



spagobi

User Manual

Release 5.X

Copyright © 2014 Engineering Group, SpagoBI Labs. All rights reserved. All other trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

WELCOME.....	3
TARGET AUDIENCE.....	3
SpagoBI INTERFACE	4
Login.....	5
Main menu.....	6
Home Page.....	6
Analytical documents menu.....	6
<i>Document browser</i>	7
<i>My Favorites</i>	10
<i>My Analysis</i>	11
<i>My Data</i>	14
<i>Select language</i>	14
<i>Select role</i>	15
<i>Infos</i>	15
<i>On-line help</i>	15
FEATURES	17
The Analytical Document.....	18
Create a Self-service Dataset.....	20
Worksheet.....	23
<i>Options for Pivot Tables</i>	26
<i>Calculated fields</i>	27
<i>Static filters</i>	28
<i>Dynamic filter</i>	29
COCKPIT.....	31
<i>Design Widgets</i>	33
<i>Associations</i>	34
QBE.....	37
<i>Calculated fields</i>	43
<i>Bands</i>	45
<i>Join-type relationships</i>	46
<i>Alias</i>	49
GIS.....	52
GEO.....	54
OLAP.....	55
<i>Configure the navigation cube</i>	56
<i>Drill operations</i>	57
<i>Hide empty cells</i>	58
<i>Invert axis</i>	58
<i>Save OLAP navigation</i>	59
REFERENCES	60

WELCOME



This document is intended to provide you with detailed guidance on the environment and main analytical tools of SpagoBI - the 100% open source Business Intelligence suite supporting traditional reporting and charting tools, as well as innovative solutions for advanced analytics, available for free download from OW2 Forge (forge.ow2.org).

TARGET AUDIENCE

Realized and managed by Engineering Group's SpagoBI Labs, this document addresses any user who wishes to design and develop a Business Intelligence project or application using SpagoBI suite. It is assumed that readers have basic Business Intelligence, SQL and XML skills.

SpagoBI INTERFACE



This section focuses on SpagoBI user interface, highlighting basic navigation features.

Login

To ensure data protection, users can access SpagoBI through a Log-in page that requires authentication credentials (user name and password).

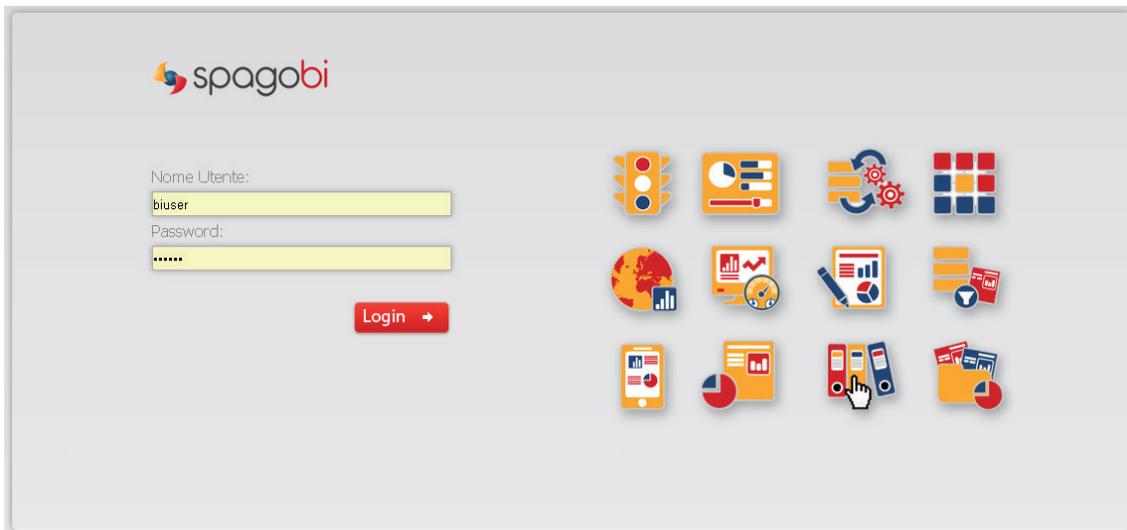


Figure 1 - Log In

Once logged in, SpagoBI interface will be displayed, which includes a sidebar on the left and a main page, whose content can be configured by the administrator according to project requirements. You can choose to keep the traditional welcome page or to customize it, for example through a predefined analysis as you can see in Figure 2.



Figure 2 – SpagoBI Welcome Page

Main menu

SpagoBI main menu consists in a set of icons associated with the following basic features:

	Home Page	Return to the Home Page, i.e. the landing page after successful authentication
	Documents Menu	Access the menu, including the analysis that were previously set up by the system administrator
	Document Browser	Show the archive folders and related analysis
	My Favorites	Quick access to my favourite analysis
	Create Document	Self-create your analysis starting from a data model
	My Data	Upload my personal files and create new datasets to develop my analysis
	Languages	Language options
	Help On-line	Access the Wiki containing on-line documentation http://wiki.spagobi.org/xwiki/bin/view/Main/SpagoBI
	Info	Infos on SpagoBI version
	Log Out	Log out
	Default Role	Select the authentication role (available if the user is associated to more than one role)

Now let's describe in detail the above-listed features

Home Page



This button allows you to get to the Home Page with a single click. This page appears once the user logs in.

Analytical documents menu



This feature gives you access to the main menu, including the documents you are allowed to see according to pre-defined visibility rules. The content and the structure of the menu are defined by the system administrator.

This approach is typically used by end users to easily reach key documents

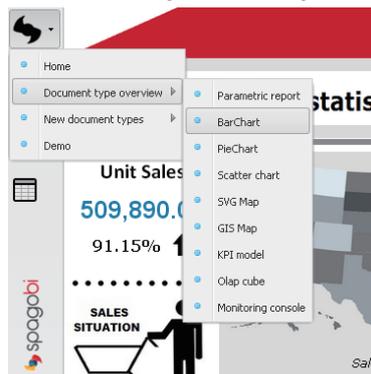


Figure 3 - Document menu example

Document browser



The **document browser** gives access to the *Functionality tree* containing all SpagoBI analysis and folders that the user is allowed to view.

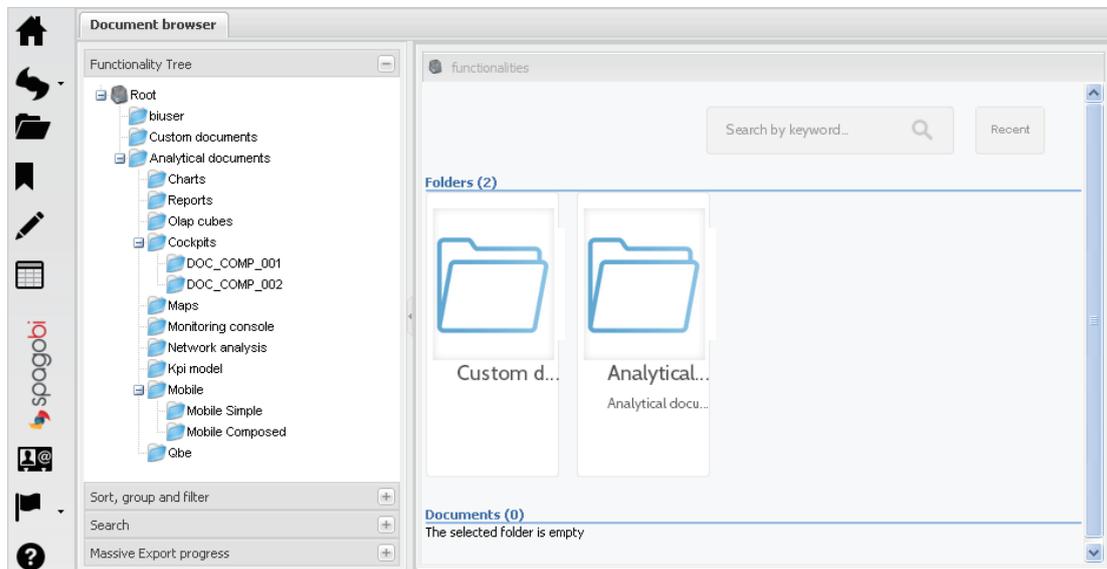


Figure 4 - Document Browser

Clicking the *document browser* button, a new page appears, which is divided into two main areas. On the left you can see the list of folders and subfolders containing the executable analysis, the right part shows the contents of your selection. By default the root node is displayed (Figure 4).

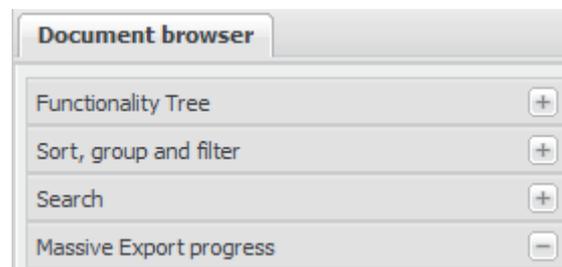


Figure 5 - Sub-menu of the Document Browser

As shown in the figure n.5, the sub-menus of the Document Browser allow you to navigate from one analytical document to another through the *Functionality tree* to perform search on them and to export them massively.

The *Sort, group and filter* tab allows users to select the display mode, as shown in Figure 6.

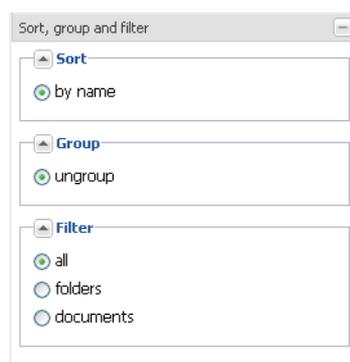


Figura 6 - Sort, group and filter

Search is the feature that allows you to search for a specific document. Enter a keyword in the **Query** section: you can refine the search by filling in the **Search in** attribute, as well as enabling advanced options to search for similar words.

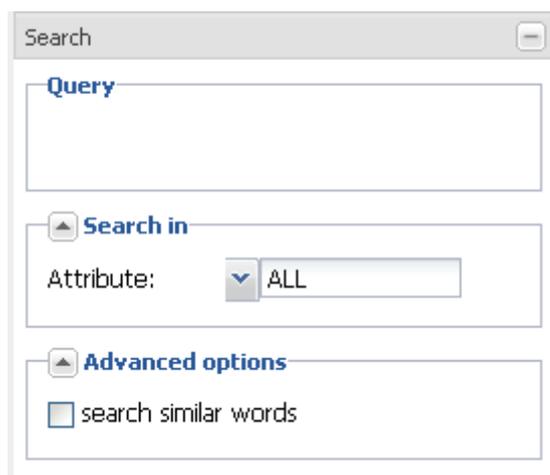


Figure 7 - Search

The **Massive Export progress** allows you to export all worksheets in the XLS format and zip them in a folder. At the end of the export process, the folder will be made available for download. In the document browser users can monitor the progress of the export process.

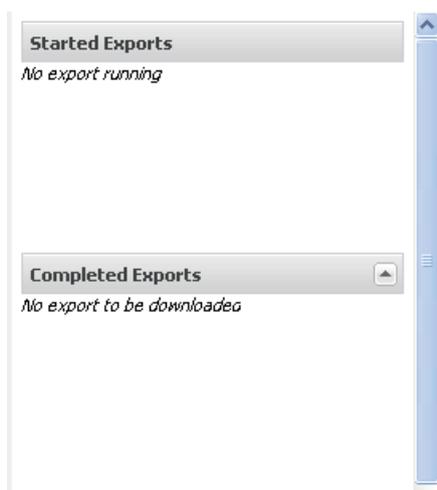


Figure 8 - Massive export

To schedule the document execution, select **“Schedule”**. On the other hand, **“Export”** allows you to execute the analysis on the fly and save it directly into a folder of your choice.

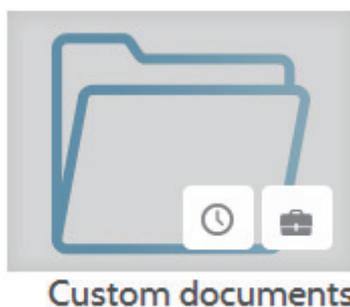


Figure 9 - Selected folder

Being both asynchronous operations, the user can activate these functions and access the results at a later time.

Let's go back to the sub-menu of the document browser showing the folder tree and examine in detail the right side of the page.

Also in this section you can perform a search, by filling the field. The search process only applies on the current folder. When completed, the list of contextual folders and analytical documents appears.

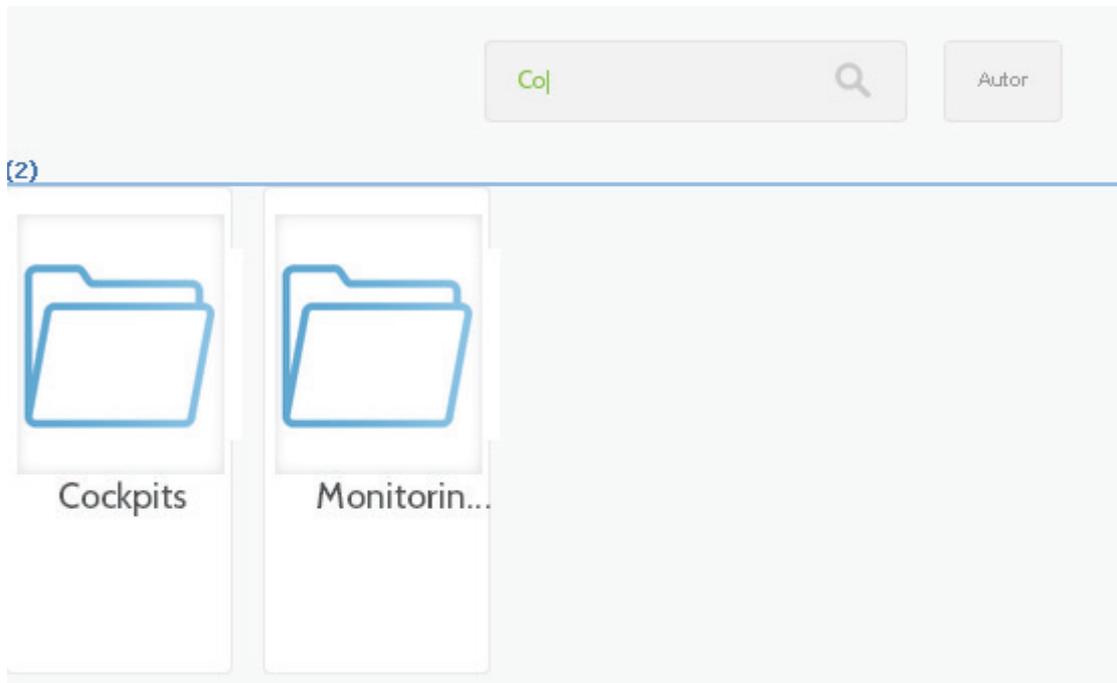


Figure 10 - Contextual search

You can view the metadata of each analytical document by clicking the  icon, which appears in transparency on mouse over.

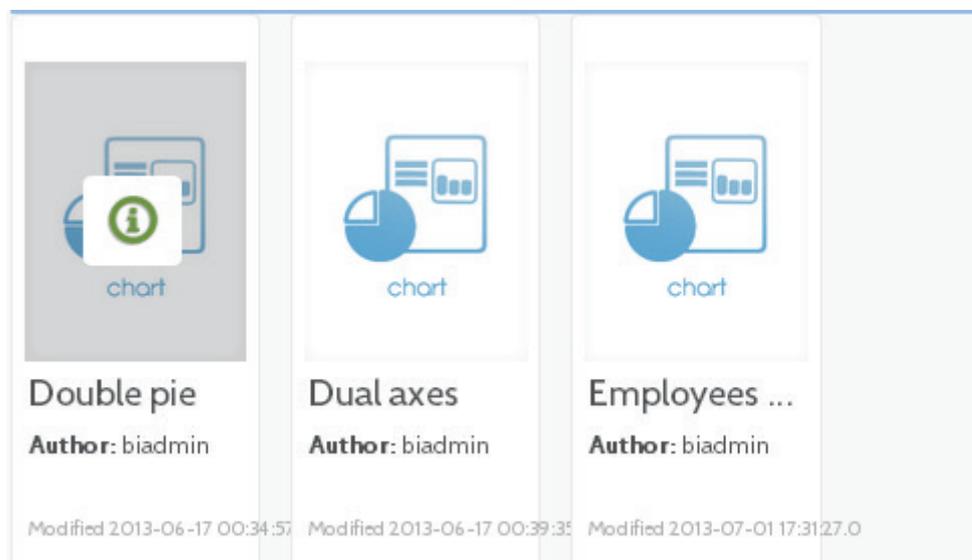


Figure 11 - How view metadata

Generally, metadata are descriptions that can be configured by administrators and consulted by users (see Figure 12).

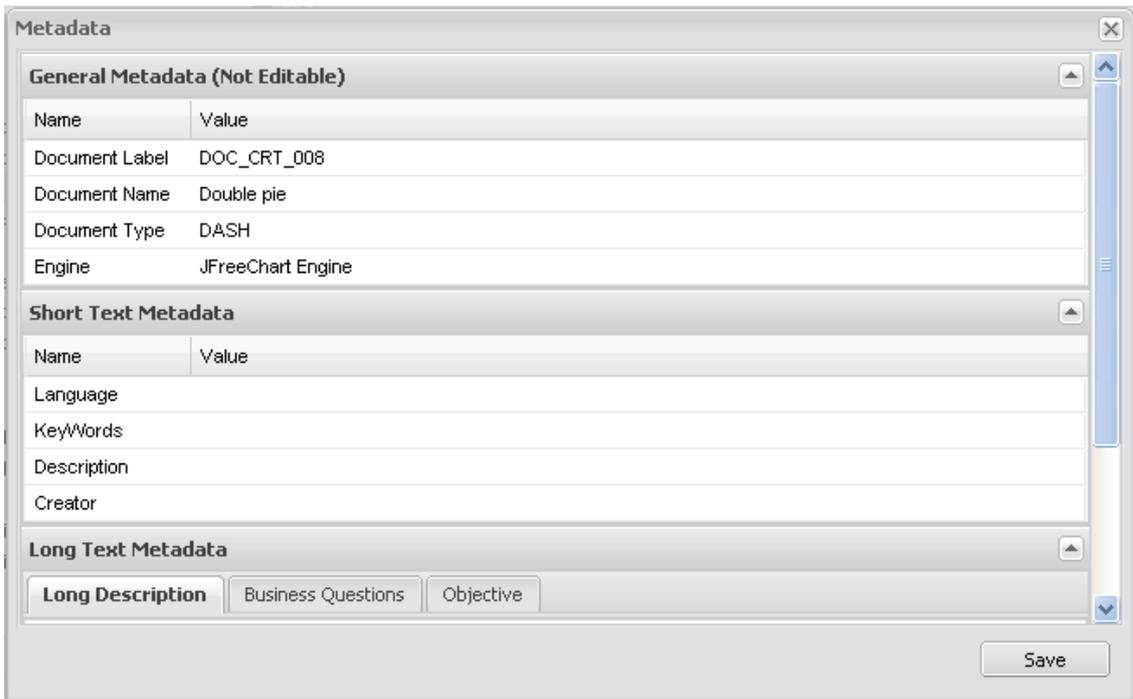


Figure 12 - Metadata

My Favorites



This menu item allows users to view pre-defined favorites, according to specific criteria, namely:

- **Remember me**, including the documents that were previously saved as Favorites (see *"Add to Favorite"* in the paragraph on analytical documents)
- **Most popular**, including most popular/clicked analytical documents, which the user is allowed to access
- **My Recently used**, which includes the list of analysis that have recently been executed by the user.

Remember Me			
Most Popular			
Document	Document name	Document description	Document type
DOC_COMP_001	Sales and revenue	Sales and revenue	DOCUMENT_COMPOS...
DOC_RPT_002	DOC_RPT_002	DOC_RPT_002	REPORT
JPalo_Sales	Sales JPalo	Sales JPalo	OLAP
DOC_NTW_001	DOC_NTW_001	DOC_NTW_001	NETWORK
DOC_KPI_001	Kpi model		KPI
My Recently Used			
Document	Document name	Document description	Document type
DOC_COMP_001	Sales and revenue	Sales and revenue	DOCUMENT_COMPOS...
DOC_OLAP_001	Sales cube		OLAP
DOC_RPT_001	Infographic	Infographic by report	REPORT
DOC_MAP_002	Sales and costs by regi...		MAP
DOC_RPT_002	DOC_RPT_002	DOC_RPT_002	REPORT

Figure 13 - My Favorites

When the user clicks on the item, the analysis is executed.

My Analysis



In the My Analysis area users can self-create their analysis with different levels of maneuverability on data and also access them, once created.

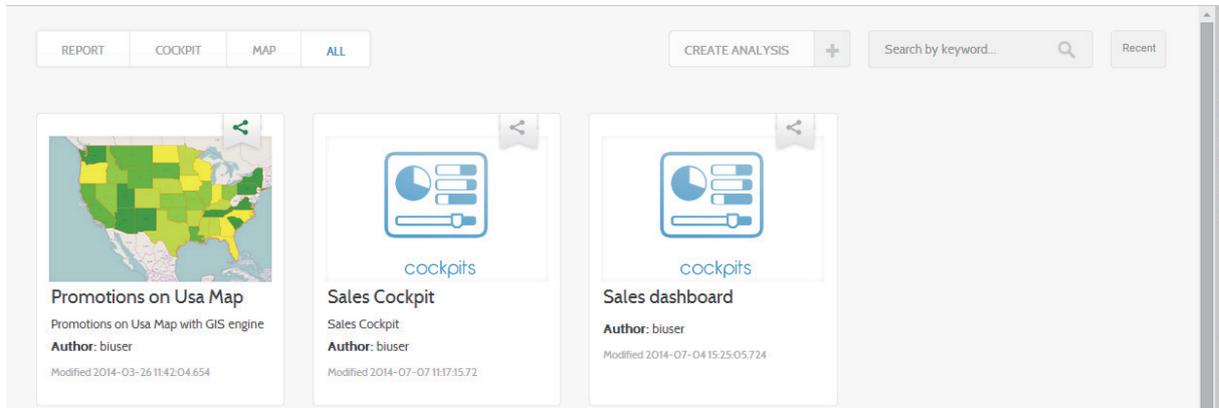


Figure 14 - Create Document

By clicking the My Analysis section, SpagoBI opens a new page containing users' analysis, including different selection and search fields.

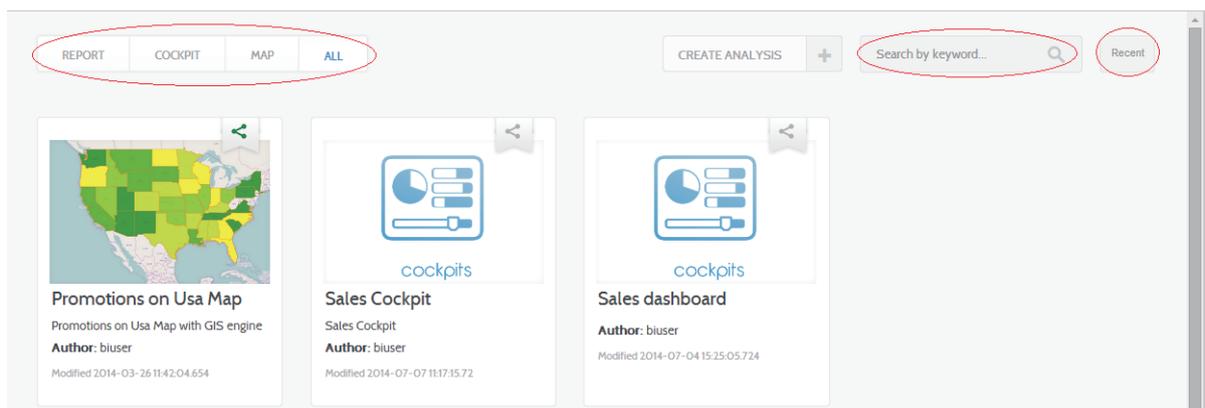


Figure 15 - Search Document

Specifically, in the upper left corner, four buttons let the user sort the analysis by typology, while in the upper right corner a menu allows the user to sort them by date of creation, name or author. It is also possible to find the analysis by typing its name or a key word in the search-by button.



In this section users can create their own analysis. The Create Analysis button displays a wizard window that allows you to select the type of document to be produced.



For all details on cockpits, please refer to the dedicated paragraphs.

By selecting ad-hoc reporting analysis, SpagoBI opens a new page with two tabs.

