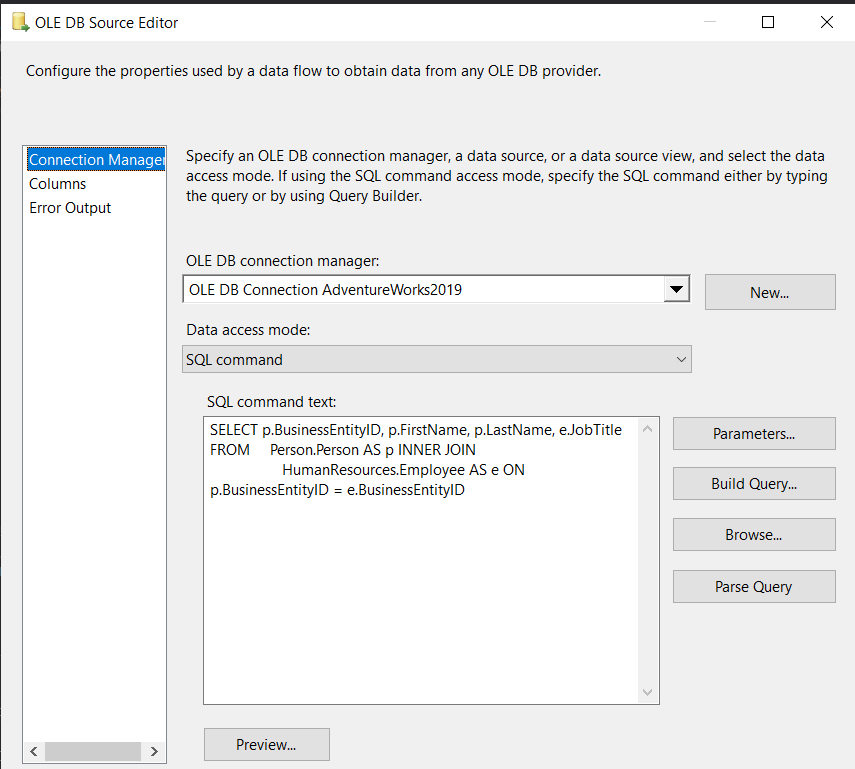


Look at the result in database

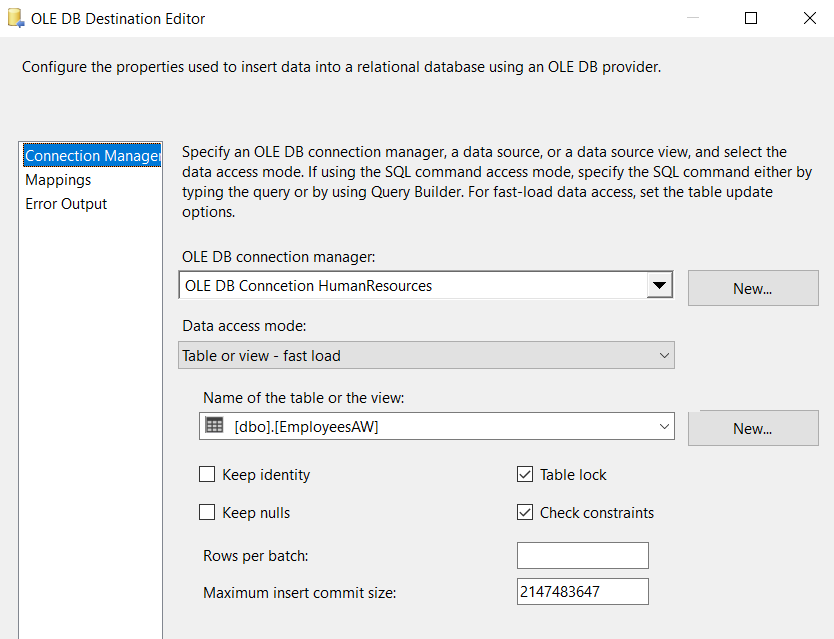


1. **Create part of the ETL pipeline which loads *EmployeesAW* table**

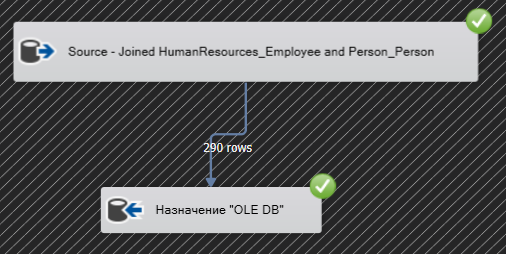
Create Source OLE DB with the SQL query which join *Person.Person* and *HumanResources.Employee* tables from *AdventureWorks2019* database.



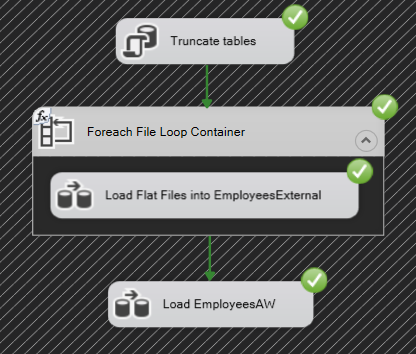
Create Destination OLE DB which loads joined data into *EmployeedAW* table.



