In-Depth User Guide

Knowage v6.3

Anthony Rousseau
08/05/2019

Table of contents

Before starting

Target

- **General User Manual**: it is aimed at end users. It provides a first approach to Knowage interface and functionalities. It can be used as a first approach to Knowage. It focuses on all those elements which are shared among the products and involves the end user.

User Guide

Introduction

This document is intended to provide a quick introduction to Knowage end user interface. It can guide you through your first approach to the suite, describing the different graphical elements and the general end user features common to all Knowage products.

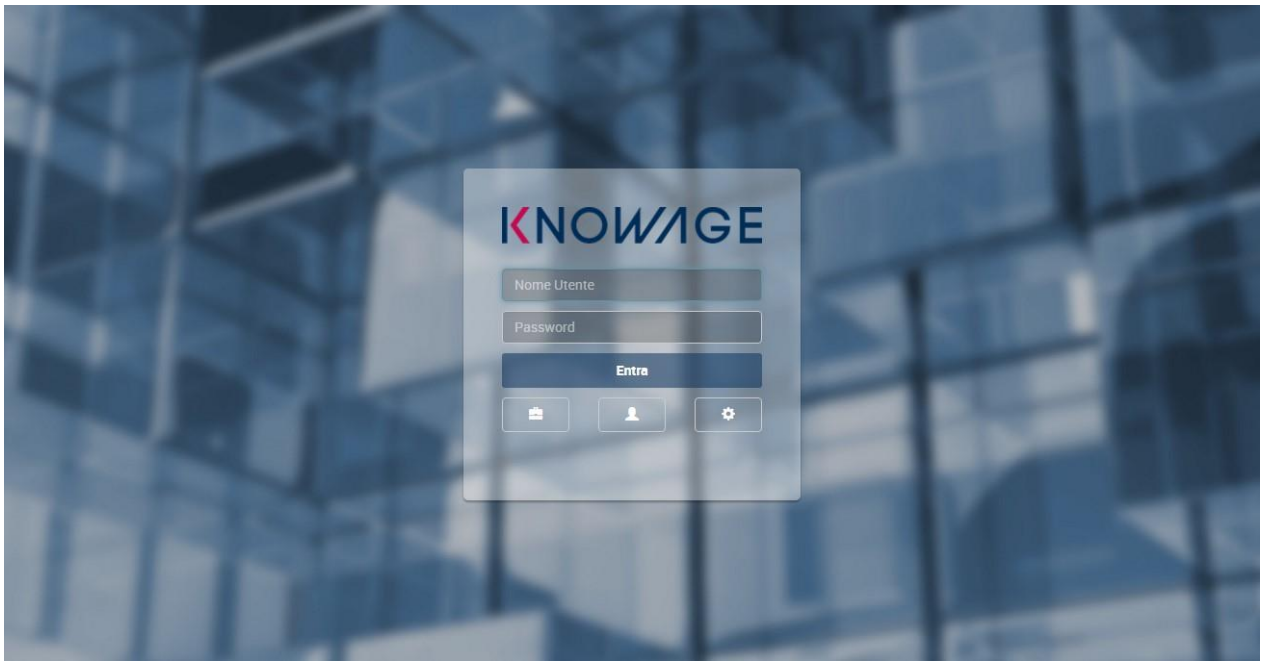
No prerequisites are needed. We suggest following the elements description and try them while reading.

User Interface

This chapter focuses on Knowage user interface, providing detailed information on the Main Menu, the Document Browser and some general settings concerning analytical documents. First, a short introduction on profiling rules it provided.

Preliminary information

The administrator will provide you a username and a password to log in Knowage environment.



Login page

These credentials identify you as user and are associated to your *role*.

In Knowage suite, roles represent categorizations of group of users, granting each user with different rights and visibility criteria on documents and data, according to their business profile.

This is what is called *Behavioural Model*. It allows to:

- reduce the required number of analytical documents,
- code only once the behavioural and visibility rules on data,
- guarantee the uniform growth of the project over time,
- guarantee the respect of the visibility rules over time, with no limit on the number analytical documents that can be added.

All the analytical documents are strictly related to the behavioural model. In fact, the behavioural model guides the behaviour of the analytical documents according to the user's role, managing the visibility of documents and data.

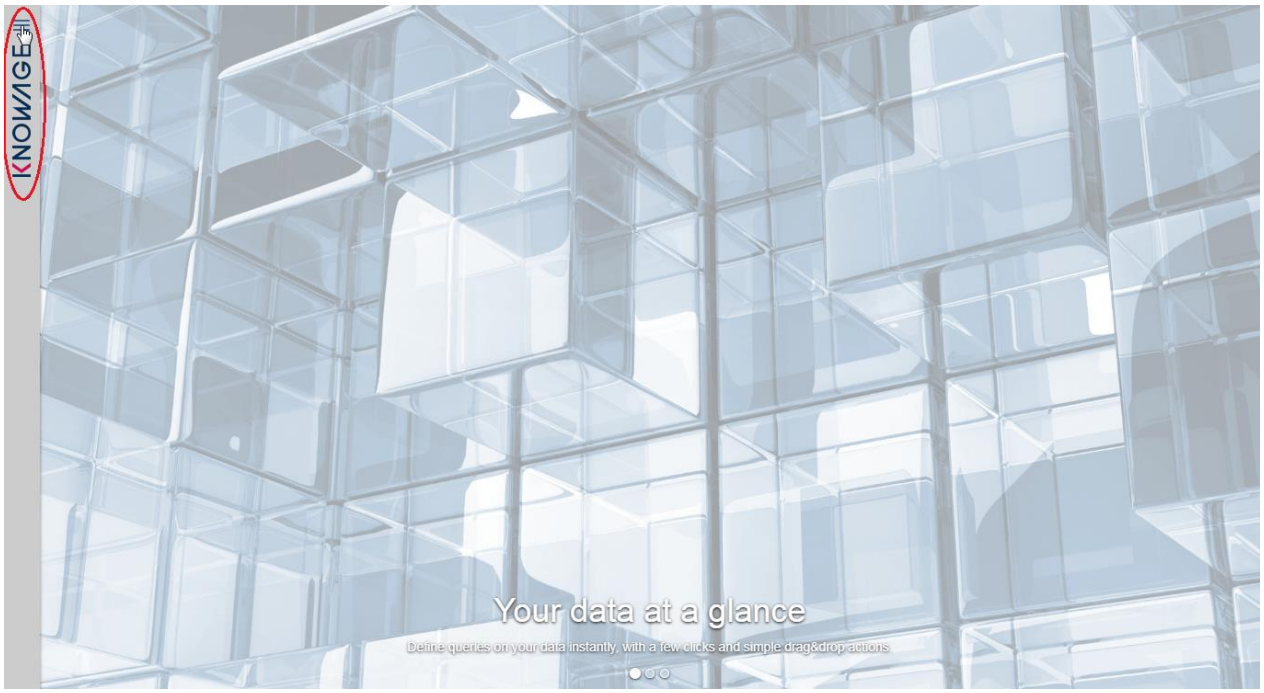
Every role belongs to a predefined *role type*. The available role types are listed and described below.

Role types

Role Type	Description
General administrator.	Manages all Knowage functionalities.
Model administrator.	Manages the behavioural model and its associated functionalities.
Developer.	Creates and modifies datasets and documents.
Test user.	Tests analytical documents.
End user.	Executes documents visible to him and creates ad-hoc reporting analysis.

From now on we suppose the reader has a role of type "End user". If some grants are optional for this role, we will state it.

Main menu



Home page

The menu gives you access to documents, data and all the functionalities that you can use. By default, the menu button is at the left bottom corner of the home page: click it to open the menu.

Main menu consists in a set of icons associated with basic features. It is divided in two submenus: the general menu, which is collapsed, and the BI functionalities menu. It is important to underline that not all the components of the menu are mandatory, but they may be configured by the administrator according to user's needs.

Menu components - General menu

Icon	Name	Description
	Knowage user	Open a hidden menu with extra functionalities.
	Select role	Select the authentication role (available if you are associated to more than one role).

Menu components - General menu

Icon	Name	Description
	Languages	Language options.
	Info	Info about Knowage version.
	Log Out	Log out.

Menu components - BI functionalities menu

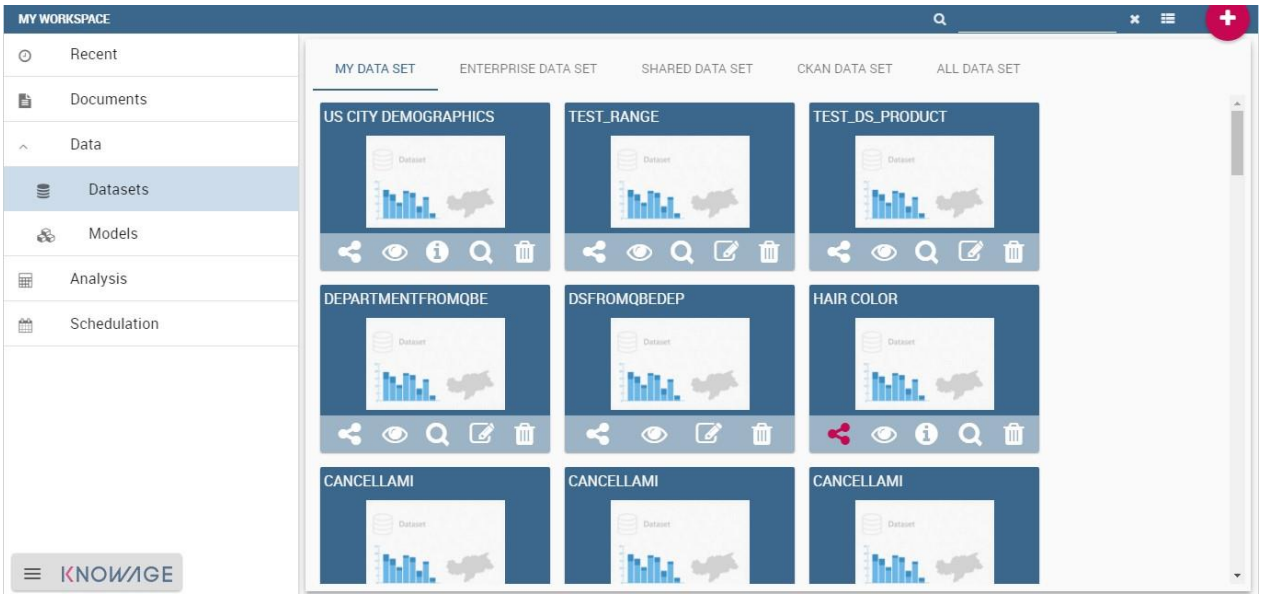
Icon	Name	Description
	Document browser	Show the archive folders and related documents.
	Workspace	Inquiry, navigate and create your data. Available only for KnowageBD and KnowageSI.

