

Integracja systemów

Laboratorium 7
Prowadzący: Marek Kowal
(M.Kowal@issi.uz.zgora.pl)

Rolling Up Data with the Aggregate Transform

1. Cel ćwiczenia

In this exercise, you create an extract file that contains a rolled-up version of the Production.TransactionHistory table. The Production.TransactionHistory table has hundreds of thousands of records in it containing very granular data of every transaction for your company's history. Your partner only needs to know how many of each product you've sold, and other data such as the last sale of that product. After this lesson, you'll know how to apply grouping to your data to see your data at a higher grain.

2. Przebieg ćwiczenia

- 1) Create a new SSIS package called Lesson22.dtsx.
- 2) Create a new OLE DB Connection Manager that connects to your AdventureWorks2012 database.
- 3) Drag a Data Flow Task onto the design pane and call the new task DFT - Aggregate Data.
- 4) In the Data Flow tab, drag a new OLE DB Source onto the Data Flow design pane and name it OLE SRC – TransactionHistory.
- 5) Double-click the OLE DB Source and change the OLE DB Connection Manager option to your only connection manager. Change the Data access mode to SQL Command and type the following query into the SQL Command text box:

```
SELECT  
TransactionID, ProductID, TransactionDate,  
TransactionType, Quantity, ActualCost, ModifiedDate  
FROM Production.TransactionHistory
```

- 6) Drag an Aggregate Transform onto the design pane and connect it to the OLE DB Source. Rename the transform AG – Roll up data.
- 7) Open the Aggregate Transformation Editor by double-clicking the new transform and check the ProductID, TransactionDate, Quantity, and ActualCost columns.
- 8) Change the Output Alias column for each of the checked columns. Change the alias to LastTransactionDate for the TransactionDate column. Change the Quantity column to TotalQuantity, and ActualCost to TotalCost
- 9) In the Operation column, change ProductID to Group by, LastTransactionDate to Maximum, TotalQuantity to Sum, and TotalCost to Sum, as shown in Fig. 1.

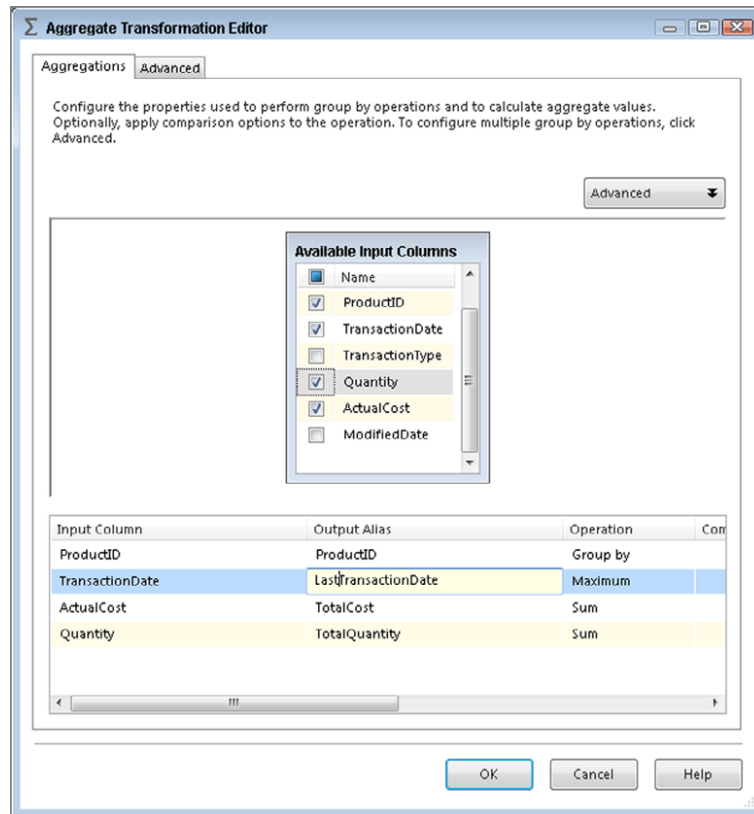


Fig. 1

- 10) Back in the designer, drag a Flat File Destination onto the design pane and connect it to the Aggregate Transform. Rename the connection FF DST – Create Extract.
- 11) Open the Flat File Destination and click New to create a new Flat File Connection Manager. When prompted, select Delimited (separated by a given symbol).
- 12) Name the connection manager Extract. Place the file wherever you'd like and check the Column names in first data row option.
- 13) Go to the Mappings page and click OK.
- 14) Execute the package