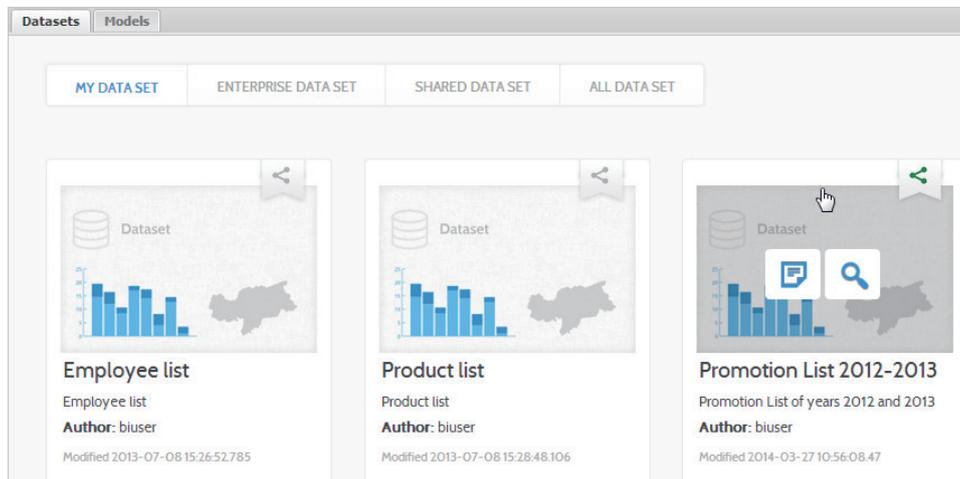


Figure 16 - Ad-hoc

The first tab -Datasets- includes

- **My Dataset** contains the datasets built by the user
- **Enterprise Data Set** gives access to the data sets the user is allowed to see according to the visibility rules previously set by the administrator
- **Shared Data Set** refers to the Data Sets created and made available by end-users so that they can be used by other users.

The second tab -Models- includes data models.

The difference between these elements follows: while in the first tab users use pre-defined data sets, in the Models tab users can inquire data models and create new data sets.

Through a data set, users can directly access the Worksheet designer and self-create their analysis, or generate further queries through the QbE.

Moving the mouse pointer over the dataset icon, two transparent buttons appear:



The first option allows you to access the Worksheet designer (see Figure 17).

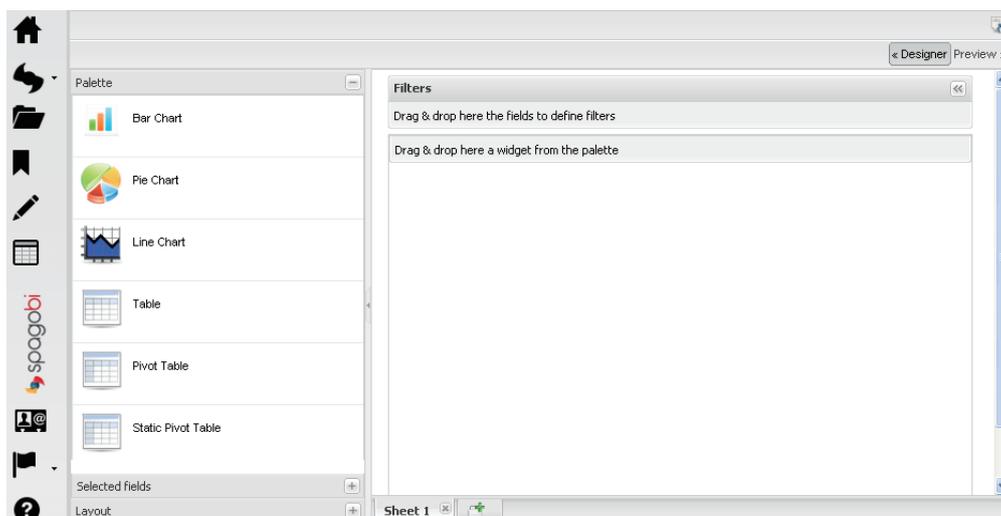


Figure 17 - Worksheet designer

The second option leads users to the QbE (Query by Example) tool (see Figure 18).

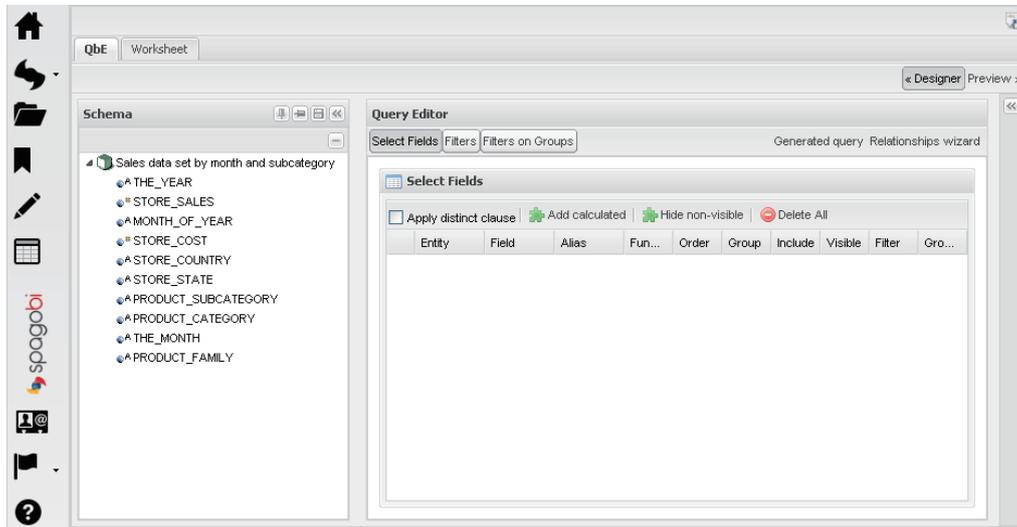


Figure 18 - QbE

Through the *Models* section, users can view the icons related to the meta models that had previously been made available by the administrator (see Figure 19). Remember that while the meta models allow high levels of manoeuvrability on data, the datasets allow users to perform their analysis using the available fields.



Figure 19 - Models

For all details on QbE and Worksheet, please refer to the next paragraphs.

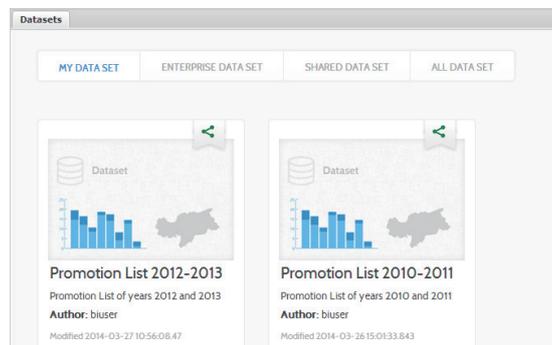


Figure 20 - Dataset Type

A specific type of data set concerns analytical tools supporting Location Intelligence. These datasets are sorted by category (My Data Set, Enterprise Data Set, Shared Data Set) as before, but the additional "Show Map" icon allows you to view data directly on the map.



My Data



In the *My Data* section users can access the *Self-Service BI* features. Once entered this section, users' data sets appear (see Figure 18).

Users can modify the existing data sets or create new ones starting from a data model (or any QbE), or you can upload a CSV or XLS file.

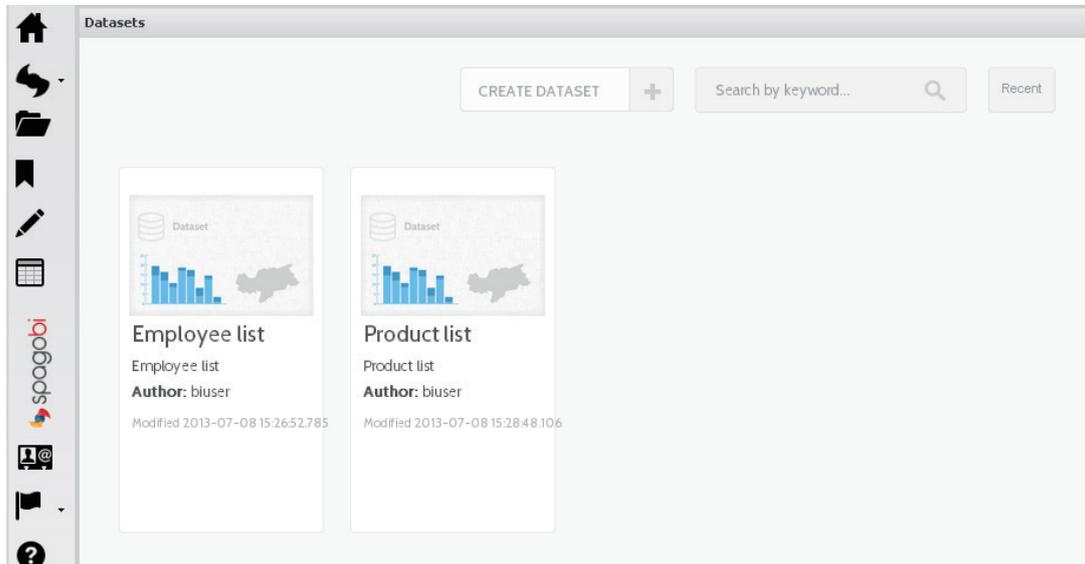


Figure 21 - My Data

As previously mentioned, Users can open either the QbE or the Worksheet designer, or again:

- Use the dataset through the geographic engine, which allows associations between the data included in the dataset and a map (see GEO paragraph) 
- View details . According to the nature of the dataset, the upload file wizard or the QbE interface will appear.
- Delete unused datasets.



Figure 22 - Self-service dataset

Select language



The *Change Language* button included in the toolbar allows users to select the language of SpagoBI environment.



Figura 23 - Change language

Select role



If a user is associated with more than one role, SpagoBI requests the user for the default role. The user can select it when executing a document, or right after authentication by clicking the  icon located in the toolbar.

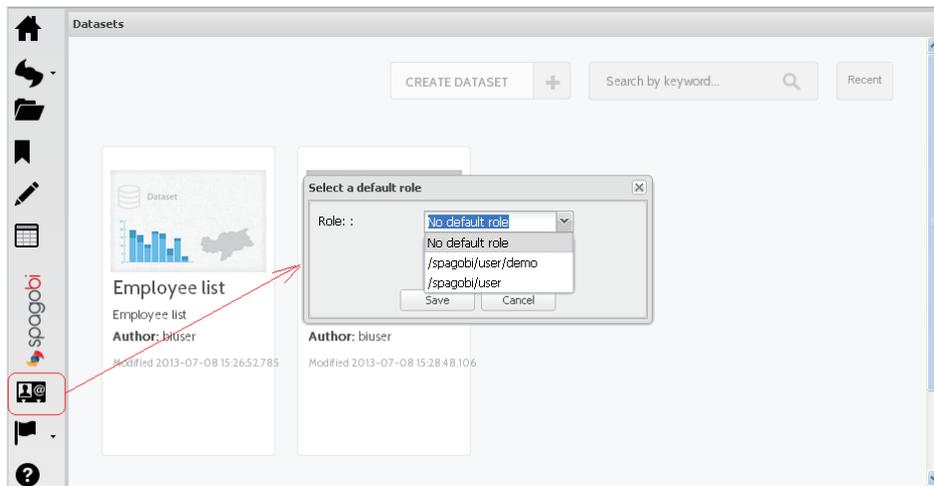


Figure 24 - Select role

Infos



This feature allows users to view details on SpagoBI version.



Figure 25 - Infos

On-line help



This icon leads to the documentation on SpagoBI suite available on the Wiki.

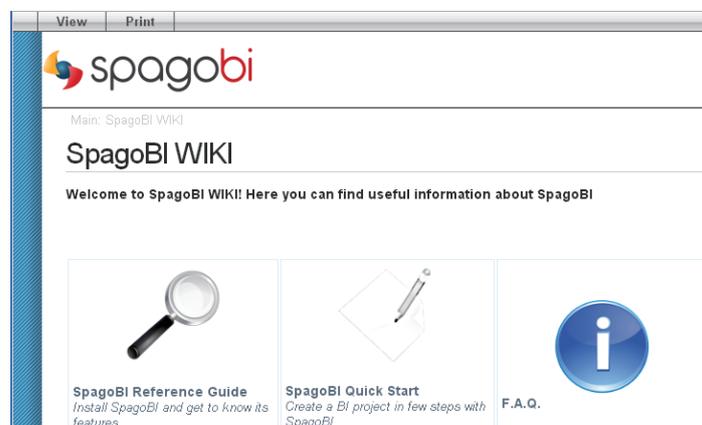


Figure 26 - On-line help

FEATURES



SpagoBI suite provides a complete range of analytical capabilities, supported by more than thirty analytical engines. Thanks to this high level of flexibility, users can perform their analysis through different engines, and easily obtain the solution that most suits their specific needs. Now let's focus on some of these analytical capabilities.

The Analytical Document

The analytical document is an analysis showing data coming from a graphic data source. The output of this analysis varies according to users' needs - e.g. a chart, a report, an OLAP cube, an interactive cockpit. Once the user has identified the analysis in the *document browser* or in the menu, the analysis is executed. If any, SpagoBI asks the user for the required parameters; otherwise, the analysis is directly executed. In Figure 27, a report shows the analysis on the customers of a grocery store requiring two entry parameters: age range and product category. Once these parameters are set, click the  icon, so that they are applied on the report.

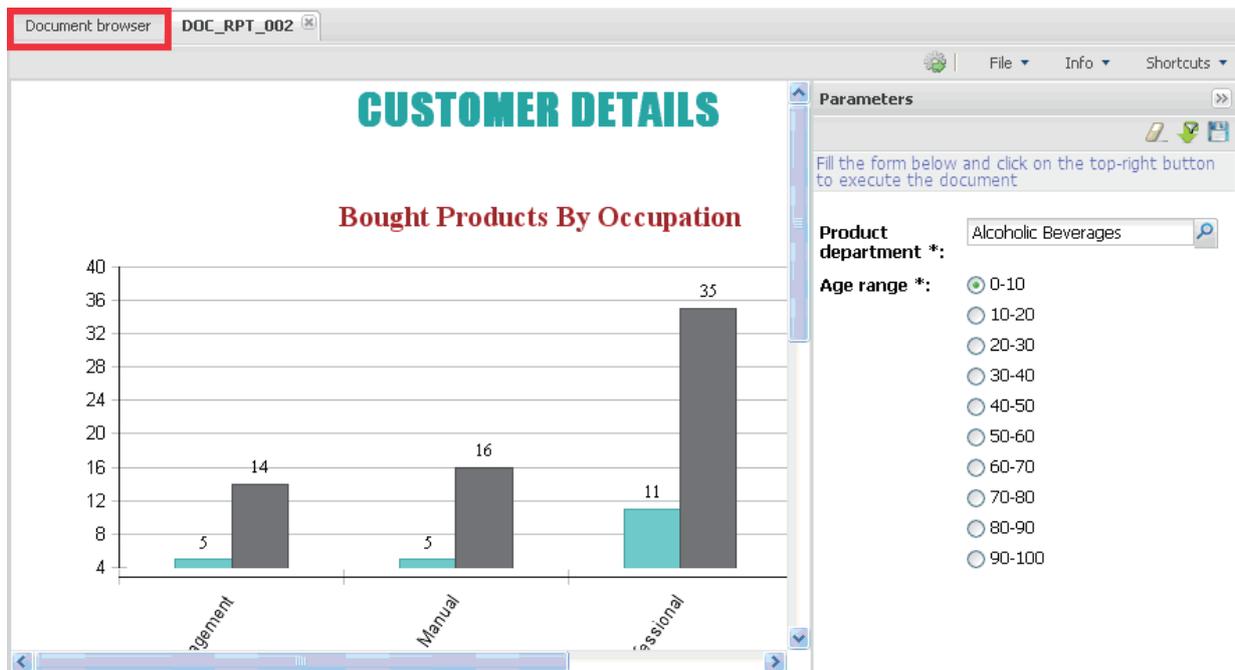


Figure 27 - Execution of an analytical document

At this point the report -just like any other analytical document- is visualized full screen; users can return to the *"document browser"* by clicking on the label located at the top left (see Figure 27). In order to show or hide the parameter window, click the  button located at the top right.

Generally, all documents inside SpagoBI environment have the same toolbar with different features



-  Execution of analysis. Users can re-execute the analysis with different parameters.
- **File** menu:

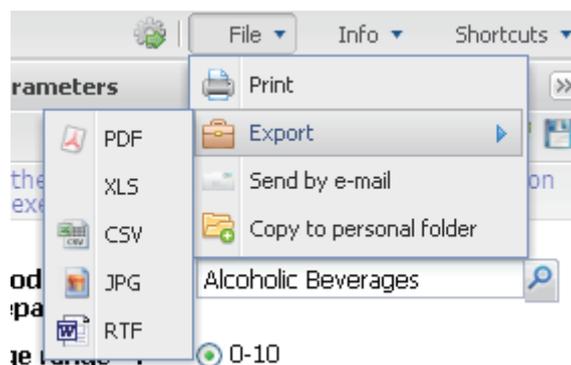


Figure 28 - File menu

-  Print
-  Export
-  Send e-mail
-  Copy in personal folder

• **Info** menu

-  Metadata
-  Note
-  Rank

• **Shortcut** menu

-  Show Favorites
-  Add to Favorites
-  Show customize view
-  Add to customize view
-  Show scheduled executions

Finally, the parameter configuration can be saved for future use. This is particularly useful when the document includes several customized parameters. This feature is accessible from the toolbar located in the parameters section.

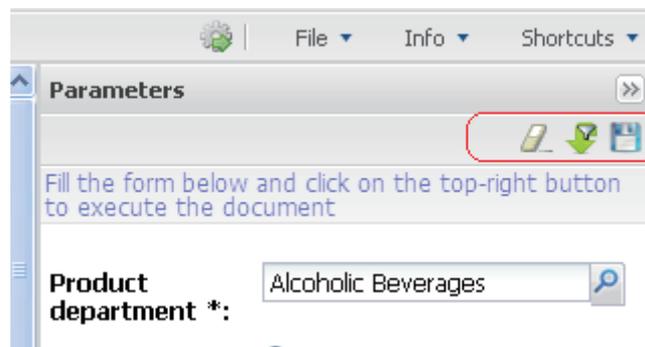


Figure 29 - Saved parameters

-  Delete inserted values for parameters
-  Open a window listing the saved parameters, so that the user can select or modify them.

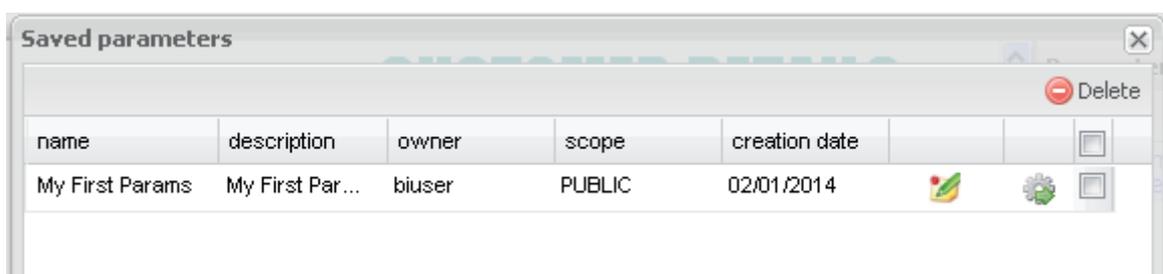


Figure 30 – Saved Parameters

 Save the parameters. **Public** means visible to all users having the same role as the authenticated user; **Private** means visible to authenticated users only. (migliorare immagine 28)

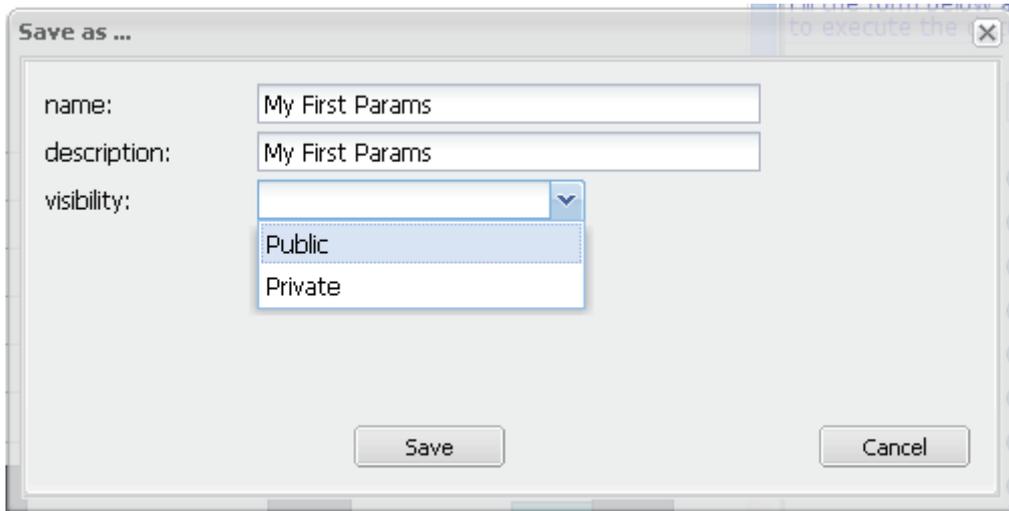


Figure 31 - Paramters - Save as

Create a Self-service Dataset

As previously mentioned, users can create new data sets using the  button, located in the "My data" menu. This paragraph is a step-by-step guide to create a self-service dataset.

Step 1: Select the files from which you need to retrieve data (see Figure 32). CSV and XLS formats are supported.

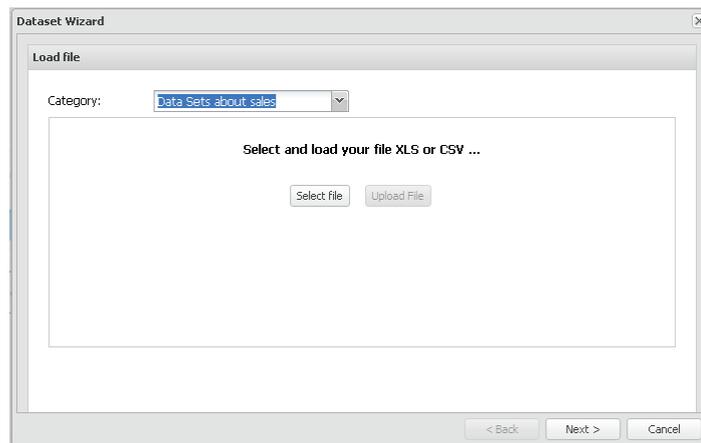


Figure 32 - Step 1: Create data set wizard

Step 2: After the file upload is completed, specify how columns are delimited (comma or semi-colon), how strings are identified (quote characters) and encoding.

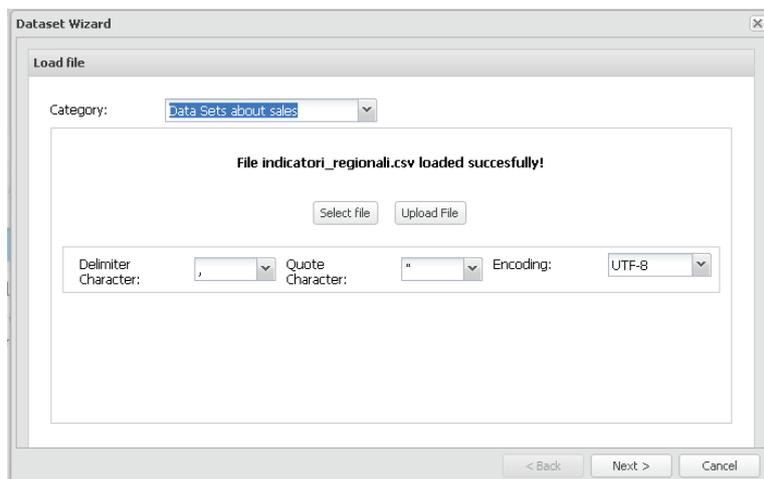


Figure 33 - Step 2 wizard

Step 3: Define meta data. Once you have defined the delimiter characters and the encoding, SpagoBI automatically identifies column headers (fields). Assign the measure or attribute value to each file and specify the data type (integer, string, etc.).

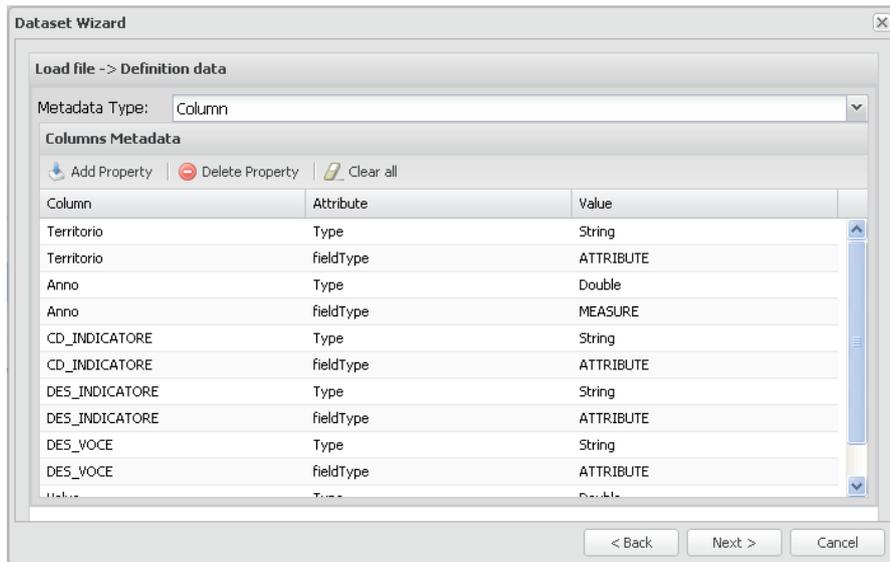


Figure 34 - Step 3 Wizard

Step 4: Data preview.

