**Power BI - Project**

**Authors:** Bartosz Olechnowicz

**Research object**

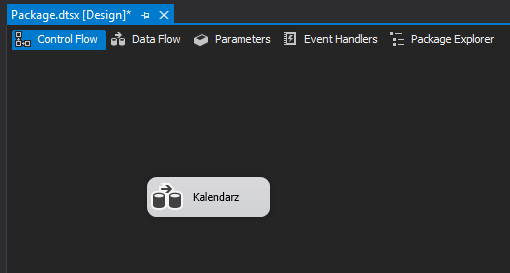
The object of research in the project was the profits generated by the online store in 2015-2018, the analysis of the most frequently purchased products and their prices.

**Import of data from the "Calendar" file to the SQL database**

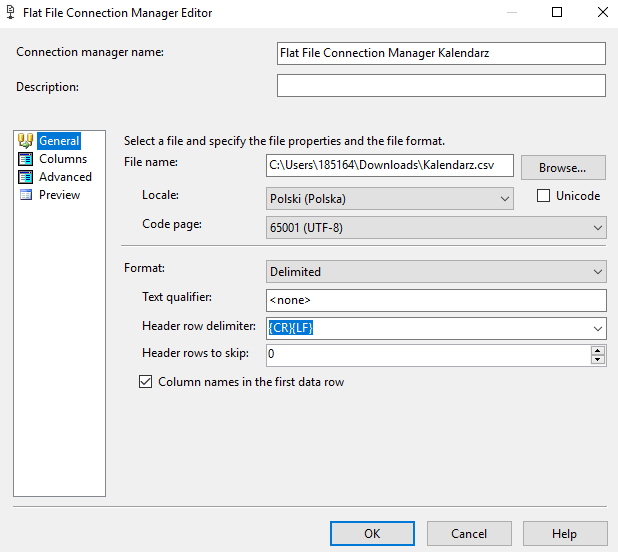
First, Excel generated a calendar by typing a date in the first cell and dragging it down. This created a column called "Date" containing dates from 2015 to 2018:



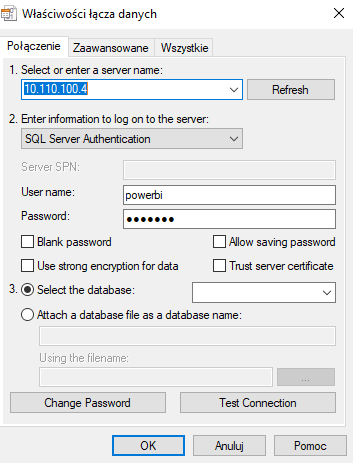
A new project has been created in Visual Studio. During its creation, the "Integration Services Project" option was selected. The next step was to drag the “Data Flow Task” to the project in Visual Studio and rename it to “Calendar”, as you can see in the screenshot below:



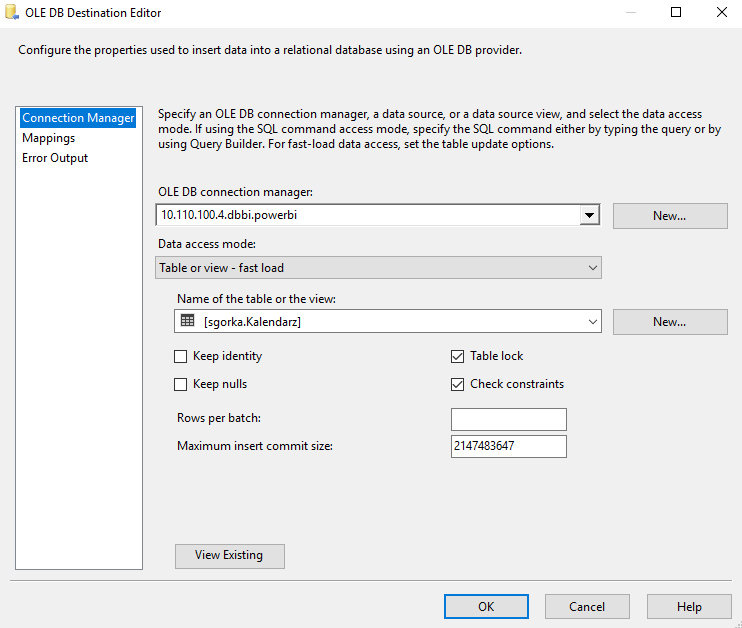
Then, "Source Assistant" was dragged to the "Calendar" item and "Flat file" was selected:



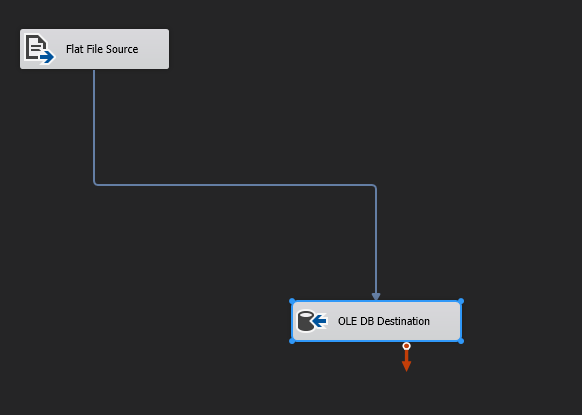
The next step was logging into the database by selecting "SQL Server Authentication" and entering the server data:



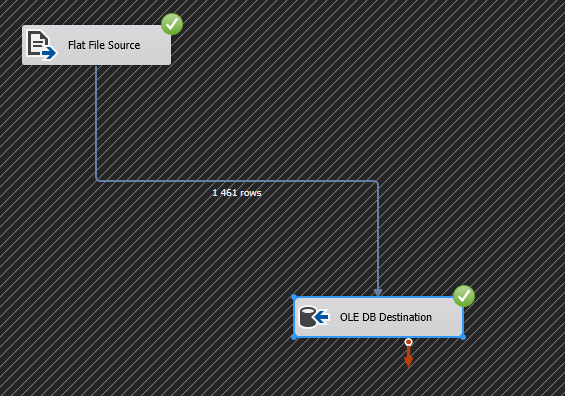
Then “Destination Assistant” was added to the project and the server data was re-entered and the name of the created table was set to “sgorka.Kalendarz”:



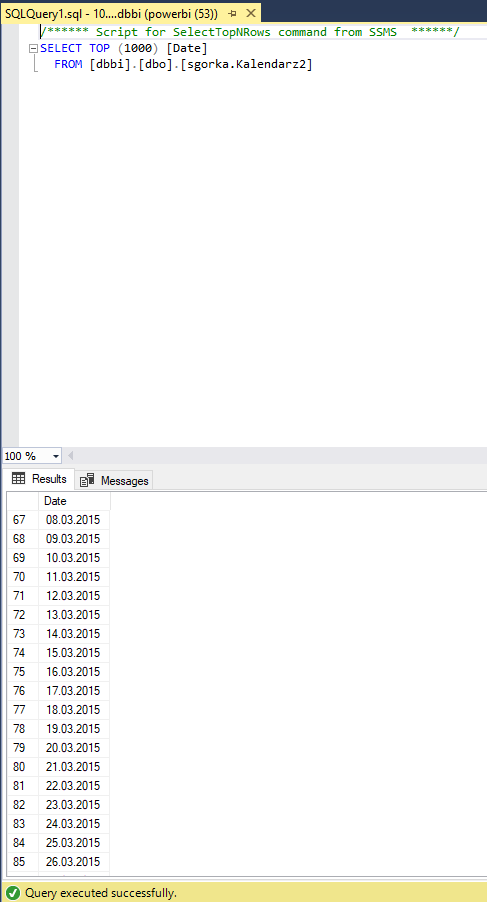
The next step was to connect the elements "Flat File Source" and "OLE DB Destination":



Then clicked the "Start" button to compile the project. The following screenshot shows that the compilation was successful:



After a successful compilation in SQL Server Management Studio on the basis of “dbbi“, the “Refresh” function was used to make the newly created table appear in the “Tables” tab. The next step was to use the “Select top 1000” function on the newly created table to display the first 1000 records from the table. The results are shown in the screenshot below:

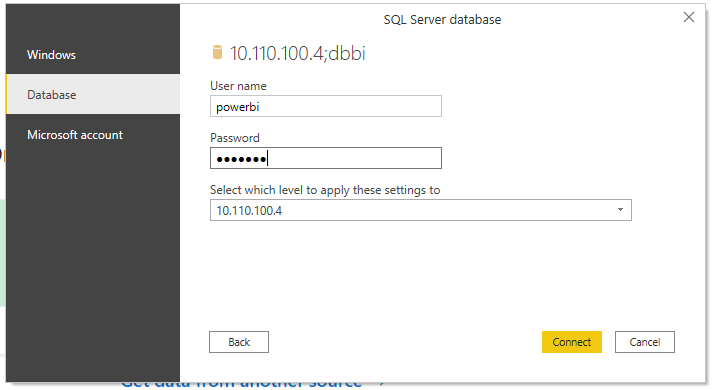


**Import of the “sgorka.Kalendarz” table from the database to Power BI**

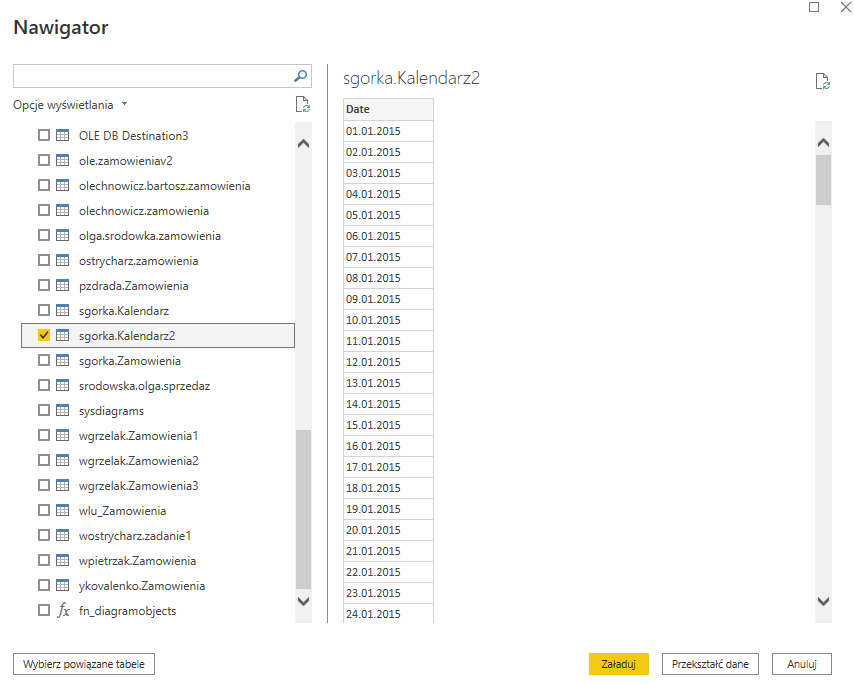
To import data from the database, the "Download data -> SQL Server" function was used and the server address and database name were entered:



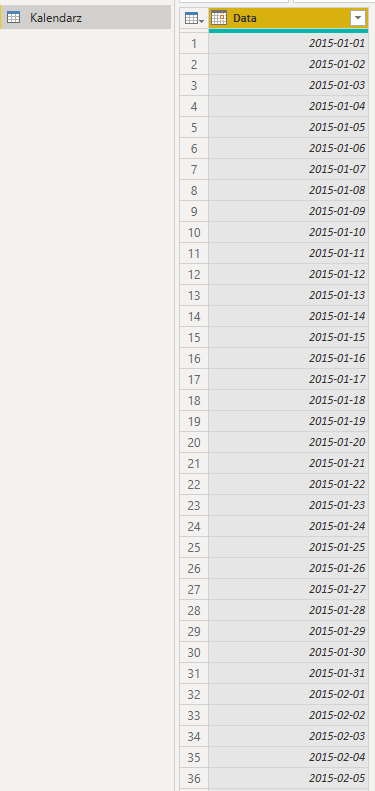
Then you logged into the database by entering your username, password and server address:



Later, our table was found in the list and the "Transform data" option was selected:

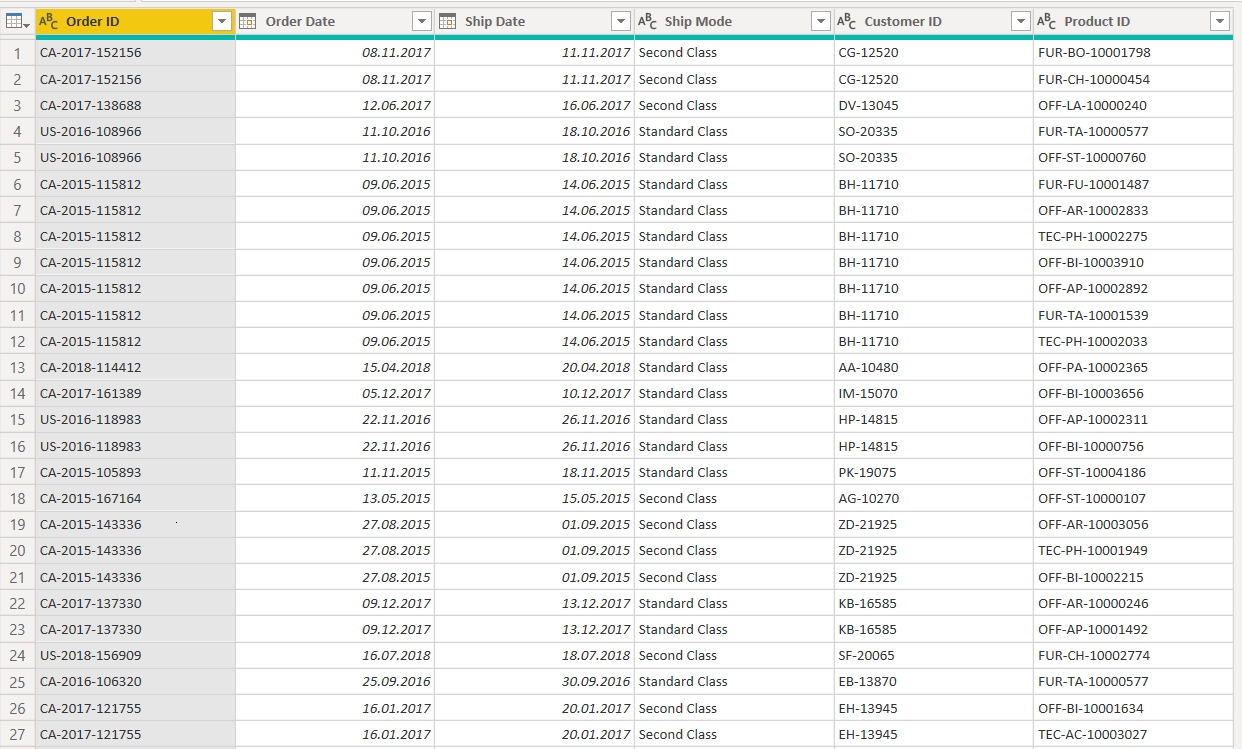


In Power Query Changed the column type "Date" using the "Change type -> Date" function and renamed the column to "Date". The transformed data is shown below:



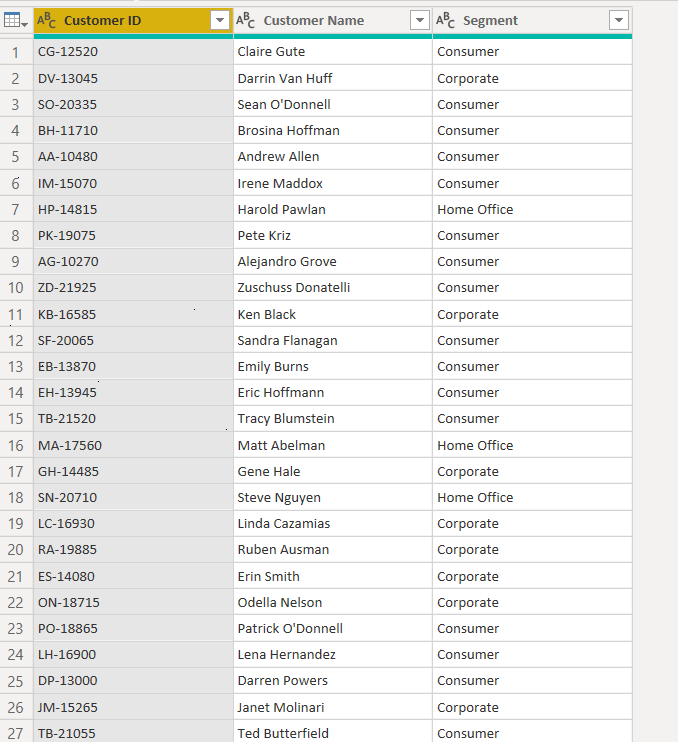
**Import of data from the "Orders" file to Power BI**

The data was imported using the "Get data -> From Excel file" function. Then the "Transform data" function was used. In the next step, the "Use first row as header" option was used so that the table header had column names instead of the table name. The "Remove Duplicates" function has been used in the "Order ID" column because the order number must be unique. Finally, the "Row ID" column was removed, because it does not add anything to the report, and the table name was changed from "Sheet 1" to "Orders". The following screenshot shows the transformed data:



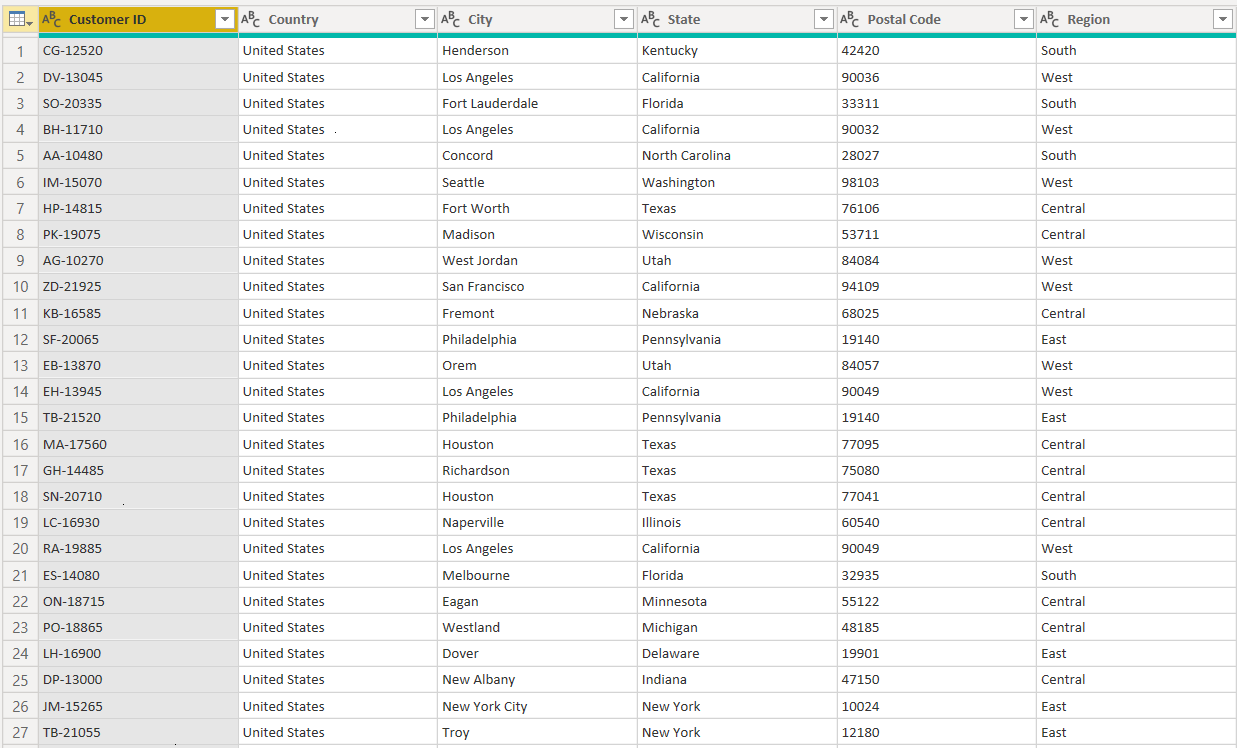
**Import of data from the "Customers" file to Power BI**

Using the "Get data -> From Excel file" function, data was imported and the "Transform data" function was used. Then, just like in the previous table, the "Use first row as header" option was used, the "Row ID" column was removed, the table name was changed from "Sheet 1" to "Customers". The column "Customer ID" uses the function "Remove Duplicates" because the customer ID must be unique. The transformed data is shown below:



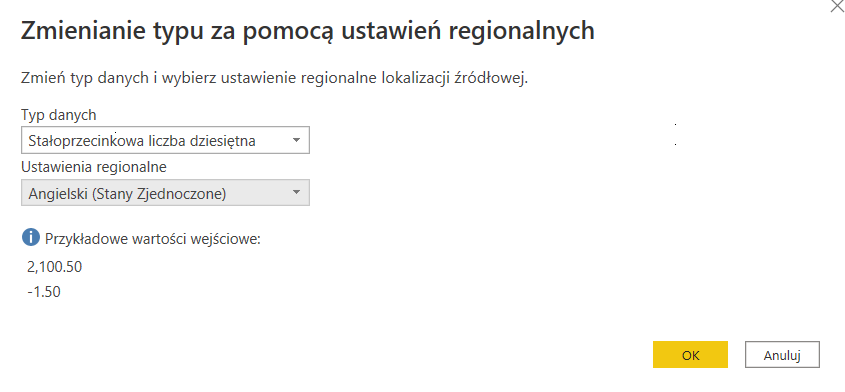
**Import of data from the “Customer Addresses” file to Power BI**

The data was imported using the "Get data -> From CSV file" function. Then the "Transform data" function was used. Then used the "Use first row as header" option twice so that the table header would have the column names instead of the table name. Then we changed the zip code from integer to text format. The column "Customer ID" uses the function "Remove Duplicates" because the customer ID must be unique. Finally, the "Row ID" column was removed, because it does not add anything to the report, and the table was renamed and renamed from "Sheet 1" to "Customer Addresses".