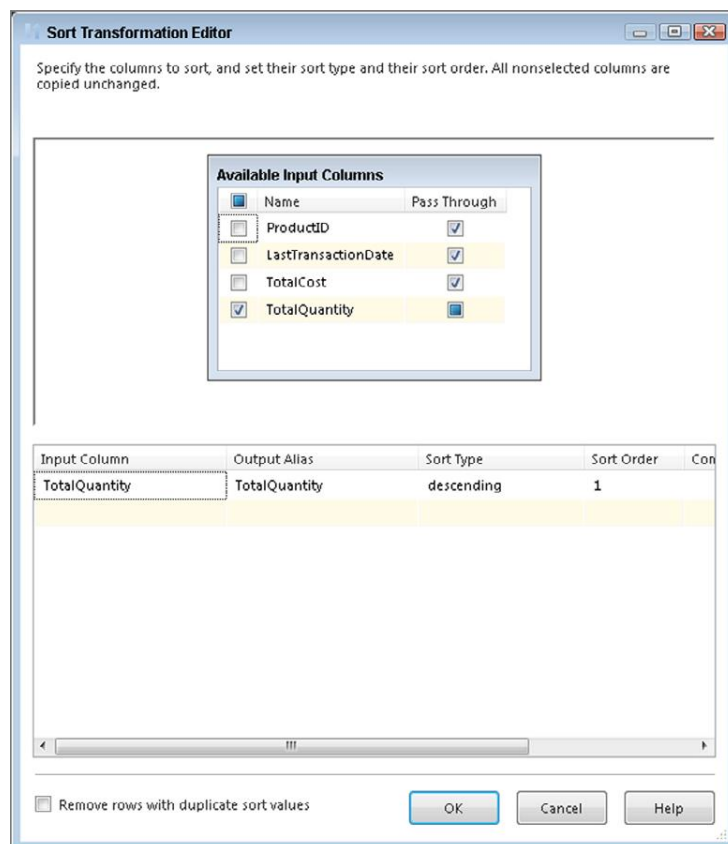
Ordering Data with the Sort Transform

1. Cel ćwiczenia

In this exercise, your company has decided it really needs the extract file you created in Lesson 22 to show the products in order by total sold. Your manager tells you to make sure that, once you've made these changes to your package, you delete the content of the extract file before you run the package again. After this lesson, you'll know how to sort data using SSIS.

2. Przebieg ćwiczenia

- 1) Open the SSIS package Lesson22.dtsx that you created in the previous lesson.
- 2) Click the Data Flow tab and delete the precedence constraint between the Aggregate Transform and the Flat File Destination.
- 3) Drag a Sort Transform into the Data Flow and connect it between the Aggregate Transform and the Flat File Destination
- 4) Open the Sort Transformation Editor, select TotalQuantity to sort by, and change the Sort Type to descending, as shown in Fig. 1. Then click OK.
- 5) Now execute the package. The flat file has now been repopulated sorted by TotalQuantity in descending order.



