

Anton Slizh's

U3M2.LW.Developing Integration Services Solutions

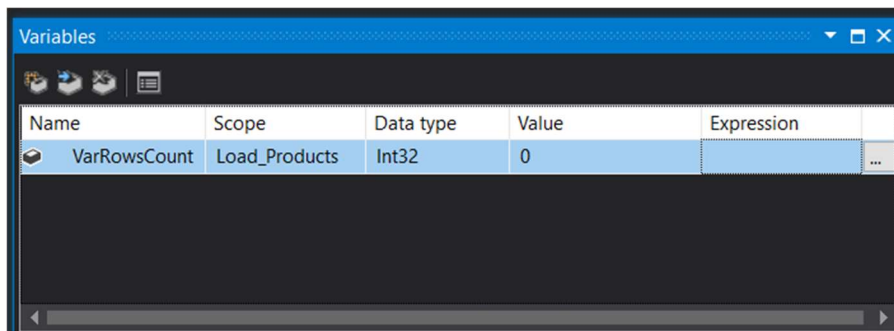
GitHub: <https://github.com/drapejny/DataCamp2022/tree/master>

It's funny that Visual Studio use English and Russian localization at the same time. I even didn't load Russian Language packet. At some screenshots you will see it.

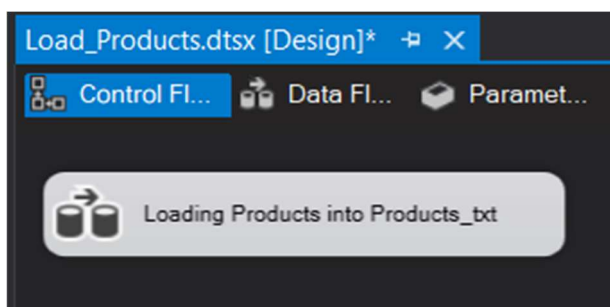
Task 1

1.1. Exercise 1: Creating an Integration Services Project and implementing a package

Creating package scope variable, which will contain the number of exported rows.



Creating the Data Flow Task to export data about products from Database into the Flat File.



Creating Source OLE DB.

Connection Manager

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

Connection

All

Server name:

DESKTOP-T6EFTML

Refresh

Log on to the server

Authentication: Windows Authentication

User name:

Password:

Save my password

Connect to a database

Select or enter a database name:

AdventureWorks2019

Attach a database file:

Browse...

Logical name:

Test Connection

OK

Cancel

Help

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:

DESKTOP-T6EFTMLAdventureWorks2019

New...

Data access mode:

Table or view

Name of the table or the view:

[Production].[Product]

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

Available External Columns

☒ Name

☒ Style

☒ ProductSubcategoryID

☒ ProductModelID

☒ SellStartDate

☒ SellEndDate

☒ DiscontinuedDate

☒ rowguid

☒ ModifiedDate

External Column	Output Column
Style	Style
ProductSubcategoryID	ProductSubcategoryID
ProductModelID	ProductModelID
SellStartDate	SellStartDate
SellEndDate	SellEndDate
DiscontinuedDate	DiscontinuedDate
rowguid	rowguid
ModifiedDate	ModifiedDate

Creating Destination Flat File:

Flat File Connection Manager Editor

Connection manager name: Flat File Connection Manager

Description:

General Columns Advanced Preview

Select a file and specify the file properties and the file format.

File name: D:\DataMola\Labs\DataCamp2022\u3\ Browse...

Locale: Russian (Belarus) ☐ Unicode

Code page: 1251 (ANSI - кириллица)

Format: Delimited

Text qualifier: <none>

Header row delimiter: [CR](LF)

Header rows to skip: 0

☒ Column names in the first data row

Flat File Connection Manager Editor

Connection manager name: Flat File Connection Manager

Description:

General Columns Advanced Preview

Specify the characters that delimit the source file:

Row delimiter: [CR](LF)

Column delimiter: Comma (,)

Preview rows 1-0:

ProductID	Name	ProductNumber	MakeFlag	FinishedGoodsFlag
-----------	------	---------------	----------	-------------------

Flat File Connection Manager Editor

Connection manager name: Flat File Connection Manager

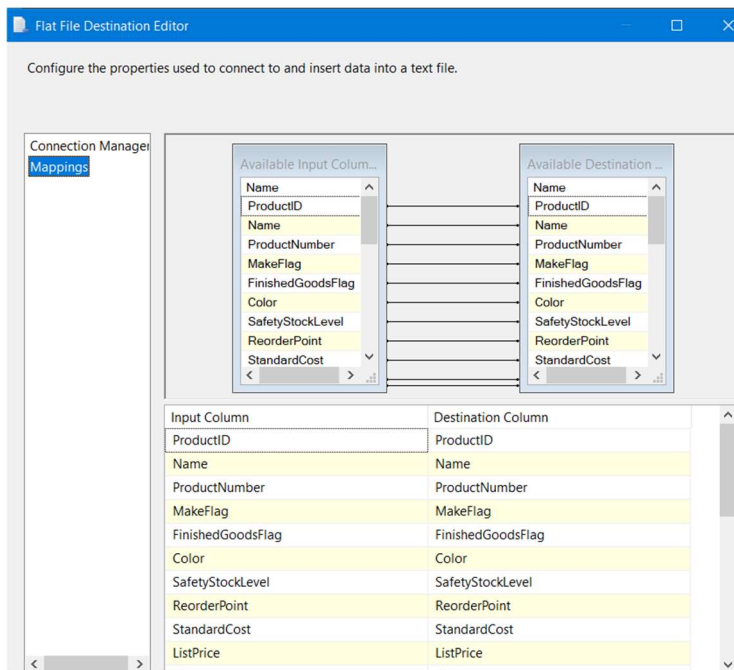
Description:

General Columns Advanced Preview

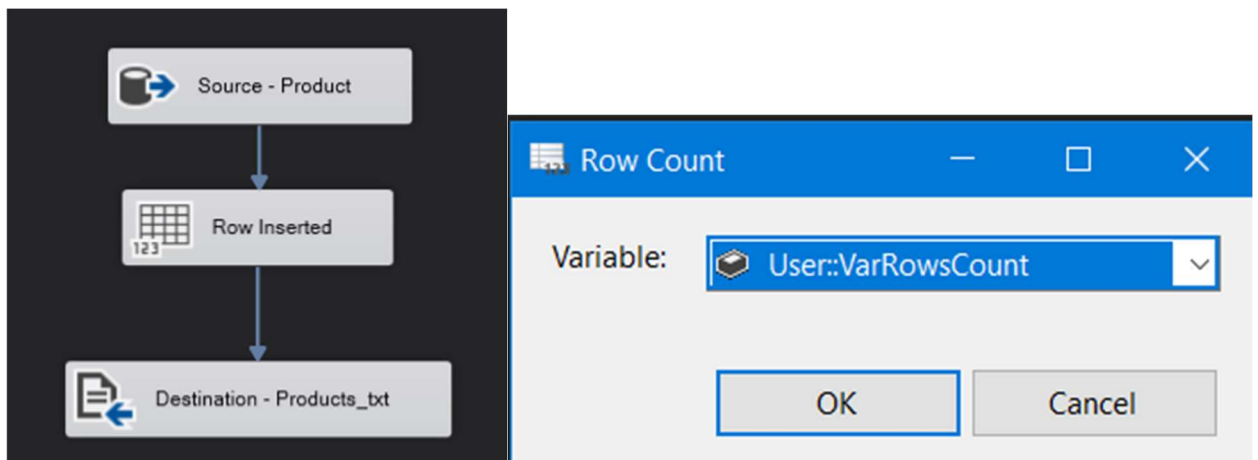
Configure the properties of each column.