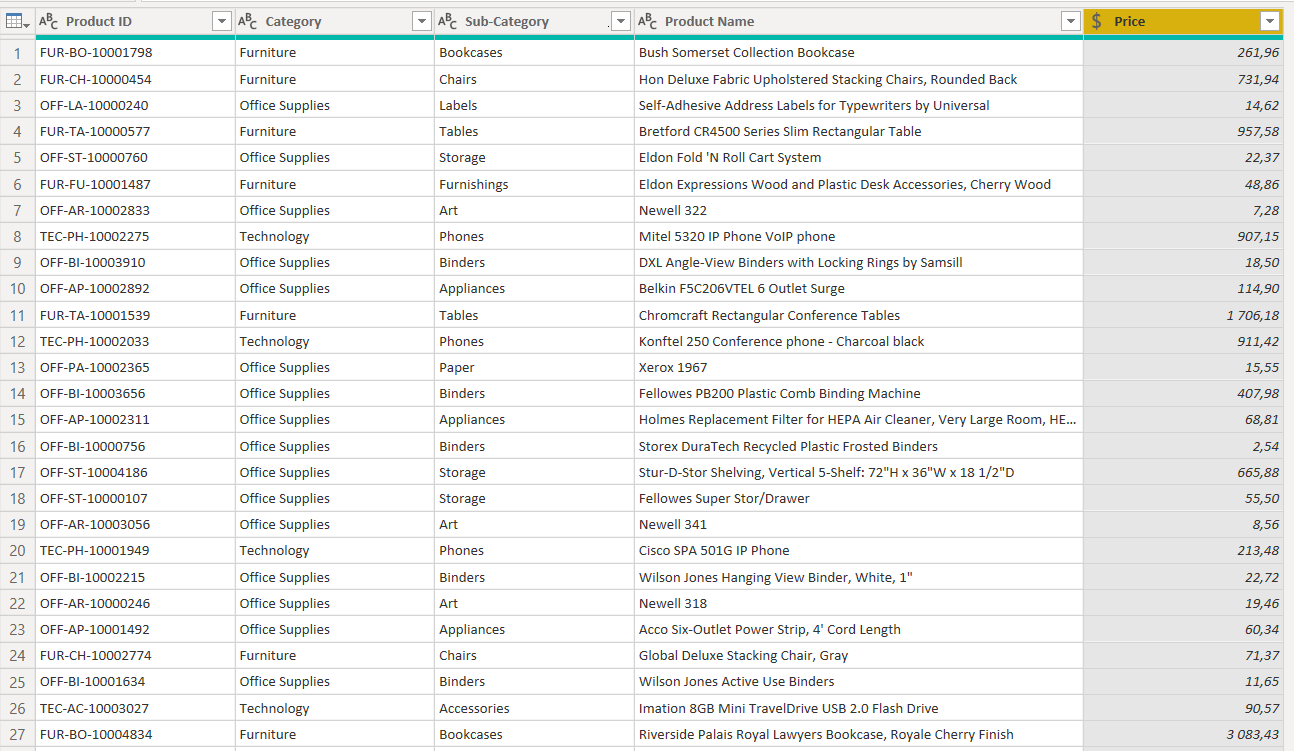


**Import of data from the "Products" file to Power BI**

The data was imported using the "Get data -> From Excel file" function. Then the "Transform data" function was used. Later, the "Use first row as header" option was used. Then, the "Sales" column was typed from text to fixed decimal via locale and renamed to "Price". The column "Product ID" uses the "Remove Duplicates" function because the customer ID must be unique. Finally, the "Row ID" column was removed, because it does not add anything to the report, and the table name was changed from "Sheet 1" to "Products".



The following screenshot shows the transformed data:

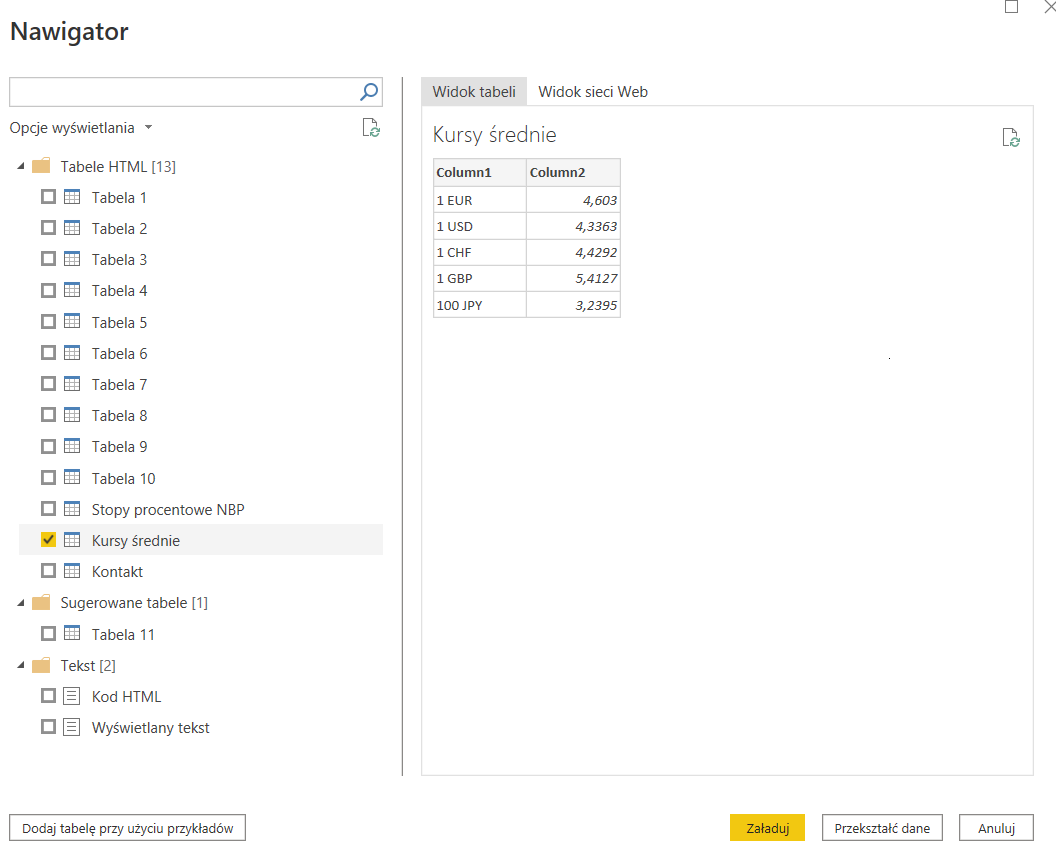


**Import of currency exchange rates from the National Bank of Poland to Power BI**

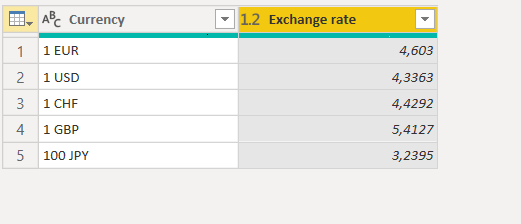
Using the "Download data -> From the Internet" function, data was imported by pasting a link to the page with the exchange rates:



In the navigator, a table called "Average odds" was found and the "Transform data" option was selected:

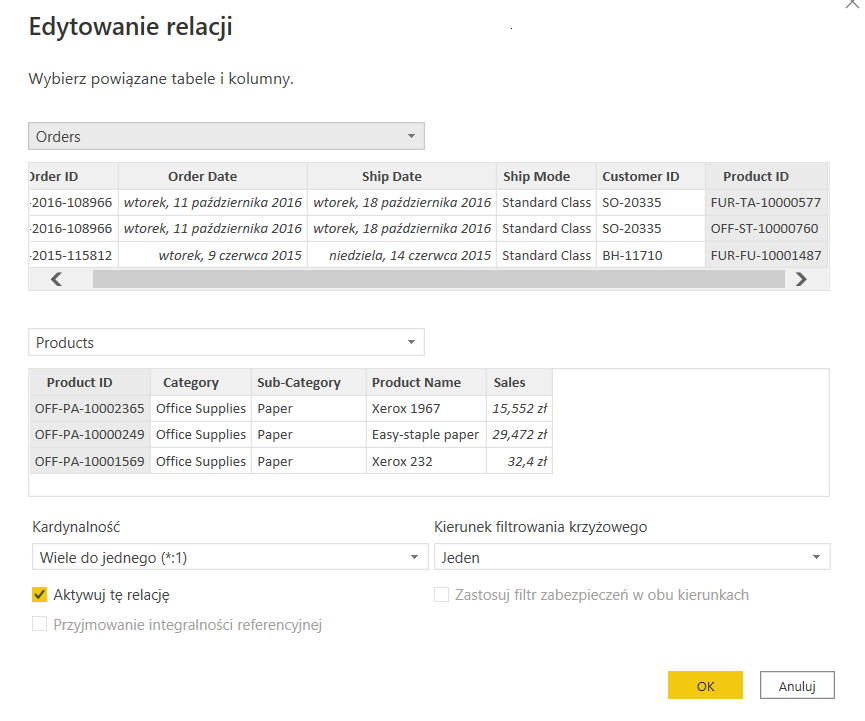


In Power Query, column names have been changed from “Column1” to “Currency” and “Column2” to “Exhange rate” respectively:

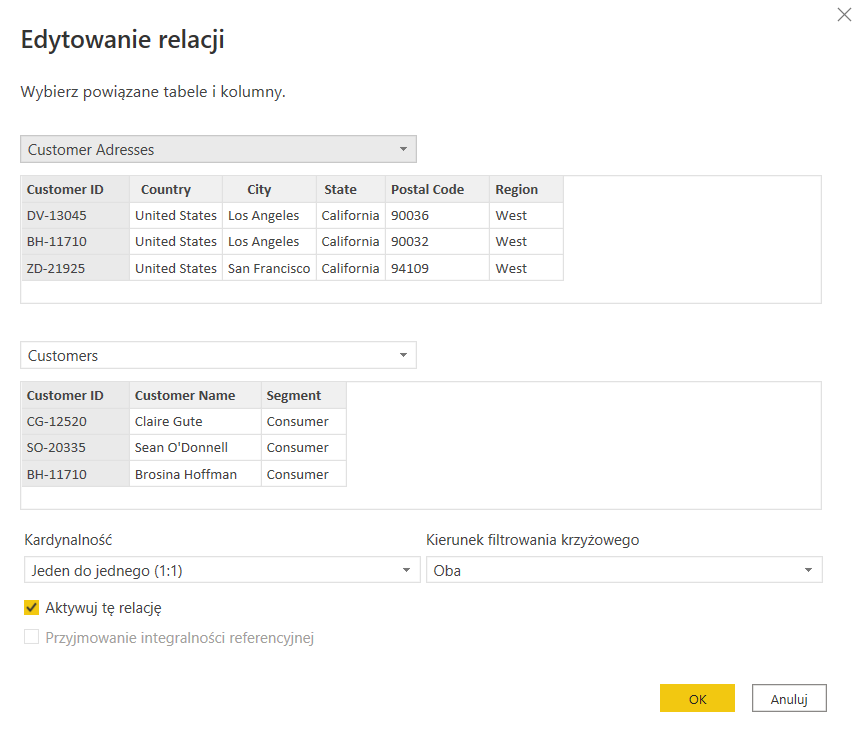


**Preparation of the data model**

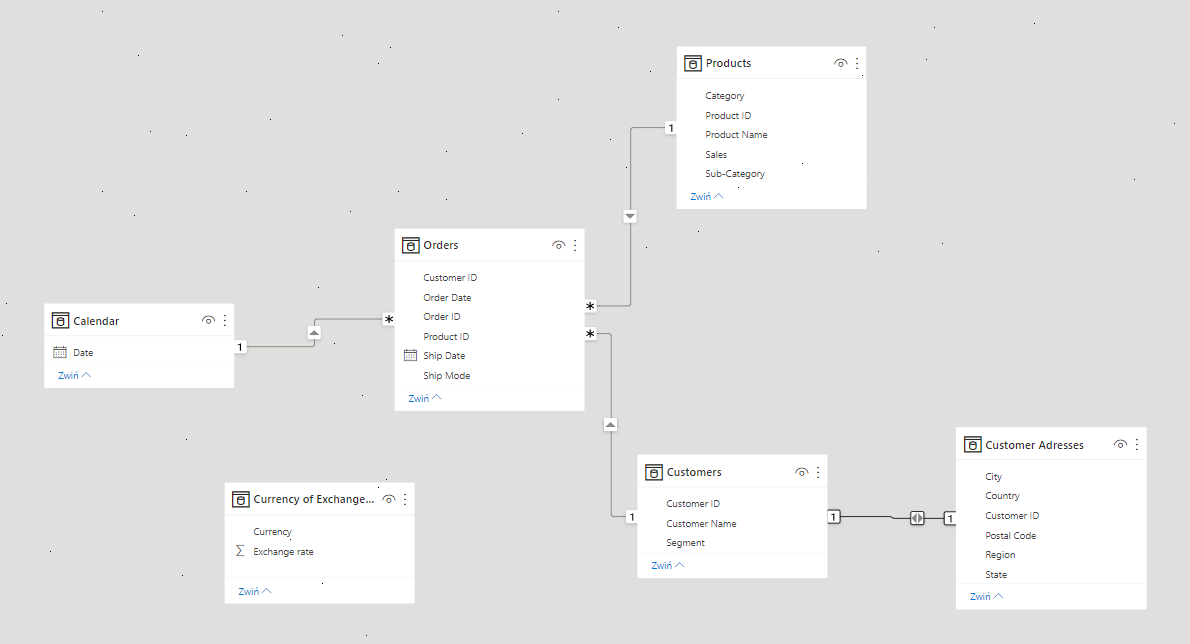
Below is how to create a relationship between the table with orders and the "Product ID" field and the table with products and the "Product ID" field. The “Many to One” cardinality was chosen because the same product can appear in multiple orders. The relationships between the tables "Orders" and "Customers" and "Orders" and "Calendar" were created in a similar way.



The screenshot below shows the relationship between the customer address table with the “Customer ID” field and the customer table with the “Customer ID” field. The cardinality has been set to one-to-one, and the direction of cross-filtering to two-way, because each customer has one residential address.



Below is the finished data model in star-like form:

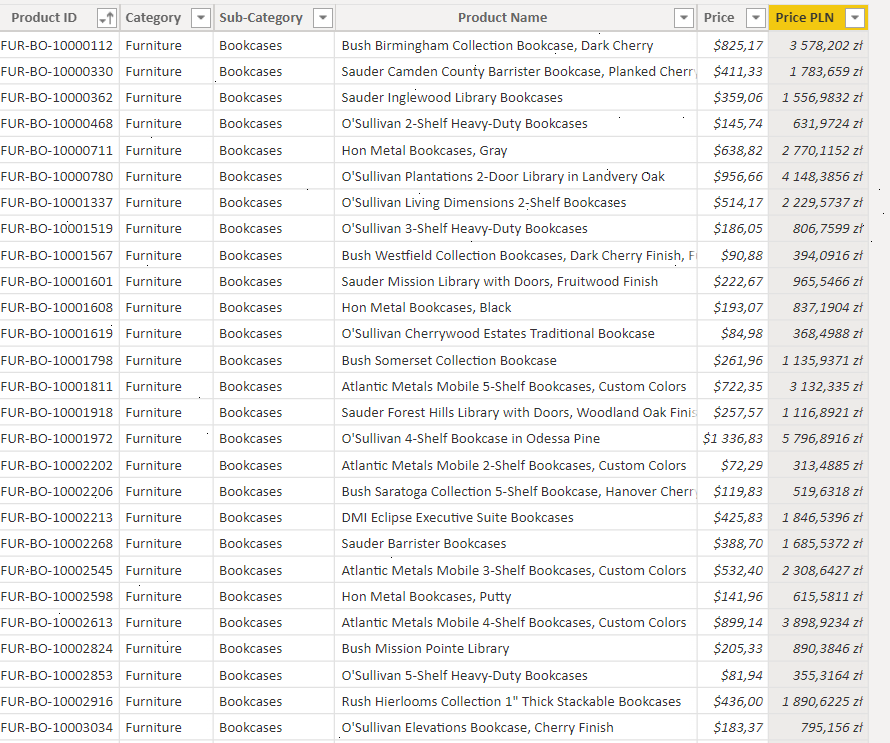


**Creation of dedicated columns and measures using the DAX language**

**The column with the converted price of the product from dollars to zlotys**

To convert the price from dollars to zlotys, the "LOOKUPVALUE" function was used, which allows you to refer to a column from another table not related to the current table. It returns a value for a row that meets all the criteria specified by one or more search conditions. The "\*" operator allows you to multiply the dollar exchange rate by the price of the product. Below is the formula for calculating the “Price PLN” column:





**The column with the converted price of the product from zlotys to euros**

The "LOOKUPVALUE" function and the "\*" operator were used to convert the price from PLN to EUR, as in the case above. Below is the formula to calculate the “Price EUR” column:





**A column showing the tax percentage by product category**

In the "Tax" column, the "SWITCH" function was used, which returns a different tax percentage for each product category, e.g. for furniture (Furniture) products, the tax value is 0.15. The column type has also been changed to percentage. The formula used is shown below:

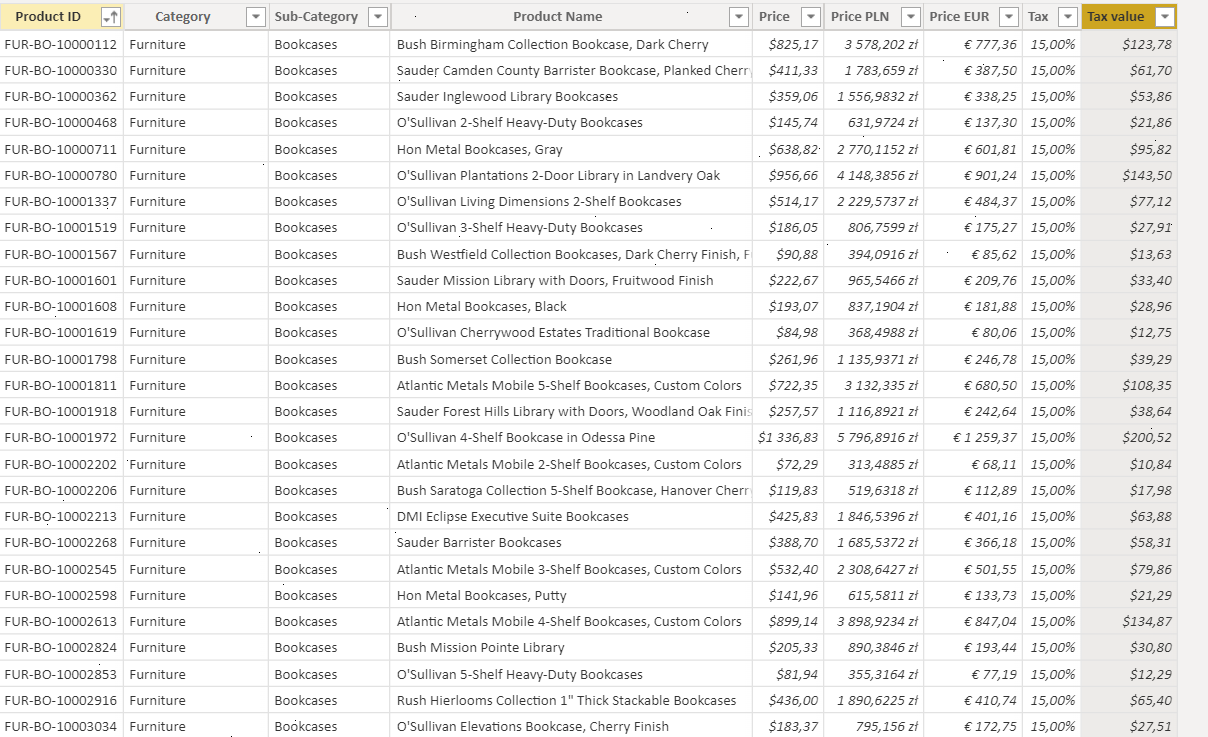
Trwa wstawianie obrazu...



**Column with the amount of tax paid in dollars**

The “Tax value” column shows the calculated tax value, taking into account the product price and the tax percentage.





**Column with showing the store's profit on a given product**

The “Price after tax” column shows the dollar value of the product after subtracting the tax from the original value of the product.

Trwa wstawianie obrazu...



