# Contents

# 1 Introduction

ACTS Accelerators includes methods and tools to support multiple languages in Power BI reports. These methods enable users to modify labels on visuals, add languages, and manage these labels and languages to tailor reports in specific implementations. The recommended approach applies Microsoft's Translations Builder and related methods to enable multiple languages for text in Power BI reports, including titles, subtitles, shapes, text boxes, axes, and slicers. The methods to use depend on the visual and are described in the three sections below, which provide sufficient information to meet most Power BI multi-language support requirements. In addition to these, Translations Builder supports other methods including those for management of metadata such as table titles and column header labels and generating machine translations for table data. Refer to Planning translation for multiple-language reports in Power BI for information on these additional capabilities. An accompanying Power BI example report 'GenericProcurement-MultiLanugaes.pbix' implements the methods described in this document.

## 1.1 Title and Subtitle Text

Once Translations Builder is installed (see Translations Builder Installation Guide) it can be used to create a set of Measures for setting the text in multiple languages in Titles and Subtitles of Power BI visuals including text boxes, buttons, and shapes. Translations Builder provides a convenient user experience for selecting the languages required, entering the names of Measures for each required label, entering translations for the language(s) you want to support, and automatically creating the new Measures in Power BI.

Translation Builder uses a table named 'Translated Localized Labels' containing all the Measures you create. For example, a Power BI Shape visual can be used to create a label for a report home page. Users can add the Shape to a report page, turn the Title for the shape to On, then enter a text for the Title as shown at top left in Figure 1.

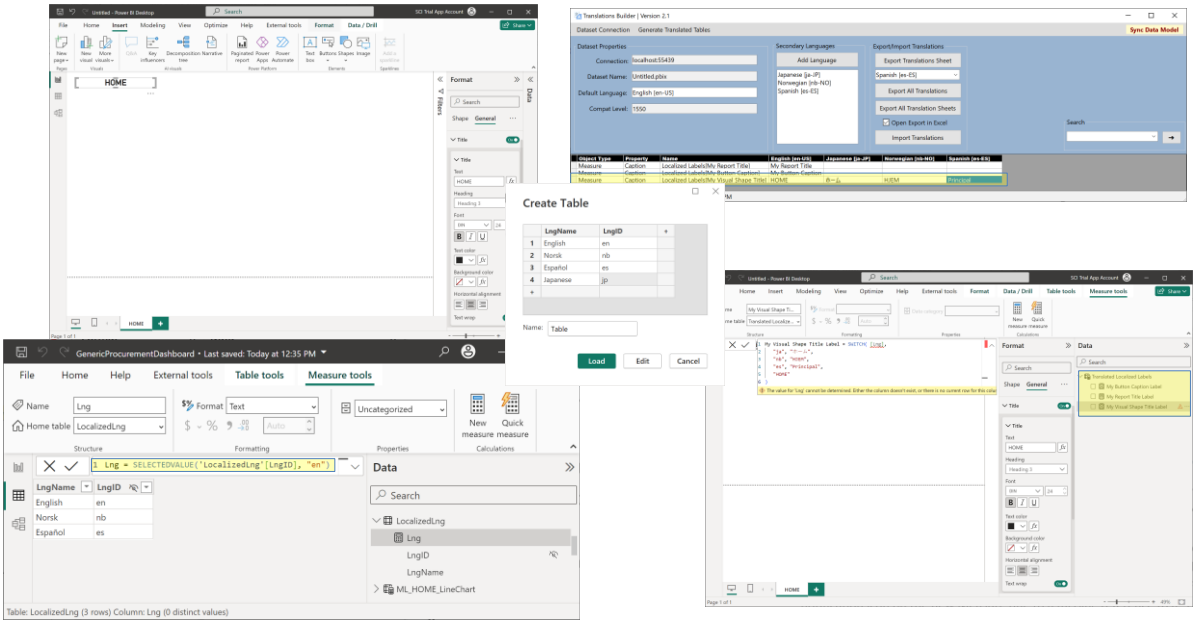



*Figure 1. Multiple language support for Shape visuals.*

To support multiple languages for this Shape, start Translations Builder, select 'Generate Translated Tables->Add Labels to Localized Labels Table…', then enter 'My Visual Shape Title' in the 'Add Localized Labels" dialog and click 'Add Label'. This will create an entry for a new Measure with columns on the right for each of the languages you selected in the Add Language group as shown at the top right of Figure 1. You can double-click on any of the cells to the right of your new Measure to add the translation. Finally, select 'Generate Translated Tables->Generate Translated Localized Labels Table' to create this table in Power BI as shown highlighted in the lower right of Figure 1. Note the error notification icon on the new Measure (My Visual Shape Title Label) due to a missing reference to an "Lng" measure.

To correct the error due to the missing reference to an "Lng" measure, create a new table named 'LocalizedLng' like that shown at lower left in Figure 1, ensuring that there are 'LngName' and 'LngID' rows for each of the languages to be supported (use the Home ribbon, Enter data option to enter the table manually as shown in the center of Figure 1, and remember to rename the table 'LocalizedLng'). Next, create for this table a new Measure named 'Lng' with the DAX expression shown highlighted at lower left in Figure 1 ( `Lng = SELECTEDVALUE('LocalizedLng'[LngID], "en")` ). Once this table and measure are created, the error messages and icons shown in the image on the lower-right side of Figure 1 automatically will be removed. This completes the setup of the measure to support multiple languages for the Shape. The final steps are to reference the new measure in the Shape's Title and create a Slicer to be able to change the language (see section 1.3 Slicer Values for supporting multi-languages in slicers).

Measures listed in the Translated Localized Labels table can be used with Power BI's Conditional Formatting option for any visual with a Title. Select the created Shape, open the Format General panel and click on the Conditional Formatting button as shown at top-center in Figure 2. Finally, add a Slicer visual to the page and set its Field property to 'LngName', which is a column with the names of the language from which a selection can be made. Selecting a different language with the Slicer will cause the Shape title to change to the selected language.

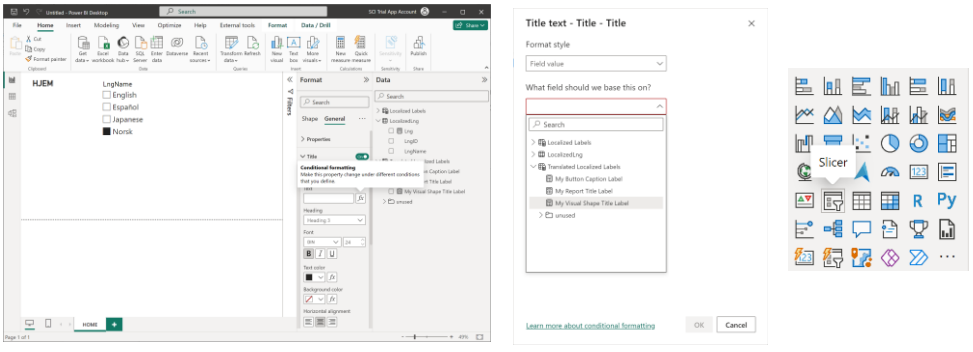



*Figure 2. Setting Conditional Formatting for the Shape Title.*

## 1.2 Chart Axes and Table Column Headers

Axis and column header titles do not support conditional formatting of the text as described in the section above. The Title text for these visuals is set to 'Auto' unless users enter a static label. Multi-

language support in Power BI for these text labels can be achieved using Power BI's Field Parameters described in this section. Translations Builder is not used in this method.

Field Parameters are a practical way to enable users to create sliders that will change the text as well as the data displayed in a table or chart. Field Parameters are calculated tables that include alternative labels, a pointer to the data fields associated with the label, and an optional integer order of the fields used in listings, for example, the order in which values are displayed in a slicer. For multi-language support, an 'LngID' column is added to this calculated table.

### 1.2.1 Chart Axes

As an example, to add multi-language support for the y axis of a chart, create a field parameter that points to the data to display as shown in Figure 3. In the top-left of this figure the Modeling ribbon is selected and the 'New parameter->Field' item has been selected to display the Parameters dialog at right. This parameter labelled 'ML_HOME_LineChart' specifies the 'RF_TendersCount' measure (this will return the count of review flags for tenders at the time specified on the x axis) of the 'TendersWithFlags' table as the field, and when it is created the calculated table will be displayed in the Data pane similar to that shown highlighted in yellow in Figure 3.

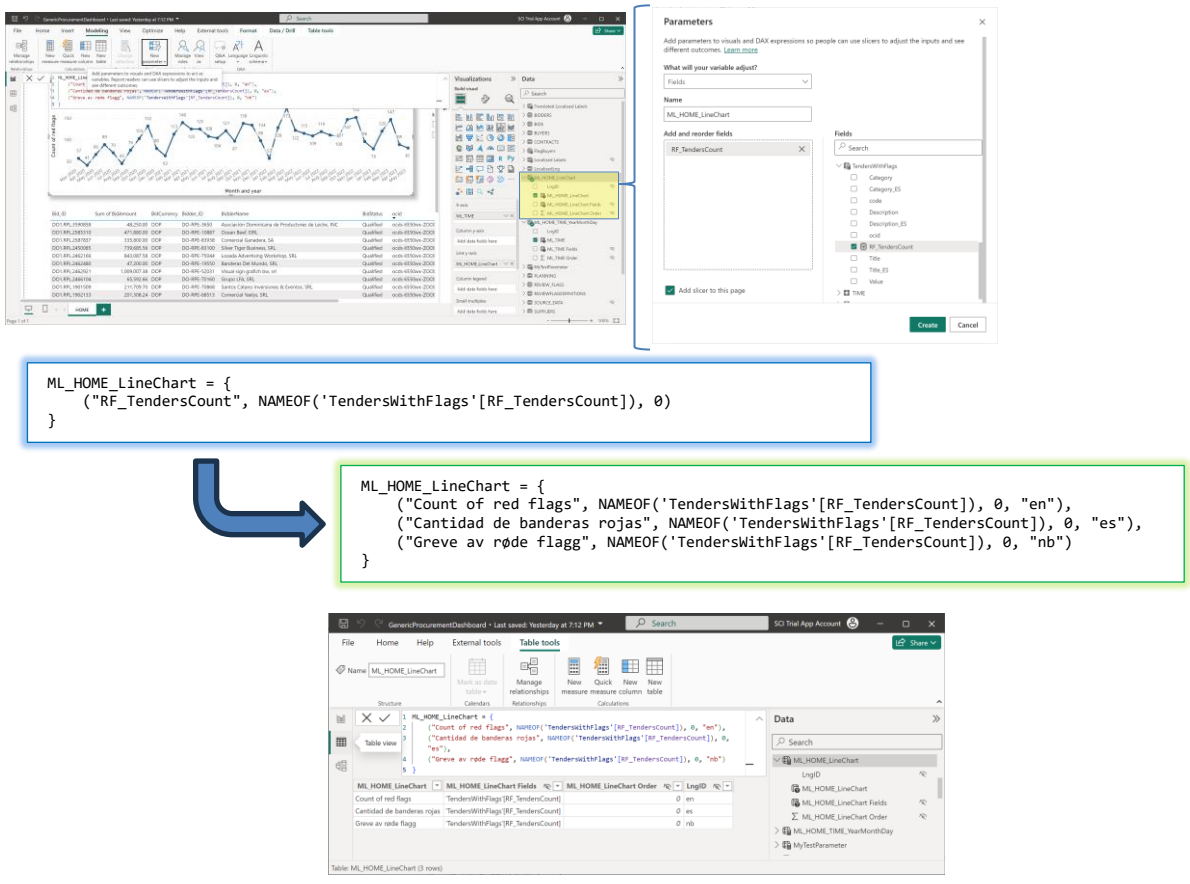



*Figure 3. Creating a Field Parmeter (a calculated table) for the X-axis data in a chart.*

Inspecting the definition of the 'ML_HOME_LineChart' calculated table shows that it has the form outlined in blue in Figure 3; however, after adding an 'LngID' column to the calculated table, and editing the DAX expression to that outlined in green in Figure 3, the field parameter now supports multiple languages using two-letter codes to designate each language. These codes must be the same codes used in the 'LocalizedLng' table created previously and shown at center in Figure 1.

The semantic (data) model in Power BI must include a relationship between the 'LngID' in the 'LocalizedLng' table to the LngID in each field parameter (a calculated table) created. This relationship is created in the Model view by editing the relationship as shown in Figure 4.

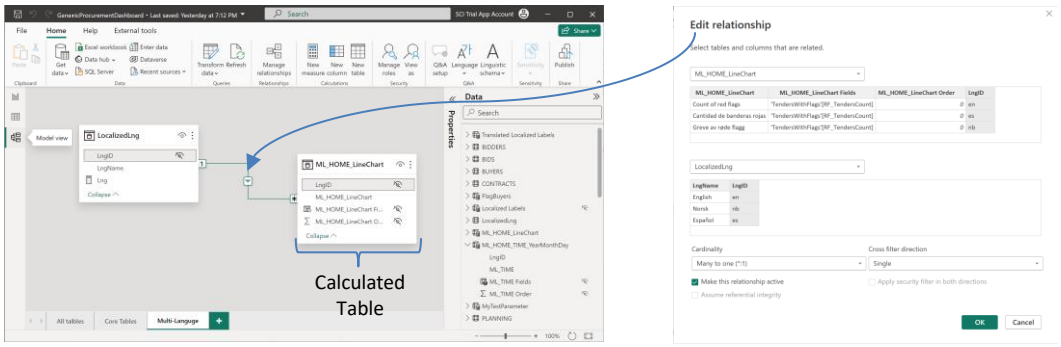



*Figure 4. Editing a relationship between 'LngID' in 'LocalizedLng' table and 'ML_HOME_LineChart' calculated table (a field parameter).*

Finally, by setting the 'Line y-axis' in the chart to field parameter 'ML_HOME_LineChart' the chart will display both the data and the label for the currently selected language using the same Slicer shown in Figure 2. This procedure can be used to set any x or y axis on charts. Chart Titles and Subtitles can be set using the methods described in section *1.1 Title and Subtitle Text*.

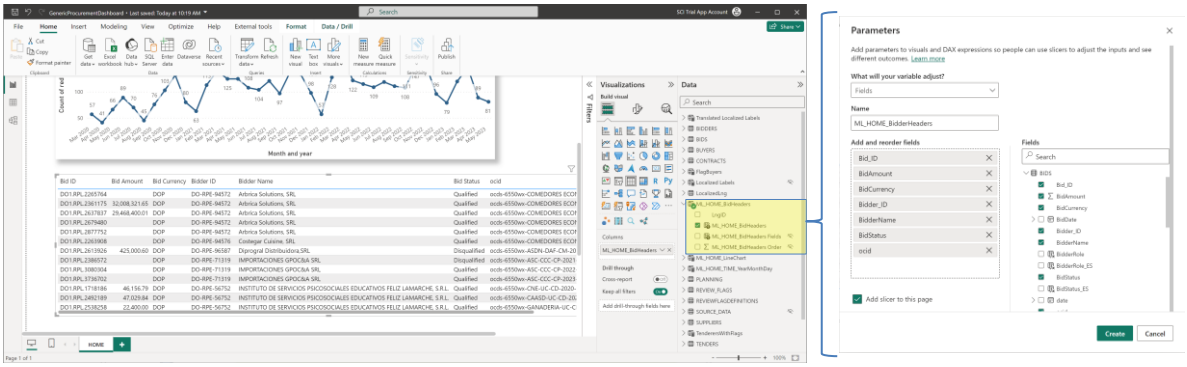
### 1.2.2 Table Column Headers

Multi-language support for table column headers is handled in the same way as chart axes by:

a) creating a field parameter referencing all header columns,
b) editing the calculated table DAX expression to include the text in the languages you want to support,
c) replacing the old DAX expression definition with the new, edited definition for the calculated table and changing the name of the last column to 'LngID', and,
d) creating a relationship between 'LngID' in the new calculated table and the 'LocalizedLng' table.