ProductID	Misc
Name	ProductID
ProductNumber	ColumnDelimiter Comma (,)
MakeFlag	ColumnType Delimited
FinishedGoodsFlag	InputColumnWidth 0
Color	DataPrecision 0
SafetyStockLevel	DataScale 0
ReorderPoint	DataType четырехбайтное целое со знаком
StandardCost	OutputColumnWidth 0
ListPrice	TextQualified True
Size	
SizeUnitMeasureCode	
WeightUnitMeasureCode	
Weight	
DaysToManufacture	
ProductLine	
Class	
Style	

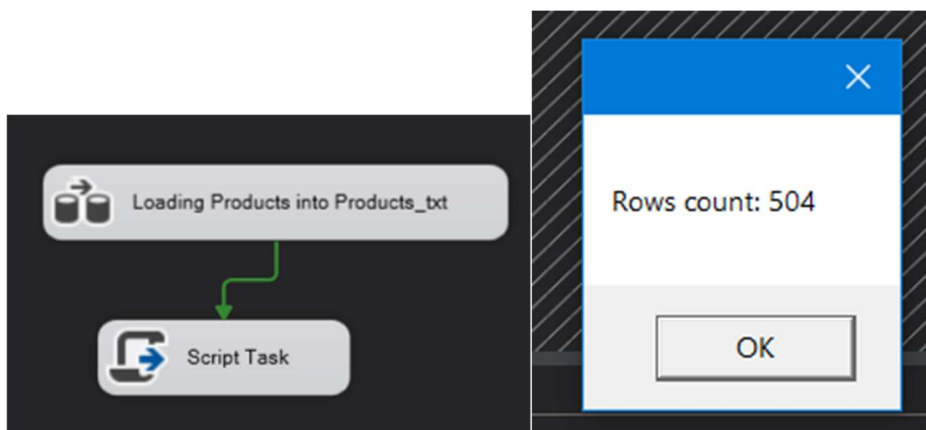
New Delete Suggest Types...



Now let's create additional task between the Source and Destination which will count all inserted rows. The rows number will be written into recently created *VarRowsCount* variable.



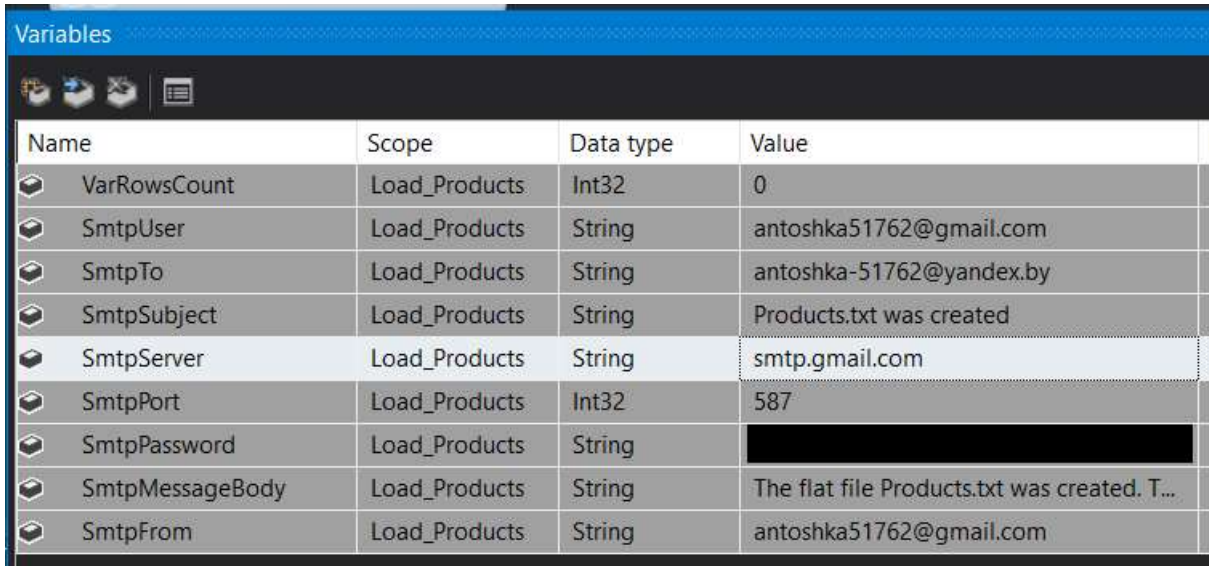
Let's make sure that all working fine. Create Script Task and print the variable value.



Now let's create the Event Handler which will send the email after loading data into *Products.txt* file.

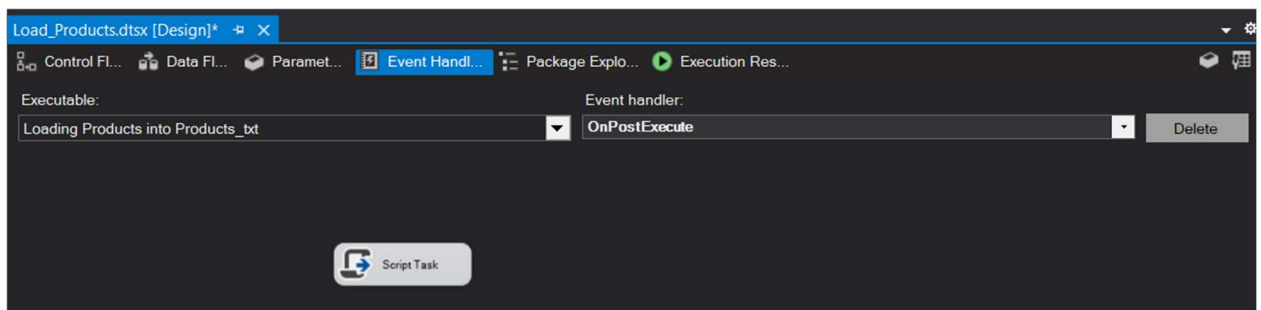
As the Send Mail Task doesn't work with server which needs the credentials, we will write our own script to send messages.

