

Print this Page

SQL-Server-Performance.com

Pivot and UnPivot with SSIS

By : Dinesh Asanka

Introduction

The presentation of your data is required for easy analysis. Turning columns into rows and rows into columns is another way of presenting your data so that end-users can understand it easily. Other than presentation purposes, you need to transform your data within your data warehouse application into different data formats. This process can be referred to as 'Pivoting', and the reversing of this process can be referred to as 'UnPivoting'.

In this article, we will discuss how you can use the Pivot and Unpivot processes in SQL Server Integration Service (SSIS).

Requirements

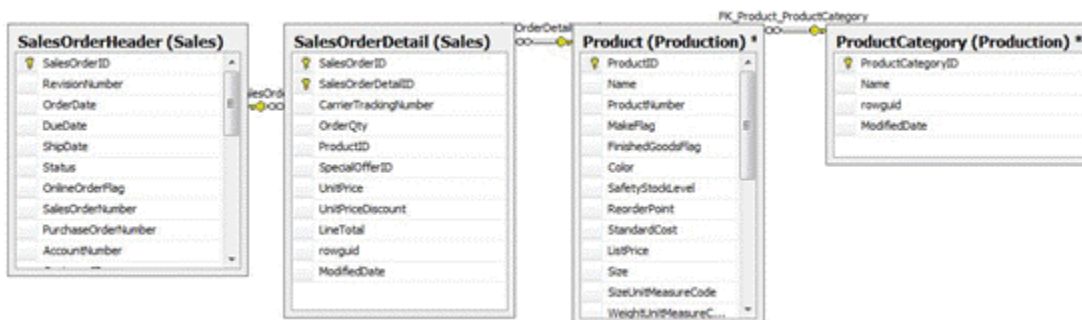
As this article will not tell you how to write SSIS packages, it is a requirement to have a fair amount of experience in creating SSIS packages. However, I will try to discuss details of creating SSIS packages wherever necessary without disturbing the main topics of the article. In addition, it is necessary to know how to write a query with joining three or more tables.

Software requirements include SQL Server 2005 installed with SQL Server Business Intelligence Development Studio. This article is written with the assumption that the Adventureworks database's data is being used. It would be optimal for the reader to have an installed Adventureworks database along with a SQL Server 2005 database server.

The data file for this exercise can be found [here](#).

Pivot

To understand what pivoting is, let's see an example. Below illustrates the relationship in the Adventureworks database of Sales.SalesOrderHeader, Sales.SalesOrderOrder, Product.Product and Production.ProductCategory tables.



From the above relationship, we can assume that we need following output.

Product	Qrt1	Qtr2	Qrt3	Qrt4
Accessories			870	411
Bikes	1167	1369	2844	2495
Components	2641	2966	6173	5253

It is known that you will not retrieve the above data set by a simple T-SQL query. However, we can build the following format with a simple T-SQL code.

Name	Qtr	OrderQty
Accessories	3	870
Accessories	4	411
Bikes	1	1167
Bikes	2	1369
Bikes	3	2844
Bikes	4	2495
Components	1	2641
Components	2	2966
Components	3	6173
Components	4	5253

Turning the above table into the required format is an example of “pivoting”.

Let’s start with the design package. First, add a new package after creating Integration Project.

Then add a Data flow task to control flow. Next, add an OLE DB connection and configure that OLE DB connection to the Adventureworks database.