Document Browser

It enables you to access the documents shared to you and the functionality trees.

Workspace

This is available only for KnowageBD and KnowageSI. Entering the Workspace, you will find the sections: **Recent**, **Documents**, **Data** and **Analysis**. The “Recent” area shows the latest documents you were working on, while “Documents” contains the analytical documents the user asks to be archived on this area. This way the user has a more rapid and efficient way to retrieve the documents of his interest. The “Data” section is made up of the “Dataset” and the “Models” subsections. In the **Dataset** one you can access all the self-service BI features. Once entered this section, your datasets appear divided into four categories:



The **General menu** is identified with the first icon in General Menu and a label containing your user name. Opening the general menu, you have the following extra buttons:

Select role
: If your user is associated with more than one role, Knowage requests you to specify the default role. You can select it when executing a document, or right after authentication by clicking on this icon and choosing a default role.

Languages
: Select the language of Knowage environment.

Info
: View the details of current Knowage version.

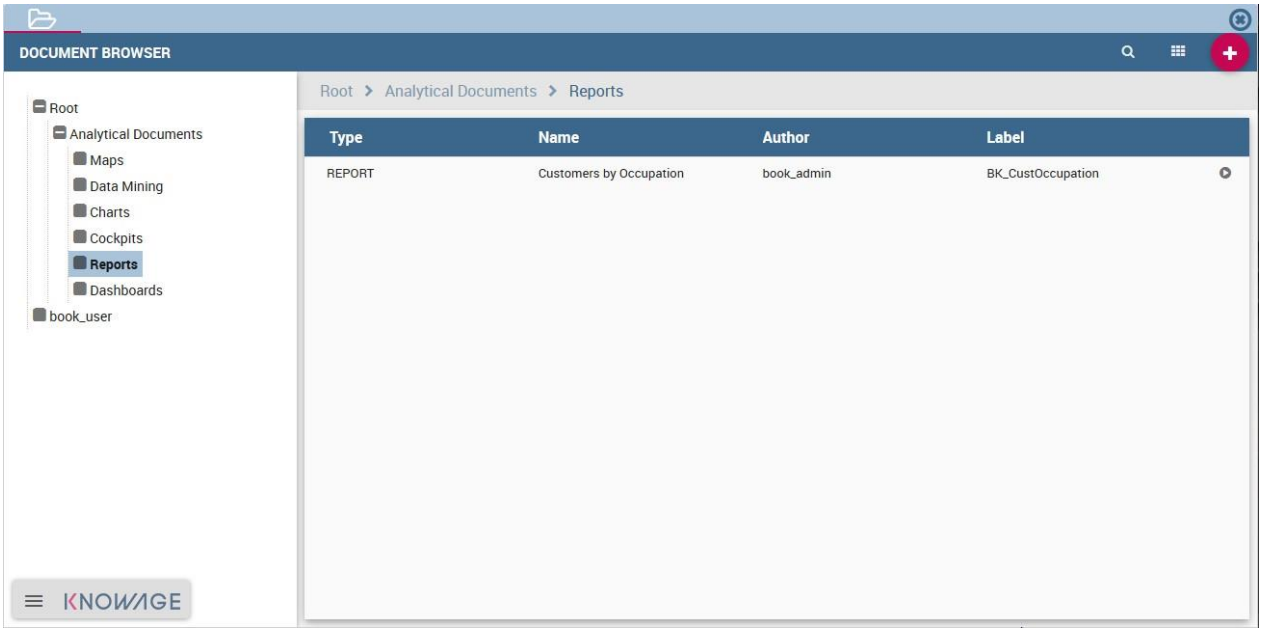
To conclude the overview of this area we describe a not mandatory element. When configured by the Knowage administrator, you can have quick links to a static page, a document, a folder or an external application, like for instance a web page. These links are displayed below the BI functionality's menu, as shown below.

Quick links

You can minimize the main menu by clicking the right arrow at its bottom right corner. This way the main menu is replaced by the menu button, so that you can close/open it according to your needs. You can move this button around the page by dragging and dropping it. Choose the position that best fits with your work.

Document Browser overview

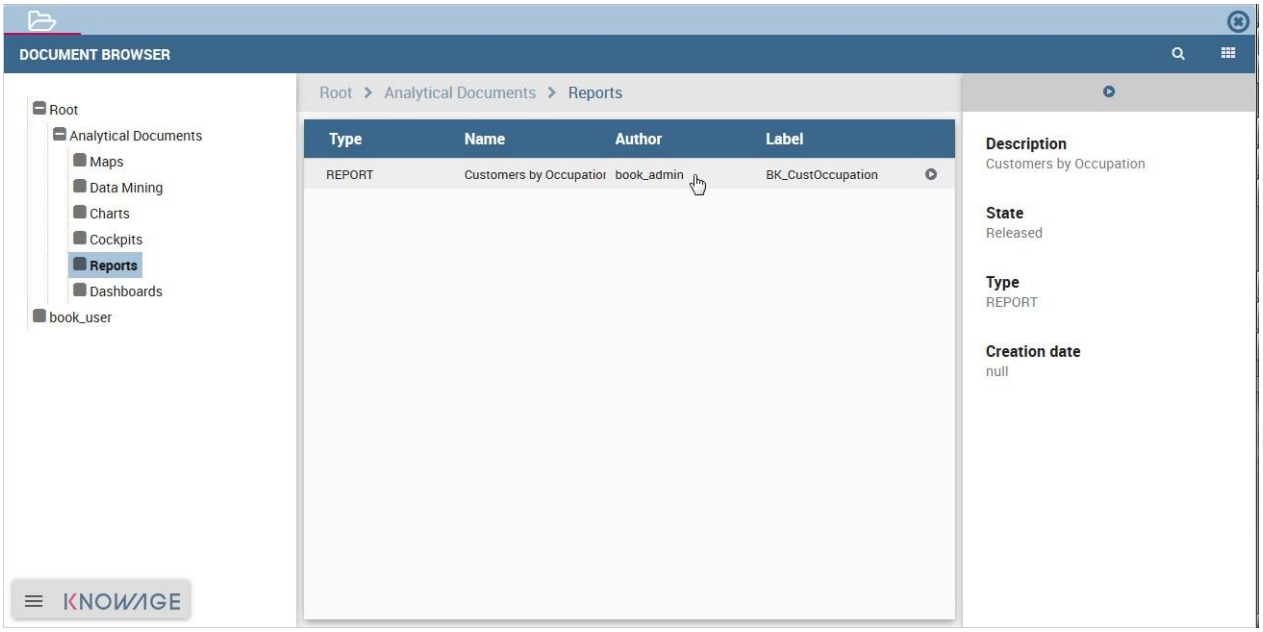
From BI functionalities Menu, select  to open the Document Browser.



Document Browser

By default, the page is divided in two parts, as shown in Figure above: in the left side there is the functionality tree representing the folder structure, while on the right you can see the list of all documents contained in the selected folder.

Each line shows the label, the name, the author and the type of the document, while the play button at the end of each row executes the document. Moreover, clicking on a line opens a side panel on the right of the page. Here you can see more metadata information such as the document description, the state and the creation date (see Figure below).



Documents details expanded.

At the top of this side panel you find the  button, the same one you see at the end of each document line. Click on it to execute the document.

Document Execution

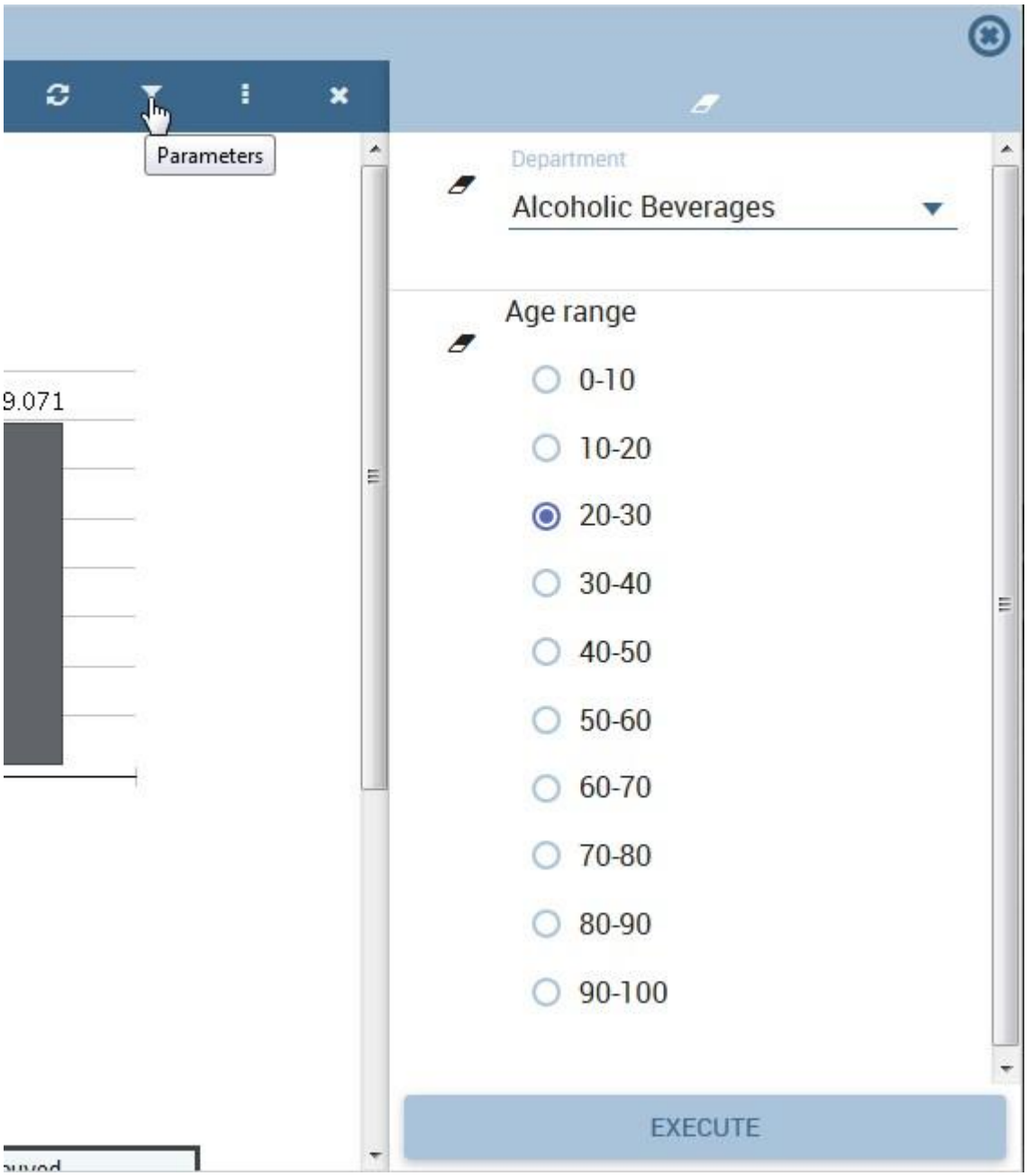
In this chapter we describe all the features related to Knowage analytical documents, such as parameters management, printing, exporting and so on.