Before writing a script we should create variables which contain our credentials and other email information.

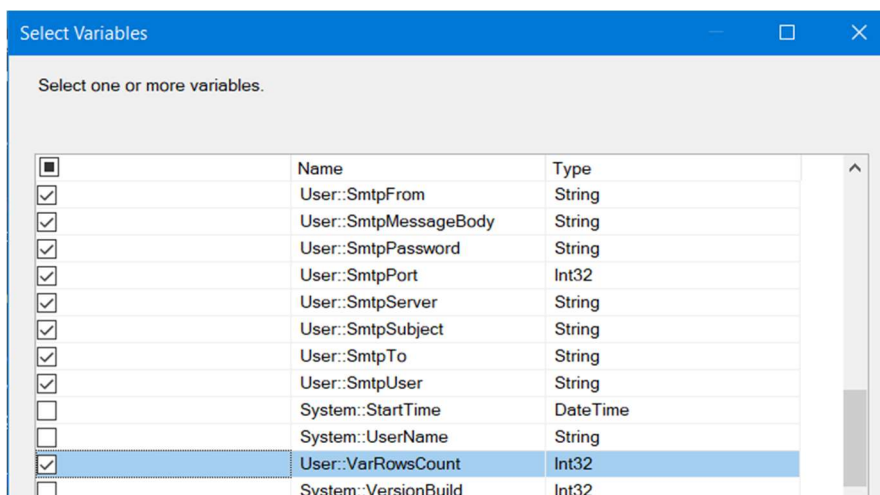


Name	Scope	Data type	Value
VarRowCount	Load_Products	Int32	0
SmtplibUser	Load_Products	String	antoshka51762@gmail.com
SmtplibTo	Load_Products	String	antoshka-51762@yandex.by
SmtplibSubject	Load_Products	String	Products.txt was created
SmtplibServer	Load_Products	String	smtp.gmail.com
SmtplibPort	Load_Products	Int32	587
SmtplibPassword	Load_Products	String	
SmtplibMessageBody	Load_Products	String	The flat file Products.txt was created. T...
SmtplibFrom	Load_Products	String	antoshka51762@gmail.com

Creating the *OnPostExecute* Handler on *Loading Products into Products_txt* executable.



Selecting our variables to use them in the script



Writing the script code

```
0 references
Public Sub Main()
    Dim SmtplibClient As New SmtplibClient
    Dim e_mail As New MailMessage()
    SmtplibClient.UseDefaultCredentials = False
    Dim SmtplibUser As String = Dts.Variables("User::SmtplibUser").Value.ToString()
    Dim SmtplibPassword As String = Dts.Variables("User::SmtplibPassword").Value.ToString()
    SmtplibClient.Credentials = New NetworkCredential(SmtplibUser, SmtplibPassword)
    SmtplibClient.Port = CInt(Dts.Variables("User::SmtplibPort").Value)
    SmtplibClient.EnableSsl = True
    SmtplibClient.Host = Dts.Variables("User::SmtplibServer").Value.ToString()

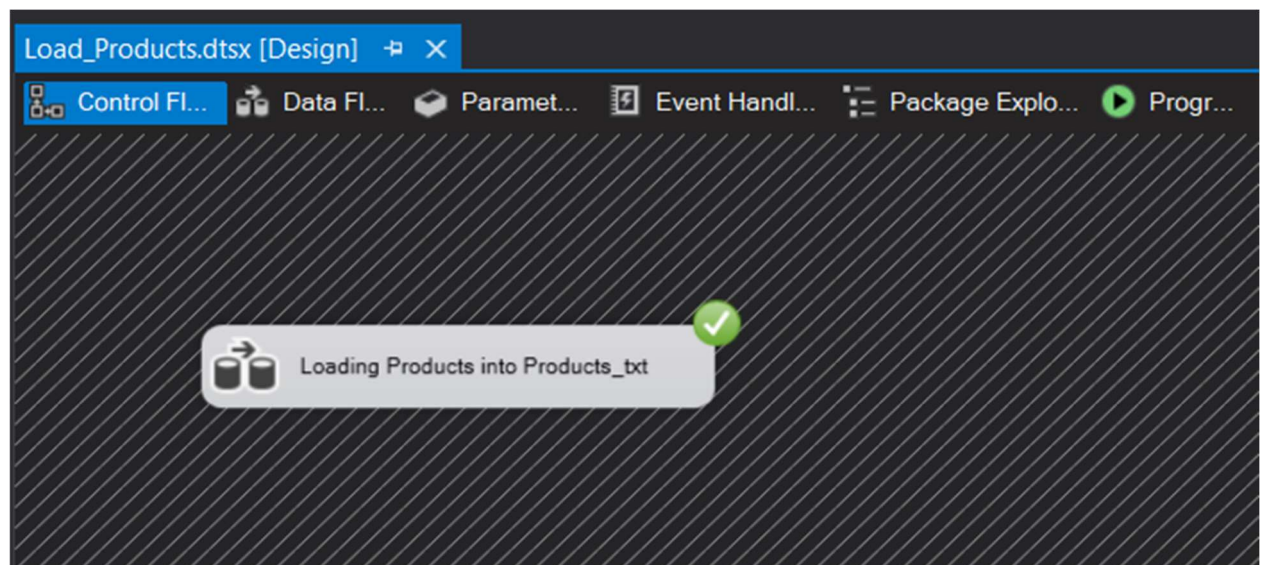
    e_mail = New MailMessage()
    e_mail.From = New MailAddress(Dts.Variables("User::SmtplibFrom").Value.ToString())
    e_mail.To.Add(Dts.Variables("User::SmtplibTo").Value.ToString())
    e_mail.Subject = Dts.Variables("User::SmtplibSubject").Value.ToString()
    e_mail.IsBodyHtml = False
    e_mail.Body = Dts.Variables("User::SmtplibMessageBody").Value.ToString() + Dts.Variables("User::VarRowCount").Value.ToString()
    SmtplibClient.Send(e_mail)
    Dts.TaskResult = ScriptResults.Success
End Sub

ScriptResults declaration

End Class
```

So, the task is completed.