**A column containing the delivery price for a given order**

The "Ship cost" column shows the value of shipping costs depending on the selected type of shipment. The "SWITCH" function was used to calculate this column, which returns the appropriate values ​​depending on the type of delivery. The formula used is shown below:

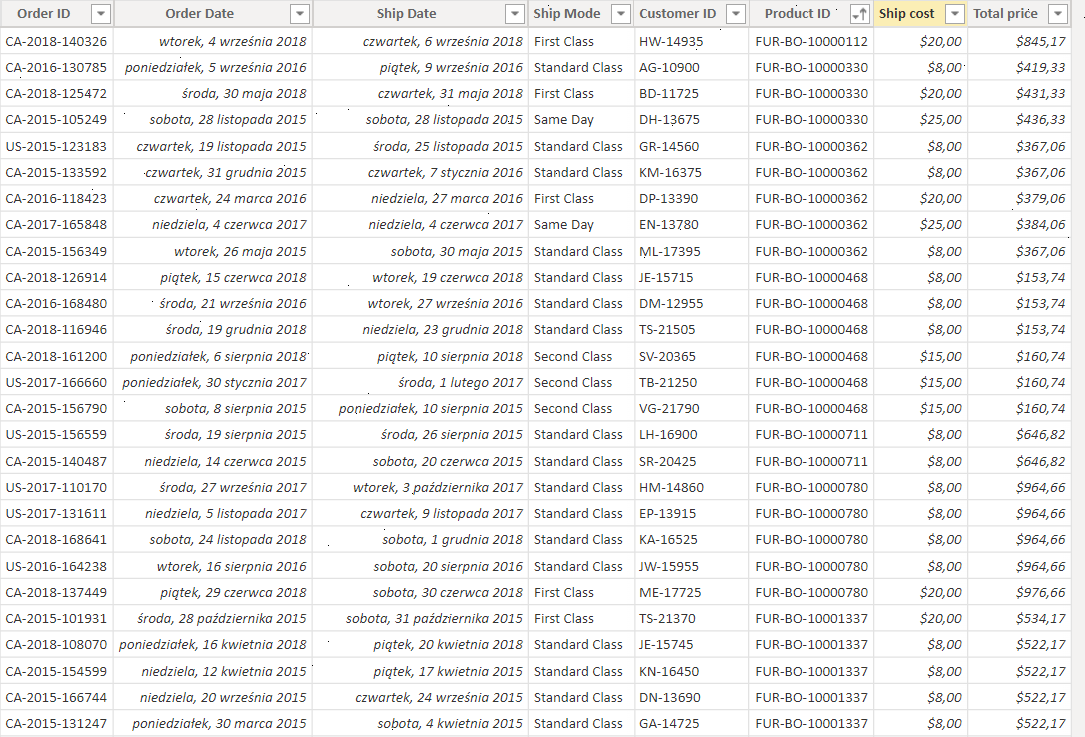




**The column containing the total price of the product including delivery**

The “Total price” column uses the “RELATED” function to fetch the price of the ordered product from the related “Products” table. The value returned by the column is the sum of the product price and the delivery price. This column has been placed in the table with orders, instead of the table with products, due to the fact that in the table with orders the same product may appear many times (in several orders), so there is no unique identifier that can be used to refer to this table. The following is the formula used for this column:





**A measure of the store's total profit**

Added measure "Total Profit", which shows the total sum of prices of all products sold. The "SUM" function was used here, which sums the values ​​from the column given as its argument:



**A measure of the previous day's revenue**

The "Total profit LD" measure has been added, which returns the sum of the prices of all products sold from the previous day. Here, the "Calculate" function is used, which calculates the expression in the modified filter context, and the "DATEADD" function, which returns the date to which the specified interval has been added (the "DAY" argument indicates that the interval is expressed in days, and "-1" that it has be the previous day). The formula used is shown below:



**A measure representing the previous month's revenue**

Added measure "Total Profit MTD" which returns a table containing a month-to-date column in the current context. The "Calculate" function is used here, which calculates an expression in a modified filter context, and the "DATESSMTD" function takes dates from the calendar as arguments.



**A measure that represents revenue for each year**

Added measure "Total Profit TotalYTD", calculates year to date expression in current context. The function sums up the obtained values ​​from the whole year and at the very end presents the value obtained after summing up the last year. The function uses the date column from the calendar file.



**Measure representing revenue from June to August 2018**

The measure "Total profit June-August 2018" has been added, which uses the previously described "CALCULATE" function, the "DATESBETWEEN" function, which selects dates from the calendar that fall within the range of dates given as arguments to this function. The DATE function returns a specified date in date (or time) format. Below is the formula used:



**The "Initials" column showing the client's initials**

The Initials column has been added, which searches the "Customer ID" table, returning the first letters of the name and surname, creating a column with the initials of the buyer. For this purpose, we use the MID function with the settings “StartPosition = 1” and “NumberOfCharacters = 2”.

Trwa wstawianie obrazu...



**Delivery time column**

Added column "Ship time" showing how many days the delivery took from the day of order to the day the package was delivered to the customer. We used the “DATEDIFF” function, which can calculate the time difference between two selected columns. In our case, we chose the interval as “day”.

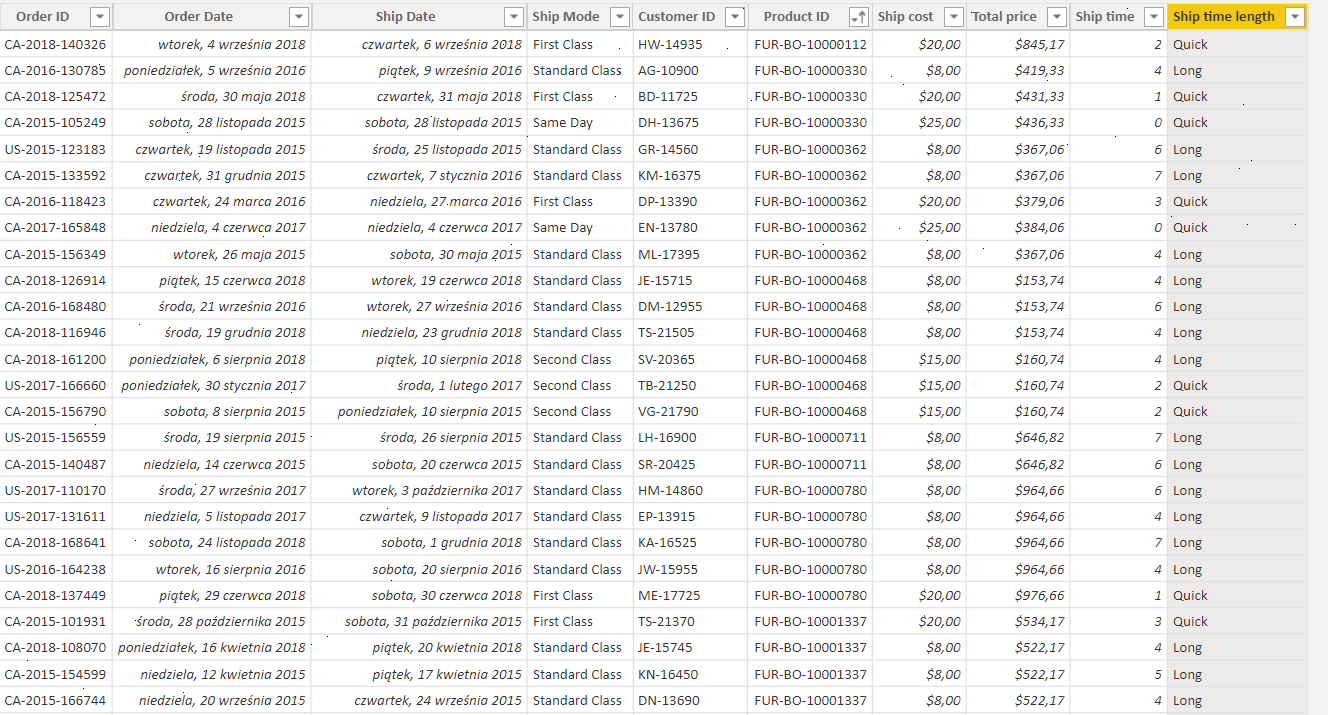




**A column specifying whether the delivery time is long**

Added column "Ship time length" showing if the delivery time was long or short. We used the “IF” function, taking the delivery time from the “Ship time” column and assessing whether the delivery time was more than 3 days or shorter.

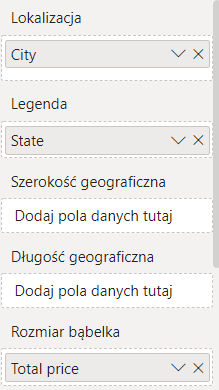
Trwa wstawianie obrazu...



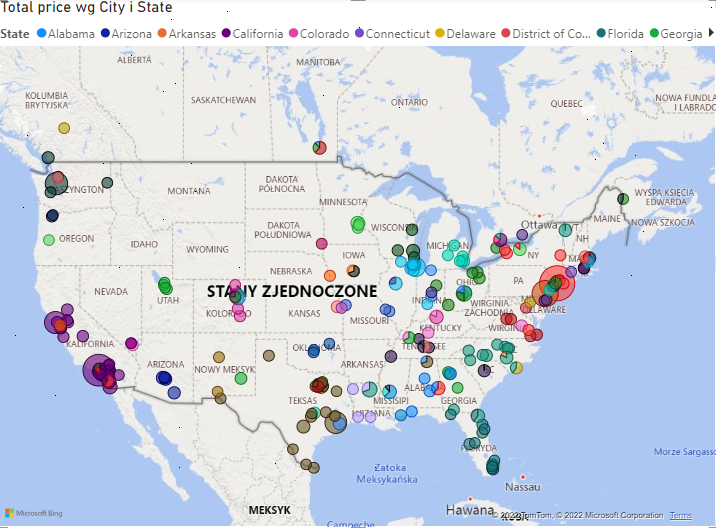
**Report 1**

**Map showing the volume of orders in individual cities**

A map has been added to the report and the following values ​​have been assigned to it:

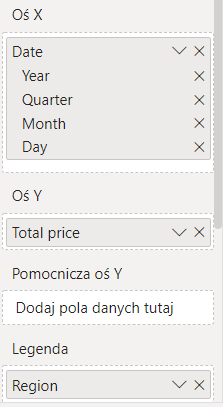


The map shows the value of orders in each city. The size of the bubble represents the total value of the order, and the color represents the status. The map shows that the highest value of orders is in the state of California:

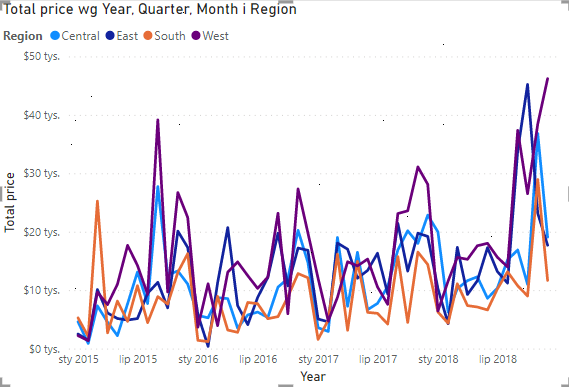


**Graph showing order volume over the years, broken down by region**

A graph has been added to the report and the following values ​​have been assigned to it:

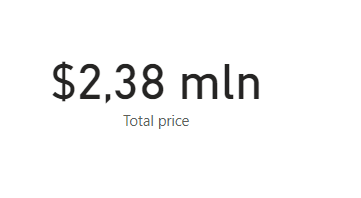


The graph shows the order volumes over the years from the beginning of 2015 to the end of 2018. The graph shows four lines representing the situation in different regions of the USA. The chart shows that the largest revenues come from the western region, and the period in which the revenues were the highest were the holiday months of 2018:



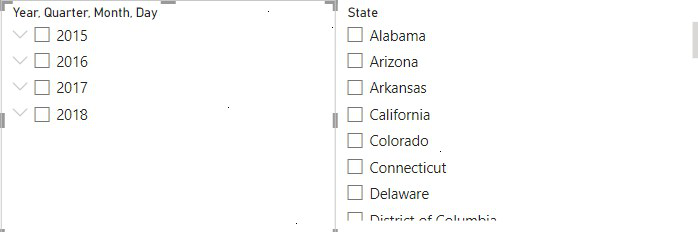
**Card**

A card showing the total value obtained from order fulfillment has been added to the report.

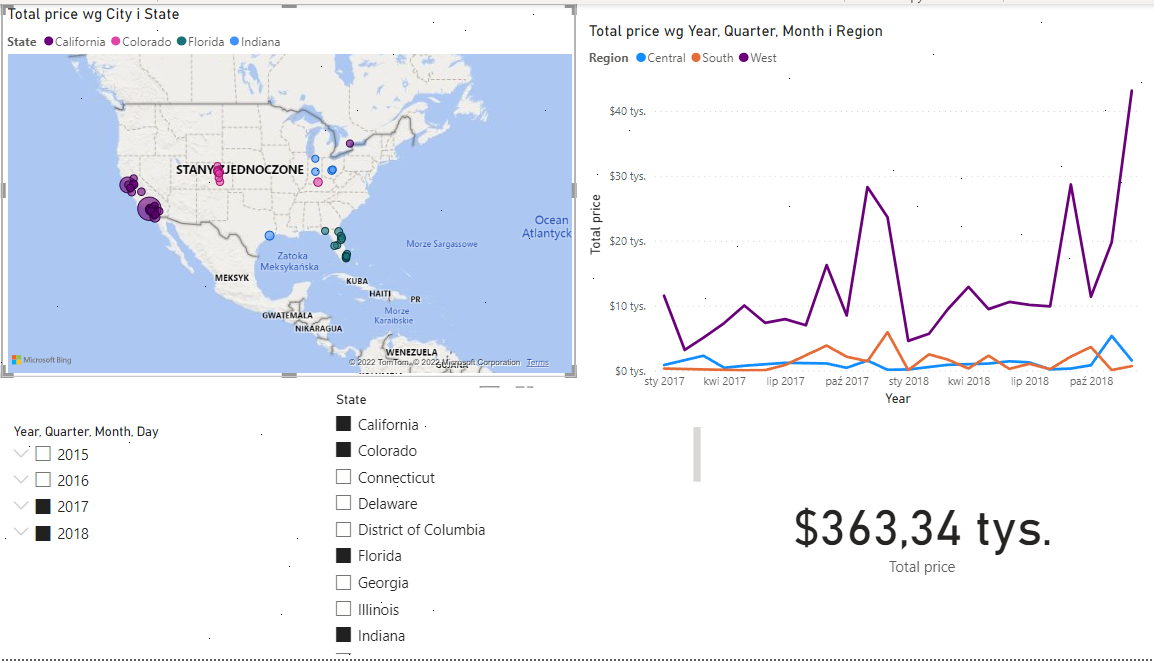


**Fragmetators**

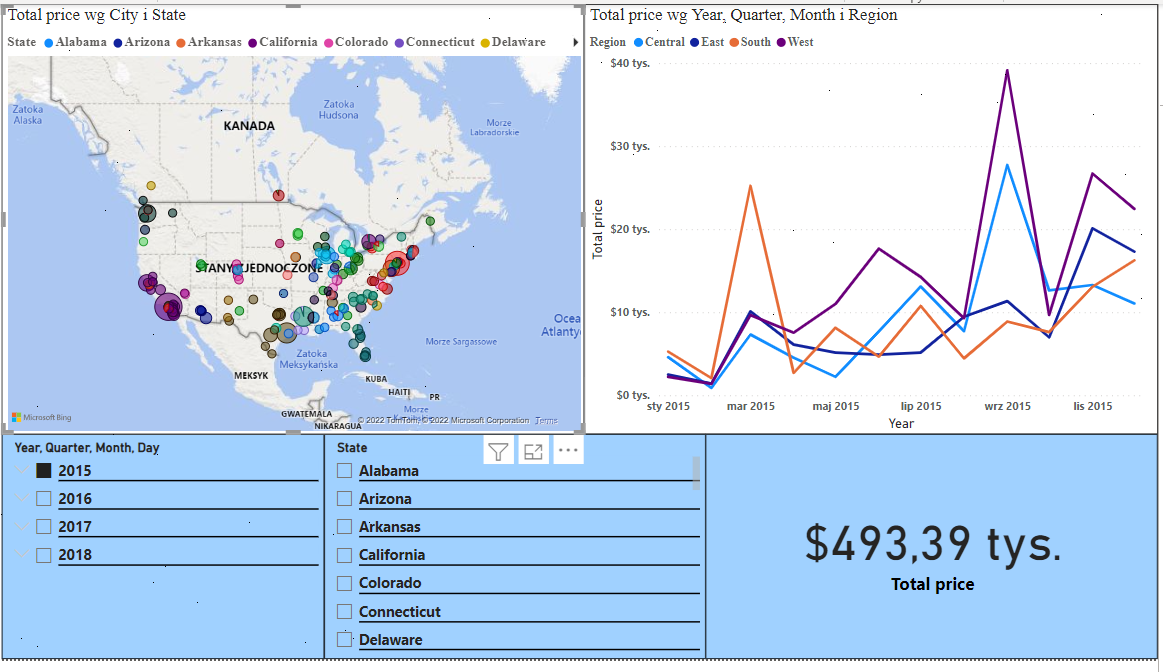
Two slicers have been added to the project. The hierarchical date division was dragged to the first one, and the "State" field from the "Customer Addresses" table to the second, which will allow us to display filtered data on visualizations:



Below are the visualizations, with slicer-filtered data, for the period from 2017 to 2018 for the states of California, Colorado, Florida and Indiana:



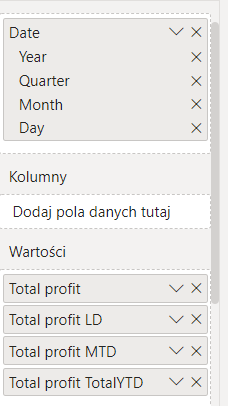
In the "Visualizations -> Format visual element" tab, the appearance of report elements was personalized by changing the background color, increasing the font size, bolding the font and adding a frame. This is what the report looks like:



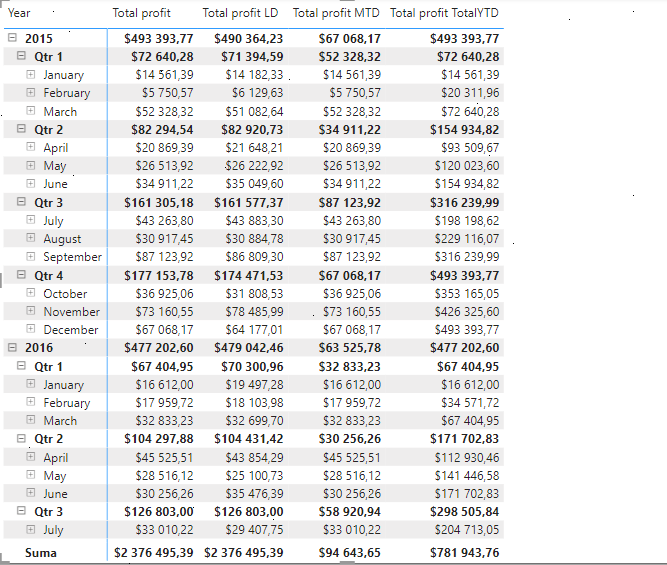
**Report 2**

**Matrix showing total revenue over time**

A matrix has been added to the report and the following values ​​have been assigned to it:

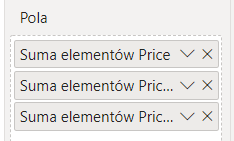


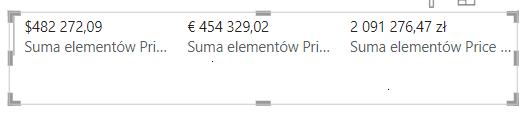
The matrix shows the total revenue over time, revenue from the previous day, month and year:



**Card with multiple lines**

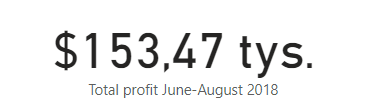
A card with multiple rows was added to the report, to which the following preset values ​​were assigned (the total was selected in the card settings):





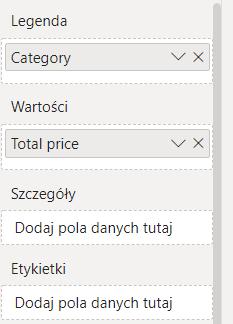
**Card**

A card has been added to the project showing the total profit for the period from June to August 2018



**Pie chart**

A pie chart has been added to the project, which shows the order values ​​broken down by the category of items sold. The following properties have been assigned to it:

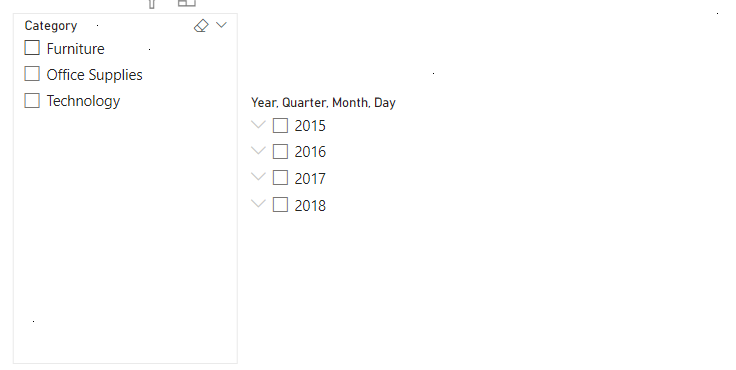


The chart shows that all three product categories bring comparable revenues to the company, but the technology category has a slight advantage in this aspect:

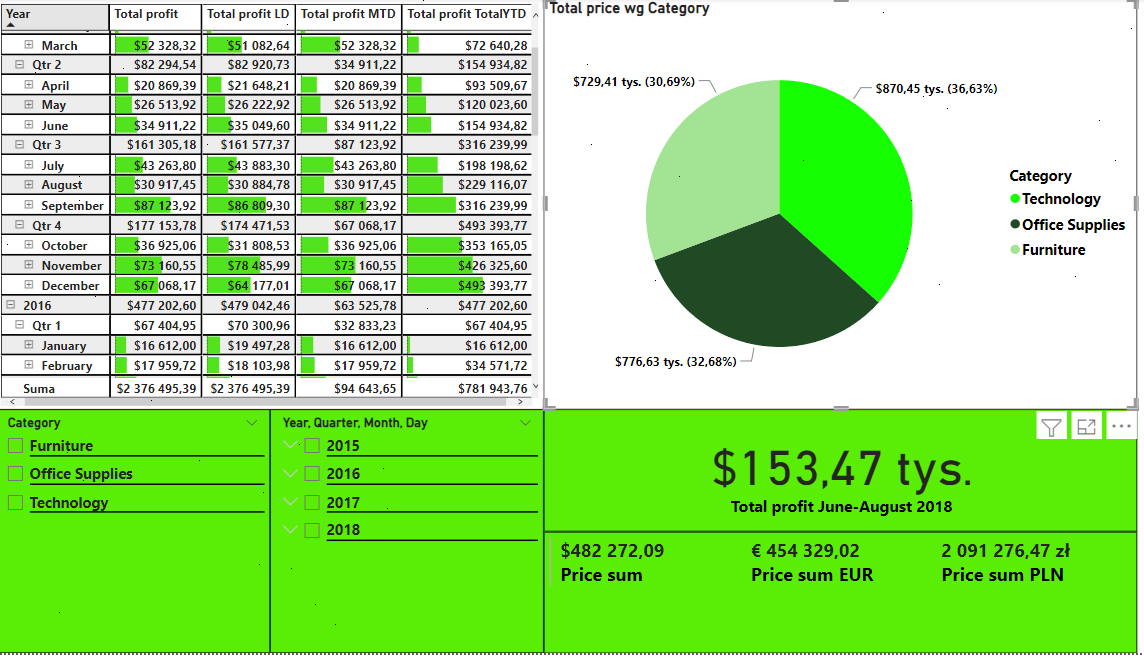


**Slicers**

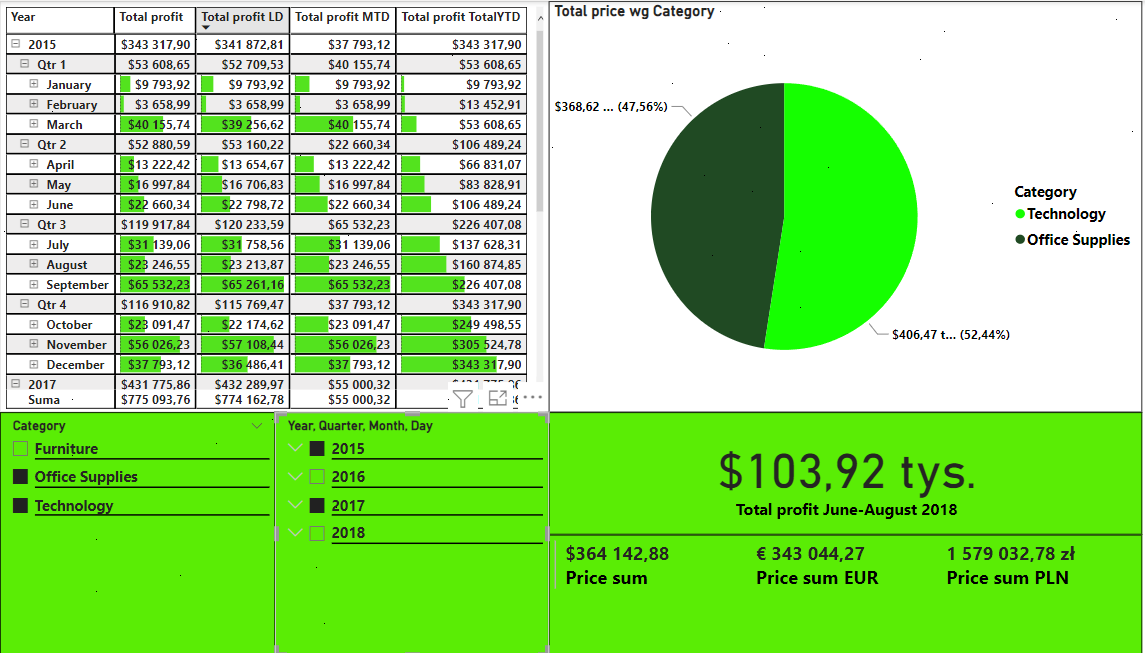
Two slicers have been added to the project. The "Category" field from the "Products" table was dragged to the first one, and the date every second, which will allow us to display filtered data on visualizations, according to a specific date and product category:



In the "Visualizations -> Format visual element" tab, the appearance of report elements has been changed again, by changing the background color, increasing the font size, bolding the font and adding a frame. Element names have also been changed for the tab. The background colors of the slices have been changed in the chart. This is what the report looks like:



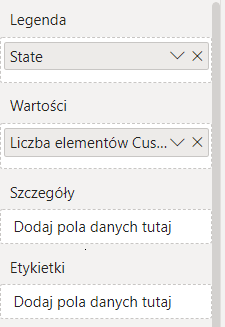
Below is a visualization using slicer filtering, as you can see in the June-August 2018 data card, after using the slicer, we still see the total profit for the mentioned period. The only change is that the category furniture is not included. The situation shows that changing the slicer's date does not affect the time functions, while category filtering does.



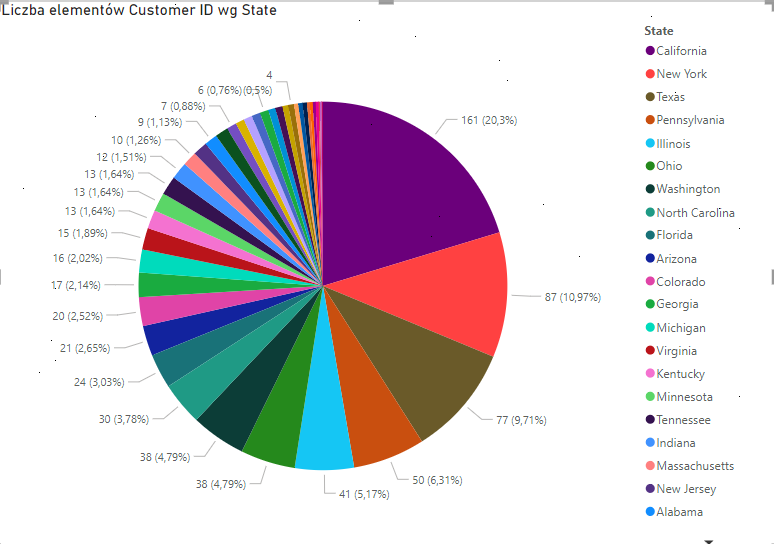
**Report 3**

**Pie chart**

A pie chart has been added to the project that shows the number of customers in a given state. The following properties have been assigned to it:

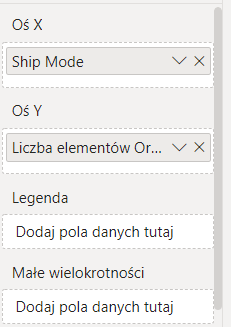


The graph shows in which region the company has the most customers and in which the least:



**A bar graph**

A column chart has been added to the project and the following values ​​have been assigned to it:

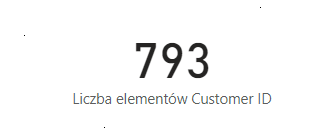


The chart shows which form of delivery is chosen most often (standard) and which is chosen least rarely (delivery on the same day):



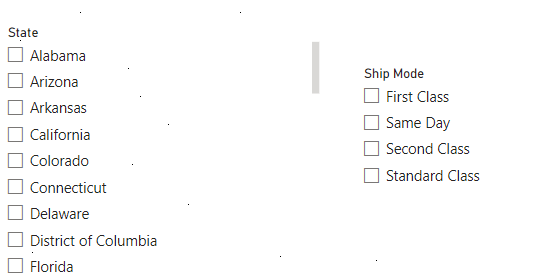
**Card**

A card has been added to the project, to which the "Customer ID" field from the "Customers" table has been added. The tab shows the total number of customers who made a purchase.