First, notice that once you execute a document from the document browser or from the menu, it is visualized full screen. In the first case, you can return to the document browser by clicking on the folder icon located at the top left.

Back to Document Browser

Parameters management

Knowage documents may have associated parameters. If any, you will be asked to select the chosen parameter's values in a collapsible panel located at the top or on the right side of the page. If this is the case, choose the parameters values and then click the **Execute** button to run the document. In case there are only optional parameters or default values are already defined, the document is directly executed after the first click on its relative icon.



Parameters panel

Mandatory parameters are shown in bold together with an asterisk on the right, while optional parameters are normal shaped.

It is possible to show or hide the parameter panel by clicking on the filter button located in the document toolbar. With the **Reset** button at the top of the panel you can clear the form.

Furthermore, 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 at the top right corner of the parameters panel.

- **Reset** inserted values for parameters;
- **Open saved** a window listing the **saved parameters**, so that you can select or modify them;
- **Save** the parameters. Here you can choose between two options: **Public** means visible to all the other users that share your role while **Private** means visible only to you.

Document Toolbar

All documents inside Knowage environment share the same toolbar with different features. We provide first a short description and next a detailed explanation.

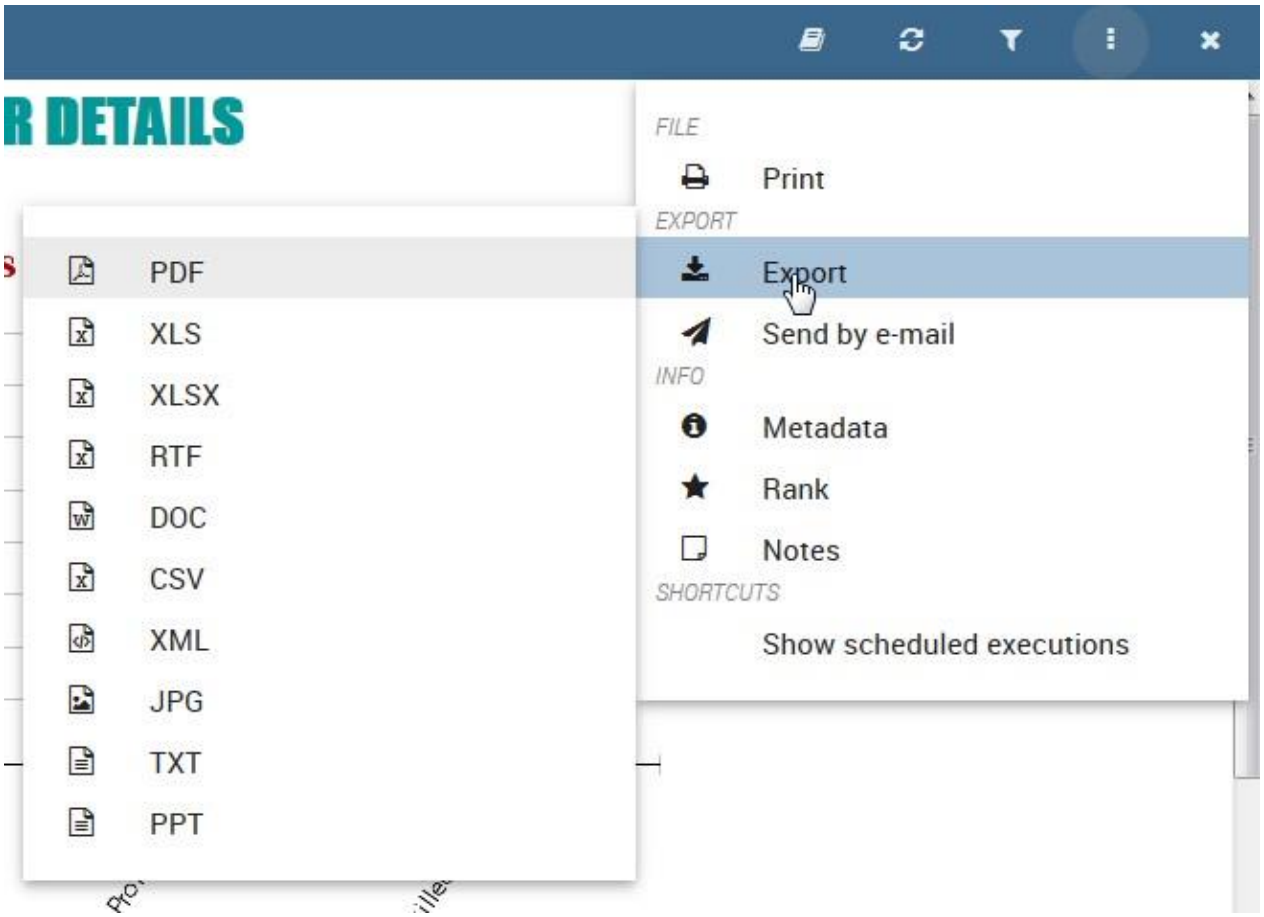
Document Toolbar

The  button is to access the help online as defined in the Glossary and it is available only in KnowageSI.

The  refreshes the document.

The  opens the parameters panel and it is visible only if there are parameters associated to the document.

The  opens the contextual menu shown in figure below. We describe the main functionalities provided by this menu in the following.



Toolbar contextual menu

Exporters

Each Knowage document can be exported into several formats, depending on the options offered by the engine.

Clicking **Export** in the document toolbar you will see the available formats for the current document. Select one and check the exported document.

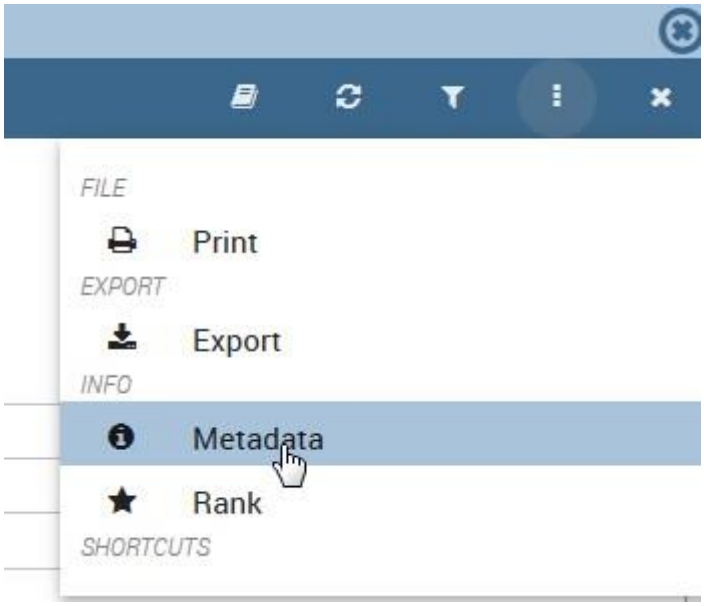
Business and structural metadata

Knowage allows the definition of business metadata to describe an object, in our case a document. Business metadata, unlike technical metadata used in Knowage Meta to build the metamodel, are business information associated to the document intended to help users to understand, access and classify it. As such, they have been mainly conceived for the end user understanding.

There are three types of business metadata, some of them are editable while others can only be read. General metadata are read-only, while short and long text metadata are editable. General metadata contain basic information about the document, which cannot Notes be altered because they are related to the structure of the document (e.g., type, engine, label). They provide useful and synthetic information on the document. Short and long text metadata should be used to add relevant business information: all the allowed users will see this information, which will help them understand the purpose and context of that document.

In general, metadata should be edited by users with adequate expertise and authority to do so. Therefore, it is possible for the administrator to assign the right to edit and save metadata only to some users. The right is not specific to a profile, but it is part of the authorizations that can be granted to any role. This applies to bookmarks as well.

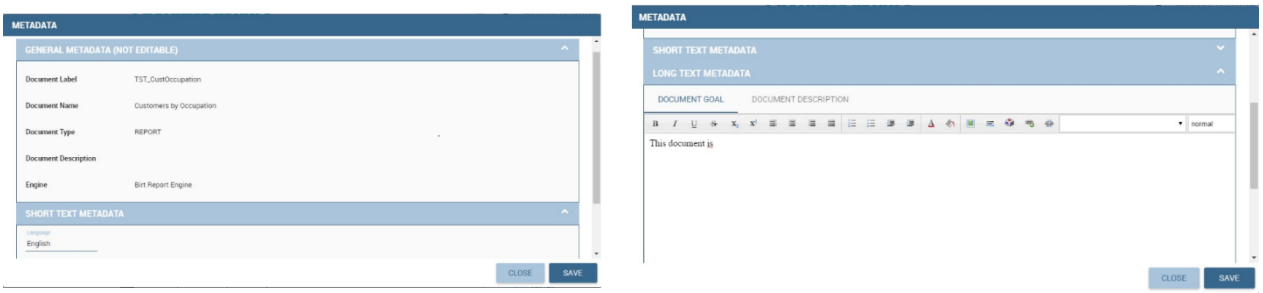
Metadata can be accessed from the toolbar clicking the corresponding item in the contextual menu as shown in the following figure.



View document metadata

As stated before, in order to see and/or edit metadata the user roles need some grants. Knowage administrator manages this authorization. If you have editing metadata authorization, you will be able to see to change them. If you want to edit short metadata just click in the value area and write what you prefer. If you want to edit long metadata just click in the value area and an HTML editor will appear.