Finally, add an OLE DB source to the added data flow. While we are going to design the SSIS package with minimum coding, we will use as many SSIS controls as possible. Thus, we will add following T-SQL to the added OLE DB Source.

```
SELECT PC.Name
```

```
,soh.OrderDate
```

```
,SOD.OrderQty
```

```
FROM Sales.SalesOrderDetail SOD
```

```
INNER
```

```
JOIN Sales.SalesOrderHeader SOH
```

```
ON SOH.SalesOrderID = SOD.SalesOrderID
```

```
INNER
```

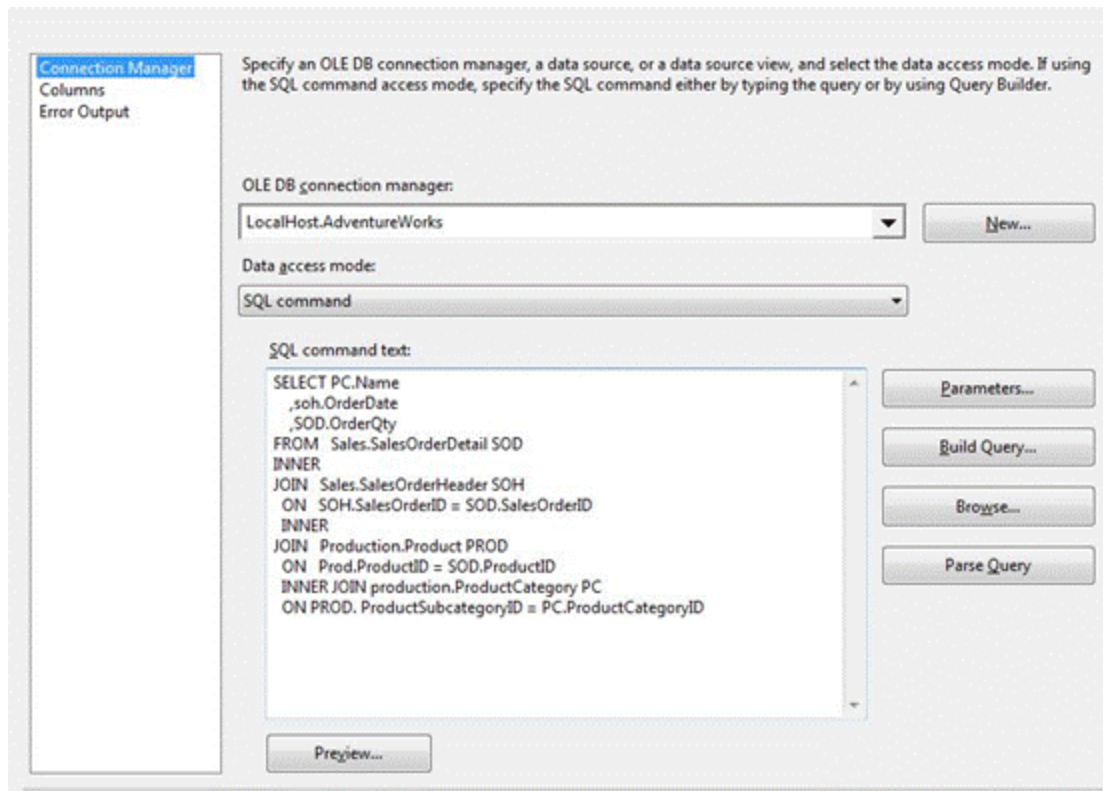
```
JOIN Production.Product PROD
```

```
ON Prod.ProductID = SOD.ProductID
```

```
INNER JOIN production.ProductCategory PC
```

