**Data Analytics with Power Bi**

1. **What is Business Intelligence (BI)?**

It is the ability to transform data into information, and information into knowledge, to streamline decision making.

* 1. **BI Flow**
* **ETL (Extract - Transform - Load):** Refers to the extraction, transformation and loading of data. It is a process required to convert data into information. Some ETL tools are *Power Center, Integration Services* and *ODI*. They allow establishing a process flow that helps to homologate and clean data, and then load it into a data warehouse; which leads us to data modeling.
* **Modeling:** Through relationships and the creation of metrics and indicators, the data model is established to answer business questions. Here we have tools such as *Erwin Data modeler or Powerdesigner.*
* **Reporting:** We move on to data visualization, reports, dashboards and storytelling. In this step we find tools such as *Power BI, Tableau, MicroStrategy,* etc.
  1. **Power BI**

Power BI is the data visualization platform focused on data reporting for bussines intelligence. Belonging to Microsoft, this unified and scalable platform for BI has the following business suite:

**Power BI Business Suite:**

* **Power BI Desktop:** Exclusive tool for Windows that allows you to connect to various data sources and create reports. It is where the process flow of a BI solution with Power BI begins.
* **Power BI Service:** Cloud service that allows you to establish a collaborative environment (information delivery).
* **Power BI Mobile:** Allows viewing and interacting with dashboards from mobile devices.

**Power BI components:**

* **Power Query:** For the ETL process.
* **Power Pivot:** For modeling in order to answer business questions.

1. **Power BI Architecture**
   1. **- Power BI Free:**

As its name suggests, Power BI Free is the free layer of Power BI, and it includes:

* It includes 1 GB of storage.
* It does not allow simultaneous collaboration (reports, dashboards and datasets).
* Sharing reports is only possible in public mode. Therefore, it is recommended to use the free plan only for reports or personal projects.

**2.2 - Power BI PRO:**

* Power BI PRO offers the services of Power BI Free, plus collaborative tools.
* It includes 10 GB of storage.
* It can be shared with internal users as long as they also have a PRO license. For example, two licenses would be needed for a developer to share a project with another person.
* Allows secure and reliable sharing of information.
* It has a Power BI Gateway option. Supports up to 8 application reloads per day.
  1. **- Power BI Premium**
* Power BI Premium offers the services of Power BI PRO with some enhancements.
* Includes 100 TB of storage.
* Can be shared with internal users without Power BI Pro.
* Greater scalability and performance than the shared capacity in Power BI Service.
* Features Power BI Gateway. Supports up to 48 reloads per day.

**3. Connection types**

* **Import:** Data is copied from local within Power BI (the most common type).
* **Direct Query:** Data is not copied, each interaction makes a query to the database. Recommended for frequently changing data
* **Live Connection or dynamic:** Reading from SSAS or from a Power BI Service dataset.
* **Composite Models:** combines import and Direct Query technologies. Allows the use of multiple datasets.
  1. **- Obtaining data**

We open Power BI Desktop, and in the home tab we go to "Get data". We will see some common sources for getting data. We can click on "More..." to see all the data sources that Power BI offers us.

To test the import type connection, we can simply open any Excel file. To test Direct Query, we select SQL Server in the home tab (or by clicking on get data). Then, we will have the option to select import or DirectQuery. Then, we connect to the database with the following access data.

**4. ETL processes**

ETL is an intermediate process between the original data sources from which we extract information, and the data modeling for further analysis. ETL consists of three steps:

* 1. **Extract**

Extract data from any source, whether flat files, binaries, databases or cloud services. These can be internal operational sources (i.e., that collect transactional information in our company), or external sources. This will depend on the analysis that you want to carry out.

* 1. **Transform**

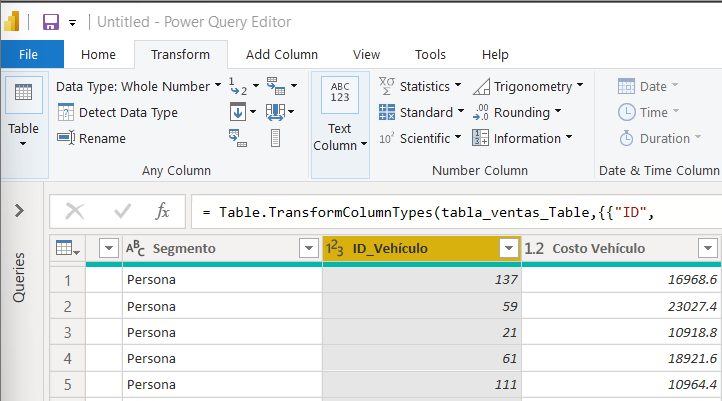
Transform, clean or enrich the extracted information without modifying the source. It is in this step that the data is adjusted according to the data model (which is designed prior to the creation of the ETL).

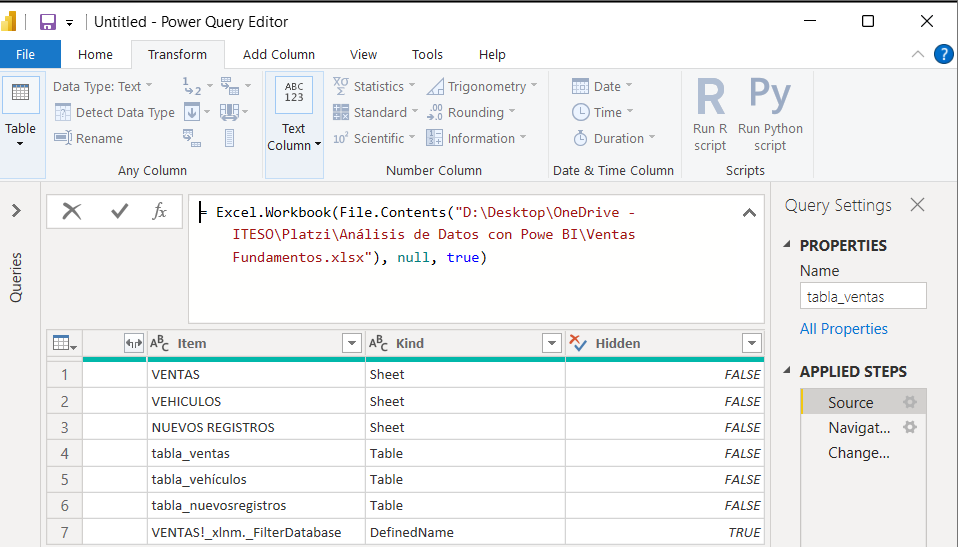
* 1. **Load**

Load the transformed data into the data model.

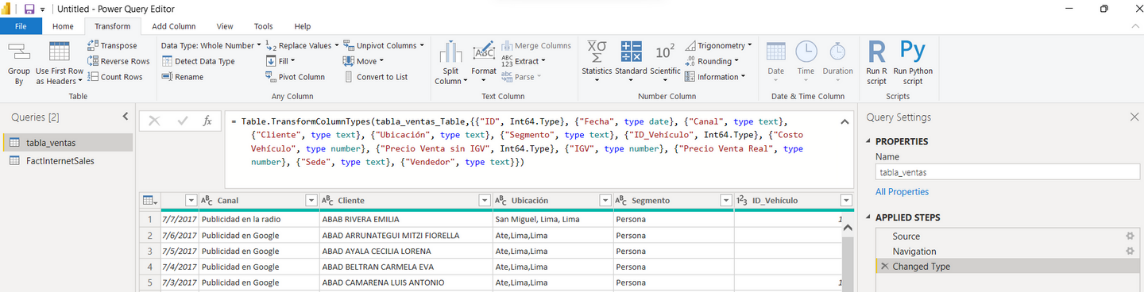
***Power query***  is a data connection technology that allows you to detect, connect, combine and refine different data sources to meet our needs. In essence, it is in charge of the ETL process, it extracts, transforms and loads data into Power BI for further analysis. Note that the goal of Power Query is not to analyze data but to manipulate it.

Power Query has a functionality called *Magic* , which is a collection of steps to reach a result, which allows you to go back or forward without modifying the data source. It is similar to Excel macros. To use Power Query select the table we want to process and click on “Transform data”. Once in Power Query, we can observe the data type of each column:





However, depending on how the data is entered in the original SQL server table or Excel file, Power Query makes an automatic estimation of the type of data present in the cells. It is necessary to check well the data entries to make sure that this automatic modification does not corrupt our analysis in the future.



**4.2 – Transformations**

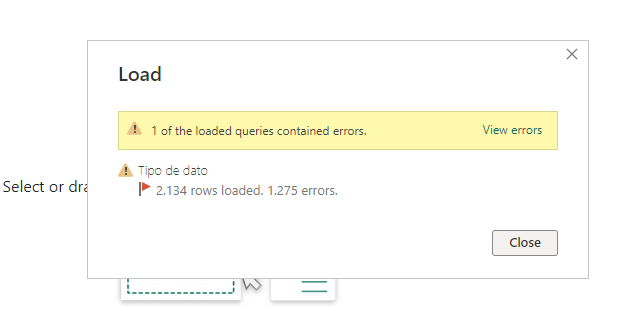
Transformations within Power Bi refer to the necessary modifications applied to the raw data to ensure its format, integrity and correct value for analysis and presentation within our visualizations. The most common transformations we can perform in Power Query are:

* Change data type
* Append queries
* Split columns
* Combine queries
* Replace values

We can also make combinations:

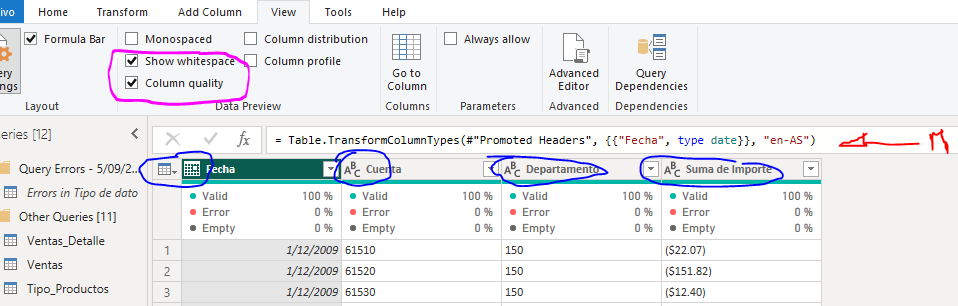
* Combine binaries
* Add columns
* Filter data

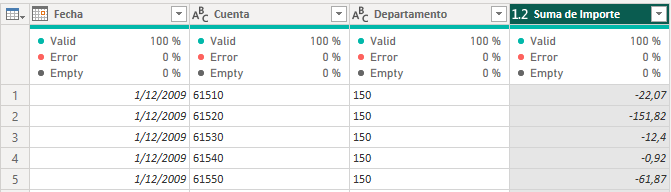
Similar to relational databases, in Power BI data types need to be well defined before using them. For example, when we import a file and this has conflicts within their cells, Power Bi outputs something like:



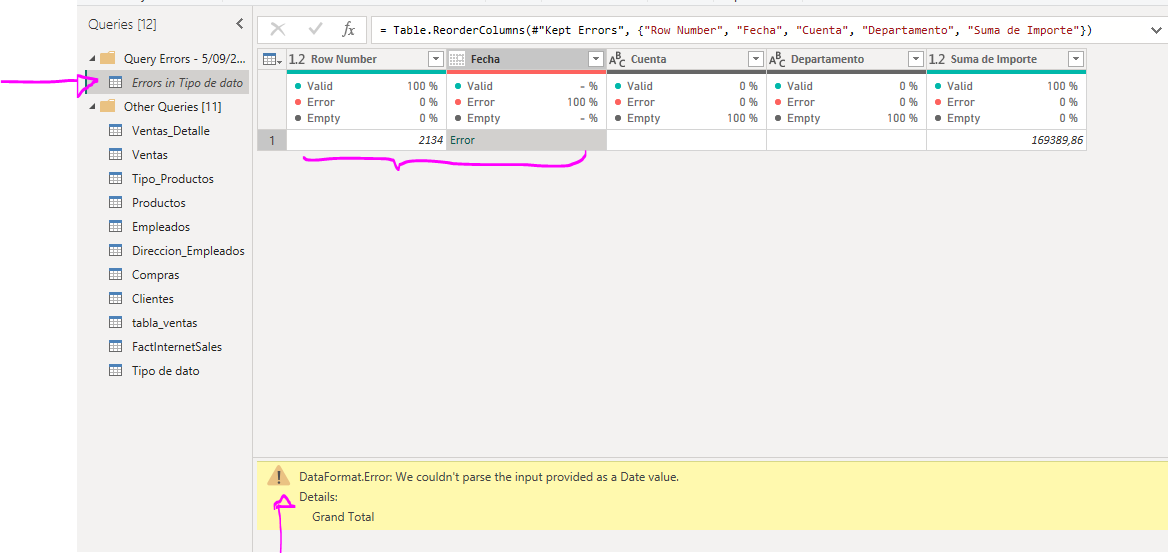
When there are errors in the data, it is necessary to remove the "changed" step inside the original query to not validate the automatic transformation that PowerBI does by default.

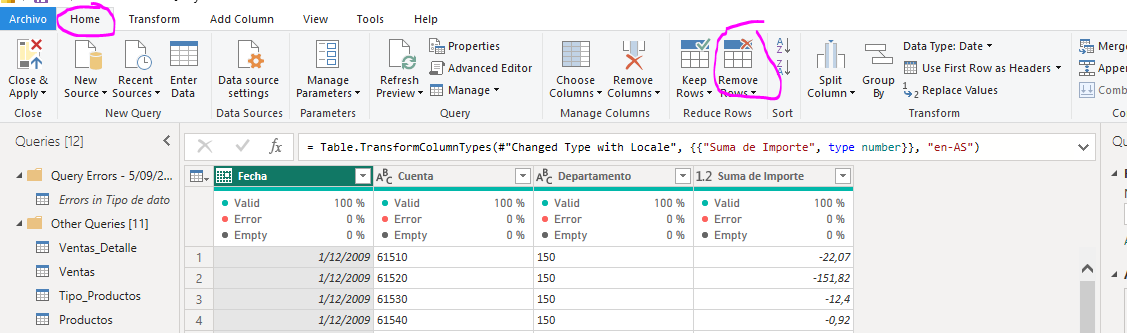
After this, we must first check that the data type is correct within the current table, to do so, we open Power Query and change the data type within each column according to our need and having in mind the type of local format that we are seeing and the local format that we expect to implement in our ETL process.

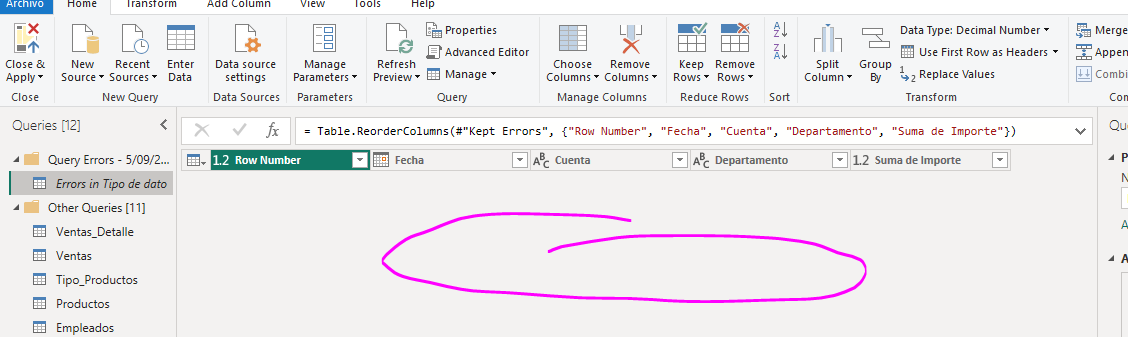




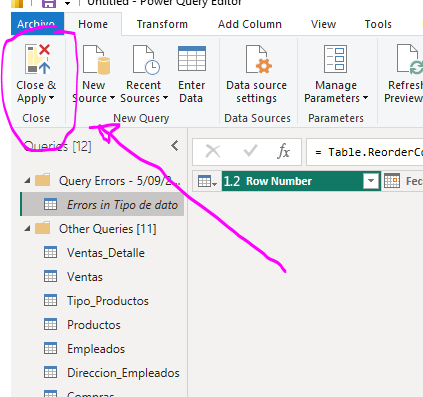
Whenever a data error is solved, the data table that Power BI creates by default in parallel to the original raw data, reduces its entries until no further information is reported there. For example if we have corrected all the errors except one, placed at the end of the table in question we can see the placement and information about the data inconsistency and then drop the unnecessary entry as follows.







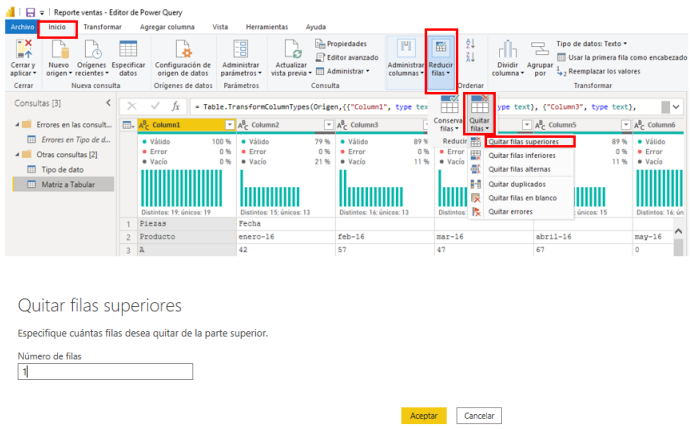
Finally, in order to apply the applied transformations, we simply select “Close and apply” in Power Query and then the changes will be applied to the data stored in Power BI.



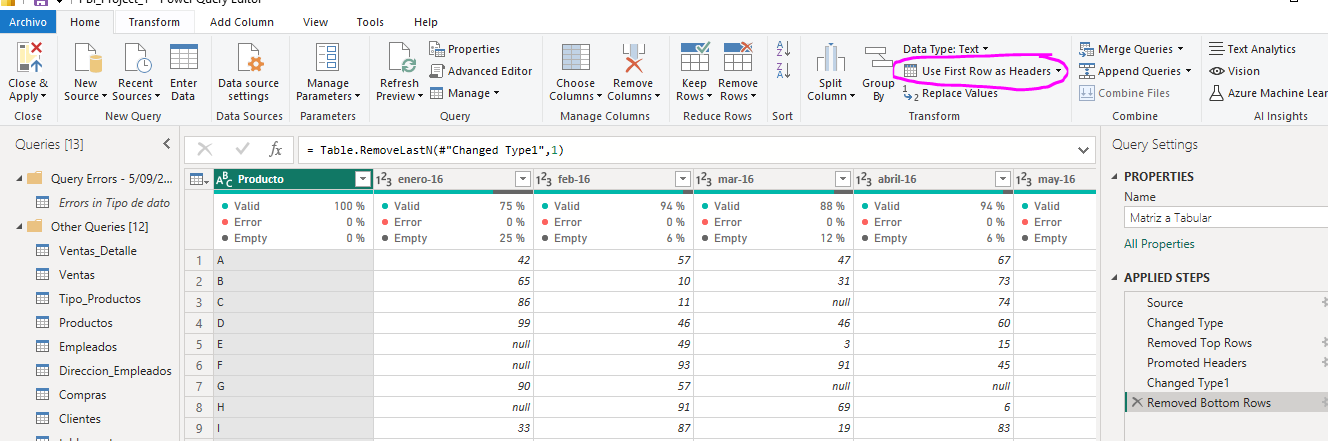
**4.2.1 – Converting a file in matrix format to tabular format**

Normalmente, las tablas que vienen en formato matriz presentan dos problemas estructurales iniciales, el primero, hace referencia a que existen encabezados artificialmente puestos que no brindan mayor información. El segundo, que los siguientes campos los cuales representan los encabezados de nuestra data no han sido destinados a precisamente ser los encabezados de la tabla y hacen que la información se presente en diferentes columnas como si cada una de ellas fuera una tabla distinta.

Primero, quitamos los encabezados innecesarios y dejamos en el top, aquellos que en verdad realizarán esta función.