Executing the package:



Looking at the data in the *Products.txt*

Products.txt 25.08.2022 18:24 Текстовый докум... 91 KB

Products.txt - Блокнот

Файл Правка Формат Вид Справка

ProductID,Name,ProductNumber,MakeFlag,FinishedGoodsFlag,Color,SafetyStockLevel,ReorderPoint,StandardCost,ListPrice,Size,SizeUnitMeasureCode

1,Adjustable Race,AR-5381,False,False,,1000,750,0,0,,,,,0,,,,,2008-04-30 00:00:00,,,{694215B7-08F7-4C0D-ACB1-D734BA44C0C8},2014-02-08 10:0
2,Bearing Ball,BA-8327,False,False,,1000,750,0,0,,,,,0,,,,,2008-04-30 00:00:00,,,{58AE3C20-4F3A-4749-A7D4-D568806CC537},2014-02-08 10:01:3
3,BB Ball Bearing,BE-2349,True,False,,800,600,0,0,,,,,1,,,,,2008-04-30 00:00:00,,,{9C21AED2-5BFA-4F18-BCB8-F11638DC2E4E},2014-02-08 10:01:3
4,Headset Ball Bearings,BE-2908,False,False,,800,600,0,0,,,,,0,,,,,2008-04-30 00:00:00,,,{ECFED6CB-51FF-49B5-B06C-7D8AC834DB8B},2014-02-08
316,Blade,BL-2036,True,False,,800,600,0,0,,,,,1,,,,,2008-04-30 00:00:00,,,{E73E9750-603B-4131-89F5-3DD15ED5FF80},2014-02-08 10:01:36.82700
317,LL Crankarm,CA-5965,False,False,Black,500,375,0,0,,,,,0,,L,,,2008-04-30 00:00:00,,,{3C9D10B7-A6B2-4774-9963-C19DCEE72FEA},2014-02-08
318,ML Crankarm,CA-6738,False,False,Black,500,375,0,0,,,,,0,,M,,,2008-04-30 00:00:00,,,{EABB9A92-FA07-4EAB-8955-F0517B4A4CA7},2014-02-08
319,HL Crankarm,CA-7457,False,False,Black,500,375,0,0,,,,,0,,,,,2008-04-30 00:00:00,,,{7D3FD384-4F29-484B-86FA-4206E276FE58},2014-02-08 10:01:3
320,Chainring Bolts,CB-2903,False,False,Silver,1000,750,0,0,,,,,0,,,,,2008-04-30 00:00:00,,,{7BE38E48-B7D6-4486-888E-F53C26735101},2014-02-08
321,Chainring Nut,CN-6137,False,False,Silver,1000,750,0,0,,,,,0,,,,,2008-04-30 00:00:00,,,{3314B1D7-EF69-4431-B6DD-DC75268BD5DF},2014-02-08
322,Chainring,CR-7833,False,False,Black,1000,750,0,0,,,,,0,,,,,2008-04-30 00:00:00,,,{F0AC2C4D-1A1F-4E3C-B4D9-68AEA0EC1CE4},2014-02-08 10:0
323,Crown Race,CR-9981,False,False,,1000,750,0,0,,,,,0,,,,,2008-04-30 00:00:00,,,{51A32CA6-65A1-4C31-AF2B-D9E4F5D631D4},2014-02-08 10:01:3
324,Chain Stays,CS-2812,True,False,,1000,750,0,0,,,,,1,,,,,2008-04-30 00:00:00,,,{FE0678ED-AEF2-4C58-A450-8151CC24DD8},2014-02-08 10:01:3
325,Decal 1,DC-8732,False,False,,1000,750,0,0,,,,,0,,,,,2008-04-30 00:00:00,,,{05CE123C-A402-478E-AE9B-75D7727AEAD},2014-02-08 10:01:36.8
326,Decal 2,DC-9824,False,False,,1000,750,0,0,,,,,0,,,,,2008-04-30 00:00:00,,,{A56851F9-1CD7-4E2F-8779-2E773E1B5209},2014-02-08 10:01:36.82
327,Down Tube,DT-2377,True,False,,800,600,0,0,,,,,1,,,,,2008-04-30 00:00:00,,,{1DAD47DD-E259-42B8-B8B4-15A0B7D21B2F},2014-02-08 10:01:36.8
328,Mountain End Caps,EC-M092,True,False,,1000,750,0,0,,,,,1,,,,,2008-04-30 00:00:00,,,{6070B1EA-59B7-4F8B-950F-2BE07D00449D},2014-02-08 10:01:36.82
329,Road End Caps,EC-R098,True,False,,1000,750,0,0,,,,,1,,,,,2008-04-30 00:00:00,,,{88399D13-719E-4545-81D6-F0650F372FA2},2014-02-08 10:01:36.82
330,Touring End Caps,EC-T209,True,False,,1000,750,0,0,,,,,1,,,,,2008-04-30 00:00:00,,,{6903CE24-D0CE-4191-9198-4231DE37A929},2014-02-08 10:01:36.82
331,Fork End,FE-3760,True,False,,800,600,0,0,,,,,1,,,,,2008-04-30 00:00:00,,,{C91D602E-DA52-43D2-BD7E-EB110A9392B9},2014-02-08 10:01:36.82
332,Freewheel,FH-2981,False,False,Silver,500,375,0,0,,,,,0,,,,,2008-04-30 00:00:00,,,{D864879A-E8B1-4F7B-BAFA-1F136089C2C8},2014-02-08 10:01:36.82
341,Flat Washer 1,FW-1000,False,False,,1000,750,0,0,,,,,0,,,,,2008-04-30 00:00:00,,,{A3F2FA3A-22E1-43D8-A131-A9B89C32D8EA},2014-02-08 10:01:36.82

Стр 1, столб 1 100% Windows (CRLF) UTF-8

Looking at the E-Mail

<

Products.txt was created

AN antoshka51762@gmail.com antoshka51762@gmail.com Сегодня в 18:24

Я >

Язык письма — английский. Перевести на русский? Перевести

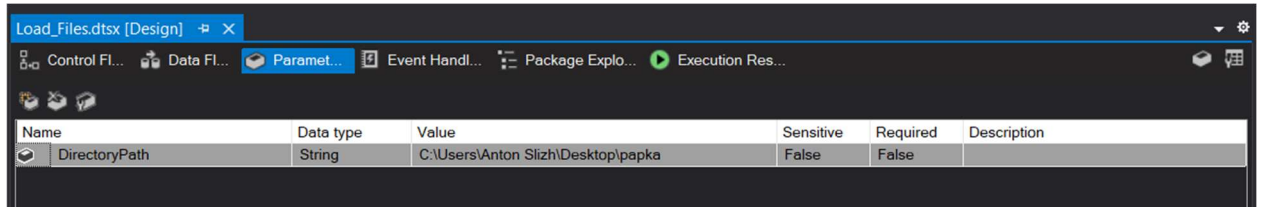
The flat file Products.txt was created. The number of rows: 504

Ответить

Task 2

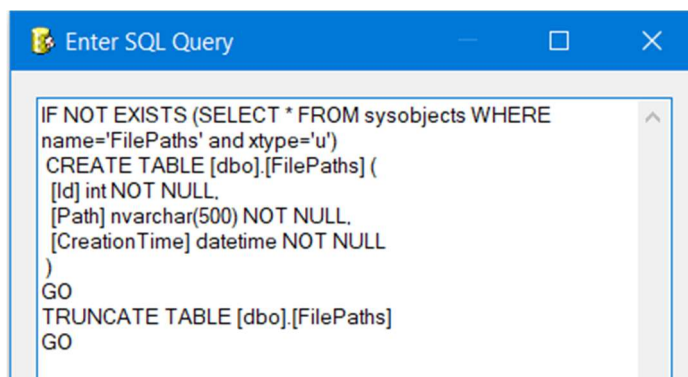
1.2. Exercise 2: Creating a package to keep list of files from a directory

Creating the package parameter which represents the directory for file search.

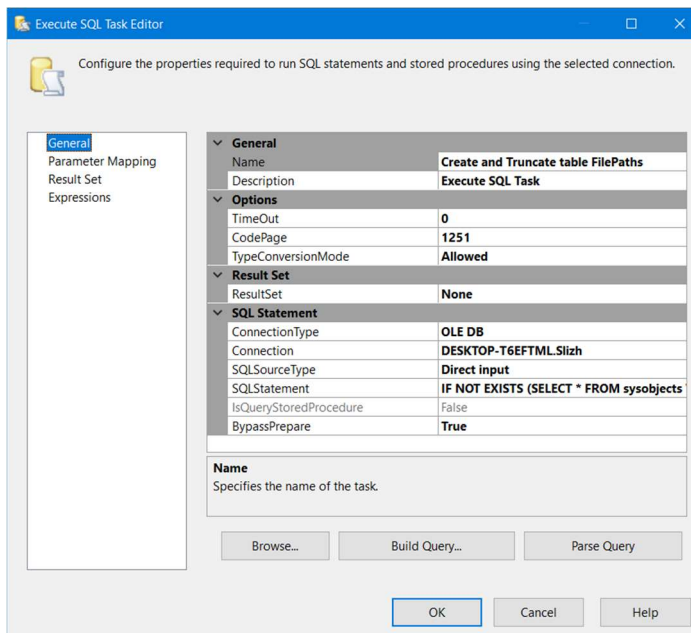


Name	Data type	Value	Sensitive	Required	Description
DirectoryPath	String	C:\Users\Anton Slizh\Desktop\pafka	False	False	

Then preparing the table in the database. Creating SQL Task which will create destination table (if not exist) and truncate it.



```
IF NOT EXISTS (SELECT * FROM sysobjects WHERE
name='FilePaths' and xtype='u')
CREATE TABLE [dbo].[FilePaths] (
[Id] int NOT NULL,
[Path] nvarchar(500) NOT NULL,
[CreationTime] datetime NOT NULL
)
GO
TRUNCATE TABLE [dbo].[FilePaths]
GO
```



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

General

Name: Create and Truncate table FilePaths
Description: Execute SQL Task

Options

TimeOut: 0
CodePage: 1251
TypeConversionMode: Allowed

Result Set

ResultSet: None

SQL Statement

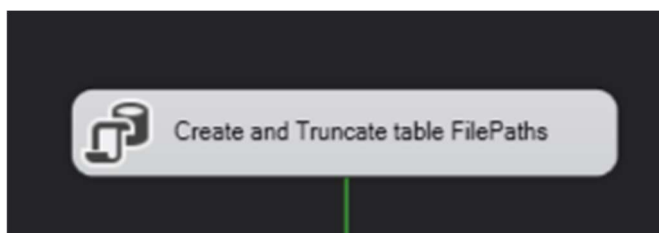
ConnectionType: OLE DB
Connection: DESKTOP-T6EFTML.Slzh
SQLSourceType: Direct input
SQLStatement: IF NOT EXISTS (SELECT * FROM sysobjects WHERE name='FilePaths' and xtype='u') CREATE TABLE [dbo].[FilePaths] (...)
IsQueryStoredProcedure: False
BypassPrepare: True

Name

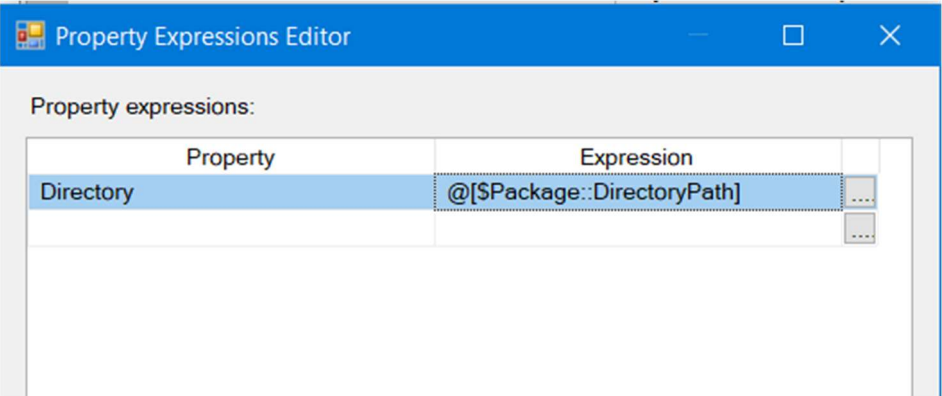
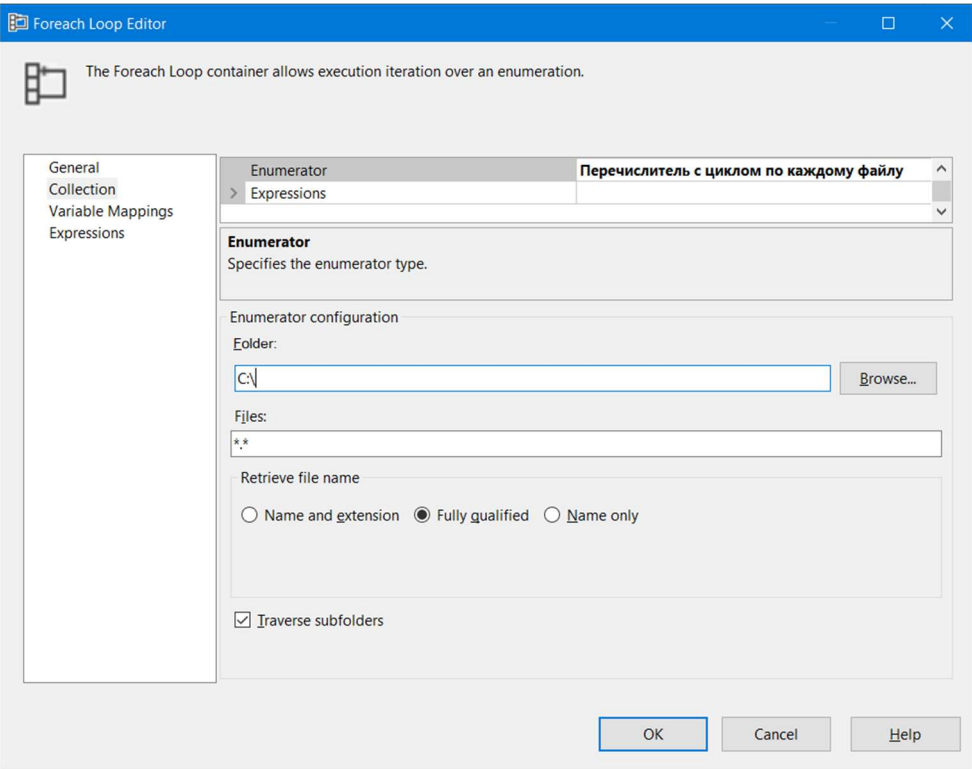
Specifies the name of the task.

Browse... Build Query... Parse Query

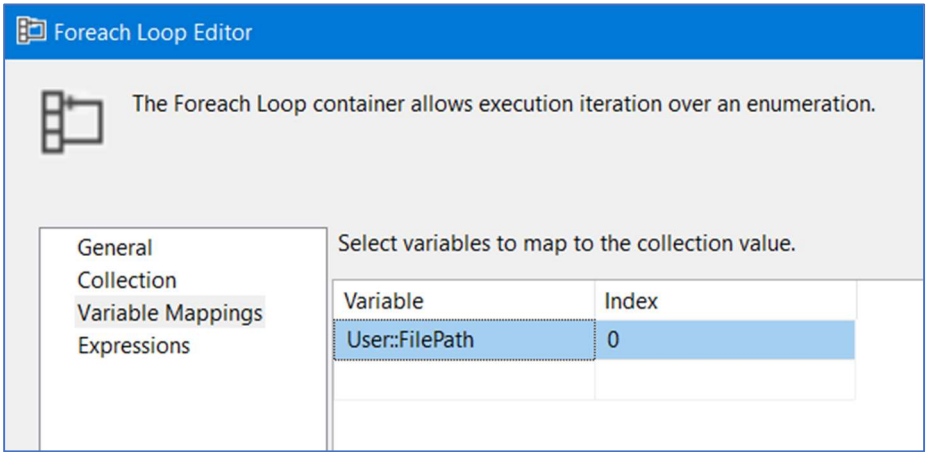
OK Cancel Help



Then creating a Foreach Loop which will iterate through all files in the specified directory. The directory for file search is taken from package parameters.



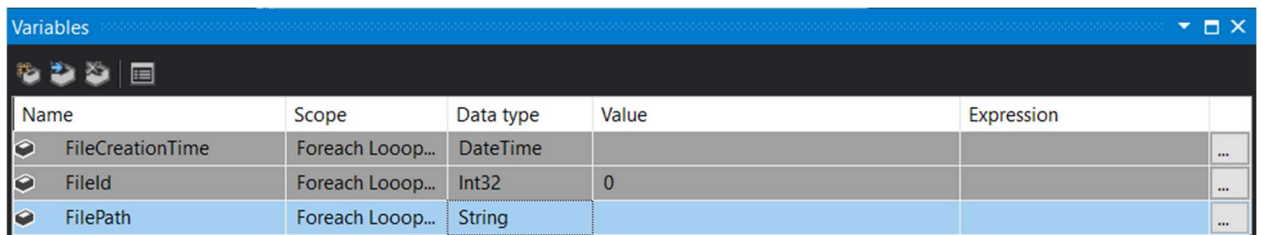
Creating the variable *FilePath* which will store the path for each file while iterating through the loop.



Then defining other variables at the Foreach Loop scope.

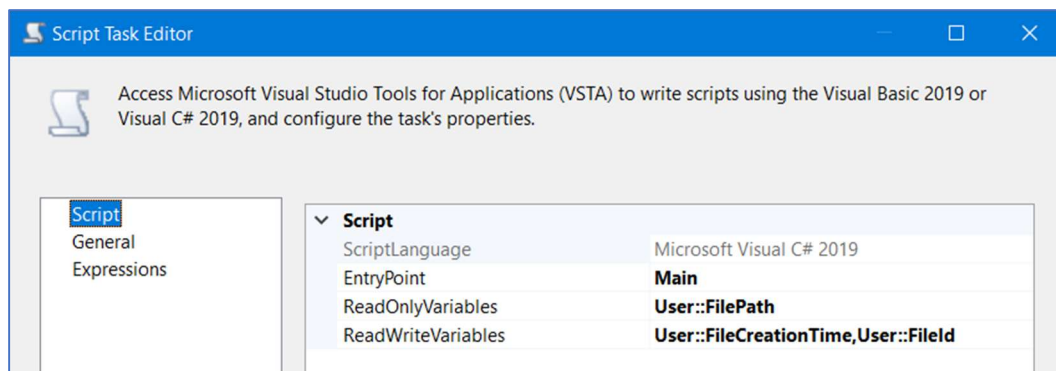
FileId – store calculated unique id for each file in the loop

FileCreationTime – store creation time for each file in the loop



Name	Scope	Data type	Value	Expression
FileCreationTime	Foreach Loop...	DateTime		
FileId	Foreach Loop...	Int32	0	
FilePath	Foreach Loop...	String		

Now, we should write information about file into defined variables. Let's create the special Script Task which will update variables values.



```
0 references
public void Main()
{
    String filePath = Dts.Variables["FilePath"].Value.ToString();

    DateTime fileCreationTime = File.GetCreationTime(filePath);

    Dts.Variables["FileCreationTime"].Value = fileCreationTime;

    Int32 fileId = Convert.ToInt32(Dts.Variables["FileId"].Value);
    fileId++;
    Dts.Variables["FileId"].Value = fileId;

    Dts.TaskResult = (int)ScriptResults.Success;
}
```

After script execution *FilePath* variable will still contain the path of the file. *FileCreationTime* variable will contain extracted creation time of the file and *FileId* will contain the calculated Int32 Id for the file.

When all necessary data was written into the variables, we can insert data into the prepared table.

Creating the SQL Task which will insert one row into the table *FilePaths*.

Also we should specify parameters for INSERT statement.

Execute SQL Task Editor

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

General

Name: Insert into FilePaths
Description: Execute SQL Task

Options

TimeOut: 0
CodePage: 1251
TypeConversionMode: Allowed

Result Set

ResultSet: None

SQL Statement

ConnectionType: OLE DB
Connection: DESKTOP-T6EFTML.Slzh
SQLSourceType: Direct input
SQLStatement: INSERT INTO [dbo].[FilePaths] VALUES(?, ?, ?)
IsQueryStoredProcedure: False
BypassPrepare: True

Name: Specifies the name of the task.