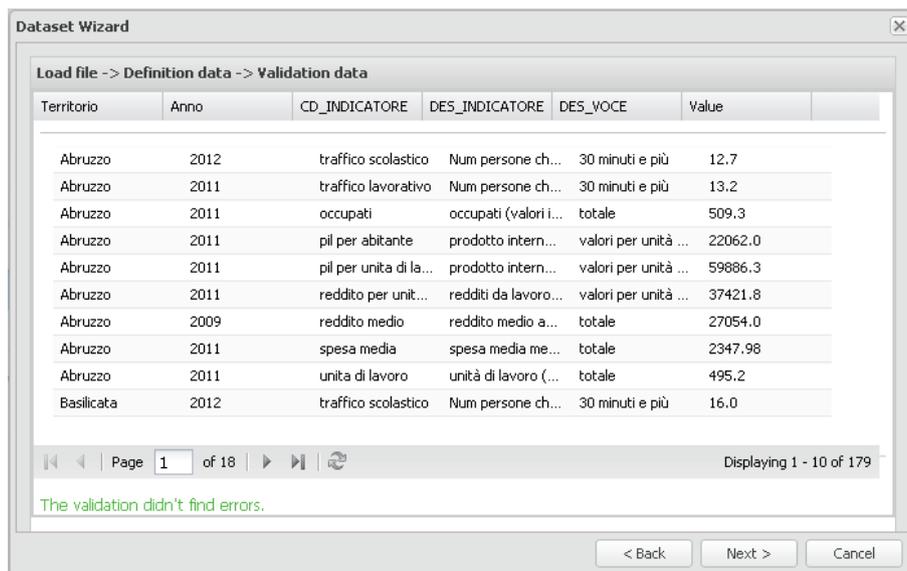


Figure 35 - Step 4 Wizard

Step 5: Save your dataset.

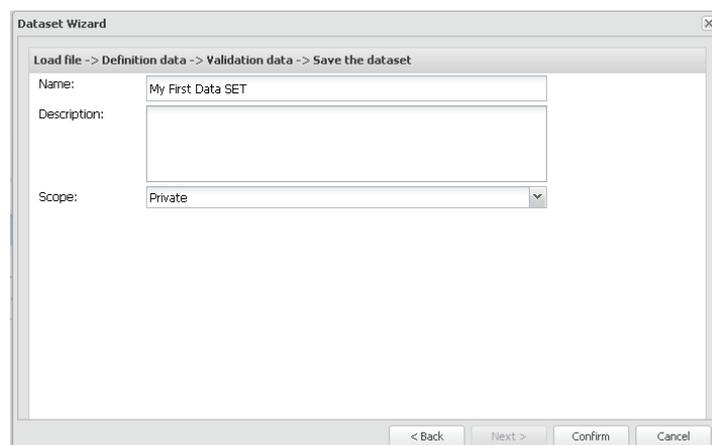


Figure 36 - Step 5 Wizard

A dataset can be used in different ways. This is the main topic of the next paragraph.

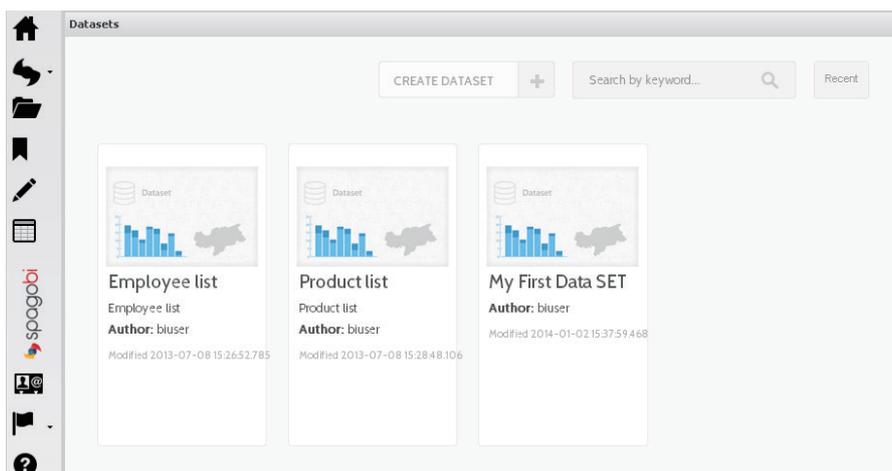


Figure 37 - New data set

Worksheet

The Worksheet is the tool that allows users to analyze data extracted through the QbE or data sets, giving them a visual representation through a user-friendly designer.

Just like the QbE, the window is composed of a *designer* for building the layout of the analytical document, and a *preview* section showing the final result.

The designer includes an area with three submenus on the left and the main window on the right. The Palette menu lists the tools and widgets to be displayed in the main window:

- histograms
- pie charts
- line or time charts
- flat tables
- interactive pivot tables
- static pivot tables

The other two submenus refer to the fields that can be dragged and dropped into the widget (*Selected fields*) and to the layout of each single sheet (*Layout*).

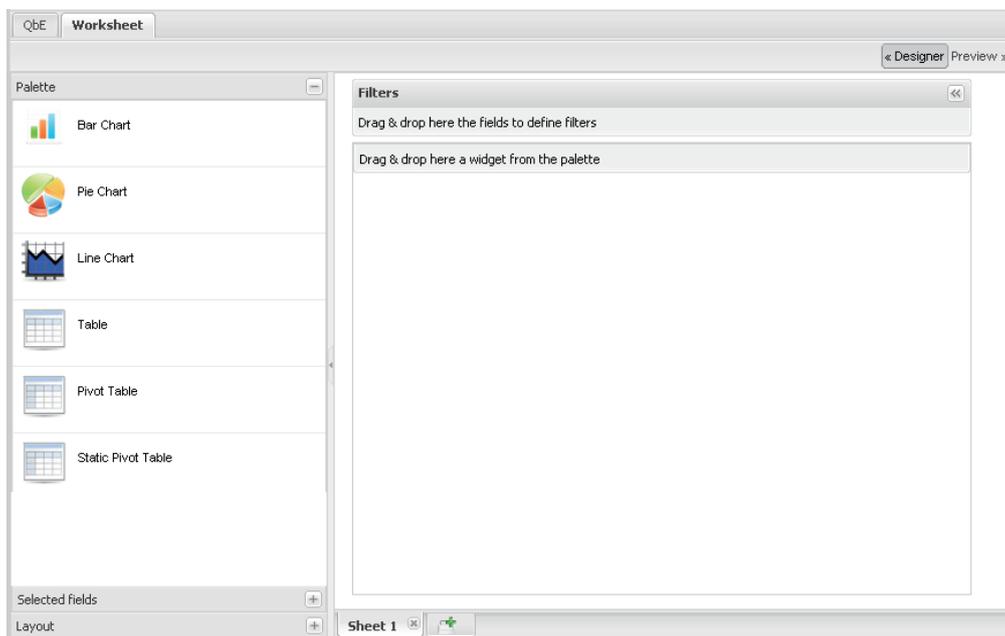


Figure 38 - Worksheet Designer

It is worth noting that the Worksheet supports the multi-sheet model. Each sheet can include maximum one element or widget.

Let's see in detail. Drag and drop the desired element from the *Palette* to the right part of the window: once you release (e.g. Figure 39), the element is shown.

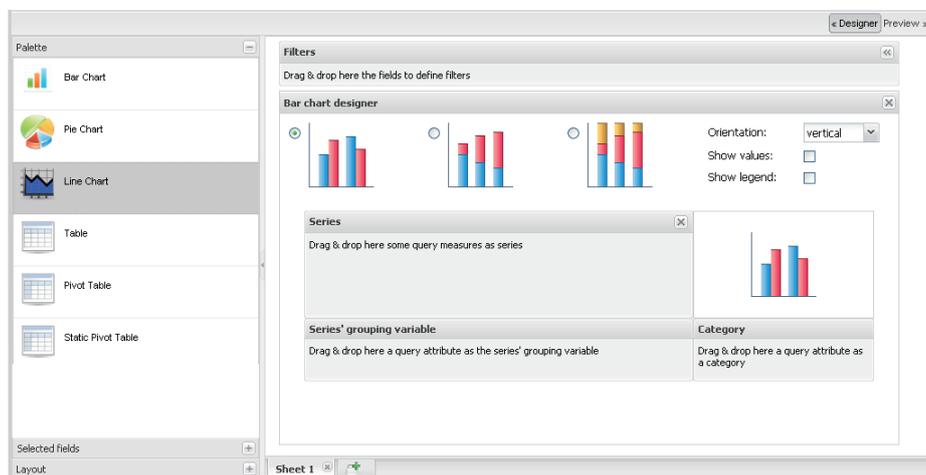


Figure 39 - Drag & Drop widget

Therefore:

- Drag the bar chart into the right part of the window to create the histogram
- Specify the fields to be used as category and measure (*Selected fields* submenu)
- Set the layout of the histogram (side-by side, stacked or %stacked)
- Define the orientation and layout of the legend, as well as of the values displayed beside the bars in the chart
- Click the *preview* button to check the results.

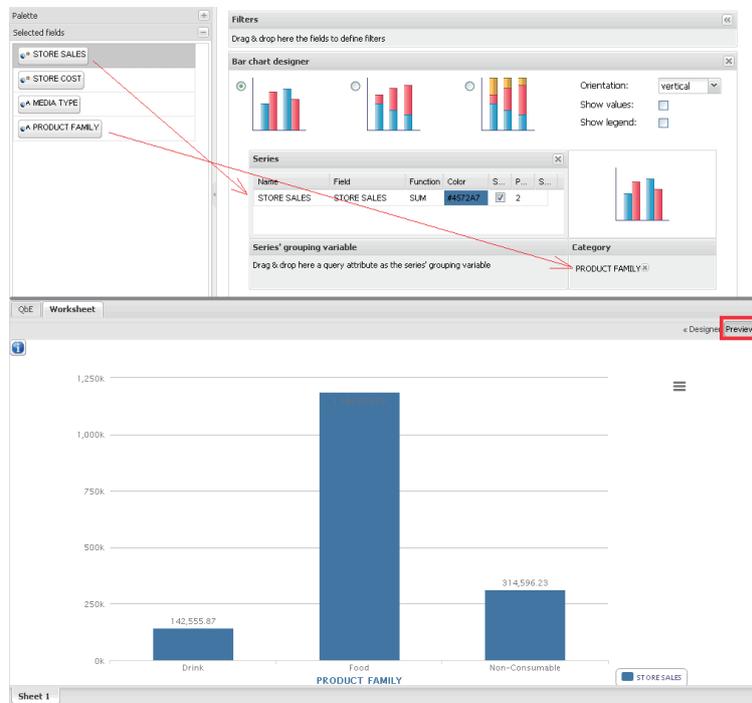


Figure 40 - Designer & preview

Users can get back to the designer anytime and add a new sheet or modify the existing one. To add a new sheet, click the  icon (see Figure 41).

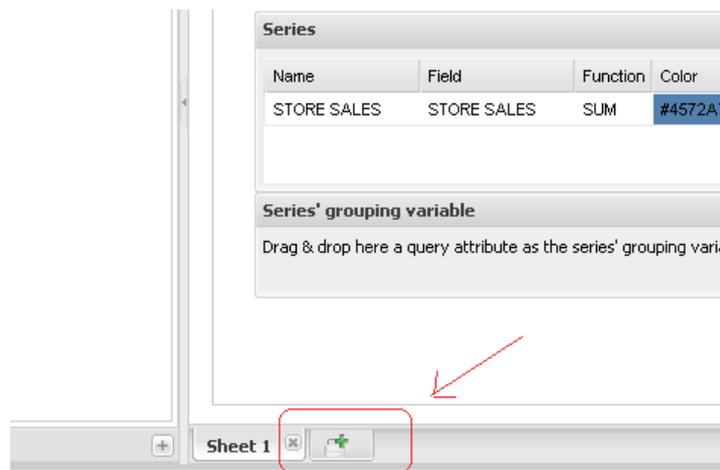


Figure 41 - Add a new sheet

For example, in order to add a pivot table, drag and drop the widgets "Pivot Table" as in Figure 42.

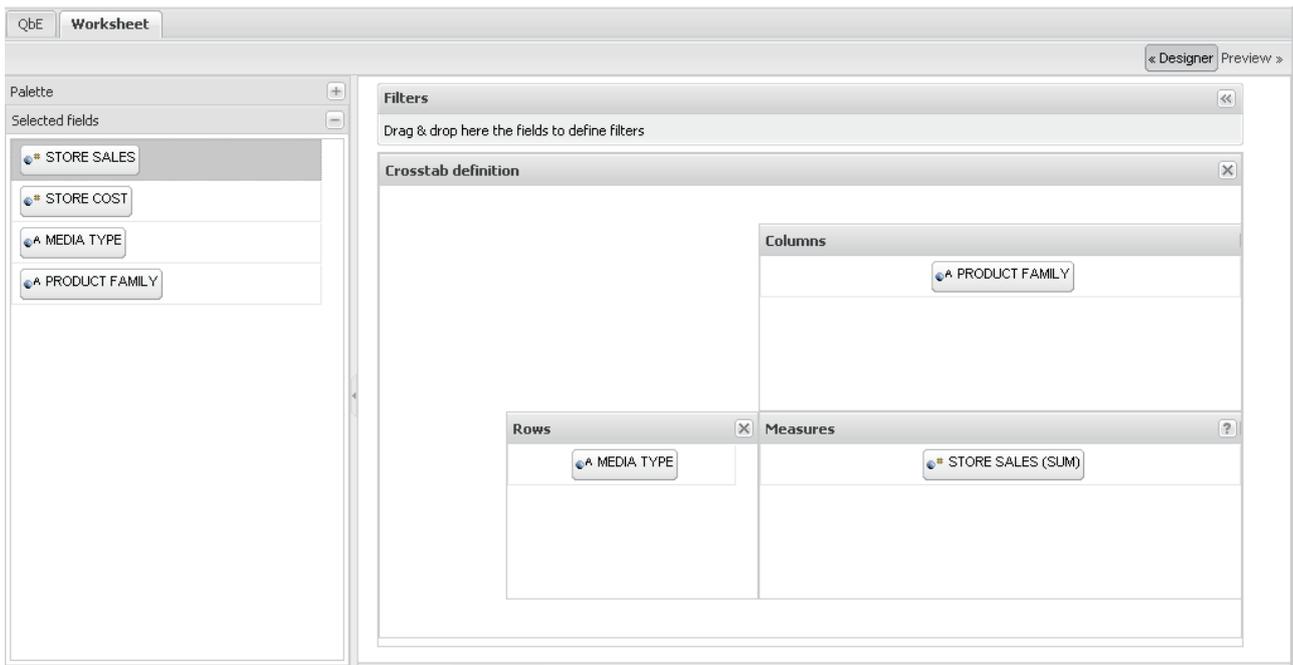


Figure 42 - Pivot table

The process for creating a pivot table is very similar to the development of a chart. Drag fields to define the contents of rows, columns and measures (see Figure 43). Click preview and the Worksheet will show the graph that you have previously created (first sheet), as well as a second sheet including the new pivot table.

MEDIA TYPE	PRODUCT FAMILY		
	Drink	Food	Non-Consumable
Bulk Mail	2,564.84	20,649.50	5,492.64
Cash Register	2,686.54	21,657.20	5,940.32
Daily Paper	3,052.59	25,525.33	6,725.02
Daily Paper,	4,168.42	35,766.85	9,953.28
Daily Paper,	3,853.07	31,343.16	8,181.01
In-Store Coupon	3,102.09	24,765.51	6,671.10
No Media	104,221.19	868,814.43	229,779.32
Product	3,372.51	27,091.52	7,362.42
Radio	3,005.86	25,749.66	7,116.56
Street Handout	3,139.51	27,885.75	6,994.58
Sunday Paper	2,539.61	22,054.15	5,935.60
Sunday Paper,	2,471.73	20,393.43	5,010.09
Sunday Paper,	1,960.56	16,433.38	4,307.61
TV	2,417.35	18,860.92	5,126.68

Figure 43 - Pivot Table preview

Options for Pivot Tables

To fully exploit the above-mentioned widget, SpagoBI offers specific features to facilitate reading of Pivot Table contents.

Measure aggregation:

Double click the measure element and drop it into the designer. SpagoBI shows a new window, where you can change the aggregation options.

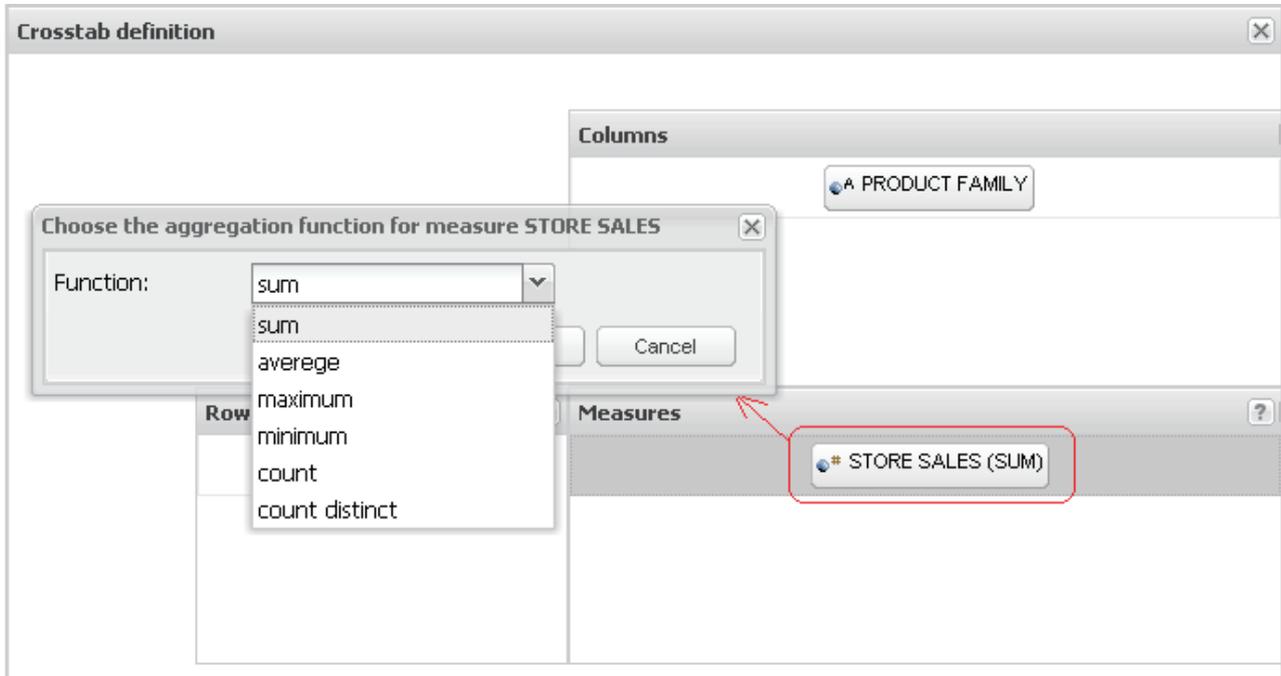


Figure 44 - Measure aggregation

Measure details:

Clicking the  button located in the right part of the *Measures* panel, a new window appears. Here users can select the measures to be visualized (columns or rows), specify how totals and subtotals shall be shown, enable percent calculation on columns/rows and set the maximum number of visible cells (see Figure 45).

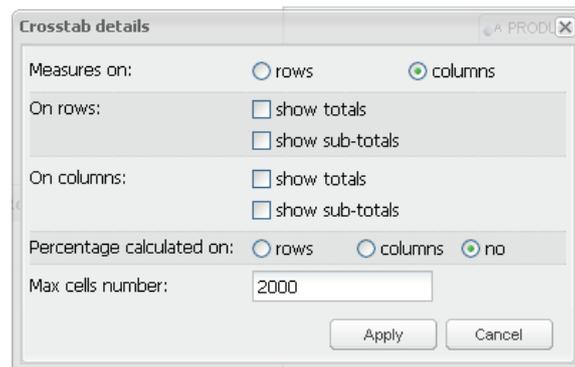


Figure 45 - Measure details



The Pivot table and the static Pivot table have different layouts: the second one does not allow you to add calculated fields but it is optimized for visualization of large tables.

Calculated fields

To improve readability of a Pivot table, you can provide additional information by defining a calculated field. In the table preview, click on the header of the measure and choose the *Add Calculated field* option.

MEDIA TYPE	PRODUCT FAMILY								
	Drink			Food			Non-Consumable		
	STORE SALES	STORE COST	ddd	STORE SALES	STORE COST	ddd	STORE SALES	STORE COST	ddd
Bulk Mail	2,564.84	1,033.39	12,396.8	20,649.50	8,252.66	12,396.8	5,492.64	2,213.21	3,279.4
Cash Register	2,686.54	1,075.26	12,989.1	21,657.20	8,668.06	12,989.1	5,940.32	2,379.26	3,561.1
Daily Paper	3,052.59	1,240.32	15,357.9	25,525.33	10,167.39	15,357.9	6,725.02	2,660.06	4,065.0
Daily Paper,	4,168.42	1,676.40	21,449.9	35,766.85	14,316.91	21,449.9	9,953.28	4,005.41	5,947.9
Daily Paper,	3,853.07	1,535.37	18,817.4	31,343.16	12,525.72	18,817.4	8,181.01	3,282.31	4,898.7
In-Store Coupon	3,102.09	1,234.87	14,837.7	24,765.51	9,927.79	14,837.7	6,671.10	2,659.69	4,011.4

Figure 46 - Calculated field

Once SpagoBI shows the related pop-up window, assign a name to the new field and define the calculation formula in the editor. In this process, you can drag and drop the measures directly from the table to get the correct syntax. In Figure 47, for instance, the new *my first CC* field calculates the profit subtracting costs from sales.

MEDIA TYPE	PRODUCT FAMILY								
	Drink			Food			Non-Consumable		
	STORE SALES	STORE COST	ddd	STORE SALES	STORE COST	ddd	STORE SALES	STORE COST	ddd
Bulk Mail	2,564.84	1,033.39	12,396.8	20,649.50	8,252.66	12,396.8	5,492.64	2,213.21	3,279.4
Cash Register	2,686.54	1,075.26	12,989.1	21,657.20	8,668.06	12,989.1	5,940.32	2,379.26	3,561.1
Daily Paper	3,052.59	1,240.32	15,357.9	25,525.33	10,167.39	15,357.9	6,725.02	2,660.06	4,065.0
Daily Paper,	4,168.42	1,676.40	21,449.9	35,766.85	14,316.91	21,449.9	9,953.28	4,005.41	5,947.9
Daily Paper,	3,853.07	1,535.37	18,817.4	31,343.16	12,525.72	18,817.4	8,181.01	3,282.31	4,898.7
In-Store Coupon	3,102.09	1,234.87	14,837.7	24,765.51	9,927.79	14,837.7	6,671.10	2,659.69	4,011.4
No Media	104,221.19	41,655.38	520,987.7	868,814.43	347,826.73	520,987.7	229,779.32	91,947.70	137,831.6
Product	3,372.51	1,332.87	16,268.3	27,091.52	10,823.27	16,268.3	7,362.42	2,956.85	4,405.6
Radio	3,005.86	1,201.06	15,412.0	25,749.66	10,337.68	15,412.0	7,116.56	2,840.22	4,276.3
Street Handout	3,139.51	1,240.06	16,719.3	27,885.75	11,166.46	16,719.3	6,994.58	2,806.7	4,011.4

Figure 47- Profit

Click OK and the Pivot table will be automatically updated.

MEDIA TYPE	PRODUCT FAMILY								
	Drink			Food			Non-Consumable		
	STORE SALES	STORE COST	My First CC	STORE SALES	STORE COST	My First CC	STORE SALES	STORE COST	My First CC
Bulk Mail	2,564.84	1,033.39	1,531.5	20,649.50	8,252.66	12,396.8	5,492.64	2,213.21	3,279.4
Cash Register	2,686.54	1,075.26	1,611.3	21,657.20	8,668.06	12,989.1	5,940.32	2,379.26	3,561.1
Daily Paper	3,052.59	1,240.32	1,812.3	25,525.33	10,167.39	15,357.9	6,725.02	2,660.06	4,065.0
Daily Paper,	4,168.42	1,676.40	2,492.0	35,766.85	14,316.91	21,449.9	9,953.28	4,005.41	5,947.9
Daily Paper,	3,853.07	1,535.37	2,317.7	31,343.16	12,525.72	18,817.4	8,181.01	3,282.31	4,898.7
In-Store Coupon	3,102.09	1,234.87	1,867.2	24,765.51	9,927.79	14,837.7	6,671.10	2,659.69	4,011.4
No Media	104,221.19	41,655.38	62,565.8	868,814.43	347,826.73	520,987.7	229,779.32	91,947.70	137,831.6
Product	3,372.51	1,332.87	2,039.6	27,091.52	10,823.27	16,268.3	7,362.42	2,956.85	4,405.6
Radio	3,005.86	1,201.06	1,804.8	25,749.66	10,337.68	15,412.0	7,116.56	2,840.22	4,276.3

Figure 48 - Pivot table and calculated field

Static filters

The static filter is a specific sort of filter that can be directly configured on the field to be filtered. Static filters include global filters and local filters: global filters apply on all sheets of the Worksheet, while local filters only apply on the current sheet.

Double click the Attribute field, which can be located among the *selected fields* or it may already have been dragged and dropped into the widget. SpagoBI shows a pop-up window in which you can choose the options to be visualized.

The Figure below shows a global filter on the *Product Family*, which only considers the *FOOD* option.

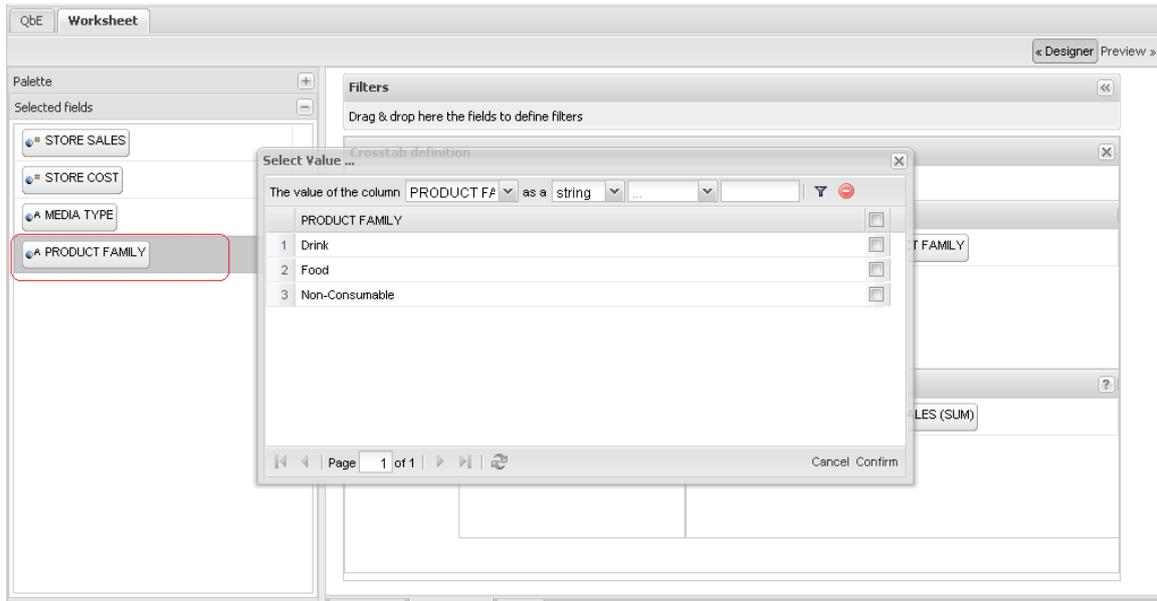


Figure 49 - Global filter

In the preview area, you can see a chart including only one bar displaying the FOOD value, and a pivot table that applies the same filter.

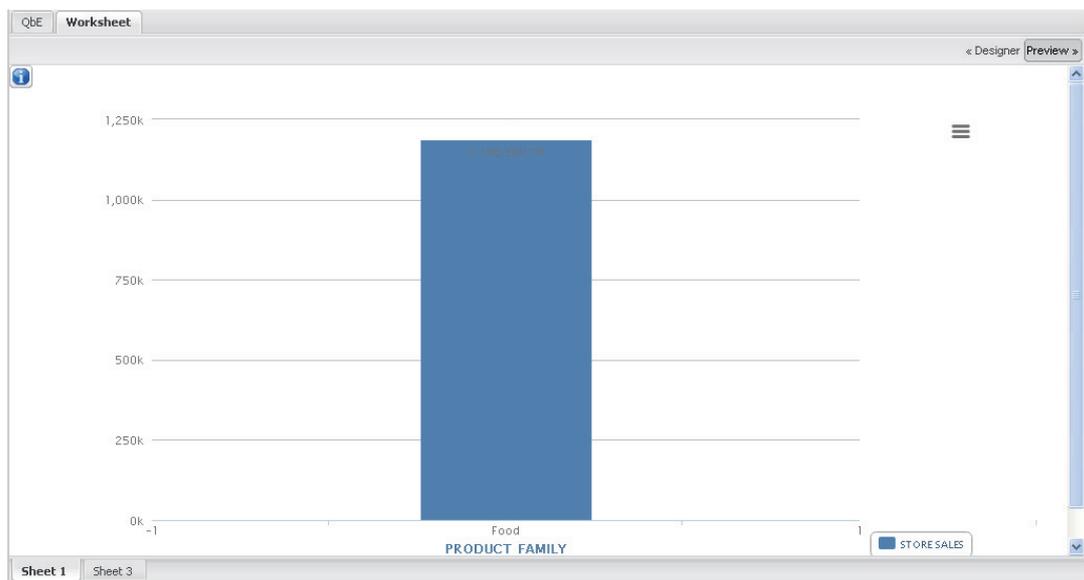


Figure 50 - Global filter

The  icon located at the left side of the window indicates that a filter is being applied. This icon opens a pop-up window as in Figure 50.

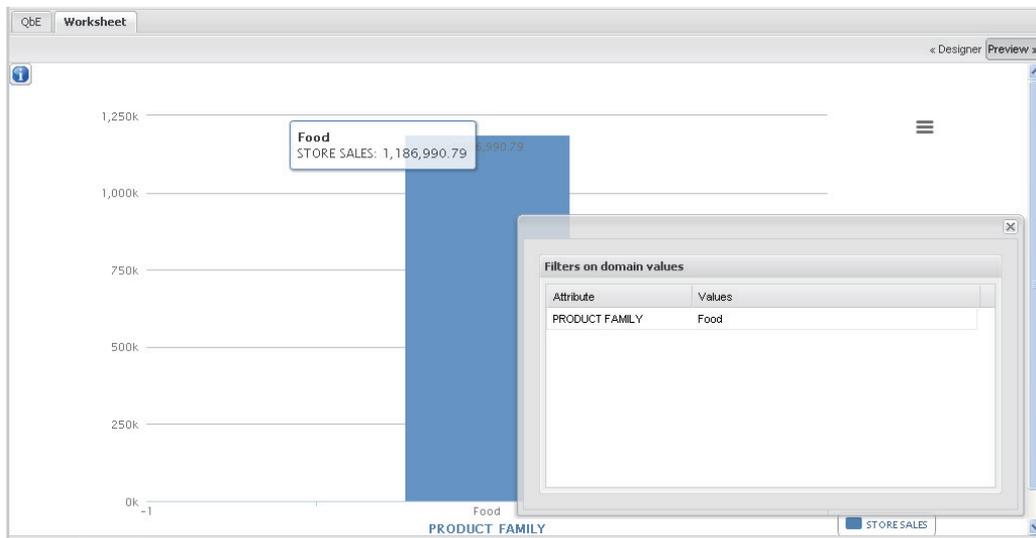


Figure 51 - Pop up window of a filter



This filter can be modified by users that simply have execute permissions, not necessarily modify permissions.

Dynamic filter

Another type of filter can be applied on a Worksheet: the dynamic filter.

It allows end-users to choose the option that suits their needs, once the analytical document has been executed.

To apply this filter, drag and drop the field into the *Filters* area in the Worksheet.

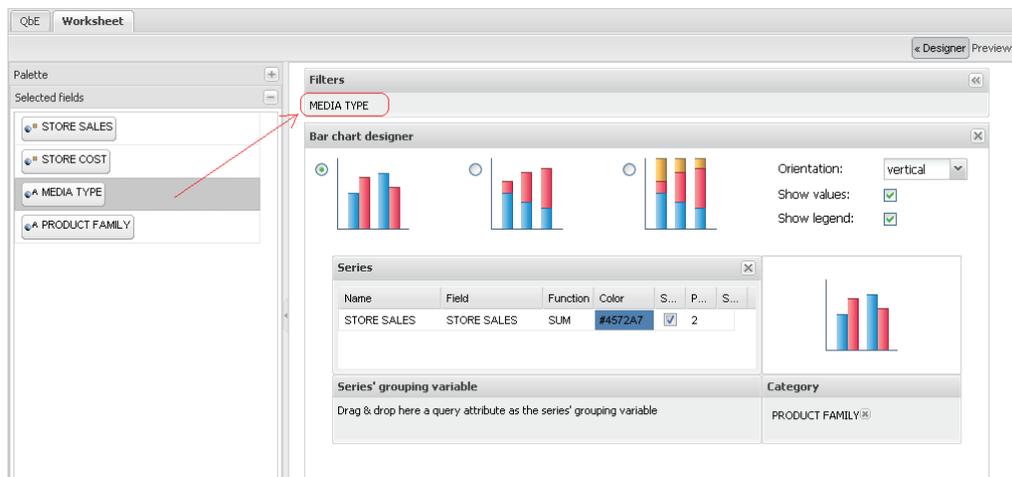


Figure 52 - Dynamic filter

The dynamic filter allows users to set some specific options including:

- Selection: the available options include single value or multi value
- Mandatory: only if this option is set, will SpagoBI show the results of the widget
- Splitting: during the export phase, a pivot table is exported for each value of the filter.

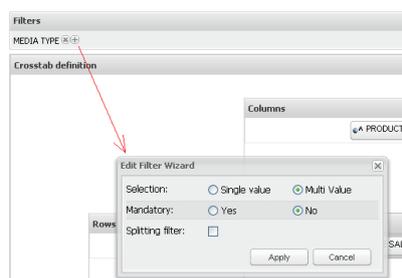


Figure 53 - Dynamic filter options

Click on Preview. The result is shown in Figure 54.

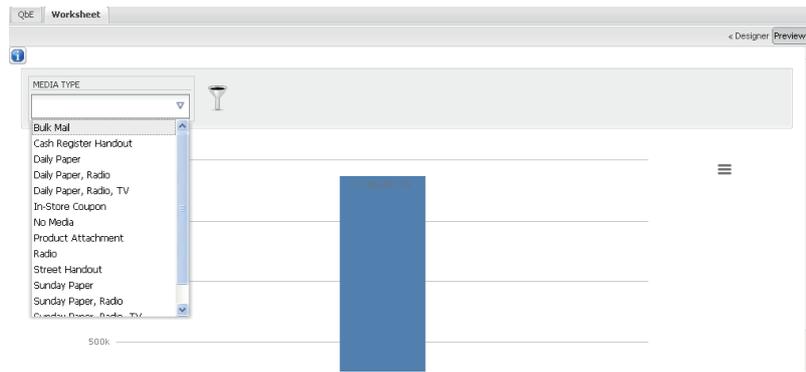


Figure 54 - Preview of a histogram with dynamic filter

Once achieved the desired results by developing the various widgets of the Worksheet, the analysis can be saved by clicking the  icon located at the top right of the page:

Figure 55 - Worksheet saving form

Fill in the fields shown in Figure 55. A document will be created, which can be executed directly using the document browser.

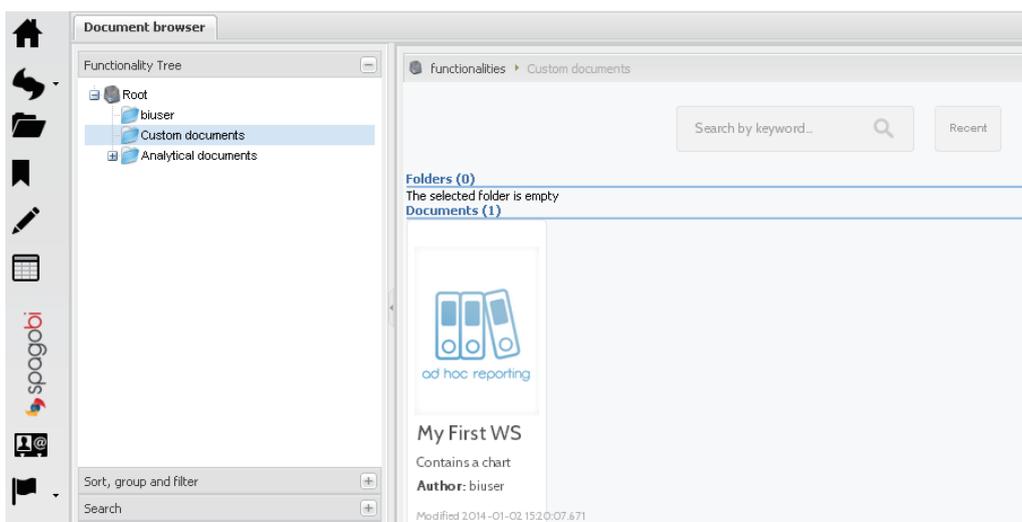


Figure 56 - Create a new document

COCKPIT

SpagoBI allows users to self-build interactive cockpits through an intuitive and interactive interface, also on in-memory technologies, in order to enable faster data insights. Moreover, SpagoBI supports data mash-up, to integrate enterprise data and externally sourced data.

Cockpit documents can be created both by technical users and end users. Cockpits are part of SpagoBI ad-hoc reporting system.

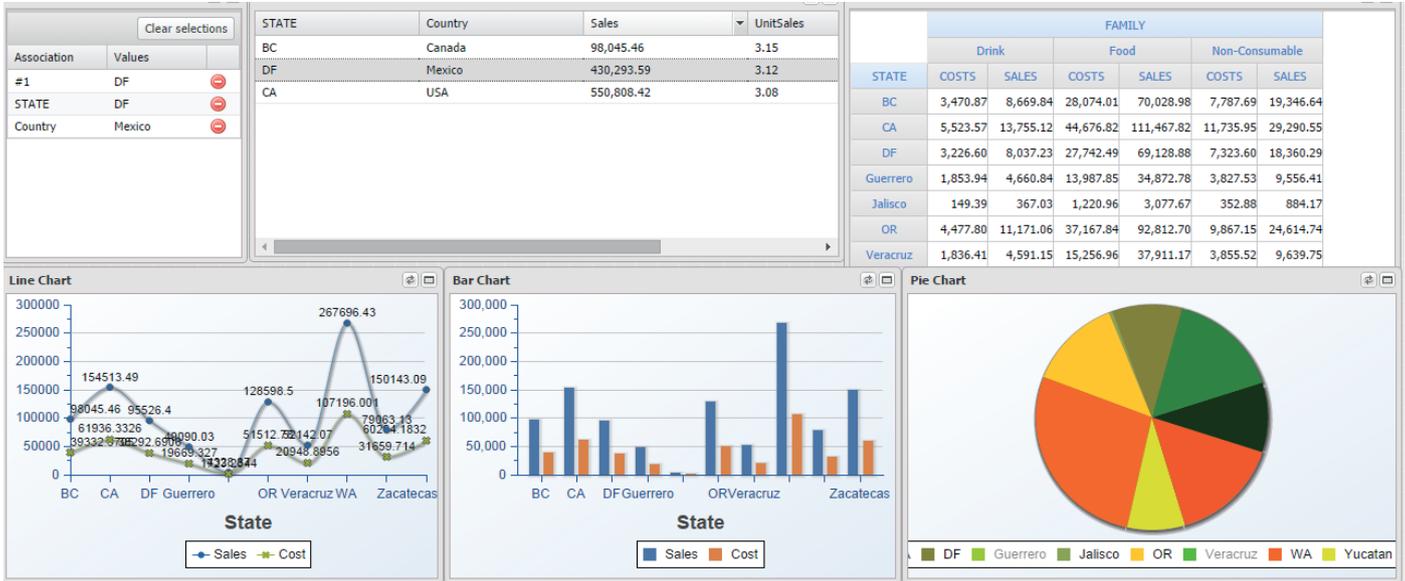
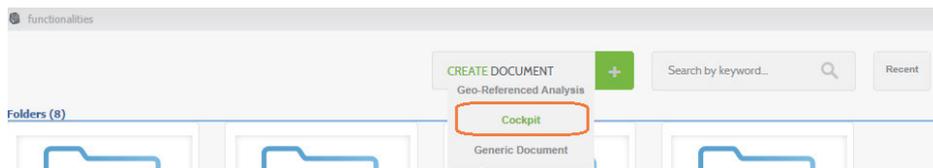


Figure 57 Cockpit

Cockpit engine designer is able from:

- **Technical menu** directly into the Document Browser for the technical users



- **My Analysis menu** for the final users through the ad hoc reporting wizard



Let's see how to build a dashboard and how is presented the interface within the server

Once opened, the Cockpit interface is an empty page with a tool bar containing different options.

-  – open the window to add single widgets and manage them
-  – open the window to manage associations between data sets
-  – selections / clear selections
-  – open the window to save the cockpit document into the personal folder or My Analysis section.

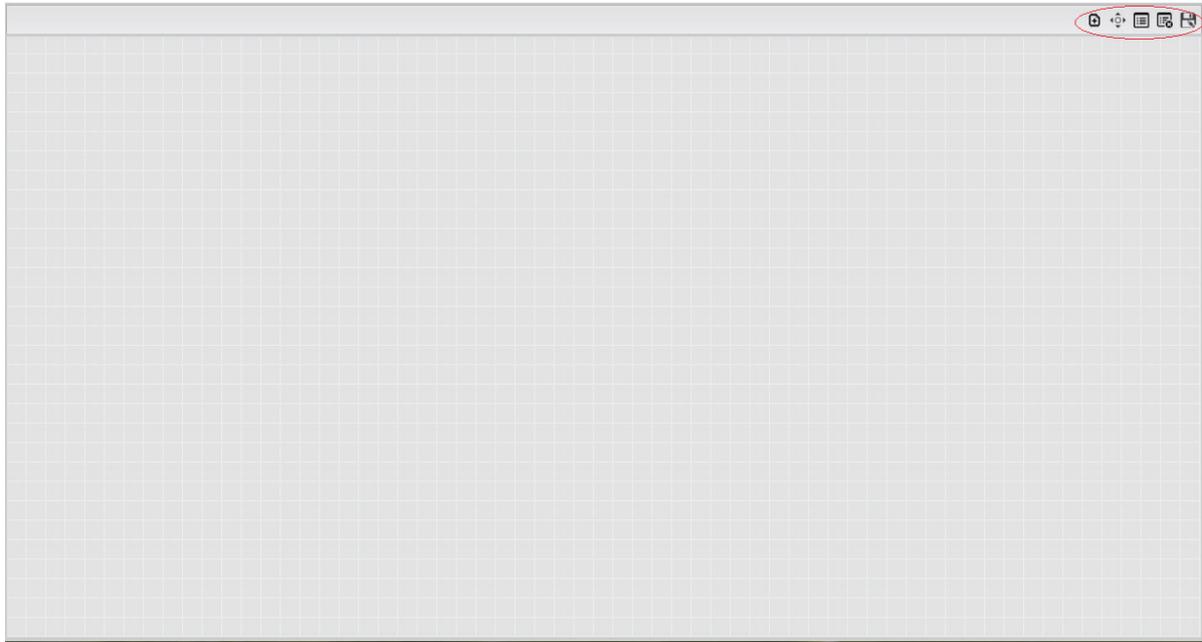


Figure 58 Cockpit

By clicking on the *add widget* button a configurable window opens.

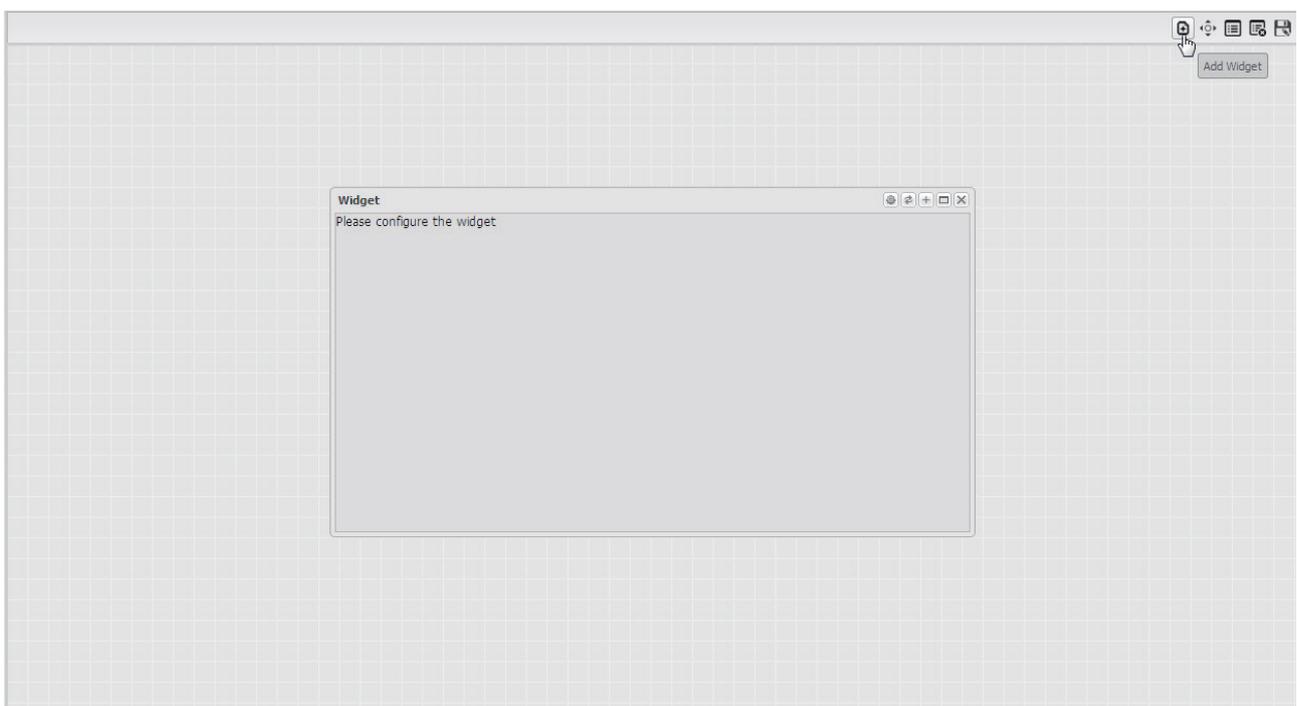


Figure 59 Cockpit – New Widget

Design Widgets

Let's see how you can design the different widgets once you click the *edit* button . As a first step, choose the dataset. The wizard window shows all datasets available according to the logged-in user role.

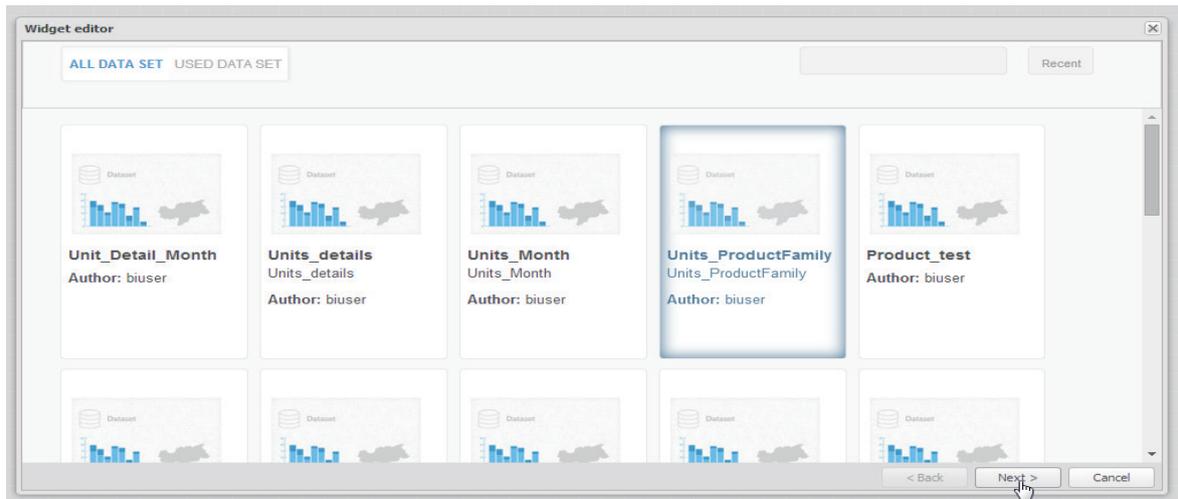


Figure 60 Cockpit – Dataset selection

As a next step, choose the widget you want to use. The wizard window offers a tool bar with all available widgets that you can configure.