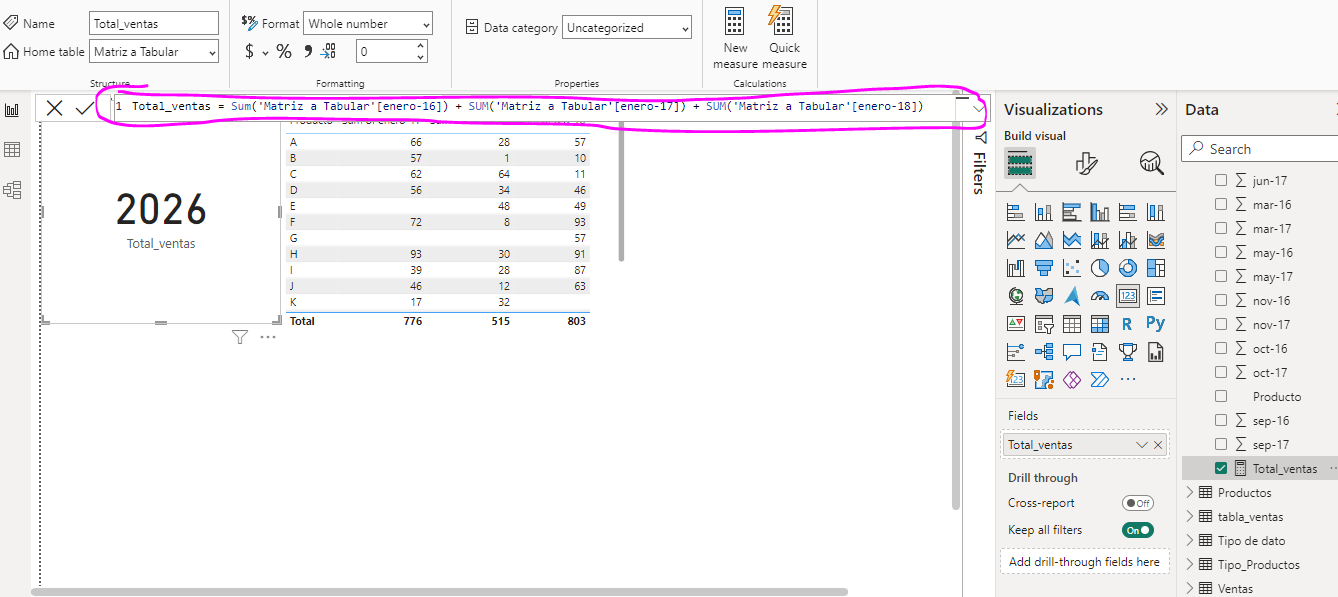
Luego, usamos la primera fila como el ncabezado de nuestra tabla. En la pestaña de inicio usamos el botón “Usar primera fila como encabezado”. En “Pasos Aplicados” se generará el paso de “Encabezados promovidos”.



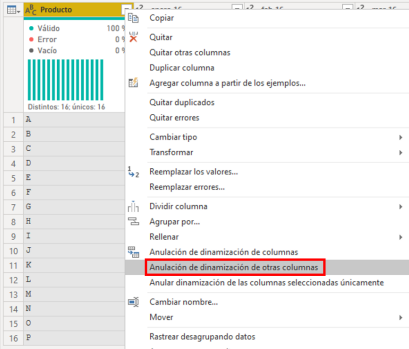
Finalmente, removemos las filas en el ultimo espacio de la tabla que generalmente puedan estar relacionadas con un “Total” haciendo click en “Reducir filas”, luego haciendo click sobre “Quitar Filas”. Seleccionamos “Quitar filas inferiores” y luego en la ventana emergente seleccionamos el número de filas a eliminar.

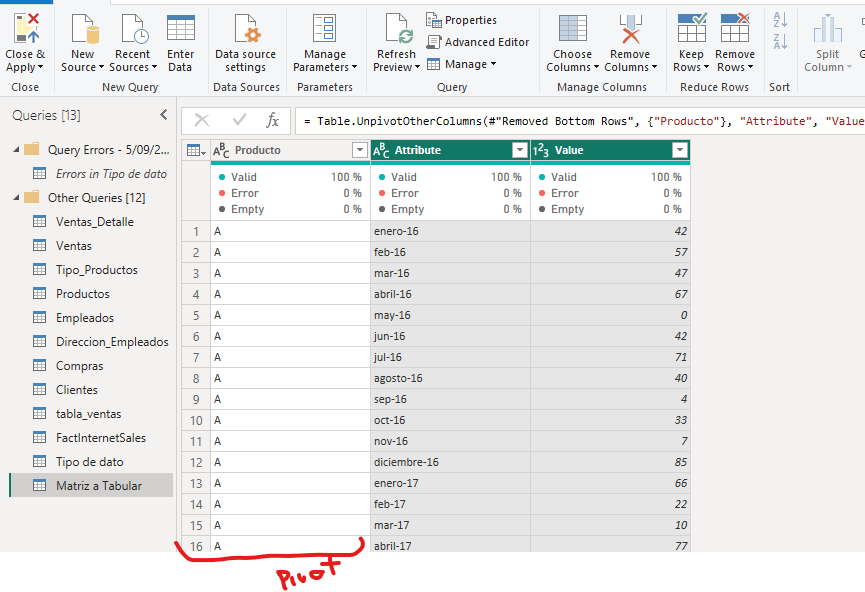
Así, la tabulación básica de una matriz consiste en agrupar de manera correcta la información de manera que el encabezado de la tabla corresponda al tipo de información que queremos por columna. No deseamos que el encabezado de cada columna sea algo de la forma “columna1 , columna 2, ….,”.

Sin embargo una tabla que no está tabulada al ser leida como un conjunto de columnas-tabla y no una tabla única, presenta desafíos mayores cuando deseamos modelar la data en una visualización, ya que al estar toda la información separada por columnas, impide que se automaticen los procesos de visualización dentro de nuestro informe haciendo que no podamos dinámicamente cambiar el valor del indicador de interés, peor aún, éstos indicadores han de ser formulados a partir de implementación de fórmulas en DAX a modo Excel, lo cual no es optimo.

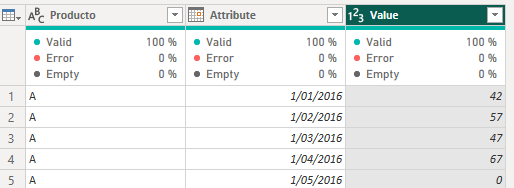


Ya que la información de la matriz se encuentra separada en “producto” y “fecha 1, fecha 2,….,” necesitamos hacer una tabla de la forma *atributo-valor*  o conocidas como tablas pivote donde el atributo sea la fecha, el valor las ventas y la característica pivote, la primera columna, es decir, la columna “producto”.





Luego, nos aseguramos que los formatos que tienen las columnas generadas estén de acuerdo a nuestras necesidades, así, cambiamos los tipos de datos al formato necesario.



Con este simple proceso, podemos reportar un indicador que si sea escalable y pueda ser controlado mediante un objeto dinámico como un slicer.



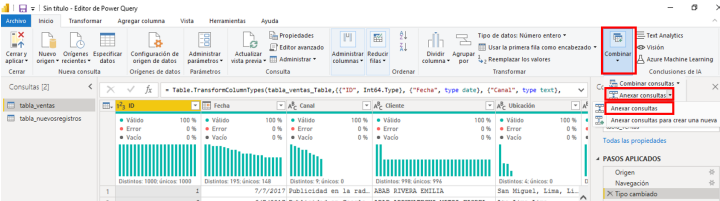
**4.3 – Combinations**

There are different types of query combinations within Power BI which are essentially the types of combinations available in SQL databases.

**4.3.1 - Append:**

* Allows you to join two or more tables "vertically" (i.e. rows are added).
* It is recommended that both tables have the same structure. If this is not the case, the system adds to the final set the fields of all other tables with null values.
* Similar to a **SQL UNION** operation.
* The results can be a new query or be added to an existing query step.

To append tables, we must make sure that the tables to be appended have the same structures, otherwise, null values will be added by default where applicable. To make an append we simply select "Merge", then "append query" and in the pop-up window we select the information table we want to add to our original table.

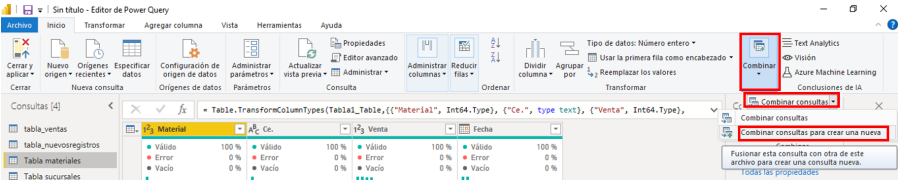


**4.3.2- Combine queries:**

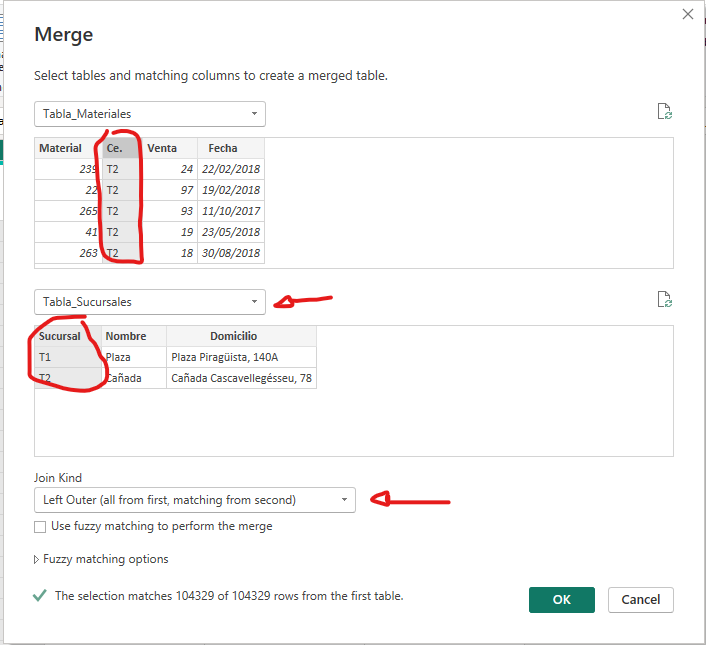
* Allows us to take two tables and cross them by a common column.
* Usually used to supplement information from a table.
* It is the closest equivalent to the **JOIN** function of the SQL standard.
* The different types of combinations (and their equivalent in SQL) are:

1. External left (LEFT JOIN)
2. External right (RIGHT JOIN)
3. Full outer (FULL OUTER JOIN)
4. Internal (INNER JOIN)
5. Anti left (LEFT EXCLUSIVE JOIN)
6. Anti right (RIGHT EXCLUSIVE JOIN)

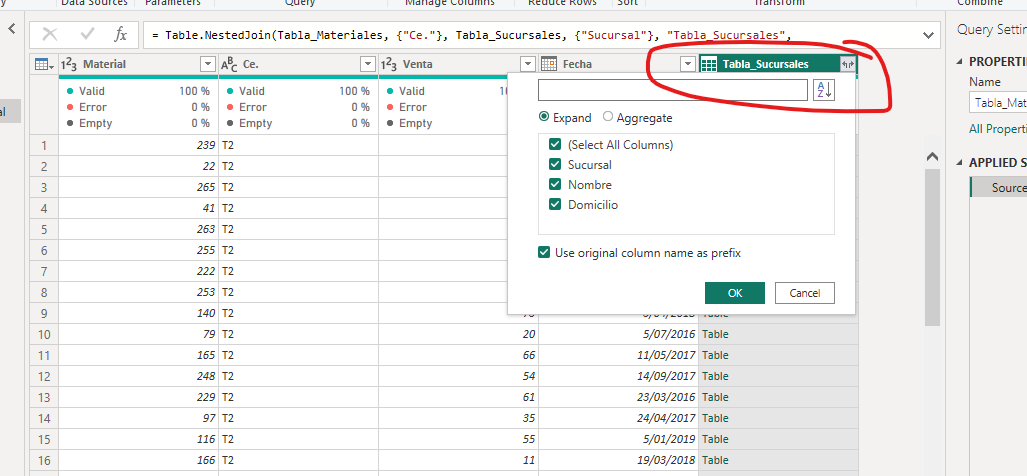
To combine queries, go to "merge queries" and click on the option "merge queries as new".