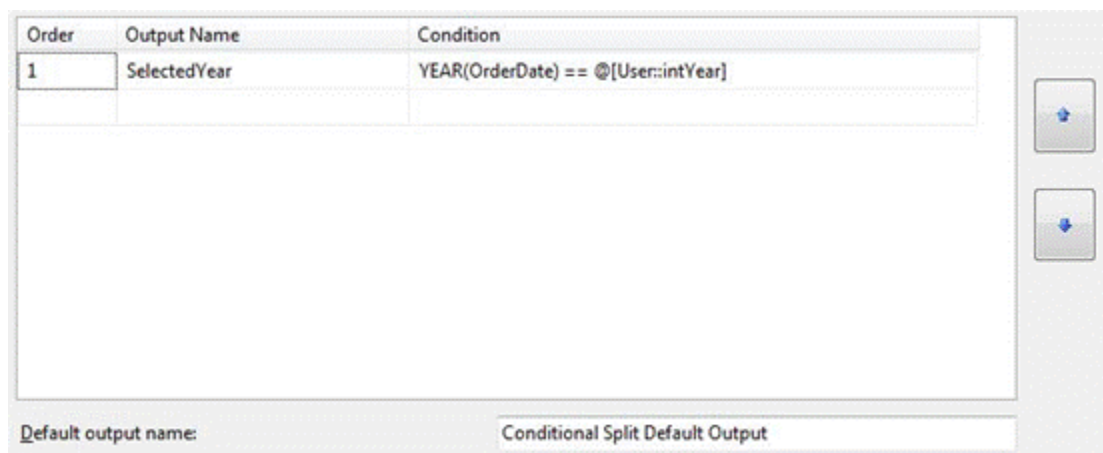
ON PROD. ProductSubcategoryID = PC.ProductCategoryID

After including the above code, the OLE DB Source will resemble the screenshot below.



We are going to add a condition split since we are doing this pivot for only a 'year'. To improve the usability, I have included a variable name called 'intYear'. Therefore, if you want to change the year, it is just a matter of changing the value of the variable 'intYear'.

The following is a screenshot of the conditional split.

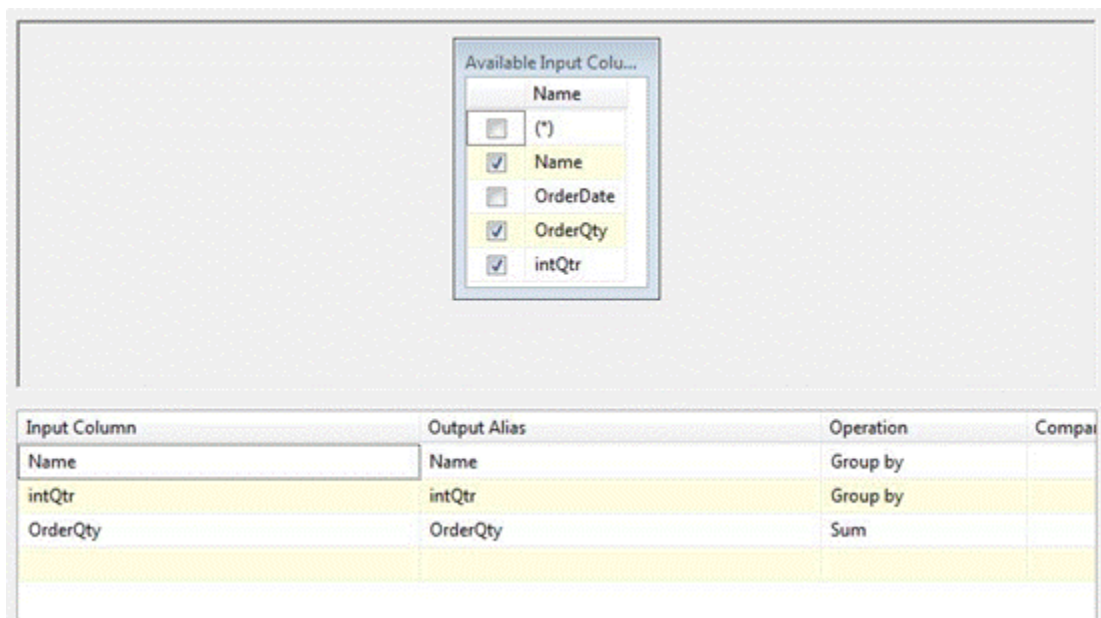


In the above case, the value of @[User::intYear] is set to 2001.

Next, we need to derive the Quarter. Even though we can modify the initial T-SQL to return the Quarter, I have used derive column data flow transformation task. The following expression is used to derive the Quarter.

MONTH(OrderDate) >= 1 && MONTH(OrderDate) <= 3 ? 1 : MONTH(OrderDate) >= 4 && MONTH(OrderDate) <= 6 ? 2 : MONTH(OrderDate) >= 7 && MONTH(OrderDate) <= 9 ? 3 : MONTH(OrderDate) >= 10 && MONTH(OrderDate) <= 12 ? 4 : 0

We now need to group the above data with Category and Quarter. We can use aggregate transformation and configure it to be grouped by Name and intQtr.



Next we need to add a sort transformation, and here I have used category to sort. We also need to sort the key column, otherwise pivot will not work properly. To see the data up to this point, you can add a data viewer.

Below is the screenshot of the data set should be getting, which is the data set we need to pivot.

Sort Output Data Viewer 1 at Sort.Sort Output

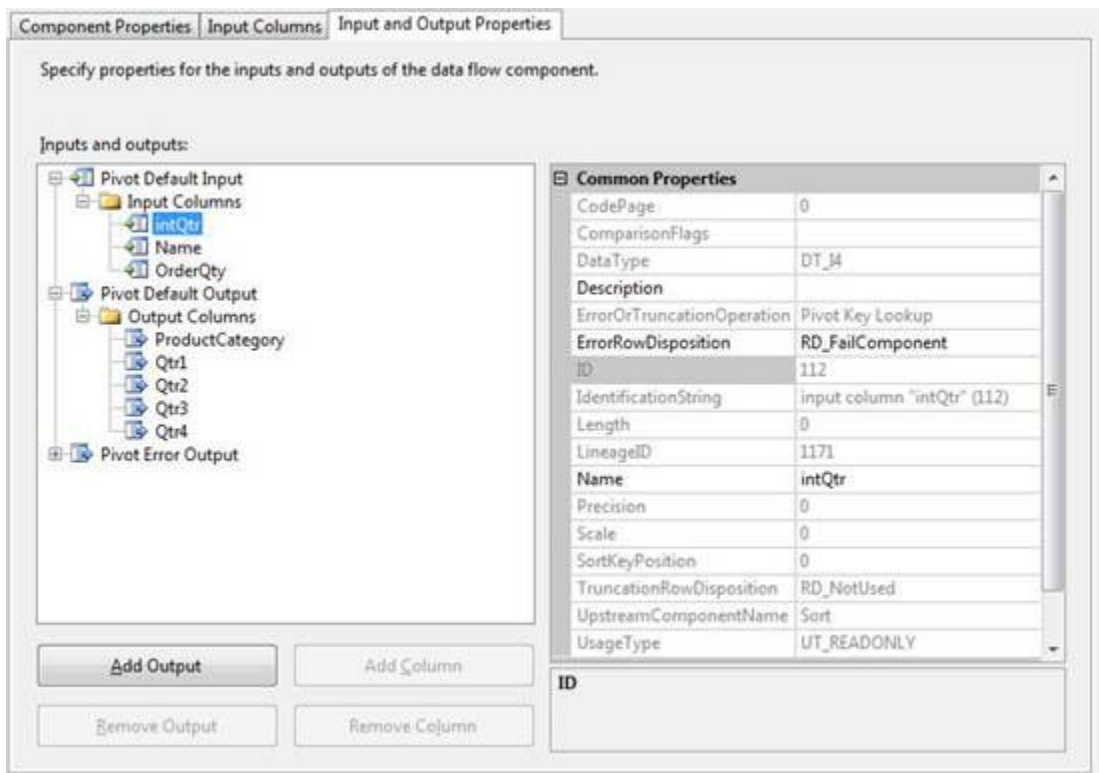
▶ Detach Copy Data

Name	OrderQty	intQtr
Accessories	870	3
Accessories	411	4
Bikes	1167	1
Bikes	1369	2
Bikes	2844	3
Bikes	2495	4
Components	2641	1
Components	2966	2
Components	6173	3
Components	5253	4

Attached Total rows: 10, buffers: 1 Rows displayed = 10

We have now reached the core part of this article- pivoting. For pivoting, there is a pivot transformation configuration which is not exactly straight forward. At input tab of the pivot transformation, you need to select columns that you would use in the pivot operation, which in this case would be all three available columns.

The next most important tab is the 'Input and Output' properties tab, pictured below.



For input columns, we need to configure the pivot usage attribute.

Option	Description
0	The column is not pivoted, and the column values are passed through to the transformation output.
1	The column is part of the set key that identifies one or more rows as part of one set. All input rows with the same set key are combined into one output row.
2	The column is a pivot column. At least one column is created from each column value.
3	The values from this column are placed in columns that are created as a result of the pivot.

Source: Books on line, SQL Server 2005

According to the above table, Name column should be Option 1 , intQtr should have Option 2 and OrderQty should have Option 3 for pivot usage attribute value.

The next step is to configure the output columns.

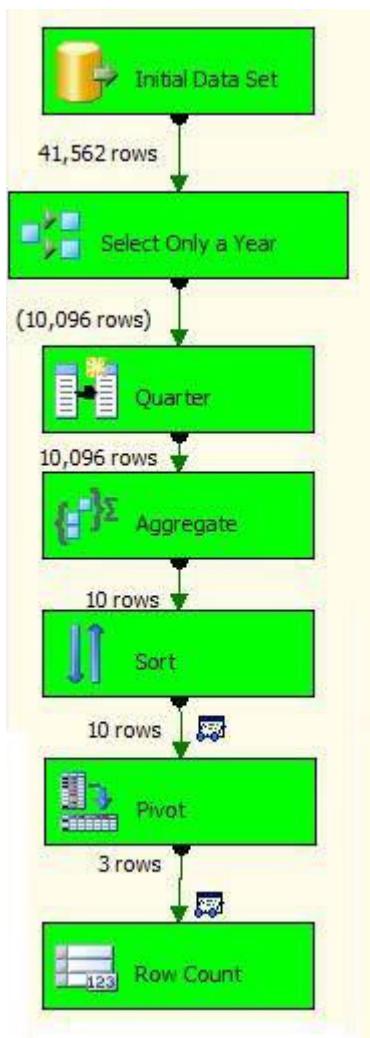
By clicking Add column button, ProductCategory, Qtr1, Qtr2, Qtr3 and Qtr4 are added. We have to configure the SourceColumn and PivotKeyValue attributes for those four columns.

For the ProductCategory column, assign LineageID of the Name input column to the SourceColumn. In this sample, it is 1161. For the Qtr1 column, assign LineageID of the OrderQty input column to the SourceColumn, which is 1166 in this example. PivotKeyValue is the column value of the initial data set. For Qtr1, we need records which has the value 1 for intQtr. So the PivotKeyValue for Qtr1 is 1.

Similary, Qtr2 will have 1166 for the SourceColumn and 2 for the PivotKeyValue. Qtr3 will have 1166 for the SourceColumn and 3 for the PivotKeyValue. Qtr4 will have 1166 for the SourceColumn and 4 for the PivotKeyValue.

Here we have also added another data viewer after the pivot transformation. Following is the output of the data viewer, thus achiving our goal.

The following depicts the final layout for the package.

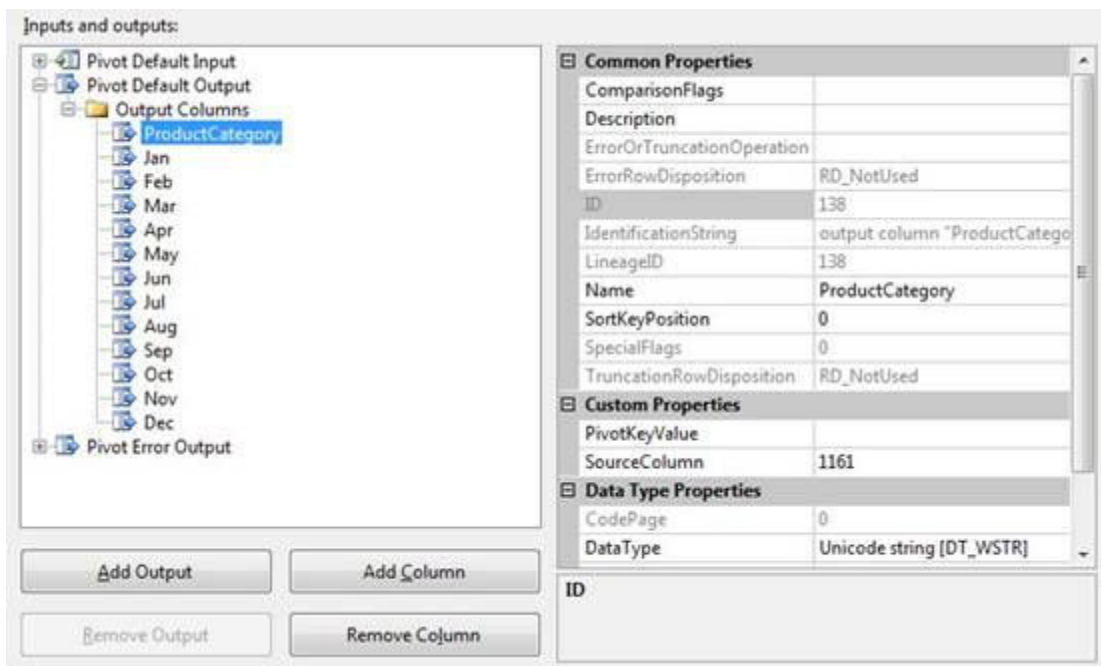


We can extend this package to view Months from Quarters. For this, we need to change the derive column transformation. The following shows how this can be done.

MONTH(OrderDate) == 1 ? "Jan" : MONTH(OrderDate) == 2 ? "Feb" : MONTH(OrderDate) == 3 ? "Mar" : MONTH(OrderDate) == 4 ? "Apr" : MONTH(OrderDate) == 5 ? "May" : MONTH(OrderDate) == 6 ? "Jun" : MONTH(OrderDate) == 7 ? "Jul" : MONTH(OrderDate) == 8 ? "Aug" : MONTH(OrderDate) == 9 ? "Sep" : MONTH(OrderDate) == 10 ? "Oct" : MONTH(OrderDate) == 11 ? "Nov" : MONTH(OrderDate) == 12 ? "Dec" : "UNK"

The sorting was done for the Month column in the Sort transformation.

The next key configuration is the Input and Output Configuration of the Pivot transformation. We will need to add all twelve columns for Month apart from the Product Category column.



The configuration is similar as in the first example. For example, the Jan column PivotKeyValue is Jan and the SourceColumn is LineageID of OrderQty of InputColumns.

Below depicts the final output of this package.

ProductCategory	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Accessories	NULL	NULL	NULL	NULL	NULL	NULL	204	353	313	117	166	128
Bikes	206	516	445	234	635	500	751	1088	1005	646	968	881
Components	618	1161	862	738	1282	946	1865	2526	1782	1468	2229	1556

In both of these cases, we have used one column to pivot, which is ProductCategory. Now we will attempt to add two columns for pivoting - Year and Month.

We will need to add the Year column, which is YEAR(OrderDate) at derive column. Year column was included for Aggregate transformation. Sorting is done for Month and Year.

Let us see what we have to do for Pivot transformation. PivotUsage of intYear column is 1. Source column of Year output column LineageID of intYear input column.

Below depicts the output for the above package.

Year	ProductCategory	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2002	Accessories	NULL	NULL	NULL	NULL	NULL	NULL	204	353	313	117	166	128
2003	Accessories	51	75	80	100	138	156	215	381	386	120	131	188
2004	Accessories	40	76	75	91	188	178	NULL	NULL	NULL	NULL	NULL	NULL
2001	Bikes	NULL	NULL	NULL	NULL	NULL	NULL	148	519	383	213	660	492
2002	Bikes	206	516	445	234	635	500	751	1088	1005	646	968	881
2003	Bikes	580	887	680	746	937	753	876	1312	1333	878	1234	1332
2004	Bikes	758	1084	1033	940	1332	1336	NULL	NULL	NULL	NULL	NULL	NULL
2003	Clothing	NULL	NULL	NULL	NULL	NULL	NULL	890	1474	1700	829	1254	1610
2004	Clothing	732	1018	1447	890	1301	1606	NULL	NULL	NULL	NULL	NULL	NULL
2001	Components	NULL	NULL	NULL	NULL	NULL	NULL	453	764	575	667	1258	1007
2002	Components	618	1161	862	738	1282	946	1865	2526	1782	1468	2229	1556
2003	Components	1176	2029	1286	1454	2283	1485	1225	1562	1311	1082	1436	1408
2004	Components	965	1374	1170	1217	1597	1379	NULL	NULL	NULL	NULL	NULL	NULL

Unpivot

Now that we have reviewed pivoting, it is time to learn how to “Unpivot”. The unpivot transformation makes a de-normalized dataset into a more normalized version by expanding values from multiple columns in a single record into multiple records within the same values in a single column. It is the exact the opposite of pivoting.

We can begin with our earlier data format, which is in a comma separated value (csv) format.

2002,Accessories,,,,,,,,,204,353,313,117,166,128

2003,Accessories,51,75,80,100,138,156,215,381,386,120,131,188

2004,Accessories,40,76,75,91,188,178,,,,,,,,

2001,Bikes,,,,,,,,,148,519,383,213,660,492

2002,Bikes,206,516,445,234,635,500,751,1088,1005,646,968,881

2003,Bikes,580,887,680,746,937,753,876,1312,1333,878,1234,1332

2004,Bikes,758,1084,1033,940,1332,1336,,,,,,,,

2003,Clothing,,,,,,,,,890,1474,1700,829,1254,1610

2004,Clothing,732,1018,1447,890,1301,1606,,,,,,,,

2001,Components,,,,,,,,,453,764,575,667,1258,1007

2002,Components,618,1161,862,738,1282,946,1865,2526,1782,1468,2229,1556

2003,Components,1176,2029,1286,1454,2283,1485,1225,1562,1311,1082,1436,1408

2004,Components,965,1374,1170,1217,1597,1379,,,,,,,,

You can get the above data in a text file name PivotedData.txt, which you will find with the sample package.

First, we will add a package to the existing SSIS solution. Then we will add a Flat File Connection Manager and

point the pivoteddata.txt file to it. Finally, we will add a Flat file source and point the previously added Flat File Connection Manager to it. Below depicts the data output you should see from the Flat File Source.

Year	ProductCategory	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2002	Accessories	0	0	0	0	0	0	204	353	313	117	166	128
2003	Accessories	51	75	80	100	138	156	215	381	386	120	131	188
2004	Accessories	40	76	75	91	188	178	0	0	0	0	0	0
2001	Bikes	0	0	0	0	0	0	148	519	383	213	660	492
2002	Bikes	206	516	445	234	635	500	751	1088	1005	646	968	881
2003	Bikes	580	887	680	746	937	753	876	1312	1333	878	1234	1332
2004	Bikes	758	1084	1033	940	1332	1336	0	0	0	0	0	0
2003	Clothing	0	0	0	0	0	0	890	1474	1700	829	1254	1610
2004	Clothing	732	1018	1447	890	1301	1606	0	0	0	0	0	0
2001	Components	0	0	0	0	0	0	453	764	575	667	1258	1007

Our next steps is to do the unpivoting. Unlike the pivot configuration, the unpivot configuration is relatively simple.

Unpivot Transformation Editor

Specify the columns to pivot into rows to make an unnormlized dataset into a more normalized version.