Rys. 1

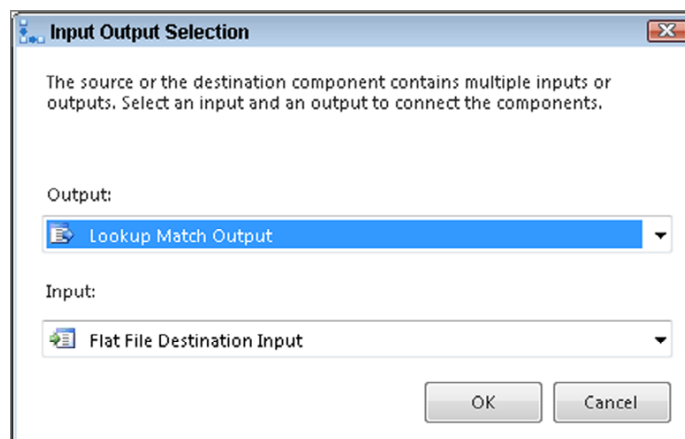
Joining Data with the Lookup Transform

1. Cel ćwiczenia

In this exercise, your company needs you to alter a package to show the product names with the sales of each product. Your manager tells you to create a new flat file to store the results. After this lesson, you'll know how to join data into the Data Flow pipeline using SSIS.

2. Przebieg ćwiczenia

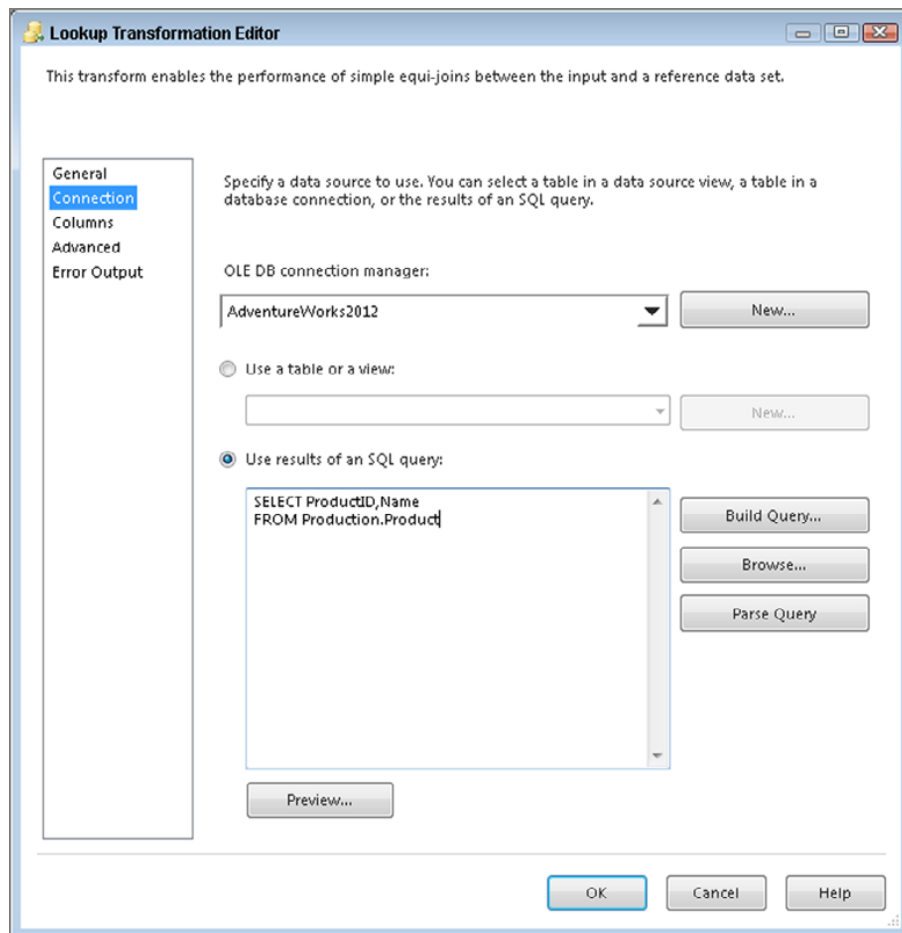
- 1) Continue the work you did in the previous lesson.
- 2) Click the Data Flow tab and delete the connecting lines between the Sort Transform and the Flat File Destination.
- 3) Drag a Lookup Component into the Data Flow and rename it LKP - Product Name; then connect it between the Sort Transform and the Flat File Destination.
- 4) Once you connect to the Flat File Destination, the Input Output Selection dialog box opens, and you should select Lookup Match Output from the Output drop-down box, as shown in Fig. 1