Browse... Build Query... Parse Query

OK Cancel Help

Enter SQL Query

INSERT INTO [dbo].[FilePaths]
VALUES(?, ?, ?)
GO

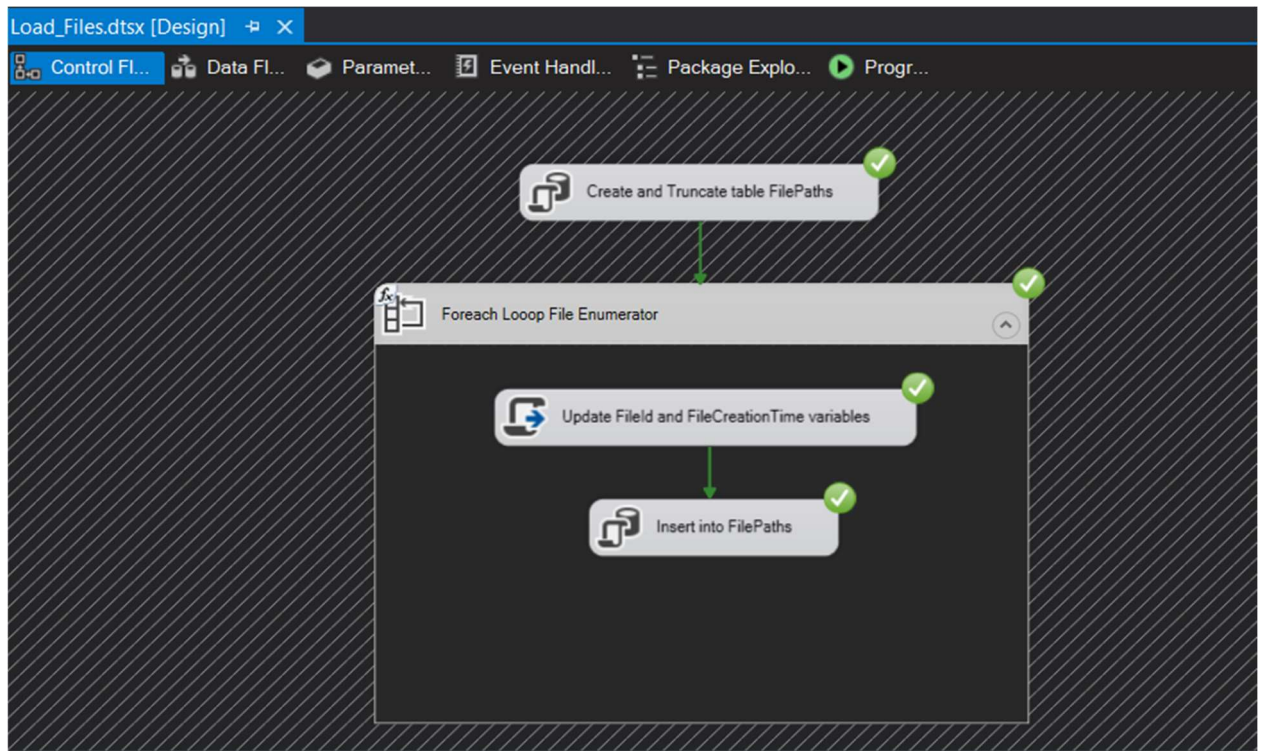
Execute SQL Task Editor

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

General
Parameter Mapping
Result Set
Expressions

Variable Name	Direction	Data Type	Paramet...	Paramet...
User::FileId	Input	LONG	0	-1
User::FilePath	Input	VARCHAR	1	-1
User::FileCreationTime	Input	DATE	2	-1

So, our ETL process is done. Let's execute it.



Now, let's look at the result.

Select result data from the table *FilePaths* in the database:

SQLQuery3.sql - DE...L\Anton Slizh (58) SQLQuery2.sql - DE...L\Anton Slizh (56))*

```
/****** Script for SelectTopNRows command from SSMS *****/  
SELECT TOP (1000) [Id]  
      , [Path]  
      , [CreationTime]  
FROM [Slizh].[dbo].[FilePaths]
```

100 %

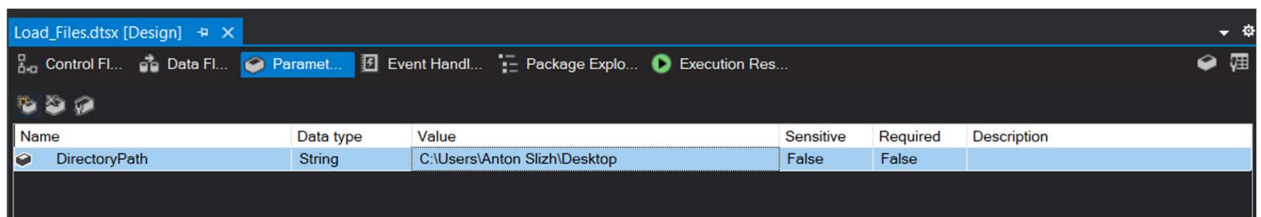
Results Messages

	Id	Path	CreationTime
1	1	C:\Users\Anton Slizh\Desktop\papka\1.txt	2022-08-25 22:07:13.000
2	2	C:\Users\Anton Slizh\Desktop\papka\subpapka\2.txt	2022-08-25 22:07:21.000

Make sure that specified directory (including subdirectories) contains only 2 files.

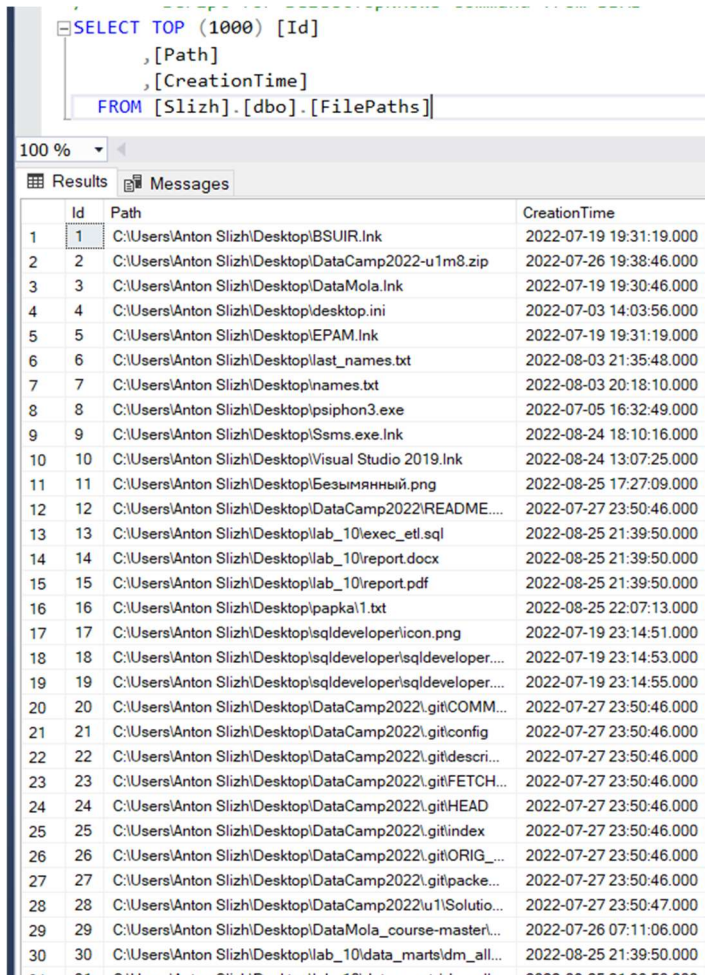
```
C:\>tree "C:\Users\Anton Slizh\Desktop\papka" /f | more  
Структура папок  
Серийный номер тома: 000000F8 F09E:5C71  
C:\USERS\ANTON SLIZH\DESKTOP\PAPKA  
1.txt  
subpapka  
2.txt
```


And let's test the directory which contains more than 2 file. For example my Desktop directory.