When you are satisfied with what you wrote just click on the **SAVE** button. We provide the following example.



Left: General and short metadata, right: editing long metadata

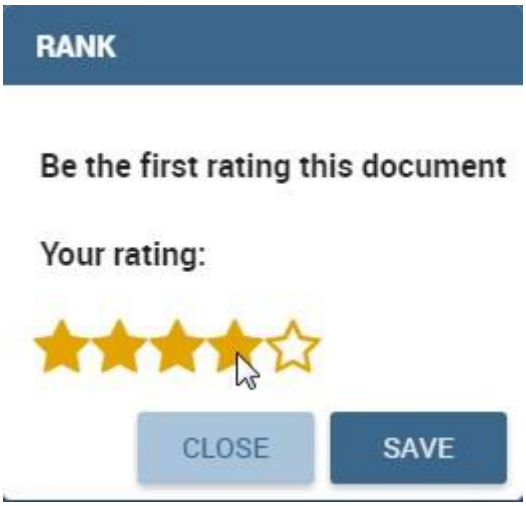
Notes

Knowage offers a simple collaborative tool to share notes and comments on documents, allowing users to share information and receive feedback. This may be useful, for example, to limit the number of exchanged e-mails: notes can be stored online and are accessible to all users sharing the same access rights. Each user can add a single note to a single document.

Click on **Notes** in the contextual menu of the document toolbar. In this way a text editor opens where you can type your comment. Make it private or public (i.e., accessible to users with the same rights as you) by selecting the preference in the appropriate box. Click **Save** to confirm. All public comments from all users, as well as your private notes, will be shown the next time you open this window. If you want to edit or delete a note, click on the Rate document corresponding symbols in the **Note list** tab. You can also export the note in PDF or RTF format.

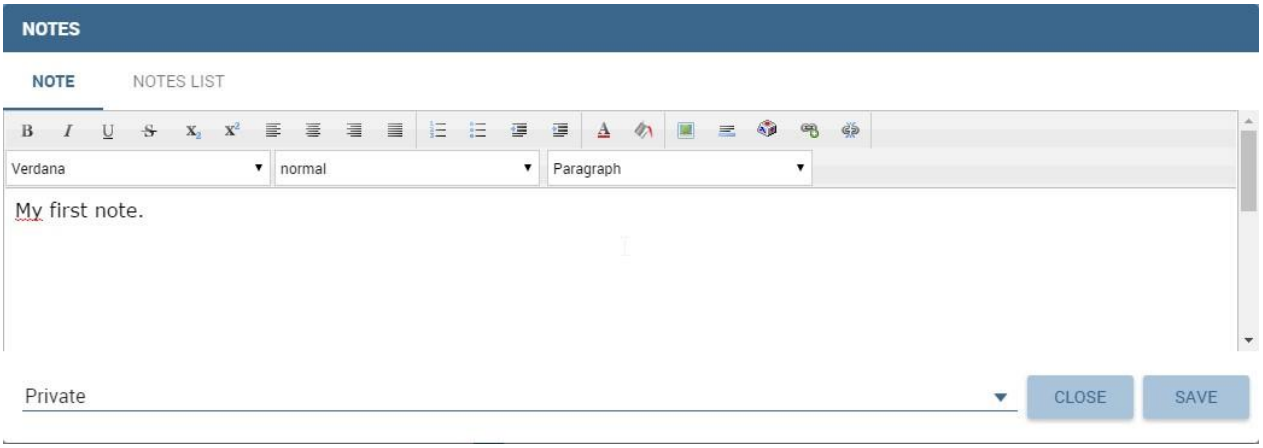
Rate document

The aim of this functionality is to acquire explicit quality data that can be further used. Specifically, it allows the administrator to identify anomalies about the use of an analytical document by end users. Rating a document means assigning it a value from the end user perspective. This brings additional information with respect to traditional audit and monitoring data, which can track the number of executions but cannot interpret users' motivations and feelings.



Rating a document

To rate documents, click on **Rank** in the contextual menu of the document toolbar. Choose the document ranking in the pop-up window and click **Save**. The administrator can exploit the result of votes, in order to evaluate and improve the quality of the document.



Share notes and comments

Scheduled Execution

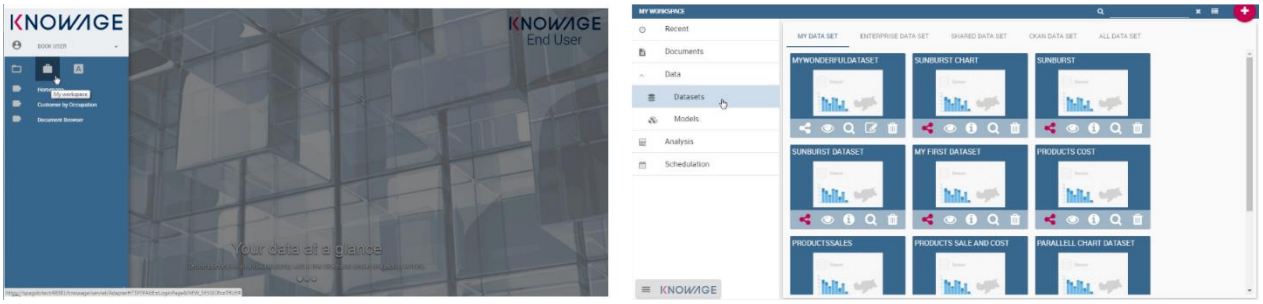
The administrator has the possibility to set schedule for documents. These schedules can be accessed by **Show Scheduled executions**.

Functionalities

Basic Data Access

A dataset is a way to read data from different sources and represents the portion of data used by various documents. Suppose you want to create a bar chart showing the sales trend for the current year; in this case you need to pass to the document the total sales amount for each month of the current year. You can create your own dataset uploading an XLS or a CSV file or use a dataset already defined. Knowage offers you also the chance to download open data from WEB thanks to CKAN integration. Moreover, you can create your own and more complete dataset from different sources through the dataset federation. In the following we will describe all these functionalities.

Let us suppose to enter, with end user credentials, the data management area clicking on the **Workspace** icon from BI functionalities menu as shown in figure below and the **Data** section of the window.



Access to My Data area

Afterward you have the subsections: **Dataset** and **Models**. Select **Models** to explore the models and the **Dataset Federation** area. Please note that the **Dataset Federation** functionality is available only in KnowageBD and KnowageSI.

Dataset

Into the “Dataset” area we find all the datasets classified according to their types. The datasets are categorised as follows:

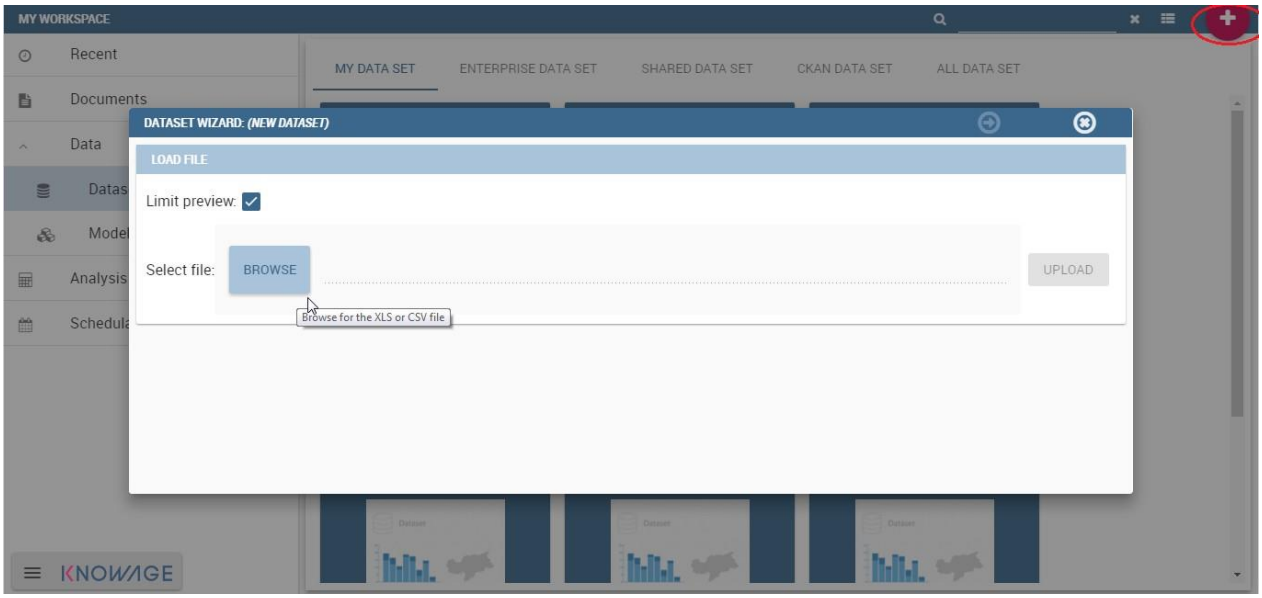
- **My dataset**: datasets created by yourself uploading a CSV or XLS file or creating a query on a business model using the Qbe interface;

- **Enterprise dataset**: certified datasets, namely datasets created by the technical/experts' users and shared with the end user.
- **Shared dataset**: datasets created and shared by other end users;
- **CKAN dataset**: in this area you can download public datasets and visualize your CKAN datasets;
- **All dataset**: in this folder are stored all the available datasets, namely all datasets contained in the classes just described.