For example, multiple languages for header text of the 'Bids' table shown in Figure 5 are enabled by creating field parameter 'ML_HOME_BidHeaders' from the Field Parameter dialog shown at top right. Inspecting the definition of the created calculated table shows that the automatically generated DAX is that shown highlighted in blue in Figure 5 and it does not contain values for each language or the 'LngID' values. First, edit the DAX expression to a form like that highlighted in green in Figure 5, replace the original DAX expression definition with the new definition, click the check mark to accept the new expression and re-create the table, then change the name of the new column to 'LngID' (to do tis,

double-click on the column header then enter the new header label). The DAX expression and resulting table are shown in Figure 6.




```
ML_HOME_BidHeaders = {
    ("Bid_ID", NAMEOF('BIDS'[Bid_ID]), 0),
    ("BidAmount", NAMEOF('BIDS'[BidAmount]), 1),
    ("BidCurrency", NAMEOF('BIDS'[BidCurrency]), 2),
    ("Bidder_ID", NAMEOF('BIDS'[Bidder_ID]), 3),
    ("BidderName", NAMEOF('BIDS'[BidderName]), 4),
    ("BidStatus", NAMEOF('BIDS'[BidStatus]), 5),
    ("ocid", NAMEOF('BIDS'[ocid]), 6)
}
```

```
ML_HOME_BidHeaders = {
    ("Bid ID", NAMEOF('BIDS'[Bid_ID]), 0, "en"),
        ("ID de oferta", NAMEOF('BIDS'[Bid_ID]), 0, "es"),
        ("Bud-ID", NAMEOF('BIDS'[Bid_ID]), 0, "nb"),
    ("Bid Amount", NAMEOF('BIDS'[BidAmount]), 1, "en"),
        ("Monto de la oferta", NAMEOF('BIDS'[BidAmount]), 1, "es"),
        ("Budbeløp", NAMEOF('BIDS'[BidAmount]), 1, "nb"),
    ("Bid Currency", NAMEOF('BIDS'[BidCurrency]), 2, "en"),
        ("Moneda de oferta", NAMEOF('BIDS'[BidCurrency]), 2, "es"),
        ("Budvaluta", NAMEOF('BIDS'[BidCurrency]), 2, "nb"),
    ("Bidder ID", NAMEOF('BIDS'[Bidder_ID]), 3, "en"),
        ("ID del postor", NAMEOF('BIDS'[Bidder_ID]), 3, "es"),
        ("Budgiver-ID", NAMEOF('BIDS'[Bidder_ID]), 3, "nb"),
    ("Bidder Name", NAMEOF('BIDS'[BidderName]), 4, "en"),
        ("Nombre del postor", NAMEOF('BIDS'[BidderName]), 4, "es"),
        ("Budgivers navn", NAMEOF('BIDS'[BidderName]), 4, "nb"),
    ("Bid Status", NAMEOF('BIDS'[BidStatus]), 5, "en"),
        ("Estado de la oferta", NAMEOF('BIDS'[BidStatus]), 5, "es"),
        ("Budstatus", NAMEOF('BIDS'[BidStatus]), 5, "nb"),
    ("ocid", NAMEOF('BIDS'[ocid]), 6, "en"),
        ("ocid", NAMEOF('BIDS'[ocid]), 6, "es"),
        ("ocid", NAMEOF('BIDS'[ocid]), 6, "nb")
}
```

*Figure 5. Adding multi-language support for table column headers.*

Finally, create the relationship between 'LngID' in the new calculated table and the 'LocalizedLng' table as shown in Figure 4. The new field parameter column name can now replace the list of columns normally specified in the Columns specification for the table as shown at top left of Figure 5 and when the language selection Slicer is changed, the column headers will also change to the selected language.

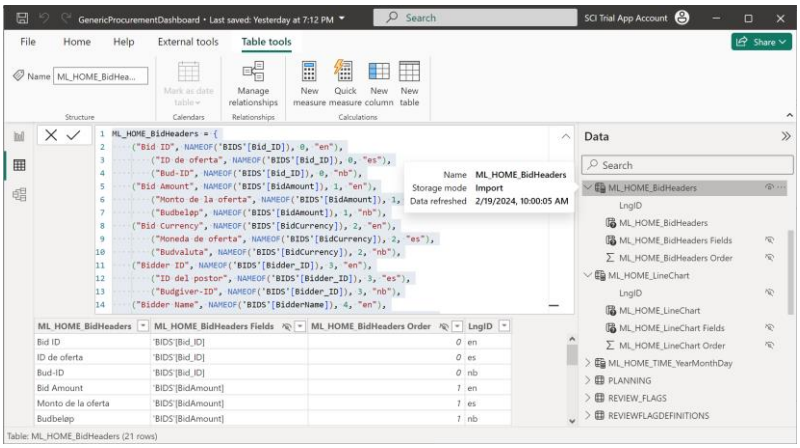



*Figure 6. Editing the Field Parameter calculated table to support multi-languages for table headers.*

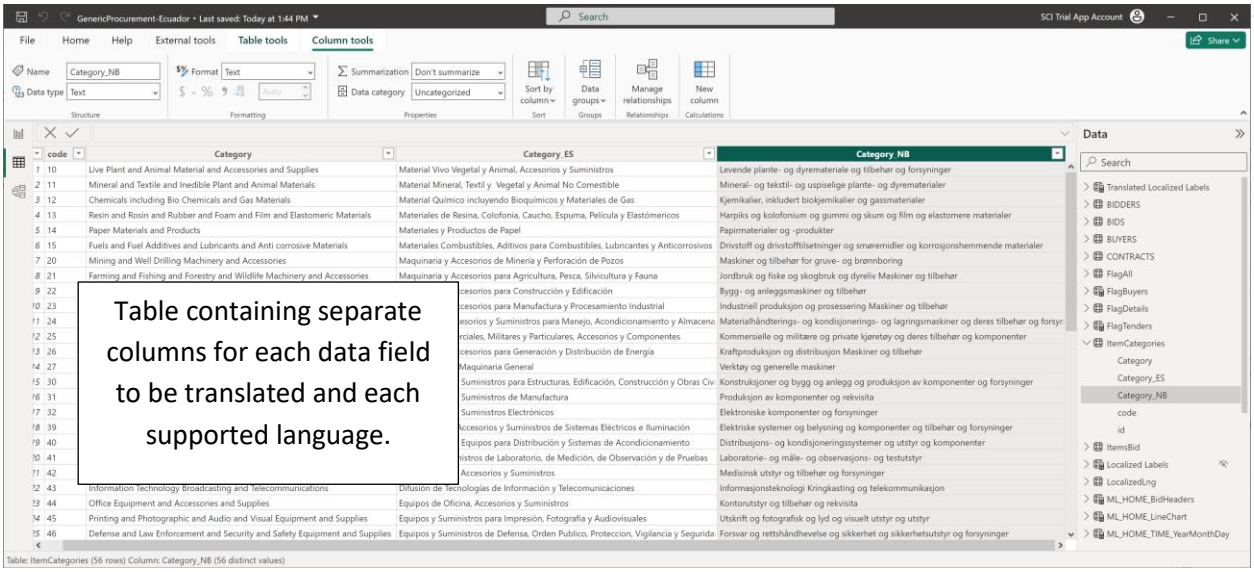
## 1.3 Slicer Values

Slicers obtain their list values from data in columns of a table. Table data translation is a challenging task as it requires either translation, optionally automated, in the source tables, or similar translation in Power BI or other data consumer application. Refer to Planning translation for multiple-language reports in Power BI for information on data translation capabilities. The methods in this section make use of Power BI field parameters described in section 1.2 and methods specific to Slicer visuals[1].

The first step in implementing multi-language support for Slicers is to import or define (with DAX) tables that contain translations of the items that will be contained in the Slicer list. If muti-language support is provided in the source data tables, then these tables can be imported and transformed in Power BI. If your data management design requires all localization to occur within Power BI, then the Slicer tables must be created in Power BI.

Figure 7 shows the two tables required to support localization of Slicer list values; one (ItemCategories) containing the translations for data items with column headers suffixed with two-letter ISO 639 Language codes (en = English, es = Spanish, nb = Norwegian), and a second table (ItemCategories_ML ) derived from the first by taking the union of all required fields from the first table with separate rows for each supported language, and the LgnID indicator in a column. This second table is in a form convenient for specifying the field required for a Slicer while also supporting language switching. The DAX expression for creating the second table is shown in Figure 8.

[1] Slicers have header labels that can also be localized using the methods of section 1.1. The methods in this section are used to localize the items listed in a Slicer.



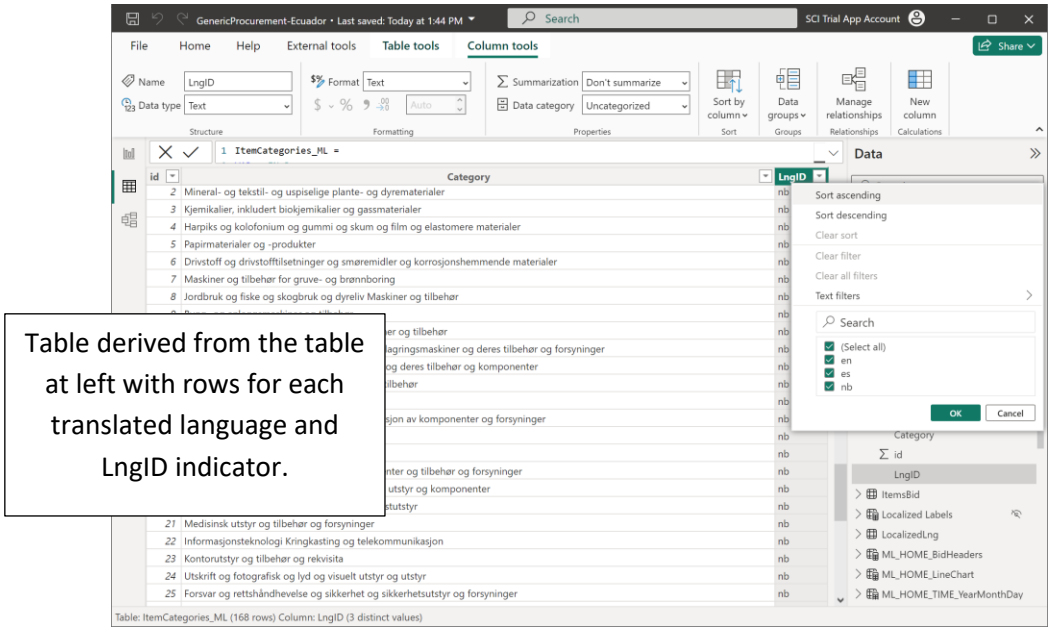




*Figure 7. Tables required to support multiple languages for Slicer item lists.*

```
        =
VAR __EN_Rows =
    SELECTCOLUMNS( 'ItemCategories'
        , "id", 'ItemCategories'[id]
        , "Category",  ItemCategories[Category]
        , "LngID", "en"
    )

VAR __ES_Rows =
    SELECTCOLUMNS( 'ItemCategories'
        , "id", 'ItemCategories'[id]
        , "Category",  ItemCategories[Category_ES]
        , "LngID", "es"
    )

VAR __NB_Rows =
    SELECTCOLUMNS( 'ItemCategories'
        , "id", 'ItemCategories'[id]
        , "Category",  ItemCategories[Category_NB]
        , "LngID", "nb"
    )

RETURN
    UNION( __EN_Rows, __ES_Rows, __NB_Rows)
```

*Figure 8. DAX Expression that creates the second (ItemCategories_ML) table shown in Figure 7.*

### 1.3.1 Time Slicers

To create multi language time slicers, a table containing time range data must be created. For example, a “TIME” table containing year, quarter and month data can be created using the following DAX expression:

CreateTimeTable( DateTime.Date( RangeStart),  Date.From(RangeEnd))

Where the parameter RangeStart and RangeEnd are parameters containing the start and end date-times for the “TIME” table. To support multiple languages, translations of each column must also be provided for each of the languages to be supported in the Power BI reports. The steps for completing a time slicer are provided below.

1. In the Power BI Table view select the "TIME" name table, then open the data table and confirm that columns for year, quarter and month data are provided for each of the languages to be supported.

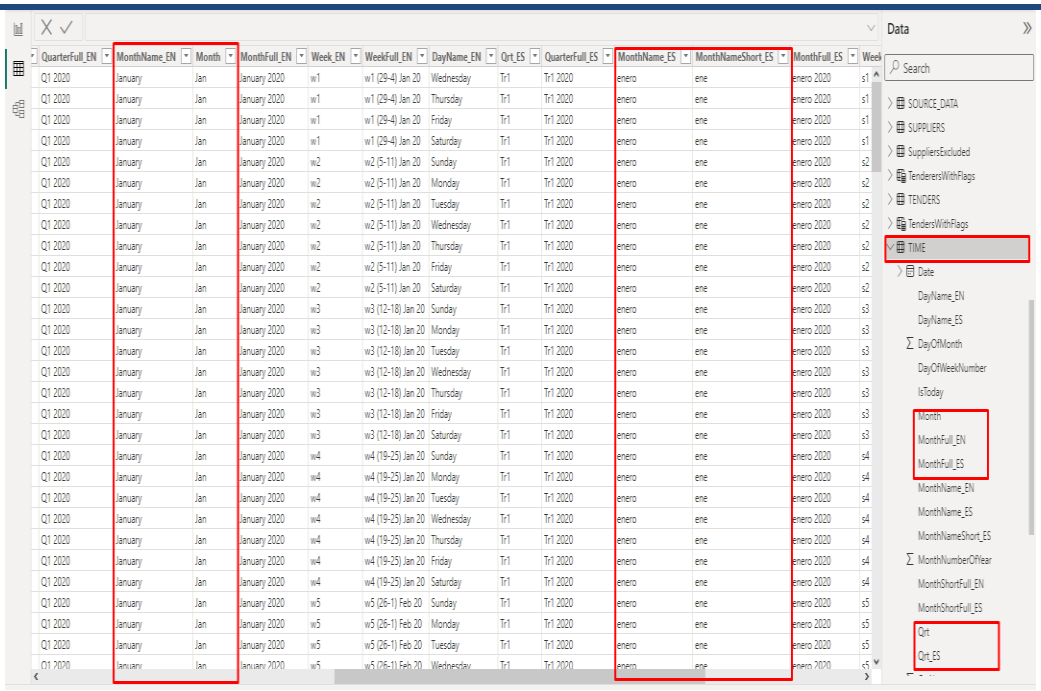


*Figure 9. : Year, quarter and month data must also be translated into three languages.*

2. To add a language, use the Advanced Editor to modify the CreateTimeTable function. Examples are provided in the example Power BI file for English, Spanish, Norwegian, and Russian (copied below for Norweigian). Choose Close & Apply to update the "TIME" table.

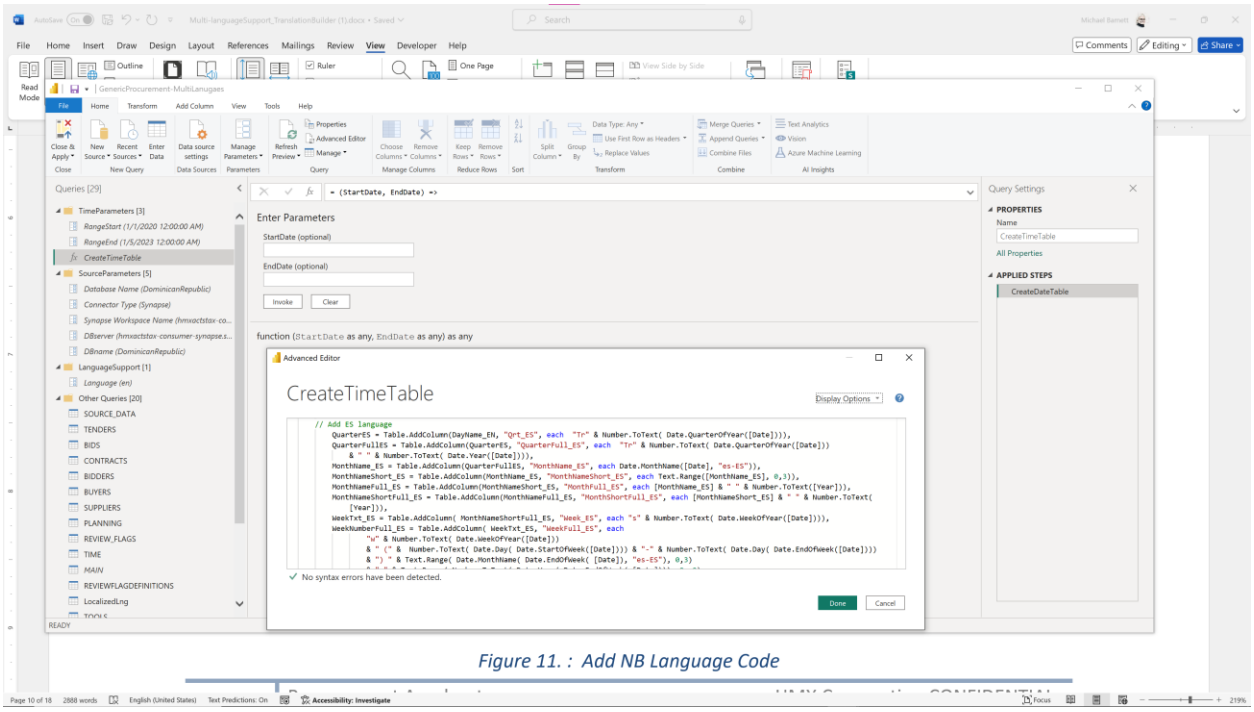


*Figure 11. : Add NB Language Code*

*Figure 10. : Modifying the CreateTimeTable function with the Advanced Editor*

```
// Add NB language
        QuarterNB = Table.AddColumn(DayName_ES, "Qrt_NB", each  "Kv" & Number.ToText(
Date.QuarterOfYear([Date]))),
        QuarterFullNB = Table.AddColumn(QuarterNB, "QuarterFull_NB", each  "Kv" &
Number.ToText( Date.QuarterOfYear([Date]))
            & " " & Number.ToText( Date.Year([Date]))),
        MonthName_NB = Table.AddColumn(QuarterFullNB, "MonthName_NB", each
Date.MonthName([Date], "nb-NO")),
        MonthNameShort_NB = Table.AddColumn(MonthName_NB, "MonthNameShort_NB", each
Text.Range([MonthName_NB], 0,3)),
        MonthNameFull_NB = Table.AddColumn(MonthNameShort_NB, "MonthFull_NB", each
[MonthName_NB] & " " & Number.ToText([Year])),
        MonthNameShortFull_NB = Table.AddColumn(MonthNameFull_NB,
"MonthShortFull_NB", each [MonthNameShort_NB] & " " & Number.ToText([Year])),
        WeekTxt_NB = Table.AddColumn( MonthNameShortFull_NB, "Week_NB", each "v" &
Number.ToText( Date.WeekOfYear([Date]))),
        WeekNumberFull_NB = Table.AddColumn( WeekTxt_NB, "WeekFull_NB", each
                "v" & Number.ToText( Date.WeekOfYear([Date]))
                & " (" &  Number.ToText( Date.Day( Date.StartOfWeek([Date]))) & "-" &
Number.ToText( Date.Day( Date.EndOfWeek([Date])))
                & ") " & Text.Range( Date.MonthName( Date.EndOfWeek( [Date]), "nb-
N0"), 0,3)
                & " " & Text.Range( Number.ToText( Date.Year( Date.EndOfWeek(
[Date]))), 2, 2)
                ), //  Text.Range([MonthName], 0,3)),
        DayName_NB = Table.AddColumn(WeekNumberFull_NB, "DayName_NB", each
Date.DayOfWeekName([Date], "nb-NO"))
    in
        DayName_NB
in
    CreateDateTable
```

3. Slicers require tabular data organized by-row with a language key as one of the columns. This table can be created from the time table created in step 2 above. The following DAX code snippet demonstrates the creation of the required multilingual dataset in Power BI. This dataset, named TIME_ML in the provided example Power BI file, combines date-related attributes from the “TIME” table into English  Spanish and Norwegian translations needed to support slicers for changing displayed lanuages.

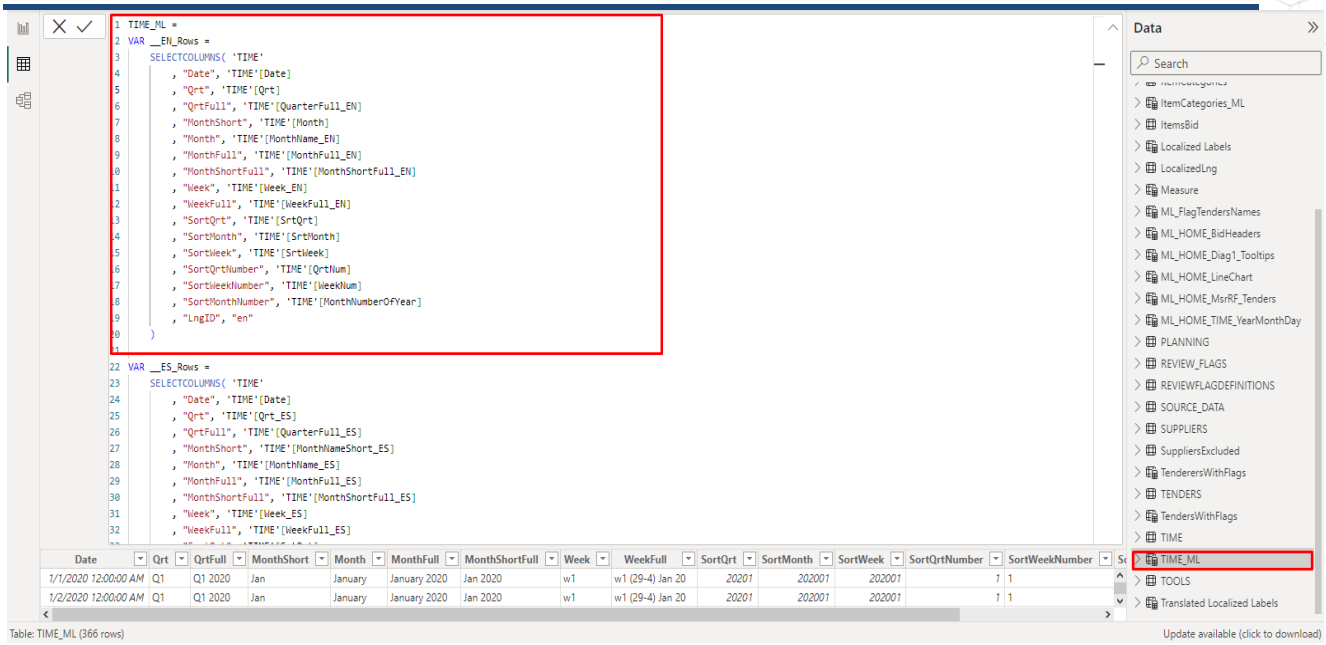



*Figure 11 : Table TIME_ML combines date-related attributes from the TIME table into English Spanish and Norwegian translations.*

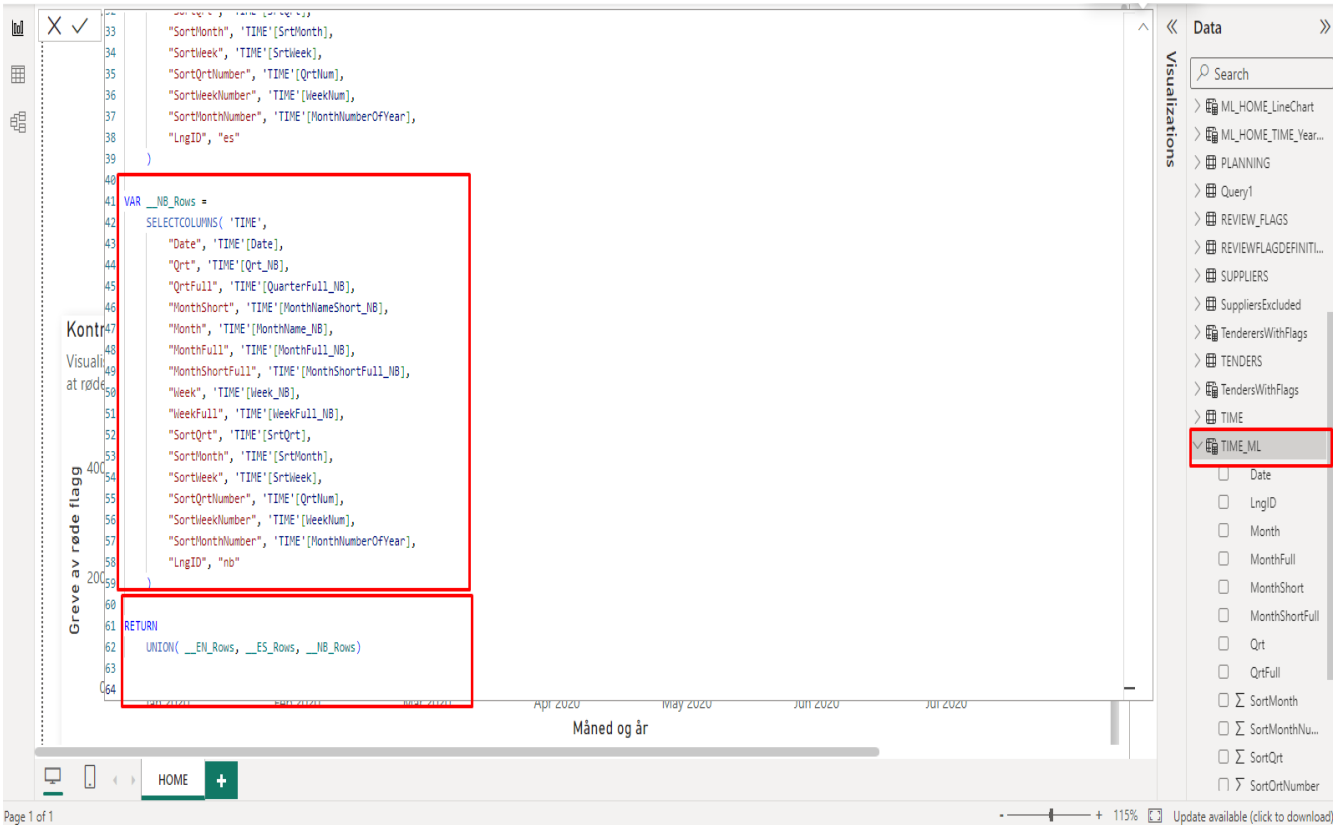



*Figure 12: Table TIME_ML, combines date-related attributes from the TIME table into English Spanish and Norwegian translations.*

- *VAR __EN_Rows*: Defines a variable to select date-related columns from the "TIME" table and renames them with English translations where applicable.
- *VAR __ES_Rows*: Defines a variable similar to __EN_Rows, selecting columns from the "TIME" table but with Spanish translations.
- *VAR __NB_Rows*: Define a variable similar to __EN_Rows, selecting columns from the "TIME" table but with Norwegian translation
- *RETURN UNION( __EN_Rows, __ES_Rows, __NB_Rows)*: Returns the union of the tables created by __EN_Rows , __ES_Rows and __NB_Rows creating the TIME_ML table with English, Spanish and Norwegian translations for date-related fields.

4. The field 'LngID' in the 'LocalizedLng' table will be selected as the 'Field' configuration of the time slicer, so a relationship between this field and the LngID field of TIME_ML must be created.

   In the Power BI Data pane, select the TIME_ML table, right-click and select 'Manage Relationships'. If there is no relation between these two fields, click on 'New' to create the relation between LngID in LngID in TIME_ML. This relation should be One-to-Many(1,*), and it should be made active.

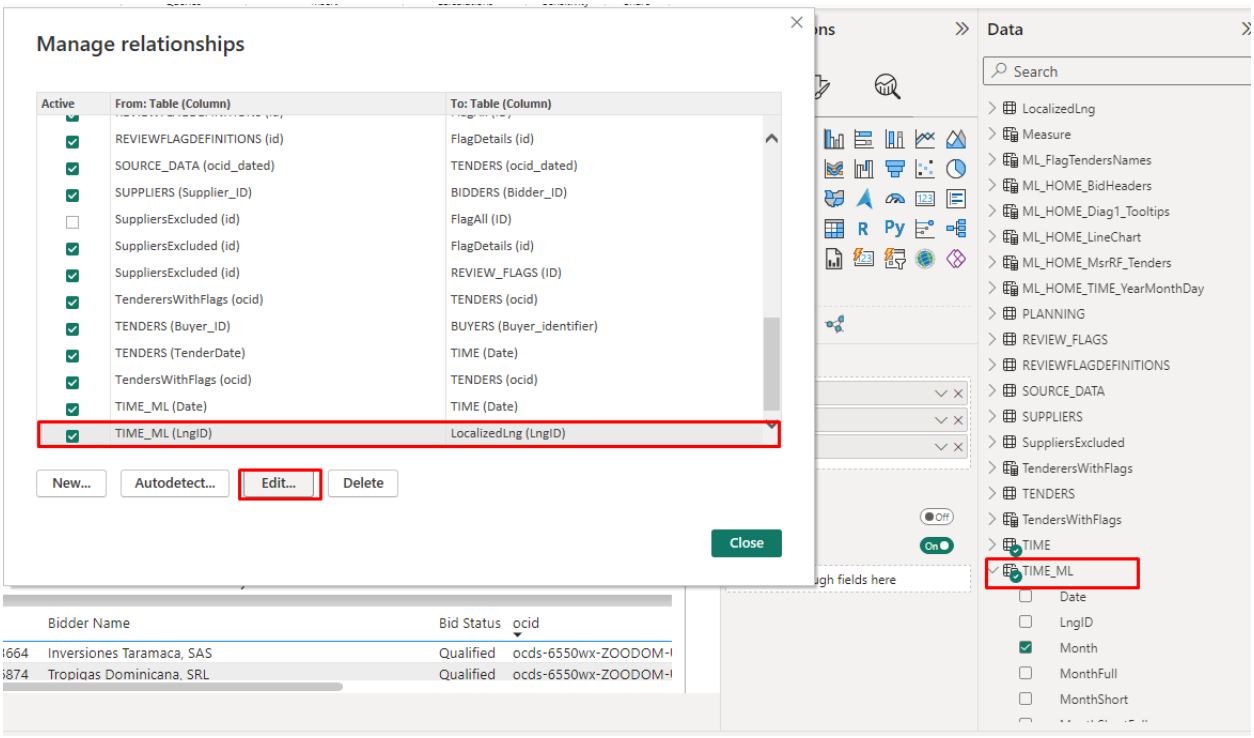



*Figure 13: Manage Relations between Time_ML(LanID) And LocalizedLng(LngID)*

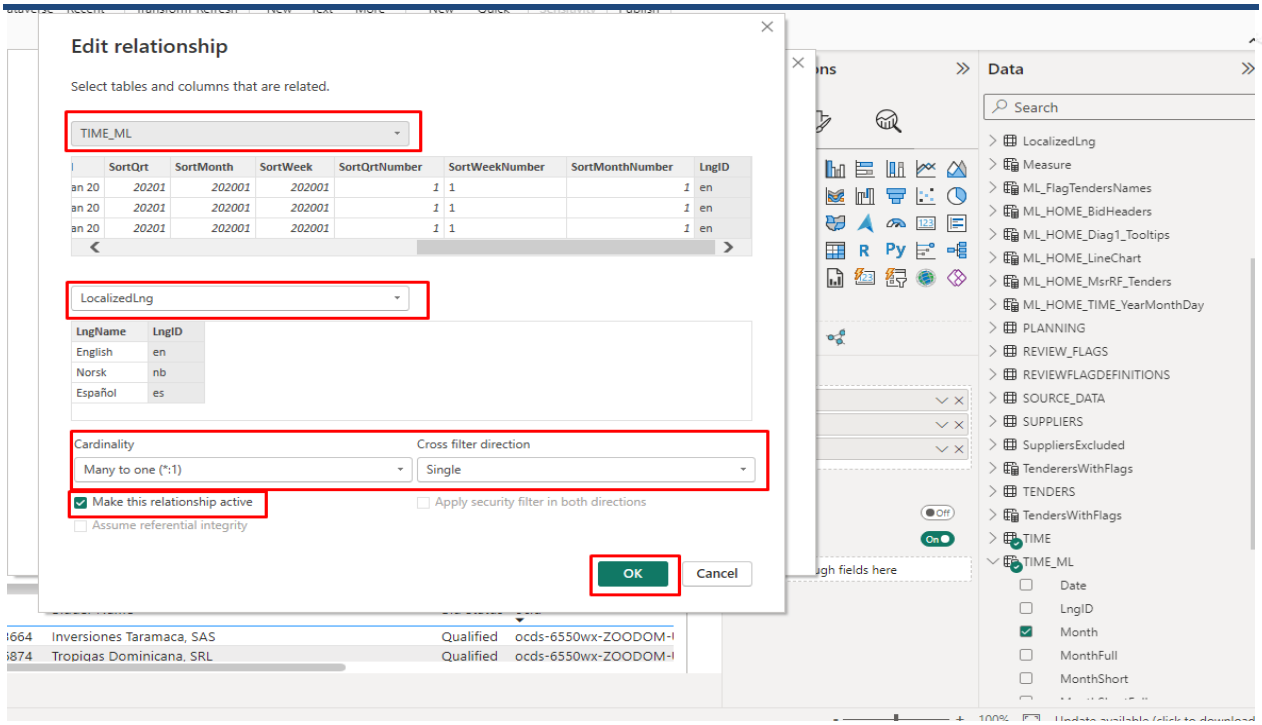



*Figure 14: Edit Relations between Time_ML(LanID) And LocalizedLng(LngID)*

5. When a time slicer drop-down list (e.g., 'Select date range' in the provided example Power BI file) is created, multiple values may be available due to the definition of multiple languages. To resolve the ambiguity, a switch measure is defined for use in a time selector Top N filter and that will choose the language based on a language selected in the language slicer. An example implementation is provided in the accompanying Power BI file.

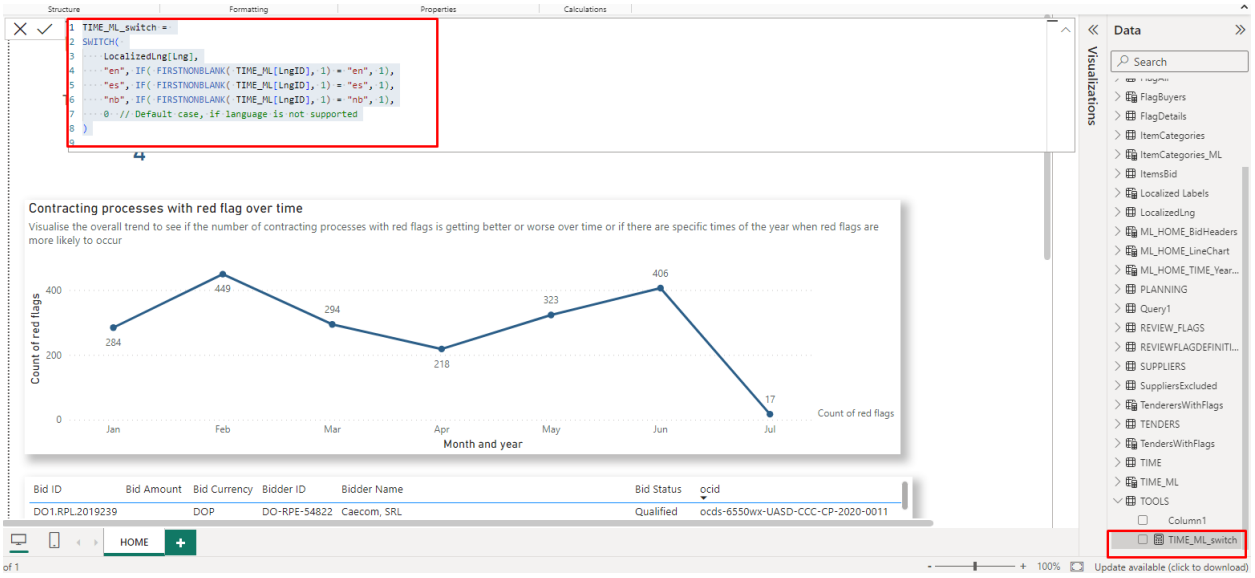



*Figure 15: Update TIME_ML_switch code and update Norwegian language.*

To define the required switch logic, in the Power Bi 'Data' pane right-click on the TOOLS table, select 'New Measure' and add the following code for the TIME_ML_switch measure.

```
TIME_ML_switch =
SWITCH(
    LocalizedLng[Lng],
    "en", IF( FIRSTNONBLANK( TIME_ML[LngID], 1) = "en", 1),
    "es", IF( FIRSTNONBLANK( TIME_ML[LngID], 1) = "es", 1),
    "nb", IF( FIRSTNONBLANK( TIME_ML[LngID], 1) = "nb", 1),
    0  // Default case, if language is not supported)
```

6. To define the time slicer filter, select the time Slicer ('Select date range' in the example Power BI file), open the Filters and configure a Top N filter as shown in Figure 16.

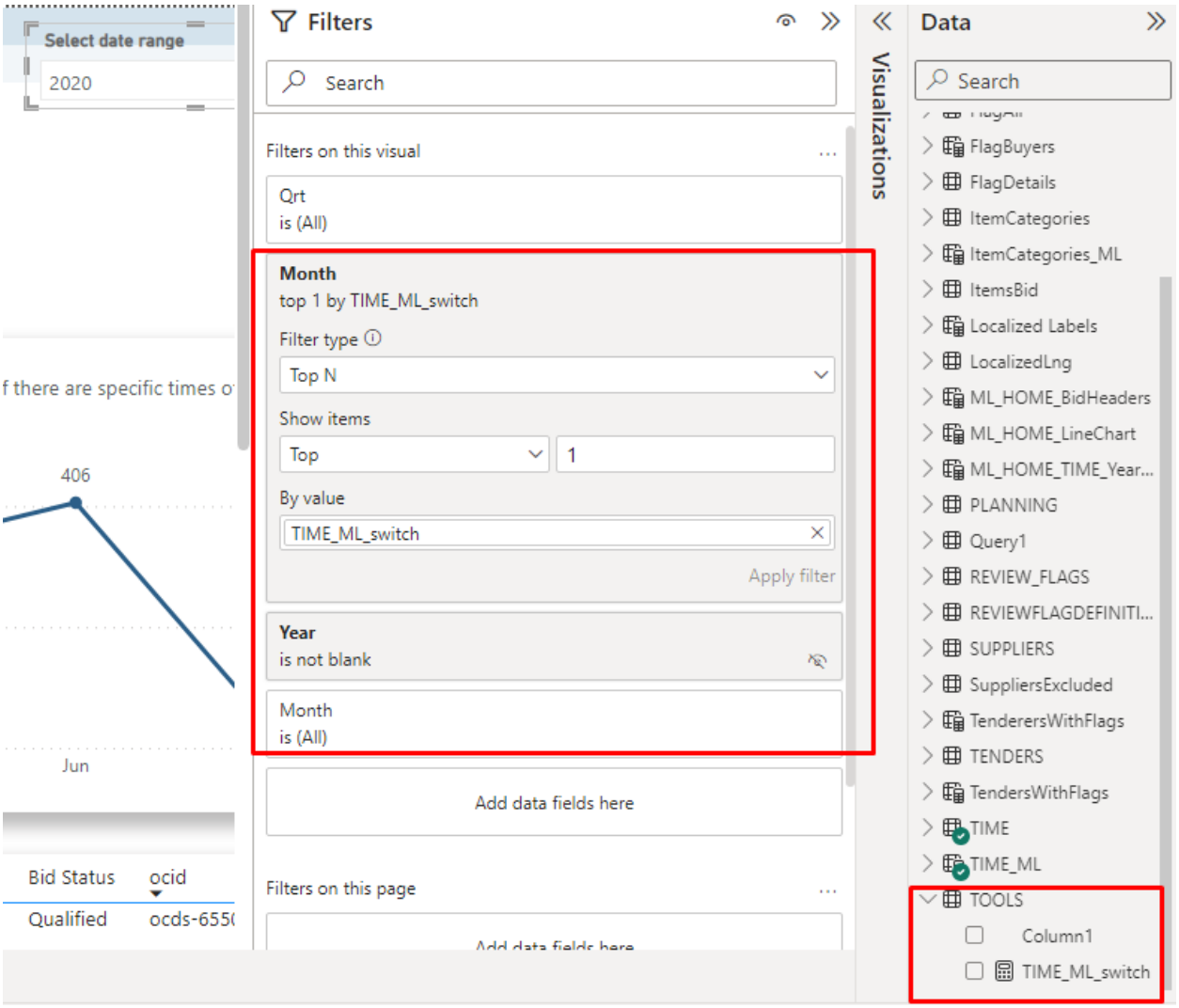



*Figure 16: Filters -> Month -> TIME_ML_switch code*

7. Finally drag and drop "Year", which requires no translation, from the “TIME” table, then "Quarter" and "Month" from the “TIME_ML” table. are selected. Check if the data in the charts and tables update according to the selected time and language.

