

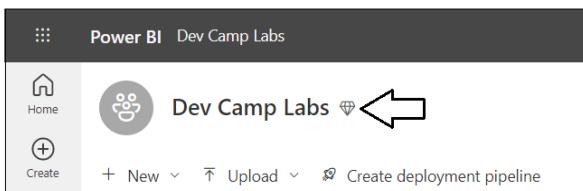
Hands-on Lab: Building Multi-language Reports for Power BI

Overview: In this lab, you will learn how to work with Power BI Desktop and [Translations Builder](#) to build and test a multi-language report for Power BI. You will start by downloading a PBIX project file named **Product Sales.pbix** and opening it in Power BI Desktop. Once you have opened the **Product Sales** project, you will launch Translations Builder and move through the steps to add metadata translations and report label translations. In later exercises you will also learn how to implement data translations. At various milestones in this lab, you will be required to publish the **Product Sales** project from Power BI Desktop to the Power BI Service so you can test out the translations you've added to the PBIX project to ensure they display correctly when loaded using different languages and locales.

Prerequisite 1: This lab assumes you're experienced with Power BI Desktop and you know how to build queries, datasets and reports.

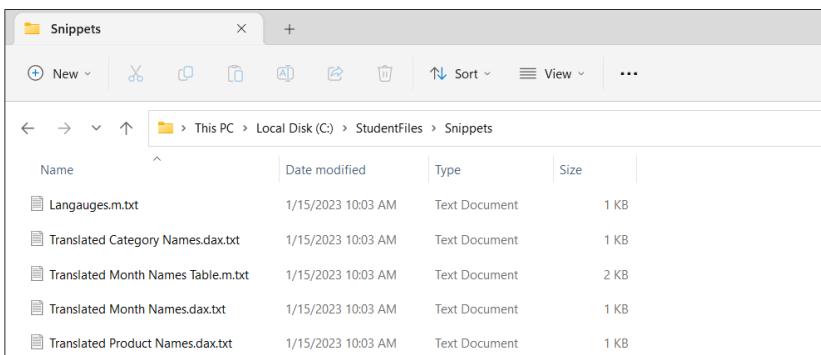
Prerequisite 2: This lab assumes you've already installed Translations Builder. If you haven't installed Translations Builder yet, you must follow steps in the [Translations Builder Installation Guide](#) before continuing with these lab exercises.

Prerequisite 3: To complete this lab, you will need a Power BI workspace where you have appropriate permissions to publish PBIX project files from Power BI Desktop to test your work. This workspace must also be associated with a Premium capacity as indicated by the diamond image (see *below*) displayed after the workspace name. A workspace in a premium capacity is required because Power BI translations do not load properly for reports in the shared capacity.



Student Files

These lab instructions are accompanied by a set of files in a ZIP archive named **StudentFiles.zip** that you can download from [HERE](#). The student files contain a completed solution for each of the six lab exercises in the form of a PBIX project file. There is also a **Snippets** folder with text files containing DAX code and M code that you will need to copy and paste in Power BI Desktop. While you can copy and paste DAX and M code from this manual, some students will find it easier to copy the code from these text files instead.



Lab Exercise Table of Contents

Hands-on Lab: Building Multi-language Reports for Power BI.....	1
Exercise 1: Creating and Testing Metadata Translations.....	2
Exercise 2: Generating Machine Translations	10
Exercise 3: Creating and Testing Report Label Translations	16
Exercise 4: Creating a Workflow Process To Gather & Integrate Human Translations	25
Exercise 5: Implementing Data Translations using Field Parameters	33
Exercise 6: Implementing Data Translations for a Calendar Table	60

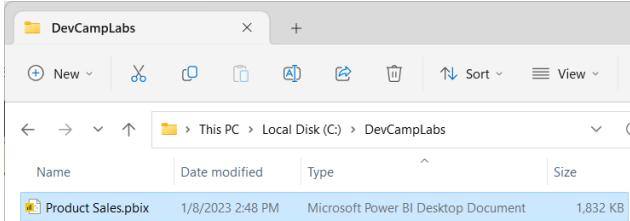
Exercise 1: Creating and Testing Metadata Translations

In this exercise, you will begin by downloading a PBIX file with a simple data model and a report with three pages. You will publish the project to the Power BI Service to set up a workflow process to test your work with translations in the browser. After that, you will use Translations Builder to add metadata translations for the column names and measure names in the dataset.

1. Download the PBIX starter file named **Product Sales.pbix** from the following link

<https://github.com/PowerBiDevCamp/TranslationsBuilder/raw/main/Labs/StarterFiles/Product%20Sales.pbix>

2. Create a new folder on your local hard drive for these lab exercises in a location such as **C:\DevCampLabs**.



3. Copy **Product Sales.pbix** into the lab folder and then open it in Power BI Desktop to examine the report inside.

Product	Image	Sales Revenue	Units Sold	Customer Count
Apples	Red Apple	€ 174,187.50	23,250	23,657
Oranges	Orange	€ 174,122.50	139,290	15,711
Potatoes	Yellow Potato	€ 167,820.00	134,256	14,895
Tomatoes	Red Tomato	€ 129,682.00	74,104	9,068
Milk	Carton	€ 117,585.00	47,034	6,259
Butter	Yellow Butter	€ 105,680.25	46,969	6,373
Cheese	Yellow Cheese	€ 97,567.50	26,018	3,892
Carrots	Carrot	€ 89,564.10	94,278	11,130
Cucumbers	Cucumber	€ 81,585.00	36,260	5,112
Bananas	Banana	€ 70,395.00	74,100	9,656
Total		€ 1,208,188.85	904,567	53,372

4. While in **Report** view, examine the **Fields** list to see the tables, columns and measure that are not hidden.

Table	Column	Type
Sales	Customer Count	Measure
	First Sale	Measure
	Last Sale	Measure
	Sales Revenue	Measure
	Units Sold	Measure
Calendar	Date	Column
	Day	Column
	Month	Column
	Year	Column
Products	Category	Column
	Image	Column
	Product	Column

5. Navigate to the second page named **Sales By Category** and examine its contents and layout.

Category	2020	2021	2022	Total
Vegetables	€ 47,500.55	€ 124,995.05	€ 296,155.50	€ 468,651.10
Potatoes	€ 14,331.25	€ 53,402.50	€ 100,086.25	€ 167,820.00
Tomatoes	€ 13,772.50	€ 49,148.75	€ 66,760.75	€ 129,682.00
Carrots	€ 11,807.55	€ 6,687.00	€ 71,069.50	€ 89,564.10
Cucumbers	€ 7,589.25	€ 15,756.75	€ 58,239.00	€ 81,585.00
Fruits	€ 72,535.00	€ 103,807.40	€ 242,544.60	€ 418,705.00
Apples	€ 28,600.50	€ 64,029.75	€ 81,557.25	€ 174,187.50
Oranges	€ 27,745.00	€ 26,181.25	€ 120,196.25	€ 174,122.50
Bananas	€ 16,007.50	€ 13,596.40	€ 40,791.10	€ 70,395.00
Dairy	€ 50,672.25	€ 94,161.50	€ 175,999.00	€ 320,832.75
Milk	€ 15,465.00	€ 35,855.00	€ 66,265.00	€ 117,585.00
Butter	€ 14,391.00	€ 31,520.25	€ 59,769.00	€ 105,680.25
Cheese	€ 20,816.25	€ 26,786.25	€ 49,965.00	€ 97,567.50
Total	€ 170,525.80	€ 322,963.95	€ 714,699.10	€ 1,208,188.85

Data translations for product names and category names will be addressed in **Exercise 5**.

6. Navigate to the third page named **Sales Over Time** and examine its contents and layout.

Product	January	February	March	April	May	June	July	August	September	October	November	December	Total
Apples	€ 11,751.00	€ 8,410.50	€ 12,003.75	€ 11,108.00	€ 14,901.00	€ 13,887.75	€ 15,162.75	€ 16,414.50	€ 15,209.25	€ 16,083.75	€ 20,175.75	€ 174,187.50	
Oranges	€ 5,346.25	€ 8,948.50	€ 10,611.25	€ 12,031.25	€ 15,517.50	€ 15,546.25	€ 14,900.00	€ 18,492.50	€ 18,056.25	€ 19,314.25	€ 17,450.00	€ 174,122.50	
Potatoes	€ 9,703.75	€ 7,207.50	€ 11,220.00	€ 10,930.00	€ 13,042.50	€ 13,612.50	€ 14,340.00	€ 15,985.00	€ 15,175.00	€ 17,004.25	€ 18,355.00	€ 20,670.00	€ 167,820.00
Tomatoes	€ 9,042.75	€ 10,461.00	€ 12,667.75	€ 12,903.25	€ 16,035.25	€ 20,020.00	€ 6,690.50	€ 19,160.00	€ 9,233.00	€ 9,981.25	€ 10,740.50	€ 12,890.25	€ 129,682.00
Milk	€ 6,345.00	€ 6,435.00	€ 6,070.00	€ 8,877.50	€ 12,490.00	€ 8,893.50	€ 5,959.00	€ 10,930.00	€ 9,472.50	€ 10,795.00	€ 11,177.50	€ 12,207.50	€ 117,585.00
Butter	€ 6,287.75	€ 6,672.50	€ 8,034.75	€ 8,032.50	€ 10,966.50	€ 7,839.00	€ 10,323.00	€ 9,036.00	€ 9,477.00	€ 9,965.25	€ 11,189.25	€ 105,680.25	
Cheese	€ 5,242.50	€ 5,115.00	€ 6,167.50	€ 10,683.75	€ 7,410.00	€ 7,782.50	€ 6,996.25	€ 7,871.25	€ 10,177.50	€ 8,970.00	€ 9,292.50	€ 97,567.50	
Carrots	€ 1,356.10	€ 1,576.05	€ 2,039.50	€ 2,039.00	€ 3,215.75	€ 3,768.85	€ 11,868.35	€ 10,629.00	€ 12,844.00	€ 11,340.15	€ 11,211.65	€ 89,564.10	
Cucumbers	€ 3,597.50	€ 5,064.75	€ 6,909.25	€ 6,959.25	€ 9,083.25	€ 6,039.00	€ 6,797.25	€ 7,452.00	€ 6,590.25	€ 7,571.50	€ 7,648.00	€ 8,160.50	€ 81,585.00
Bananas	€ 3,249.00	€ 3,386.75	€ 4,429.00	€ 4,886.75	€ 5,848.20	€ 7,240.90	€ 6,952.10	€ 6,730.05	€ 6,629.10	€ 7,530.05	€ 7,505.10	€ 70,395.00	
Total	€ 61,821.60	€ 61,826.75	€ 84,771.75	€ 88,854.30	€ 111,783.70	€ 96,904.50	€ 104,133.25	€ 116,858.50	€ 107,898.25	€ 121,933.30	€ 119,644.75	€ 131,732.25	€ 1,208,188.85

Data Translations for month names and day names will be addressed in **Exercise 6**.

7. Now, navigate to **Model** view so you can see the entire data model including the columns hidden from **Report** view.

8. Navigate to **Data view** and examine the rows of the **Products** table.

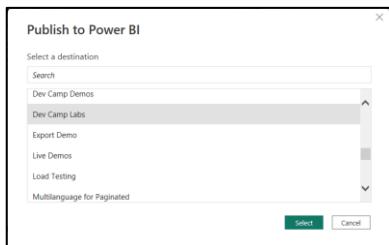
The screenshot shows the Power BI Data view interface. On the left, there's a list of tables: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, and Products. The 'Products' table is highlighted with a red box and an arrow pointing to it from the left. The main area displays the 'Products' table with columns: Image, Category, and Product. The data includes rows for Apples, Bananas, Oranges, Carrots, Potatoes, Tomatoes, Milk, Butter, and Cheese, each with a corresponding image icon.

Next, you are going to publish the **Product Sales** project to a Premium workspace in the Power BI Service.

9. Navigate to the **Home** tab and then click the **Publish** button.



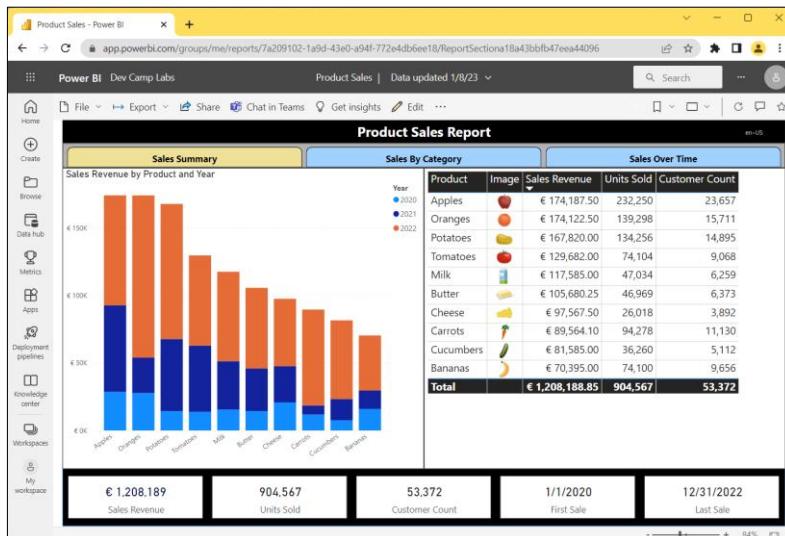
10. When prompted by the **Publish to Power BI** dialog, choose your test workspace and then click **Select**.



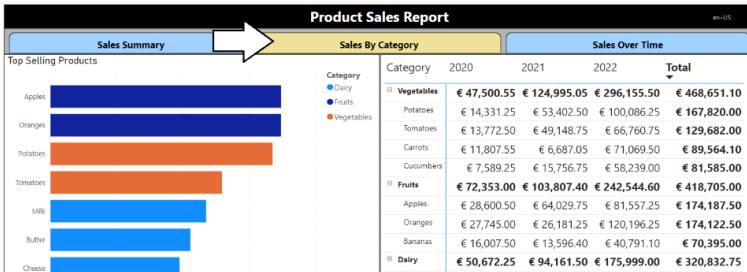
11. Once you see **Success!**, click **Open 'Product Sales.pbix'** in Power BI to view the report in the Power BI Service.



12. The report named **Product Sales** should appear like the report shown in the screenshot below.



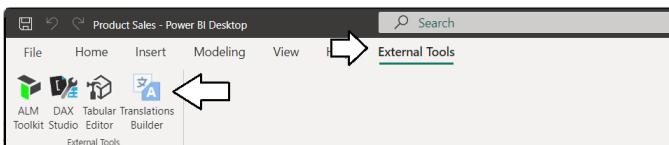
13. Navigate to the two other pages using the navigation buttons in the toolbar under the report tile..



Now it's time to begin adding translations. As you begin to add translations to a PBIX project, you will often follow this set of steps: **(1)** make changes in Power BI Desktop, **(2)** publish the project, **(3)** check your work in the Power BI Service, **(4)** repeat until happy

14. Return to Power BI Desktop.

15. Navigate to the **External Tools** tab and launch **Translations Builder**.



16. Translations Builder should start and load the data model for the **Product Sales** project.

17. The **Dataset properties** section on the left provides details about the dataset connection and the PBIX project file.

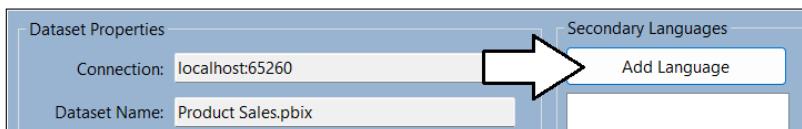
Every PBIX project file has a default language and a default locale. The **Product Sales** project has a default language of **English (en)** and a default locale of the **United States (US)**.

18. Examine the translation grid down below which displays a row for each non-hidden dataset object in the data model.

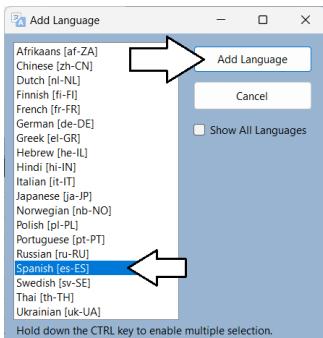
Object Type	Property	Name	English [en-US]
Table	Caption	Products	Products
Column	Caption	Products[Image]	Image
Column	Caption	Products[Category]	Category
Column	Caption	Products[Product]	Product
Table	Caption	Sales	Sales
Measure	Caption	Sales[Sales Revenue]	Sales Revenue
Measure	Caption	Sales[Units Sold]	Units Sold
Measure	Caption	Sales[Customer Count]	Customer Count
Measure	Caption	Sales[First Sale]	First Sale
Measure	Caption	Sales[Last Sale]	Last Sale
Table	Caption	Calendar	Calendar
Column	Caption	Calendar[Date]	Date
Column	Caption	Calendar[Year]	Year
Column	Caption	Calendar[Month]	Month
Column	Caption	Calendar[Day]	Day

Tables, columns and measures that are hidden from report view in the data model are not displayed. You don't need to translate them.

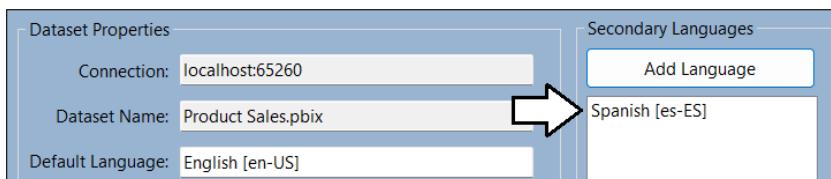
19. Click the **Add Language** button to add your first secondary language.



20. Select Spanish [es-ES] and click Add Language.



21. You should now see that Spanish [es-ES] appears as the first language in the Secondary Languages list.



22. You should also notice that a new column has been added to the translation grid for Spanish translations.

Object Type	Property	Name	English [en-US]	Spanish [es-ES]
Table	Caption	Products	Products	
Column	Caption	Products[Image]	Image	
Column	Caption	Products[Category]	Category	
Column	Caption	Products[Product]	Product	
Table	Caption	Sales	Sales	
Measure	Caption	Sales[Sales Revenue]	Sales Revenue	

23. In the row with the **Products** table, click on the cell for the **Spanish** column. It should turn blue when selected.

	Object Type	Property	Name	English [en-US]	Spanish [es-ES]
▶	Table	Caption	Products	Products	
	Column	Caption	Products[Image]	Image	
	Column	Caption	Products[Category]	Category	

24. Type **Hello World**. You should see that you can just start typing in the selected cell to add or edit a translation.

	Object Type	Property	Name	English [en-US]	Spanish [es-ES]
▶	Table	Caption	Products	Products	Hello World
	Column	Caption	Products[Image]	Image	
	Column	Caption	Products[Category]	Category	

25. Press the **ENTER** key to save your changes. Note that pressing **ENTER** will move the selection to the cell below.

	Object Type	Property	Name	English [en-US]	Spanish [es-ES]
	Table	Caption	Products	Products	Hello World
▶	Column	Caption	Products[Image]	Image	
	Column	Caption	Products[Category]	Category	

26. Now, type some more text and press **ENTER** repeatedly to quickly add text to each cell in the Spanish column.

	Object Type	Property	Name	English [en-US]	Spanish [es-ES]
	Table	Caption	Products	Products	Hello World
	Column	Caption	Products[Image]	Image	Hey
	Column	Caption	Products[Category]	Category	This
	Column	Caption	Products[Product]	Product	Is
	Table	Caption	Sales	Sales	Really
	Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Easy
▶	Measure	Caption	Sales[Units Sold]	Units Sold	
	Measure	Caption	Sales[Customer Count]	Customer Count	

The point of the last few steps has been for you to become comfortable with the translation editing experience. You can see the grid provides an editing experience similar to working with Excel. You can even use the **{F2}** key to move a cell with content into edit mode.

27. [OPTIONAL STEP] Add Spanish translations using the following table or translate them yourself if you know Spanish.

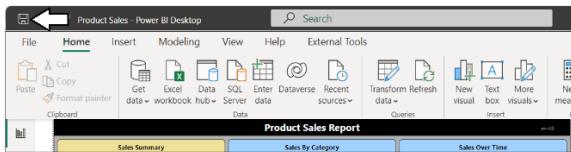
English	Spanish
Products	Productos
Image	Imagen
Category	Categoría
Product	Producto
Sales	Ventas
Sales Revenue	Ingresos Por Ventas
Units Sold	Unidades Vendidas
Customer Count	Número De Clientes
First Sale	Primera Venta
Last Sale	Última Venta
Calendar	Calendario
Date	Fecha
Year	Año
Month	Mes
Day	Día

It's OK if you don't want to type in all the Spanish translations from the table. Just add whatever text you'd like for each translation.

28. When you are done with your edits, the Spanish translations should match the following screenshot (or not).

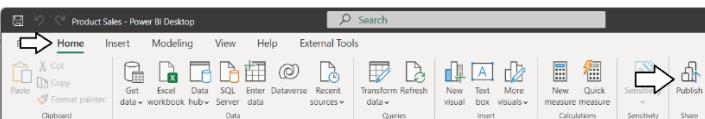
Object Type	Property	Name	English [en-US]	Spanish [es-ES]
Table	Caption	Products	Products	Productos
Column	Caption	Products[Image]	Image	Imagen
Column	Caption	Products[Category]	Category	Categoría
Column	Caption	Products[Product]	Product	Producto
Table	Caption	Sales	Sales	Ventas
Measure	Caption	Sales(Sales Revenue)	Sales Revenue	Ingresos Por Ventas
Measure	Caption	Sales(Units Sold)	Units Sold	Unidades Vendidas
Measure	Caption	Sales(Customer Count)	Customer Count	Número De Clientes
Measure	Caption	Sales(First Sale)	First Sale	Primerá Venta
Measure	Caption	Sales(Last Sale)	Last Sale	Última Venta
Table	Caption	Calendar	Calendar	Calendario
Column	Caption	Calendar[Date]	Date	Fecha
Column	Caption	Calendar[Year]	Year	Año
Column	Caption	Calendar[Month]	Month	Mes
Column	Caption	Calendar[Day]	Day	Día

29. Return to the **Product Sales** project in Power BI Desktop and save your work by clicking the **Save** button.



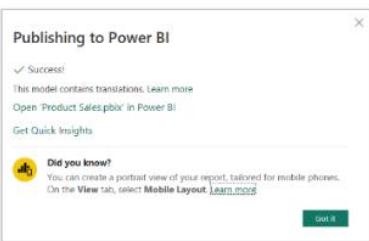
It's easy to forget to save your changes in Power BI Desktop. Be aware that any changes made by Translations Builder are just made to the data model loaded in memory. None of your changes are saved back to the PBIX project file until you save in Power BI Desktop.

30. Publish the **Product Sales** project to push the changes to the project's translations to the Power BI Service.

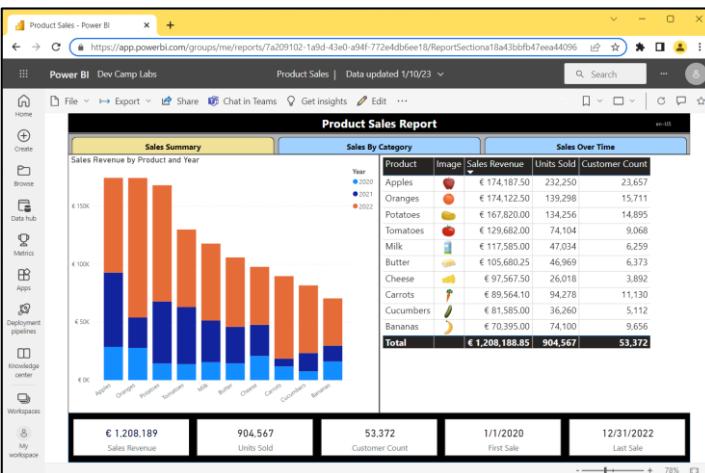


31. When prompted by the **Replace this dataset?** Dialog, click the **Replace** button to continue.

32. Once you see **Success!**, click **Open 'Product Sales.pbix'** in Power BI to view the report in the Power BI Service.



33. The report should load with its default behavior showing all text in English at first.

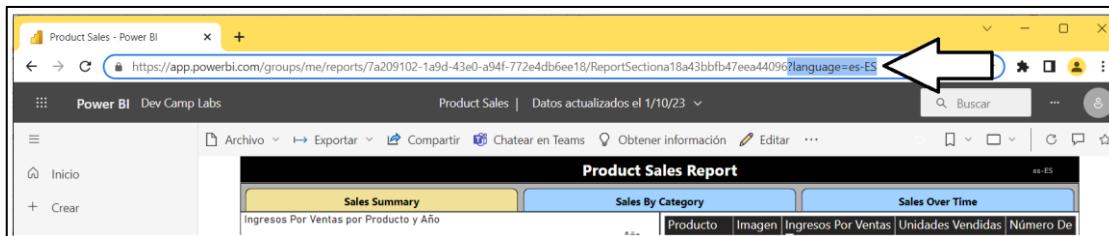


Now, it's time to test your translations. You will accomplish this by using the **language** query string parameter to load the report.

34. Click the browser address bar and add the following **language** parameter to the end of the report URL and press **ENTER**.

`?language=es-ES`

35. When you press **ENTER**, you should see the **language** parameter accepted by the browser as it reloads the report.



When the report reloads, you should see the UI experience for the entire Power BI Service UI switch from English to Spanish.

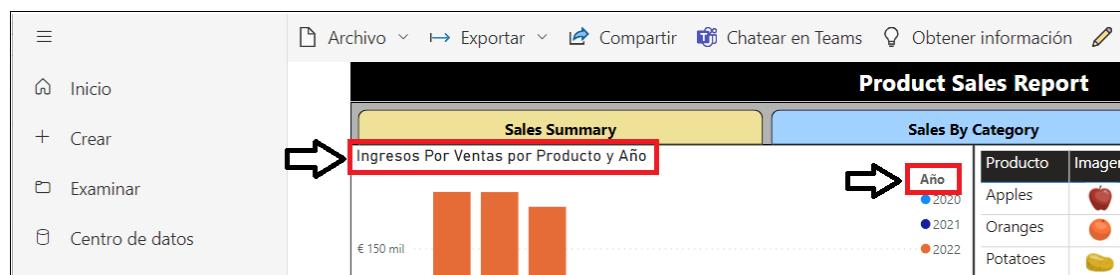
36. Inspect the Card visuals at the bottom of the **Sales Summary** page. Verify the column and measure names are displayed with the Spanish translations you added earlier in this exercise.

€ 1.208.189	904.567	53.372	01/01/2020	31/12/2022
Ingresos Por Ventas	Unidades Vendidas	Número De Clientes	Primera Venta	Última Venta

37. Inspect the column headers in the Table visual. Verify the column and measure names are also displayed with the Spanish translations you added.

Product Sales Report				
Sales Summary		Sales By Category		
Ingresos Por Ventas por Producto y Año		Producto	Imagen	Ingresos Por Ventas
€ 150 mil	2020	Apples		€ 174.187,50
	2021	Oranges		€ 174.122,50
	2022	Potatoes		139.298
				23.657
				15.711

38. Inspect the Column Chart visual. Verify the title and legend title are displayed with the Spanish translations.



39. Navigate to the **Sales By Category** page.

40. You should see the title of the Bar Chart visual is not translated because the text for the title is hard-coded in the report.



You have now successfully added the metadata translations to display this report in both English and Spanish. Leave Power BI Desktop and Translations Builder open as you will continue using them in your next exercise.

Exercise 2: Generating Machine Translations

In this lab you will configure Translations Builder to support generating machine translations using the Azure Translator Service. While machine translations might not provide the level of quality required for some production scenarios, they do provide a great step forward in generating the first round translations which can be used for testing and for getting *something* into production sooner.

To complete this lab you will require a **Key** and **Location** which provide access to the **Azure Translator Service**. If you have an Azure subscription, you can learn how to obtain this key and its location by reading [Obtaining a Key for the Azure Translator Service](#). If you do not have a Azure subscription or you cannot obtain a key of your own, you can use the following **Key** and **Location** value when working on these lab exercises up through February 28, 2023.

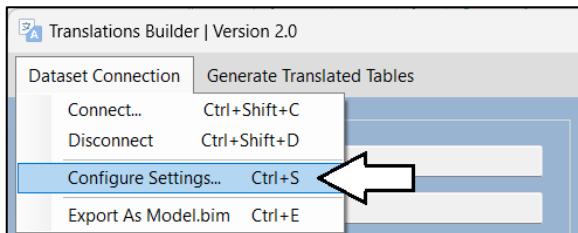
Key: **7bf614164d924cd9a9ba3d9cf351d15f**

Location: **eastus2**

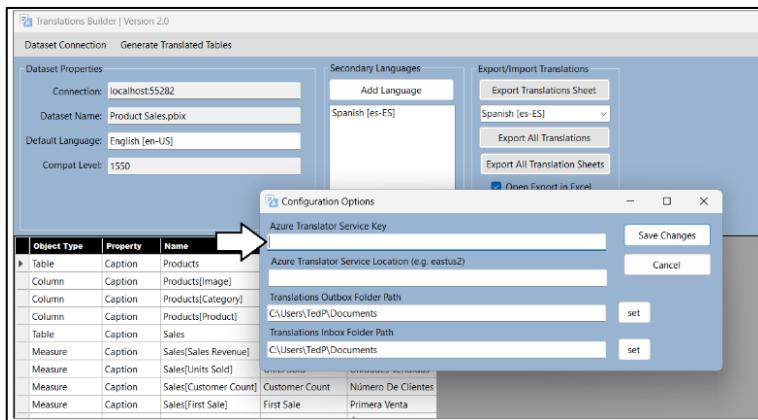
1. Return to Translations Builder and drop down the **Dataset Connection** menu.



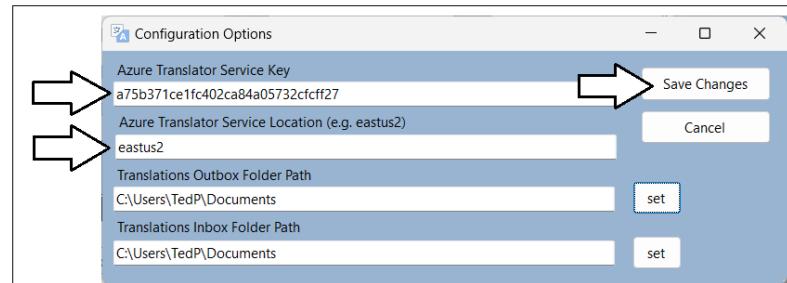
2. Select the **Configure Settings...** menu command to display the **Configuration Options** dialog.



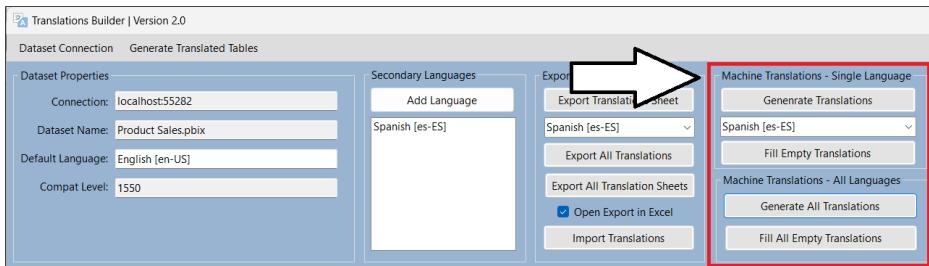
3. In the **Configuration Options** dialog, enter the **Key** and **Location** for the Azure Translator Service.



4. Once you have added the **Key** and **Location**, click **Save Changes**.

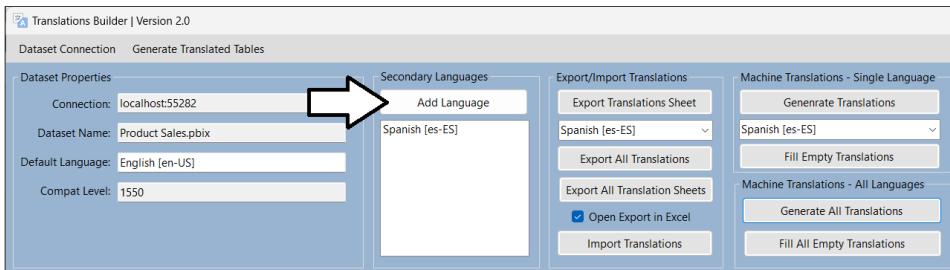


5. After you have configured the **Key** and **Location** for the Azure Translator Service, new command will appear on the main window.



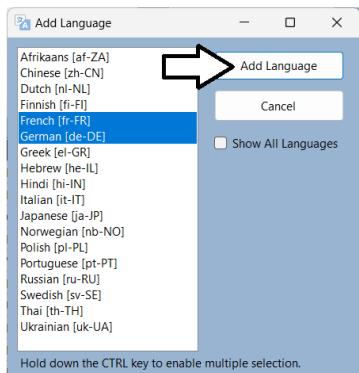
Now that you have added support for generating machine translations, it's time to put that automatic translation support to work!

6. Click the **Add Language** button to add your second secondary language.



You can hold down the **CTRL** key in the **Add Language** dialog while selecting languages to enable multiple selection

7. Hold down the **CTRL**, and select **French** and **German**. And then click **Add Language**.

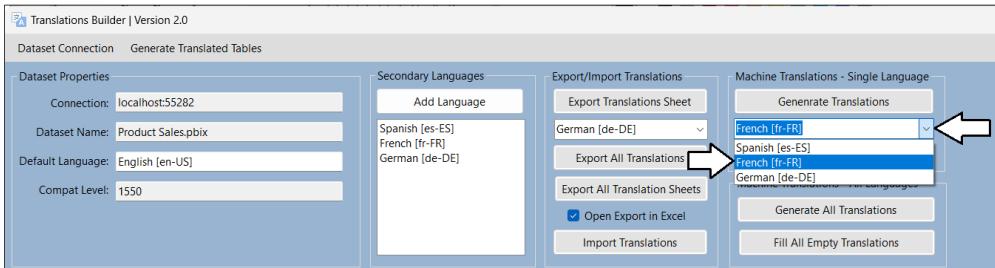


8. You should now see the two new languages appear in the **Secondary Languages** list.

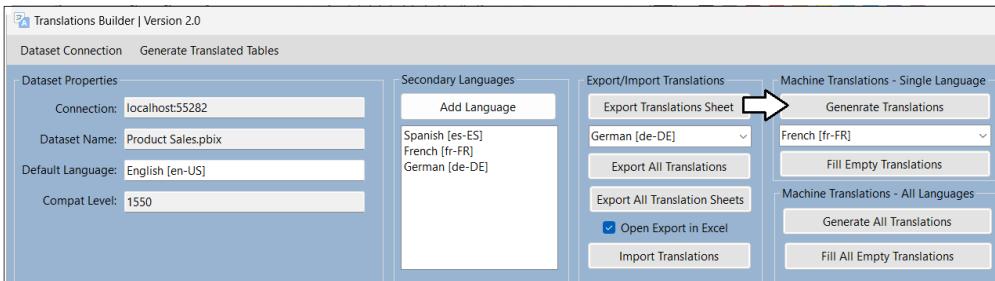
9. You will also notice that new columns have been added to the translation grid for each new language.

Object Type	Property	Name	English [en-US]	Spanish [es-ES]	French [fr-FR]	German [de-DE]
Table	Caption	Products	Products	Productos		
Column	Caption	Products[Image]	Image	Imagen		
Column	Caption	Products[Category]	Category	Categoría		
Column	Caption	Products[Product]	Product	Producto		

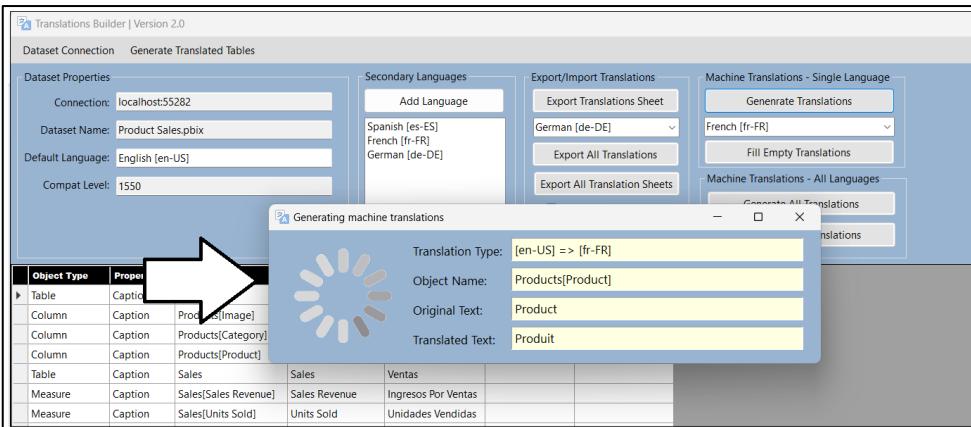
10. In the **Machine Translations – Single Language** section, select **French [fr-FR]** from the drop down menu.



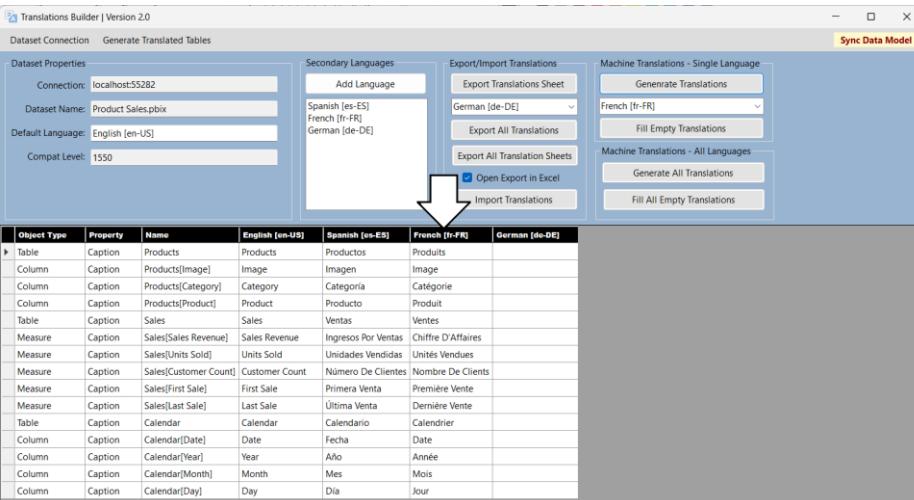
11. Once you have selected **French [fr-FR]**, click **Generate Translations** to create French translations for all rows in the grid.



12. As the code runs to interact with the Azure Translator Service, the **Generating machine translation** dialog shows the progress.

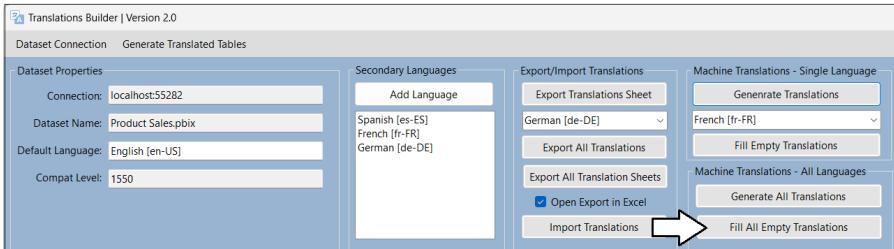


13. Once this dialog closes, you should see all cells in the French column have been filled with machine-generated translations.



If your Spanish translations are less than perfect, you can use the same technique to replace what you have with machine translations.

14. Click the **Fill All Empty Translation** button in the **Machine Translations - All Languages** section.



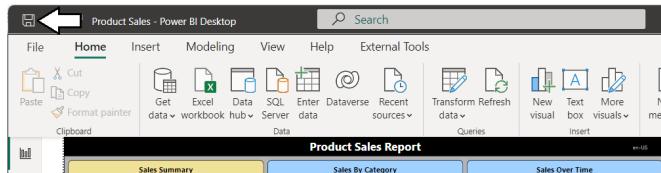
In this scenario, you want to use the **Fill All Empty Translation** command generate machine translations just for translations that are empty. Use the **Generate All Translations** command to replace all existing translations with newly-generated machine translations.

15. You should now see that the empty cells for all secondary languages have been populated with machine-generated translations.

Object Type	Property	Name	English [en-US]	Spanish [es-ES]	French [fr-FR]	German [de-DE]
Table	Caption	Products	Products	Productos	Produits	Produkte
Column	Caption	Products[Image]	Image	Imagen	Image	Bild
Column	Caption	Products[Category]	Category	Categoría	Catégorie	Kategorie
Column	Caption	Products[Product]	Product	Producto	Produit	Produkt
Table	Caption	Sales	Sales	Ventas	Ventes	Umsatz
Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas	Chiffre D'Affaires	Umsatz
Measure	Caption	Sales[Units Sold]	Units Sold	Unidades Vendidas	Unités Vendues	Verkaufte Einheiten
Measure	Caption	Sales[Customer Count]	Customer Count	Número De Clientes	Nombre De Clients	Anzahl Der Kunden
Measure	Caption	Sales[First Sale]	First Sale	Primerá Venta	Première Vente	Erster Verkauf
Measure	Caption	Sales[Last Sale]	Last Sale	Última Venta	Dernière Vente	Letzter Verkauf
Table	Caption	Calendar	Calendar	Calendario	Calendrier	Kalender
Column	Caption	Calendar[Date]	Date	Fecha	Date	Datum
Column	Caption	Calendar[Year]	Year	Año	Année	Jahr
Column	Caption	Calendar[Month]	Month	Mes	Mois	Monat
Column	Caption	Calendar[Day]	Day	Día	Jour	Tag

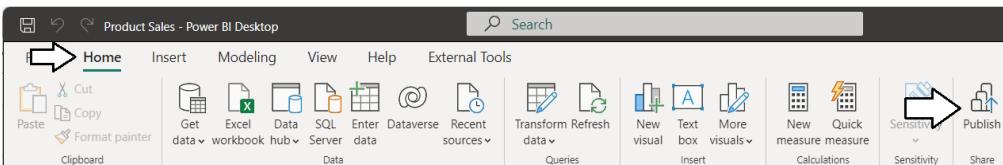
Now, it's time once again to test your work in the Power BI Service,

16. Return to the **Product Sales** project in Power BI Desktop and save your work by clicking the **Save** button.



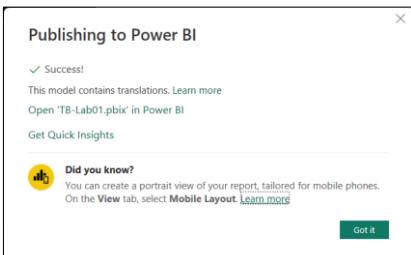
Don't forget to save your work! Did we mention it's easy to forget which can inadvertently lead to you losing your work.

17. Publish the **Product Sales project** to push the changes to the project's translations to the Power BI Service.



18. When prompted by the **Replace this dataset?** Dialog, click the **Replace** button to continue.

19. Once you see **Success!**, click **Open 'Product Sales.pbix'** in Power BI to view the report in the Power BI Service.



20. The report should load as normal showing all text in English at first.

Product	Image	Sales Revenue	Units Sold	Customer Count
Apples	🍎	€ 174,187,50	232,250	23,657
Oranges	🍊	€ 174,122,50	139,298	15,711
Potatoes	🥔	€ 167,820,00	134,256	14,895
Tomatoes	🍅	€ 129,682,00	74,104	9,068
Milk	🥛	€ 117,585,00	47,034	6,259
Butter	🧈	€ 105,680,25	46,999	6,373
Cheese	🧀	€ 97,567,50	26,018	3,892
Carrots	🥕	€ 89,564,10	94,278	11,130
Cucumbers	🥒	€ 81,585,00	36,260	5,112
Bananas	🍌	€ 70,395,00	74,100	9,656
Total		€ 1,208,188,85	904,567	53,372

€ 1,208,189 Sales Revenue
904,567 Units Sold
53,372 Customer Count
1/1/2020 First Sale
12/31/2022 Last Sale

Now, it's time to test your French and German translations using the **language** query string parameter to load the report.

21. Click the browser address bar and add the **language** parameter value of **fr-FR** for French to the end of the report URL.

?language=fr-FR

22. When the report reloads, you should see the UI experience for the Power BI Service UI switch from English to French.

23. Verify the column and measure names used in all three visuals are displayed with French translations.

Product	Image	Chiffre D'Affaires	Unités Vendues	Nombre De Clients
Apples	🍎	€ 174,187,50	232,250	23,657
Oranges	🍊	€ 174,122,50	139,298	15,711
Potatoes	🥔	€ 167,820,00	134,256	14,895
Tomatoes	🍅	€ 129,682,00	74,104	9,068
Milk	🥛	€ 117,585,00	47,034	6,259
Butter	🧈	€ 105,680,25	46,999	6,373
Cheese	🧀	€ 97,567,50	26,018	3,892
Carrots	🥕	€ 89,564,10	94,278	11,130
Cucumbers	🥒	€ 81,585,00	36,260	5,112
Bananas	🍌	€ 70,395,00	74,100	9,656
Total		€ 1,208,188,85	904,567	53,372

€ 1 208 189 Chiffre D'Affaires
904 567 Unités Vendues
53 372 Nombre De Clients
01/01/2020 Première Vente
31/12/2022 Dernière Vente

Now that you have tested the French translations, it's time to test German.

24. Click the browser address bar and add the **language** parameter value of **de-DE** for German to the end of the report URL.

?language=de-DE

25. When the report reloads, you should see the UI experience for the Power BI Service UI switch to German.

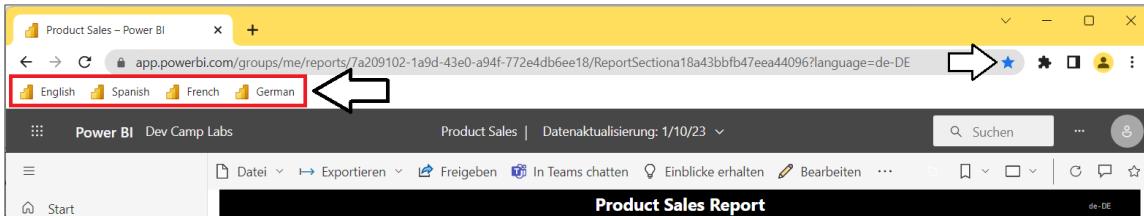
26. Verify the column and measure names in columns headings are displayed with the German translations.



Take a moment to appreciate how the date and numeric formatting automatically changes across all four regional locales.

en-US	€ 1.208.189 Sales Revenue	904.567 Units Sold	53.372 Customer Count	1/1/2020 First Sale	12/31/2022 Last Sale
es-ES	€ 1.208.189 Ingresos Por Ventas	904.567 Unidades Vendidas	53.372 Número De Clientes	01/01/2020 Primera Venta	31/12/2022 Última Venta
fr-FR	€ 1 208 189 Chiffre D'Affaires	904 567 Unités Vendues	53 372 Nombre De Clients	01/01/2020 Première Vente	31/12/2022 Dernière Vente
de-DE	€ 1.208.189 Umsatz	904.567 Verkaufte Einheiten	53.372 Anzahl Der Kunden	01.01.2020 Erster Verkauf	31.12.2022 Letzter Verkauf

27. As a final step in this exercise, add a browser bookmark for each language with a **language** parameter at the end.



Creating a browser bookmark for each language might take a minute or two to set up at first. However, it will save lots of time in the long run as you continue to test the translations you create for this report in the lab work that remains ahead.

In the previous step you created bookmark in the browser to easily test the report using different languages. Later in Exercise 5, you will create bookmarks inside the context of the report in the **Product Sales** project. To avoid any potential confusions, these lab exercises will use the term **browser bookmark** when referring to the type of bookmark you have just created in the browser. These lab exercises will use the term **report bookmark** when referring to the type of bookmark that is created in a Power BI report.

Let's summarize where you are at. You have now successfully created and tested metadata translations to display the names of columns and measures in four different languages. In the next exercise, you will move ahead to implement report label translations.

Exercise 3: Creating and Testing Report Label Translations

In this exercise, you will work through the process of adding report label translations. You will add translations for the report title, navigation button captions and the title of a visual. This will give you experience working with the localized labels table strategy that Translations Builder uses to quickly and easily implement report label translations.

1. Return to the **Product Sales** project in Power BI Desktop and move to **Report view**.
2. Make sure the current page is **Sales Summary**.



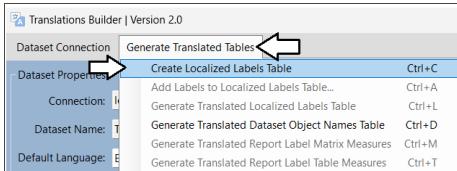
3. Examine the **Product Sales Report** label which is displayed using a Rectangle shape object.



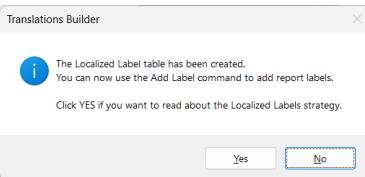
4. Examine the three buttons which provide the navigation menu allowing users to switch between pages.



5. Move back to Translations Builder and drop down the **Generate Translated Tables** menu.
6. Select the **Create Localized Labels Table** to create the **Localized Labels Table**.

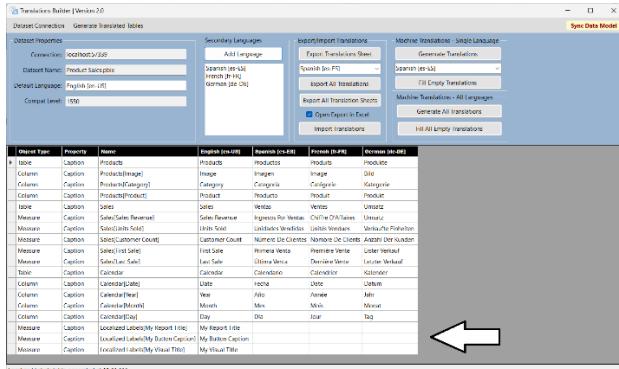


7. When you create the **Localized Labels** table, you will be prompted with the following dialog. Click No to continue.



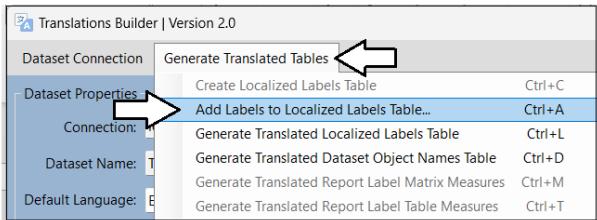
If you click Yes, you will be redirected to [this web page](#) which provides a detailed explanation of the Localized Labels table strategy.

8. Once Translations Builder has created the **Localized Labels** table, it will also add three sample report labels.



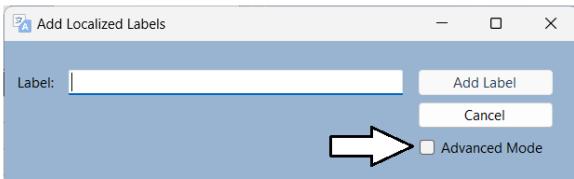
Over the next few steps, you will delete these three sample report labels and replace them by adding five report labels of your own.

9. Drop down the **Generate Translated Tables** menu and select click **Add Labels to the Localized Labels Table**.

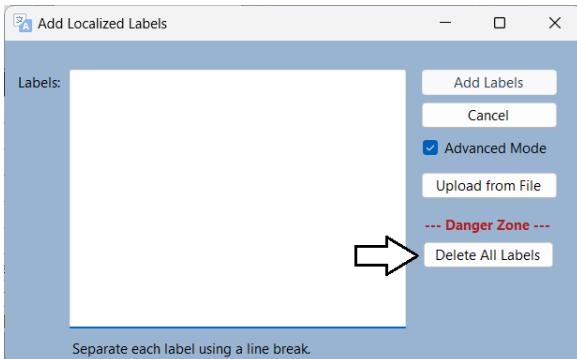


Note you can also execute the **Add Labels to the Localized Labels Table** command using the shortcut key of **Ctrl+A**.

10. In the **Add Localized Labels** dialog, click the **Advanced Mode** checkbox.



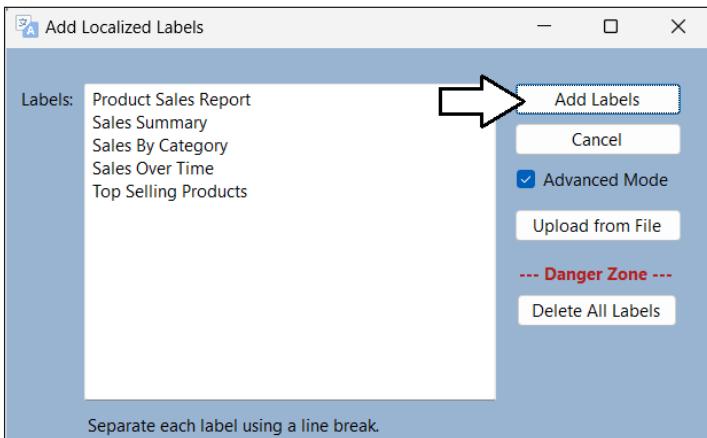
11. Once the dialog is in **Advanced Mode**, click the **Delete All Labels** button to remove the sample report labels.



12. In the **Labels** textbox, type the following labels separated by line breaks.

- **Product Sales Report**
- **Sales Summary**
- **Sales By Category**
- **Sales Over Time**
- **Top Selling Products**

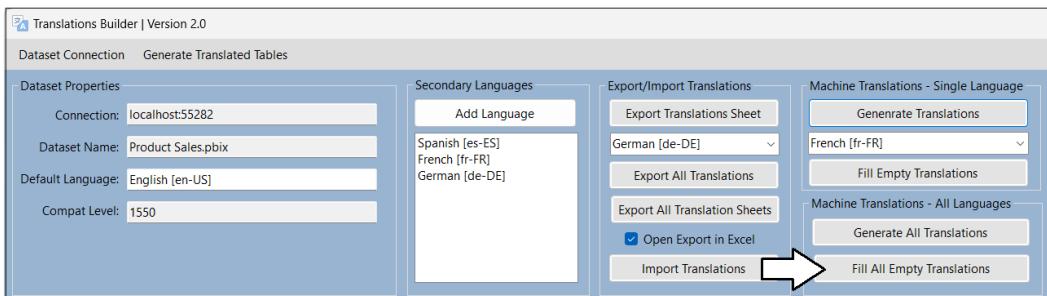
13. Click the **Add Labels** button to add the five new report labels to your project.



14. You should now see five new rows have been added to the translations grid for the new report labels.

Measure	Caption	Sales[Last Sale]	Last Sale	Última Venta	Dernière Vente	Letzter Verkauf
Table	Caption	Calendar	Calendar	Calendario	Calendrier	Kalender
Column	Caption	Calendar[Date]	Date	Fecha	Date	Datum
Column	Caption	Calendar[Year]	Year	Año	Année	Jahr
Column	Caption	Calendar[Month]	Month	Mes	Mois	Monat
Column	Caption	Calendar[Day]	Day	Día	Jour	Tag
Measure	Caption	Localized Labels[Product Sales Report]	Product Sales Report			
Measure	Caption	Localized Labels[Sales Summary]	Sales Summary			
Measure	Caption	Localized Labels[Sales By Category]	Sales By Category			
Measure	Caption	Localized Labels[Sales Over Time]	Sales Over Time			
Measure	Caption	Localized Labels[Top Selling Products]	Top Selling Products			

15. Click the **Fill All Empty Translations** button to generate translations for all the new report labels.

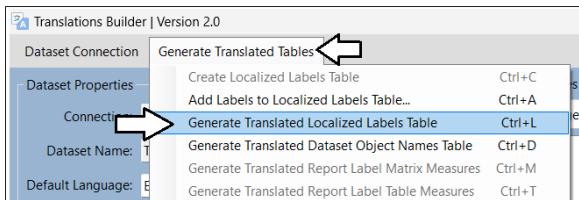


16. At this point, the translations grid should be completely filled with machine-generated translations.

Object Type	Property	Name	English [en-US]	Spanish [es-ES]	French [fr-FR]	German [de-DE]
Table	Caption	Products	Products	Productos	Produits	Produkte
Column	Caption	Products[Image]	Image	Imagen	Image	Bild
Column	Caption	Products[Category]	Category	Categoría	Catégorie	Kategorie
Column	Caption	Products[Product]	Product	Producto	Produit	Produkt
Table	Caption	Sales	Sales	Ventas	Ventes	Umsatz
Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas	Chiffre D'Affaires	Umsatz
Measure	Caption	Sales[Units Sold]	Units Sold	Unidades Vendidas	Unités Vendues	Verkauft Einheiten
Measure	Caption	Sales[Customer Count]	Customer Count	Número De Clientes	Nombre De Clients	Anzahl Der Kunden
Measure	Caption	Sales[First Sale]	First Sale	Primerá Venta	Première Vente	Erster Verkauf
Measure	Caption	Sales[Last Sale]	Last Sale	Última Venta	Dernière Vente	Letzter Verkauf
Table	Caption	Calendar	Calendar	Calendario	Calendrier	Kalender
Column	Caption	Calendar[Date]	Date	Fecha	Date	Datum
Column	Caption	Calendar[Year]	Year	Año	Année	Jahr
Column	Caption	Calendar[Month]	Month	Mes	Mois	Monat
Column	Caption	Calendar[Day]	Day	Día	Jour	Tag
Measure	Caption	Localized Labels[Product Sales Report]	Product Sales Report	Informe De Ventas De Productos	Rapport Sur Les Ventes De Produits	Produktverkaufsbericht
Measure	Caption	Localized Labels[Sales Summary]	Sales Summary	Resumen De Ventas	Récapitulatif Des Ventes	Zusammenfassung Der Verkäufe
Measure	Caption	Localized Labels[Sales By Category]	Sales By Category	Ventas Por Categoría	Ventes Par Catégorie	Umsatz Nach Kategorie
Measure	Caption	Localized Labels[Sales Over Time]	Sales Over Time	Ventas A Lo Largo Del Tiempo	Ventes Au Fil Du Temps	Umsatz Im Laufe Der Zeit
Measure	Caption	Localized Labels[Top Selling Products]	Top Selling Products	Productos Más Vendidos	Produits Les Plus Vendus	Meistverkaufte Produkte

There is one critical step you must complete after modifying report labels in the **Localized Labels** table. More specifically, you must execute **Generate Translated Localized Labels Table** to create the measures that will be used to surface report labels on a report.

17. Drop down the **Generate Translated Tables** menu and select click **Generate Translated Localized Labels Table**.



Note you can also execute the **Generate Translated Localized Labels Table** command using the shortcut key of **Ctrl+L**.

18. Return to the **Product Sales** project in Power BI Desktop and navigate to **Report** view.

19. Locate the **Translated Localized Labels** table in the **Fields** list.

The screenshot shows the Power BI Desktop interface with the 'Fields' pane open. The 'Translated Localized Labels' table is selected, indicated by a red arrow pointing to it. The table contains measures like 'Product Sales Report Label' and 'Sales By Category Label'.

The measures in the **Translated Localized Labels** table are used to display report labels on a Power BI report.

20. Select the measure named **Product Sales Report Label** and examine the DAX expression behind this measure.

The screenshot shows the Power BI Desktop interface with the 'Fields' pane open. The 'Product Sales Report Label' measure is selected, indicated by a red arrow pointing to it. The DAX expression for this measure is: `SWITCH(LEFT(USERCULTURE(), 2), "es", "Reporte de Venta de Productos", "fr", "Report sur les Ventes De Produits", "de", "Produktverkaufbericht", "Product Sales Report")`.

You should not edit the DAX expressions of any measures in the **Translated Localized Labels** table. Any changes you make will be lost as all the measures in this table are deleted and recreated each time you execute **Generate Translated Localized Labels Table**.

21. Take time to examine the DAX expression behind each of the measures in the **Translated Localized Labels** table.

Now that you have created measures for report labels in the **Translated Localized Labels** table, it's time to use them in the report.

22. Ensure that **Sales Summary** is still the active page in **Report** view.



23. In the report layout, select the large black Rectangle shape that displays the report title **Product Sales Report**.

24. With the Rectangle shape selected, move to the **Format** pane and locate the **Text** section inside the **Style** selection.

The screenshot shows the Power BI Desktop interface with the 'Format' pane open. The 'Text' section is expanded, indicated by a red arrow pointing to it. The 'Text' property is set to 'Product Sales Report'.

25. Expand the **Text** section to see the **Text** property is configured with the literal string value of **Product Sales Report**.

The screenshot shows the Power BI Desktop interface with the 'Format' pane open. The 'Text' section is expanded, indicated by a red arrow pointing to it. The 'Text' property is set to 'Product Sales Report'.

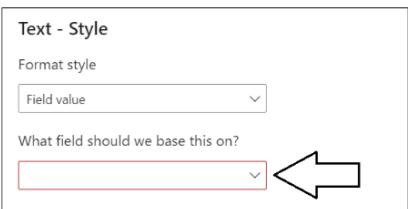
Literal string values in a report layout cannot be localized. You will replace this string using a measure with dynamic translations.

26. Click on the **fx** button to the right of the textbox to replace the literal string value.

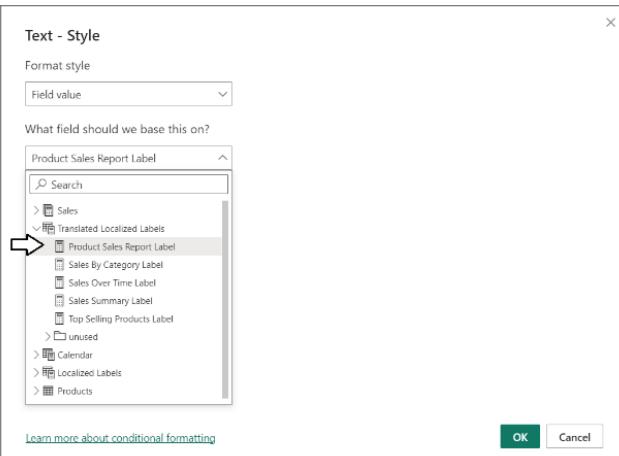


27. In the **Text – Style** dialog, select **Field value** as the **Format style**.

28. Drop down the select control with the caption of **What field should we base this on?**



29. Select the **Product Sales Report Label** measure from the **Translated Localized Labels** table and click **OK**.

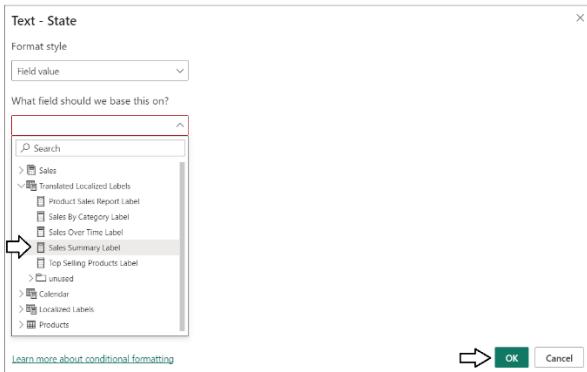


Now that you've updated the Rectangle shape with the report title, it's time to make similar updates to the navigation buttons.

30. Select the first button in the navigation menu with the caption of **Sales Summary**.
 31. With the **Sales Summary** button selected, move to the **Format** pane and click the **Button** tab.
 32. Locate the **Text** section and the **Text** property inside with the literal text value of **Sales Summary**.

Year	Product	Image	Sales Revenue	Units Sold	Customer Count
2022	Apples		€ 174,187.50	232,250	23,657
2022	Oranges		€ 174,122.50	139,298	15,711
2022	Potatoes		€ 167,820.00	134,256	14,895
2022	Tomatoes		€ 129,682.00	74,104	9,058
2022	Milk		€ 117,585.00	47,034	6,259
2022	Butter		€ 105,680.25	46,969	6,373
2022	Cheese		€ 97,567.50	26,018	3,892
2022	Carrots		€ 89,564.10	94,278	11,180
2022	Cucumbers		€ 81,585.00	36,260	5,112
2022	Bananas		€ 70,395.00	74,100	9,656
	Total		€ 1,208,188.85	904,567	53,372

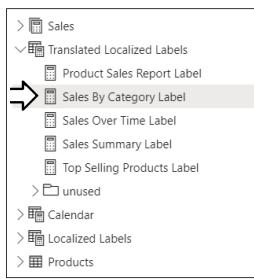
33. Click on the **fx** button to display the **Text - State** dialog.
34. Select the **Sales Summary Label** measure from the **Translated Localized Labels** table.



35. Select the second button in the navigation menu with the caption of **Sales By Category**.



36. With the **Sales By Category** button selected, move to the **Format** pane and click the **Button** tab.
37. Locate the **Text** section and the **Text** property inside with the literal text value of **Sales By Category**.
38. Click on the **fx** button to display the **Text - State** dialog.
39. Select the **Sales By Category Label** measure from the **Translated Localized Labels** table.

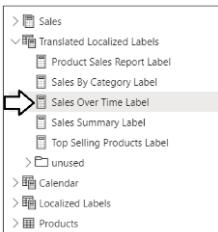


40. Select the third button in the navigation menu with the caption of **Sales Over Time**.



41. With the **Sales Over Time** button selected, move to the **Format** pane and click the **Button** tab.
42. Locate the **Text** section and the **Text** property inside with the literal text value of **Sales Over Time**.
43. Click on the **fx** button to display the **Text - State** dialog.

44. Select the **Sales Over Time Label** measure from the **Translated Localized Labels** table.



You have just implemented report label translations for the report title and for navigation button captions on the **Sales Summary** page. However, there is a separate Rectangle shape with the report title and a separate set of navigation buttons for each page in the report. Over the next few steps, you must repeat the work you did in the **Sales Summary** page on the other two pages.

45. Navigate to the second page named **Sales By Category**.



46. Repeat the same set of steps to display the report title and button captions using measures from **Translated Localized Labels**.



Remember the **Sales By Category** page has the visual with the hard-coded title. Now you will configure this visual to use a measure.

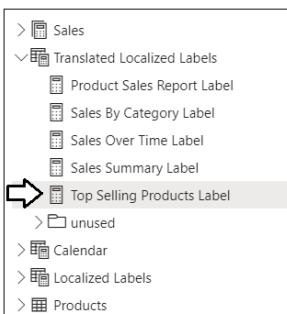
47. Select the Bar Chart visual with the title **Top Selling Products**.

48. With the Table visual selected, move to the **Format** pane and click the **General** tab.

49. Locate the **Title** section and the **Text** property inside with the literal text value of **Top Selling Products**.



50. Select the **Top Selling Products Label** measure from the **Translated Localized Labels** table.



51. Navigate to the third page named **Sales Over Time**.



52. Repeat the same steps to display the report title and button captions using measures from **Translated Localized Labels**.

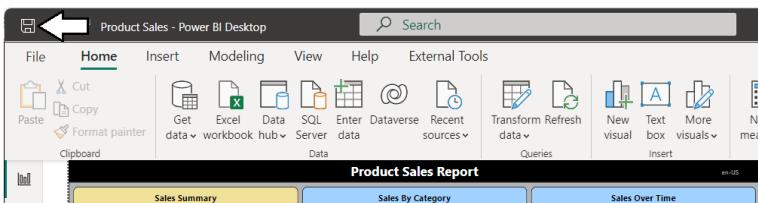


You've now completed the work of updating visuals to use report label translations. Now you need to prepare the report for deployment.

53. Navigate back to the **Sales Summary** page before saving to ensure this page is the startup page.

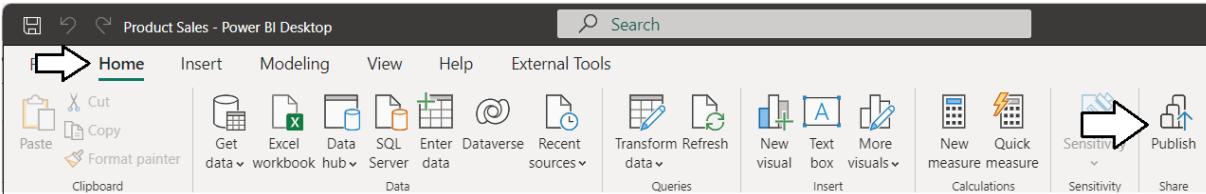


54. Save your work by clicking the **Save** button.



Now, it's once again time to test your work in the Power BI Service,

55. Publish the **Product Sales** project to push your changes to the project's translations to the Power BI Service.



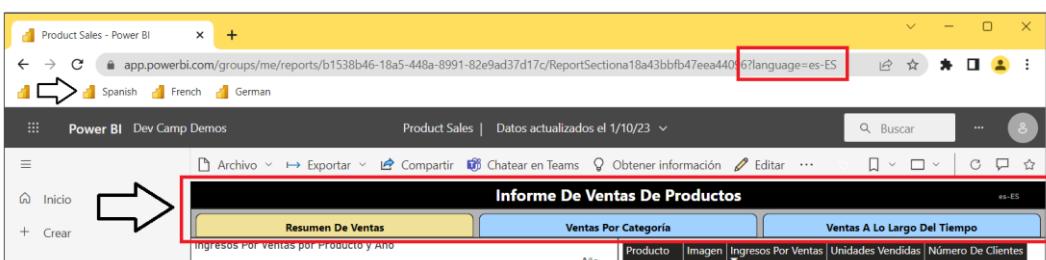
56. When prompted by the **Replace this dataset?** dialog, click the **Replace** button to continue.

57. Once you see **Success!**, click **Open 'Product Sales' in Power BI** to view the report in the Power BI Service.

58. The report should load as normal showing all text in English at first.

59. Use the browser bookmark created earlier to load the report in Spanish.

60. Verify the report title and navigation button captions are displayed using Spanish translations.



61. Navigate to the second page and verify the title for the **Top Selling Products** visual is displayed using Spanish translations.

The screenshot shows the Power BI interface with the report title 'Informe De Ventas De Productos'. The navigation bar has three main items: 'Resumen De Ventas' (highlighted with a red box), 'Ventas Por Categoría', and 'Ventas A Lo Largo Del Tiempo'. Below the navigation bar, there are filters for 'Categoría' (set to '2020'), 'Año' (set to '2021'), and a dropdown menu set to 'Total'. The URL in the browser's address bar includes '?language=es-ES'.

62. Use the browser bookmark created earlier to load the report in French.

63. Verify the report title and navigation button captions are displayed using French translations.

The screenshot shows the Power BI interface with the report title 'Rapport Sur Les Ventes De Produits'. The navigation bar has three main items: 'Récapitulatif Des Ventes' (highlighted with a red box), 'Ventes Par Catégorie', and 'Ventes Au Fil Du Temps'. Below the navigation bar, there are filters for 'Produit' (set to 'Image'), 'Chiffre D'Affaires', 'Unités Vendues', and 'Nombre De Clients'. The URL in the browser's address bar includes '?language=fr-FR'.

64. Navigate to the second page and verify the title for the **Top Selling Products** visual is displayed using French translations.

The screenshot shows the Power BI interface with the report title 'Rapport Sur Les Ventes De Produits'. The navigation bar has three main items: 'Récapitulatif Des Ventes' (highlighted with a red box), 'Ventes Par Catégorie', and 'Ventes Au Fil Du Temps'. Below the navigation bar, there are filters for 'Categoría' (set to '2020'), 'Año' (set to '2021'), and a dropdown menu set to 'Total'. The URL in the browser's address bar includes '?language=fr-FR'.

65. Use the browser bookmark created earlier to load the report in German.

66. Verify the report title and navigation button captions are displayed using German translations.

The screenshot shows the Power BI interface with the report title 'Produktverkaufsbericht'. The navigation bar has three main items: 'Zusammenfassung Der Verkäufe' (highlighted with a red box), 'Umsatz Nach Kategorie', and 'Umsatz Im Laufe Der Zeit'. Below the navigation bar, there are filters for 'Umsatz nach Produkt und Jahr', 'Produkt', 'Bild', 'Umsatz', 'Verkaufte Einheiten', and 'Anzahl Der Kunden'. The URL in the browser's address bar includes '?language=de-DE'.

67. Navigate to the second page and verify the title for the **Top Selling Products** visual is displayed using German translations.

The screenshot shows the Power BI interface with the report title 'Produktverkaufsbericht'. The navigation bar has three main items: 'Zusammenfassung Der Verkäufe' (highlighted with a red box), 'Umsatz Nach Kategorie', and 'Umsatz Im Laufe Der Zeit'. Below the navigation bar, there are filters for 'Kategorie' (set to '2020'), 'Año' (set to '2021'), and a dropdown menu set to 'Gesamt'. The URL in the browser's address bar includes '?language=de-DE'.

You have now implemented report label translations using the Translations Builder Localized Label table strategy. You should be able to see that this will add a significant level of productivity to your future efforts to build Power BI reports that support multiple languages.

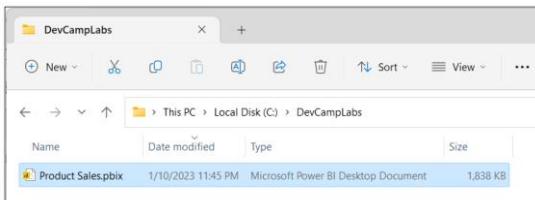
Exercise 4: Creating a Workflow Process To Gather & Integrate Human Translations

Up to this point, you have done the work required to get the report and its underlying dataset into a structure to support translations for secondary languages. You were able to complete this work in a quick and efficient manner using Translations Builder together with machine-generated translations. However, it's important to acknowledge that machine-generated translations alone will not be adequate for many production scenarios. You will need a way to integrate other people acting as translators into a human workflow process.

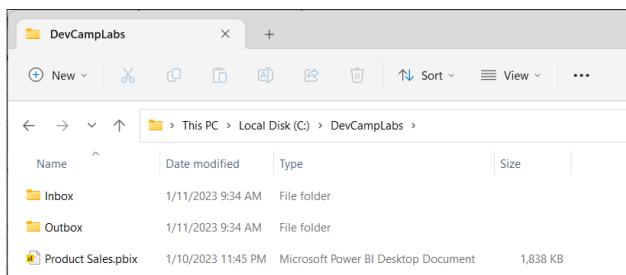
In this lab exercise, you will work with the Translations Builder features to export and import translations using a CSV file format. This will provide a quick way to generate translations sheets that can be sent to human translators. As you will see, translators can make their edits to a translation sheet using Microsoft Excel. Once you've received an updated translation sheet back from a translator, Translations Builder provides an import operation to integrate those updated translations back into the dataset of the current project.

Prerequisite: To complete this exercise, you will need Microsoft Excel installed on the same PC that's running Translations Builder.

1. Launch Windows Explorer and navigate to the folder where you copied the project file **Product Sales.pbix**.

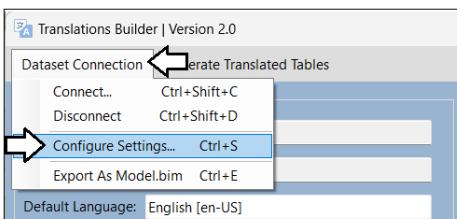


2. Create two new folders inside the lab folder named **Outbox** and **Inbox**.

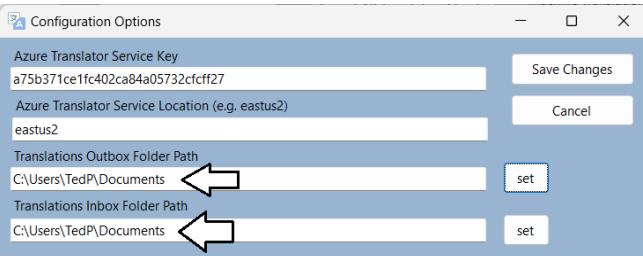


Next, you need to configure settings in Translations Builder so that these folders are used as targets for export and import operations.

3. Return to Translations Builder and drop down the **Dataset Connection** menu.
4. Click **Configure Settings...** to display the **Configuration Options** dialog.

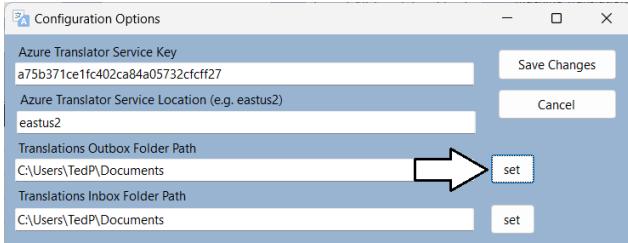


5. By default, folder paths for the **Outbox** and **Inbox** are configured to target the current user's **Documents** folder.

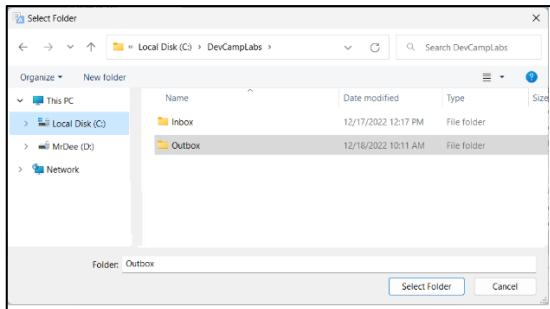


Why does **Outbox** come before **Inbox**? That's because you generally work with the **Outbox** first when you export translation sheets that you will send to translators. Once you get updated translations sheets back from translators, you add them to the **Inbox** for import.

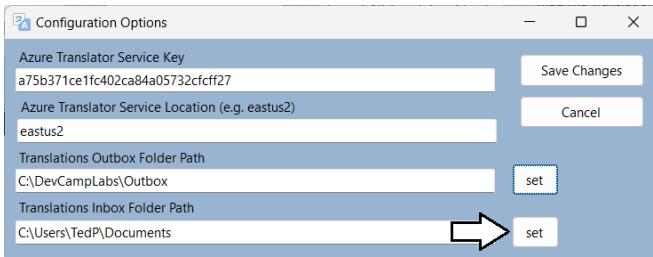
6. Click the **set** button to update the setting for **Translations Outbox Folder Path**.



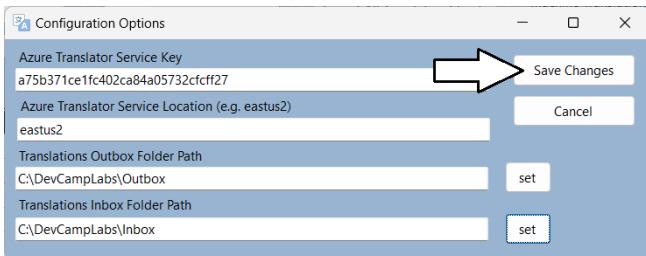
7. Select the **Outbox** folder you created earlier in this exercise.



8. Click the **set** button for **Translations Inbox Folder Path** and select the **Inbox** folder you created earlier

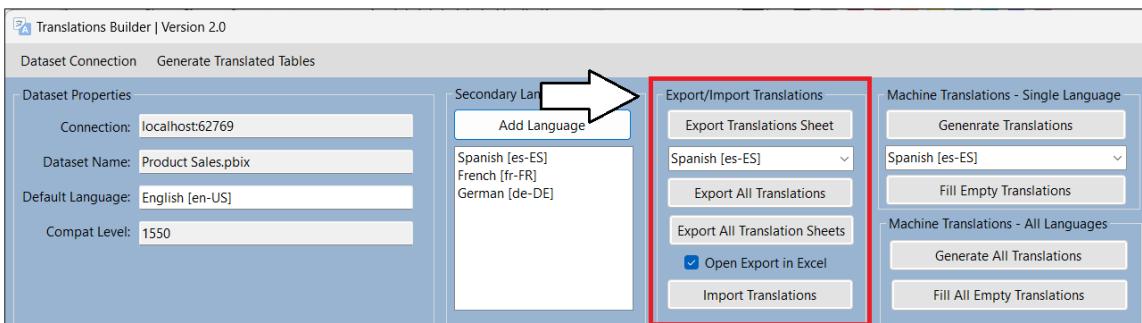


9. Click **Save Changes**.



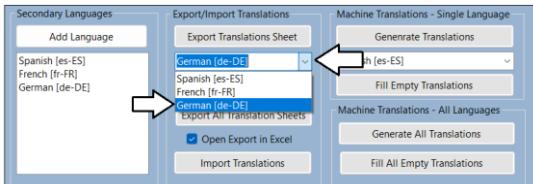
Now that you have configured the folder paths for the **Outbox** and **Inbox**, it's time to begin exporting translation sheets.

10. Examine what's inside the **Export/Import Translations** section.



Let's start by creating a translation sheet for a single language.

- Drop down the selection menu under the **Export Translations Sheet** button and select **German [de-DE]**.



- After selecting **German [de-DE]**, click the **Export Translations Sheet** button.



Translations Builder should create a translation sheet named **Product Sales-Translations-German.csv** and open it in Excel

- Examine the contents of **Product Sales-Translations-German.csv**.

	A	B	C	D	E	F	G	H
1	Object Type	Property	Name	English [en-US]	German [de-DE]			
2	Table	Caption	Products	Products	Produkte			
3	Column	Caption	Products[Image]	Image	Bild			
4	Column	Caption	Products[Category]	Category	Kategorie			
5	Column	Caption	Products[Product]	Product	Produkt			

Over the next two steps you will use a trick in Excel to widen the columns so you can see all their contents.

- Click on the top left corner where the row headers and the column headers meet. This should select all columns and rows.

	A	B	C	D	E	F	G	H
1	Object Type	Property	Name	English [en-US]	German [de-DE]			
2	Table	Caption	Products	Products	Produkte			
3	Column	Caption	Products[Image]	Image	Bild			

- Double-click on the column heading divider between the column headers showing **A** and **B**.

	A	B	C	D	E
1	Object Type	Property	Name	English [en-US]	German [de-DE]
2	Table	Caption	Products	Products	Produkte
3	Column	Caption	Products[Image]	Image	Bild
4	Column	Caption	Products[Category]	Category	Kategorie
5	Column	Caption	Products[Product]	Product	Produkt

16. You should now be able to see all the text from each column.

A	B	C	D	E
1 Object Type	Property Name	English [en-US]	German [de-DE]	
2 Table	Caption Products	Products	Produkte	
3 Column	Caption Products[Image]	Image	Bild	
4 Column	Caption Products[Category]	Category	Kategorie	
5 Column	Caption Products[Product]	Product	Produkt	
6 Table	Caption Sales	Sales	Umsatz	
7 Measure	Caption Sales[Sales Revenue]	Sales Revenue	Umsatz	
8 Measure	Caption Sales[Units Sold]	Units Sold	Verkaufte Einheiten	
9 Measure	Caption Sales[Customer Count]	Customer Count	Anzahl Der Kunden	
10 Measure	Caption Sales[First Sale]	First Sale	Erster Verkauf	
11 Measure	Caption Sales[Last Sale]	Last Sale	Letzter Verkauf	
12 Table	Caption Calendar	Calendar	Kalender	
13 Column	Caption Calendar[Date]	Date	Datum	
14 Column	Caption Calendar[Year]	Year	Jahr	
15 Column	Caption Calendar[Month]	Month	Monat	
16 Column	Caption Calendar[Day]	Day	Tag	
17 Measure	Caption Localized Labels[Product Sales Report]	Product Sales Report	Produktverkaufsbericht	
18 Measure	Caption Localized Labels[Sales Summary]	Sales Summary	Zusammenfassung Der Verkäufe	
19 Measure	Caption Localized Labels[Sales By Category]	Sales By Category	Umsatz Nach Kategorie	
20 Measure	Caption Localized Labels[Sales Over Time]	Sales Over Time	Umsatz Im Laufe Der Zeit	
21 Measure	Caption Localized Labels[Top Selling Products]	Top Selling Products	Meistverkaufte Produkte	

This translation sheet is what you will send to translators. They can then use Excel to review the machine translations and make changes wherever they are required.

17. Close **Product Sales-Translations-German.csv** and return to Translations Builder.

18. Click the **Export All Translations** button to export a master translation sheet with the translations for all languages.



Translations Builder creates a translation sheet named **Product Sales-Translations-Master.csv** and opens this CSV file in Excel

19. When **Product Sales-Translations-Master.csv** open in Microsoft Excel, you cannot see the contents of all columns at first.

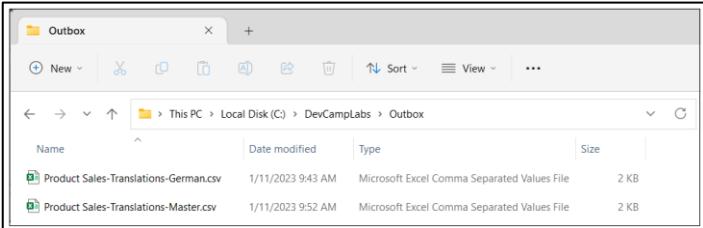
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1 Object Type	Property Name	English [en-US]	Spanish [es-ES]	French [fr-FR]	German [de-DE]										
2 Table	Caption Products	Products	Productos	Produkte											
3 Column	Caption Products[Image]	Image	Imagen	Image											
4 Column	Caption Products[Category]	Category	Categoría	Kategorie											
5 Column	Caption Products[Product]	Product	Producto	Produkt											
6 Table	Caption Sales	Sales	Ventas	Verkäufe											
7 Measure	Caption Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas	Chiffre D'Affaires	Umsatz										
8 Measure	Caption Sales[Units Sold]	Units Sold	Unidades Vendidas	Unités Vendues	Verkaufte Einheiten										
9 Measure	Caption Sales[Customer Count]	Customer Count	Número De Clientes	Número De Clients	Anzahl Der Kunden										
10 Measure	Caption Sales[First Sale]	First Sale	Primerá Venta	Première Vente	Erster Verkauf										
11 Measure	Caption Sales[Last Sale]	Last Sale	Última Venta	Dernière Venta	Letzter Verkauf										
12 Table	Caption Calendar	Calendar	Calendario	Calendrier	Kalender										
13 Column	Caption Calendar[Date]	Date	Fecha	Date	Tag										
14 Column	Caption Calendar[Year]	Year	Año	Année	Jahr										
15 Column	Caption Calendar[Month]	Month	Mes	Mois	Monat										
16 Column	Caption Calendar[Day]	Day	Día	Tag											
17 Measure	Caption Localized Labels[Product Sales Report]	Product Sales Report	Informe De Ventas De Productos	Informé De Ventas De Produkten	Produktverkaufsbericht										
18 Measure	Caption Localized Labels[Sales Summary]	Sales Summary	Resumen De Ventas	Récapitulatif Des Ventes	Zusammenfassung Der Verkäufe										
19 Measure	Caption Localized Labels[Sales By Category]	Sales By Category	Ventas Por Categoría	Ventes Par Catégorie	Umsatz Nach Kategorie										
20 Measure	Caption Localized Labels[Sales Over Time]	Sales Over Time	Ventas A Lo Largo Del Tiempo	Ventes Au Fil Du Temps	Umsatz Im Laufe Der Zeit										
21 Measure	Caption Localized Labels[Top Selling Products]	Top Selling Products	Productos Más Vendidos	Produkte Les Plus Vendus	Meistverkaufte Produkte										

20. Use the Excel trick you learned earlier to expand all columns so you can see the entire contents of all cells.

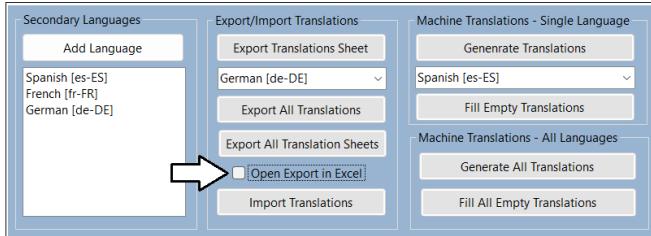
A	B	C	D	E	F	G
1 Object Type	Property Name	English [en-US]	Spanish [es-ES]	French [fr-FR]	German [de-DE]	
2 Table	Caption Products	Products	Productos	Produkte		
3 Column	Caption Products[Image]	Image	Imagen	Image		
4 Column	Caption Products[Category]	Category	Categoría	Kategorie		
5 Column	Caption Products[Product]	Product	Producto	Produkt		
6 Table	Caption Sales	Sales	Ventas	Verkäufe		
7 Measure	Caption Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas	Chiffre D'Affaires	Umsatz	
8 Measure	Caption Sales[Units Sold]	Units Sold	Unidades Vendidas	Unités Vendues	Verkaufte Einheiten	
9 Measure	Caption Sales[Customer Count]	Customer Count	Número De Clientes	Número De Clients	Anzahl Der Kunden	
10 Measure	Caption Sales[First Sale]	First Sale	Primerá Venta	Première Venta	Erster Verkauf	
11 Measure	Caption Sales[Last Sale]	Last Sale	Última Venta	Dernière Venta	Letzter Verkauf	
12 Table	Caption Calendar	Calendar	Calendario	Calendrier	Kalender	
13 Column	Caption Calendar[Date]	Date	Fecha	Date	Tag	
14 Column	Caption Calendar[Year]	Year	Año	Année	Jahr	
15 Column	Caption Calendar[Month]	Month	Mes	Mois	Monat	
16 Column	Caption Calendar[Day]	Day	Día	Tag		
17 Measure	Caption Localized Labels[Product Sales Report]	Product Sales Report	Informe De Ventas De Productos	Informé De Ventas De Produkten	Produktverkaufsbericht	
18 Measure	Caption Localized Labels[Sales Summary]	Sales Summary	Resumen De Ventas	Récapitulatif Des Ventes	Zusammenfassung Der Verkäufe	
19 Measure	Caption Localized Labels[Sales By Category]	Sales By Category	Ventas Por Categoría	Ventes Par Catégorie	Umsatz Nach Kategorie	
20 Measure	Caption Localized Labels[Sales Over Time]	Sales Over Time	Ventas A Lo Largo Del Tiempo	Ventes Au Fil Du Temps	Umsatz Im Laufe Der Zeit	
21 Measure	Caption Localized Labels[Top Selling Products]	Top Selling Products	Productos Más Vendidos	Produkte Les Plus Vendus	Meistverkaufte Produkte	

Now that you have learned to export translations sheets, it's time to examine how to manage translation sheet files.

21. In Windows Explorer, navigate to the **Outbox** folder. You should see the two files you generated using export operations.



22. Return to Translations Builder and uncheck the checkbox with the caption **Open Export in Excel**.



All three export commands use this checkbox to decide whether to open a translation sheet in Excel after it's generated. In some cases, it's handy to have the translation sheet open in Excel. In other scenarios like the one ahead, it's unnecessary and distracting.

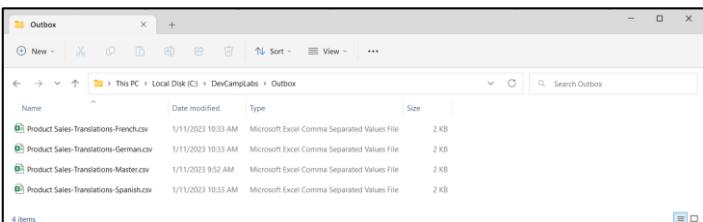
23. Click the **Export All Translation Sheets** button.



The **Export All Translation Sheets** command generates the complete set of translation sheets to be sent to translators.

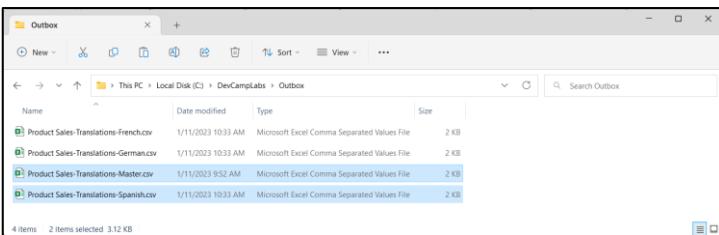
24. Return to the **Outbox** folder in Windows Explorer.

25. You should see that a separate translation sheet has been generated for each secondary language.



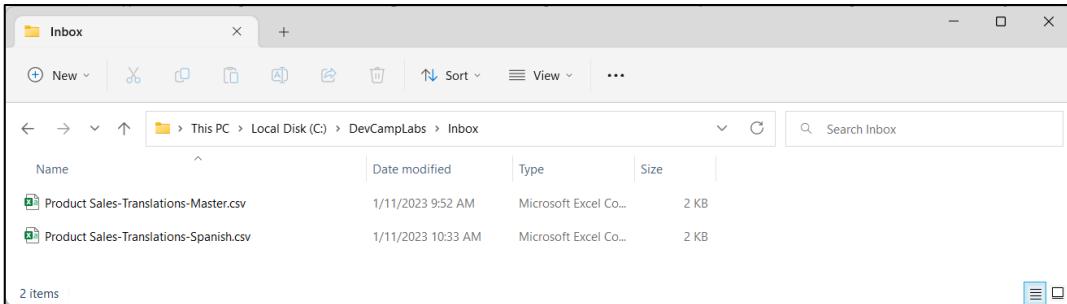
Now that you have experienced how to export translation sheets, you will now learn how to import translation sheets.

26. In the **Outbox** folder, select **Product Sales-Translations-Master.csv** and **Product Sales-Translations-Spanish.csv**.



27. Copy the two selected translation sheet files to the Windows clipboard.

28. Navigate from the **Outbox** folder to the **Inbox** folder.
29. Paste the two translation sheet files from the Windows clipboard into the **Inbox** folder.



30. Open the translation sheet named **Inbox\Product Sales-Translations-Spanish.csv** in Microsoft Excel.

	A	B	C	D	E
1	Object Type	Property	Name	English [en-US]	Spanish [es-ES]
2	Table	Caption	Products	Products	Productos
3	Column	Caption	Products[Product]	Product	Producto
4	Column	Caption	Products[Image]	Image	Imagen
5	Table	Caption	Sales	Sales	Ventas
6	Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas
7	Measure	Caption	Sales[Units Sold]	Units Sold	Unidades Vendidas
8	Measure	Caption	Localized Labels[Product Sales Report]	Product Sales Report	Informe De Ventas De Productos
9	Measure	Caption	Localized Labels[Product List by Sales Revenue]	Product List by Sales Revenue	Lista De Productos Por Ingresos Por Ventas

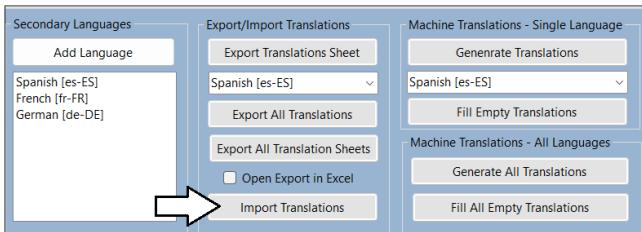
The job of the translator is to review all translations in the fifth column and to make updates where appropriate. From the perspective of the translator, the top row with column headers and the first four columns should be treated as read-only values.

31. Enter new values for a few Spanish translations in the fifth column as shown in the following screenshot.

	A	B	C	D	E
1	Object Type	Property	Name	English [en-US]	Spanish [es-ES]
2	Table	Caption	Products	Products	This
3	Column	Caption	Products[Image]	Image	Is
4	Column	Caption	Products[Category]	Category	A
5	Column	Caption	Products[Product]	Product	Test
6	Table	Caption	Sales	Sales	Ventas
7	Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas
8	Measure	Caption	Sales[Units Sold]	Units Sold	Unidades Vendidas
9	Measure	Caption	Sales[Customer Count]	Customer Count	Número De Clientes
10	Measure	Caption	Sales[First Sale]	First Sale	Primera Venta

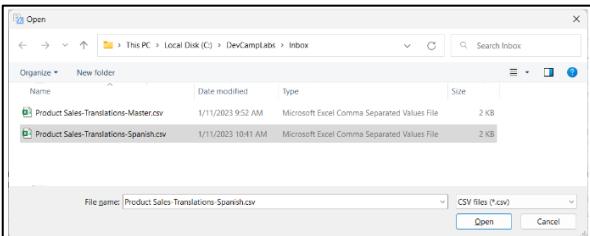
Don't worry whether the text values you add are valid translations. You just need to add text that is different for testing purposes.

32. Save your changes to **Product Sales-Translations-Spanish.csv** and then close the file in Microsoft Excel.
33. Return to Translations Builder and click the **Import Translations** button.



Remember to close translation sheet files in Microsoft Excel before importing them with Translations Builder to prevent errors.

34. In the **Open** file dialog, select **Product Sales-Translations-Spanish.csv** and click **Open**.



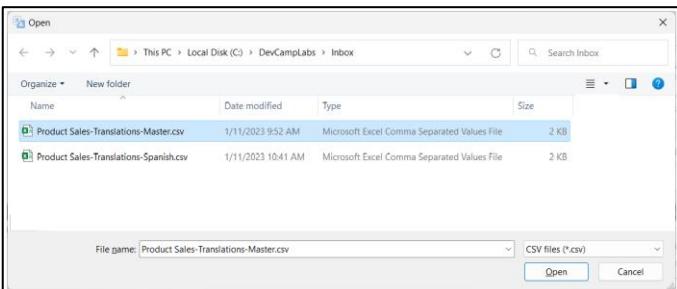
35. You should see that your updates to the Spanish translation sheet now appear in the translation grid.

Object Type	Property	Name	English [en-US]	Spanish [es-ES]	French [fr-FR]	German [de-DE]
Table	Caption	Products	Products	This	Produits	Produkte
Column	Caption	Products[Image]		Is	Image	Bild
Column	Caption	Products[Category]		A	Catégorie	Kategorie
Column	Caption	Products[Product]	Product	Test	Produit	Produkt
Table	Caption	Sales	Sales	Ventas	Ventes	Umsatz

Now that you have seen how to import translations for a single language using an updated translations sheet, it's time to move ahead and learn how to import translations for all languages at once by importing a master translation sheet.

36. click the **Import Translations** button.

37. In the **Open** file dialog, select **Product Sales-Translations-Master.csv** and click **Open**.



38. You should see that the original, machine-generated Spanish translations are now back in the translation grid.

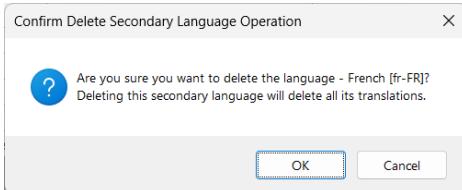
Object Type	Property	Name	English [en-US]	Spanish [es-ES]	French [fr-FR]	German [de-DE]
Table	Caption	Products	Products	Productos	Produits	Produkte
Column	Caption	Products[Image]	Image	Imagen	Image	Bild
Column	Caption	Products[Category]	Category	Categoría	Catégorie	Kategorie
Column	Caption	Products[Product]	Product	Producto	Produit	Produkt
Table	Caption	Sales	Sales	Ventas	Ventes	Umsatz
Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas	Chiffre D'Affaires	Umsatz
Measure	Caption	Sales[Units Sold]	Units Sold	Unidades Vendidas	Unités Vendues	Verkaufté Einheiten

You can see that the master translation sheet can provide a great way to backup and restore your translations work. To make this point, you are now going to delete the French column from the translations grid which will remove all the French translations from the current project. As you will see, Translations Builder will automatically add this column and all its translations back into the project when importing a translation sheet that contains translations for a secondary language that do not exist in the current project.

39. Right-click on the **French [fr-FR]** column header and click **Delete Secondary Language**.

Object Type	Property	Name	English [en-US]	Spanish [es-ES]	French [fr-FR]	German [de-DE]
Table	Caption	Products	Products	Productos	Produits	Produkte
Column	Caption	Products[Image]	Image	Imagen	Image	Bild
Column	Caption	Products[Category]	Category	Categoría	Catégorie	Kategorie
Column	Caption	Products[Product]	Product	Producto	Produit	Produkt
Table	Caption	Sales	Sales	Ventas	Ventes	Umsatz
Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas	Chiffre D'Affaires	Umsatz

40. When prompted by the **Confirm Delete Secondary Language Operation** dialog, click **OK** to continue.



The **Delete Secondary Language** command is useful when you'd like to remove a secondary language and all its translations from the current PBIX project.

41. You should see that the column for French has been removed from the translations grid.

Object Type	Property	Name	English [en-US]	Spanish [es-ES]	German [de-DE]
Table	Caption	Products	Products	Productos	Produkte
Column	Caption	Products[Image]	Image	Imagen	Bild
Column	Caption	Products[Category]	Category	Categoría	Kategorie
Column	Caption	Products[Product]	Product	Producto	Produkt
Table	Caption	Sales	Sales	Ventas	Umsatz
Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas	Umsatz

Now that you have removed all French translations, you will restore them by importing the master translation sheet.

42. Click the **Import Translations** button.

43. In the **Open** file dialog, select **Product Sales-Translations-Master.csv** and click **Open**.

44. After the import operation completes, the **French [fr-FR]** column should reappear as the last column on the right.

Object Type	Property	Name	English [en-US]	Spanish [es-ES]	German [de-DE]	French [fr-FR]
Table	Caption	Products	Products	Productos	Produkte	Produits
Column	Caption	Products[Image]	Image	Imagen	Bild	Image
Column	Caption	Products[Category]	Category	Categoría	Kategorie	Catégorie
Column	Caption	Products[Product]	Product	Producto	Produkt	Produit
Table	Caption	Sales	Sales	Ventas	Umsatz	Ventes
Measure	Caption	Sales[Sales Revenue]	Sales Revenue	Ingresos Por Ventas	Umsatz	Chiffre D'Affaires

The key takeaway with this last step is that master translation sheets make it possible to backup and restore all localization work.

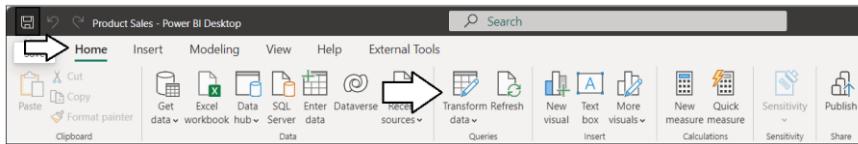
Exercise 5: Implementing Data Translations using Field Parameters

In this exercise you will implement data translations for product names and category names. You will begin by extending the **Products** table by adding new columns that contain translations for product and category names. After that, you will implement data translations by adding Field Parameters and a new table named **Languages**. This will provide a design which makes it possible to filter to select the language for data translations.

1. Return to the **Product Sales** project in Power BI Desktop and move to **Report** view.
2. Make sure the current page is **Sales Summary**.



3. From the **Home** tab on the ribbon, click **Transform Data** to display the Power Query window with queries for the current project.



4. In the left navigation of the Power Query window, select the **Products** query and examine the columns in its output table.

A screenshot of the Power Query Editor. On the left, the 'Queries' pane shows 'Products' selected. The main area displays a table with columns: ProductId, Image, Category, and Product. The table contains 10 rows of data. On the right, the 'Query Settings' pane shows 'Properties' for 'Products' and an 'APPLIED STEPS' list.

ProductId	Image	Category	Product
1	http://classresources.blob.core.windows.net/Images/Apples.png	Fruits	Apples
2	http://classresources.blob.core.windows.net/Images/Bananas.png	Fruits	Bananas
3	http://classresources.blob.core.windows.net/Images/Oranges.png	Fruits	Oranges
4	http://classresources.blob.core.windows.net/Images/Carrots.png	Vegetables	Carrots
5	http://classresources.blob.core.windows.net/Images/Cucumbers.png	Vegetables	Cucumbers
6	http://classresources.blob.core.windows.net/Images/Potatoes.png	Vegetables	Potatoes
7	http://classresources.blob.core.windows.net/Images/Tomatoes.png	Vegetables	Tomatoes
8	http://classresources.blob.core.windows.net/Images/Milk.png	Dairy	Milk
9	http://classresources.blob.core.windows.net/Images/Butter.png	Dairy	Butter
10	http://classresources.blob.core.windows.net/Images/Cheese.png	Dairy	Cheese

5. Examine the query steps in the **APPLIED STEPS** list on the right and locate the step named **RemoveTranslatedColumns**.

A screenshot of the Power Query Editor showing the 'Applied Steps' list. The 'RemovedTranslationColumns' step is highlighted with a large white arrow pointing to it.

6. Right-click the **RemoveTranslatedColumns** step and select the **Delete** menu command to delete this step from the query.

A screenshot of the Power Query Editor context menu for the 'RemovedTranslationColumns' step. The 'Delete' option is highlighted with a large white arrow pointing to it.

7. You should see the query's output table now contains four columns with translations for product names.

The screenshot shows the Power Query Editor interface with the 'Products' query selected. The table has four additional columns: 'ProductTranslationEnglish', 'ProductTranslationSpanish', 'ProductTranslationFrench', and 'ProductTranslationGerman'. The data in these columns is identical to the original 'Product' column, providing multi-language support for product names.

8. If you scroll to the right, you will see there is also four columns with translations for category names as well.

The screenshot shows the Power Query Editor interface with the 'Categories' query selected. The table has four additional columns: 'CategoryTranslationEnglish', 'CategoryTranslationSpanish', 'CategoryTranslationFrench', and 'CategoryTranslationGerman'. The data in these columns is identical to the original 'Category' column, providing multi-language support for category names.

9. Click **Close and Apply** to close the Power Query windows and to execute the **Products** query with the update you made.

The screenshot shows the Power Query Editor ribbon with the 'File' tab selected. An arrow points to the 'Close & Apply' button, which is used to save changes and close the editor.

10. Switch to **Model** view to see the changes that have been made to the **Products** table.

The screenshot shows the Power BI Desktop interface in Model view. The 'Products' table is selected, and its structure is displayed. The 'Category' and 'Product' columns are expanded, revealing their respective multi-language translation columns: 'CategoryTranslationEnglish', 'CategoryTranslationFrench', 'CategoryTranslationGerman', 'CategoryTranslationSpanish' for categories, and 'ProductTranslationEnglish', 'ProductTranslationFrench', 'ProductTranslationGerman', 'ProductTranslationSpanish' for products.

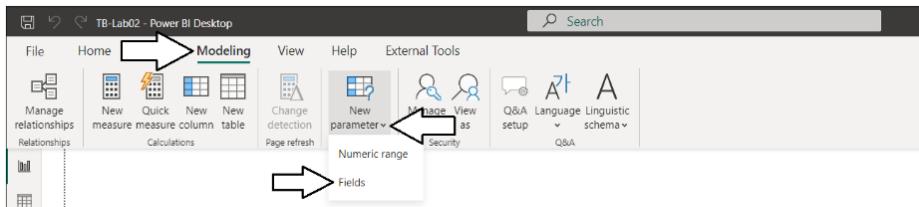
Now that the dataset has been extended with extra translation columns, it's time to create a new Field Parameter.

11. Switch to **Report** view and create a new page in the report. Rename the new page **Test Page**.

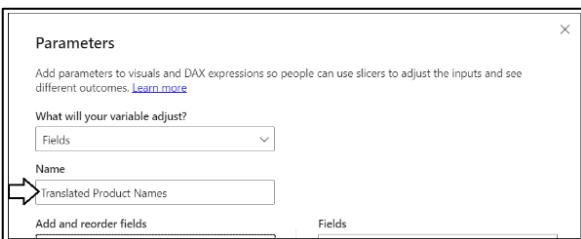


This test page will be used temporarily to test out the Field Parameter. You will delete this page later in this exercise.

12. Navigate to the **Modeling** tab and select **New parameter > Fields** to create a Field Parameter.

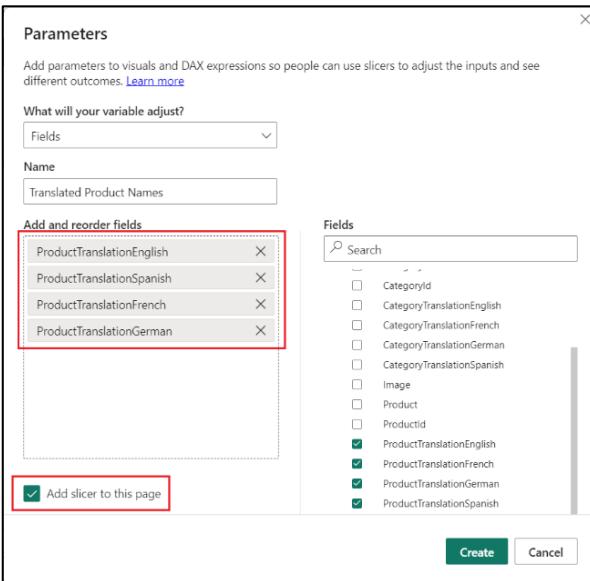


13. Enter the **Name** for a new Field Parameter as **Translated Product Names**.

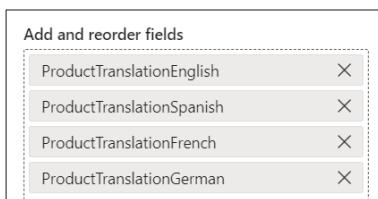


14. Populate the fields collection with the four columns from the **Products** table with product name translations.

15. Leave the **Add slicer to this page** checkbox enabled.



16. Make sure the translation columns are ordered starting with English and then Spanish, French and German,



17. Click the **Create** button in the **Parameters** dialog to create the new Field Parameter.

After you have created a new Field Parameter, it appears in the **Fields** list on the right as a new table.

18. Examine the new Field Parameter named **Translated Product Names** in the **Fields** list.

The screenshot shows the 'Fields' list in the Power BI Desktop ribbon. A new entry, 'Translated Product Names', has been added to the list. It is represented by a green circular icon with a white dot, indicating it is a Field Parameter.

19. If you expand the node for **Translated Product Names**, you will see a single child field with the same name as the parent table.

The screenshot shows the 'Fields' list with the 'Translated Product Names' node expanded. Underneath, there is a single child field also named 'Translated Product Names', which is highlighted with a checkmark.

20. Select the top node for **Translated Product Names** in the **Fields** list to view the DAX expression that defines the fields collection.

The screenshot shows the Power BI Desktop interface with the 'Fields' list open. The top node, 'Translated Product Names', is selected. The DAX editor on the left displays the following code:

```

1 Translated Product Names = {
2     ("ProductTranslationEnglish", NAMEOF('Products'[ProductTranslationEnglish]), 0),
3     ("ProductTranslationSpanish", NAMEOF('Products'[ProductTranslationSpanish]), 1),
4     ("ProductTranslationFrench", NAMEOF('Products'[ProductTranslationFrench]), 2),
5     ("ProductTranslationGerman", NAMEOF('Products'[ProductTranslationGerman]), 3)
6 }

```

The 'Structure' pane on the left shows the table structure with four columns: 'ProductTranslationEnglish', 'ProductTranslationSpanish', 'ProductTranslationFrench', and 'ProductTranslationGerman'. The 'Visualizations' pane on the right shows various visualization icons, and the 'Fields' list on the far right shows the 'Translated Product Names' table.

Now you will conduct a quick experiment so you can better understand how Field Parameters work.

21. Add a **Table** visual to the report page to the right of the slicer.
22. Add the child field from inside the Field Parameter into the **Columns** data role of the Table visual.

The screenshot shows a report page with a 'Table' visual. The table has four columns: 'ProductTranslationEnglish', 'ProductTranslationSpanish', 'ProductTranslationFrench', and 'ProductTranslationGerman'. To the right of the table, the 'Fields' list is open, showing the 'Translated Product Names' table. An arrow points from the 'Fields' list towards the 'Columns' dropdown menu of the table visual, indicating where to add the child field.

As long as nothing is selected in the slicer, the Table visual displays all four source columns.

23. Select a specific field in the slicer. When you select a single slicer value such as the **ProductTranslationSpanish** field, the slicer applies filtering that reduces the number of columns displayed in the Table visual from four columns to a single column.

In the previous screenshot, you can see that the column values for product names have been translated into Spanish. However, there is still an issue with the column header. The column header still displays the column name from the underlying datasource which is **ProductTranslationSpanish**. The reason for this is that those column header values were hard-coded into the DAX expression when Power BI Desktop created the new Field Parameter.

24. Switch to **Data** view.
25. Select the top node for **Translated Product Names** to see the DAX expression that defines the Field Parameter's fields collection.

```
Translated Product Names = {
    {"ProductTranslationEnglish", NAMEOF('Products'[ProductTranslationEnglish]), 0},
    {"ProductTranslationSpanish", NAMEOF('Products'[ProductTranslationSpanish]), 1},
    {"ProductTranslationFrench", NAMEOF('Products'[ProductTranslationFrench]), 2},
    {"ProductTranslationGerman", NAMEOF('Products'[ProductTranslationGerman]), 3}
}
```

26. Currently, the DAX expression has hard-coded column names like **ProductTranslationEnglish** and **ProductTranslationSpanish**.

```
Translated Product Names = {
    {"ProductTranslationEnglish", NAMEOF('Products'[ProductTranslationEnglish]), 0},
    {"ProductTranslationSpanish", NAMEOF('Products'[ProductTranslationSpanish]), 1},
    {"ProductTranslationFrench", NAMEOF('Products'[ProductTranslationFrench]), 2},
    {"ProductTranslationGerman", NAMEOF('Products'[ProductTranslationGerman]), 3}
}
```

The way to resolve this issue is to update the DAX expression to replace the column names with localized translations.

27. Replace the existing DAX expression with the follow DAX expression.

```
Translated Product Names = {
    {"Product", NAMEOF('Products'[ProductTranslationEnglish]), 0},
    {"Producto", NAMEOF('Products'[ProductTranslationSpanish]), 1},
    {"Produit", NAMEOF('Products'[ProductTranslationFrench]), 2},
    {"Produkt", NAMEOF('Products'[ProductTranslationGerman]), 3}
}
```

28. After you have replaced the DAX expression, verify that the column header is now translated properly along with product names.

Up to this point we have only examined the Field Parameter in **Report** view. Now it's time to switch over to **Data** view where you will be able to see two addition fields inside the Field Parameter that are hidden from **Report** view.

29. Switch to **Data** view and select the top-level node for **Translated Product Names**.

The screenshot shows the Power BI Desktop interface with the 'Table tools' ribbon selected. In the 'Structure' pane, there is a field parameter named 'Translated Product ...'. Below it, a table named 'Translated Product Names' is defined with four columns: 'Product', 'Fields', 'Translated Product Names Order', and 'SortOrder'. The 'Translated Product Names' node is expanded in the 'Data' pane, showing its three child fields: 'Translated Product Names', 'Translated Product Names Fields', and 'Translated Product Names Order'.

30. Expand the top-level node for **Translated Product Names** and examine the fields inside.

A callout arrow points from the 'Translated Product Names' node in the Data pane to its expanded state, which lists the three fields it contains: 'Translated Product Names', 'Translated Product Names Fields', and 'Translated Product Names Order'.

The names of the fields inside a Field Parameter are automatically generated based on the name you gave to the top-level Field Parameter. The fields inside a Field Parameter can (and should) be renamed to simplify the data model and to improve readability.

31. In the **Fields** list, double-click on the field named **Translated Product Names** and rename it to **Product**.

The screenshot shows the 'Column tools' ribbon selected. In the 'Fields' list, the 'Translated Product Names' field is highlighted. A callout arrow points from this field to the 'Product' node in the 'Data' pane, indicating where the rename operation should be performed.

32. Using the same technique, rename the two other hidden fields with the shorter names **Fields** and **SortOrder**.

The screenshot shows the 'Table tools' ribbon selected. In the 'Fields' list, the 'Translated Product Names' field has been renamed to 'Product'. The 'Fields' and 'SortOrder' fields are highlighted with red boxes. A callout arrow points from the 'Fields' field in the Data pane to the 'Fields' field in the Fields list, indicating the target for renaming.

Currently, **Translated Product Names** contains three columns named **Product**, **Fields** and **SortOrder**. In the following steps, you will configure support data translations by adding a fourth column named **LanguageId** to enable filtering to switch between languages.

33. Replace the DAX expression for **Translated Product Names** with the following DAX code which adds a fourth string parameter to the row for each language with the lower-case, two character language identifier.

```
Translated Product Names = {
    ("Product", NAMEOF('Products'[ProductTranslationEnglish]), 0, "en"),
    ("Producto", NAMEOF('Products'[ProductTranslationSpanish]), 1, "es"),
    ("Produit", NAMEOF('Products'[ProductTranslationFrench]), 2, "fr"),
    ("Produkt", NAMEOF('Products'[ProductTranslationGerman]), 3, "de")
}
```

34. Once you've updated the DAX expression with a language identifier for each language, a new column will appear named **Value4**.

35. Rename the **Value4** column to **Langageld**.

36. Finally, don't forget to configure the sort column for the new column named **Langageld**.

You do not have to worry about configuring the sort column for the two pre-existing fields named **Fields** and **Product**. That is done automatically by Power BI Desktop when you create a new Field Parameter. However, you need to explicitly configure the sort column when you add additional columns such as **Langageld**.

37. Switch to **Model** view to inspect the Field Parameter named **Translated Product Names**.
 38. Hide the **LanguageId** column from **Report** view. Report authors will never need to see or use this column in a report.

At this point, you no longer need the slicer that was automatically added by Power BI Desktop when creating the Field Parameter. While the slicer added by Power BI Desktop is great for simple demos, it can only control a single Field Parameter at a time. You need a more scalable, report-wide strategy to enable switching between languages that works across multiple Field Parameters.

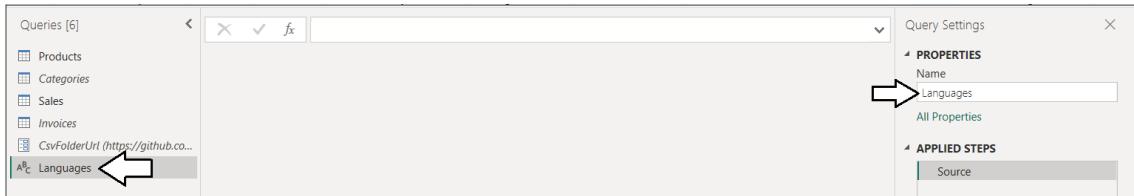
39. Delete the page you created earlier in this exercise named **Test Page**.

Let's summarize where you are at. You have created a Field Parameter named **Translated Product Names** and extended it with an extra column named **LanguageId**. The **LanguageId** column will be used to filter and effectively select which language is used for data translations. Over the next few steps, you will continue building out the data translations strategy by adding the **Languages** table.

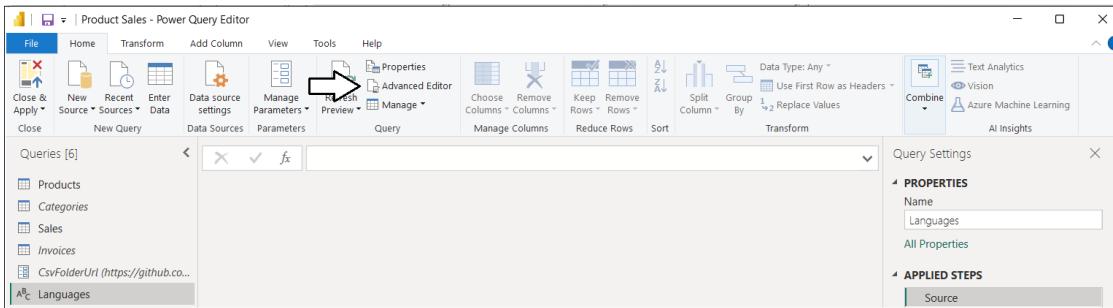
40. Create a new query by dropping down the **Get data** menu and selecting **Blank query**.

41. You should see the Power Query window open with a new query named **Query1**.

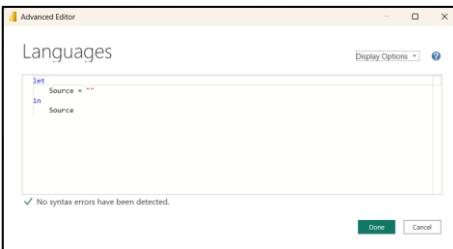
42. Rename the new query from **Query1** to **Languages**.



43. From the **Home** tab, click the **Advanced Editor** button to display the Advanced Editor window for the **Languages** query.



44. The Advanced Editor window will initially contain a minimal amount of M code.



45. Copy and paste the following M code into the Advanced Editor window

```
let
    LanguagesTable = #table(type table [
        Language = text,
        LanguageId = text,
        DefaultCulture = text,
        SortOrder = number
    ], {
        {"English", "en", "en-US", 1 },
        {"Spanish", "es", "es-ES", 2 },
        {"French", "fr", "fr-FR", 3 },
        {"German", "de", "de-DE", 4 }
    }),
    SortedRows = Table.Sort(LanguagesTable,{{"SortOrder", Order.Ascending}}),
    QueryOutput = Table.TransformColumnTypes(SortedRows,{{"SortOrder", Int64.Type}})
in
    QueryOutput
```

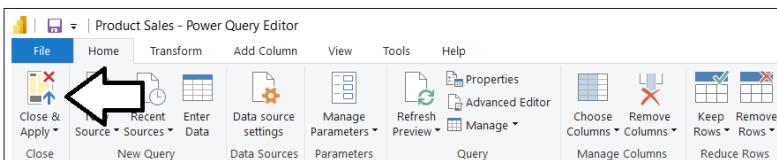
46. Once you have copied the M code from above into the **Advanced Editor** window, click **Done** to save your changes.



47. When the **Advanced Editor** window closes, the **Languages** query should run and generate an output table with four rows.

Language	Languageld	DefaultCulture	SortOrder
English	en	en-US	1
Spanish	es	es-ES	2
French	fr	fr-FR	3
German	de	de-DE	4

48. Click **Close & Apply** to close the Power Query window and to add the **Languages** table to the project's data model.

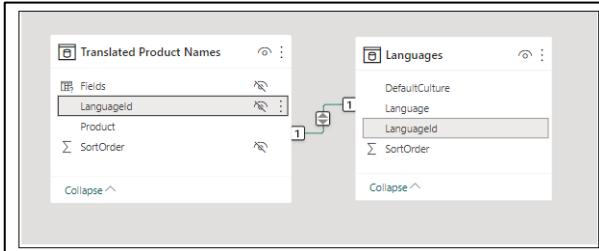


49. In the main Power BI Desktop window, switch to **Data** view and select the **Languages** table in the **Data** pane.

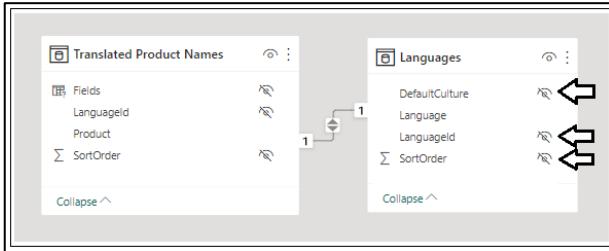
50. Configure the columns named **Language**, **Languageld** and **DefaultCulture** to use **SortOrder** as their sort column.

51. Switch to **Model** view and reposition **Translated Product Names** and the **Languages** table so they are next to each other.

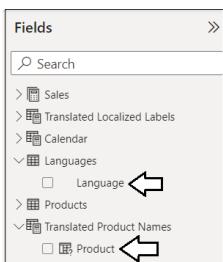
52. Create a one-to-one relationship between **Translated Product Names** and **Languages** based on the **LangugaeId** column.



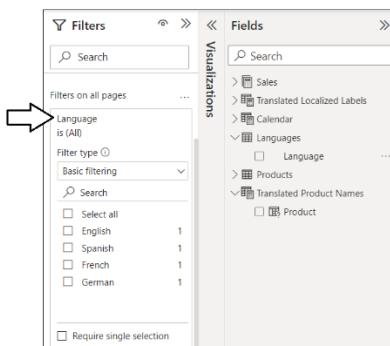
53. Hide the columns **DefaultCulture**, **Language** and **SortOrder** so that **Language** is the only non-hidden column.



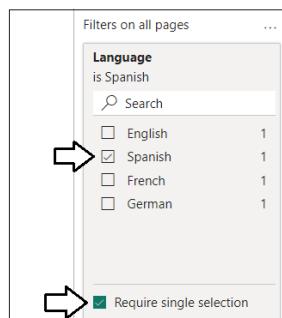
54. Switch to **Report** view and expand the nodes for **Translated Product Names** and **Languages** in **Fields** pane. Each of these tables should now only display a single child field when in **Report** view.



55. Open the **Filters** pane and create a report-level filter based on the **Language** column from the **Languages** table.



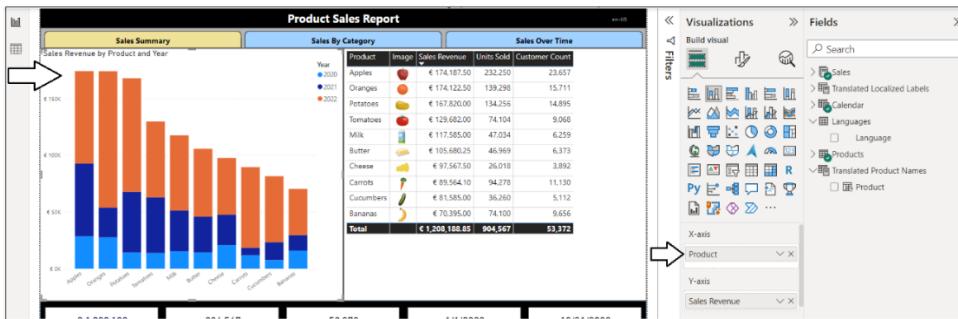
56. Enable the **Require single selection** filter option and set the filter to **Spanish**.



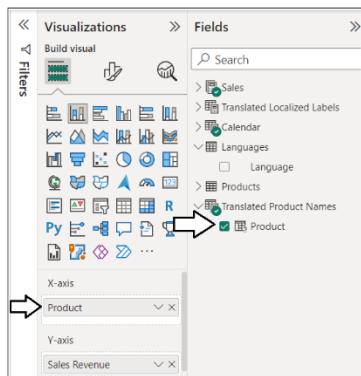
57. Make sure the current page is **Sales Summary**.



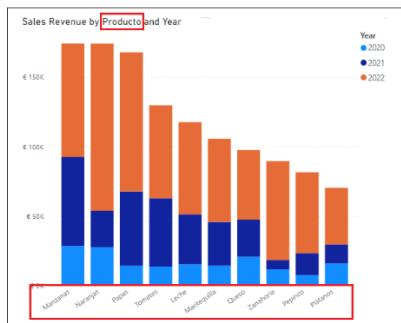
58. Select the Column chart on the left and verify it's using the **Product** column from the **Products** table for the **X-axis** data role.



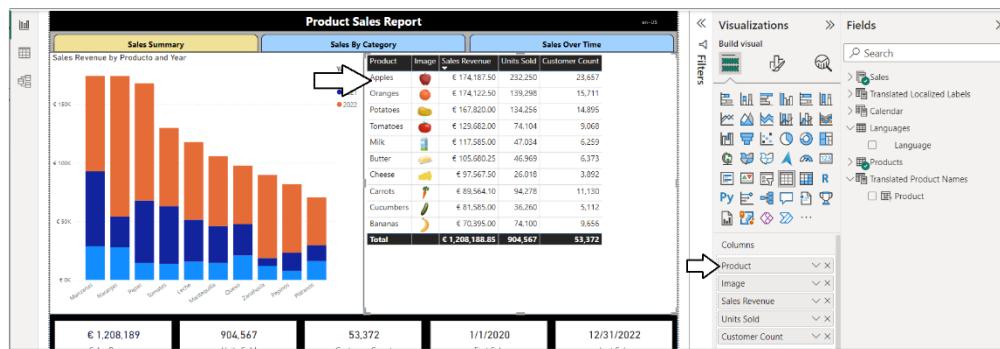
59. Remove the **Product** column from the **Products** table and replace it with the **Product** field from **Translated Product Names**.



60. You should now see the data translations in the Column chart display in Spanish.



61. Select the Table visual on the right. You can see that it is also using the **Product** column from the **Products** table.



62. Remove the **Product** column from the **Products** table and replace it with the **Product** field from **Translated Product Names**. The table should now update and display product names in Spanish.

Product	Image	Sales Revenue	Units Sold	Customer Count
Manzanas	🍎	€ 174,122.50	139,298	15,711
Naranjas	🍊	€ 167,820.00	134,256	14,895
Papas	🥔	€ 129,632.00	74,104	9,068
Tomates	🍅	€ 117,885.00	47,054	6,259
Mantequilla	🧈	€ 105,600.25	46,969	6,373
Queso	🧀	€ 97,567.50	26,018	3,892
Zanahoria	🥕	€ 69,564.10	94,279	11,130
Pepinos	🥒	€ 81,585.00	36,260	5,112
Pitáyanos	🥑	€ 70,395.00	74,100	9,656
Total		€ 1,208,188.85	904,567	53,372

63. Now you can test your data translations by changing the filter on the **Languages** table to quickly switch between languages.

Product	Image	Sales Revenue	Units Sold	Customer Count
Äpfel	🍏	€ 174,187.50	232,250	23,657
Orangen	🍊	€ 174,122.50	139,298	15,711
Kartoffeln	🥔	€ 167,820.00	134,256	14,895
Tomaten	🍅	€ 129,632.00	74,104	9,068
Milch	🥛	€ 117,885.00	47,034	6,259
Butter	🧈	€ 105,600.25	46,969	6,373
Käse	🧀	€ 97,567.50	26,018	3,892
Möhren	🥕	€ 69,564.10	94,279	11,130
Gurken	🥒	€ 81,585.00	36,260	5,112
Bananen	🥑	€ 70,395.00	74,100	9,656
Total		€ 1,208,188.85	904,567	53,372

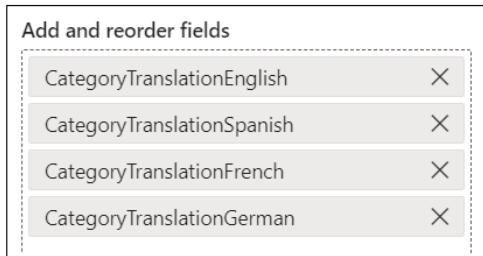
Now you have just implemented data translations for product names. Now, you will continue by adding the same type of data translation support for category names as well.

64. Navigate to the **Modeling** tab and select **New parameter > Fields** to create a Field Parameter.

65. Create a new Field Parameter named **Translated Category Names**. Populate the fields collection with the four columns from the **Products** table with category name translations and be sure to disable the **Add slicer to this page** option.

Fields
CategoryTranslationEnglish
CategoryTranslationSpanish
CategoryTranslationFrench
CategoryTranslationGerman

66. Make sure the fields collection is sorted like the fields shown in the following screenshot and then click the **Create** button.



67. Move to **Data** view and select the Field Parameter named **Translated Category Names** in the Data pane on the right.

Category	Fields	SortOrder
CategoryTranslationEnglish	'Products'[CategoryTranslationEnglish]	0
CategoryTranslationSpanish	'Products'[CategoryTranslationSpanish]	1
CategoryTranslationFrench	'Products'[CategoryTranslationFrench]	2
CategoryTranslationGerman	'Products'[CategoryTranslationGerman]	3

68. Expand the node **Translated Category Names** and inspect the three child fields inside.

- ✓ Translated Category Names
 - ☒ Translated Category Names
 - ☒ Translated Category Names Fields
 - Σ Translated Category Names Order
- > Translated Product Names

69. Update the child field names to **Category**, **Fields** and **SortOrder** as shown in the following screenshot.

Category	Fields	SortOrder
CategoryTranslationEnglish	'Products'[CategoryTranslationEnglish]	0
CategoryTranslationSpanish	'Products'[CategoryTranslationSpanish]	1
CategoryTranslationFrench	'Products'[CategoryTranslationFrench]	2
CategoryTranslationGerman	'Products'[CategoryTranslationGerman]	3

70. Update the DAX expression for **Translated Category Names** using the following DAX code.

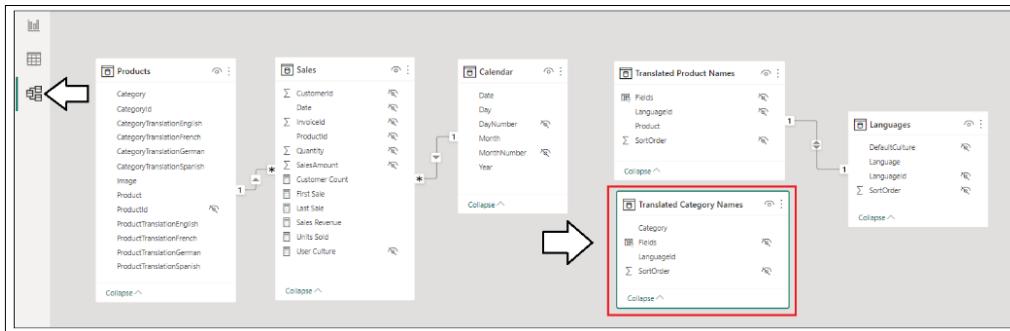
```
Translated Category Names = {
    ("Category", NAMEOF('Products'[CategoryTranslationEnglish]), 0, "en"),
    ("Categoría", NAMEOF('Products'[CategoryTranslationSpanish]), 1, "es"),
    ("Catégorie", NAMEOF('Products'[CategoryTranslationFrench]), 2, "fr"),
    ("Kategorie", NAMEOF('Products'[CategoryTranslationGerman]), 3, "de")
}
```

71. You should see that a new column has been added named **Value4**.

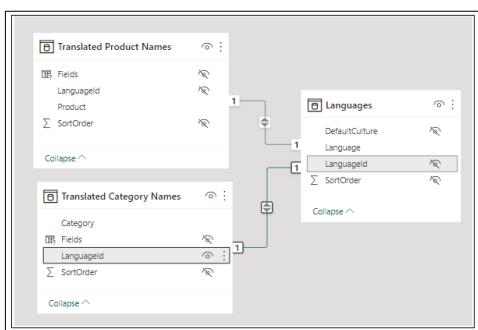
72. Rename the **Value4** field to **Languageld**.

73. Configure the **Languageld** column to use **SortOrder** as its sort column.

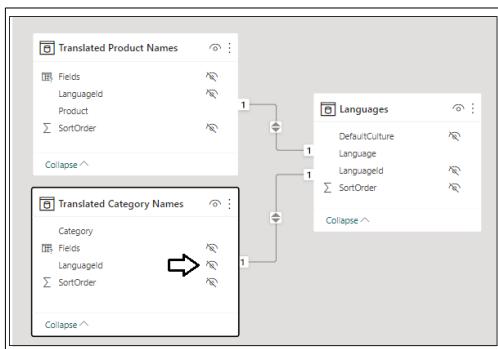
74. Switch to **Model** view.
75. Reposition **Translated Category Names** so its underneath **Translated Product Names**. as shown in the following screenshot.



76. Create a one-to-one relationship between **Translated Category Names** and **Languages** based on the **LangugaeId** column.



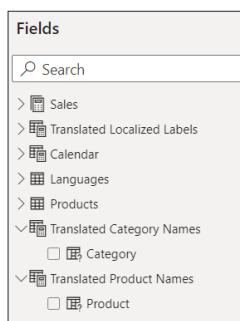
77. Hide the column **LangugaeId** in **Translated Category Names** so it does not show up in **Report** view.



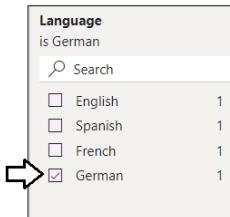
78. Switch to **Report** view and then navigate to the **Sales By Category** page.



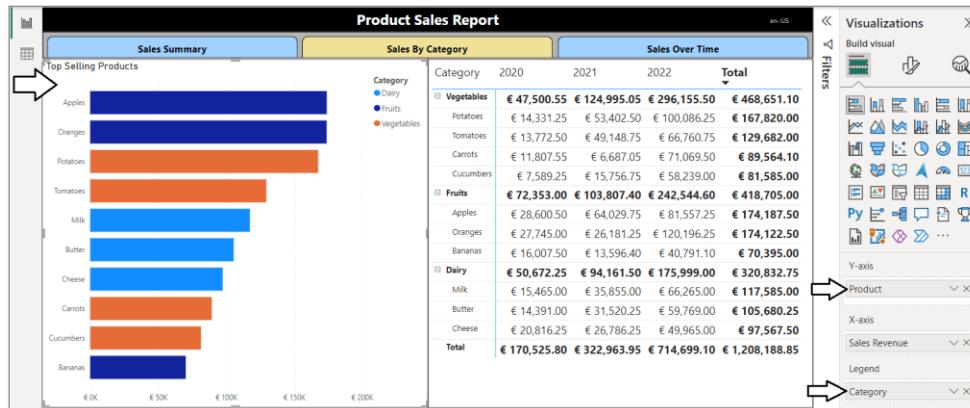
79. When inspecting **Translated Category Names** in **Report** view, it should only contain a single field named **Category**.



80. Set the report-level filter on the **Languages** table to **German** for testing purposes.

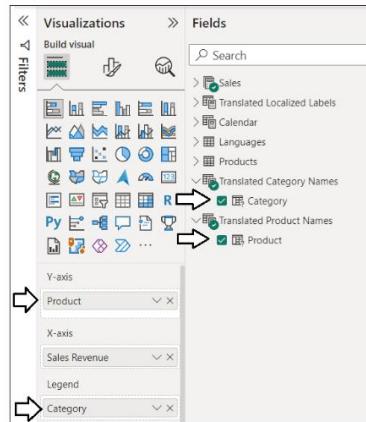


81. Select the Bar chart on the left and verify it uses the **Product** column and the **Category** column from the **Products** table.

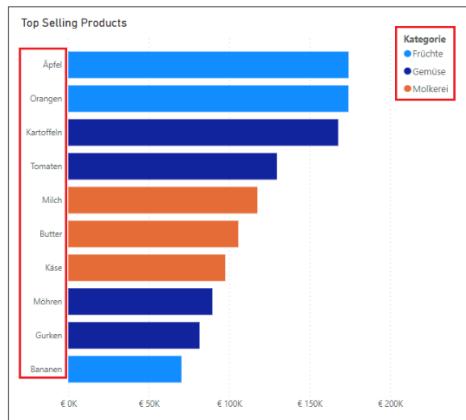


82. Update the Bar chart visual to use **Product** from **Translated Product Names**.

83. Update the Bar chart visual to use **Category** from **Translated Category Names**.



84. You should now see data translations for German in the Y-axis and in the legend.



85. Select the Matrix visual and verify it uses the **Product** column and the **Category** column from the **Products** table.

Category	2020	2021	2022	Total
Vegetables	€ 47,500.55	€ 124,995.05	€ 296,155.50	€ 468,651.10
Potatoes	€ 14,331.25	€ 53,402.50	€ 100,086.25	€ 167,820.00
Tomatoes	€ 13,772.50	€ 49,148.75	€ 66,760.75	€ 129,682.00
Carrots	€ 11,807.55	€ 6,687.05	€ 71,069.50	€ 89,564.10
Cucumbers	€ 7,589.25	€ 15,756.75	€ 58,239.00	€ 81,585.00
Fruits	€ 72,353.00	€ 103,807.40	€ 242,544.60	€ 418,705.00
Apples	€ 28,600.50	€ 64,029.75	€ 81,557.25	€ 174,187.50
Oranges	€ 27,745.00	€ 26,181.25	€ 120,196.25	€ 174,122.50
Bananas	€ 16,007.50	€ 13,596.40	€ 40,791.10	€ 70,395.00
Dairy	€ 50,672.25	€ 94,161.50	€ 175,999.00	€ 320,832.75
Milk	€ 15,465.00	€ 35,855.00	€ 66,265.00	€ 117,585.00
Butter	€ 14,391.00	€ 31,520.25	€ 59,769.00	€ 105,680.25
Cheese	€ 20,816.25	€ 26,786.25	€ 49,965.00	€ 97,567.50
Total	€ 170,525.80	€ 322,963.95	€ 714,699.10	€ 1,208,188.85

86. Update the **Rows** data role of the Matrix visual to use **Product** from **Translated Product Names** and **Category** from **Translated Category Names**. You should now see data translations for German for category names.

Kategorie	2020	2021	2022	Total
Gemüse	€ 47,500.55	€ 124,995.05	€ 296,155.50	€ 468,651.10
Früchte	€ 72,353.00	€ 103,807.40	€ 242,544.60	€ 418,705.00
Molkerei	€ 50,672.25	€ 94,161.50	€ 175,999.00	€ 320,832.75
Total	€ 170,525.80	€ 322,963.95	€ 714,699.10	€ 1,208,188.85

87. Click the **Expand** button with the pitchfork icon to drill down and display product names.

Kategorie	2020	2021	2022	Total
Gemüse	€ 47,500.55	€ 124,995.05	€ 296,155.50	€ 468,651.10
Früchte	€ 72,353.00	€ 103,807.40	€ 242,544.60	€ 418,705.00
Molkerei	€ 50,672.25	€ 94,161.50	€ 175,999.00	€ 320,832.75
Total	€ 170,525.80	€ 322,963.95	€ 714,699.10	€ 1,208,188.85

88. You should now see an expanded matrix that displays both category names and product names in German

Kategorie	2020	2021	2022	Total
Gemüse	€ 47,500.55	€ 124,995.05	€ 296,155.50	€ 468,651.10
Kartoffeln	€ 14,331.25	€ 53,402.50	€ 100,086.25	€ 167,820.00
Tomaten	€ 13,772.50	€ 49,148.75	€ 66,760.75	€ 129,682.00
Möhren	€ 11,807.55	€ 6,687.05	€ 71,069.50	€ 89,564.10
Gurken	€ 7,589.25	€ 15,756.75	€ 58,239.00	€ 81,585.00
Früchte	€ 72,353.00	€ 103,807.40	€ 242,544.60	€ 418,705.00
Äpfel	€ 28,600.50	€ 64,029.75	€ 81,557.25	€ 174,187.50
Orangen	€ 27,745.00	€ 26,181.25	€ 120,196.25	€ 174,122.50
Bananen	€ 16,007.50	€ 13,596.40	€ 40,791.10	€ 70,395.00
Molkerei	€ 50,672.25	€ 94,161.50	€ 175,999.00	€ 320,832.75
Milch	€ 15,465.00	€ 35,855.00	€ 66,265.00	€ 117,585.00
Butter	€ 14,391.00	€ 31,520.25	€ 59,769.00	€ 105,680.25
Käse	€ 20,816.25	€ 26,786.25	€ 49,965.00	€ 97,567.50
Total	€ 170,525.80	€ 322,963.95	€ 714,699.10	€ 1,208,188.85

You will need to click the **Expand** button again any time you switch between languages and want to drill into product names.

89. Select the Slicer visual at the bottom of the page and verify it uses the **Category** column from the **Products** table

The screenshot shows a Power BI report titled "Product Sales Report". It includes three main sections: "Sales Summary", "Sales By Category", and "Sales Over Time". The "Sales By Category" section contains a bar chart of top-selling products and a table of sales data by category. Below these is a Slicer visual with three options: "Dairy", "Fruits", and "Vegetables". A large arrow points to the Slicer.

90. Update the **Fields** data role of the Slicer visual to use **Category** from **Translated Category Names**.

This screenshot shows the same report structure as the previous one, but the Slicer visual now displays the translated category names: "Früchte", "Gemüse", and "Molkerei". The rest of the report remains the same, showing the original English category names in the tables and charts.

Slicer visuals work differently when configured with a field from a Field Parameter. The default behavior is to provide a field picker for the fields collection of the Field Parameter instead of showing the values. You will correct the slicer's behavior in the next step

91. Use the dropdown menu for the **Category** field inside the **Field** data role and select the option to **Show values of selected field**.

The screenshot shows the "Fields" pane in Power BI. In the "Translated Category Names" group, the "Category" field is selected. A context menu is open over this field, with an arrow pointing to the "Show values of selected field" option.

92. The slicer visual should now display the three category names using data translations in German.

The screenshot shows the Slicer visual at the bottom of the report, now correctly displaying the German category names: "Früchte", "Gemüse", and "Molkerei".

93. Now experiment by switching the report-level filter on the **Languages** table to view the page in each of the supported languages.

The screenshot shows a multi-page report titled "Product Sales Report". The first page contains three tabs: "Sales Summary", "Sales By Category", and "Sales Over Time". The "Sales By Category" tab is selected, displaying a bar chart of top-selling products. The "Sales Over Time" tab shows a table of sales data for 2020, 2021, and 2022. To the right of the report is a "Filters" pane. An arrow points from the "Filters" pane to the "Languages" section, which lists "Language is Spanish" and "Language is English". Below the report, there are three tabs: "Frutas", "Lecheria", and "Verduras".

You have now implemented data translations for product names and category names. The next step is to create a set of report bookmarks which will make it possible to set the filter for data translations at report load time.

94. From the **View** tab, click the **Bookmarks** button to display the **Bookmarks** pane.

The screenshot shows the Power BI Desktop interface with the "View" tab selected. In the ribbon, the "Bookmarks" button is highlighted with an arrow. On the right side, there is a "Bookmarks" pane. Another arrow points to the "Add" button in the "Bookmarks" pane, which is used to create new bookmarks.

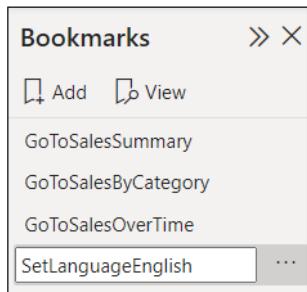
95. Set the report-level filter on the **Languages** table to English and then click the **Add** button in the **Bookmarks** pane.

The screenshot shows the "Product Sales Report" with the "Language" filter set to "English". The "Bookmarks" pane on the right shows a new bookmark named "Bookmark4" has been added. The "Bookmarks" pane also lists other bookmarks: "GoToSalesSummary", "GoToSalesByCategory", and "GoToSalesOverTime".

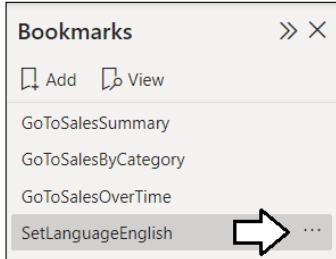
96. You should see a new report bookmark has been created named **Bookmark4**.

The screenshot shows the "Bookmarks" pane with a list of bookmarks. "Bookmark4" is highlighted with a gray background, indicating it is the active bookmark. Other bookmarks listed include "GoToSalesSummary", "GoToSalesByCategory", and "GoToSalesOverTime".

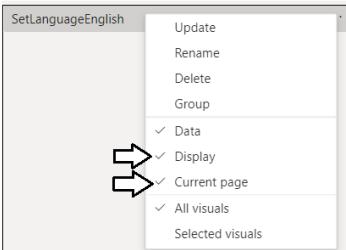
97. Rename the new report bookmark to **SetLanguageEnglish**.



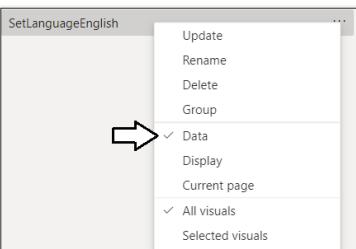
98. Drop down the context menu for the report bookmark named **SetLanguageEnglish**.



99. By default, the behaviors of **Data**, **Display** and **Current page** are enabled. Disable the behaviors for **Display** and **Current page**.

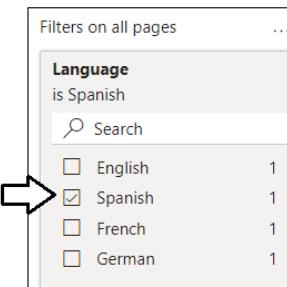


100. Ensure that the only behavior that remains enabled is **Data** as shown in the following screenshot.

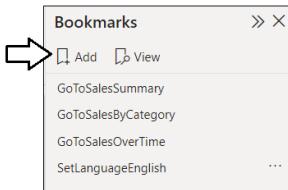


You have now created the first report bookmark. Now you will create three more for the other languages using the same set of steps.

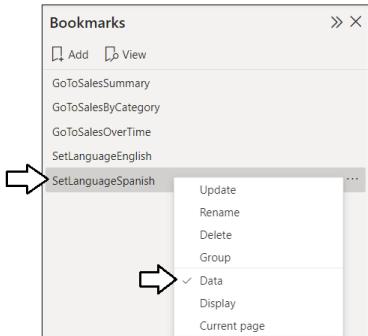
101. Update the report-level filter on the **Languages** table to **Spanish**.



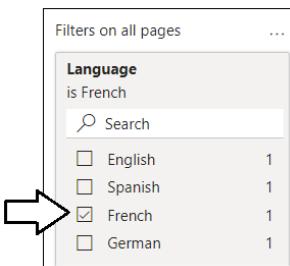
102. Click the **Add** button in the **Bookmarks** pane.



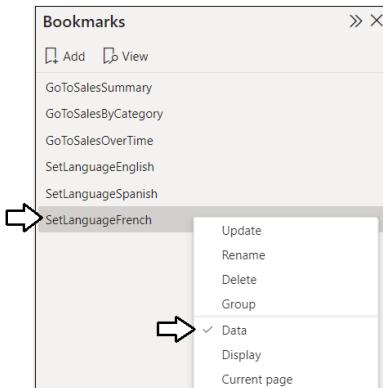
103. Rename the report bookmark to **SetLanguageSpanish** and ensure that the only behavior that remains enable is **Data**.



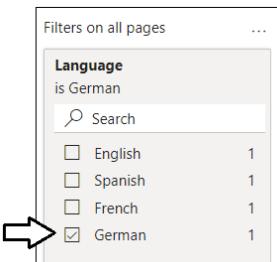
104. Update the report-level filter on the **Languages** table to **French** and then click **Add** in the **Bookmarks** pane.



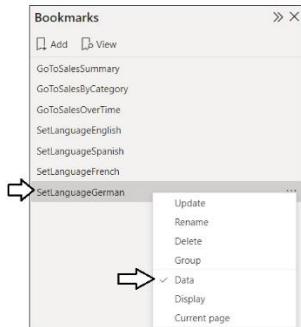
105. Rename the report bookmark to **SetLanguageFrench** and ensure that the only behavior that remains enable is **Data**.



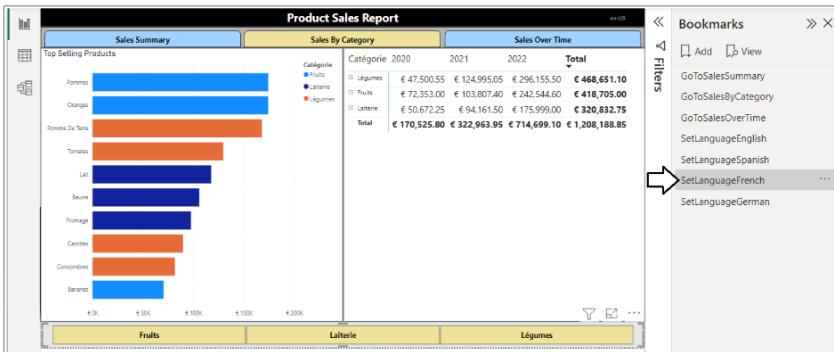
106. Update the report-level filter on the **Languages** table to **German** and then click **Add** in the **Bookmarks** pane.



107. Rename the report bookmark to **SetLanguageGerman** and ensure that the only behavior that remains enable is **Data**.

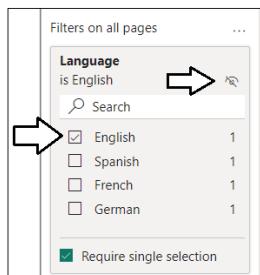


108. Now you should be able to apply these report bookmarks to change the language for data translations.



Before testing the report in the Power BI Service, you should explicitly set the default language filter and the start page.

109. Set the report-level filter on the **Languages** table to **English**. Also hide the **Languages** filter so it is not seen by report consumers.

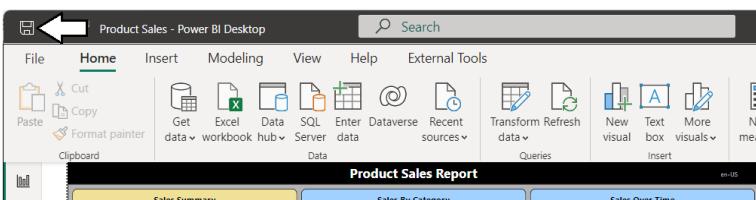


Don't give users the option of changing this filter as it only changes data translations but not the other two types of translations.

110. Navigate back to the **Sales Summary** page and make it active before saving to ensure this page is the startup page.

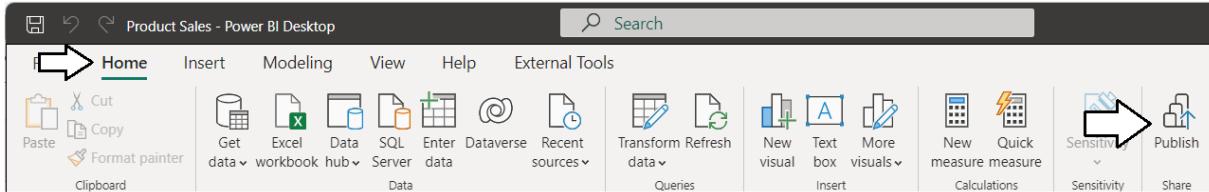


111. Save your work by clicking the **Save** button.



Now, it's time once again to test your work in the Power BI Service.

112. Publish the **Product Sales** project to push your changes to the project's translations to the Power BI Service.



113. When prompted by the **Replace this dataset?** dialog, click the **Replace** button to continue.

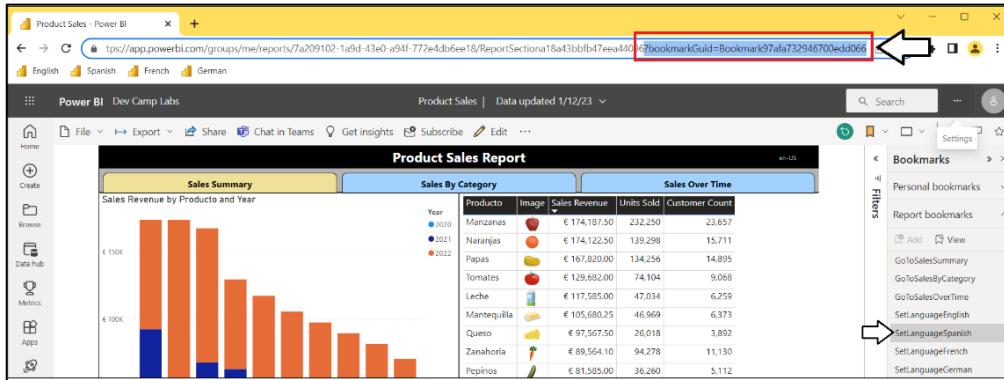
114. Once you see **Success!**, click **Open 'Product Sales'** in **Power BI** to view the report in the Power BI Service.

115. The report should load as normal showing all text in English at first.

116. Drop down the **Bookmarks** menu on the toolbar and select **Show more bookmarks** to display the **Bookmarks** pane.

117. Experiment by applying each of the report bookmarks which set the language used for data translations.

118. After applying a report bookmark, examine the report URL in the address bar and locate the **bookmarkGuid** parameter value



Yes, we agree. The parameter name of **bookmarkGuid** isn't the best name. But you can think of the **bookmarkGuid** value as the identifier for a report bookmark which is generated in the format of **Bookmarkdfa67c1b956e45694481**.

119. Open a text editor such as Notepad and copy and paste the following text.

```
English
?language=en-US&bookmarkGuid={bookmarkGuid for SetLanguageEnglish}

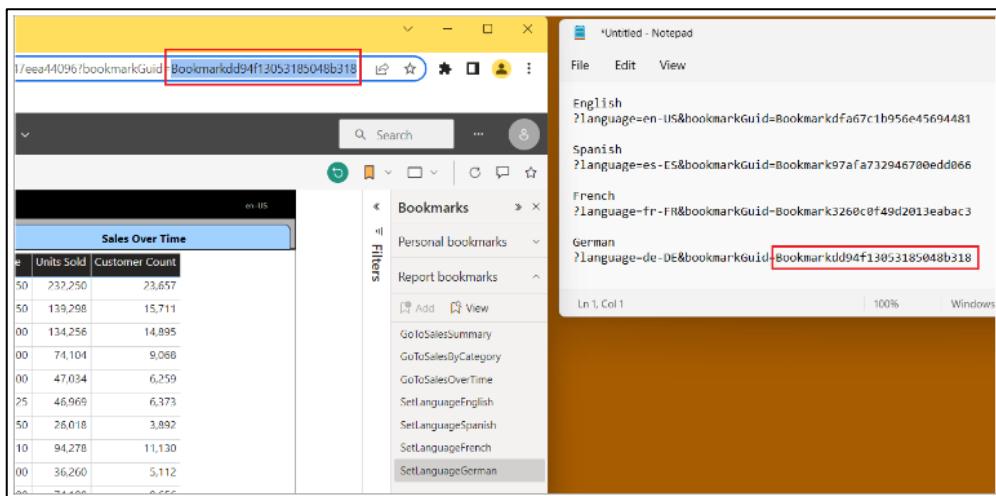
Spanish
?language=es-ES&bookmarkGuid={bookmarkGuid for SetLanguageSpanish}

French
?language=fr-FR&bookmarkGuid={bookmarkGuid for SetLanguageFrench}

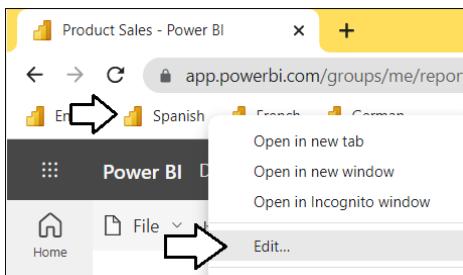
German
?language=de-DE&bookmarkGuid={bookmarkGuid for SetLanguageGerman}
```

120. Discover the **bookmarkGuid** values by applying report bookmarks and copying the parameter value from the address bar.

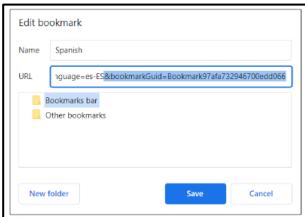
121. Add the **bookmarkGuid** values into the text editor to record the URL parameters you need to load each of the four languages.



122. Edit each of the four browser bookmarks you created earlier by adding the **bookmarkGuid** parameter to load data translations.



123. For each browser bookmark, add the **bookmarkGuid** parameter in addition to **language** to apply the correct report bookmark..



Now all four browser bookmarks should pass the appropriate **bookmarkGuid** parameter value to load data translations correctly.

124. Click the browser bookmark for **Spanish**. The **Sales Summary** page should now be fully translated into Spanish.

125. Navigate to the **Sales By Category** page. This page should also be fully translated into Spanish.

126. Click the browser bookmark for **French**. The **Sales Summary** page should now be fully translated into French.

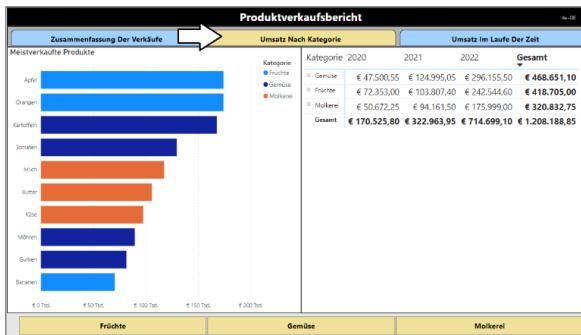
127. Navigate to the **Sales By Category** page. This page should also be fully translated into French.



128. Click the browser bookmark for **German**. The **Sales Summary** page should now be fully translated into German.

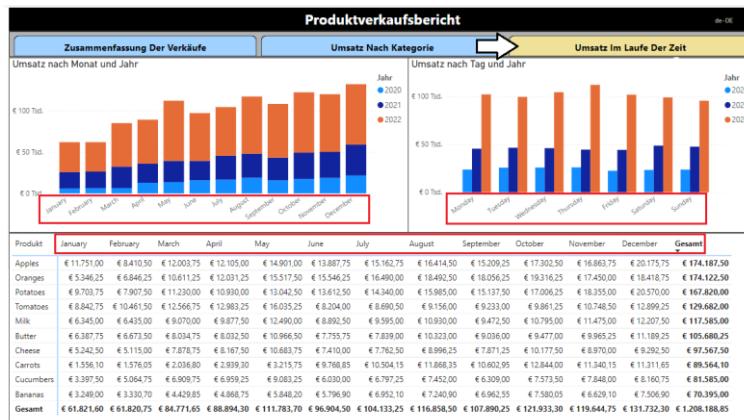


129. Navigate to the **Sales By Category** page. This page should also be fully translated into German.



You have now successfully implemented and tested the data translations for product names and category names. However, your work is not yet done. The **Sales Over Time** page still contains text-based values for calendar names that require your attention.

130. Navigate the **Sales Over Time** page. You can see that the names of months and days are still in English.



You will now move on to the final exercise where you will implement data translations for the names of months and days of the week..

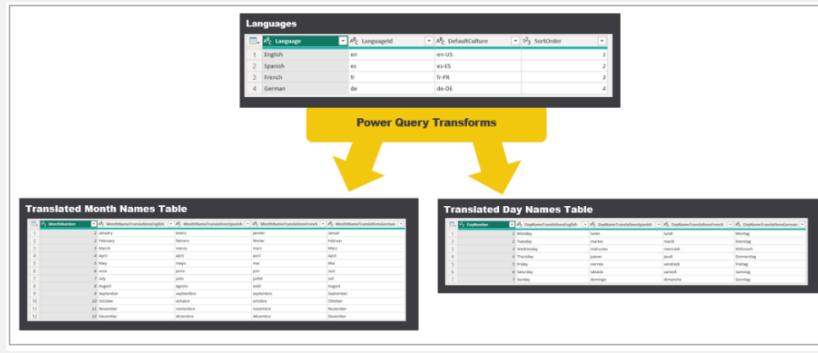
Exercise 6: Implementing Data Translations for a Calendar Table

In this exercise you will implement data translations for the names of months and days in a **Calendar** table. To properly implement data translations for columns in a calendar table, you need a strategy to translate month names and day-of-week names into the secondary languages you plan to support.

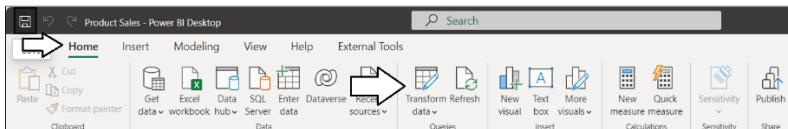
The strategy presented in this lab exercise for implementing calendar table column translations is based on Power Query and the power of the M query language. Power Query provides several built-in functions such as **Date.MonthName** which accept a **Date** parameter and return a text-based calendar name. The following Power Query function call will evaluate to a text-based value of January in French.

```
Date.MonthName( #date(2022, 12, 1), "fr-FR")
```

Power Query is built on a functional query language named M which makes it possible to enumerate through the rows of the **Languages** table to discover what languages and what default cultures are supported in the current project. This makes it possible to write a query which uses the **Languages** table as its source to generate translation tables with the names of months or weekdays.



- From the Home tab on the ribbon, click **Transform Data** to display the Power Query window with queries for the current project.



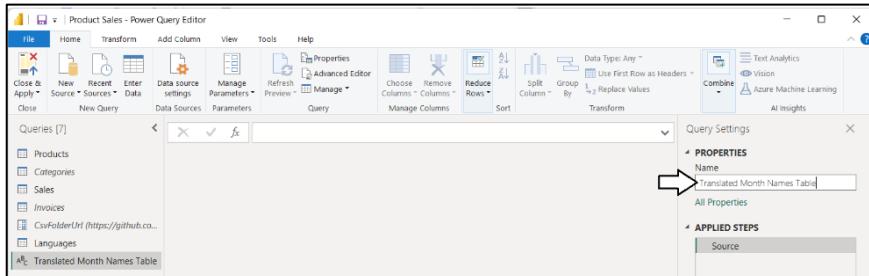
- In the left navigation of the Power Query window, select the **Languages** query and examine the columns in its output table.

A screenshot of the Power Query Editor window. The 'Queries' list on the left shows 'Languages' as the selected query. The main area displays a table with columns: Language, LanguageId, DefaultCulture, and SortOrder. The data rows are: English (en, en-US, 1), Spanish (es, es-ES, 2), French (fr, fr-FR, 3), and German (de, de-DE, 4). A large orange arrow points to the 'Languages' entry in the 'Queries' list.

- Create a new query by dropping down the **New Source** menu and selecting **Blank Query**.

A screenshot of the Power Query Editor window. The 'Queries' list on the left shows 'Languages' as the selected query. The main area displays a table with columns: Language, LanguageId, DefaultCulture, and SortOrder. The data rows are: English (en, en-US, 1), Spanish (es, es-ES, 2), French (fr, fr-FR, 3), and German (de, de-DE, 4). A large orange arrow points to the 'New Source' button in the ribbon, which is highlighted in red. Another orange arrow points to the 'Blank Query' option in the dropdown menu.

4. Rename the new query to **Translated Month Names Table**.

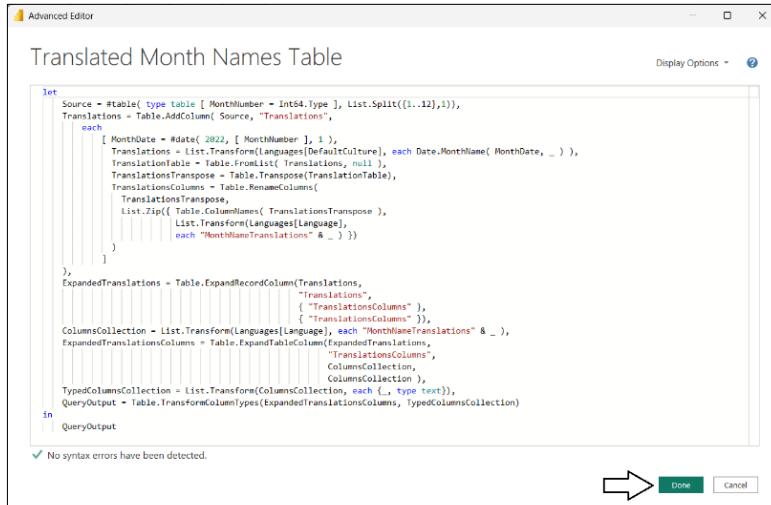


5. Click on the **Advanced Editor** button in the ribbon to open the Advanced Editor widow.
6. Copy the following M code listing and paste it into the Advanced Editor window.

```
let
    Source = #table( type table [ MonthNumber = Int64.Type ], List.Split({1..12},1)),
    Translations = Table.AddColumn( Source, "Translations",
        each
            [ MonthDate = #date( 2022, [ MonthNumber ], 1 ),
                Translations = List.Transform(Languages[DefaultCulture], each Date.MonthName( MonthDate, _ ) ),
                TranslationTable = Table.FromList( Translations, null ),
                TranslationsTranspose = Table.Transpose(TranslationTable),
                TranslationsColumns = Table.RenameColumns(
                    TranslationsTranspose,
                    List.Zip({ Table.ColumnNames( TranslationsTranspose ),
                        List.Transform(Languages[Language],
                            each "MonthNameTranslations" & _ ) })
                )
            ]
        ),
    ExpandedTranslations = Table.ExpandRecordColumn(Translations,
        "Translations",
        { "TranslationsColumns" },
        { "TranslationsColumns" }),
    ColumnsCollection = List.Transform(Languages[Language], each "MonthNameTranslations" & _ ),
    ExpandedTranslationsColumns = Table.ExpandTableColumn(ExpandedTranslations,
        "TranslationsColumns",
        ColumnsCollection,
        ColumnsCollection ),
    TypedColumnsCollection = List.Transform(ColumnsCollection, each {_, type text}),
    QueryOutput = Table.TransformColumnTypes(ExpandedTranslationsColumns, TypedColumnsCollection)
in
    QueryOutput
```

If you have trouble copying the M code from this page, it might be easier to copy it from [here](#) or from the Student Files folder.

7. Once you have copied the M code into the Advanced Editor, click **Done** to save your changes and execute the new query.



8. You should now see the output table of the **Translated Month Names Table** query as shown in the following screenshot.

	MonthNameTranslationsEnglish	MonthNameTranslationsSpanish	MonthNameTranslationsFrench	MonthNameTranslationsGerman
1	January	enero	janvier	Jänner
2	February	febrero	février	Februar
3	March	marzo	mars	März
4	April	abril	avril	April
5	May	mayo	mai	Mai
6	June	junio	juin	Juni
7	July	julio	juillet	Juli
8	August	agosto	août	August
9	September	septiembre	septembre	September
10	October	octubre	octobre	Oktober
11	November	noviembre	novembre	November
12	December	diciembre	décembre	Dezember

You have now created the translation table for month names. Next you will create a second table for day names.

9. Create a new query by dropping down the **New Source** menu and selecting **Blank Query**.

10. Rename the new query to **Translated Day Names Table**.

	Source
1	Source

11. Click on the **Advanced Editor** button in the ribbon to open the Advanced Editor widow.

12. Copy the following M code listing and paste it into the Advanced Editor window.

```

let
    Source = #table( type table [ DayNumber = Int64.Type ], List.Split({1..7}, 1) ),
    NextSunday = Date.AddDays( Date.From(DateTime.LocalNow() ),
        Value.Subtract(6,
            Date.DayOfWeek( Date.From(DateTime.LocalNow()),
                Day.Monday) ) ),
    Translations = Table.AddColumn( Source, "Translations",
        each
            [ DayOfWeek = Date.AddDays(NextSunday, [DayNumber] ),
                Translations = List.Transform(Languages[DefaultCulture],
                    each Date.DayOfWeekName(DayOfWeek, _) ),
                TranslationTable = Table.FromList( Translations, null ),
                TranslationsTranspose = Table.Transpose(TranslationTable),
                TranslationsColumns = Table.RenameColumns(
                    TranslationsTranspose,
                    List.Zip({ Table.ColumnNames( TranslationsTranspose ),
                        List.Transform(Languages[Language],
                            each "DayNameTranslations" & _ ) })
                )
            ]
        ),
    ExpandedTranslations = Table.ExpandRecordColumn(Translations,
        "Translations",
        { "TranslationsColumns" },
        { "TranslationsColumns" }),
    ColumnsCollection = List.Transform(Languages[Language], each "DayNameTranslations" & _ ),
    ExpandedTranslationsColumns = Table.ExpandTableColumn(ExpandedTranslations,
        "TranslationsColumns",
        ColumnsCollection,
        ColumnsCollection ),
    TypedColumnsCollection = List.Transform(ColumnsCollection, each {_, type text}),
    QueryOutput = Table.TransformColumnTypes(ExpandedTranslationsColumns, TypedColumnsCollection)
in
    QueryOutput

```

If you have trouble copying this M code from this page, it might be easier to copy it from [here](#) or from the Student Files folder

13. Once you have copied the M code into the Advanced Editor, click **Done** to save your changes and execute the new query.

```

let
    Source = #table{ type: table, columns: {DayNumber = Int64.Type } },
    BackSunday = Date.AddDays( Date.From(Datetime.LocalNow()), Value.Subtract(0, Date.DayOfWeek( Date.From(Datetime.LocalNow()) ) ) ),
    DayNumber = Date.AddDays( Date.From(Datetime.LocalNow()), Value.Add(0, Date.DayOfWeek( Date.From(Datetime.LocalNow()) ) ) )
in
    Translations = Table.AddColumn( Source, "Translations", each
        [ DayWeek = Date.AddDays(BackSunday, [DayNumber]),
            Translations = List.Transform( Languages[DefaultCulture], each Data.DayNameValue( DayWeek, _ ) ),
            TranslationTable = Table.FromList( Translations, Splitter.SplitByNothing(), null, null, false ),
            TranslationTranspose = Table.Transpose( TranslationTable ),
            TranslationColumns = Table.RenameColumn( TranslationTranspose, List.Zip( Table.ColumnNames( TranslationTranspose ), List.Transform( Languages[Language], each "DayName" & _ ) ) )
        ]
    )
in
    ExpandedTranslations = Table.ExpandRecordColumn( Translations, "Translations", { "Translations", { "DayNameTranslationsEnglish", "DayNameTranslationsSpanish", "DayNameTranslationsFrench", "DayNameTranslationsGerman" } } ),
    ColumnsCollection = List.Transform( Languages[Language], each "DayNameTranslations" & _ ),
    ExpandedTranslationsColumns = Table.ExpandListColumn( ExpandedTranslations, ColumnsCollection, { "DayNameTranslationsEnglish", "DayNameTranslationsSpanish", "DayNameTranslationsFrench", "DayNameTranslationsGerman" } ),
    TypedColumnsCollection = List.Transform( ColumnsCollection, each type: type.Text ),
    QueryOutput = Table.TransformColumnTypes( ExpandedTranslationsColumns, TypedColumnsCollection )
in
    QueryOutput

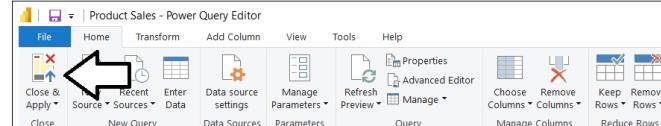
```

No syntax errors have been detected.

14. You should now see the output of the **Translated Day Names Table** query as shown in the following screenshot.

DayNumber	DayNameTranslationsEnglish	DayNameTranslationsSpanish	DayNameTranslationsFrench	DayNameTranslationsGerman
1	Monday	Lunes	Lundi	Montag
2	Tuesday	Martes	Mardi	Dienstag
3	Wednesday	Miércoles	Mercredi	Mittwoch
4	Thursday	Jueves	Jeudi	Donnerstag
5	Friday	Viernes	Vendredi	Freitag
6	Saturday	Sábado	Samedi	Samstag
7	Sunday	Domingo	Domingo	Sonntag

15. Click **Close & Apply** to close the Power Query window and to add two new tables to the project's data model.



16. Switch to **Data** view and locate **Translated Month Names Table** and **Translated Day Names Table** in the **Data** pane.

MonthNumber	MonthNameTranslationsEnglish	MonthNameTranslationsSpanish	MonthNameTranslationsFrench	MonthNameTranslationsGerman
1	January	Enero	Janvier	Januar
2	February	Febrero	Février	Februar
3	March	Marzo	Mars	März
4	April	Abril	Avril	April
5	May	Mayo	Mai	Mai
6	June	Junio	Juin	Juni
7	July	Julio	Juillet	Juli
8	August	Agosto	août	August
9	September	Septiembre	septembre	September
10	October	Octubre	octobre	Oktober
11	November	Noviembre	novembre	November
12	December	Diciembre	décembre	Dezember

17. Select **Translated Month Names Table** and configure all four translation columns to use **MonthNumber** as their sort column.

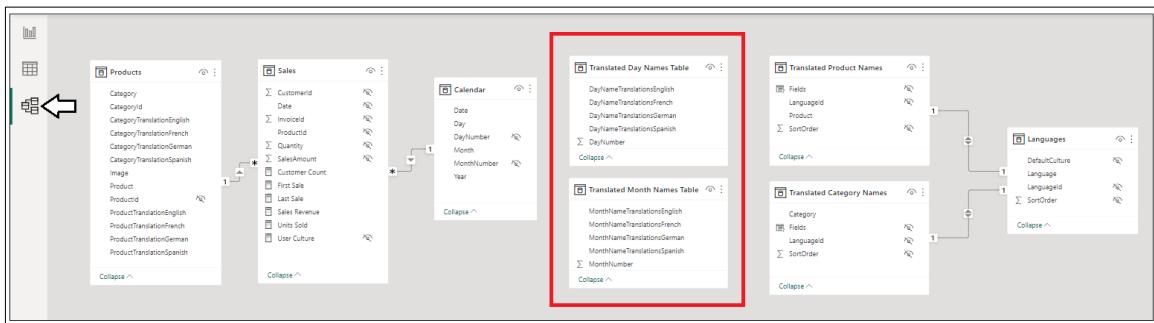
MonthNumber	MonthNameTranslationsEnglish	MonthNameTranslationsSpanish	MonthNameTranslationsFrench	MonthNameTranslationsGerman
1	January	Enero	Janvier	Januar
2	February	Febrero	Février	Februar
3	March	Marzo	Mars	März
4	April	Abril	Avril	April
5	May	Mayo	Mai	Mai
6	June	Junio	Juin	Juni
7	July	Julio	Juillet	Juli
8	August	Agosto	août	August
9	September	Septiembre	septembre	September
10	October	Octubre	octobre	Oktober
11	November	Noviembre	novembre	November
12	December	Diciembre	décembre	Dezember

18. Select **Translated Day Names Table** and configure all four translation columns to use **DayNumber** as their sort column.

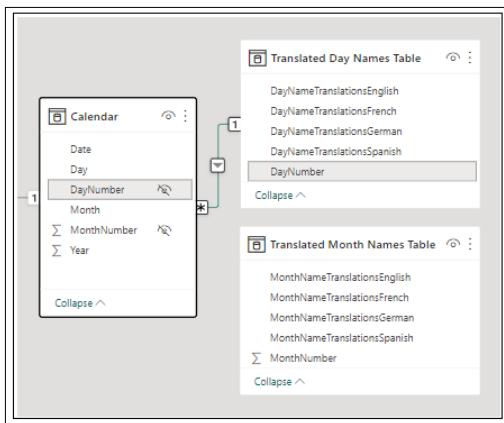
The screenshot shows the Power BI Column Tools pane. A red arrow points to the 'Sort by column' button in the top right. Another red arrow points to the 'DayNameTranslationsEnglish' column header. The 'DayNumber' column is highlighted in the list of columns.

19. Switch to **Model** view.

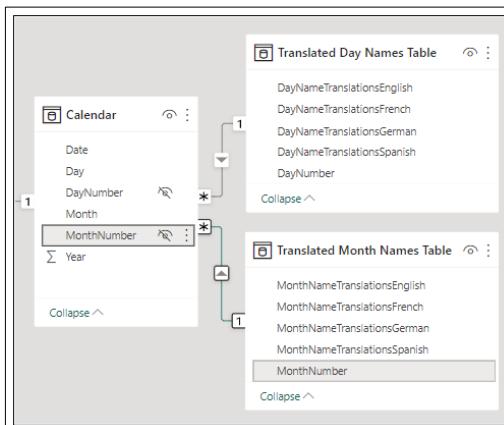
20. Reposition **Translated Day Names Table** and **Translated Month Names Table** just to the right of the **Calendar** table.



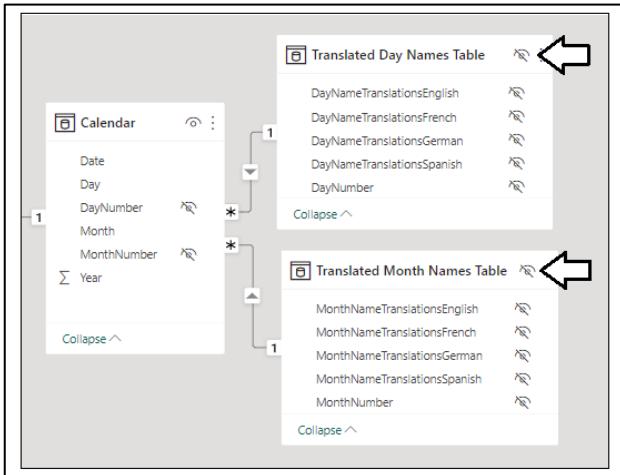
21. Create a one-to-many relationship between **Translated Day Names Table** and **Calendar** based on the **DayNumber** column.



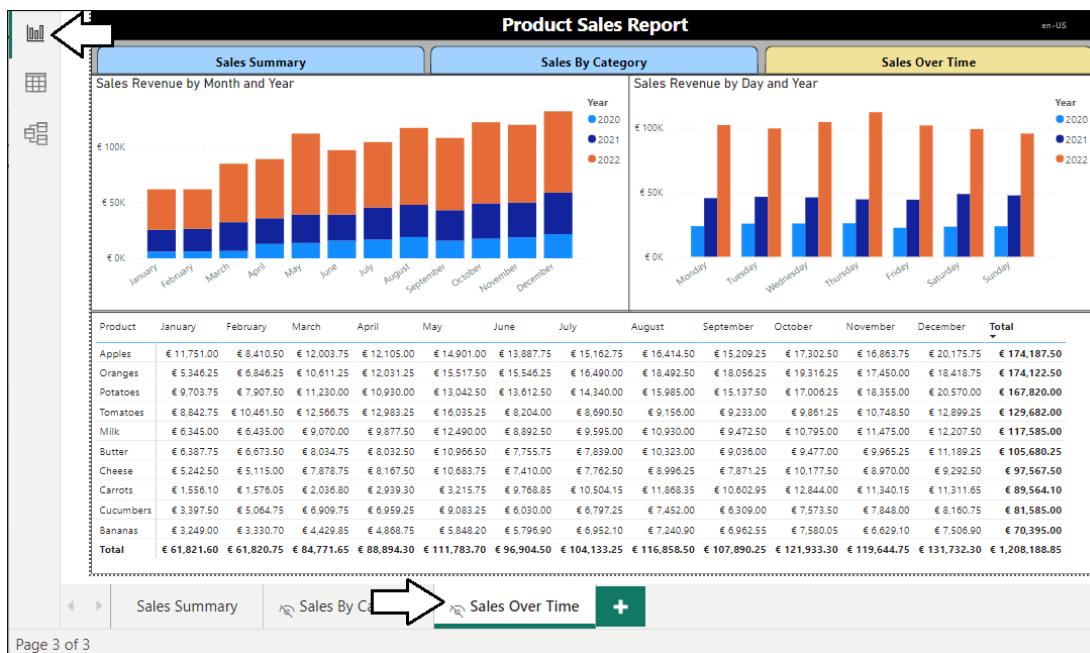
22. Create a one-to-many relationship between **Translated Month Names Table** and **Calendar** based on the **MonthNumber** column.



23. Hide **Translated Day Names Table** and **Translated Month Names Table** so they do not appear in **Report** view.

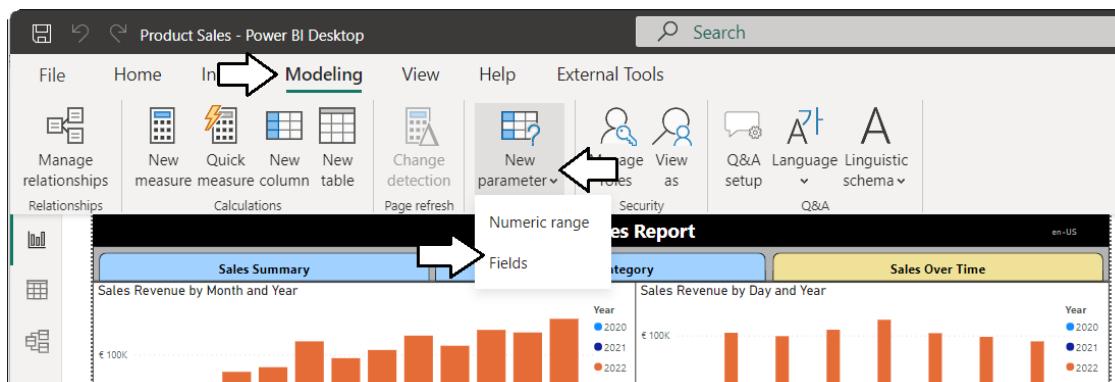


24. Switch to **Report** view and navigate to the **Sales Over Time** page.

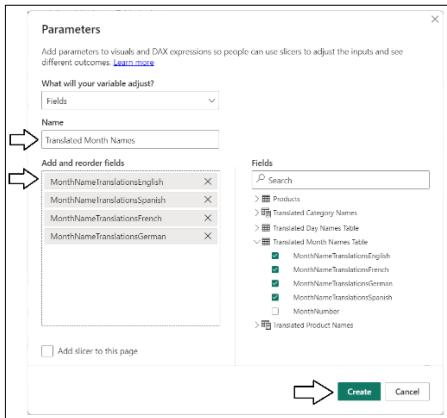


Now you need to create two new Field Parameters to implement data translations for month names and day names. However, you have created and configured two Field Parameters in the previous exercise so the steps should already be familiar.

25. Navigate to the **Modeling** tab and select **New parameter > Fields** to create a Field Parameter.



26. Name the Field Parameter Translated Month Names and add the translation columns from Translated Month Names Table.



27. Switch to Data view and select Translated Month Names in the Data pane so you can view its DAX expression.

28. Update the DAX expression for Translated Month Names with the following DAX code.

```
Translated Month Names = {
    {"Month", NAMEOF('Translated Month Names Table'[MonthNameTranslationsEnglish]), 0, "en"),
    {"Mes", NAMEOF('Translated Month Names Table'[MonthNameTranslationsSpanish]), 1, "es"),
    {"Mois", NAMEOF('Translated Month Names Table'[MonthNameTranslationsFrench]), 2, "fr"),
    {"Monat", NAMEOF('Translated Month Names Table'[MonthNameTranslationsGerman]), 3, "de")
}
```

29. After you have updated the DAX expression for Translated Month Names, you should see a new column named Value4.

30. Update the field names inside Translated Month Names to Month. Fields, SortOrder and LanguageId.

31. Configure the **Languageld** column to use **SortOrder** as its sort column.

The screenshot shows the Power BI Data view with the 'Translated Month Names' table selected. The 'Languageld' column is currently selected. A context menu is open over the 'Sort by column' dropdown, with the option 'SortOrder' highlighted. This indicates that the 'Languageld' column is being configured to use 'SortOrder' as its sort column.

Now there is one more Field Parameter you need to create for day names.

32. Switch to **Report** view.

33. Navigate to the **Modeling** tab and select **New parameter > Fields** to create a Field Parameter.

The screenshot shows the Power BI Desktop interface with the 'Modeling' tab selected. The 'Fields' button is highlighted. A context menu is open over the 'Fields' button, with the option 'New parameter' highlighted. This indicates that a new field parameter is being created.

34. Name the Field Parameter **Translated Day Names** and add the translation columns from **Translated Day Names Table**.

The screenshot shows the 'Parameters' dialog box in Power BI Desktop. The 'Name' field is set to 'Translated Day Names'. The 'Fields' list shows four entries: 'DayNameTranslationsEnglish', 'DayNameTranslationsSpanish', 'DayNameTranslationsFrench', and 'DayNameTranslationsGerman'. The 'Create' button at the bottom right is highlighted.

35. Switch to **Data** view and select **Translated Day Names** in the **Data** pane so you can view its DAX expression.

The screenshot shows the Power BI Data view with the 'Translated Day Names' table selected. The 'DayNameTranslationsEnglish' column is currently selected. The 'Data' pane on the right shows the table structure, including columns like 'DayNameTranslationsEnglish', 'DayNameTranslationsSpanish', 'DayNameTranslationsFrench', and 'DayNameTranslationsGerman'. This indicates that the 'Translated Day Names' table is being viewed in its raw data form.

36. Update the DAX expression for **Translated Day Names** with the following DAX code.

```
Translated Day Names =
    ("Day", NAMEOF('Translated Day Names Table'[DayNameTranslationsEnglish]), 0, "en"),
    ("Día", NAMEOF('Translated Day Names Table'[DayNameTranslationsSpanish]), 1, "es"),
    ("Jour", NAMEOF('Translated Day Names Table'[DayNameTranslationsFrench]), 2, "fr"),
    ("Tag", NAMEOF('Translated Day Names Table'[DayNameTranslationsGerman]), 3, "de")
}
```

37. After you have updated the DAX expression for **Translated Day Names**, you should see a new column named **Value4**.

	Translated Day Names	Translated Day Names Fields	Translated Day Names Order	Values
Day	'Translated Day Names Table'[DayNameTranslationsEnglish]		0	en
Dia	'Translated Day Names Table'[DayNameTranslationsSpanish]		1	es
Jour	'Translated Day Names Table'[DayNameTranslationsFrench]		2	fr
Tag	'Translated Day Names Table'[DayNameTranslationsGerman]		3	de

38. Update the field names inside **Translated Day Names** to **Day**, **SortOrder** and **Languageld**.

	Translated Day Names	Day	SortOrder	Languageld
Day	'Translated Day Names Table'[DayNameTranslationsEnglish]	0	en	
Dia	'Translated Day Names Table'[DayNameTranslationsSpanish]	1	es	
Jour	'Translated Day Names Table'[DayNameTranslationsFrench]	2	fr	
Tag	'Translated Day Names Table'[DayNameTranslationsGerman]	3	de	

39. Configure the **Languageld** column to use **SortOrder** as its sort column.

File Home Help External Tools Table tools Column tools

Name: Languageld Data type: Text

Structure: Day Fields SortOrder Languageld

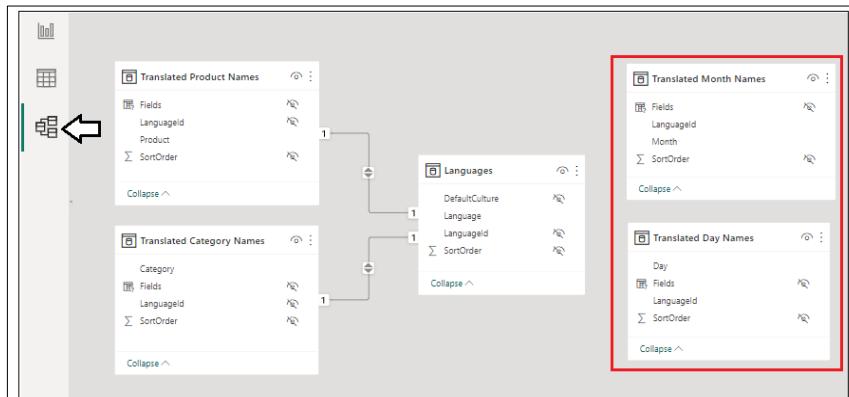
Properties: Summarization: Don't summarize Data category: Uncategorized

Sort by column: SortOrder

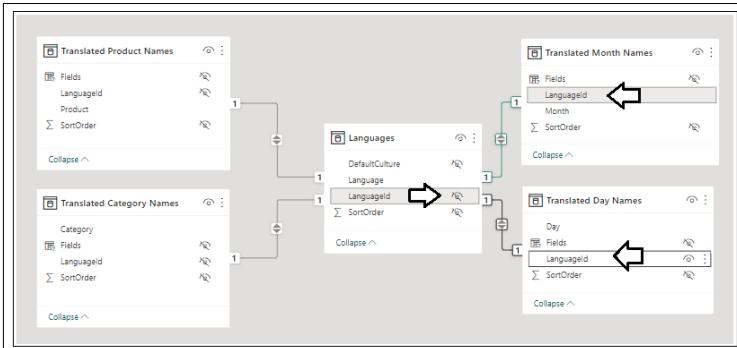
	Translated Day Names	Day	SortOrder	Languageld
Day	'Translated Day Names Table'[DayNameTranslationsEnglish]	0	en	
Dia	'Translated Day Names Table'[DayNameTranslationsSpanish]	1	es	
Jour	'Translated Day Names Table'[DayNameTranslationsFrench]	2	fr	
Tag	'Translated Day Names Table'[DayNameTranslationsGerman]	3	de	

40. Switch to **Model** view.

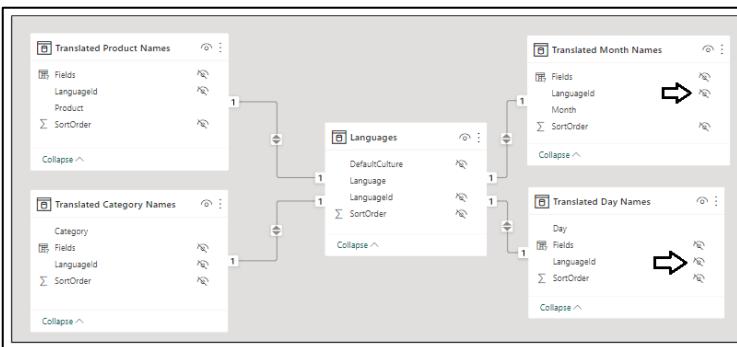
41. Reposition **Translated Month Names** and **Translated Day Names** so they appear just to the right of the **Languages** table.



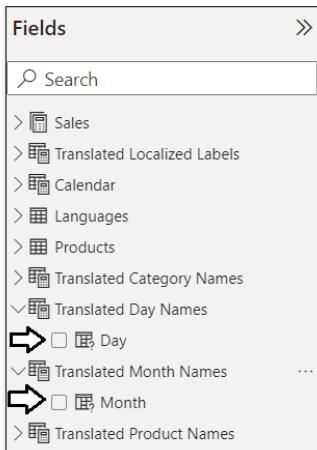
42. Create a one-to-one relationship between **Languages** and **Translated Month Names** based on the **LangaugeId** column.
 43. Create a one-to-one relationship between **Languages** and **Translated Day Names** based on the **LangaugeId** column.



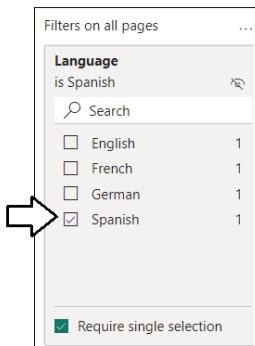
44. Hide the **LangaugeId** field from **Report view** in both **Translated Month Names** and **Translated Day Names**.



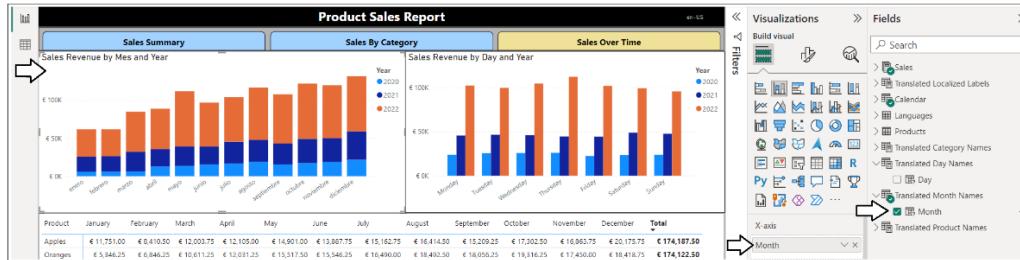
45. Switch to **Report view**, navigate to the **Sales Over Time** page and inspect the two new Field Parameters in the **Fields** pane.



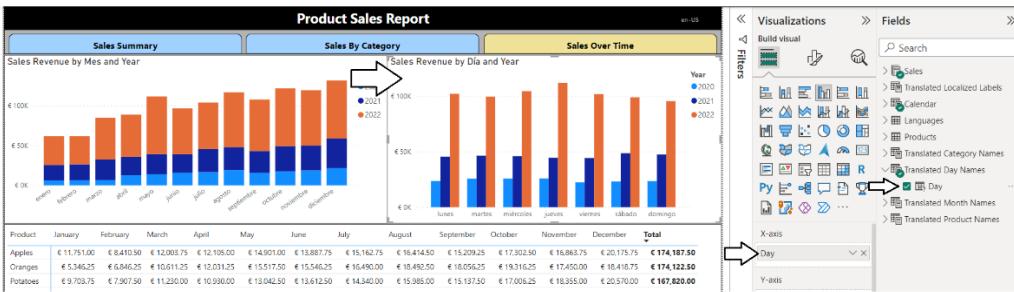
46. Display the **Filter** pane if it's not already showing and set the filter on the **Languages** table to **Spanish**.



47. Select the Column chart on the left which displays **Sales Revenue by Month and Year**.
 48. In the **X-axis** data role, replace **Month** from the **Calendar** table with **Month from Translated Month Names**.



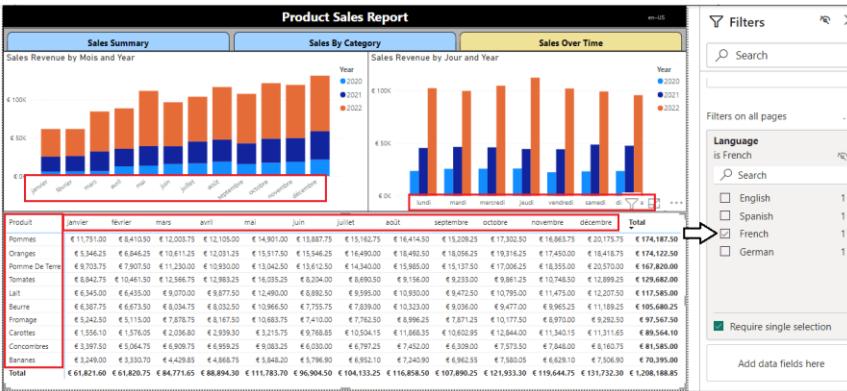
49. Select the Column chart on the right which displays **Sales Revenue by Day and Year**.
 50. In the **X-axis** data role, replace **Day** from the **Calendar** table with **Day from Translated Day Names**.



51. Select the Matrix visual at the bottom of the page which displays sales revenue by product and month.
 52. In the **Rows** data role, replace **Product** from the **Products** table with **Product** from **Translated Product Names**.
 53. In the **Columns** data role, replace **Month** from the **Calendar** table with **Month from Translated Month Names**.

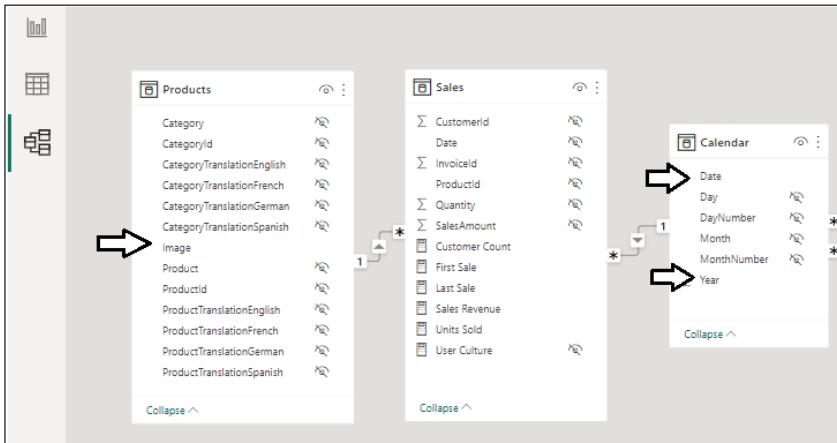


54. Now you can test the data translations for each language by switching the filter on the **Languages** table.



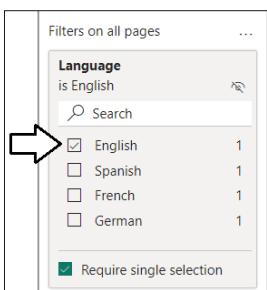
Now that you have completed the work to implement data translations, you will clean up the data model by hiding several columns that should not be shown to reporter authors in **Report** view.

55. Switch to **Model** view.
56. Hide every column in the **Products** table except for the **Image** column.
57. Hide the **Day** column and the **Month** column in the **Calendars** table so only the **Date** and **Year** columns are seen in **Report** view.



Now you need to prepare the report for deployment to the Power BI Service.

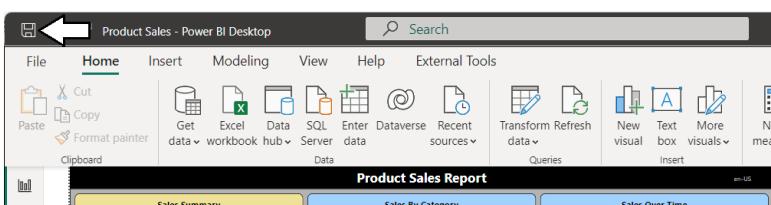
58. Set the report-level filter on the languages table to **English**.



59. Navigate back to the **Sales Summary** page and make it active before saving to ensure this page is the startup page.

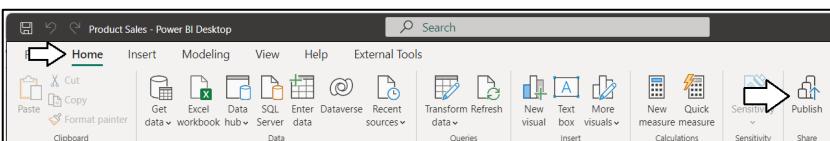


60. Save your work by clicking the **Save** button.



Now, it's time once again to test your work in the Power BI Service.

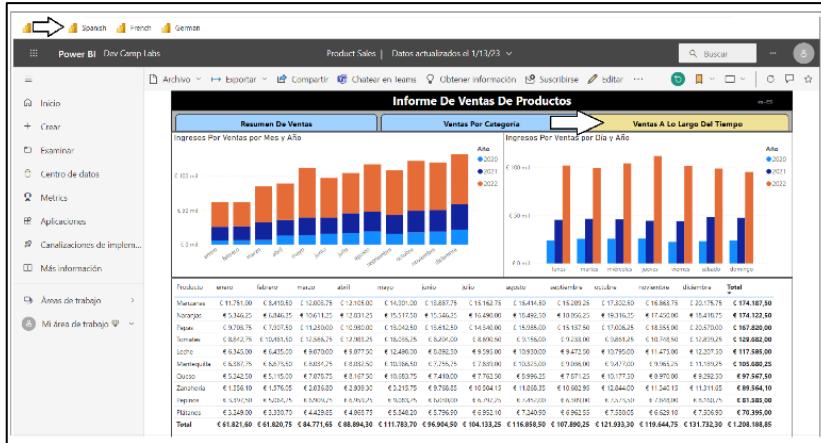
61. Publish the **Product Sales** project to push your changes to the project's translations to the Power BI Service.



62. When prompted by the **Replace this dataset?** dialog, click the **Replace** button to continue.

63. Once you see **Success!**, click **Open 'Product Sales'** in **Power BI** to view the report in the Power BI Service.

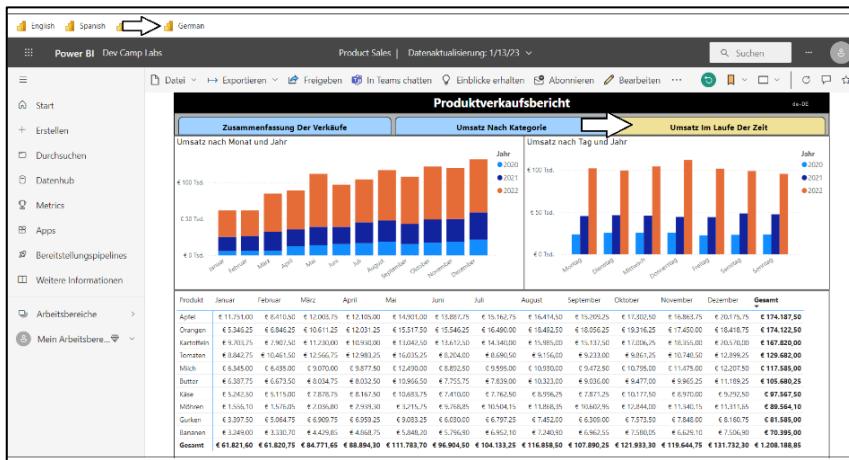
64. The report should load as normal showing all text in English at first.
65. Click on the browser book for **Spanish**. Once the report has loaded, navigate to the **Sales Over Time** page.
66. Verify that the data translations are loading Spanish names for the names of months and days in all three visuals.



67. Click on the browser book for **French**. Once the report has loaded, navigate to the **Sales Over Time** page.
68. Verify that the data translations are loading French names for the names of months and days in all three visuals.



69. Click on the browser book for **German**. Once the report has loaded, navigate to the **Sales Over Time** page.
70. Verify that the data translations are loading German names for the names of months and days in all three visuals.



Congratulations. You have now completed this hands-on lab and you can now begin using your newfound localization skills in working with translations and building multi-language reports on your own PBIX projects.