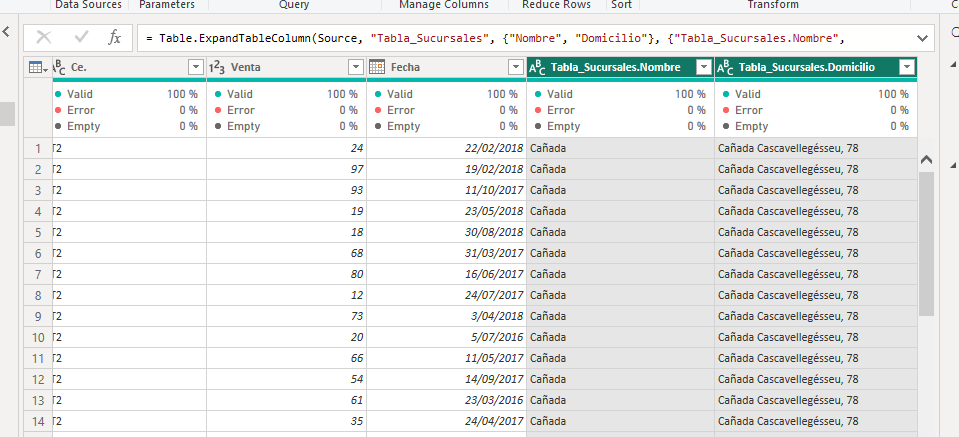


Then, within the dropdown, we select the external table with which we want to feed our original query. Then we select the relationship field through which Power BI will examine the possible matches. Finally we select the type of JOIN we want to execute.



Once the new query is created, a new column referring to the set of matches found will be added. Within the header dropdown, we select those changes from the external table that we want to preserve within our new query.



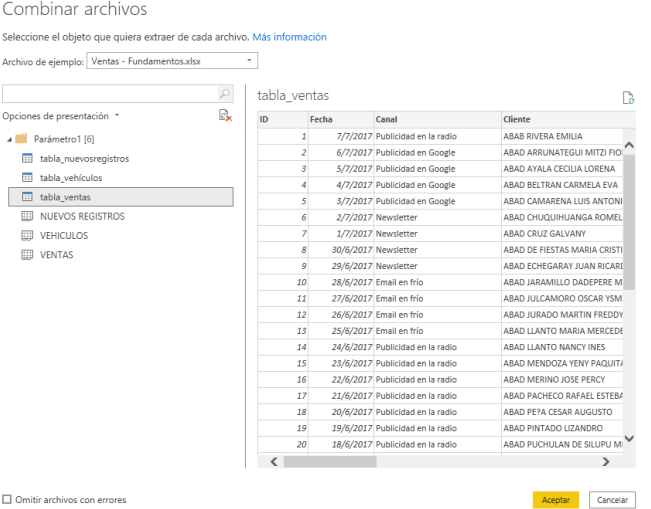


**4.3.3- Combine binary:**

* Allows tables to be extracted from files using an automated process.
* Usually used through the folder connector.
* It is especially useful when the information source is too fragmented for the append operation.

The binary comination actually corresponds to an automatic linking of different tables located in a common source by means of a table-parameter common to all files. This means that Power BI will append in a single table all those excel sheets that are common to the books in question.

To do this, first click on "Get Data", select "More..." and then click on the "Folder" option. Power Bi will automatically detect all the files present in the folder in question and click on the "Merge and Load" option.



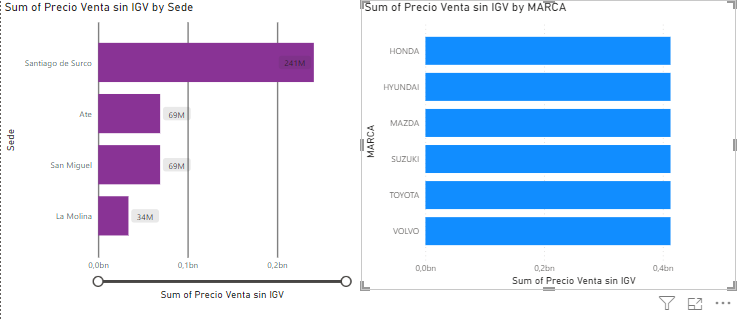
**5. Data Modeling**

The data modeling within Power BI is executed with the Power Pivot tool, which in essence, is a communication platform of tables within the same database which establishes through relationships or common fields between tables, interchange of queries. Thus, when performing a manipulation or visualization of a table, the visualization can be fed by the underlying information in the tables connected to the main table that we want to visualize.

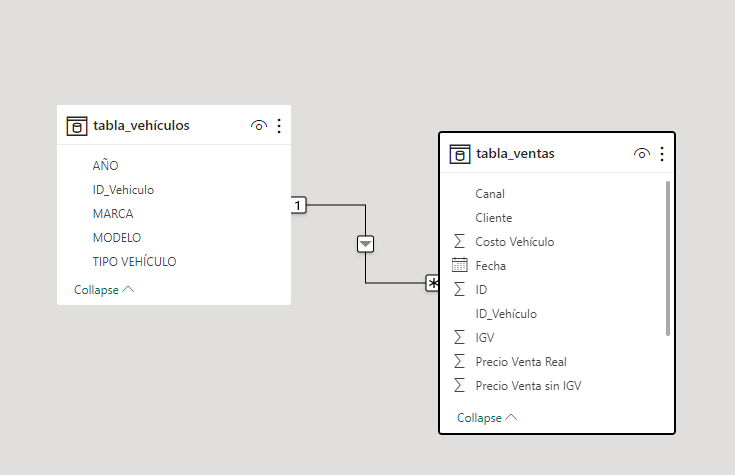
In essence, Power Pivot is the tool that allows you to generate the SQL communication model between the different types of tables available.

For example, let's imagine that we have a table which has the sale price information and the information of the location that made a certain sale. If we make a bar chart containing the total sales by location (left panel) there will be no problem in getting the visualization as we want.

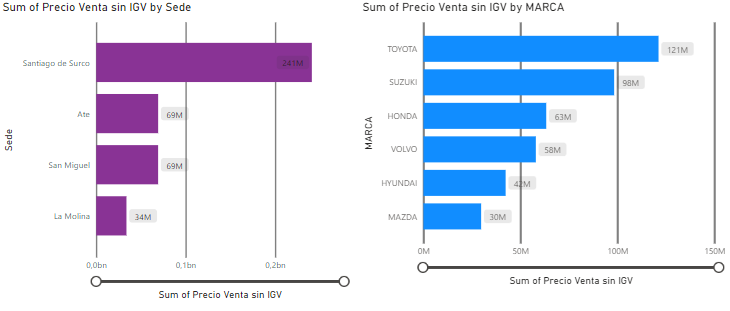
Now if we want to observe the total sales price by vehicle brand, being the vehicle brand information stored in another table, if we do not make the communication through a common field between tables, Powe BI will not know how to assign these totals and by default will assign a single value corresponding to the total value of sales made repeated for each brand.



If we run Power Pivot, we can see that the sales table and the make and vehicle information table will have a common id, "id\_vehicle". By establishing the relationship between these two fields by simply dragging this item to the table we want to relate, the visualization of the total sales by brand, visualization fed by two tables, will use the vehicle id to extract the information regarding what type of brand each vehicle id belongs to, thus, you can use the information from the external table as a sub query information repository for the main sales table. This way Power BI will know what data to use to generate the appropriate visualization.



The importance of database modeling lies in the fact that it allows to have a scalable form that facilitates the visualization of information distributed across multiple tables. It corresponds to the communication map between queries without the need to use JOIN statements.



**5.1 – Relationships and filters**

In databases we encounter the concepts of relationships and filters. Relationships refer to the correspondence between tables. Filters, on the other hand, refer to including (or not) certain records when querying a database, based on certain criteria.

**Table Relationships:**

When we talk about relationships between tables, we have two key concepts:

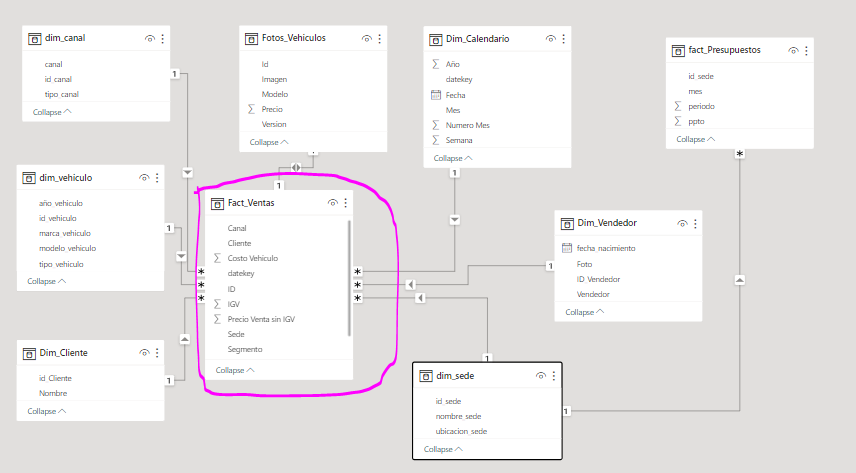
* **Primary Keys:** they define the main key of the table. They cannot contain null values or duplicates.
* **Foreign Keys:** is a column, or set of columns, that contains a value that refers to a row of another table.