Available Input Columns

Name	Pass Through
<input type="checkbox"/> Year	<input checked="" type="checkbox"/>
<input type="checkbox"/> ProductCategory	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Jan	<input type="checkbox"/>
<input checked="" type="checkbox"/> Feb	<input type="checkbox"/>
<input checked="" type="checkbox"/> Mar	<input type="checkbox"/>
<input checked="" type="checkbox"/> Apr	<input type="checkbox"/>
<input checked="" type="checkbox"/> May	<input type="checkbox"/>

Input Column	Destination Column	Pivot Key Value
Jan	Qty	Jan
Feb	Qty	Feb
Mar	Qty	Mar
Apr	Qty	Apr
May	Qty	May
Jun	Qty	Jun
Jul	Qty	Jul
Aug	Qty	Aug
Sep	Qty	Sep

Pivot key value column name: Month

OK Cancel

You will need to select the pivot columns, which in this case the pivot columns are Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov and Dec. The pivot column name will be in column titled Month, which you can configure at the bottom of the screen. The values of these pivotedkey columns (Jan, Feb etc.) into column called Qty which

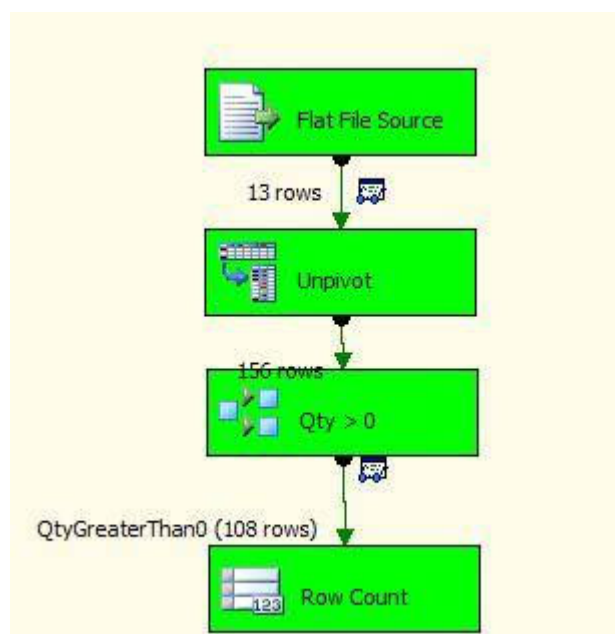
can be configured at destination column.

In this unpivot example there is a small issue. Here the columns which have a value of "0" in the Qty column will be transformed into rows. For example, the Qty of accessories in January of 2002 is '0' and there will be an unnecessary row for that record. To alleviate these records, a conditional split with a simple condition is used. The condition is $Qty > 0$, which will eliminate any row with "0" in the Qty field.

Below is the output of the unpivoted transformation.

Month	Year	ProductCategory	Qty
Jan	2003	Accessories	51
Jul	2003	Accessories	215
Jun	2003	Accessories	156
Mar	2003	Accessories	80
May	2003	Accessories	138
Nov	2003	Accessories	131
Oct	2003	Accessories	120
Sep	2003	Accessories	386
Apr	2004	Accessories	91
Feb	2004	Accessories	76
Jan	2004	Accessories	40
Jun	2004	Accessories	178
Mar	2004	Accessories	75
May	2004	Accessories	188
Aug	2001	Bikes	519
Dec	2001	Bikes	492
Jul	2001	Bikes	148

Below depicts the image of the final unpivoted package.



Key to Sample SSIS Package

Package	Description
PivotSampleForQuarter.dtsx	Pivot data for One Year Quarter wise
PivotSampleForMonths.dtsx	Pivot data for One Year monthly wise
PivotSampleForAllYears.dtsx	Pivot data for all data including year
UnpivotSample.dtsx	UnPivot Data

Conclusion

Pivot and unpivot transformations are important controls from which you can improve your data presentation. With minimum time and knowledge, you can successfully change your data representation.

© 2000 - 2007 vDerivatives Limited All Rights Reserved.