My dataset

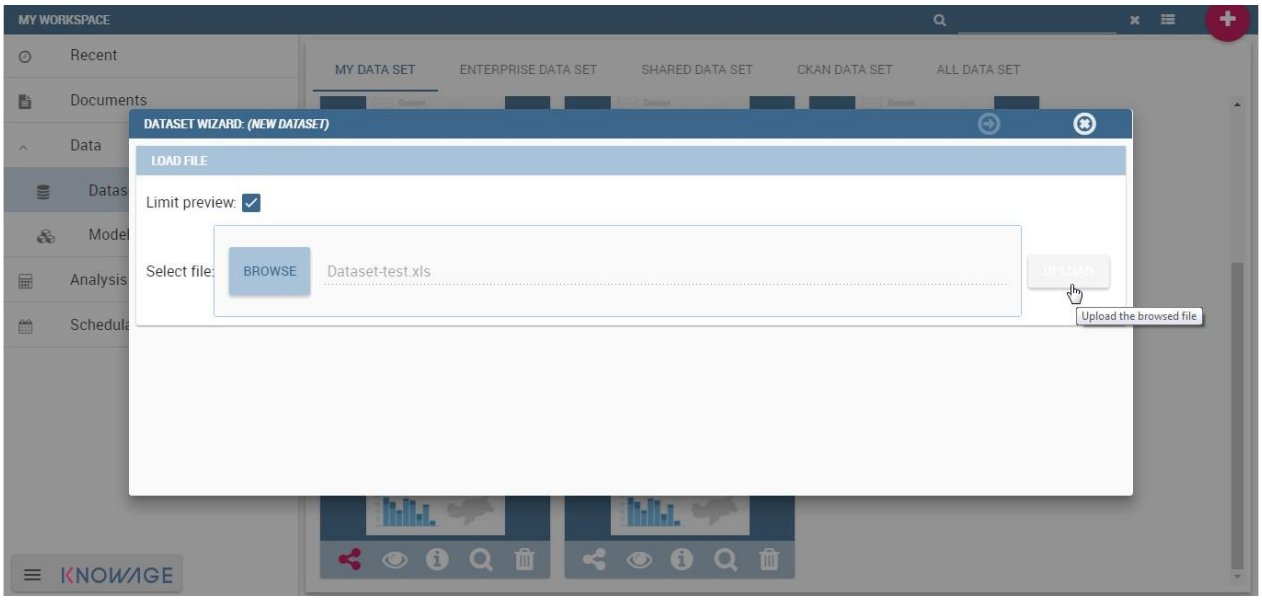
In this area you can create datasets uploading your own files.

Click **Create Dataset** to open the dataset wizard which guides you through the dataset creation. You can choose between XLS or CSV file as in the following figure.



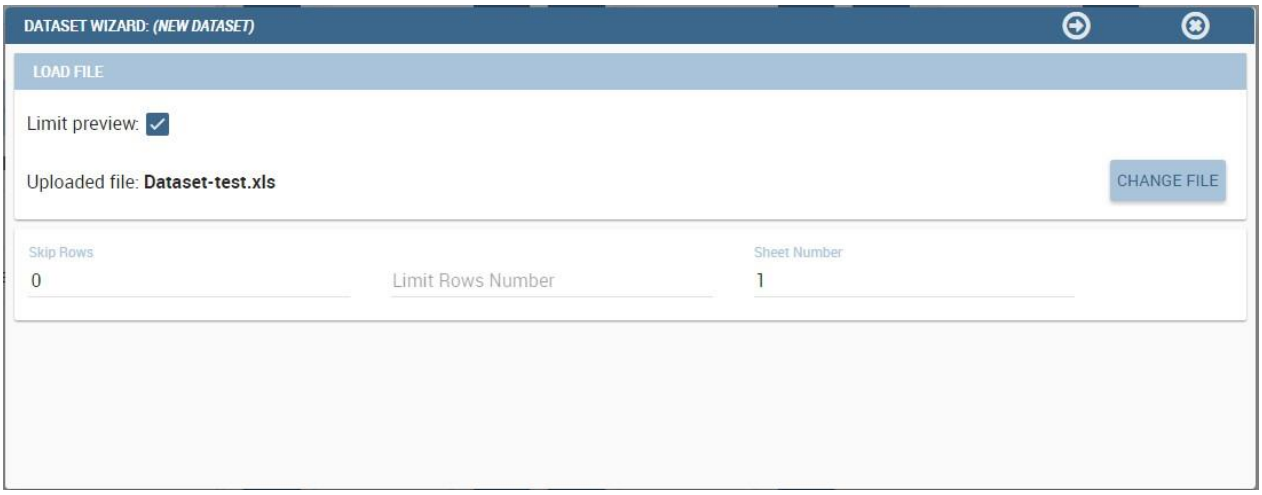
Dataset creation.

In the example shown in the next figure, we upload an XLS file.



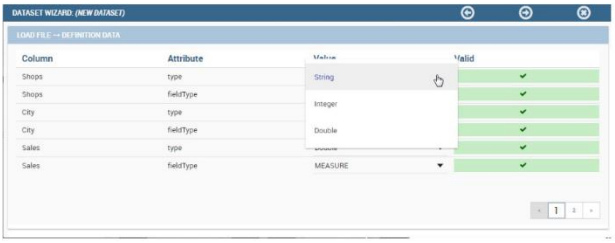
Uploading XLS for dataset.

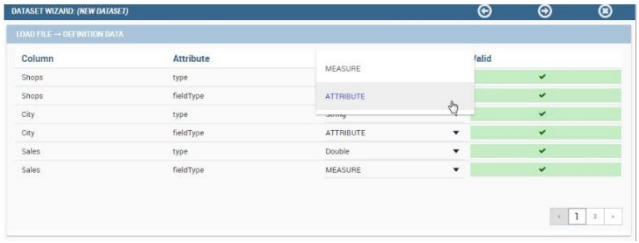
The wizard, shown below, leads the user to insert some information to configure the dataset. For instance, to specify the number of rows to skip or to limit and which sheet (of the XLS file) to pick up values from.



Configuration features.

Once you have uploaded the file, you can check and define the metadata (measure or attribute) of each column. To switch a measure to an **attribute** (or vice versa), click on **Value** column of the interested row field as shown below.





Change metadata.

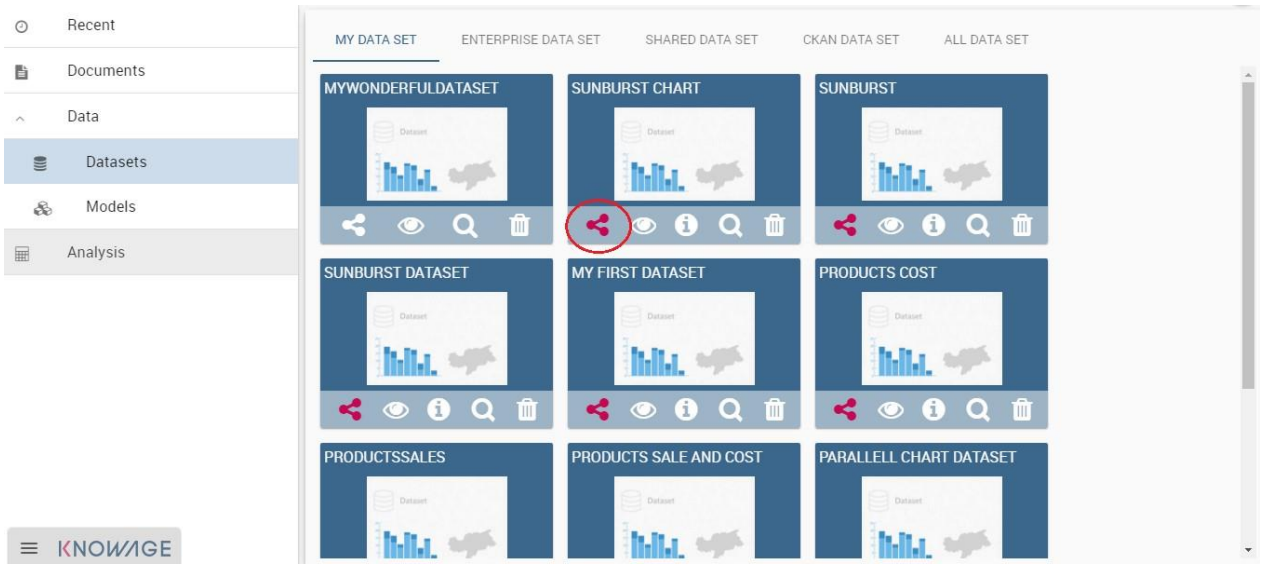
Just few steps before saving the dataset:

- Check the data preview in order to verify the accuracy of your data;
- enable or disable the persistence of dataset. Thanks to this functionality the server creates a snapshot of the extracted data in order to avoid reloading the dataset each time that the user revokes it;
- finally, name and save the dataset as shown below.

Saving dataset.

As we discussed previously, you find all created datasets under **My dataset** area. You can share/unshare them by clicking on the **share** icon (have a look at the next figure). The colour of the icon changes from white to red when sharing is turned to active. A shared dataset is visible to all other users having your same role.

Note that dedicated area "**Shared Dataset**" contains all acquired datasets thanks to the sharing of other users.



Share a dataset.

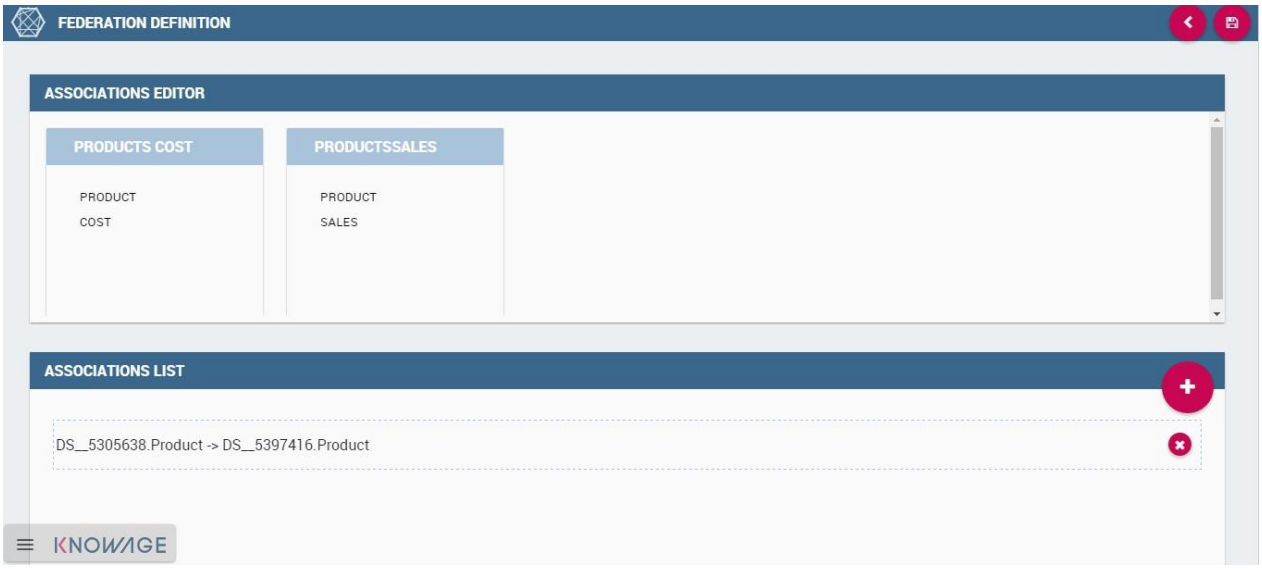
Models

Here you find the models that a technical user has built for you. You can query it using the QbE interface and create your own dataset from them.