**Types of Relationships**

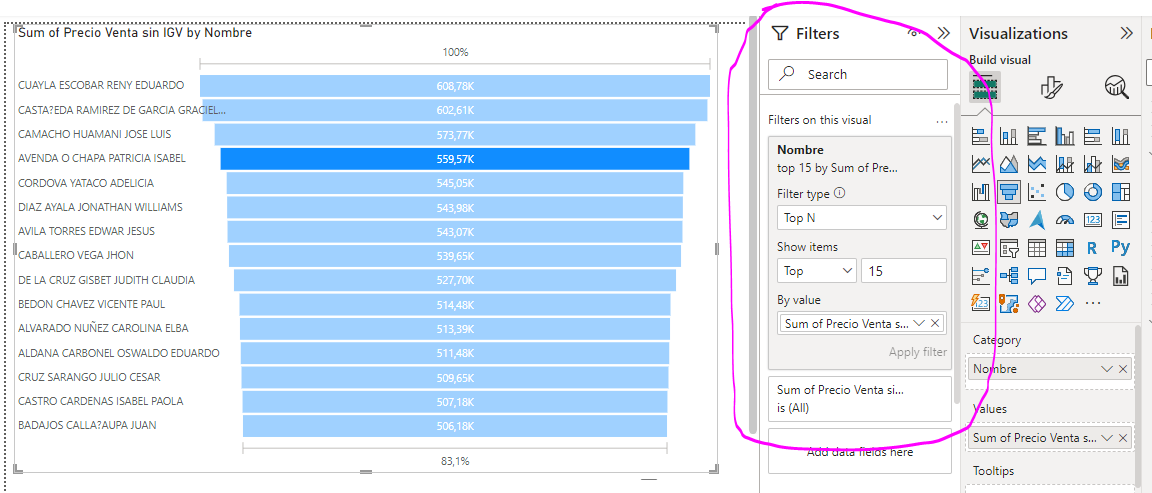
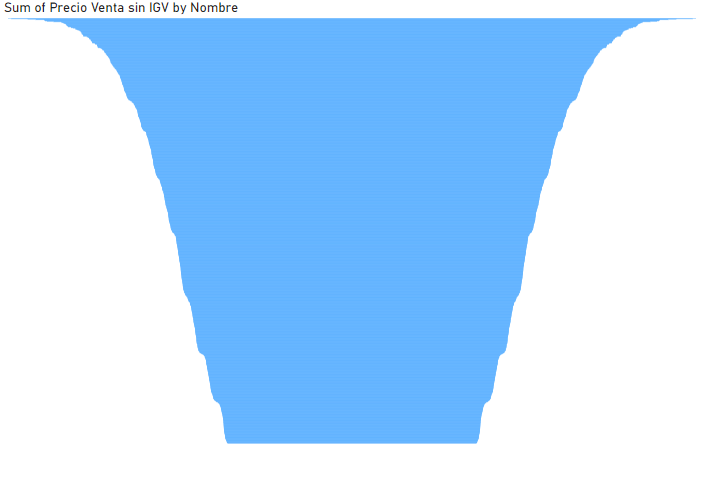
* 1 to 1 (1-1): both tables are connected with their primary keys.
* 1 to many (1-\*): when a primary key is connected with a foreign key from another table. This is the one to be searched in Power BI.
* Many to many (\* - \*): occurs when both tables are related by their foreign keys (none of the columns have unique values). It is recommended to avoid this type of relationship.

For example, when loading a database that has a main table where all the economic activity of a period is recorded and whose complementary tables contain information concerning external keys, i.e. information sometimes not related to the period of commercial activity but information concerning the details of the resources, types of customers, brands, categories, etc, involved in the commercial activity, Power BI automatically generates the possible relations-matches by field between tables, although we can modify it by hand.

Note that an optimal data model is one that handles one-to-many or one-to-one relationships to avoid inconsistencies in the data as in the case of many-to-many, and generally employs one or a few main tables where the activity period and other relevant information is recorded, connected to different sub-tables that will serve for the main tables to extract relevant information about the different categories that this main table manipulates, handles or talks about.



On the other hand, filters are information extraction criteria applied to visualizations with the aim of presenting only the most relevant data for storytelling, allowing not only to show the most relevant insights but also to generate cleaner visualizations. For example, below is a funnel visualization for the number of sales made to each customer, where the best 15 customers have been filtered out in the right image.



**5.2– Data modeling reviewed**

There are several types of data models: star, snowflake, etc. For Power BI, the most efficient model is the star model, because it results in tables with one-to-many relationships. The star model is composed of:

* Dimension (lookup) table: has descriptions of the fact table. Dimensions add context to the facts. For example: dates, location, etc.
* Fact table (transactional or fact): has the bulk of the information. For example: sales, subscriptions, orders, etc.

**Data model engine**

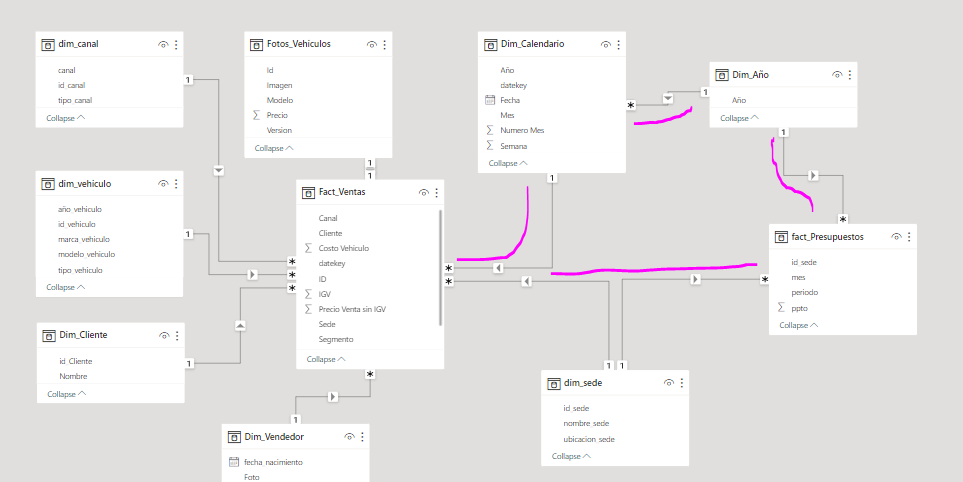
* Its name is Vertipaq
* It is in charge of all data analysis operations (DAX).
* Uses in-memory technology.
* Allows short development cycles.

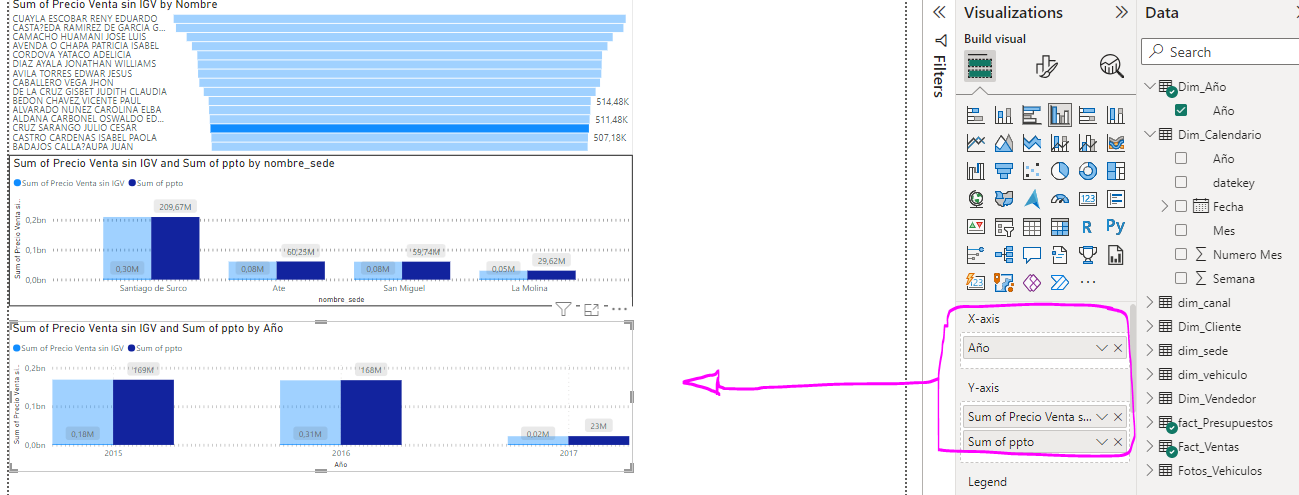
An example of the star model is shown below, where we have two star models whose centers are the sales fact table and the budget fact table. The idea of the model is to make a study of the organization by comparing the sales statistics with the budget statistics. To do this, we generate star models for each fact, the nodes of which correspond to the different dimensionality tables in our database.

It is common practice to generate pivot dimensions which correspond to tables of unique elements extracted from another table using Power Query.

For example, if we have that sales\_fact and budget\_fact are connected by calendar\_dim, making a bar chart of these measures using as a comparison dimensionality the calendar dimension can be counterproductive because Power BI although it creates the reference dimensions correctly as categories, it does not assign well the values to compare between sales and budget because within calendar\_dim there is an ambiguous date format.

To be precise, for example if we want to make this comparison correctly through the years of activity, we must generate a pivot table containing the years of activity, then relate this table to dim\_calendar and fact\_budget to establish a proper relationship between sales and budget through the date, which will make its sub-query in the years table.





In essence, in order to correct errors not as obvious as the previous example in the data modelling, it is necessary to generate pivot tables according to the dimension of comparison every time we want to communicate two tables, especially if they communicate with each other through another calendar table. For example, note that the sales table does not have a date but has the sales\_id, an external key present in the calendar table. Thus, the date of sales and budget is not explicit because within the calendar table there are several fields that contain the information of the years, confusing Power BI in the assignment of display values.

**5.3– DAX Language**

*Data Analysis Expression, or DAX*, allows us to create analytical formulas. It was created to manipulate a tabular data model. Originally, it was generated as an extension to Excel. It is a collection of functions and operators that can be used in expressions to calculate one or more values. It is also found in PBI, Excel and SSAS Tabular.

**Advantages of DAX**

DAX has several advantages. It is designed for BI users, and has a lower learning curve for data analysts. It takes advantage of the knowledge of working with Excel formulas, adding more capabilities such as:

* Navigation relationships.
* Dynamic calculation of dimensions.
* Handling of time dimensions (time intelligence).

**DAX Format**

Table name'[Column name].