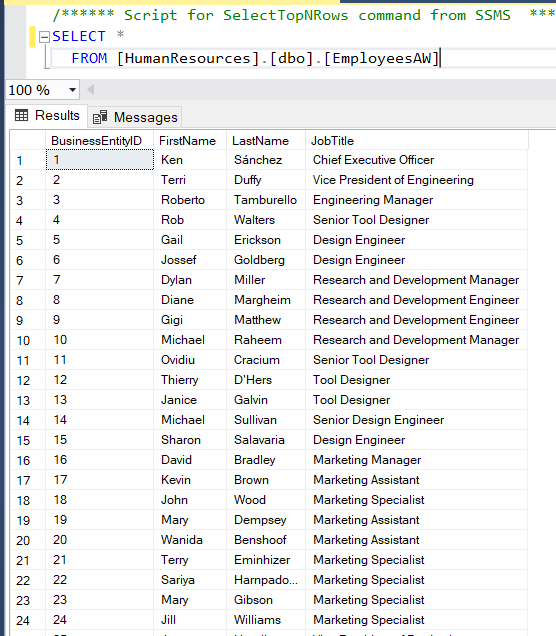
Prepared Data Flow:



Prepared part of ETL pipeline:

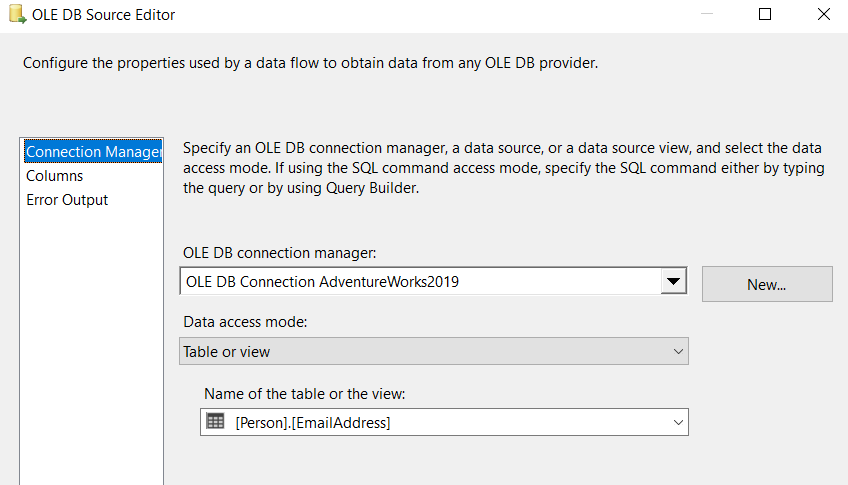


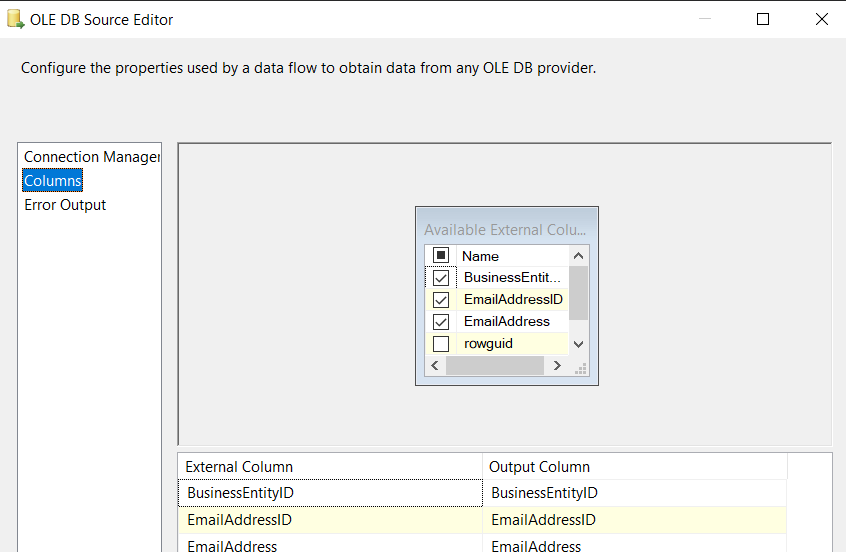
Look at the result in the database table.



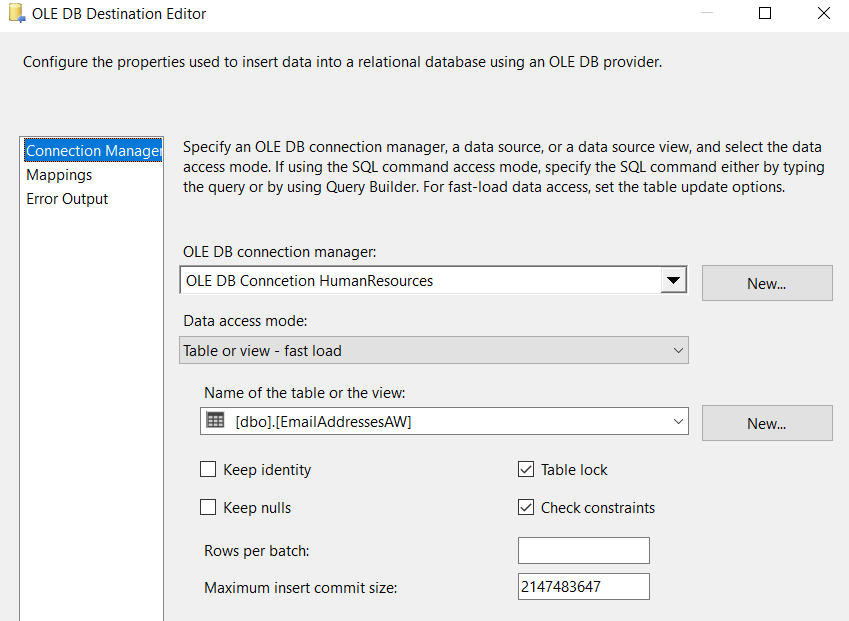
1. **Create part of the ETL pipeline which loads data into *EmailAddressesAW***