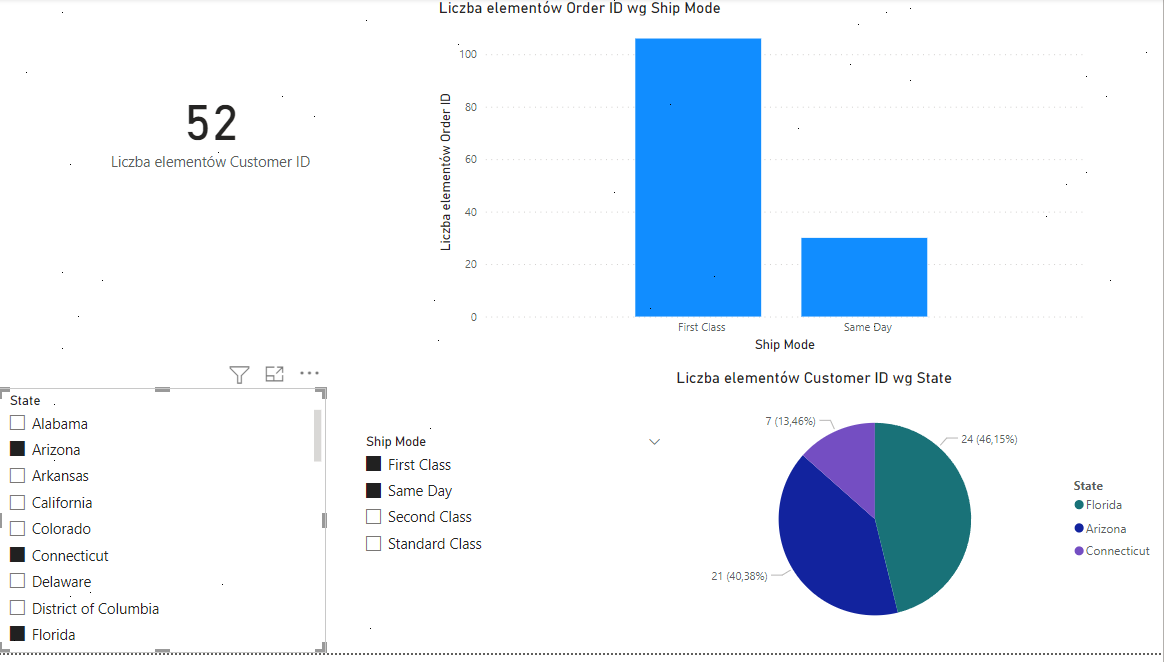


**Slicers**

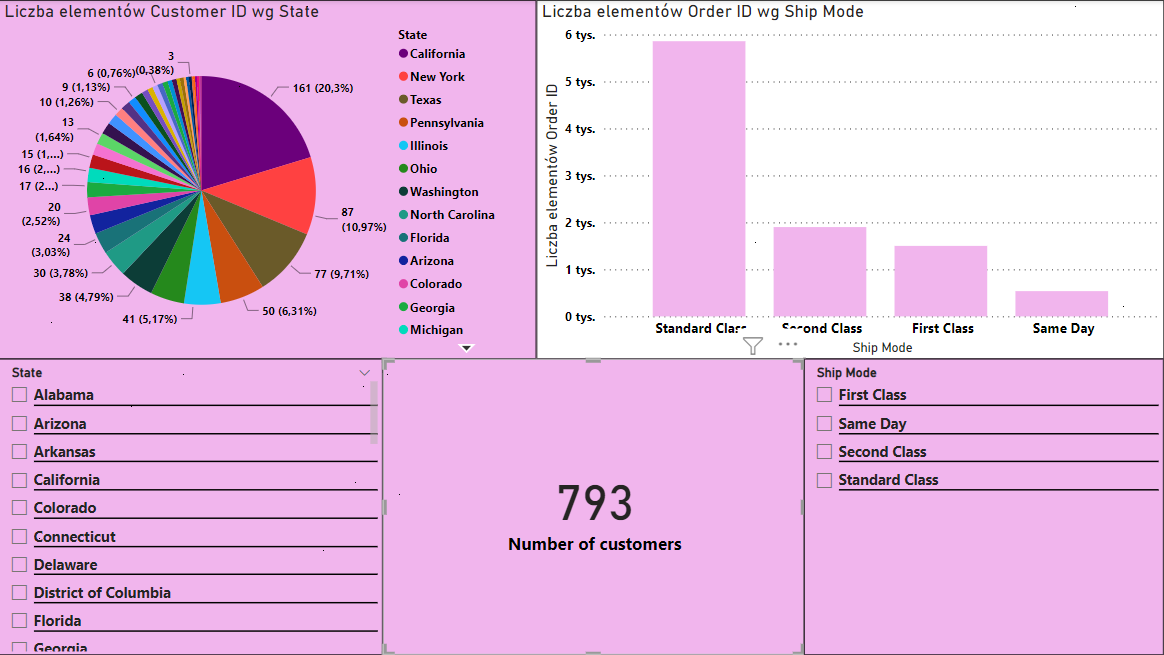
Two slicers have been added to the project: one filtering states in the USA and the other filtering delivery types, the addition of which will allow us to filter data on visualizations for selected attributes.



Below is a visualization using a slicer set to get results for Arizona, Connecticut, and Florida with the shipping type set to "First Class" and "Same Day".



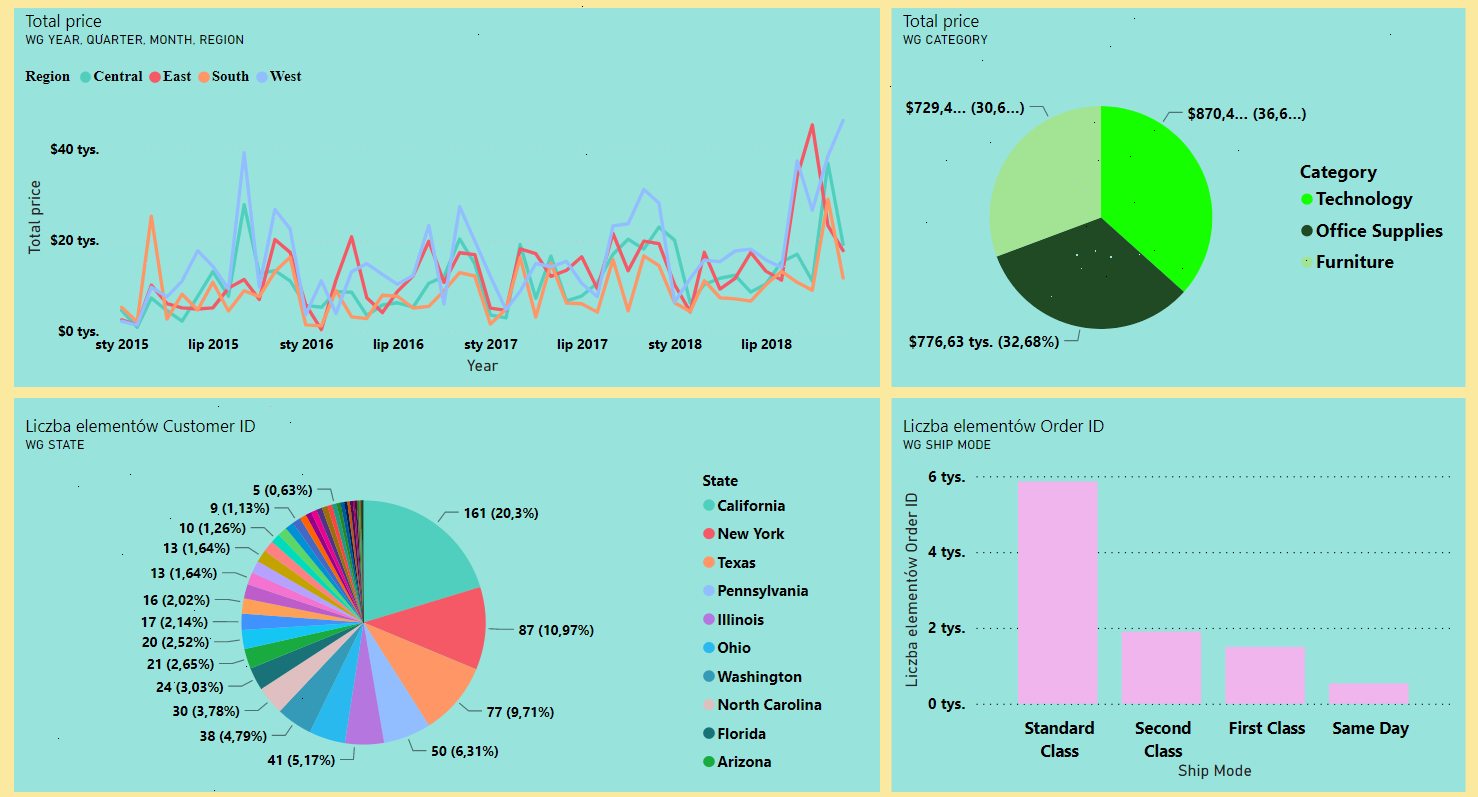
In the "Visualizations -> Format visual element" tab, the appearance of report elements has been changed once again, by changing the background color, increasing the font size, bolding the font and adding a frame. This is what the report looks like:

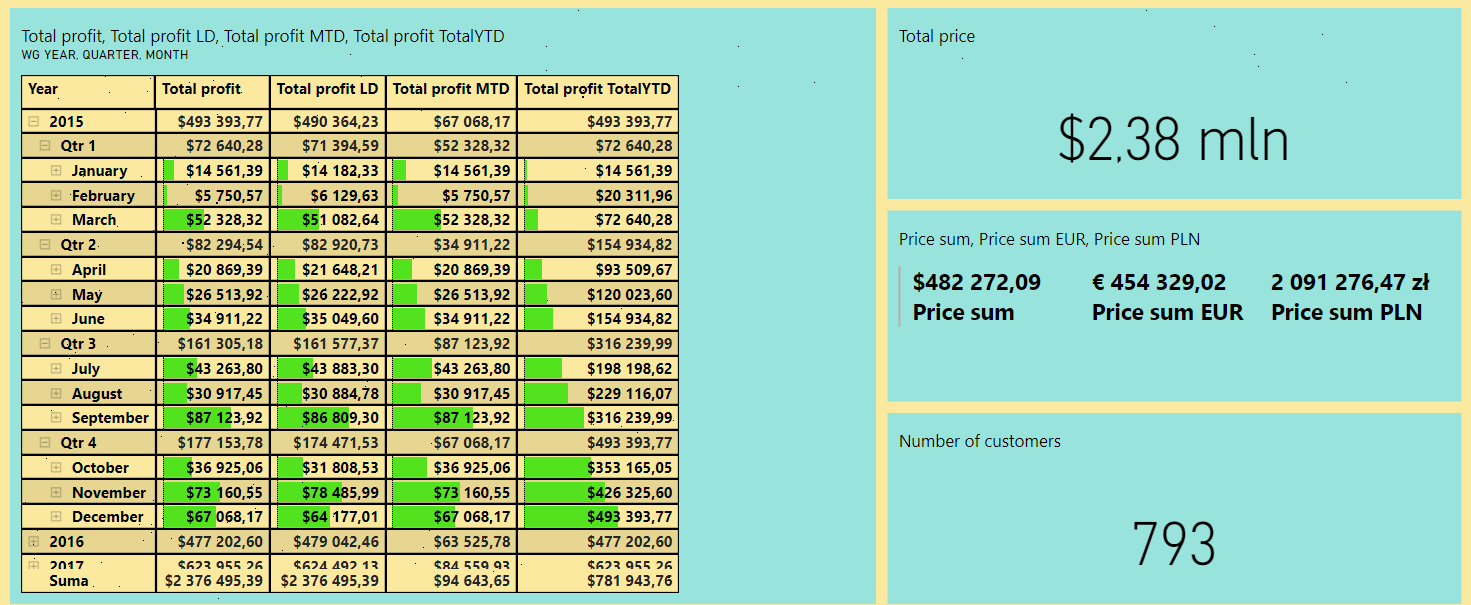


**Dashboards at powerbi.com**

The developed report was published and opened on powerbi.com. Then, by clicking on individual elements of the created report, the "Pin visual element" function was used. Using the "Edit -> Dashboard Theme" function, the appearance of the dashboard was changed (text, tiles and background color).

This is what the finished dashboard looks like:





This is how the finished dashboard looks like in the mobile view:

