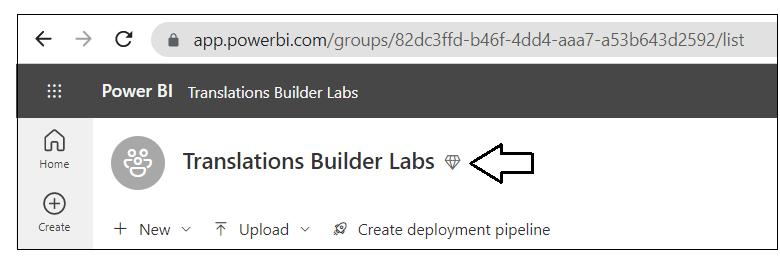
## **Lab 01: Getting Started with Translation Builder**

**Overview**: In this lab, you will learn how Translations Builder together with Power BI Desktop to build multi-language reports for Power BI. You will start by downloading a PBIX project file named **TB-Lab01.pbix** and opening it in Power BI Desktop. Once you have opened **TB-Lab01.pbix,** you will launch Translations Builder and practice by moving through the steps to add metadata translations and report label translations. At various milestones in this lab, you will be required to publish **TB-Lab01.pbix** from Power BI Desktop to the Power BI Service so you can test how the translations you’ve added appear to users when loaded using different languages and cultures.

**Prerequisite 1**: To complete this lab, you will need a Power BI workspace where you have appropriate permissions to publish project 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. This is required because Power BI translations do not load properly for reports tht are running in workspaces associated with the shared capacity.



**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](https://github.com/PowerBiDevCamp/TranslationsBuilder/blob/main/Docs/Installation%20Guide.md) before continuing with these lab exercises.

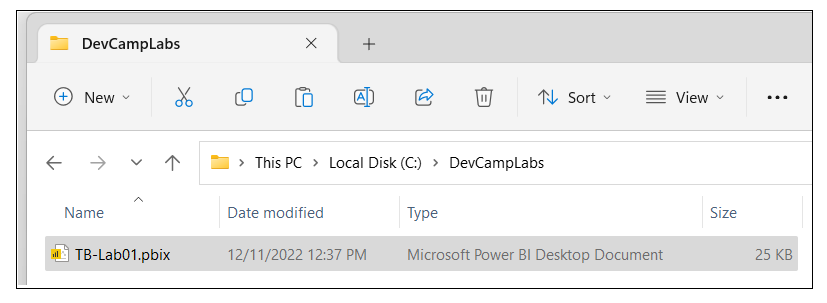
### Exercise 1: Creating and Testing Metadata Translations

In this exercise, you will begin by downloading a PBIX file which already has a simple data model and a single page report. You will then publish the project to the Power BI Service to set up a process whereby you can test out how your translations appear to report consumers.

1. Download the PBIX starter file named **TB-Lab01.pbix** from the following link

<https://github.com/PowerBiDevCamp/TranslationsBuilder/raw/main/Labs/StarterFiles/TB-Lab01.pbix>

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

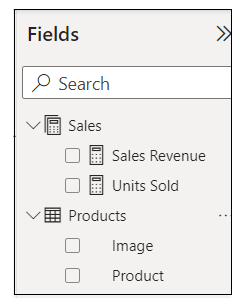


1. Copy **TB-Lab01.pbix** into the lab folder and then open it in Power BI Desktop to examine the report inside.

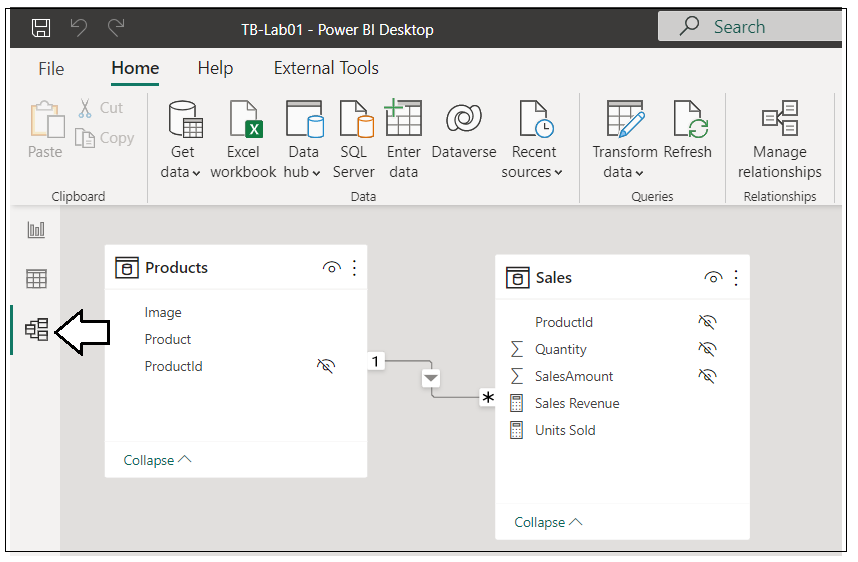
Graphical user interface, application, table

Description automatically generated

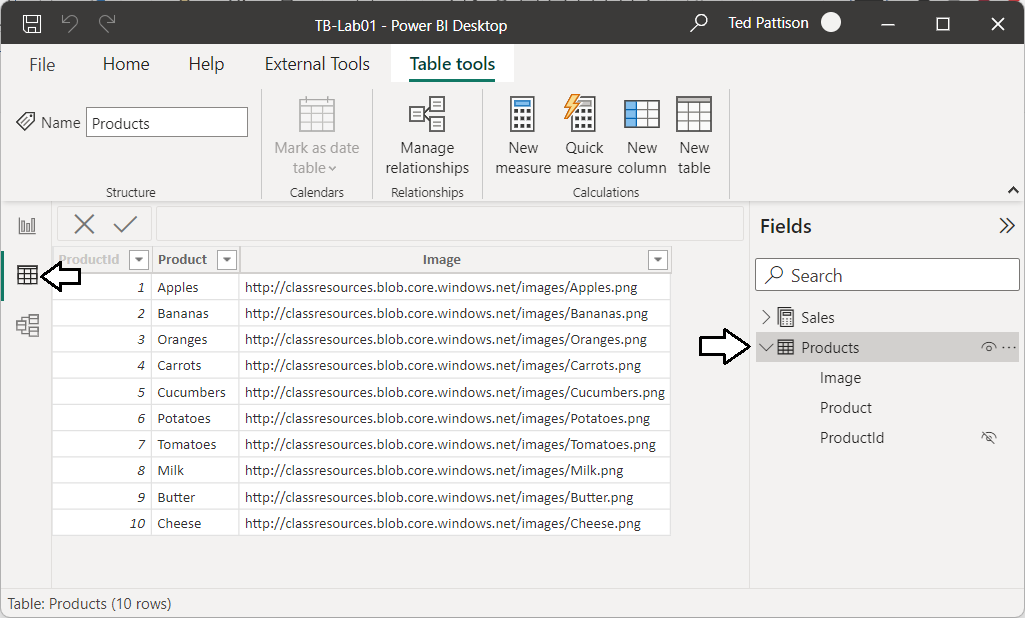
1. While in Report view, examine the **Fields** list to see the tables, columns and measure that are not hidden.



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

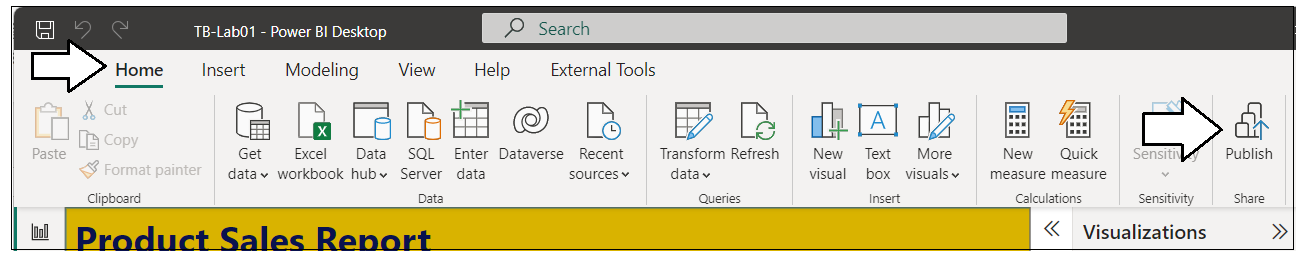


1. Navigate to Data view and examine the rows of the **Products** table.

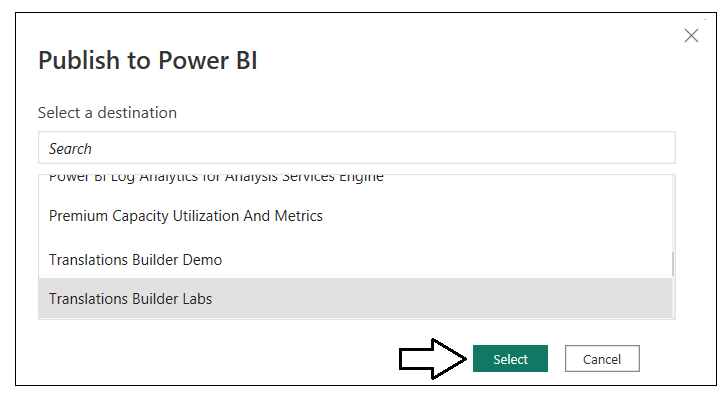


Now you are going to publish the TB-Lab01.pbix project to your test workspace in the Power BI Service.

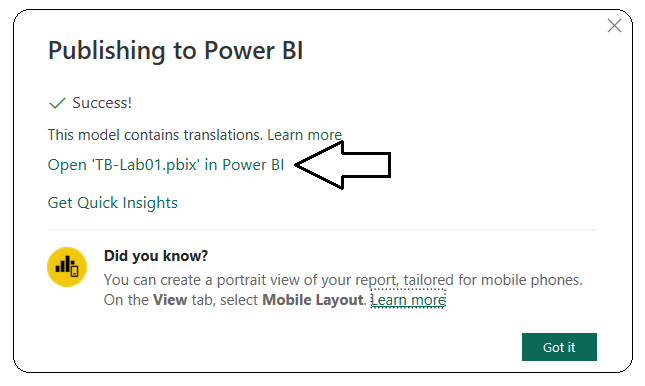
1. Navigate to the **Home** tab and then click the **Publish** button.



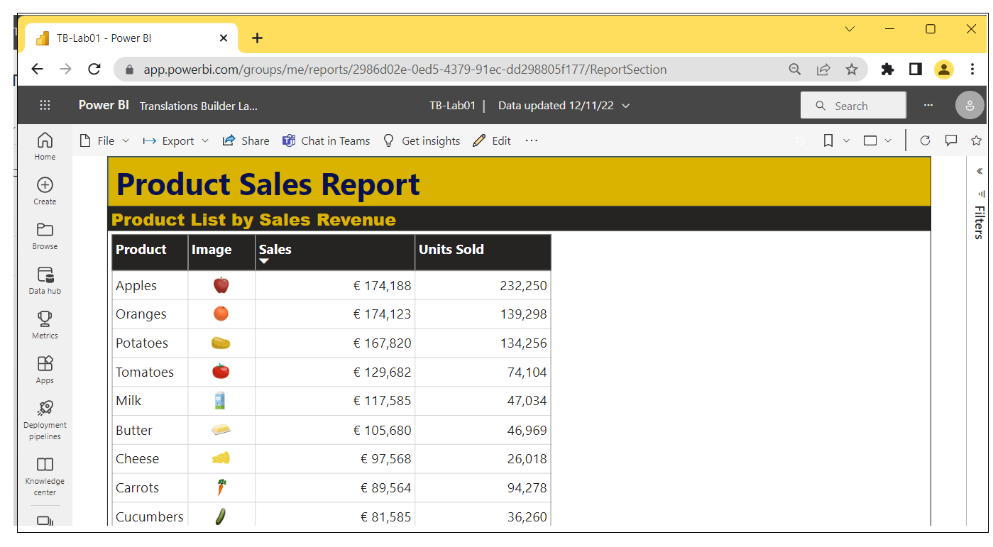
1. When prompted by the **Publish to Power BI** dialog, choose your test workspace and then click **Select**.



1. Once you see **Success!**, click **Open ‘TB-Lab01.pbx’ in Power BI** to view the report in the Power BI Service.



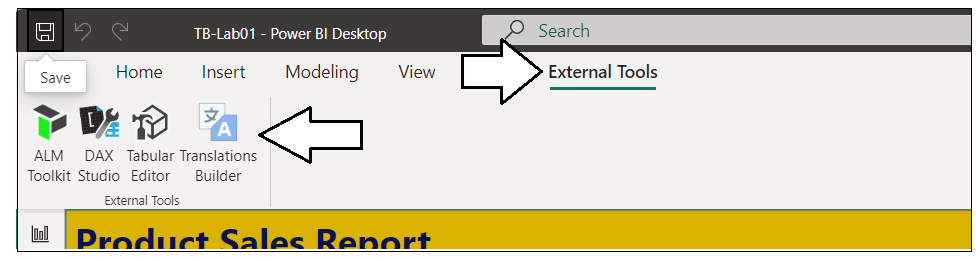
1. The report named **TB-Lab01** should appear like the report in the screenshot below.



Now you are setup to follow this workflow: (1) make changes in Power BI Desktop, (2) publish the project, (3) check your work

It’s time to begin using Translations Builder to begin adding translations to the **TB-Lab01.pbix** project.

1. Return to Power BI Desktop, navigate to the **External Tools** tab and launch **Translations Builder**.

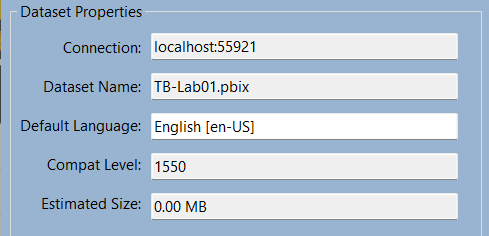


1. Translations Builder should start and load the data model for **TB-Lab01.pbix** as shown in the following screenshot.

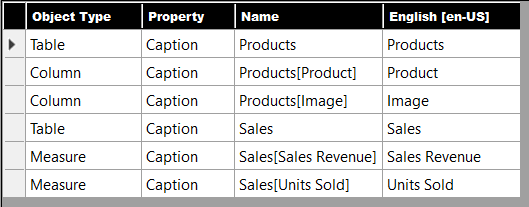
Graphical user interface, application

Description automatically generated

1. The Dataset properties section provides details about the connection and the PBIX project file.

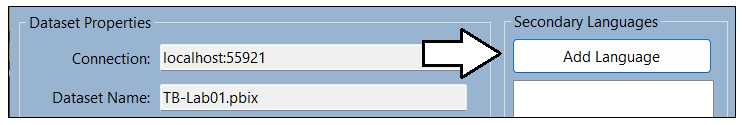


1. The grid on the bottom of the page display a row for each non-hidden dataset object in the data model.

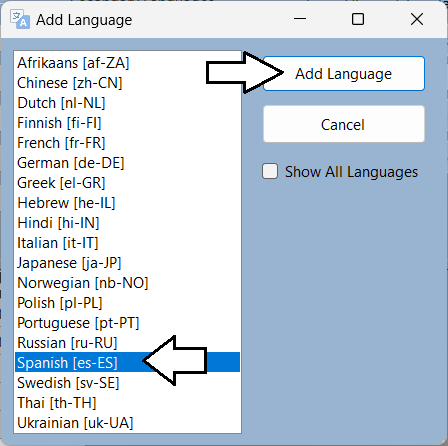


Tables, columns and measures that are hidden from report view in the data model are not displayed. You don’t need to translate them.

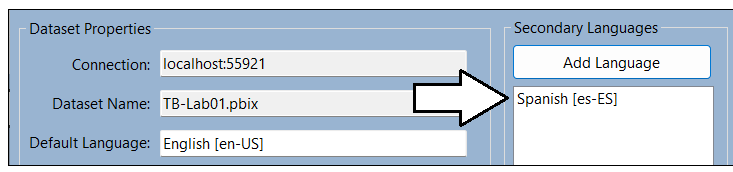
1. Click the **Add Language** button to add your first secondary language.



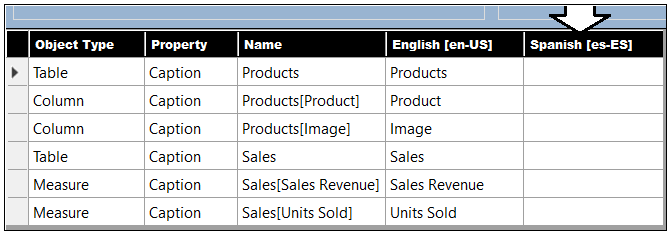
1. Select **Spanish [es-ES]** and click **Add Language**.



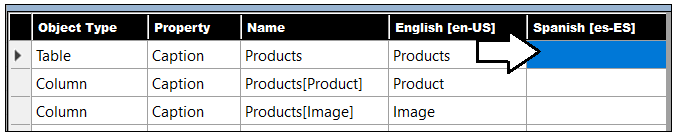
1. You should now see that **Spanish [es-ES]** appears the first language in the **Secondary Languages** list.



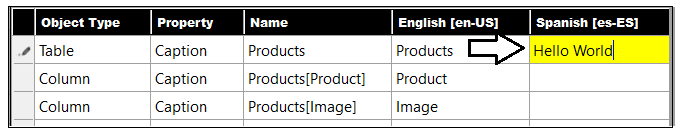
1. You will also notice that a new column has been added for Spanish translations.



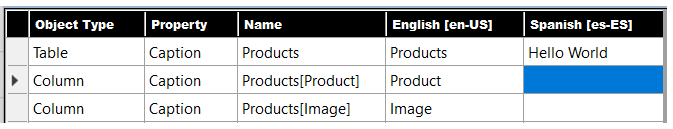
1. Locate the row for the **Products** table and click on the cell for the **Spanish** column. It should turn blue when selected.



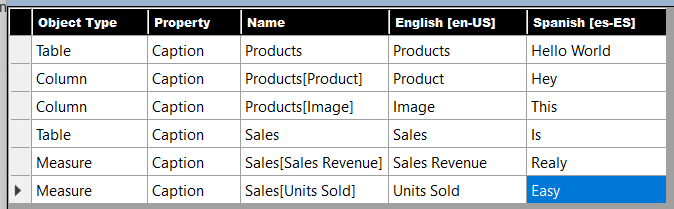
1. Type Hello World. You should see that you can start typing in a translation select that is selected to add a translation.



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

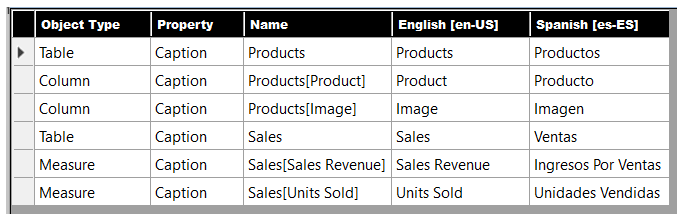


1. Now, type more text and press **ENTER** repeatedly to experiment quickly adding text to all cells in the Spanish column.

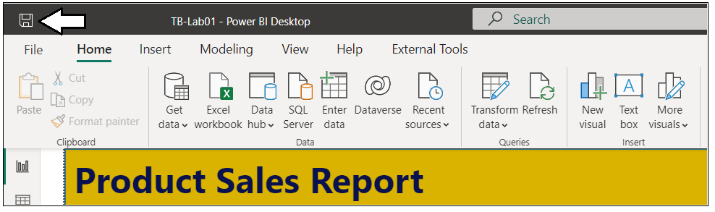


The point of the last few steps has been for you to become comfortable with the translation editing experience. You can see the grid provides an editing experience similar to working with Excel. You can even use the **{F2}** key to move a cell with content into edit mode.

1. Now edit the translations in the Spanish columns with better translated values. Use the following translations.
   1. For the **Products** table, enter the Spanish translation of **Productos**.
   2. For the **Product** column, enter the Spanish translation of **Producto**
   3. For the **Image** column, enter the Spanish translation of **Imagen**.
   4. For the **Sales** table, enter the Spanish translation of **Ventas**.
   5. For the **Sales Revenue** measure, enter the Spanish translation of **Ingresos Por Ventas**.
   6. For the **Units Sold** measure, , enter the Spanish translation of **Unidades Vendidas**.
2. When you are done with your edits, the Spanish translations should match the following screenshot.

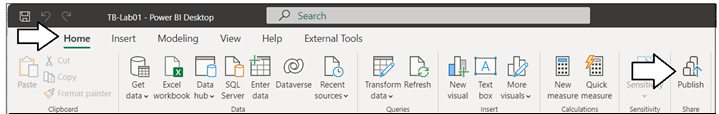


1. Return to the **TB-Lab01.pbix** project in Power BI Desktop and save your work by clicking the **Save** button.

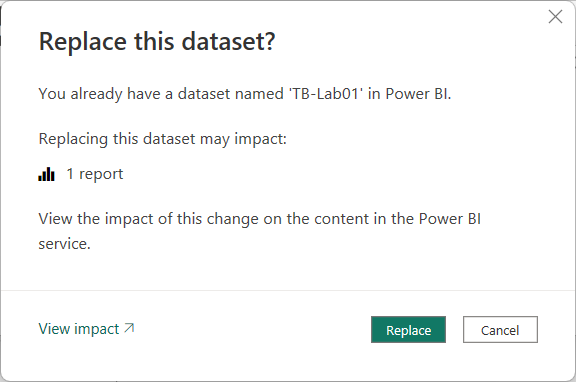


It’s easy to forget to save your changes in Power BI Desktop. Don’t forget any changes made by Translations Builder are just made to the data model loaded in memory. None of your changes are saved back to the PBIX project file until you save in Power BI Desktop.

1. Publish the **TB-Lab01.pbix** project to push the changes to the project’s translations to the Power BI Service.



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



1. Once you see **Success!**, click **Open ‘TB-Lab01.pbx’ in Power BI** to view the report in the Power BI Service.

Graphical user interface, text, application, email

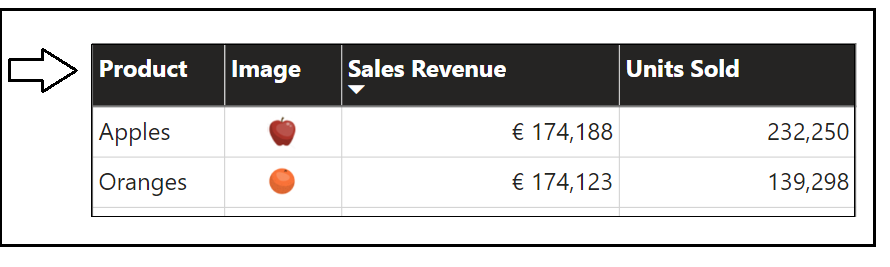
Description automatically generated

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

Table

Description automatically generated

1. In particular, inspect the table visual columns which displays column names and measure names in English.

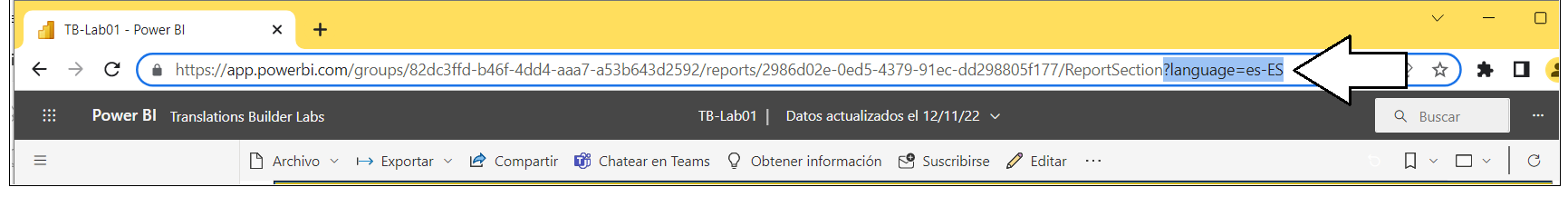


Now, it’s time to test your translations for the first time by using the **language** query string parameter to load the report.

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

/?language=es-ES

1. You should see the **language** query string parameter accept by the browser as it reloads the report.



1. You should see the Power BI Service UI has switch from English to Spanish
2. You should also see columns headings with column names and measure names are also displayed in Spanish.



You have now successfully added the metadata translations to display this report in both English and Spanish.

### Exercise 2: Generating Machine Translations

In this lab you will configure Translations Builder’s support to automatically provide machine translations using the Azure Translation Service. While machine translations might not prove good enough to use in all production scenarios, they do provide a great first step in generating translations for testing and getting something into production sooner.

**Key**: a75b371ce1fc402ca84a05732cfcff27

**Location**: eastus2

xx

### Exercise 3: Creating and Testing Report Label Translations

In this exercise ddddd

1. Rub your belly
2. Tap your forehead