Create Source OLE DB which load data from *Person.EmailAddress* table

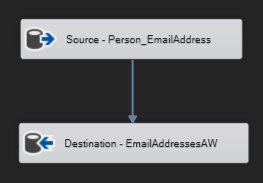




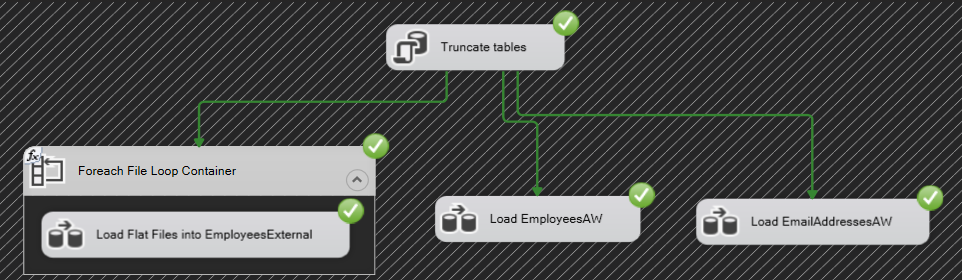
Create Destination OLE DB which loads data into *EmailAddressesAW*

**

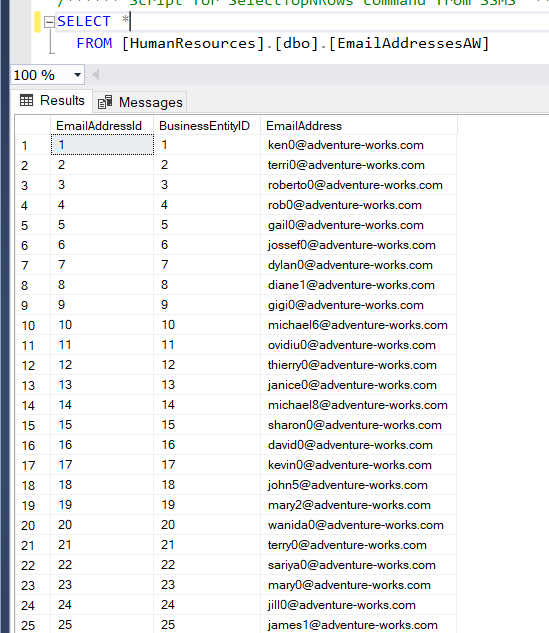
Prepared Data Flow:



Prepared part of ETL pipeline:

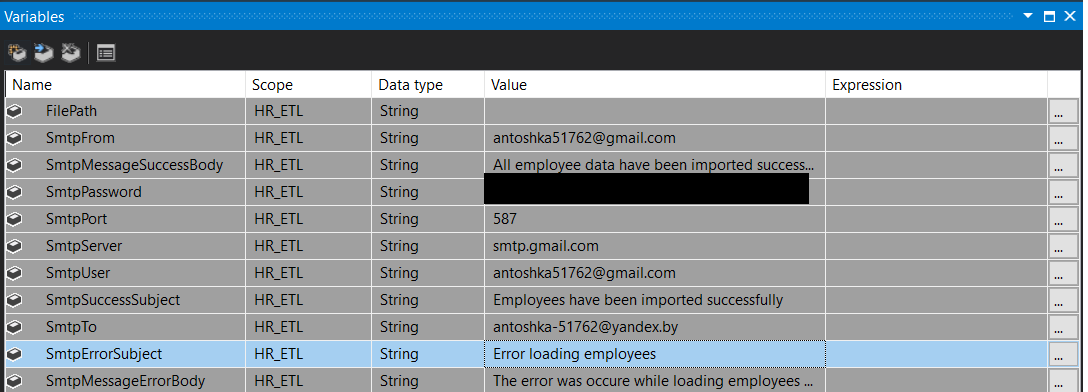


Look at the result in the database table

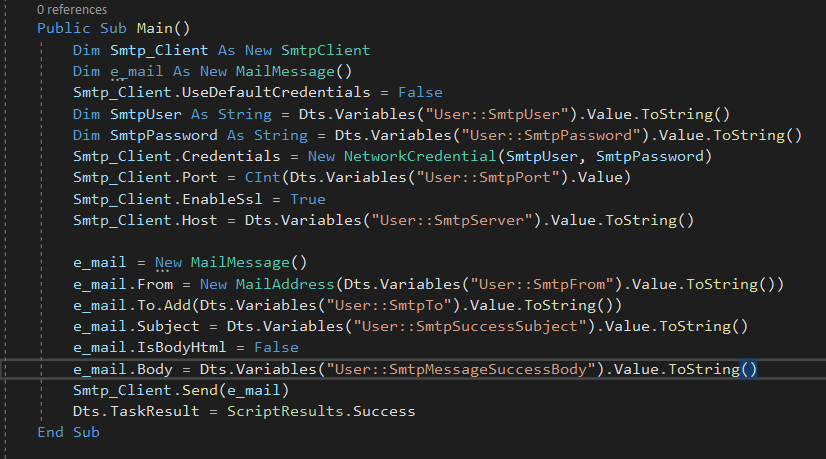


1. **Create error handling logic for ETL pipeline**

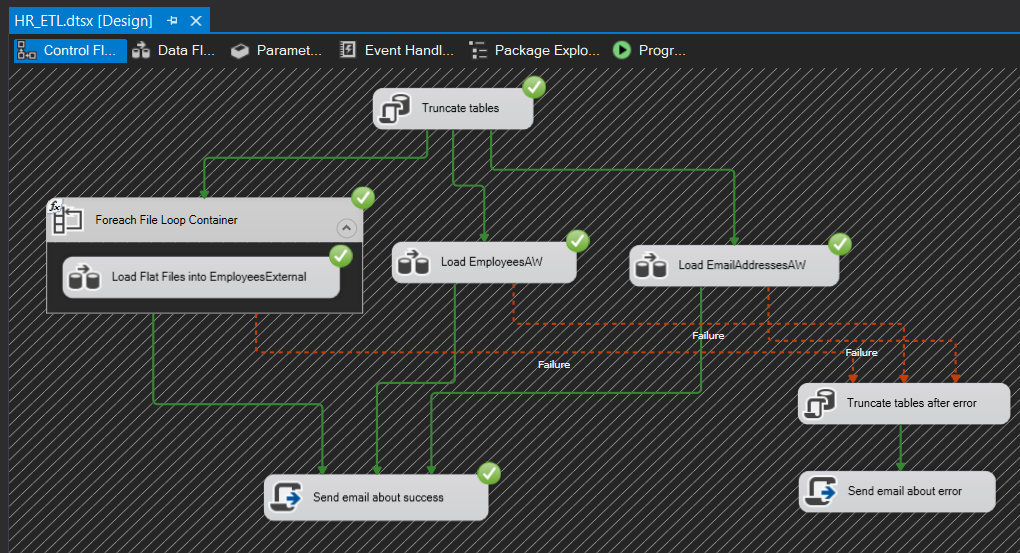
Create all variables which will be used for email sending and error handling



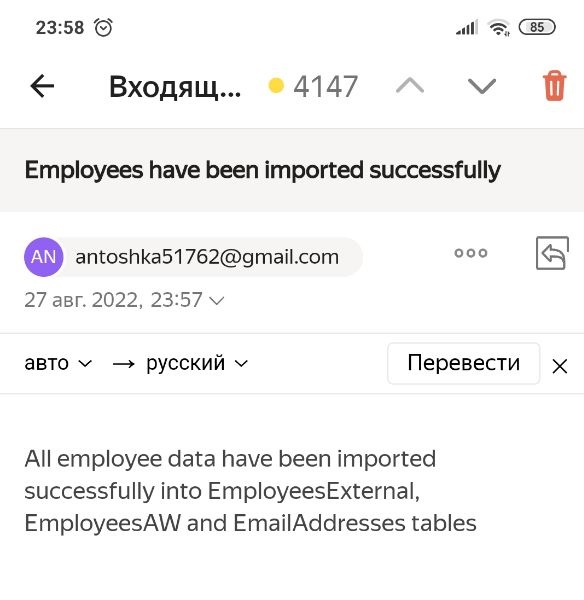
Creating Script Task which send email about successful loading process.



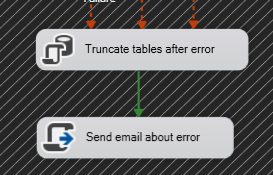
Executing package without errors



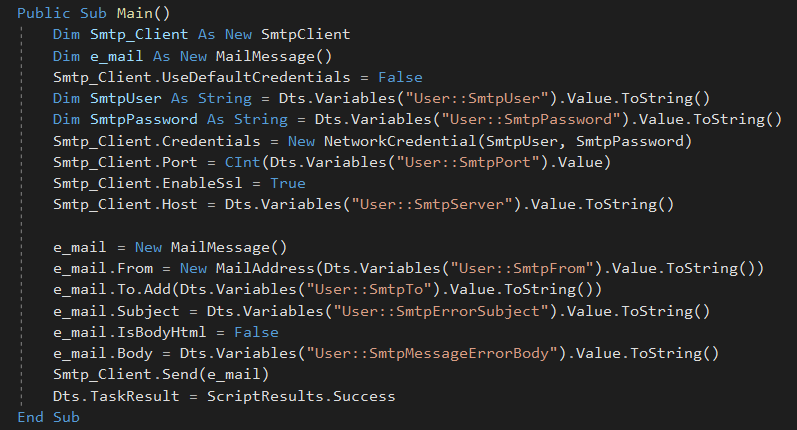
Look at the email



Then add error handling process. First, we should truncate all tables, to avoid partial data loading. Then we should send email about error.

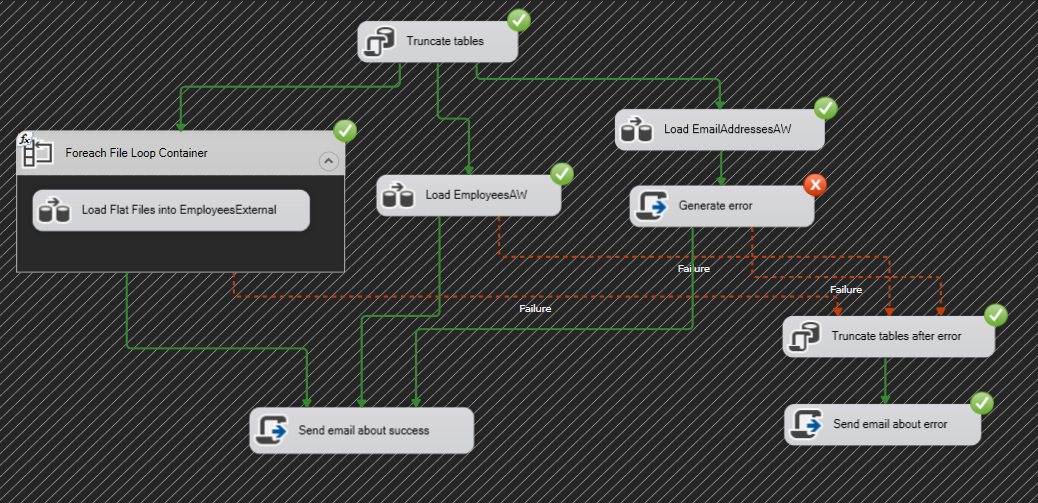


Send email about error script code:



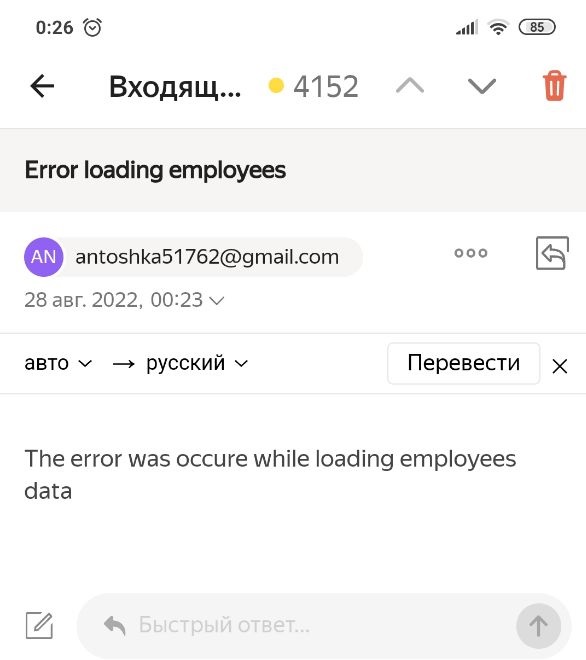
Let’s test error handling process.

For example, create Script Task which generates error and add it into our ETL pipeline

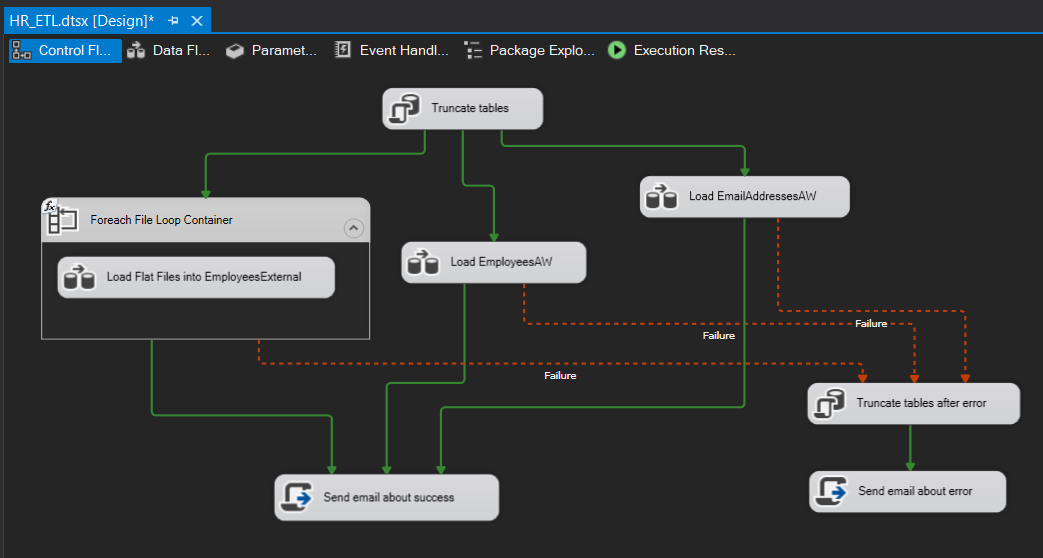


So, the error handling branch will handle the existing error and send mail.

Look at the email



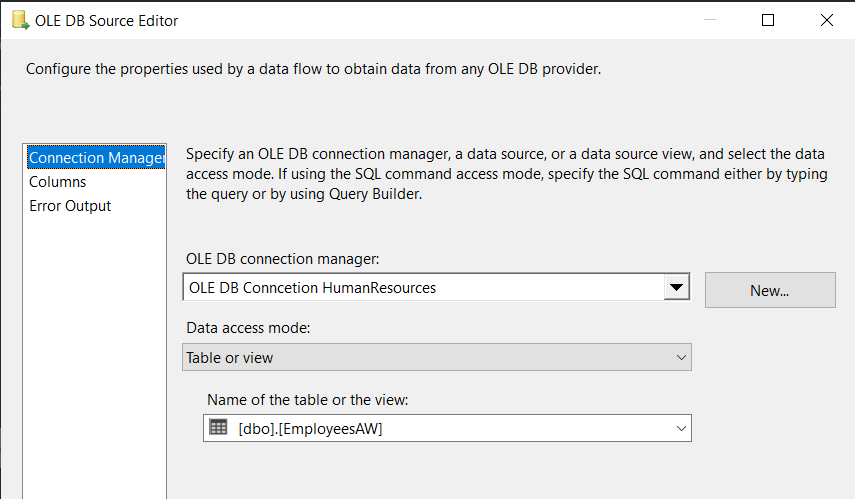
So, our pipeline successfully works and here its final Control Flow:



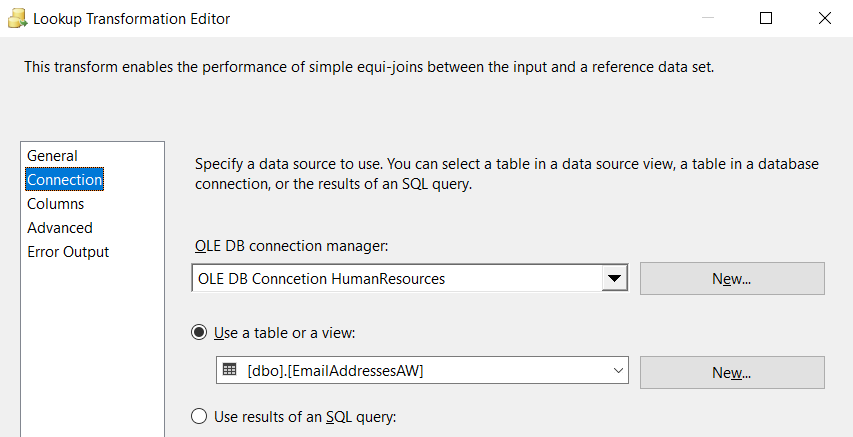
**Task 3**

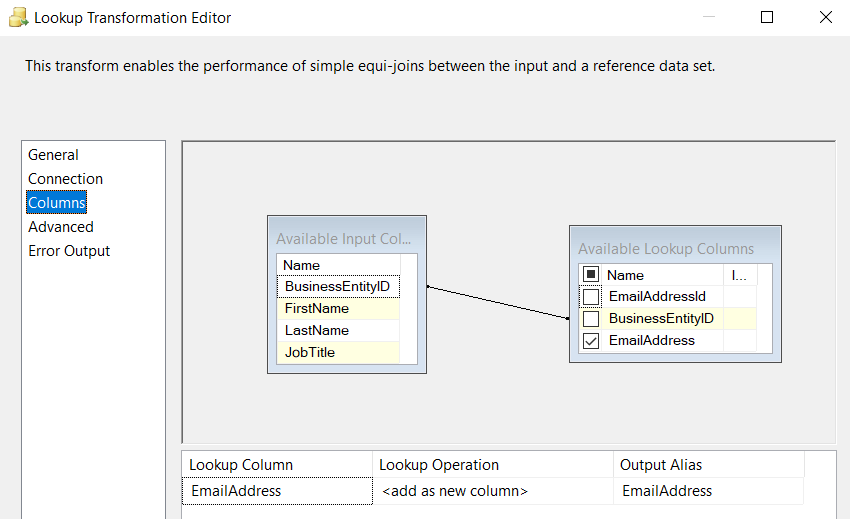
# 3. Exercise 3: Creating ETL solution to export the data from HumanResourcesdb

Create Source OLE DB which loads data from *EmployeesAW* table

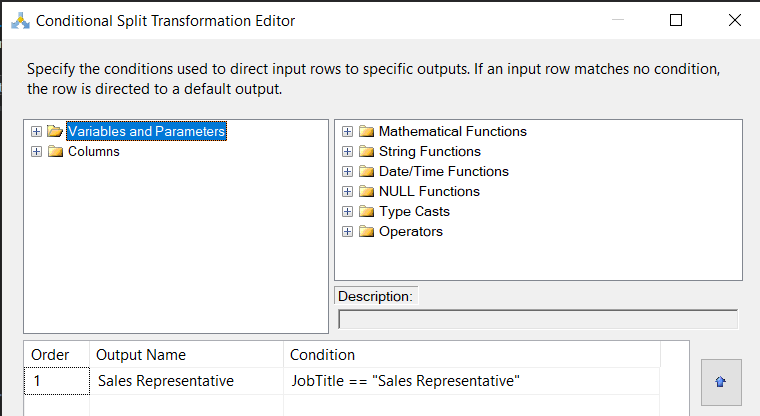


Then create Lookup Transformation which join data from *EmailAddressesAw* table on *BusinessEntityID* column

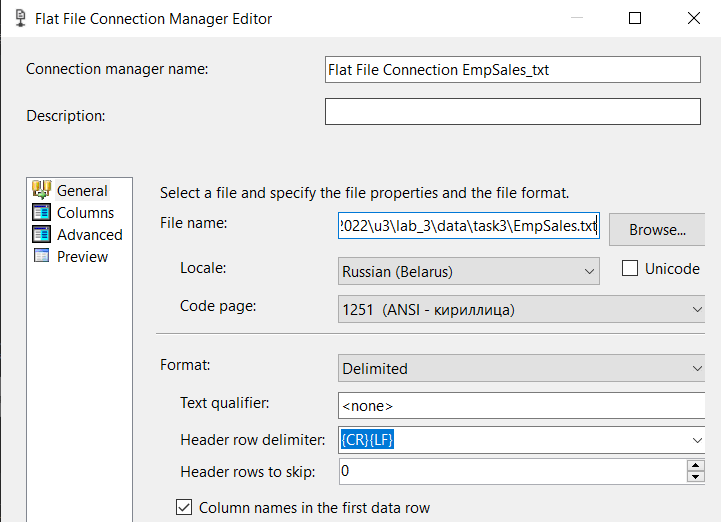


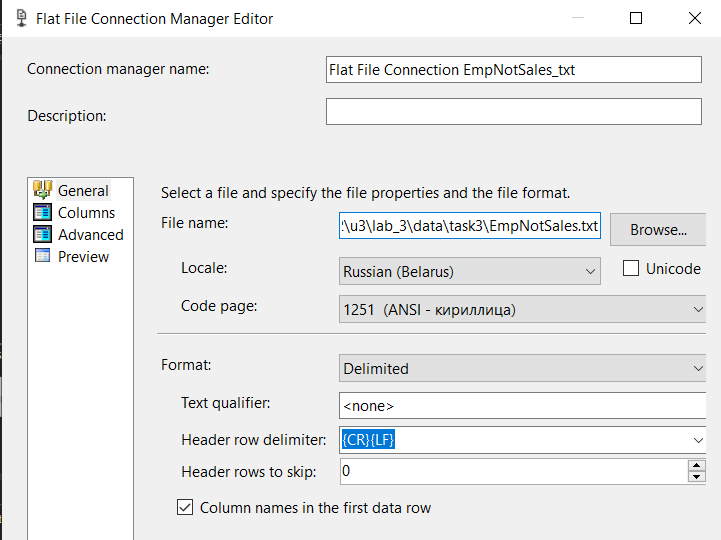


Then we should create Condition Split to split data flow into 2 separate flows. First flow comes into *EmpSales.txt* and the second flow comes into *EmpNotSales.txt* flat file.

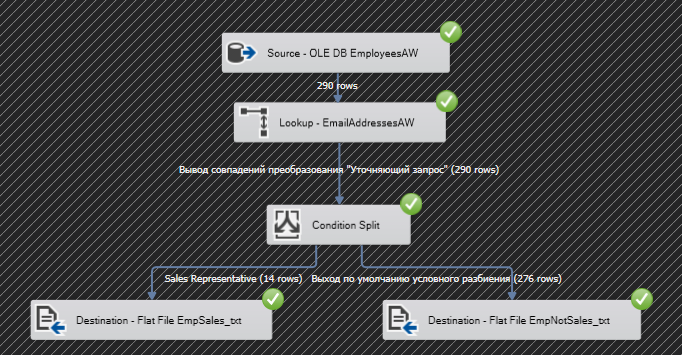


Then just create two Destinations Flat Files to save our divided data

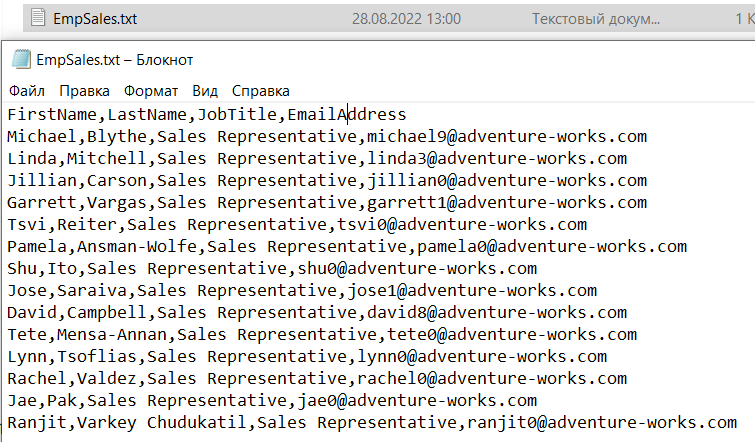


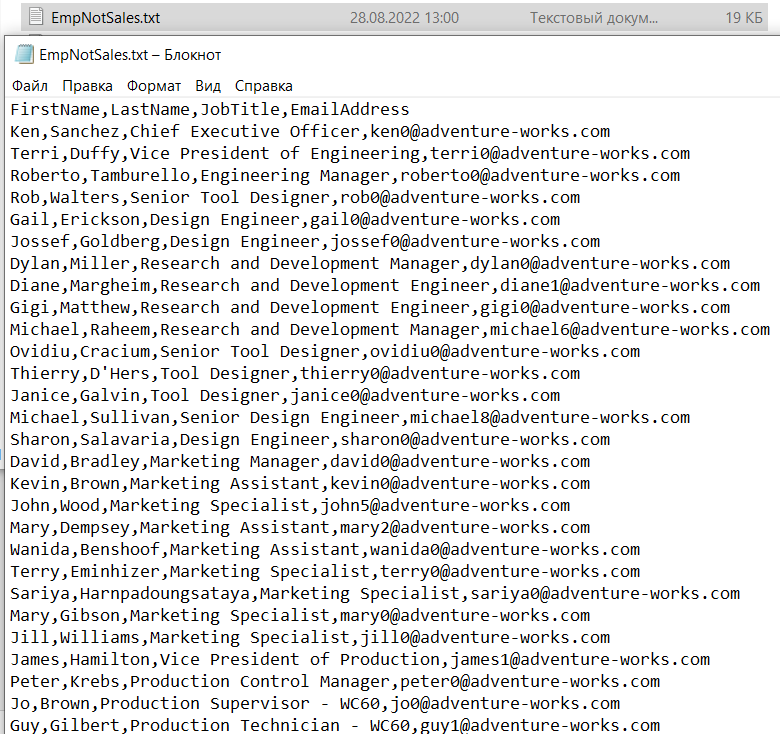


Final Data Flow



Look at the result

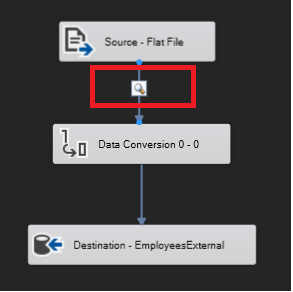




**Task 4**

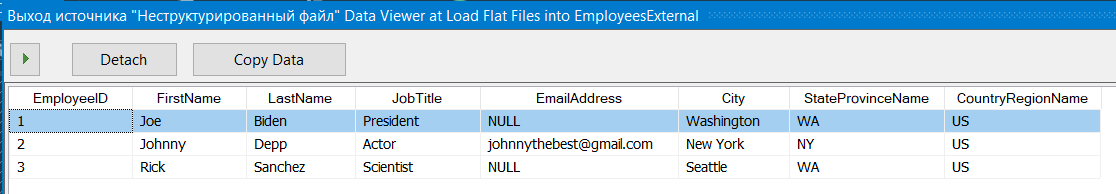
# 4. Exercise 4: Using Data Viewers

For example, enable data viewer at the data flow in the Foreach Loop, to see data from every loaded Flat File.

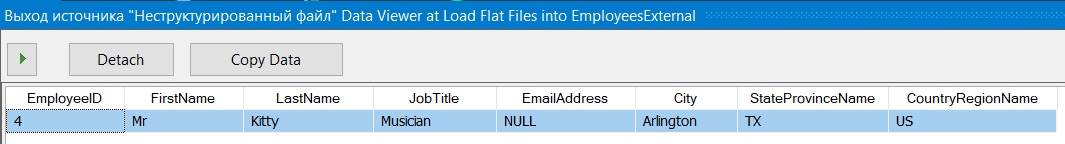


Execute package

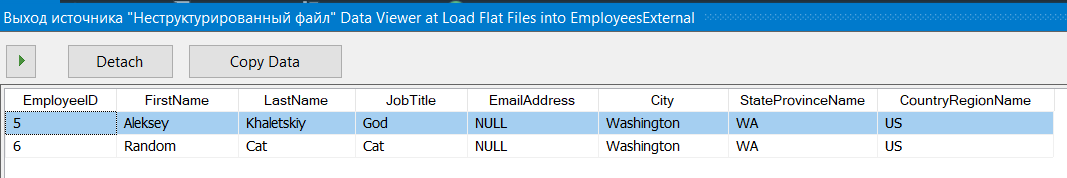
First file:



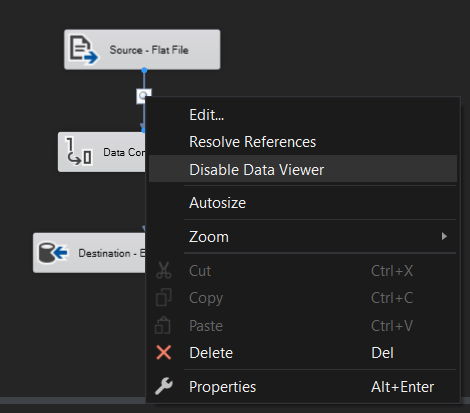
Second file:



Third file:

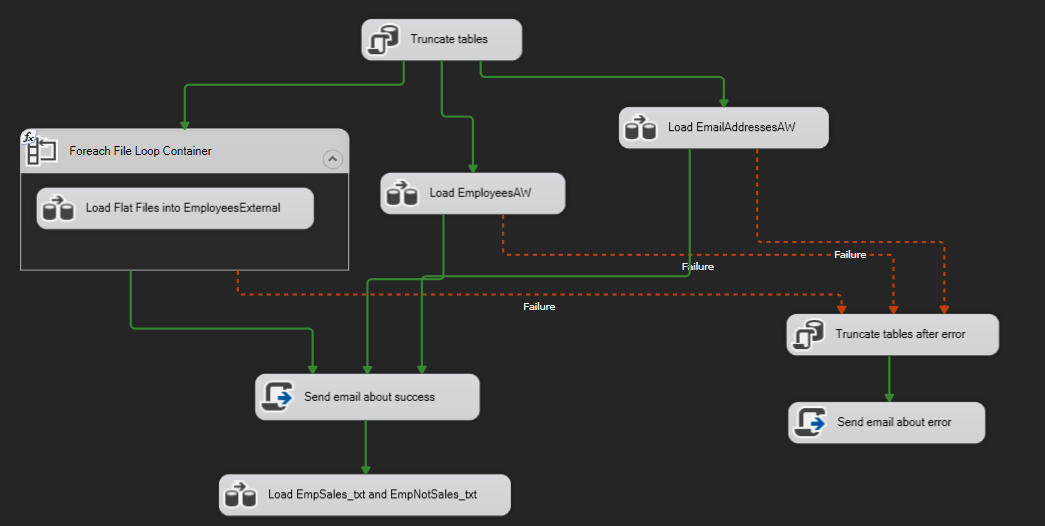


If we want to disable data viewer, we should just right click on Data Flow Path and disable it:



So, data viewers help us with managing and debugging our data flows. It’s really good and helpful option.

**Final ETL Pipeline**

****