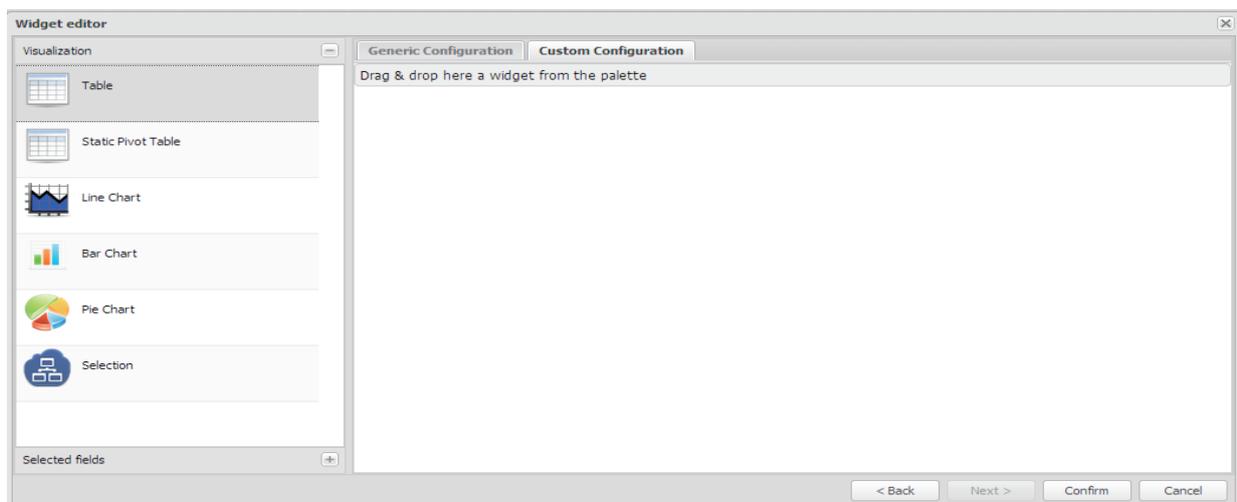


Figure 61 Cockpit – Data Visualization

Widgets include:

-  Flat Table
-  Static Pivot Table
-  Line Chart
-  Bar Chart
-  Pie Chart
-  Selection

For all different data visualization details please refer to the Worksheet paragraph.

The number of boxes that you can insert into the cockpit varies according to your needs, but the procedure remains the same. Add a new widget

- Choose the dataset
- Choose how to display your data
- Place your object within the available space

During the second step, multiple datasets can be selected. The dataset already used will be orange-bordered, while the other one(s) will be blue-bordered.

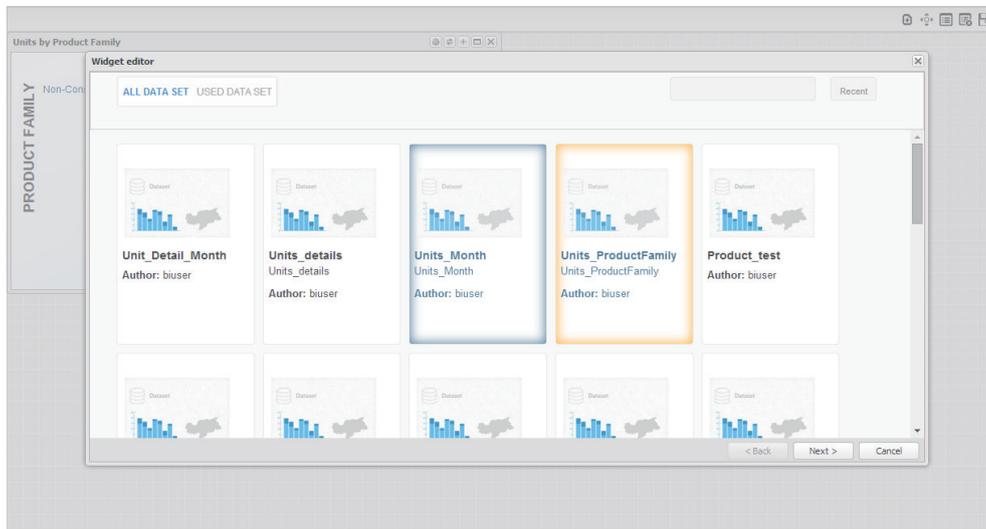


Figure 62 Cockpit – Multiple dataset selection

Associations

If your goal is to show data from a single dataset, it's not necessary to define any association. Associations should be set within the designer when widgets are built on different data sets. To set associations, click the dedicated button.

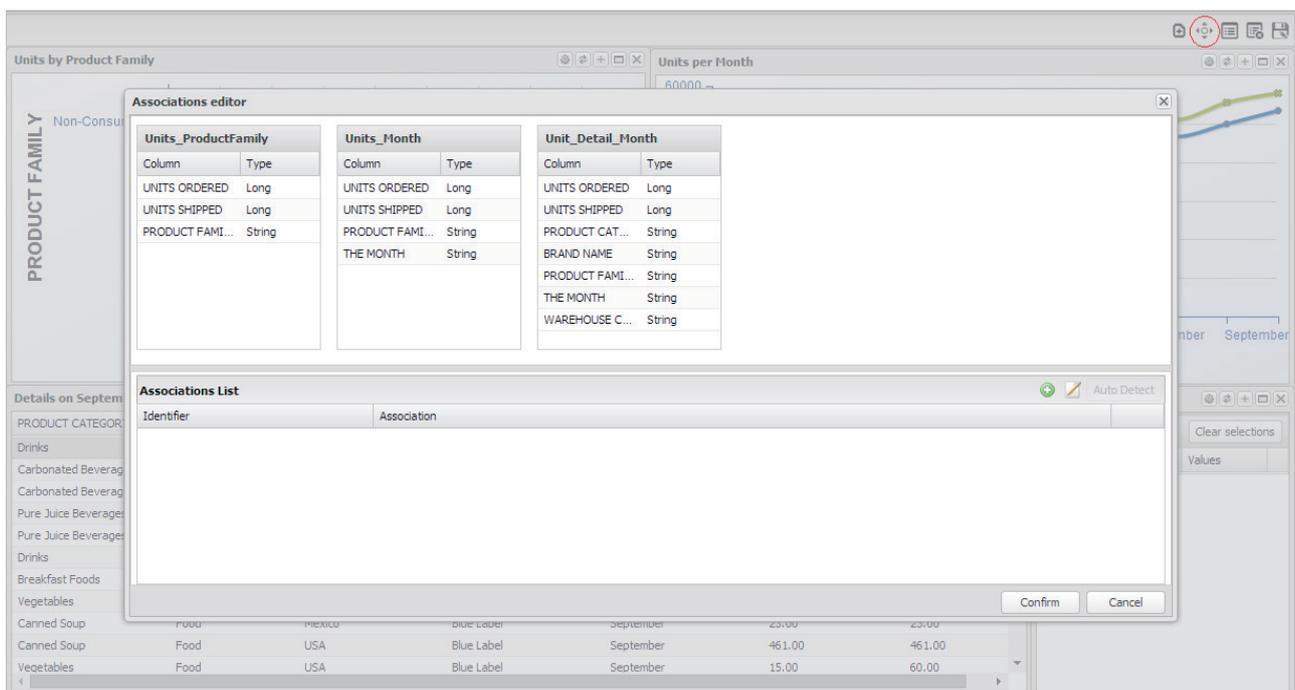


Figure 63 Cockpit –Association editor

The editor shows one column for each data set. To create associations, select the columns to be correlated and click the [icon] button to make them effective.

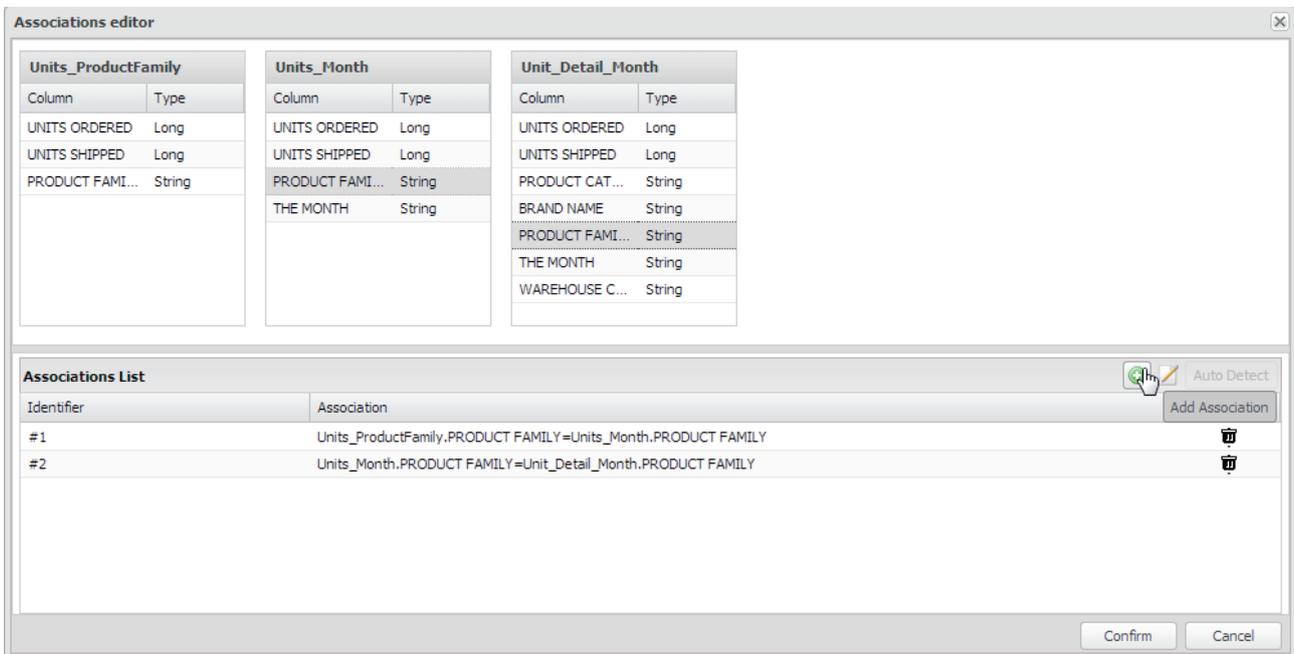


Figure 64 Cockpit – Add Association

Once you have your desired result you can save the cockpit by clicking on the save button  which opens a dialog in which to set some parameters to respectively save within the personal folder (technical users) or in the *My analysis* section.

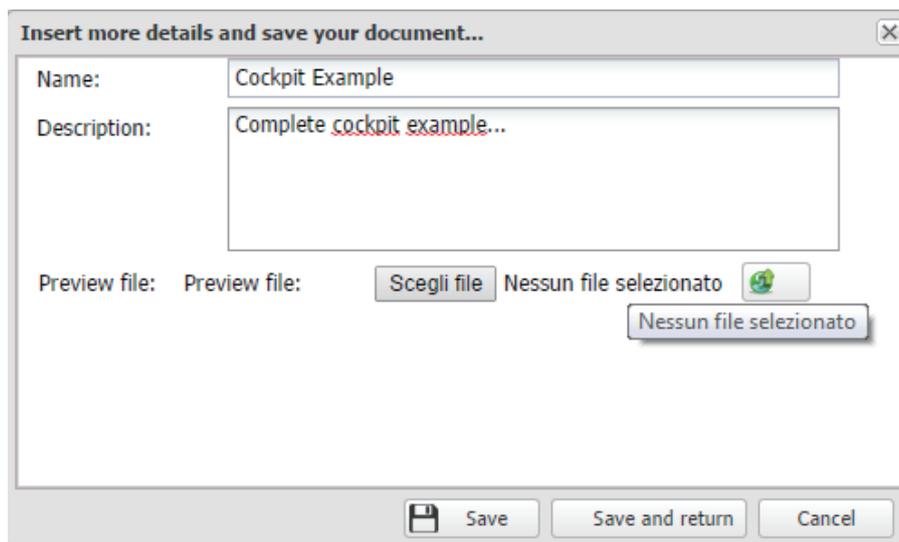


Figure 65 Cockpit – Save cockpit

Note - It's possible to share the new cockpit with other users with the apposite icon



Figure 66 Cockpit – Share cockpit

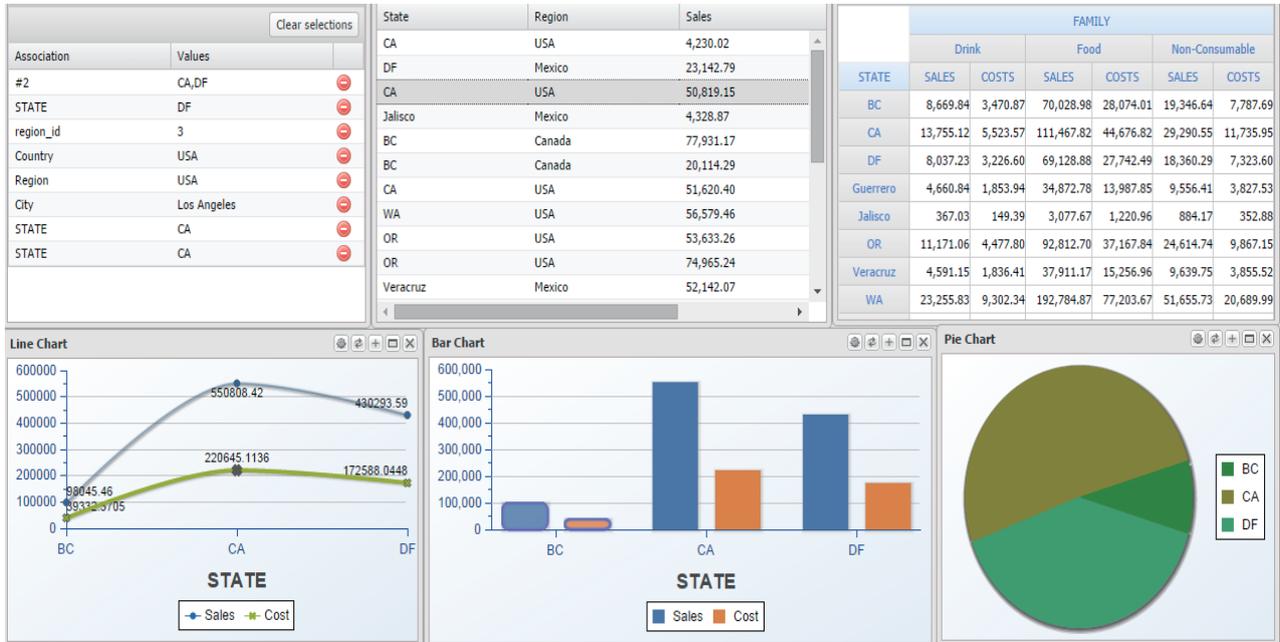


Figure 67 Cockpit – Final Result

QBE

The QbE (Query-by-Example) feature is suitable for those cases in which free inquiry and extraction of data is more relevant than data graphical representation and layout. Users can define their own query graphically, execute it, check the results, export them, save the query for future use and generate a reporting template.

The QbE interface includes a *Schema* area showing the data model, as well as an *Editor* area in which the user can build the query (see Figure 68).

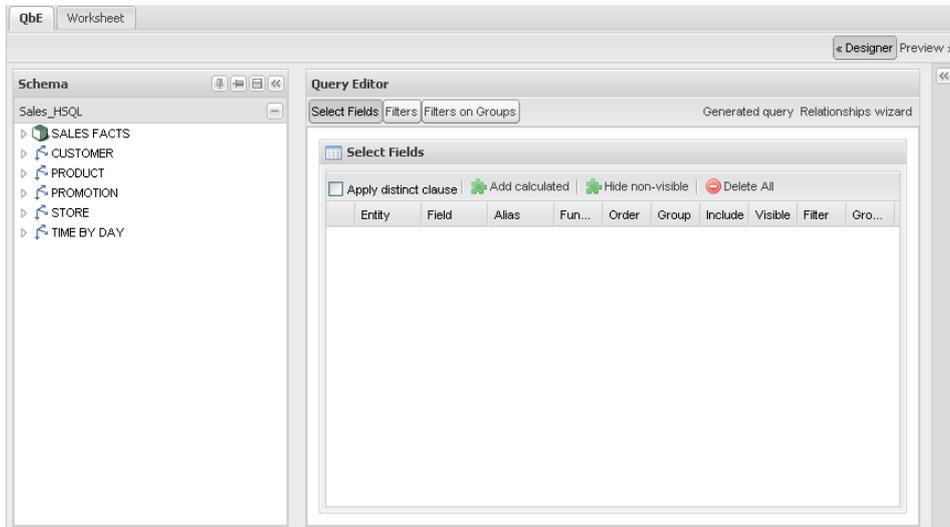


Figure 68 - QbE interface

The data model represents a business model with different levels of complexity, namely a high-level representation of the data included in the data base. More than this, it's a set of tables and entities that can be used to create one's own queries.

Each single entity is composed of a title, some attributes and relationships with other entities:

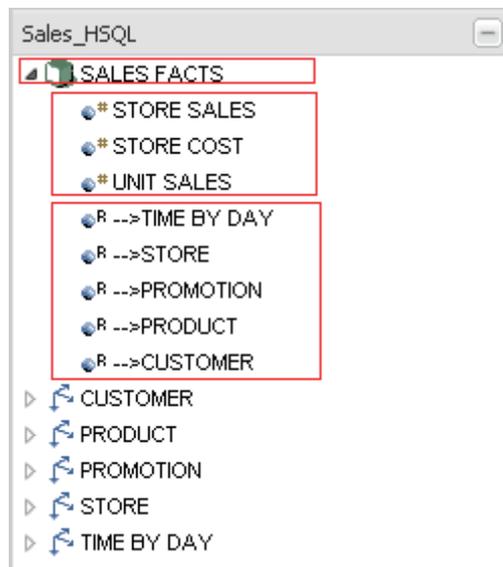


Figure 69 - Data model

By exploring the content of an entity (i.e. *SALES FACT* as in Figure 69), you encounter the following elements:

- #: **Misure**. Attribute made of numeric data and additional data (e.g. *sold items*)
- ^A: **Attribute**. It refers to fields that can be associated to a category (e.g. *product category*)
- ^R: **Relation**. It refers to relationships or connections between two entities (e.g. *product sales*)

The main section offers a query editor, including 3 sub-sections, which allows users to visually compose their query.

Select Fields

It includes the list of columns to be returned by the query. To add a new attribute in this section, just drag and drop a field into the item tree or click it.

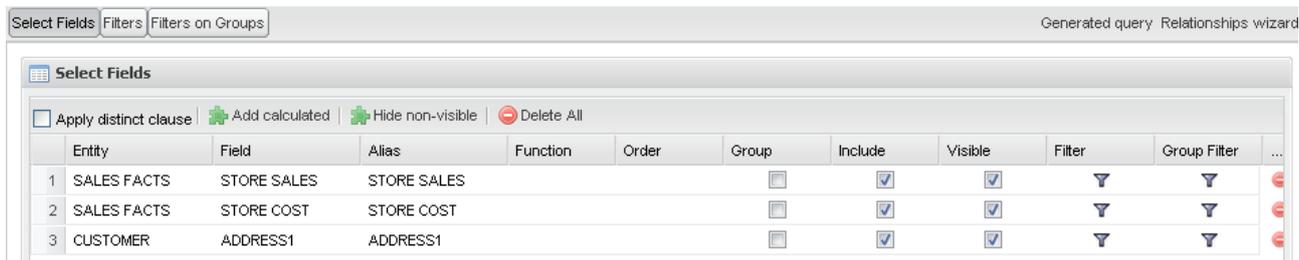


Figure 70 - Select fields

For each dropped item, it is possible to:

- Define an alias
- Apply a mathematical function, in case of aggregation
- Set a sort criteria
- Group
- Indicate the column(s) to be included in the result
- Indicate whether a column shall be visible in the result
- Delete a column
- Add a filter criteria
- Add a filter on groups.

Filters

In this section, a filter criteria may be added. Similarly, to add an element in this section, drag and drop it and then click on the data model.

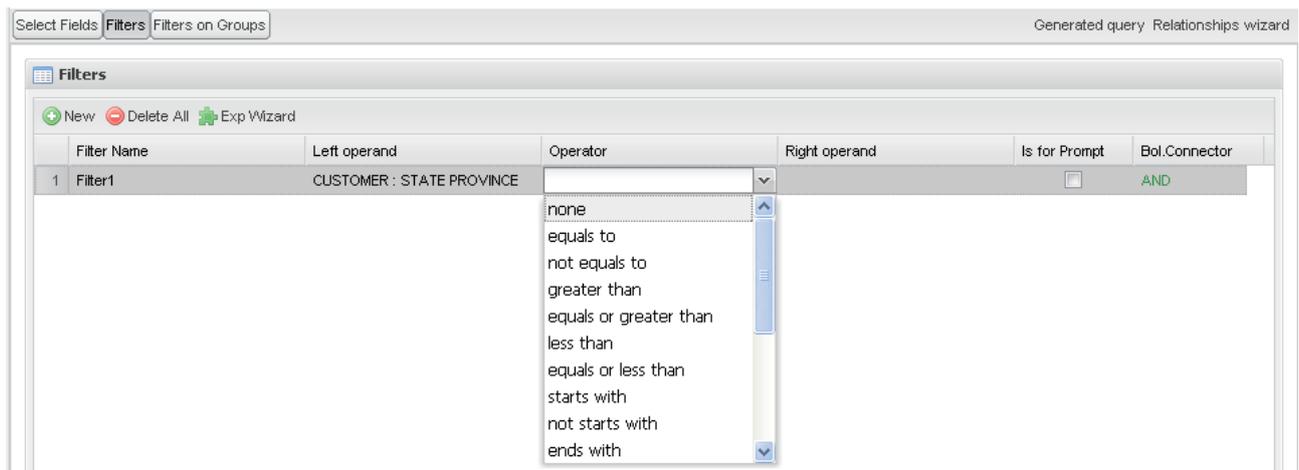


Figure 71 - Filters

For each dropped element, it is possible to:

- Specify a filter operator
- Define the right operand using a fix value or choosing one of the options included in the lookup (see Figure 72)
- Define the right operand using a model attribute
- Specify whether the filter value shall be requested during the query execution (it is for Prompt).

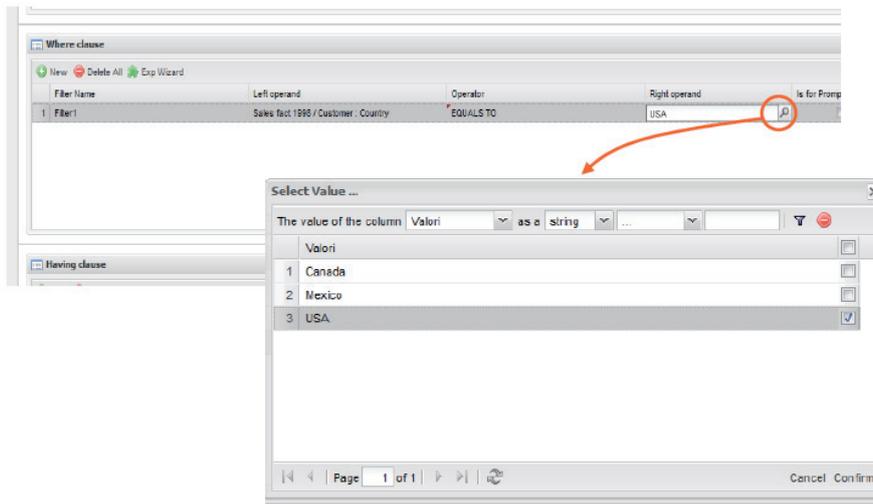


Figure 72 - Filter lookup

The types of filters that can be applied to a QbE query are listed in the following table.

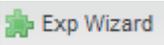
Filter type	Left operand	Operator	Right operand	Example
Basic	Entity.attribute	Any	value	Product.family = "Food"
Basic	Entity.attribute	Any	Entity.attribute	Sales 1998.Store sales > Sales 1998.Store cost
Parametric	Entity.attribute	Any	[parameter]	Product.family = [Product family]
Dynamic	Entity.attribute	Any	prompt	Product.family = ?
Explicit join	Entity1.attribute	Any	Entity2.attribute	Sales 1997.product name = Sales 1998. product name
Value list from subquery	Entity.attribute	In /not in	subquery	Sales 1998.customer in subquery
Single value from subquery	subquery	< = >	value	Subquery > 0

Therefore the *Filters* section enables the definition of filters with multiple relationships using the AND and OR operators.

Moreover SpagoBI offers a tool for building complex expressions: Figure 73 shows a list of filters, according to which the client's city is equal to Beverly Hills, the country is equal to USA, the store type is Deluxe Supermarket and the year is greater than 2011.



Figure 73 - Multiple filters

By clicking the  icon, a pop-up window allows users to build expressions to execute filters.

