To produce a valid result, you have to change the connection string, located in app.config.

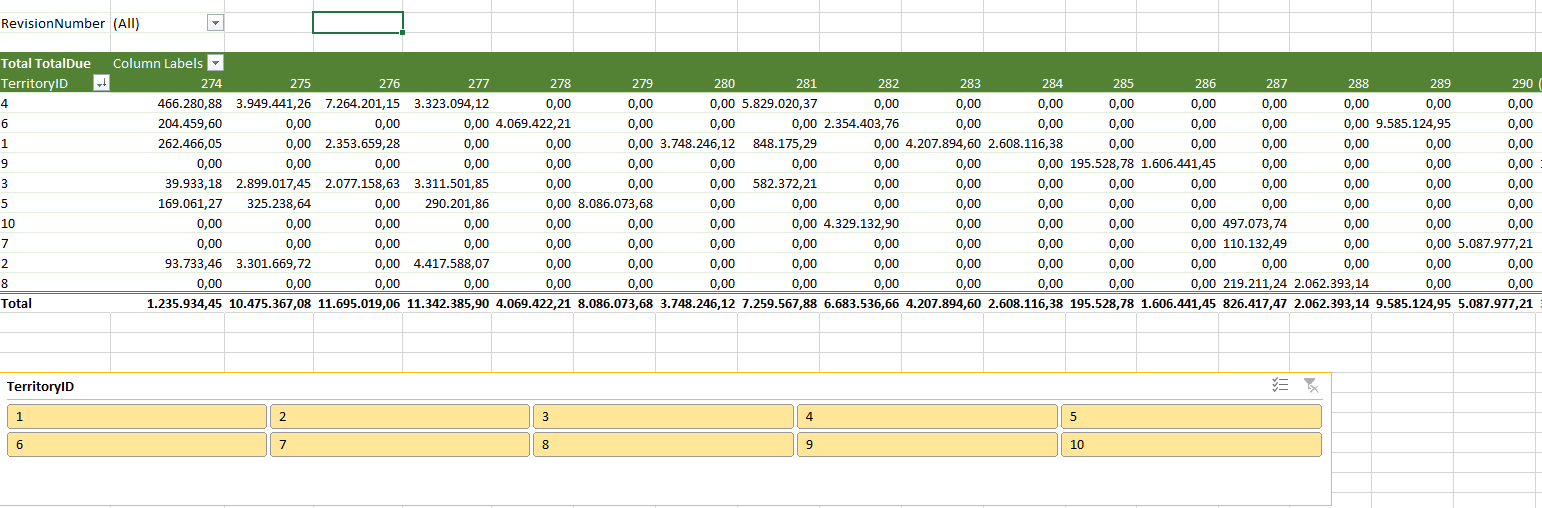
Currently, in the app.config under the element Data Source is written ‘your server name’.



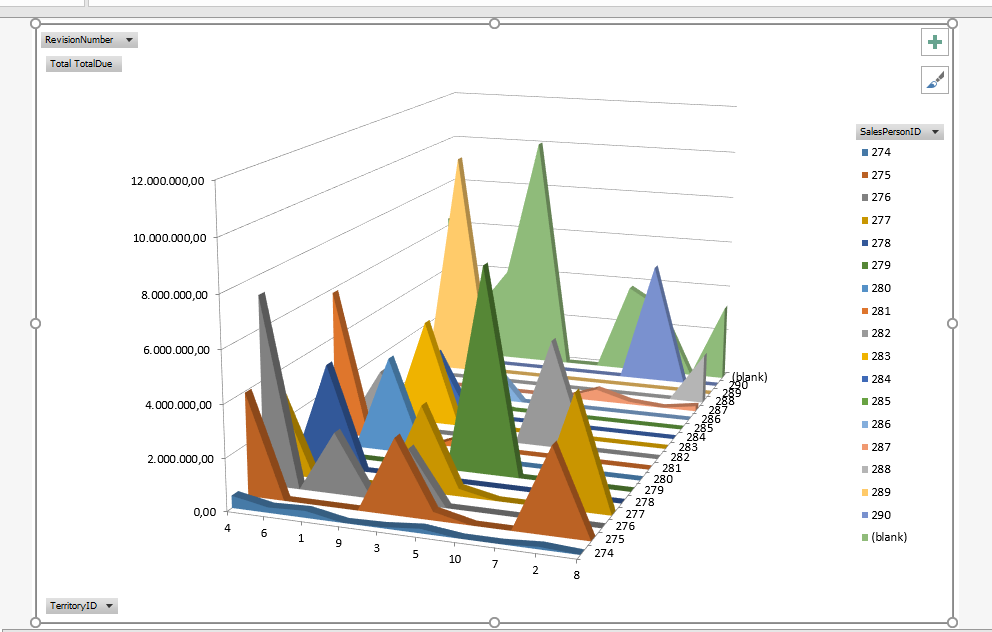
I presume that at first, you will point to the instance that contains the AdventureWorks2016 database.