Dataset federation

Thanks to the Data federation functionality, you can create a new dataset combining two or more datasets according to your role permissions. Let us give you an example. Suppose you have stored in a database your products information, i.e. sales, costs, promotion, etc.) and you find as open data the customers feedbacks on these products. If you create datasets on these Dataset federation resources sharing at least one column, then you can join them on the common column and improve your analysis.

Click on **Create Federation** to see all available datasets and choose the ones you want to federate. Click **Next** and choose which columns the join must be made on and click the plus icon to add it to the **Association list**. In our example in the following figure we choose Product.



Federated dataset details.

Once saved, the new federation has been created in **Federation definition** and you can find it in Federation definition. Open it by clicking the magnifier icon on the federation. In this way you open it with QbE tool. All details on how to use the QbE interface to perform free inquiries can be found in the dedicated chapter. You can create new datasets, save them and retrieve them from the **Dataset** section.

Free Inquiry

This detailed user guide is dedicated to the Qbe (acronym of Query by Example), a Free Inquiry instrument which empowers users with easy and free access to information via graphical interfaces.

Free Inquiry indicates the modus operandi of analysts and operational users that are usually seeking for business analysis that are not limited to pre-arranged lists of results. This method has a medium level of difficulty since it requires an adequate knowledge of data management and a structured organization of work.

QbE is the tool that lets you develop your free inquiry through an entirely graphical modality. Moreover, you can execute the query, check the results, export them and save the query for further use.

The material will be divided in two main sections. The first is dedicated to build queries in the Knowage Server environment, supposing that an expert user has already created

a suitable business model to analyze. In the second part, we will provide the user for the principal steps to build a proper business model through the Qbe designer available in Knowage Meta.

My first Query by Example

QbE (i.e., Query by Example) allows you to query (a subset of) a database through a high-level representation of the entities and relations. Its main characteristics are:

- it has a rich end user GUI;
- it allows to select attributes and set filters;
- it does not require any knowledge of data structures;
- it requires a semantic knowledge of data;
- it is useful every time the free inquiry on data is more important than their graphical layout;
- it leaves the management of results free;
- it supports export capabilities;
- it allows the repeatable execution of inquiries;
- it works on a data domain with limitations.

Building a QbE query does not require any technical knowledge, but data domain knowledge: technical aspects, such as creating filters, aggregation and ordering criteria, are managed by a user-friendly graphical interface.

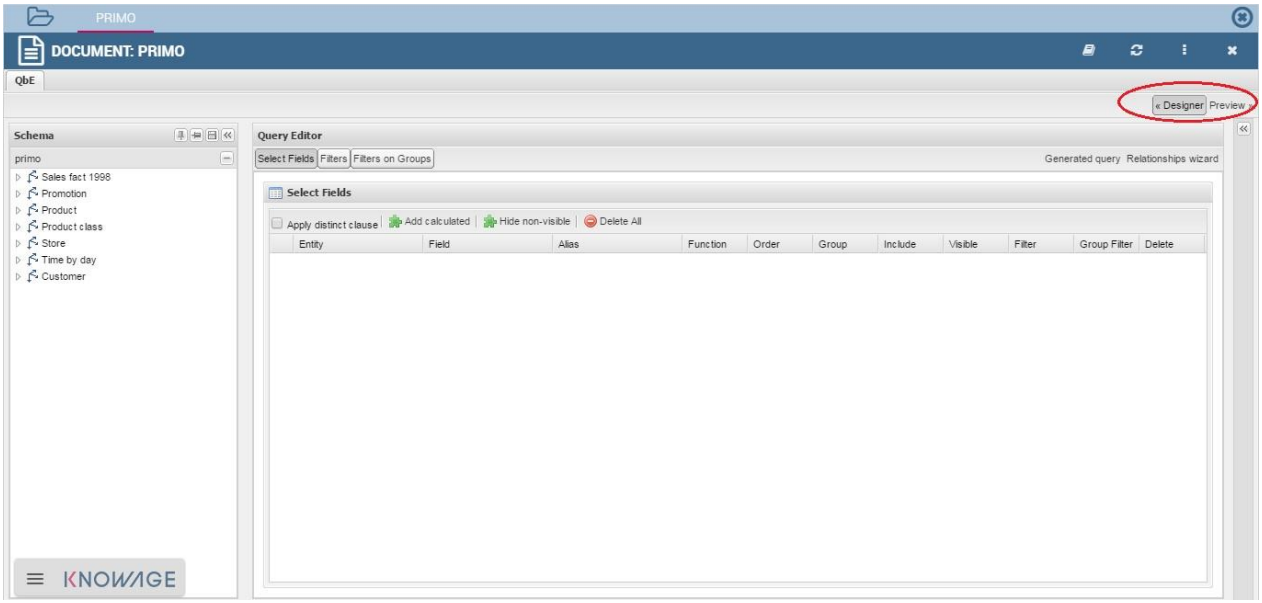
Let’s suppose that an administrator has built a business model and, consequently, released it on Knowage Server. This permits the user to access the model, query the available entities and save results as a dataset, usable later in other Knowage documents, such as cockpits.

In the following we discuss each step-in detail, showing basic and advanced functionalities of the **QbE Editor**.

Query design and execution

To open the QbE editor, access the **Models** section, available in the end user’s **Workspace**. Then, simply click on the model icon to reach the QbE graphical interface.

In this paragraph we show how to build a simple query with the QbE editor.



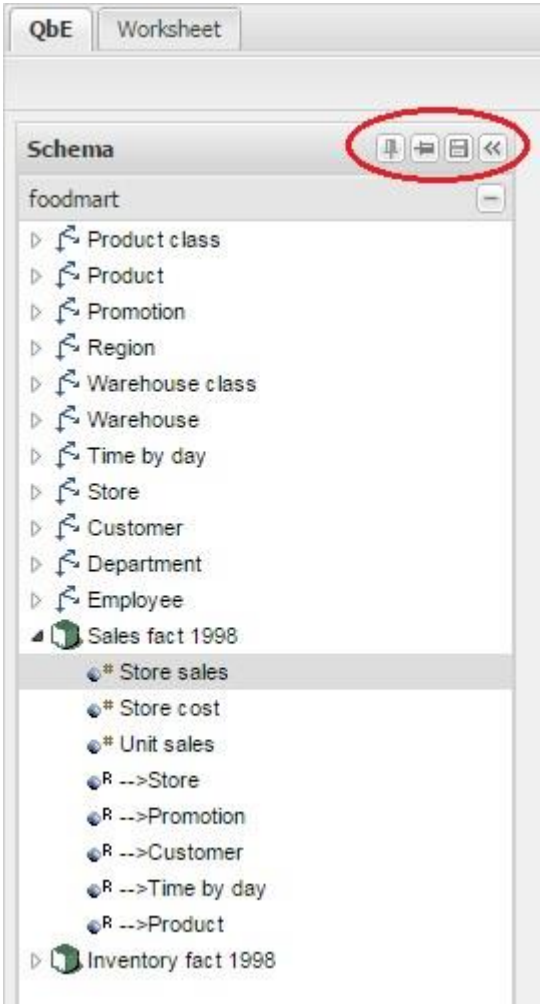
QbE editor.

As shown in Figure 9.1 the window of the QbE editor contains the **Query designer**. In next sections we explain in detail all the areas of the **Query Designer**, the **Datamart Schema** tab, the query editor and a hidden tab dedicated to the management of queries, subqueries and parameters catalogue.

Datamart Schema

Starting from the left side, the first Panel shows the searchable logical schema and the list of entities that can be queried to generate the query. Entities and relationships are represented in a tree structure, with user-defined names. Fields can be dragged from here and dropped onto the editor area.

In the top right corner of the panel you can find a small toolbar to configure the panel (e.g., expand, reduce) and to save changes made to the model (i.e. **Calculated Field** or **Range**), as shown below.



Datamart schema toolbar.

There are two types of entities: *facts*, represented by a cube symbol.(i.e., the Sales fact 1998 entity) and *dimensions*, represented by a three-arrows symbol (i.e., the Product entity).

Each single entity is composed of a title, some attributes or measures and relationships with other entities. By exploding the content of an entity (i.e. Sales fact 1998 as in figure above), you may encounter the following elements:

- **measure**: it refers to fields associated with numeric data (e.g. number of sold items);
- **attribute**: it refers to fields that can be associated to a category (e.g. product category);
- **relation**: it refers to relationships or connections between two entities (e.g. relationship between the product sales fact and the product dimension).

Right clicking on an item in the tree, the contextual menu opens and shows some additional features:

- **Add calculated field**: to add a field that can be obtained via simple expressions combining existing fields and operators. Clicking on the contextual menu item, the wizard opens. Here you can combine fields with arithmetic and date functions. When you create a calculated field, you can add it to the model by clicking the **Save** button located in the top right corner of the panel. In addition, they can be used in queries. Calculated fields may also be managed by expert users via advanced functionalities, which will be described at the end of this section.
- **Edit field**: to rename a field.
- **Add/Edit Range**: to add or manage a range of values of the selected attribute (details are provided below).
- **Remove calculated field**: to remove a calculated field that was added before.

Let us see more in detail how to add calculated fields and ranges.

Calculated fields management

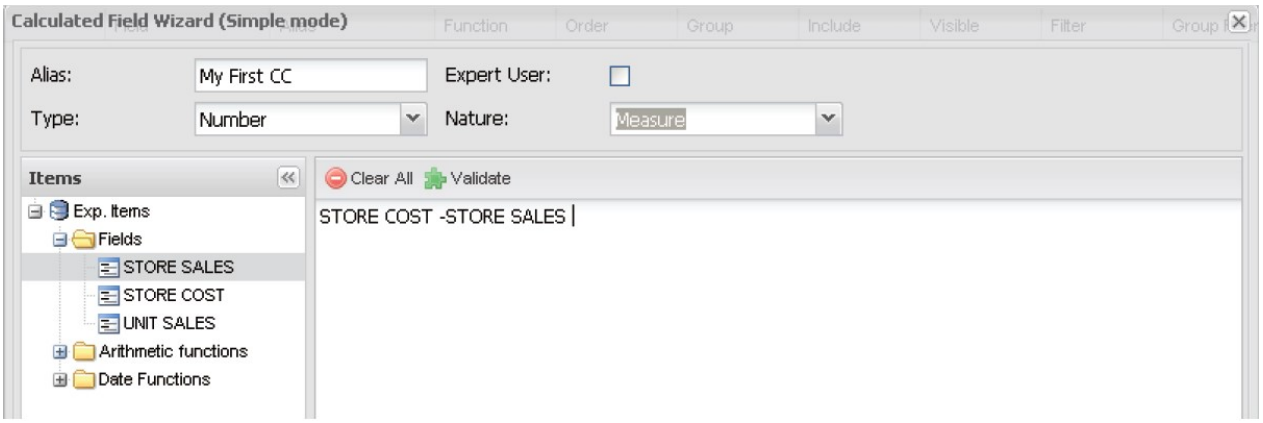
You can create new calculated fields either inside a query or in the schema panel. Calculated fields defined in this second way can be saved for future use.