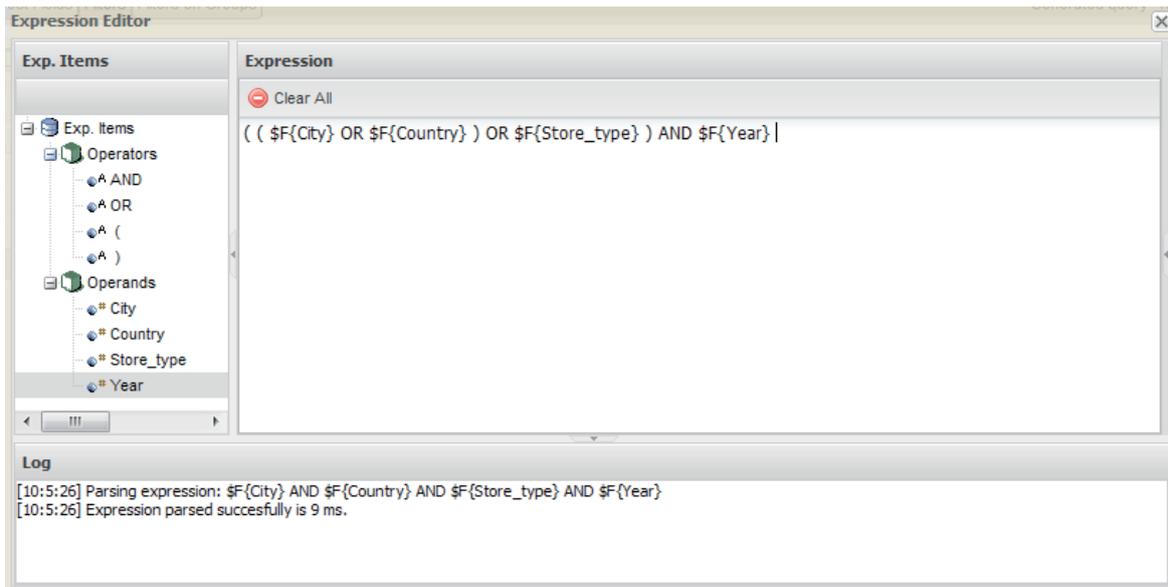


Figure 74 - Expression Wizard

The expression wizard is built through the *Exp.Items* section that includes the Operators list (AND and OR filter relationships and brackets defining the execution order), and the list of available filters in the Operands section. The “expression” section shows the expression for execution of filters.

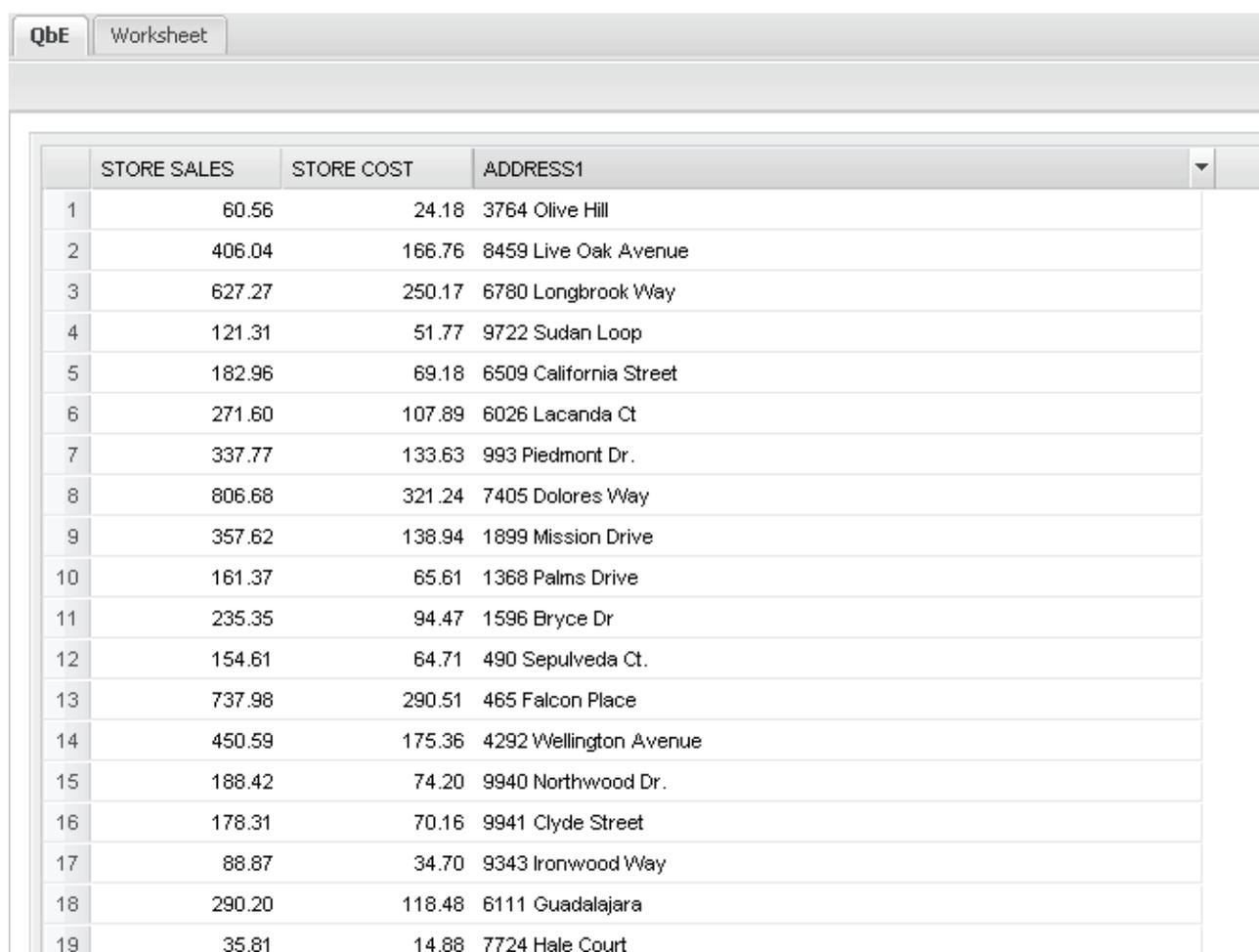
To test the expression, use the Log box as shown in Figure 74.

Once you close the window and execute the query, the content of the expression will be considered, if included.

The Filters on Groups feature corresponds to a condition of HAVING clauses of a SQL query. It works like a normal filter, with some additional columns for grouping functions. Once the user has created the **HAVING** filter:

- in the two **Function** columns, it is possible to define the aggregation function to be used on right/left operands;
- set the **Left/Right Operand Type** columns, to define operand types;
- if the filter is assigned a dynamic value, the **Is for Prompt** column should be flagged. This way users can choose the value of the operand at query execution.

Once the query has been defined, it can be executed using the **Preview** button.



	STORE SALES	STORE COST	ADDRESS1
1	60.56	24.18	3764 Olive Hill
2	406.04	166.76	8459 Live Oak Avenue
3	627.27	250.17	6780 Longbrook Way
4	121.31	51.77	9722 Sudan Loop
5	182.96	69.18	6509 California Street
6	271.60	107.89	6026 Lacanda Ct
7	337.77	133.63	993 Piedmont Dr.
8	806.68	321.24	7405 Dolores Way
9	357.62	138.94	1899 Mission Drive
10	161.37	65.61	1368 Palms Drive
11	235.35	94.47	1596 Bryce Dr
12	154.61	64.71	490 Sepulveda Ct.
13	737.98	290.51	465 Falcon Place
14	450.59	175.36	4292 Wellington Avenue
15	188.42	74.20	9940 Northwood Dr.
16	178.31	70.16	9941 Clyde Street
17	88.87	34.70	9343 Ironwood Way
18	290.20	118.48	6111 Guadalajara
19	35.81	14.88	7724 Hale Court

Figure 75 - QbE preview

Similarly to the Worksheet, in order to get back and continue building and designing the query, click the **Designer** button at the top right of the page.

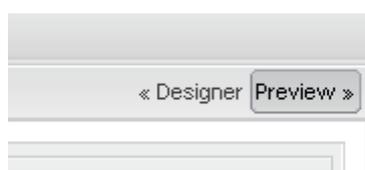


Figure 76 - QbE designer

The QbE designer offers a section dedicated to the query catalogue (see Figure 77 on the left).

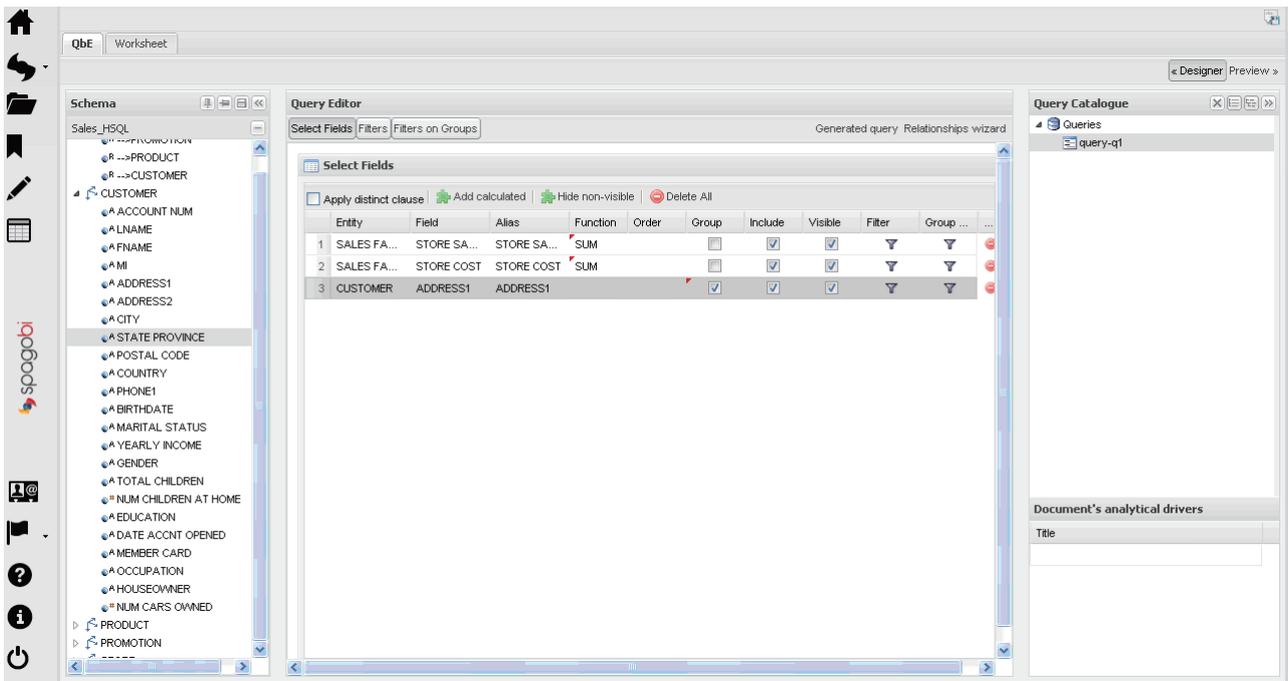


Figure 77 - Query catalogue

The query catalogue allows the definition of sub-queries in a filter whose *right operand* is the query itself: in this case, the clause to be used is IN or NOT IN.
 If the QbE analytical document has one or more analytical drivers (parameters), they will be listed here.

Calculated fields

End-users can create new calculated fields either inside a query or inside a data model. For instance, let's see how you can create a calculated field that shows the difference between Store Sales and Store Cost.

Right click on the menu of the chosen entity and select *Add calculated field* :

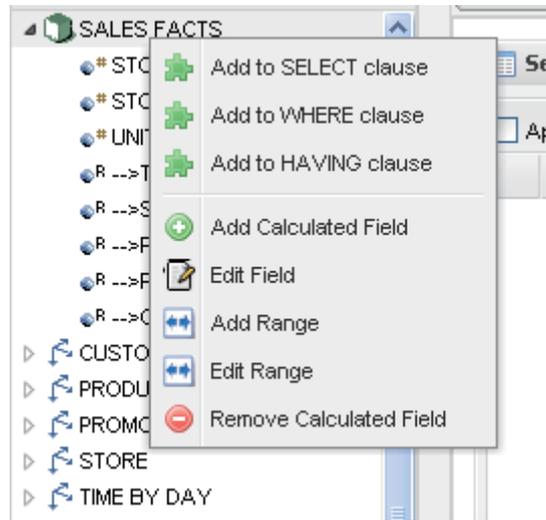


Figure 78 - Entity contextual menu

The new window offers an editor in which you can define the calculated field.

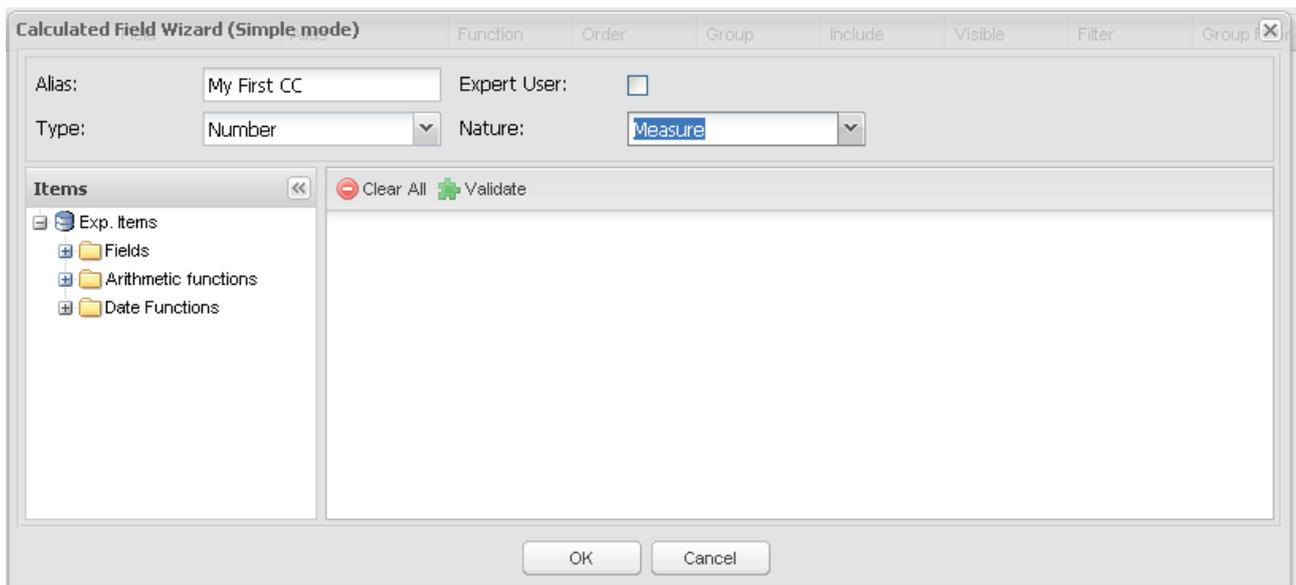


Figure 79 - Calculated field

To build a calculated field, you shall define:

- **Name**
- **Type** (string, number, date)
- **Nature** (measure or attribute)
- **Formula**: click on the fields included in the item tree on the left (or drag and drop them) and build the formula, as shown in Figure 80.

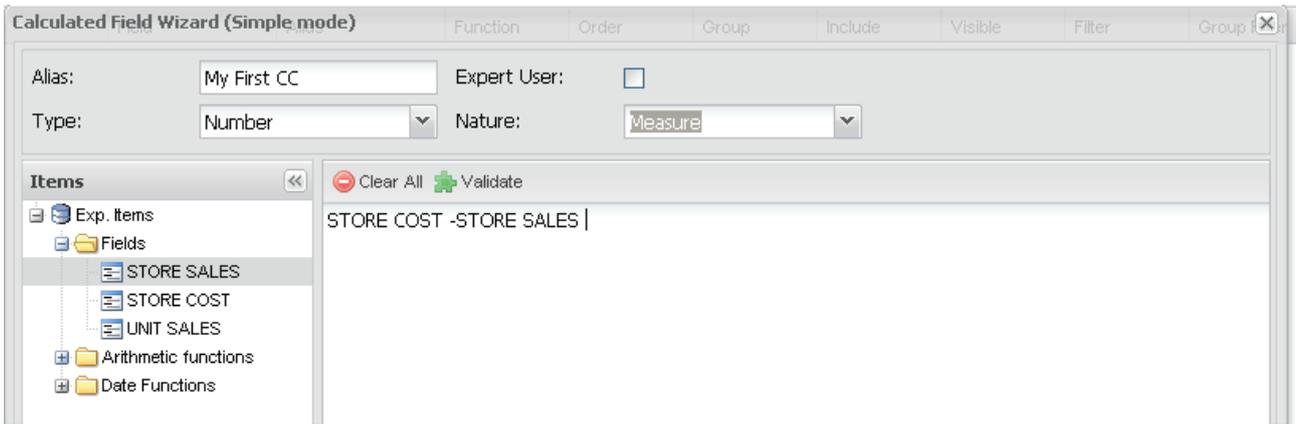


Figure 80 - Calculated field

This field appears in the data model of the QbE main page. This new item can be used as a model attribute.

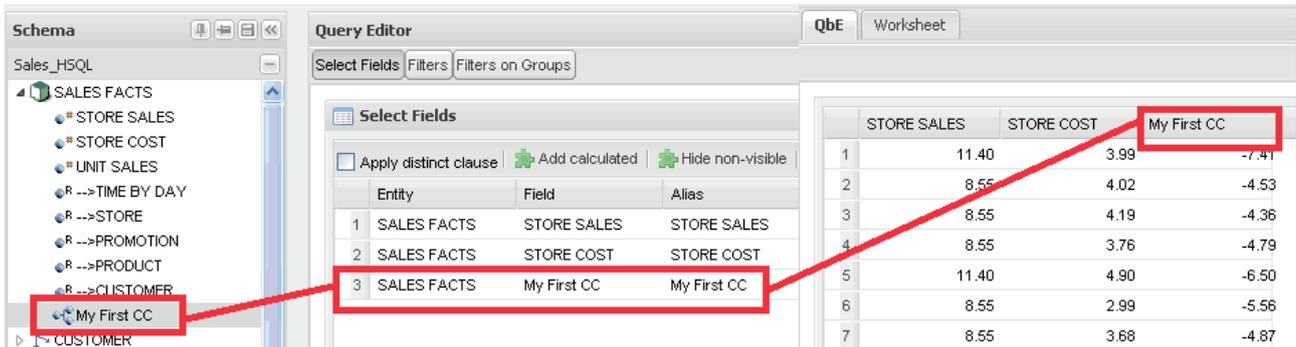


Figure 81 - Data model and preview with a calculated field

Once a new calculated field is created in a query, it can be edited by simply double-clicking it in the "select fields" section.

The following table includes some functions that can be used to build an advanced calculated field.

Function	Description	Notes
<i>MOD(num1, num2)</i>	Given two numbers a and b, where b ≠ 0, a mod. b returns the remainder after division of a first item by the second.	
<i>BIT_LENGTH(str)</i>	It returns the length of a bit string	
<i>UPPER(str)</i>	It converts a string to uppercase	upper(Store type)
<i>LOWER(str)</i>	It converts a string to lowercase	
<i>SUBSTRING(str, idx, length)</i>	It returns the substring of str string with the given length to the idx position.	substring(Sales district,2,4)
<i>CONCAT(str1, str2)</i>	It joins two strings	
<i>LENGTH(str)</i>	It returns the length of a string	length(Sales district)
<i>LENGTH(str, s, idx)</i>	It returns the position in which the S character is located, starting the search at a specific idx point of the string	
<i>TRIM([type] str)</i>	TRIM([BOTH LEADING TRAILING] [remstr] FROM] str) It eliminates spaces in a string	
<i>CURRENT_DATE()</i>	It returns the current date	
<i>CURRENT_TIME()</i>	It returns the current time	
<i>CURRENT_TIMESTAMP()</i>	It returns current date and time	
<i>HOUR(date), MINUTE(-date), SECOND(date)</i>	It returns time, minutes and seconds	minute(current_timestamp())
<i>YEAR(date), MONTH(date), DAY(date)</i>	It returns year, month, day	
<i>CASE</i>	case when COUNTRY = 'USA' then '1' else '0' end	
<i>CASE</i>	case when Unit sales >= 10 then 'Too Many' when (Unit sales < 10 and Unit sales >=5) then 'Many' when (Unit sales < 5 and Unit sales >1) then 'Some' else 'Zero' end	

Bands

The attributes of an entity relating to a specific model may have different values. In order to analyze data, it is worth grouping these values into categories. For instance, let's consider customers' age: often the analysis does not aim at knowing the specific customers' age but the age band to which customers belong to (e.g. young, adults, elders). For this reason, the QbE engine can define and manage intervals in queries.

To create a new band, click **Add interval** in the contextual menu: the guided procedure for creating a new band will open. As shown in Figure 82, it is possible to select numeric fields and data functions to define intervals.

For example, the **AA_up_today** function returns the interval between a certain data and today's date, which is useful to identify the customers' age. The procedure to define a band follows.

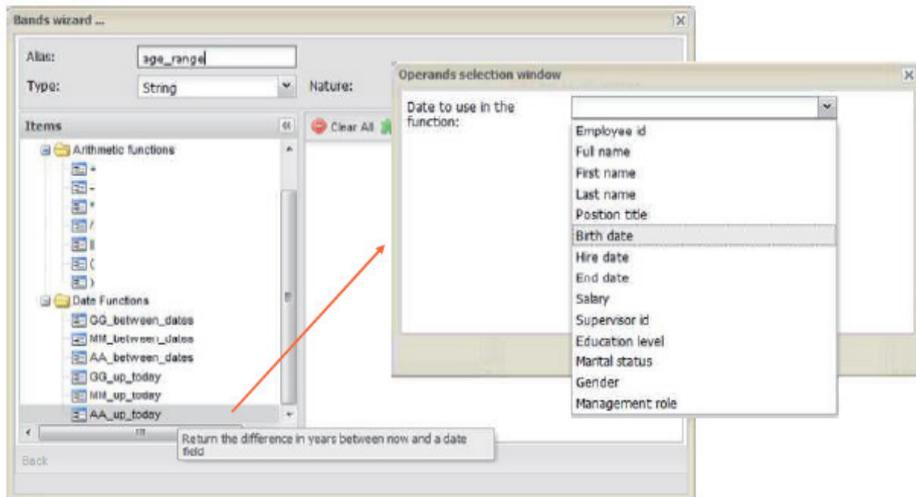


Figure 82 - Bands

Click the **Next** button at the bottom right corner of the guided procedure and a new window will appear to create new instances of the bands.

Click **Add Band** to add a new instance, set values and related labels. Click **Add Default** to include a default interval: this will create a new category called **Other**, which groups all the values that do not belong to the defined intervals. Click **Finish**. The band will appear as a node in the **Schema** panel. If you wish to modify the interval, click **Edit Range**.

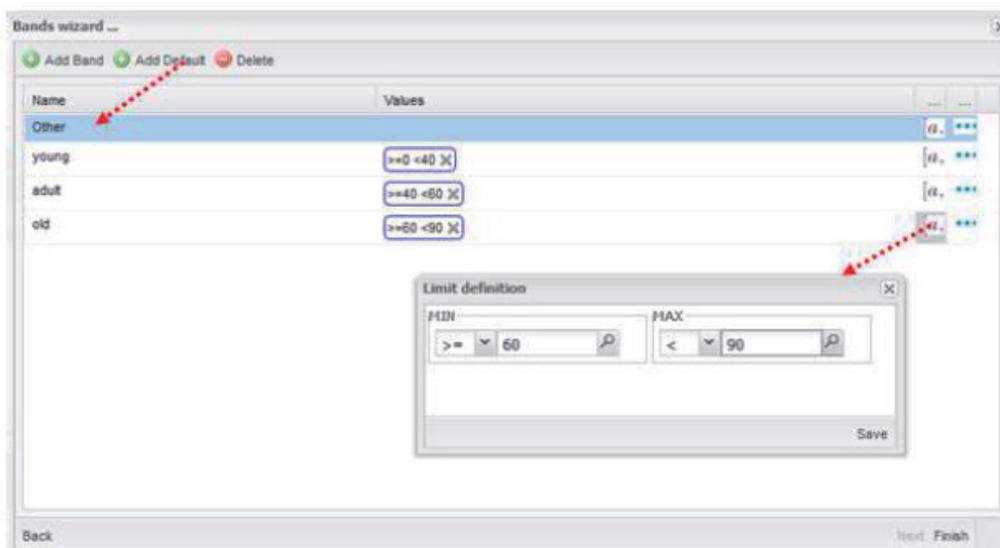


Figure 83 - Interval definition

Join-type relationships

The QbE includes a specific feature to thoroughly manage relationships among entities: users can create join paths from one table to another to be used in case of ambiguity.

Let's see in detail how it works through an example.