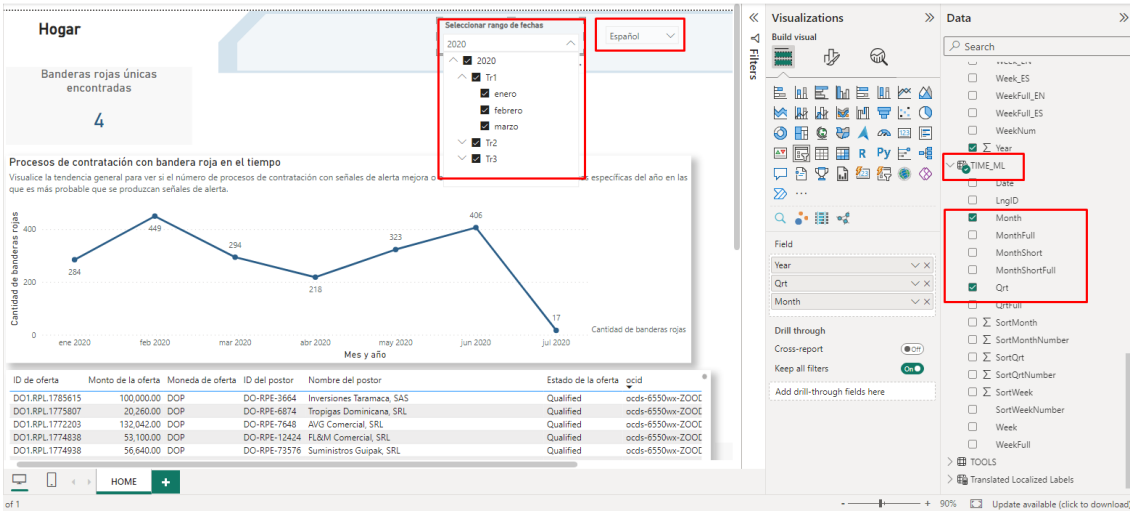



*Figure 17. Navigate to the report view and open the table named "TIME_ML".*

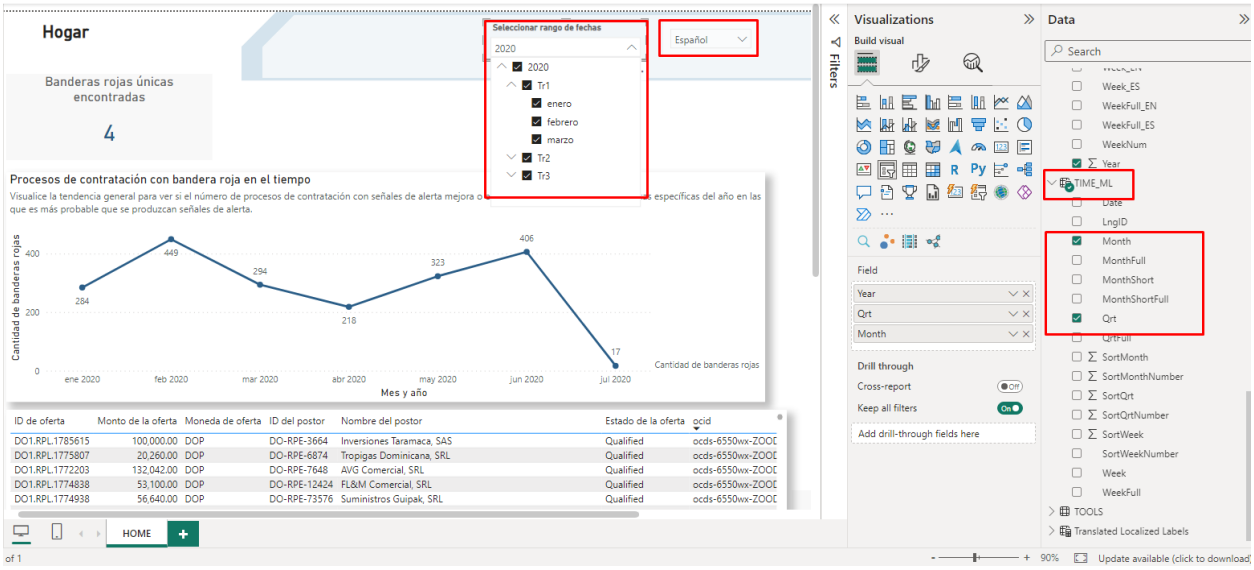
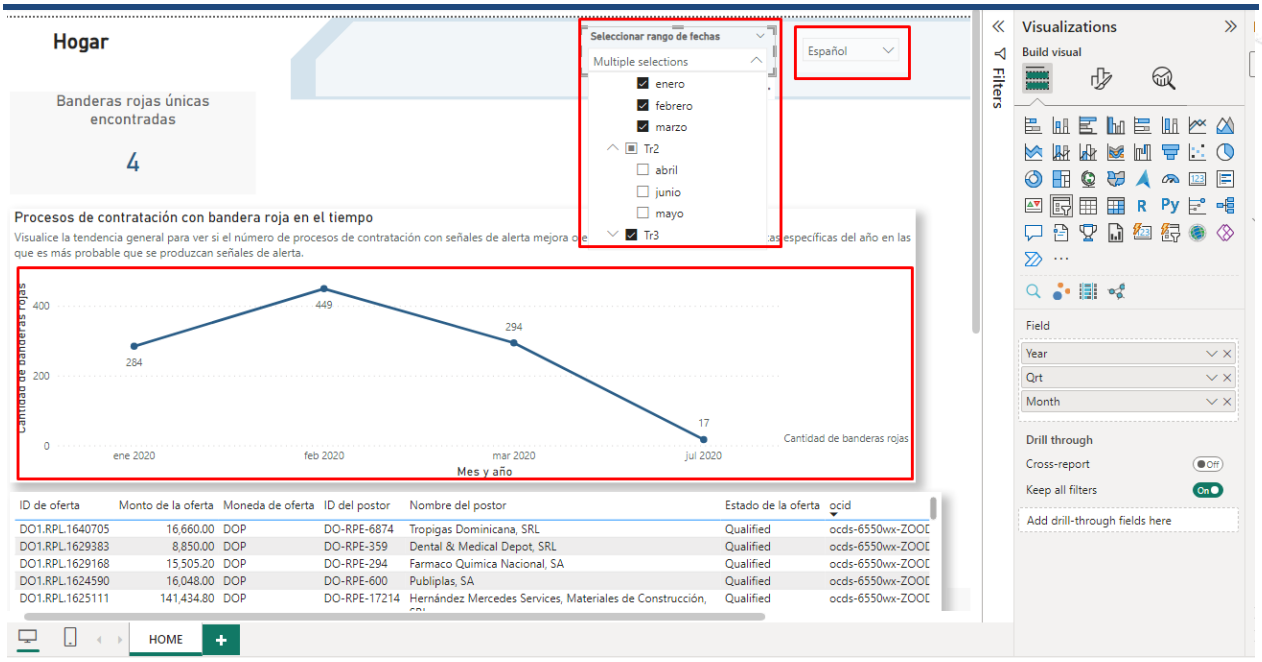



*Figure 18. reports will display based on selected Data Range parameters.*



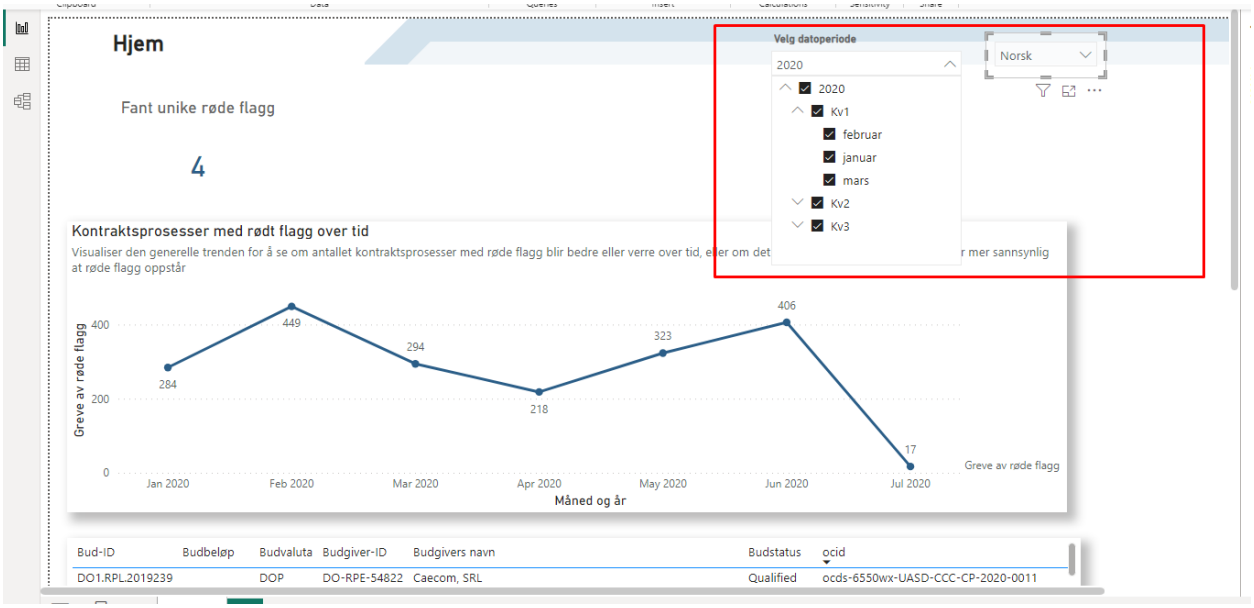




*Figure 19. In the charts in the reports the month label changes according to the slicer language.*