If you set the project as the startup project and press F5, the Excel file will be generated. The file is part of the project as an embedded resource. Defaults are that the file will be generated on ‘C:\TMP’ folder.

The file contains two sheets, the pivot, and the chart. The pivot is shown in the image below.



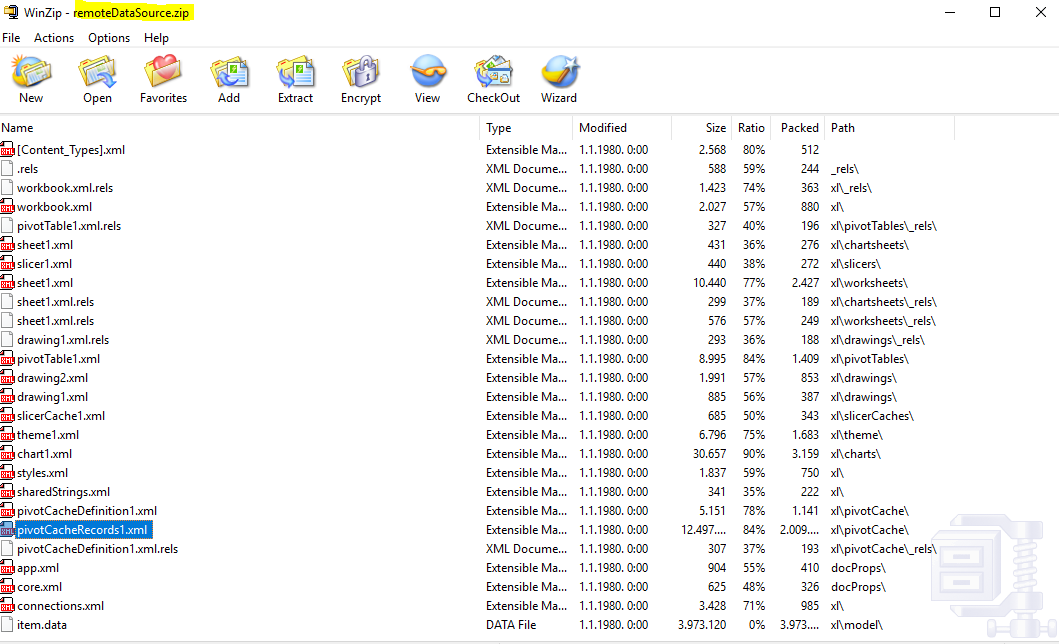
On the other tab(sheet), the chart is displayed.



Using the technique shown in the project we could extend Excel limitation ( 1,048,576 rows) and produce a pivot with more than one million records.

If we change the file extension from XLSX to ZIP, we will be able to browse the structure of the pivot file.

Actual data is stored in PivotCacheRecords.xml, see in the image below.



By using this approach we could achieve a couple of benefits. First, we will reduce the size of the Excel file, then we could refresh the pivot when the data has been changed in the database. How to prove that?

By issuing the following T-SQL statement

WITH X

AS (SELECT \*

FROM sales.SalesOrderHeader

WHERE SalesOrderID = 43706)

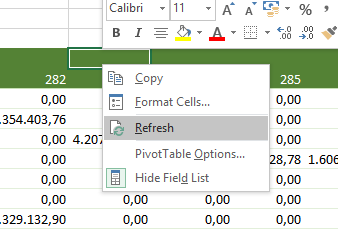
UPDATE x

SET

SubTotal = SubTotal + 1000000000000;