In order to define a new calculated field in the model, right click on the chosen entity and select **Add calculated field**. The wizard offers an editor in which you can define the calculated field.

To build a calculated field, you shall define:

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

An example is provided below.



Calculated field wizard.

There are two types of calculated fields that you can add to the QbE query: *standard* and *expert*. The *standard* ones are SQL expressions that are injected into the query. With the *expert* calculated fields (you should mark the **Expert user** box in the calculated fields wizard) you can build Groovy scripts, show images, add links. This second type of calculated field is computed after the query has been executed.

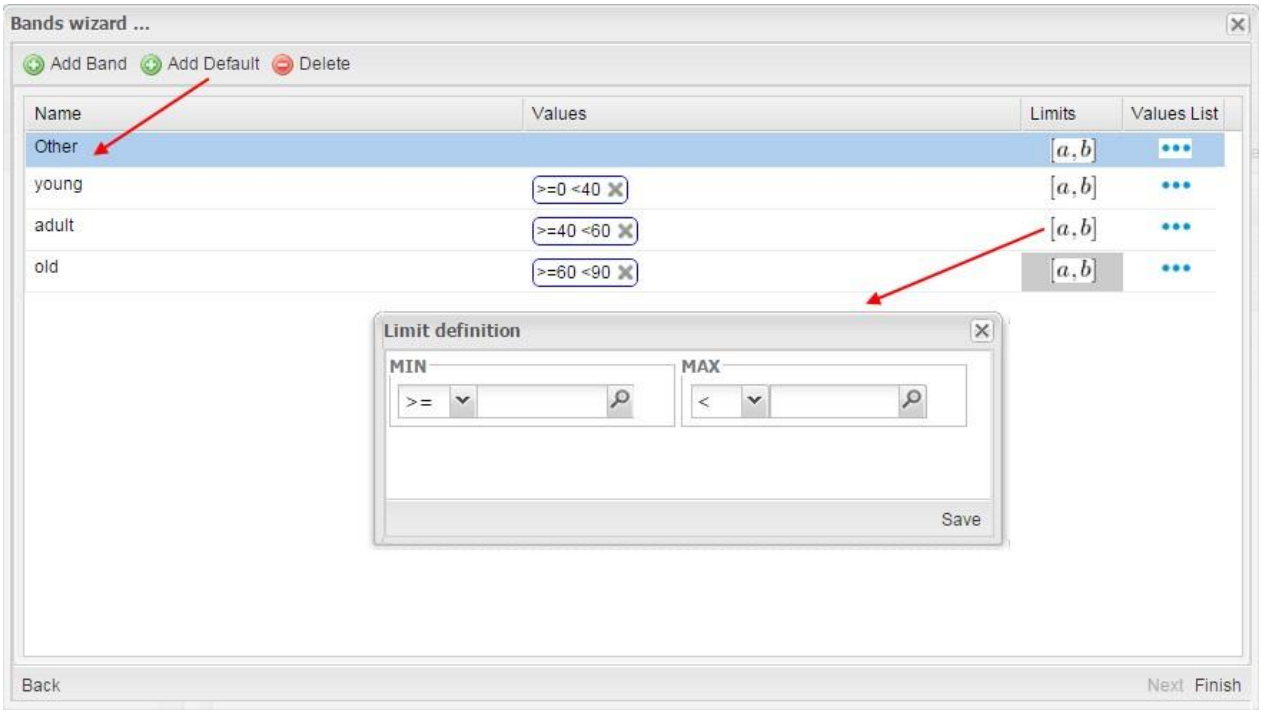
Range management

It frequently happens that attributes of entities in a model have several different values. However, for the purpose of analysing data, it is often more useful to group those values into categories.

For example, let's consider the customers' age: often analysts do not aim to know the exact age of customers, but rather if they belong to a certain age range, e.g., young, adult and elderly. For this and similar cases, the **QbE Engine** can define and manage ranges in queries. To create a new range for an attribute you can:

- right click a field and click on **Add Range** in the contextual menu;
- right click on an entity (a cube or a dimension), click on **Add Range** , then in the wizard choose a field or define a calculated expression, give a name to it and click on **Next**.

Both operations open the band creation wizard. Here click on **Add Band** to add a new instance and set the corresponding values and labels, as shown below.



Range instance creation.

You can set your band values by clicking on blue points under the Values List column to add them one by one, or by clicking on *[a,b]* under the **Limits** column to give only the end points of the interval. Then you can name your band by double clicking on the related field under the Name. Repeat the procedure to generate all the bands you need.

Finally, you can click on **Add Default**: this creates a new category called **Others**, which groups all values not belonging to already defined range intervals. At this point, click on **Finish**. The range appears as a node in the schema panel on the left. If you want to edit the range, click on **Edit Range**.

Query Editor

The central panel provides a query editor, including three different tabs:

- **Select Fields**, containing the list of columns to be returned by the query;
- **Filters**, containing filtering conditions on fields values;
- **Filters on Groups**, containing filtering conditions on aggregated measures.

Elements from the Datamart schema on the left can be dragged and dropped onto the query editor tabs. If a whole entity is selected, all its attributes are dropped into the

editor. Alternatively, you can drag and drop single entity fields, as said before. To remove an attribute from the query editor, just click on the dedicated icon in the delete column or select the corresponding row and press **Delete** on your keyboard.

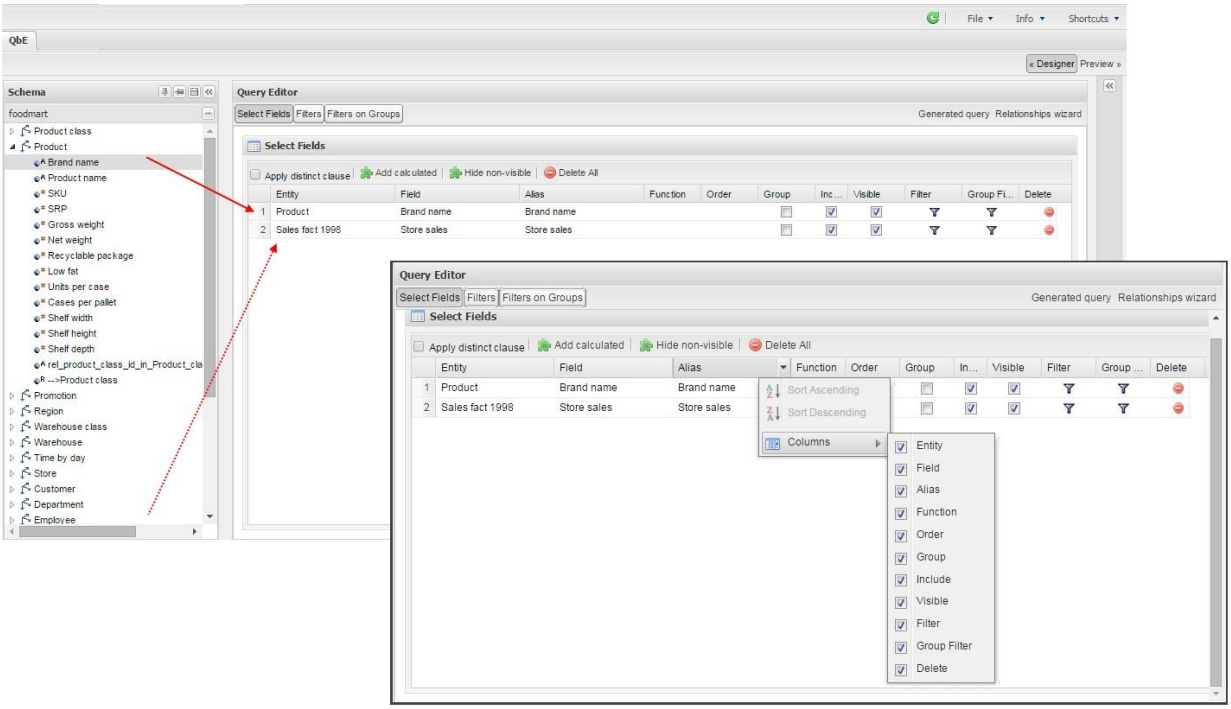
The expert user can visualize the query matching his selections by clicking on the **Generated query** button at the top right corner of the panel. This way it is possible to check the SQL generated by the graphical interface.

Let us now see in detail the three functionalities, listed above, which split the query editor area in different sections.

Select Fields

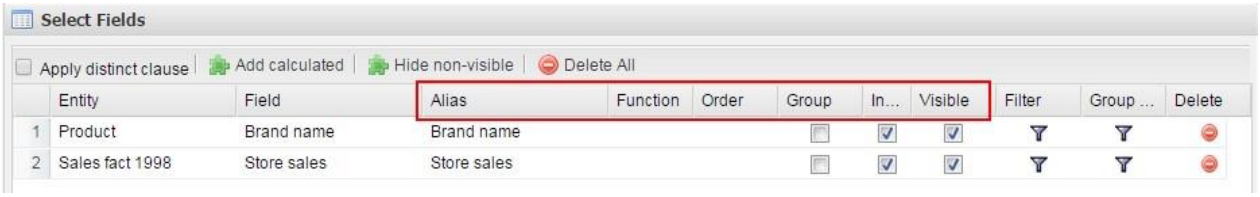
This tab contains the list of columns to be returned by the query. To add a new attribute in this section, just click on a field in the schema panel tree or drag and drop it onto the query editor.

This panel is structured as a table: rows contain the attributes selected from the Datamart schema, while columns include applicable functions as shown below.



Select fields interface.

For each dropped item, the first two columns Entity and Field show the entity and the related attribute field respectively, and they are not editable.