The screenshot shows the 'Load_Files.dtsx [Design]' window in SQL Server Data Tools. It displays a table named 'DirectoryPath' with the following columns: Name, Data type, Value, Sensitive, Required, and Description. The 'Value' column contains the path 'C:\Users\Anton Slizh\Desktop'.

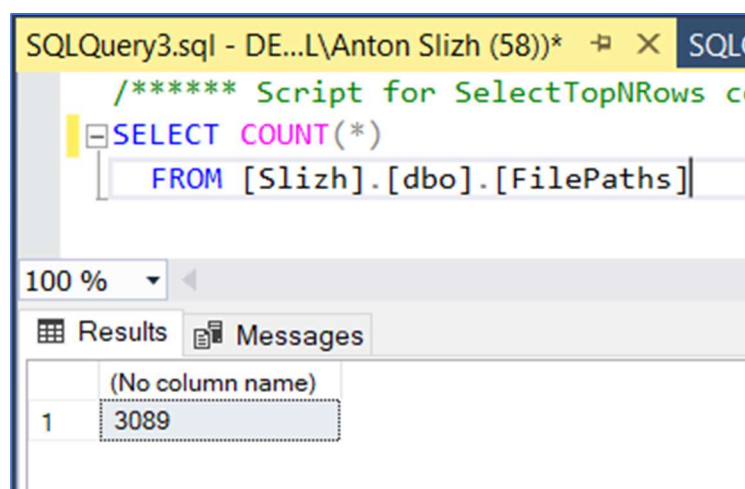
Name	Data type	Value	Sensitive	Required	Description
DirectoryPath	String	C:\Users\Anton Slizh\Desktop	False	False	



The screenshot shows a SQL query in the 'SQLQuery3.sql' window. The query is a 'SELECT TOP (1000)' statement. Below the query, the 'Results' pane shows the output of the query, which is a table with three columns: Id, Path, and CreationTime. The results are sorted by Id in ascending order.

```
SELECT TOP (1000) [Id]
, [Path]
, [CreationTime]
FROM [Slizh].[dbo].[FilePaths]
```

Id	Path	CreationTime
1	C:\Users\Anton Slizh\Desktop\BSUIR.Ink	2022-07-19 19:31:19.000
2	C:\Users\Anton Slizh\Desktop\DataCamp2022-u1m8.zip	2022-07-26 19:38:46.000
3	C:\Users\Anton Slizh\Desktop\DataMola.Ink	2022-07-19 19:30:46.000
4	C:\Users\Anton Slizh\Desktop\desktop.ini	2022-07-03 14:03:56.000
5	C:\Users\Anton Slizh\Desktop\EPAM.Ink	2022-07-19 19:31:19.000
6	C:\Users\Anton Slizh\Desktop\last_names.txt	2022-08-03 21:35:48.000
7	C:\Users\Anton Slizh\Desktop\names.txt	2022-08-03 20:18:10.000
8	C:\Users\Anton Slizh\Desktop\psiphon3.exe	2022-07-05 16:32:49.000
9	C:\Users\Anton Slizh\Desktop\Ssms.exe.Ink	2022-08-24 18:10:16.000
10	C:\Users\Anton Slizh\Desktop\Visual Studio 2019.Ink	2022-08-24 13:07:25.000
11	C:\Users\Anton Slizh\Desktop\Безымянный.png	2022-08-25 17:27:09.000
12	C:\Users\Anton Slizh\Desktop\DataCamp2022\README...	2022-07-27 23:50:46.000
13	C:\Users\Anton Slizh\Desktop\lab_10\exec_etl.sql	2022-08-25 21:39:50.000
14	C:\Users\Anton Slizh\Desktop\lab_10\report.docx	2022-08-25 21:39:50.000
15	C:\Users\Anton Slizh\Desktop\lab_10\report.pdf	2022-08-25 21:39:50.000
16	C:\Users\Anton Slizh\Desktop\papakal1.txt	2022-08-25 22:07:13.000
17	C:\Users\Anton Slizh\Desktop\sqldeveloper\icon.png	2022-07-19 23:14:51.000
18	C:\Users\Anton Slizh\Desktop\sqldeveloper\sqldeveloper....	2022-07-19 23:14:53.000
19	C:\Users\Anton Slizh\Desktop\sqldeveloper\sqldeveloper....	2022-07-19 23:14:55.000
20	C:\Users\Anton Slizh\Desktop\DataCamp2022\git\COMM...	2022-07-27 23:50:46.000
21	C:\Users\Anton Slizh\Desktop\DataCamp2022\git\config	2022-07-27 23:50:46.000
22	C:\Users\Anton Slizh\Desktop\DataCamp2022\git\descri...	2022-07-27 23:50:46.000
23	C:\Users\Anton Slizh\Desktop\DataCamp2022\git\FETCH...	2022-07-27 23:50:46.000
24	C:\Users\Anton Slizh\Desktop\DataCamp2022\git\HEAD	2022-07-27 23:50:46.000
25	C:\Users\Anton Slizh\Desktop\DataCamp2022\git\index	2022-07-27 23:50:46.000
26	C:\Users\Anton Slizh\Desktop\DataCamp2022\git\ORIG...	2022-07-27 23:50:46.000
27	C:\Users\Anton Slizh\Desktop\DataCamp2022\git\packe...	2022-07-27 23:50:46.000
28	C:\Users\Anton Slizh\Desktop\DataCamp2022\u1\Solutio...	2022-07-27 23:50:47.000
29	C:\Users\Anton Slizh\Desktop\DataMola_course-master...	2022-07-26 07:11:06.000
30	C:\Users\Anton Slizh\Desktop\lab_10\data_marts\dm_all...	2022-08-25 21:39:50.000



The screenshot shows a SQL query in the 'SQLQuery3.sql' window. The query is a 'SELECT COUNT(*)' statement. Below the query, the 'Results' pane shows the output of the query, which is a single row with the value 3089.

```
SELECT COUNT(*)
FROM [Slizh].[dbo].[FilePaths]
```

(No column name)
3089

All works fine.