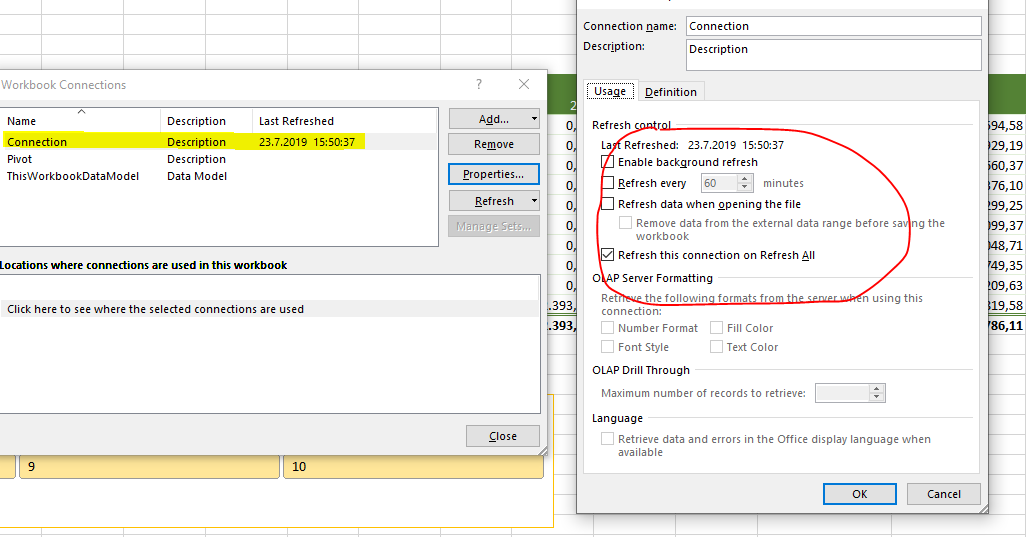
So, the statement above is just an update statement. For SalesOrderId we will enlarge the column SubTotal, and because the column TotalDue is computed column which depends on the column SubTotal, we will enlarge the values in the Excel file.

In the Excel object model, there is a possibility to update pivot when opening the Excel file. I did not use this approach. Instead, you have to manually refresh the pivot by clicking the mouse right button and pressing refresh, as shown in the image below.



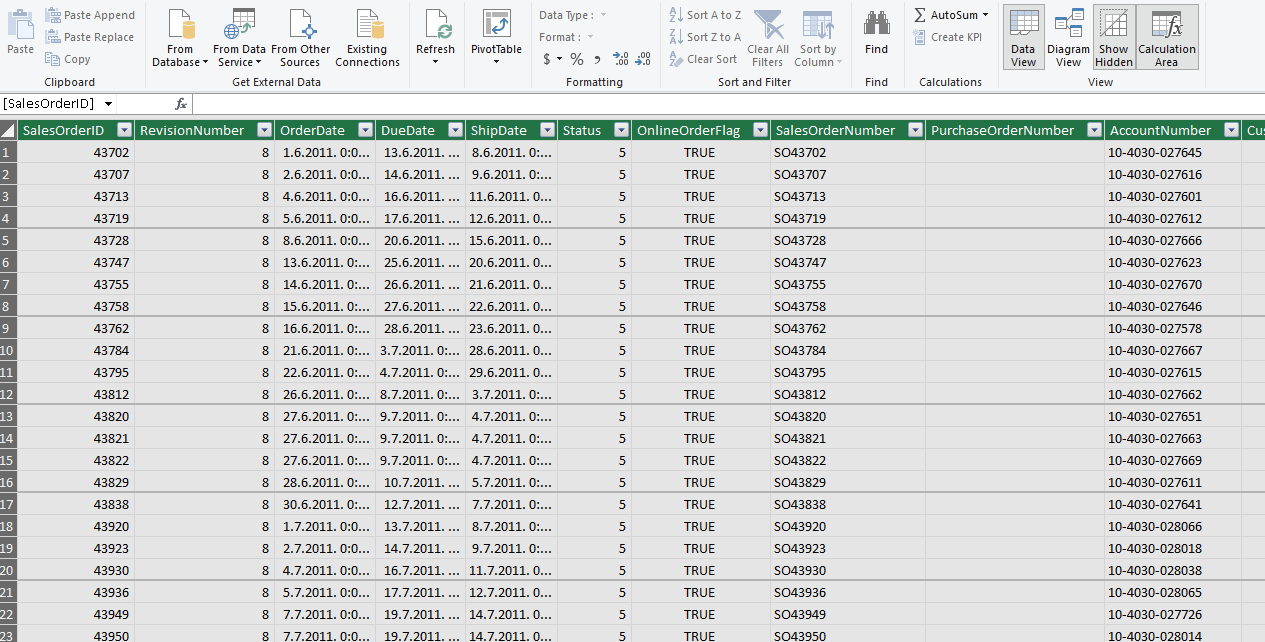
Refreshing won’t work if we can not establish a connection to the database. How we know the connection and when the refresh is the last time completed?

There is the connection command that is part of the Excel UI interface.



If someone asks yourself how we can browse the actual data that build the pivot, there is an option for that as well.

If we click on the power pivot menu and choose Manage command the actual data will be displayed in the separate window.



The source code in the project file is divided into two parts. In the file ExcelHelper, I developed a simple class that handles the pivot creation in a method CreatePivotWithRemoteDataSource.

The test is accomplished in the file Program.cs.