Rys. 1

- 5) Open the Lookup Transformation Editor, navigate to the Connection tab, and select the Use results of an SQL query option
- 6) In the query window, write the following select statement. Figure 2 shows how the editor should look at this point:

```
SELECT ProductID, Name  
FROM Production.Product
```



Rys. 2

- 7) Open the Flat File Destination and click Update, which is a new feature in SSIS 2012 that enables you to easily update flat file metadata. Click OK when the Flat File Connection Manager opens to confirm the update and return back to the destination. The Name column has been added to the file connection, but still needs to be mapped to the Data Flow input. You do this by drawing the mapping from the input columns to the destination columns on the Mappings page.
- 8) Now execute the package. The new flat file has now been created with the new column included.

Auditing Data with the Row Count Transform

1. Cel ćwiczenia

In this exercise, your company needs you to create a package that runs only if the ErrorLog table in the AdventureWorks2012 database contains any rows. After this lesson, you'll know how to insert a row count into a variable and use it dynamically in your package.

2. Przebieg ćwiczenia

1. Create an SSIS package named Lesson25.
2. In the Control Flow tab, add a variable named MyRowCount. Ensure that the variable is package-scoped and of type Int32 (Fig. 1). If you don't know how to add a variable, select Variable from the SSIS menu and click the Add Variable button.

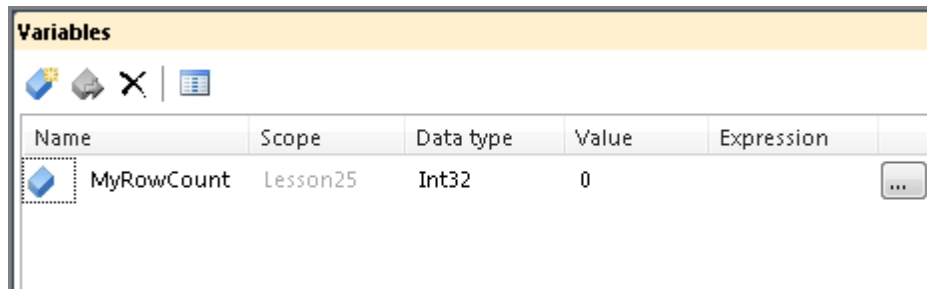


Fig. 1

3. Create a connection manager that connects to the AdventureWorks2012 database. Add an OLE DB Data Source to the Data Flow design surface. Configure the source to point to your AdventureWorks2012 database's connection manager and the ErrorLog table.
4. Add a Row Count Transform to the Data Flow and connect it to the Data Source. Double-click the transform to open the Row Count Editor and select the variable named User::MyRowCount in the Variable property. Your editor should resemble Figure 2.

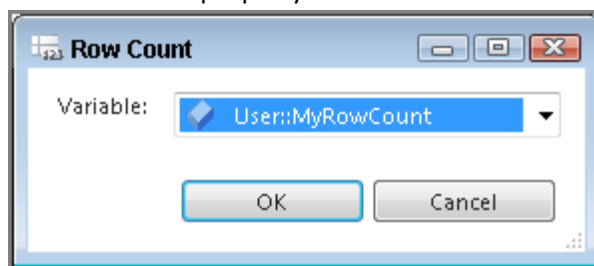


Fig. 2

5. Return to the Control Flow tab and add a Script Task. This task is not really going to perform any action. Instead, it will be used to show the conditional ability to perform steps based on the value returned by the Row Count Transform.
6. Connect the Data Flow Task to the Script Task.
7. Right-click the arrow connecting the Data Flow Task and Script Task. Select the Edit menu. In the Precedence Constraint Editor, change the Evaluation Operation to Expression. Set the Expression to `@MyRowCount>0` (Fig. 3).
8. Now execute the package. The Script Task should not change to green because no rows exist in the ErrorLog table.