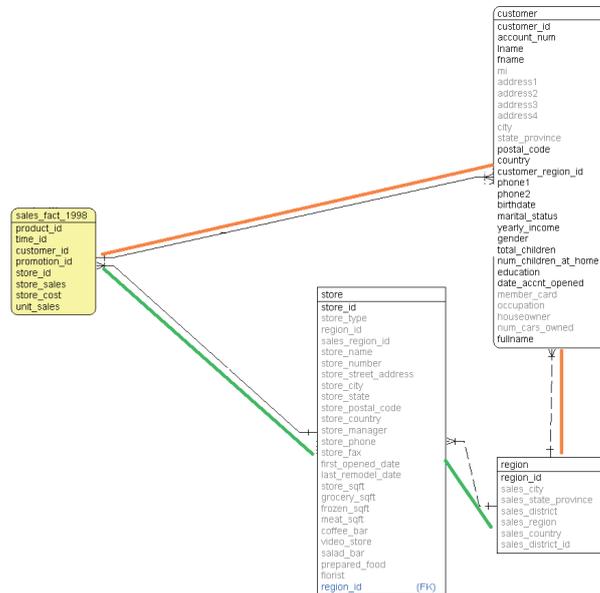


Figure 84 - Ambiguity

Using the schema and data model as in Figure 84, let's build a model with the following relationships:

- Store - Region entity
- Customer - Region entity
- Sales Fact - Store entity
- Sales Fact - Customer entity.

Ambiguity arises when attributes coming from the various tables are dragged and dropped into the query that is build in the QbE, as in Figure 85.

In this case, in order to identify the items sold by region, the Join relationship may be *Sales Fact – Customer – Region* or *Sales Fact – Store – Region*.

Entity	Field	Alias	Function	Order	Group	Include	Visible	Filter	Group ...	Delete
4	Store	Store name	Store name		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	Sales fact 1998	Unit sales	Unit sales	SUM	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	Customer	Fname	Fname		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	Region	Sales region	Sales region		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 85 - Ambiguity

Click *Relationship Wizard* in the Query editor and a pop-up window will appear, where users can define the path:





These images show the double relationship between *Sales Fact* and *Region*, specifically:

- relationship between Sales Fact and Customer
- relationship between Sales Fact and Store.

At this point, modify the relationship so as to eliminate ambiguity: if you wish to view the region related to a specific customer, select the *Region* entity in the *Entity* panel on the left and the correct path in the panel on the right, as shown here:



Follow the same procedure for fact tables.



If you wish to check the SQL code generated by the QbE query, click the “Generated Query” button (see Figure 86).



Figure 86 - Generated query

```

select
  customer0_.`fname` as `Fname`,
  store1_.`store_name` as `Store_name`,
  region2_.`sales_region` as `Sales_region`,
  sum(sales_fact3_.`unit_sales`) as `Unit_sales`
from
  `foodmart_key`.`customer` customer0_ cross
join
  `foodmart_key`.`store` store1_ cross
join
  `foodmart_key`.`region` region2_ cross
join
  `foodmart_key`.`sales_fact_1998` sales_fact3_
where
  sales_fact3_.`customer_id`=customer0_.`customer_id`
  and customer0_.`customer_region_id`=region2_.`region_id`
  and sales_fact3_.`store_id`=store1_.`store_id`
group by
  customer0_.`fname`,
  store1_.`store_name`,
  region2_.`sales_region`

```

The relationship between Customer and Region is shown in bold. On the other hand, if the query aims to show the region of the Store, it is necessary to modify the wizard as follows:



The generated query follows:

```

select
  customer0_.`fname` as `Fname`,
  store1_.`store_name` as `Store_name`,
  region2_.`sales_region` as `Sales_region`,
  sum(sales_fact3_.`unit_sales`) as `Unit_sales`
from
  `foodmart_key`.`customer` customer0_ cross
join
  `foodmart_key`.`store` store1_ cross
join
  `foodmart_key`.`region` region2_ cross
join
  `foodmart_key`.`sales_fact_1998` sales_fact3_
where
  store1_.`region_id`=region2_.`region_id`
  and sales_fact3_.`store_id`=store1_.`store_id`
  and sales_fact3_.`customer_id`=customer0_.`customer_id`
group by
  customer0_.`fname`,
  store1_.`store_name`,
  region2_.`sales_region`

```

The relationship between Store and Region is shown in bold.

Alias

If the data model includes various relationships between two tables, the QbE allows users to manage them using aliases.

To describe this feature, it is worth using an example. Let's suppose there is a double relationship between *Promotion* and *Time by Day* entities (see Figure 87).

The screenshot shows the QbE interface with a schema tree on the left and a query editor on the right. The schema tree shows the 'Promotion' entity with relationships to 'Time by day' (BR_Promotion_Time by day and t2_fk). The query editor shows a 'Select Fields' table with three rows. Red circles and numbers 1-5 highlight specific elements: 1 and 2 point to relationships in the schema; 3 points to a tooltip for a relationship; 4 and 5 point to the 'Alias' and 'Function' columns in the 'Select Fields' table.

	Entity	Field	Alias	Function
1	Promotion	Promotion name	Promotion name	
2	Time by day	The date	Start date	4
3	Time by day	The date	End Date	5

Figure 87 - Alias

The two relationships concern the start date and end date of the promotion.

As shown in the Figure 87, this information can be retrieved from the QbE graphical interface. The Promotion entity includes two relationships (see points 1 and 2 in the figure), whose tooltip returns information on how the relationship is structured (see point 3).

If you wish to see the list of promotions with a specific start date and end date, it is necessary to drag and drop the name of the promotion (*Promotion* entity) and the *The Date* field (*Time by Day* entity) two times by changing the alias, as well as the name of the column to be visualized in the results of the query (see points 4 and 5).

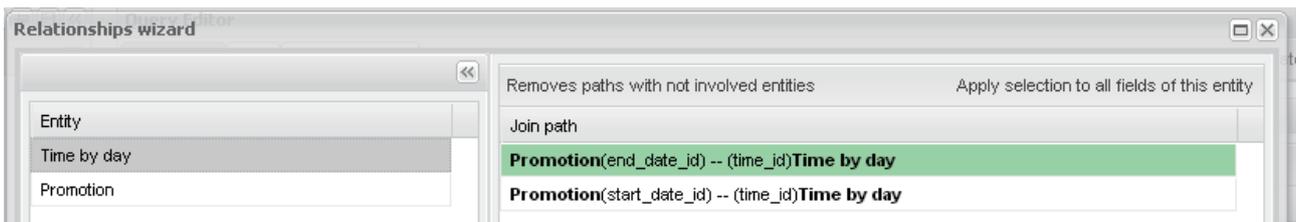
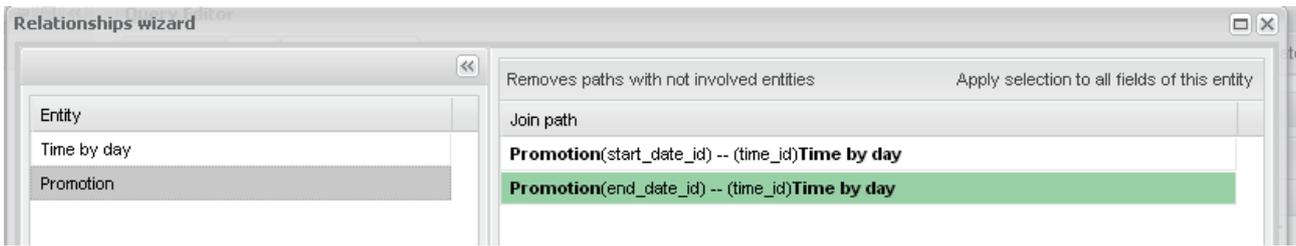
By executing the query, users can investigate the system behaviour: in the absence of specific indications, the system selects two relationships (path) at random.

Query:

```
select
  promotion0_.`promotion_name` as `Promotion_name`,
  time_by_da1_.`the_date` as `Start_date`,
  time_by_da1_.`the_date` as `End_Date`
from
  `foodmart_key`.`promotion` promotion0_cross
join
  `foodmart_key`.`time_by_day` time_by_da1_
where
  promotion0_.`end_date_id`=time_by_da1_.`time_id`
```



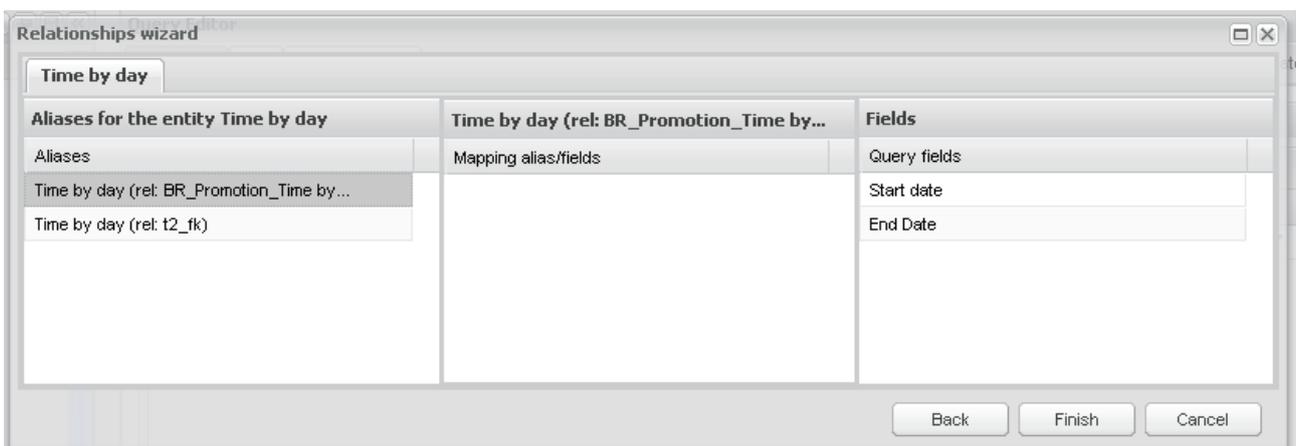
As mentioned in the previous paragraph, by opening the relationships wizard, users can see the list of entities relating to various paths, as well as the list of paths involving various entities:



In this case, we are only using one relationship with the *END_DATE*. If you wish to use both relationships (*end date* and *start date*), select both.



The tooltip shows the complete path using an intuitive tree layout. Once the relationships are selected in both entities, click *Apply*. SpagoBI will show the following window:



It includes 3 sections:

- **List of Aliases** - In this case the Alias contains a different name, but it refers to the same entity (like SQL alias)
- **List of Fields** associated to the entities selected in section 1
- **List of fields** involved in the query.

To distinguish the fields during the execution of the query, it is necessary to identify all the fields involved in the query (included in section 3) with the aliases of the entities that contain them (included in section 1). Select the "Time by day (rel BR_Promotion_...)" entity in the first column, then drag and drop the "Start Date" field from the third column to the one in the middle. Repeat the same operation with the "Time by day (rel: t2_fk)" entity and the "Start date" field. The result follows:



Figure 88 - Alias association

To check whether the association was correctly set, you can refer to the relationship specified in the tooltip:



Figure 89 - Tooltip

Click *Finish* and check the SQL code clicking *Generated query*:

```
select
  promotion0_.`promotion_name` as `Promotion_name`,
  time_by_da2_.`the_date` as `Start_date`,
  time_by_da1_.`the_date` as `End_Date`
from
  `foodmart_key`.`promotion` promotion0_cross
join
  `foodmart_key`.`time_by_day` time_by_da1_cross
join
  `foodmart_key`.`time_by_day` time_by_da2_
where
  promotion0_.`start_date_id`=time_by_da1_.`time_id`
  and promotion0_.`end_date_id`=time_by_da2_.`time_id`
```

Comparing the text of the generated queries, you can notice that in this last case the query was correctly generated according to our needs. Figure 90 shows the desired result.

	Promotion name	Start date	End Date
1	Dollar Days	01/03/1997 00:00:00	01/02/1997 00:00:00
2	Coupon Spectacular	01/03/1997 00:00:00	01/01/1997 00:00:00
3	I Cant Believe It Sale	01/03/1997 00:00:00	01/02/1997 00:00:00
4	Two Day Sale	01/04/1997 00:00:00	01/01/1997 00:00:00

Figure 90 - Query result

GIS

The GIS engine of the SpagoBI suite allows the visualisation of business information on a cartographic layer provided by spatial systems using WFS/WMS standards.

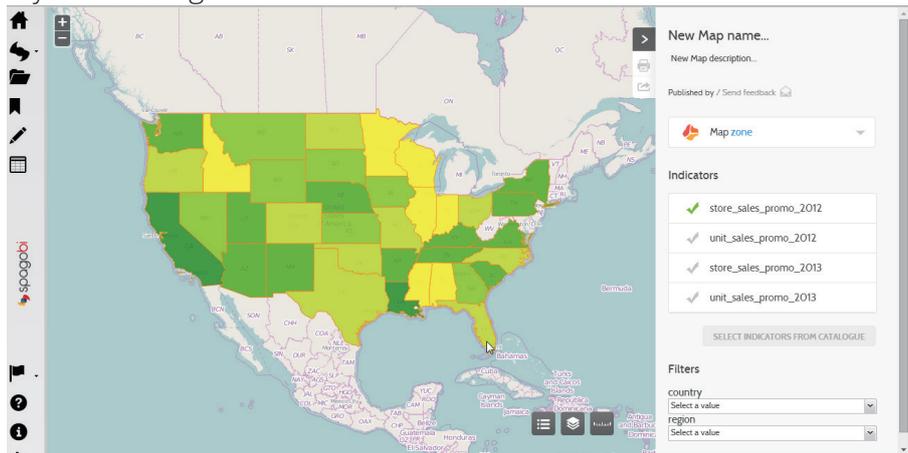


Figura 91 - GIS

This feature is based on a user-friendly interface. Let's see how the user can interact:

- Click the  icon to view the legend

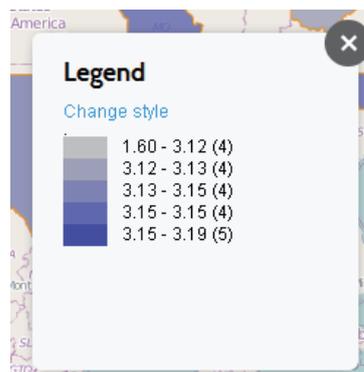


Figure 92 - GIS legend

- Click the  icon to select the layers to be visualized or the list of map providers

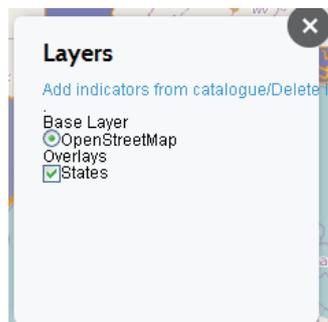


Figure 93 - GIS layers

- Click the  icon to select the layers to be visualized or the list of map providers



- Choose whether you want your business data to be represented on a map with zones or points.

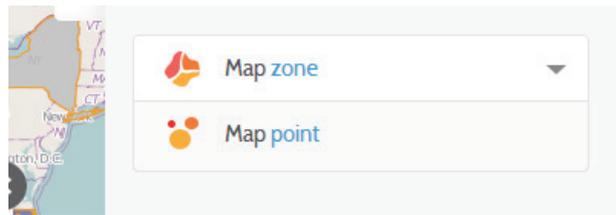


Figure 94 - Data representation levels

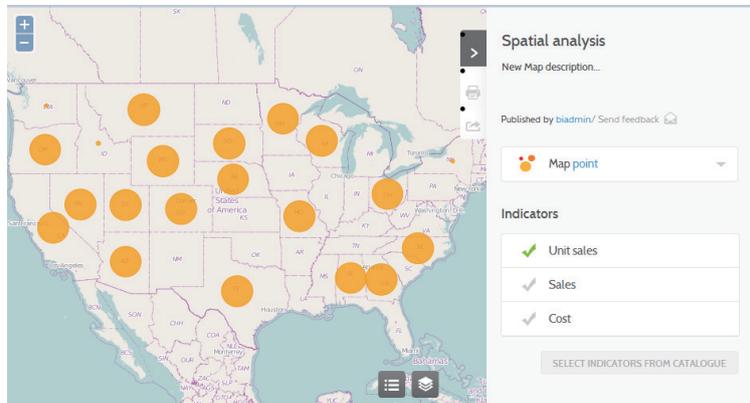


Figure 95 - Map Point view

Finally the following feature allows users to easily identify the data to be visualized on the map by selecting the indicator or a filter that can make visible only a selected part of the analysis, filtering on some characteristics as shown here.

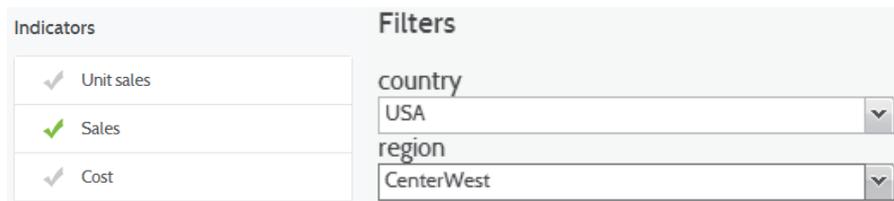


Figure 96 - Indicators and filters

Users' preferences can be saved in a new analytical document as follows:

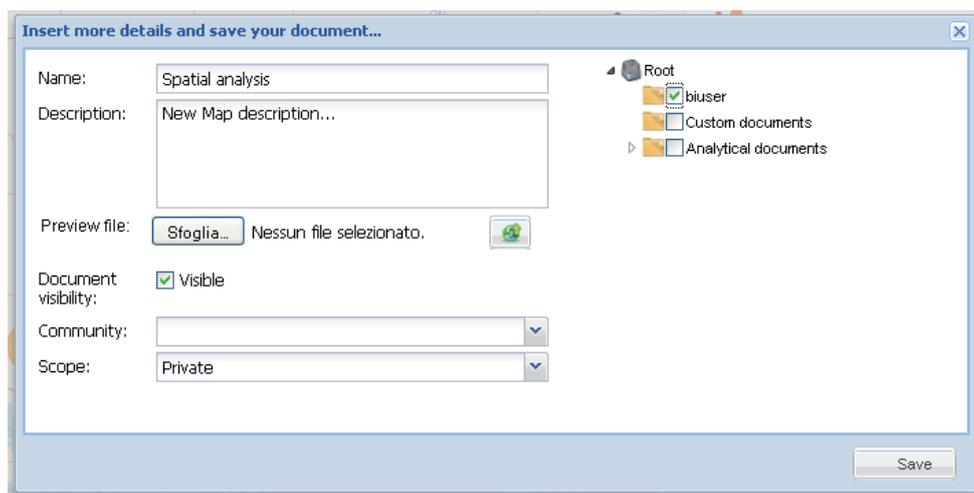


Figure 97 - Save new map

GEO

The GEO engine uses a static map catalogue and allows users to re-aggregate information dynamically according to a geographic hierarchy (e.g. nation, region, district) defined by the Administrator. This engine can be used irrespectively of the so-called geographic context - it allows to display the distribution of indicators on any structure that can be represented on a map (e.g. process flow diagrams, hardware infrastructure topology, etc.)

Generally, the interface is composed of the following three elements:

- Hierarchy navigator
- Level navigator, allowing users to integrate additional levels in the map (e.g. borders, charts, values)
- Map.

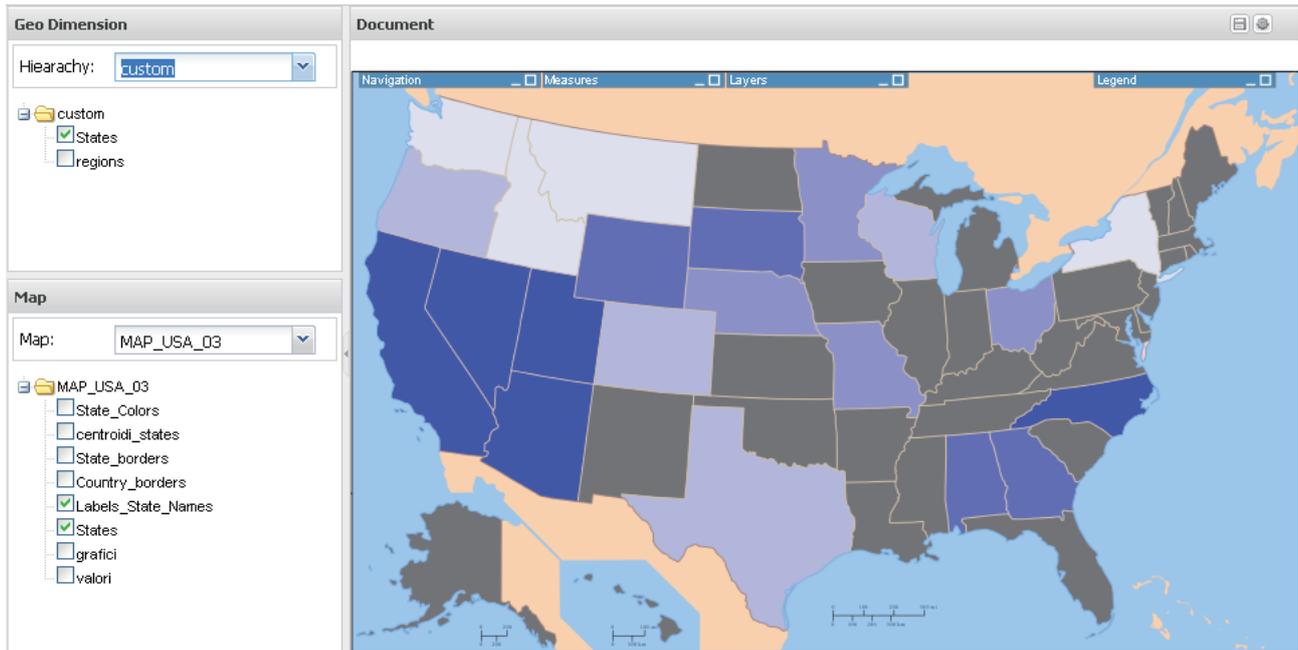


Figure 98 - GEO interface



Each time hierarchies and/or levels options are modified, it is necessary to re-execute the map using the  button at the top right

To change hierarchy from *STATE* to *REGION*, select the region level and re-execute the map.

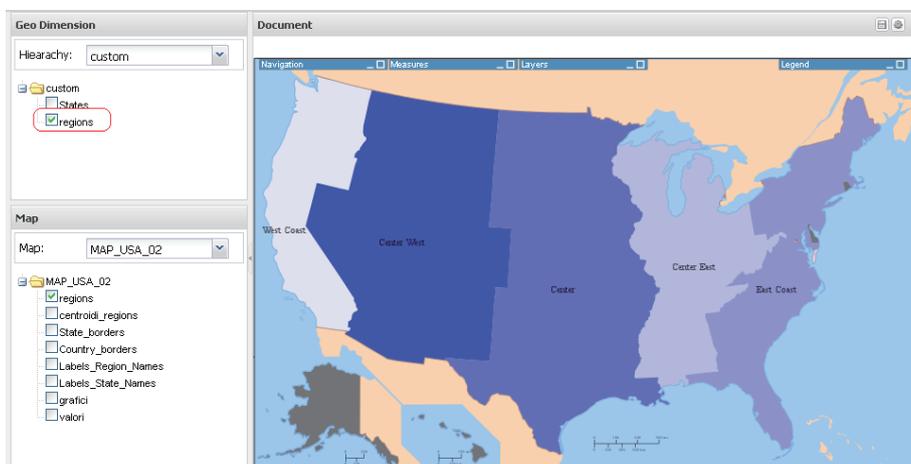


Figure 99 - REGION hierarchy

To add or modify a level of the map, use the options available in the left side of the window. For example, as shown in Figure 100, you can add the charts associated to each region.

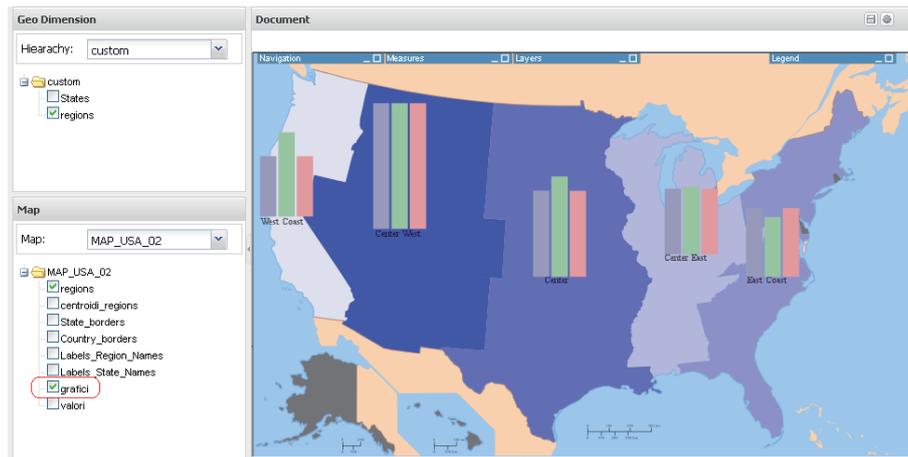


Figure 100 - Charts level

Users can navigate the analytical document, specifically:

- Zoom-in and zoom-out using the Navigation window
- Access the legend
- View the measure catalogue and choose the measures to be visualized
- View the list of available levels and select the desired ones
- Associate an area or a point on the map to an analytical document - alternatively you can make information appear in a box. This option can be configured by administrators.