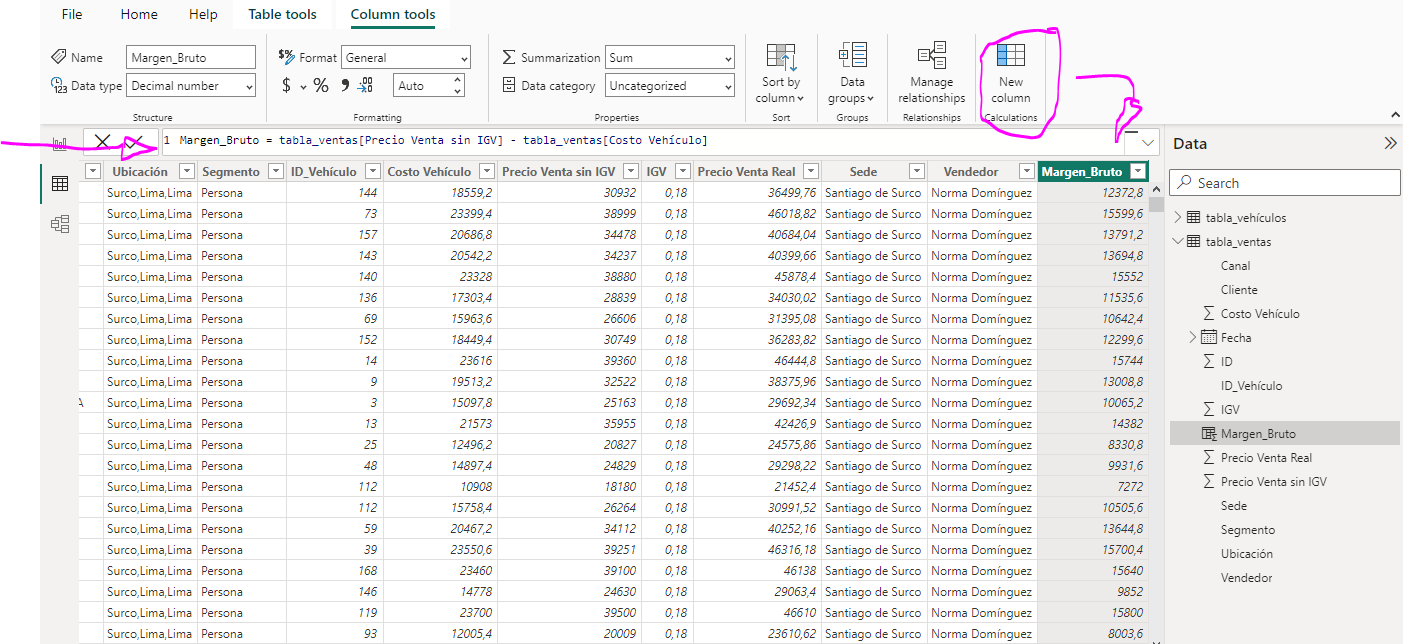
Example: 'Table Products'[Price].

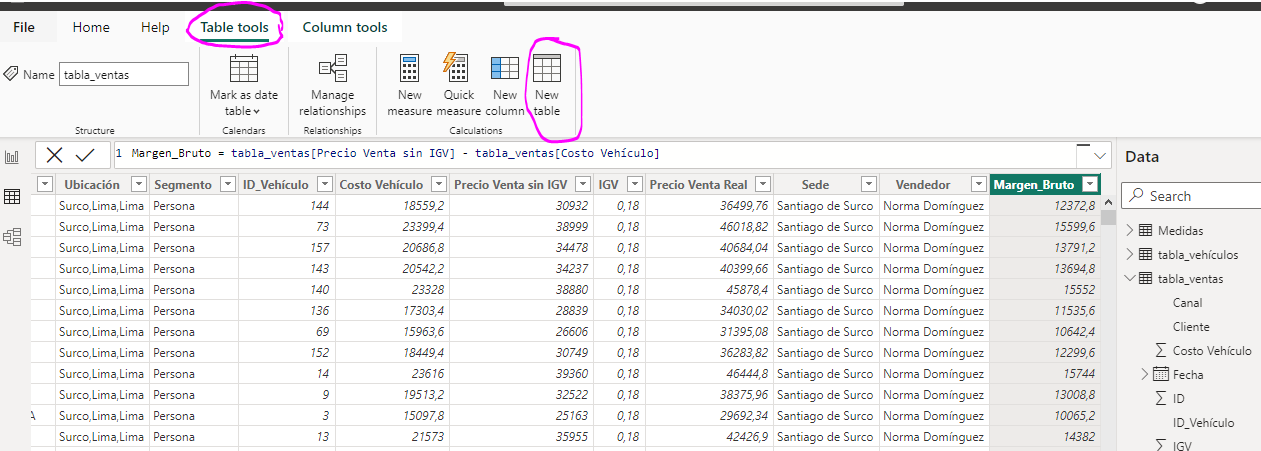
The table name can be omitted when used in calculated columns, but it is not recommended for ambiguity reasons.

**What can we generate with DAX?**

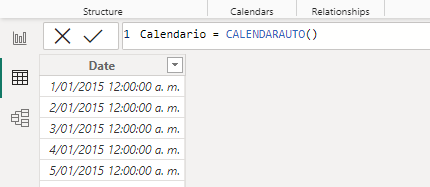
* Calculated columns: creates new columns in the data model. It is a method to connect tables with multiple key columns.
* Calculated Tables: creates a table derived from another table.
* Measures: creates dynamic calculations stored in memory. More efficient than calculated columns. Support time intelligence.

For example, to create a calculated column within a table, select the table of interest and click on "Data". Then, click on "Add Column" and Power BI will automatically give us the option to implement a Dax statement with which it will calculate the fields.

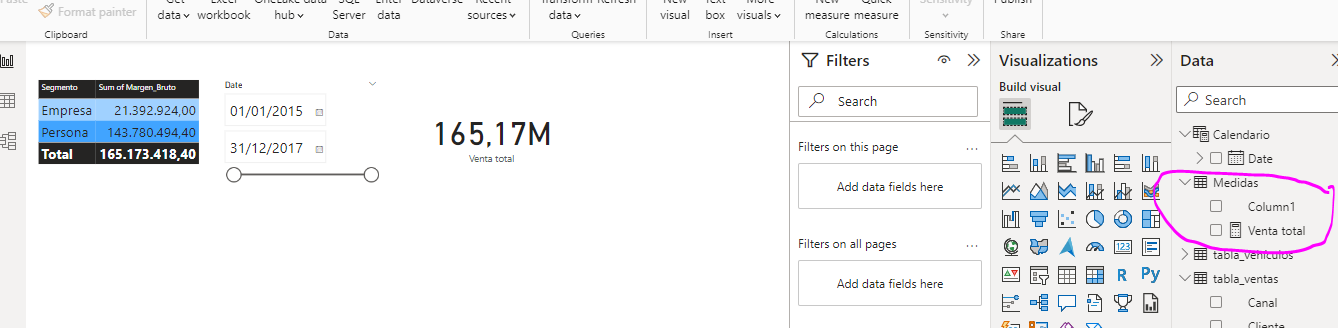




To create a new table, click on "Table View" -> "Table Tools" -> "Create new table". Then, using the DAX language, we specify which type of table population we want to create.



Finally, to create a measure, it is advisable to generate a new empty table using "Specify data" where we will store all the metrics of our presentation. Then, right click on this metrics table and click on "New metric", then a DAX sentences space will open where we must specify the metric to be calculated.



Measurements in DAX are characterised by the use of aggregations. CALCULATE is an aggregation that allows us to "modify the filter context", as well as to create a row context within our calculations, iterating row by row.

The CALCULATE function receives at least two parameters. The first is the expression we want to filter The next parameters are the filters we want to apply

For example, if we want to get the total sales for a specific location, we can use CALCULATE and the Total Sales measure (created in the previous class) to filter the total sales for that location.

That is, we would create a new measure with the following DAX

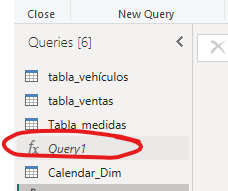
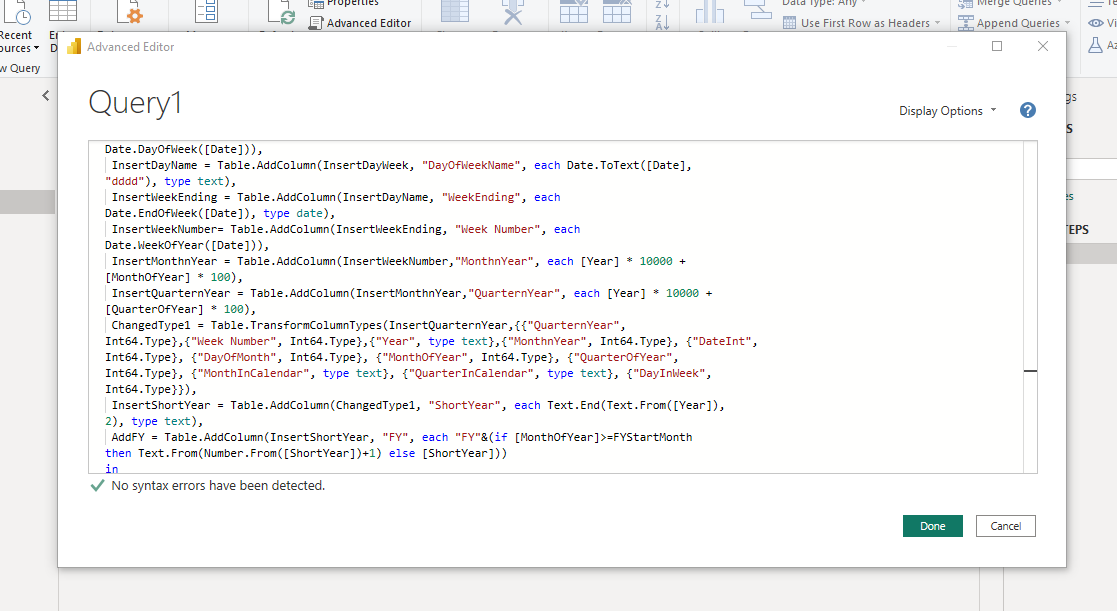


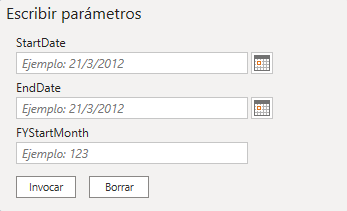
**5.4– Time Intelligence**

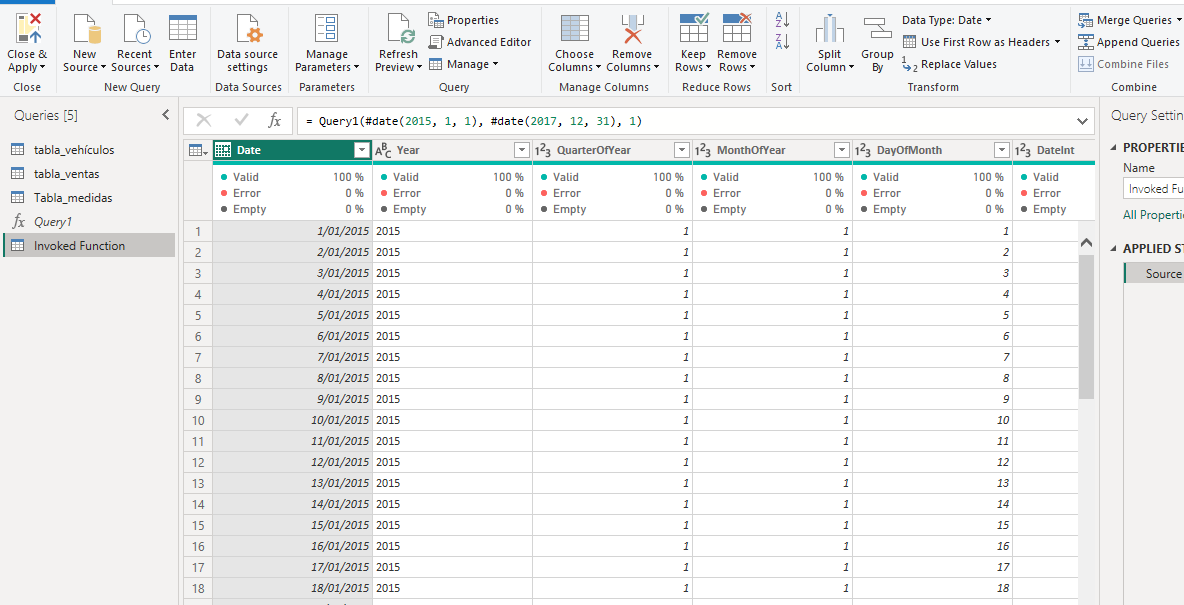
Time intelligence refers to the possible analyses, visualisations and conclusions that can be drawn from a metric of interest considered over a time series. Within the DAX language, the most common functions related to functions such as CALCULATE are the following, and correspond to queries, calculated metrics or filtered data extracted from a time series.