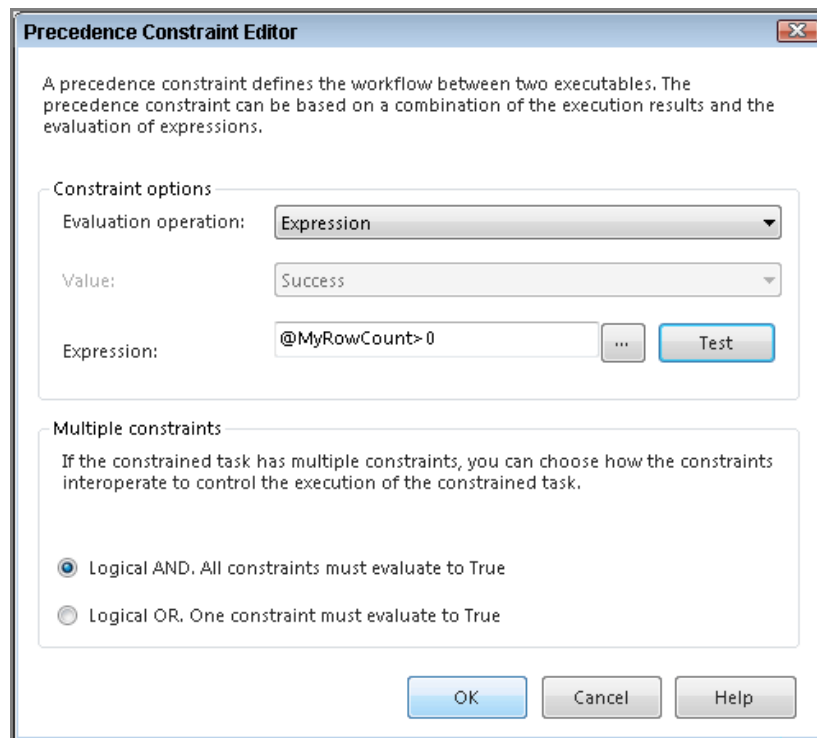


Fig. 3

Combining Multiple Inputs with the Union All Transform

1. Cel ćwiczenia

In this exercise, your company needs you to create a package that has three different sources, but places the data into one flat file. After this lesson, you will know how to combine data from different sources and place that data in one Flat File Destination.

2. Przebieg ćwiczenia

1. Create an SSIS package named Lesson26.
2. In the Control Flow tab, add a new Data Flow Task to the design surface and name it DFT - Union All Sales.
3. Create a new OLE DB Connection Manager using the AdventureWorksDW2012 database as the source. Then drag two OLE DB Sources on the designer and rename them Reseller Sales and Internet Sales.
4. In the Internet Sales Source, select SQL Command as the Data access mode and enter the following query:


```
Select ProductKey,SalesAmount
From FactInternetSales
```

5. In the Reseller Sales Source, select **SQL Command** as the Data access mode and enter the following query:

```
Select ProductKey, SalesAmount  
From FactResellerSales
```
6. Drag a **Union All Transform** and connect both blue arrows from the sources to it. Verify that the columns mapped correctly by opening the **Union All Transformation Editor** (Fig. 1).

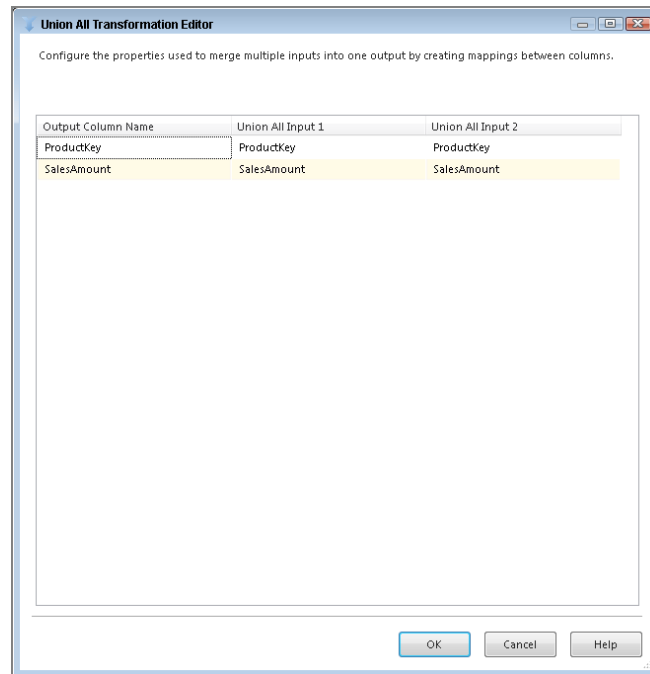


Fig. 1

7. Now bring a **Flat File Destination** to the design surface and connect the **Union All Transform** to it. Name the destination **Sales Export**.
8. Open the **Flat File Destination** and select **New** to create a delimited Flat File Connection Manager.
9. Name the Flat File Connection Manager **Flat File Sales Export**. Then call the file **SalesExport.txt**, and select location for it. Also, check the **Column names in the first data row** option. Click **OK** on the connection manager. Ensure that you select the **Mappings** page on the destination so each column is set correctly. Click **OK** on the destination.
10. The package is now complete. When the package is executed, your results will look like Figure 2.

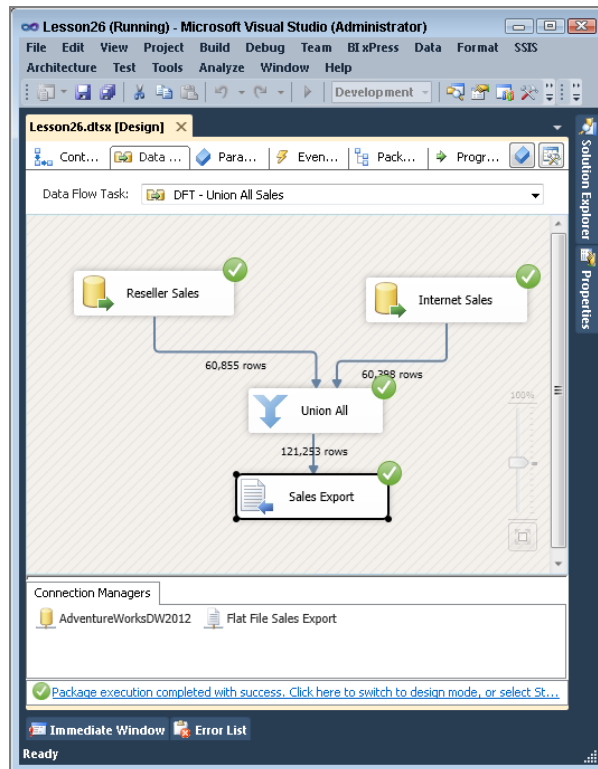


Fig. 2

Bibliografia

- 1) Knight B., Knight D., Davis M, Snyder W. (2013): Knight's Microsoft® SQL Server® 2012 Integration Services 24-Hour Trainer, John Wiley & Sons.
- 2) Knight B., Veerman E., Moss J.M., Davis M., Rock C. (2012): PROFESSIONAL Microsoft® SQL Server® 2012 Integration Services, John Wiley & Sons.
- 3) <http://www.wrox.com/WileyCDA/Section/id-814197.html>
- 4) [https://msdn.microsoft.com/library/ms169917\(SQL.120\).aspx](https://msdn.microsoft.com/library/ms169917(SQL.120).aspx)
- 5) Tok W-H., Parida R. Masson M. Ding X. Sivashanmugam (2012): Microsoft SQL Server 2012 Integration Services, Promise (tłumaczenie j. polski).
- 6) Kimball R. (2004): The Data Warehouse ETL Toolkit. John Wiley & Sons