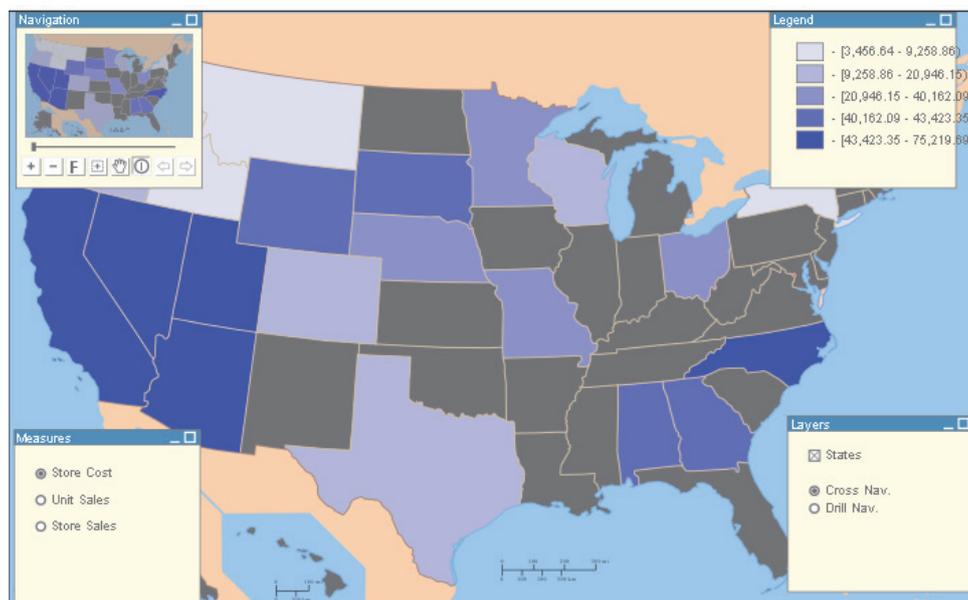
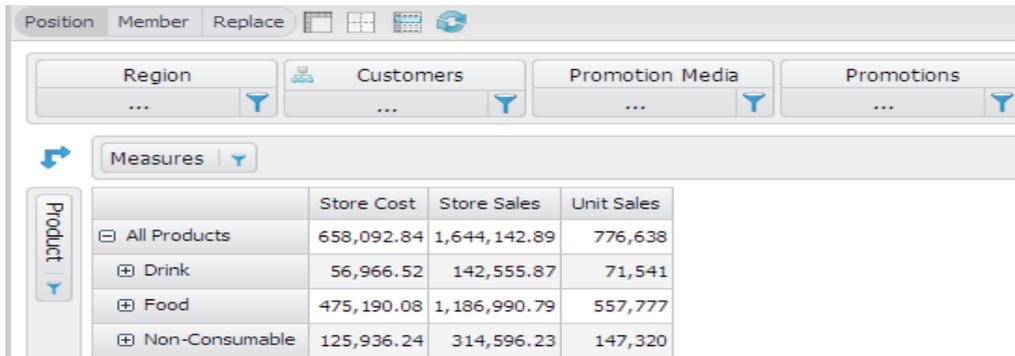


Figure 101 - GEO options

OLAP

SpagoBI supports multi-dimensional analysis thanks to its OLAP engines offering higher flexibility than structured reports. After having defined the analytical axis with specific hierarchies and measures, users can analyze data on different levels of detail and from different perspectives through traditional drill-down, drill-across, slice-and-dice, drill-through processes.



	Store Cost	Store Sales	Unit Sales
All Products	658,092.84	1,644,142.89	776,638
Drink	56,966.52	142,555.87	71,541
Food	475,190.08	1,186,990.79	557,777
Non-Consumable	125,936.24	314,596.23	147,320

Figure 102 - OLAP

The OLAP interface includes a toolbar allowing users to set up data navigation and visualization options.

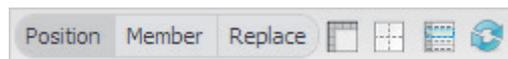
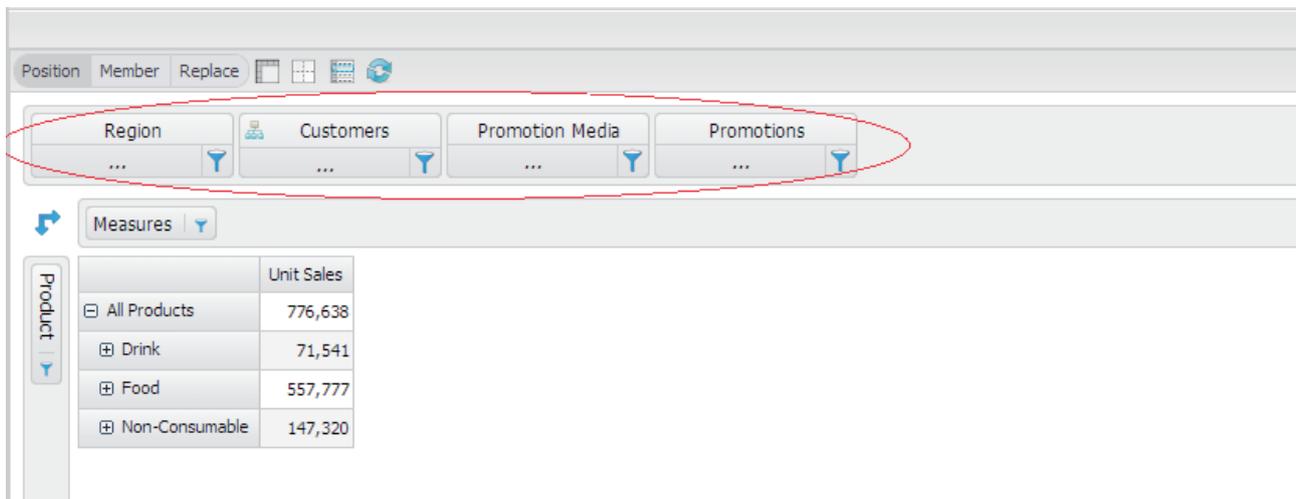


Figure 103 - Toolbar



	Unit Sales
All Products	776,638
Drink	71,541
Food	557,777
Non-Consumable	147,320

Figure 104 - Configuration window for OLAP cube

Configure the navigation cube

The Olap cube can be set up by adding attributes or filters. Users can filter on characteristics by clicking the filter icon 

A wizard window appears and customization can be performed on data, without inserting any other row in the cube.

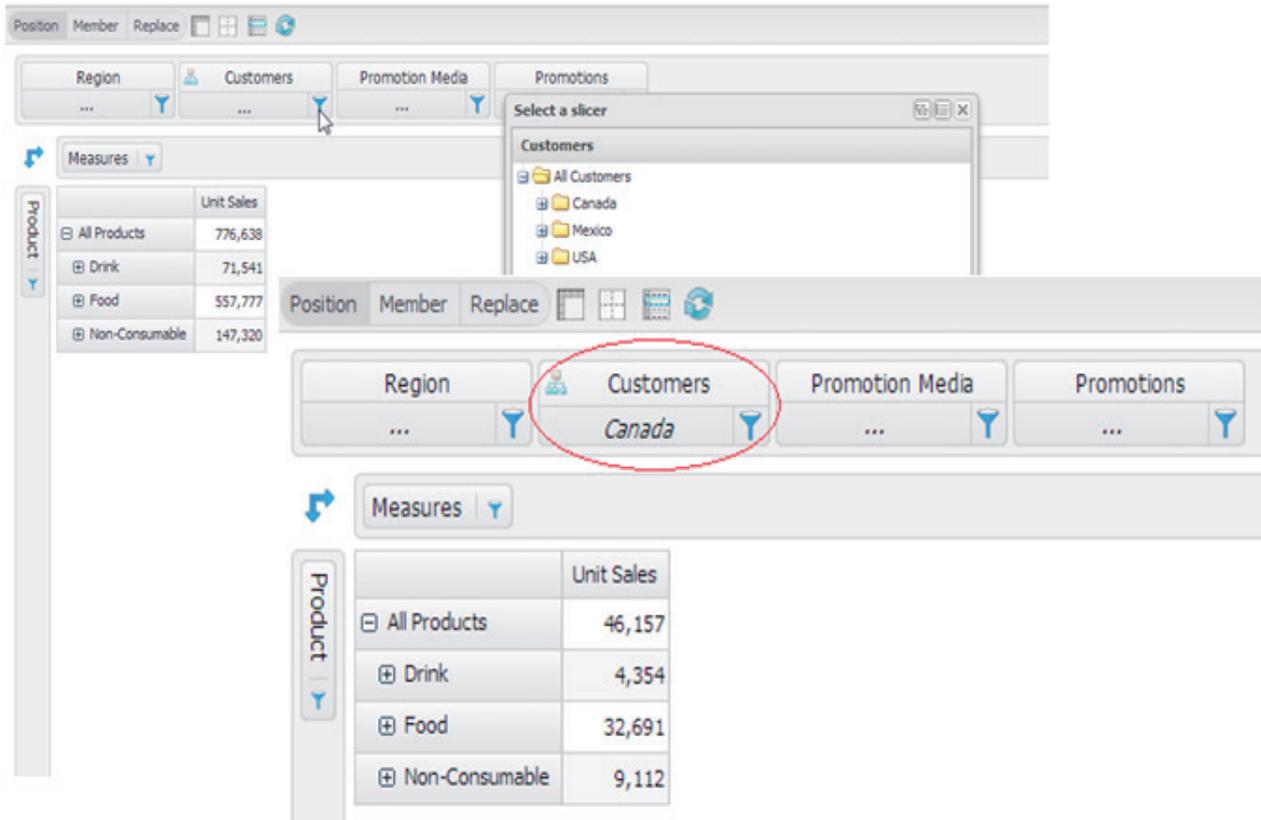


Figura 105 – Filters

Users can modify the cube in different ways, by simply drag and drop attributes and measures from the top bar into the side bar, and filtering them there.

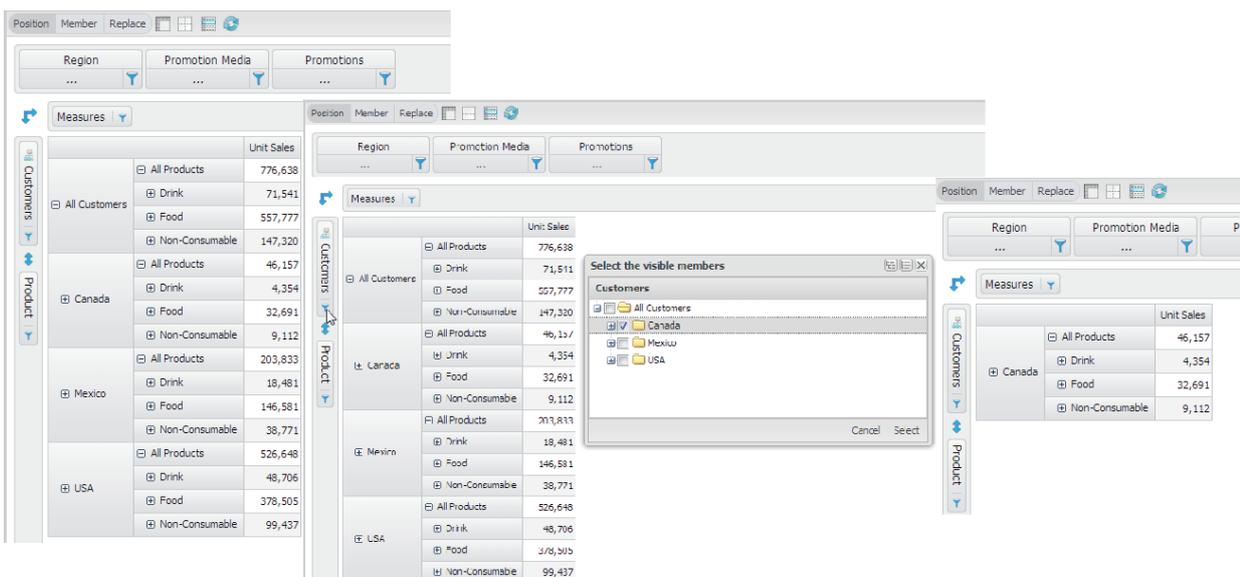


Figure 106 – Filters on cube

Moreover can reorganize attributes in the bar according to their needs, by clicking on the double arrowed icon .

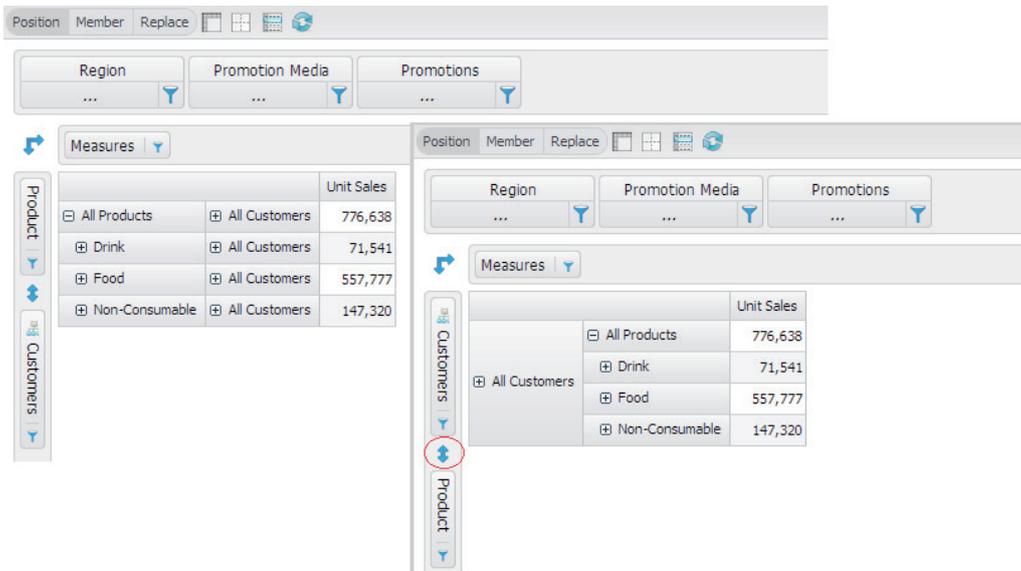


Figure 107 – Swap attributes

Drill operations

OLAP supports drill operations, enabling users to display data at different levels of detail, navigating through the hierarchies within the OLAP cube. Three different drill operations are available:

Drill member & Drill position. This feature allows users to expand fields that have detail data at a specific hierarchy level, keeping the upper hierarchy levels visible.

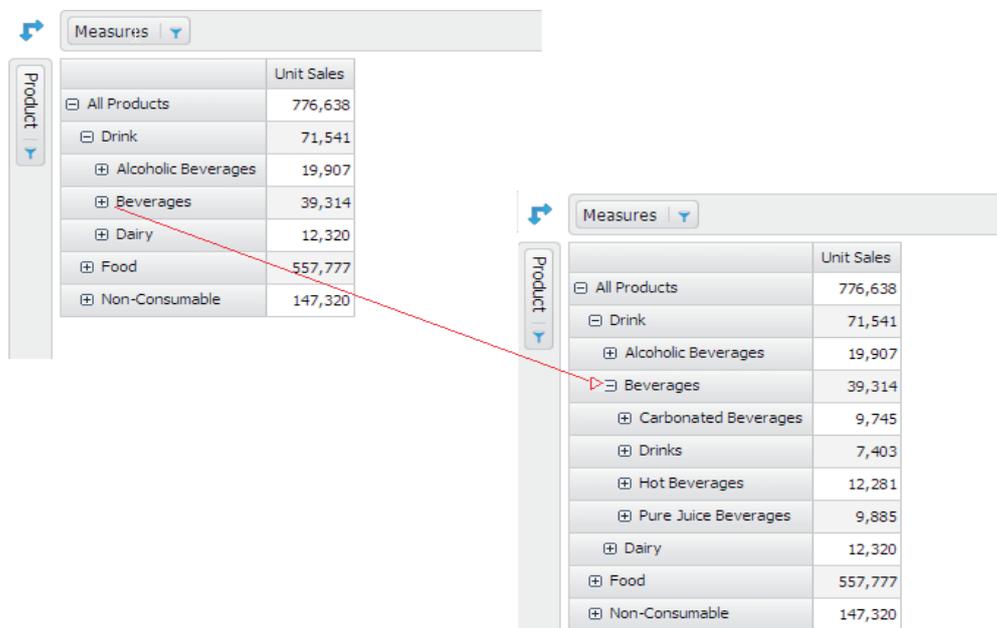


Figure 108 - Drill member & Drill position

Drill Replace. This feature allows users to expand fields that have detail data at a specific hierarchy level, losing the information related to the selected parent level in the analytical hierarchy. It displays the members of the selected level.

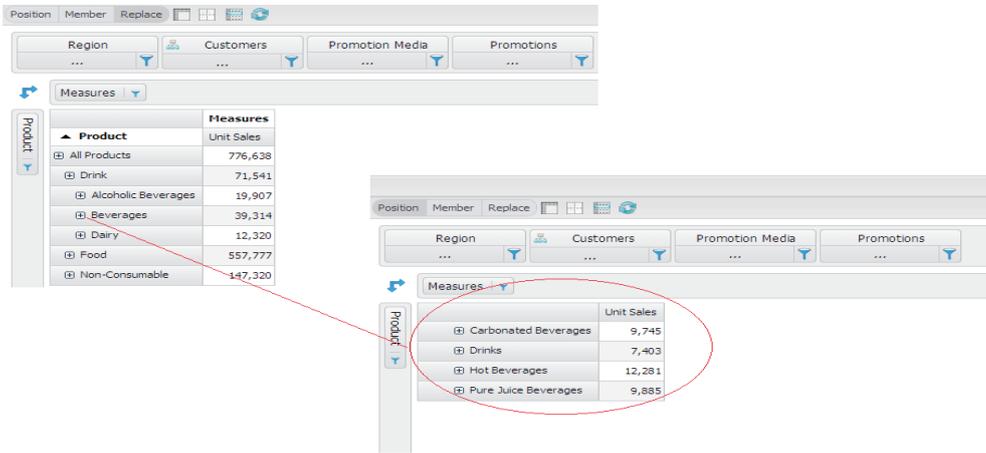


Figure 109 - Drill replace

Hide empty cells

This feature hides all empty cells in the table, which have no value.

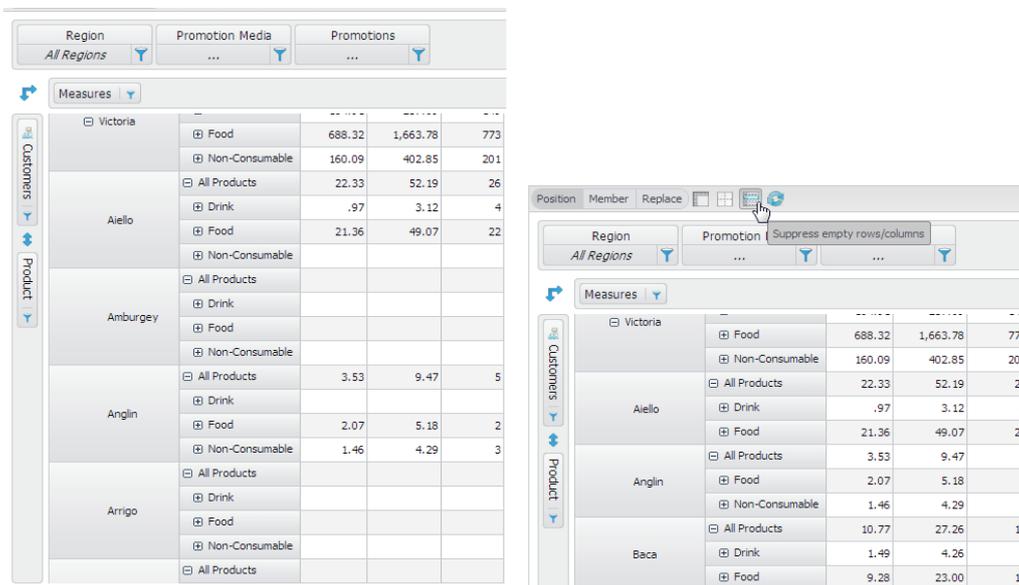


Figura 110 - Hiding empty cells

Invert axis

This operation is called “pivoting”: it moves rows to columns and vice versa.

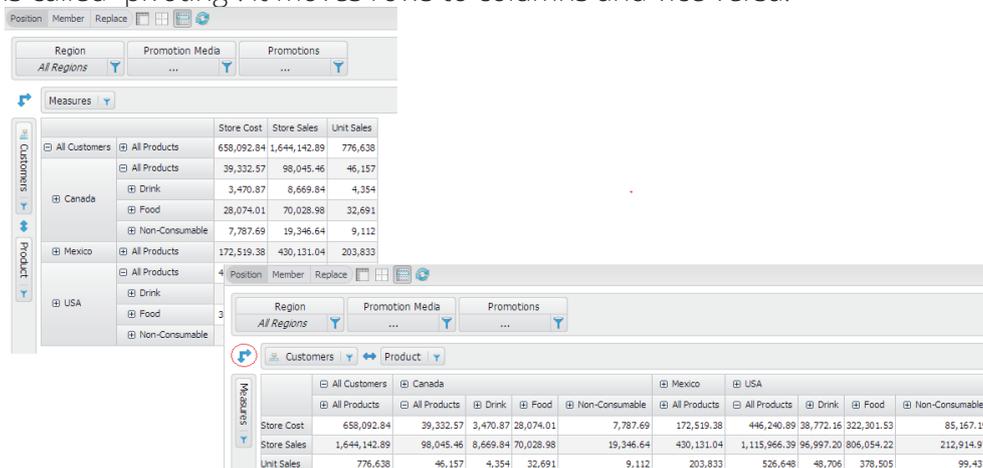


Figure 111- Pivoting

Save OLAP navigation

This feature allows users to save a level of navigation for future use. In this case, the query is made available in the OLAP execution window as a customized view.

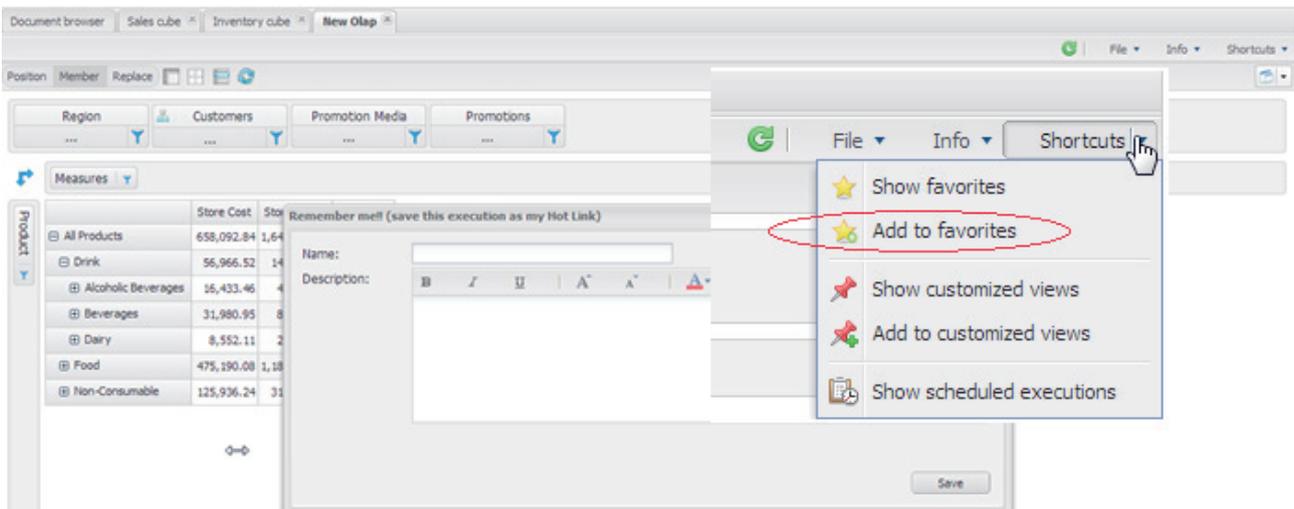


Figure 112 - Saving an OLAP query

REFERENCES



General resources

A full range of public resources for all those who wish to learn more, get involved and contribute to SpagoBI growth:

SpagoBI website: www.spagobi.org
On-line demo: www.spagoworld.org/SpagoBI/
Success Stories: <http://bit.ly/1pDEDOj>
Videos: <http://bit.ly/1uyr0aR>
Webinars: <http://bit.ly/1CA4DDj>
Newsletter: <http://bit.ly/1rBCyHW>

Technical support and services

Various resources and types of service to let SpagoBI Labs support you through all project stages.

Professional services: <http://bit.ly/1qR6LxG>
Documentation: <http://bit.ly/1yxMdEx>
Shop: <https://spagoshop.spagoworld.org/spagoshop/>