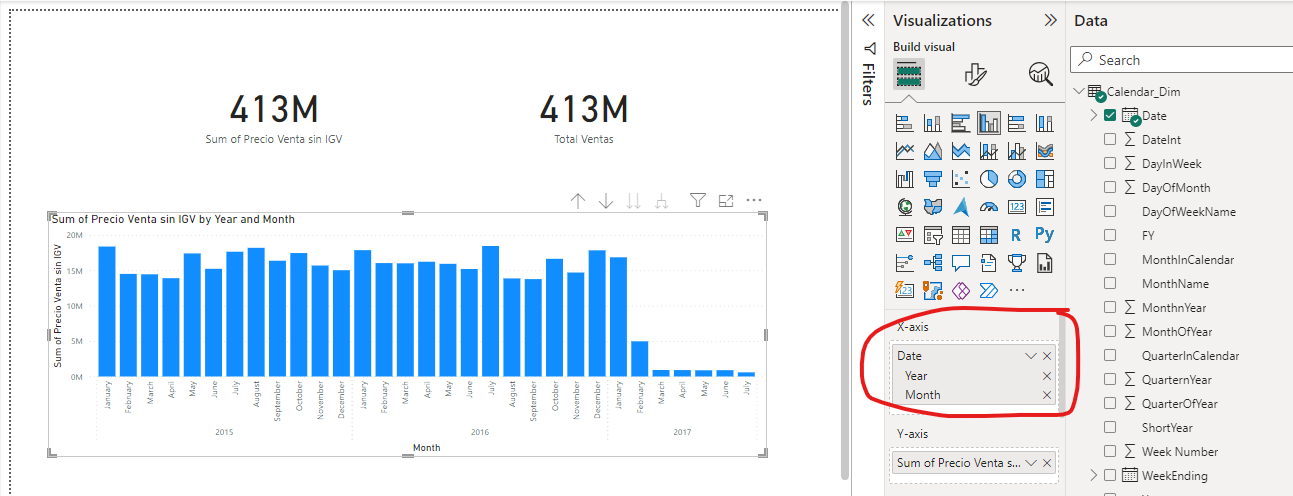
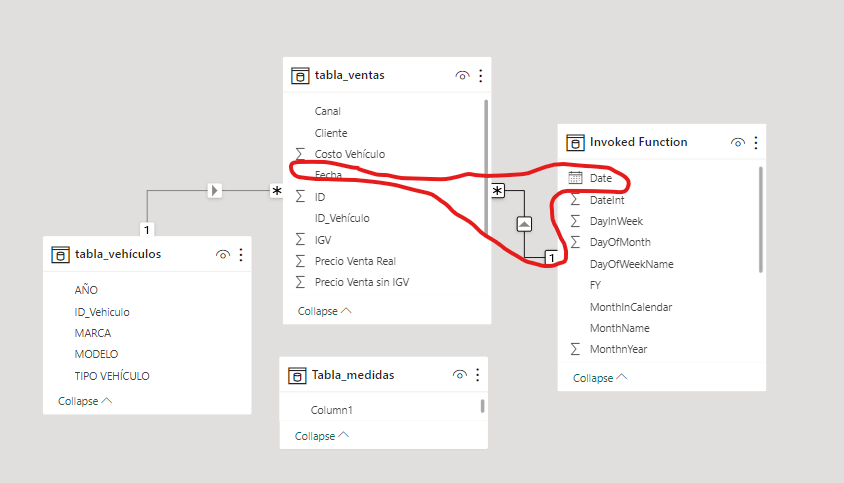
An example of the consideration of time in our analyzes is the following. Let's consider the following Query, which is created by going to the option "Insert Data" -> "Blank Query". Then, we right click on the blank query generated in Power Query and select the "Advanced Editor" option. Immediately afterwards, a pop-up screen will appear to make queries by Macros language. Suppose then we enter the appropriate code that generates a table that continuously contains all the dates from a start date in question to an end date of interest.



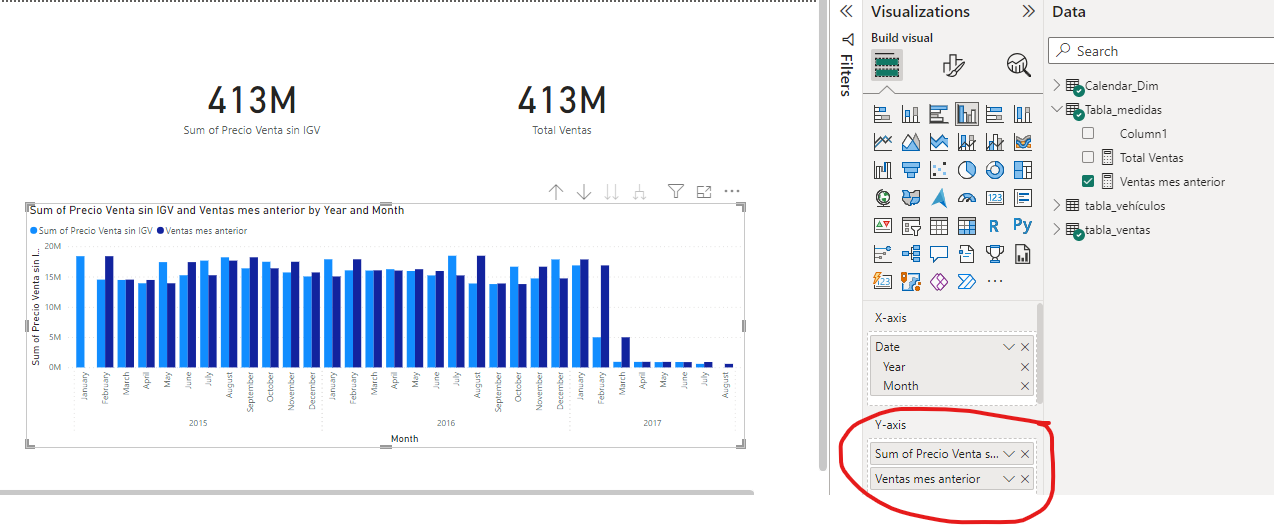


Once the date table is created, we proceed to generate the relational data model between the new table and the fact table.



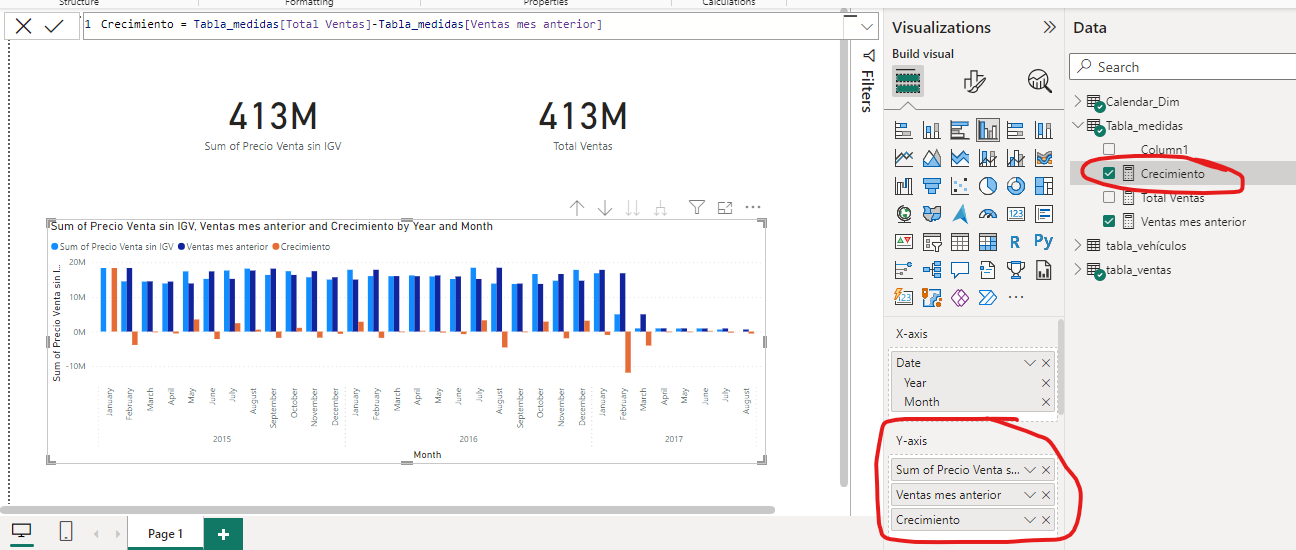
Once the relationship is created, we can generate a visualization of the sales time series throughout a year in all its months. However, by entering the following DAX command we generate a metric in the metrics table which contains the sales for the month directly prior to the month of interest.





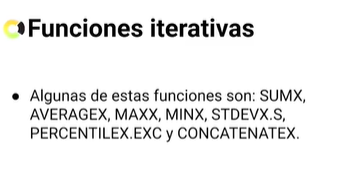
Now, we generate the growth metric as the difference between the sales made in the current month and the previous month.





**5.5– X Iterators**

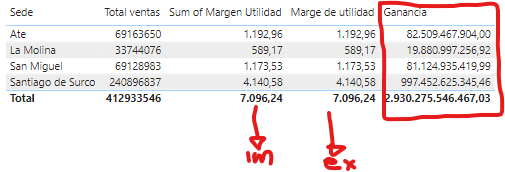
Iterators in DAX operate for the same purpose as iterators in Excel, which are applications of formulas to a set of rows-columns. However, the way DAX works is a little different, being generally for the case of X iterators, a direct iteration by rows to later apply the desired formula on the result of this iteration done on each row. Which can lead to erroneous results if not well understood how to use them. The following functions corresponds to the most common X-iteratiors.



The X iterators are used to explicitly tell Power Bi through DAX language, to execute a function taking values from specified columns, applying this function to each row of the columns in question and perform this process iteratively.

For example, consider a table where we have displayed the total sales by location. Then, we have included the profit margin by location. If we wanted to know the profit per location, we would simply multiply the total sales per location by the profit margin.

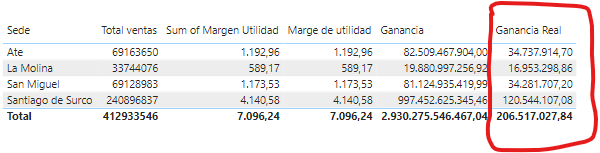




If we do this by simply multiplying the column-fields in question, we would get a wrong result because DAX would first calculate the total sales and the total margin per location and then multiply these totals, i.e. it first operates in a columnar way and then applies the formula. This erroneous result can be corrected if we use an iterator X, in this case SUMX since we want to know the total profit per site. To do this we enter the formula shown below which applies the function set by row and then finds the total by column.

Operating in the opposite way, the X iterators operate by rows and then by columns, while without the X iteration, DAX operates columnarly and then row-wise.





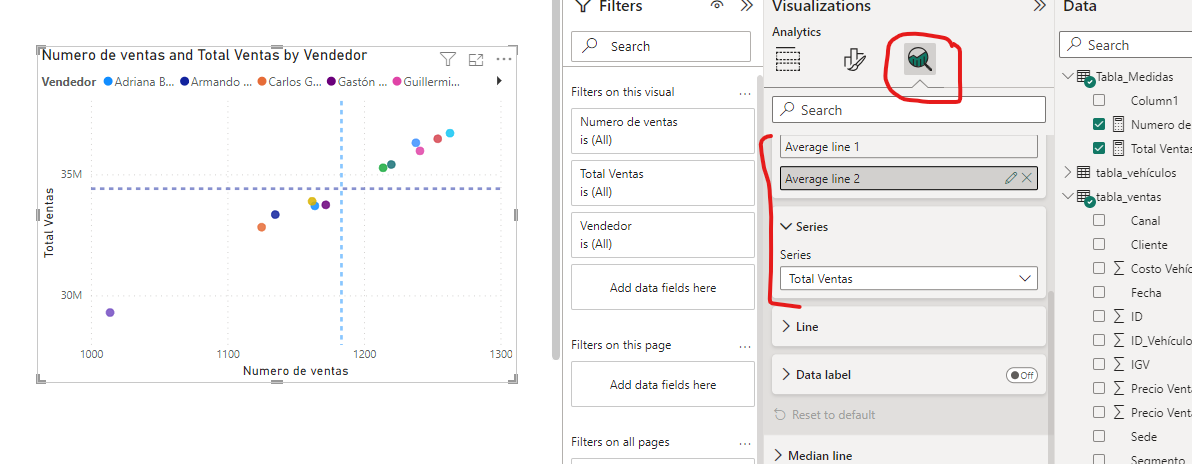
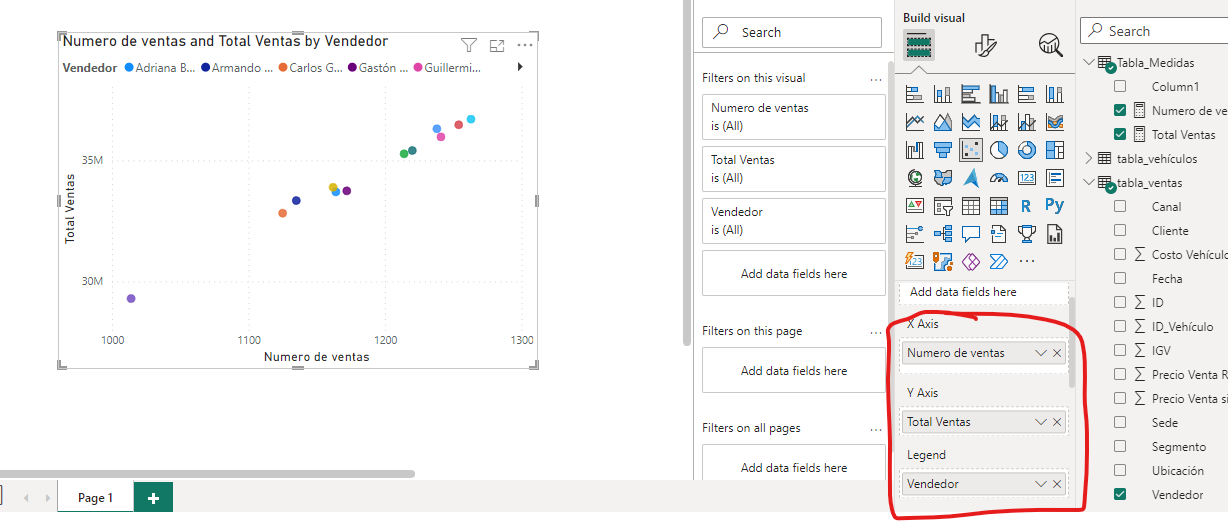
**6. Data Analytics**

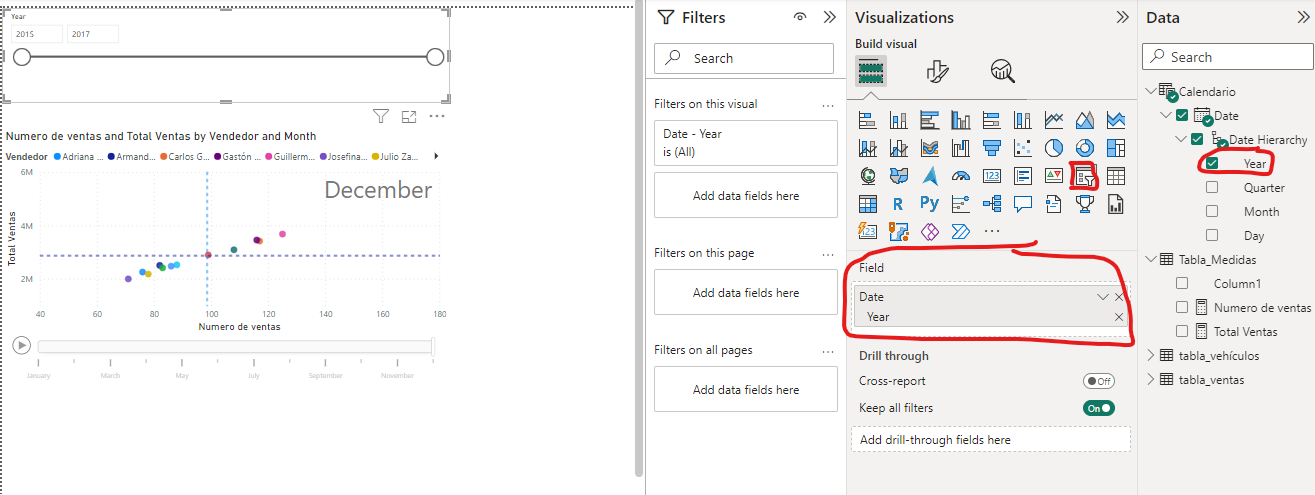
Power BI offers different data analytics tools focused on story telling. For example, it allows you to filter data by category, generate lines of averages, maximums, among others, and observe the theoretical evolution of the data.

For example, to create a filtered scatter plot with story playback we select scatter plot as visualization, we implement in the x and y axes the data we want to observe.

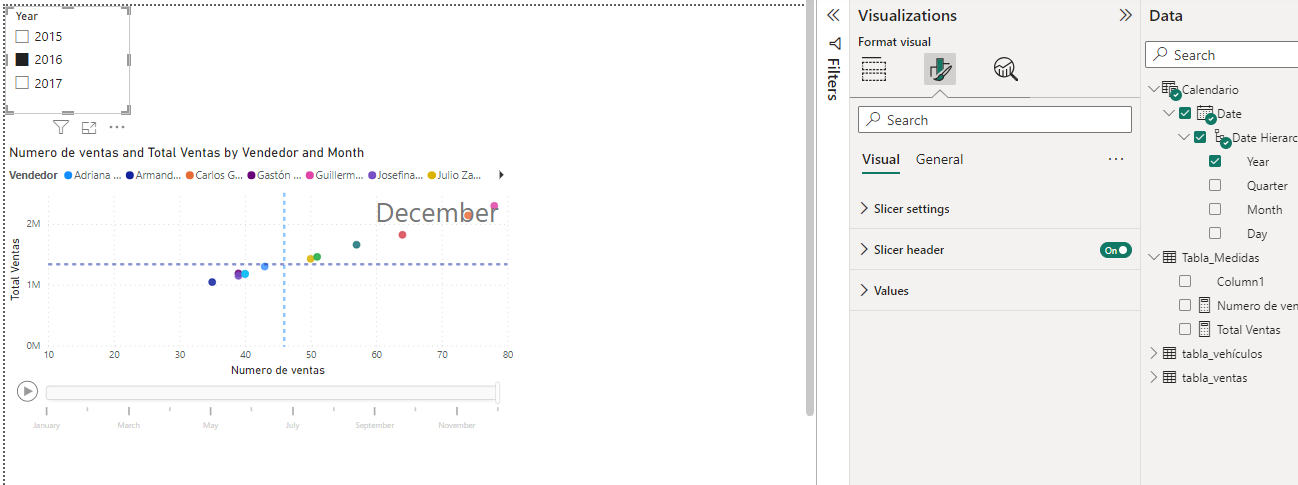
Then, in legend, we add the categories we want to compare in case the initial variables inserted in x and y have many subcategories.

Then, we go to "Format" -> "Play Axis" and add the time data that we want to evolve with the visual. Finally, we add a filter by slicer, we go to "Slicer settings" and click on "vertical list" to generate a dynamic chart filtered by year criteria in this case.

****

****

****



On the other hand, we can extend the study of time series by implementing trend lines, averages, maxima, minima and constant lines entered by the user to a time plot. It is also possible to perform time series forecast analysis where Power BI employs a Gaussian smoothing internally to predict a certain number of stipulated points, with a certain margin of confidence. Also, if the trend is too steep in the last part, it is possible to have Power BI omit a certain number of end points in order to obtain a prediction closer to the past trend.