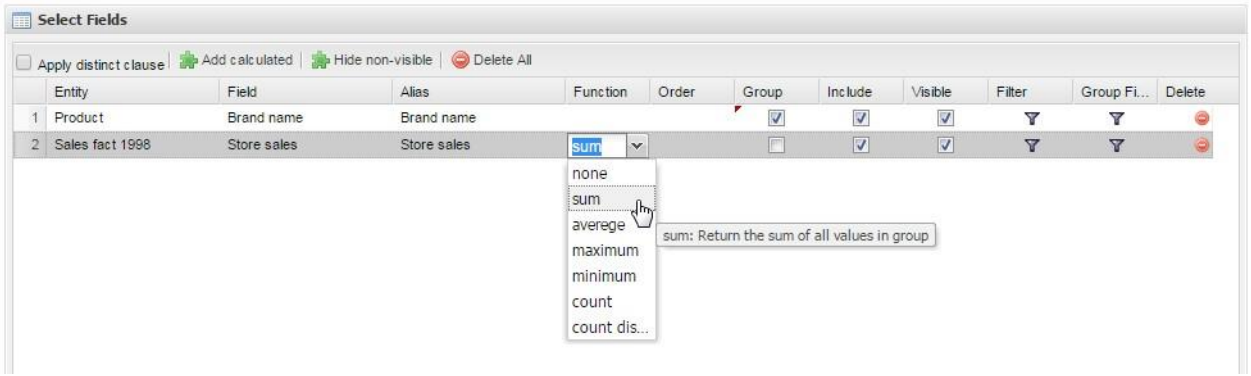


Select Fields panel options.

With the other columns it is possible to:

- **Alias**: define aliases for fields: those aliases are shown as column headers in the result table;
- **Function**: in case of aggregation, define the aggregation function (e.g., **SUM**, **AVERAGE**, …) on the non-grouped items;
- **Order**: define a sorting criteria: double click on the **Order** column to set the ordering criteria;
- **Group**: in case of aggregations, define the attribute that you want to group on (if you know SQL syntax, these attributes are the ones you should place in the GROUP BY clause);
- **Include**: indicate the column(s) to be included in the result (please notice that non-included attributes will not be returned by the query, but can be used in it, e.g. to apply grouping criteria);
- **Visible**: indicate whether a column shall be visible in the result (hidden attributes are used and returned by the generated query, but are not shown in the result table);
- **Filter**: add a filter criteria: clicking on this filter icon redirects you to the **Filters** tab;
- **Group Filter**: add a filter on groups: clicking on this filter icon redirects you to the **Filters on Groups** tab;

Pay attention to grouping options: if you want to define an aggregation function on a field (like, for instance, the **COUNT** of the sold items), you shall tick the Group checkbox for all the other fields dragged in the **Select Filters** panel without an aggregation function defined, otherwise you will get an SQL exception. The possible grouping functions are shown in the following figure.



Aggregation functions.

When you drag attributes belonging to entities that are linked through a relationship path, the QbE automatically resolves relationships between attributes (implicit join).

Moreover, multiple relationships may occur among entities. A typical example concerns dates. Suppose you have two relationships between the **Order** fact table and the **Time** dimension table: the first links the order_date column of the first table to the *time_id* column of the latter, while the second relationship joins the *shipping_date* column to the *time_id column*.

In this case, when dragging fields from both the **Order** entity and the **Time** entity you may want to specify which relationship will join the two tables: for instance, you may want to know the total number of orders according to the ordering month, the shipping month or for both. In all these situations, you can set the relationship to be used by clicking the **Relationships wizard** button at the top right corner of the panel. A pop-up window opens where you can define the path to be used. Please refer to Multiple relationships section for all details regarding the disambiguation of relationships.

The select sub-section has a toolbar with additional functionalities summarized in Table below.

Select fields toolbar options

Button	Description
Apply distinct clause	Remove duplicated rows from results, if any
Hide non visible	Hide fields set as non visible in query results

Select fields toolbar options

Button	Description
Add calculated	Add a calculated field to the query
Delete all	Remove all rows from select area

Filters

The **Filters** panel allows you to define filter criteria (WHERE clause). Similarly, to the select area, filters are structured as a table: here rows contain filters, while columns represent the elements of the filter.

There are three ways to create a filter:

Delete all Remove all rows from the select area

- drag an attribute from the Datamart schema to the **Filters** panel;
- click the filter symbol on the row of an attribute in the **Select Fields** panel;
- click the **New** button in the **Filters** panel.

To remove a filter from the query editor, select the left side of the row (multiple rows can be selected as well) and press the **Delete** button on your keyboard.

Filters are expressions of type:

Left operand + Operator + Right operand.

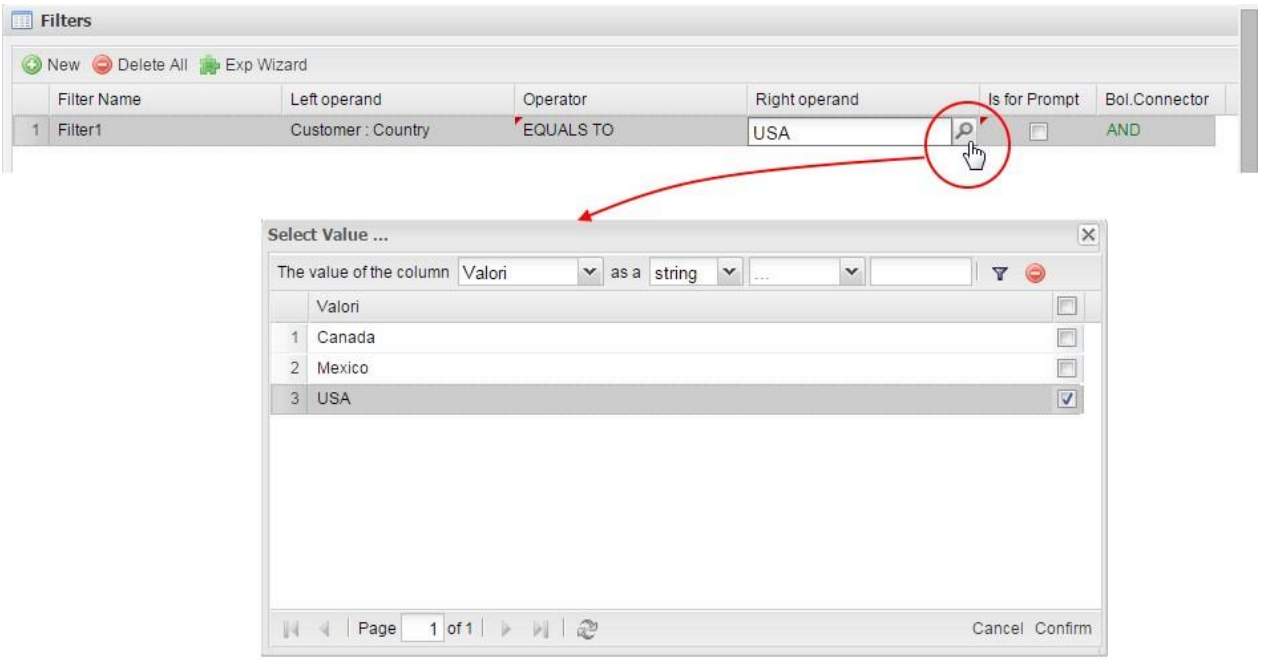
Once you have selected the left operand, you can configure the filter by using the proper setting values on columns. In particular:

- the **Filter Name** column contains the (editable) name of the filter while the Filter Description column contains an editable description;
- the **Left operand, Operator, Right operand** columns allow you to define filters according to the syntax defined above. Double clicking in the Right operand column, a lookup function is activated to facilitate selection of values;

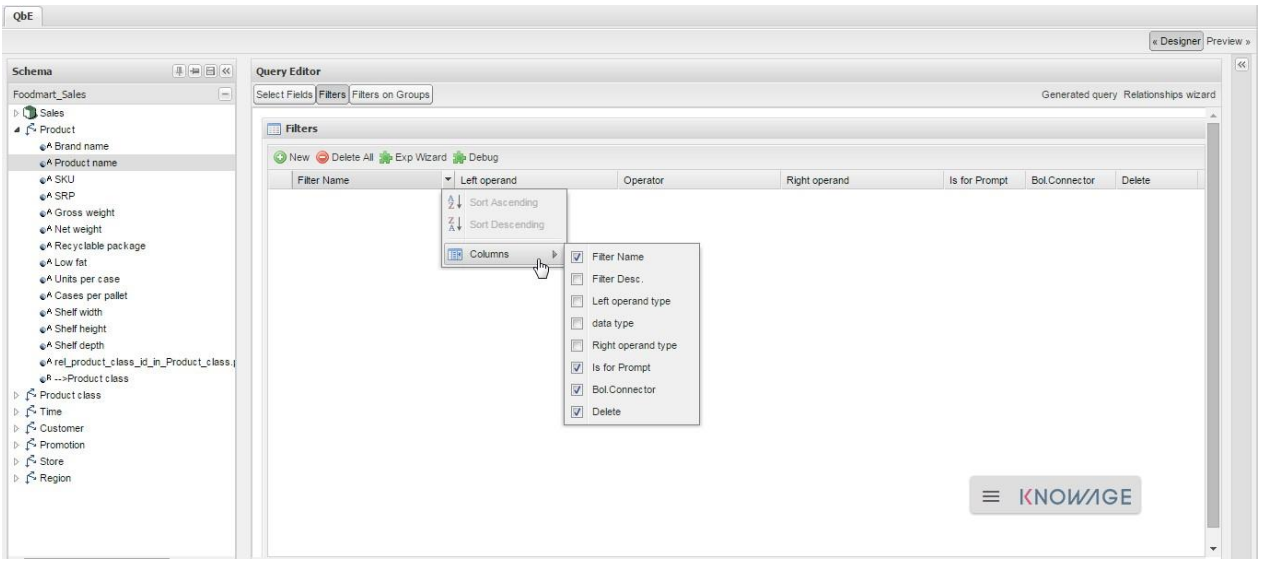
- the **LeftOperandType** and **RightOperandType** columns define the types of operands;
- the **Is for Prompt** column should be checked in order to insert dynamically the value for the parameters at execution time;
- the **Boolean Connector** column shall be used to control the evaluation order of the different filters' conditions;

Not all available features of the editor panel are visible by default. To customize the editor appearance, double click on the arrow located on each column header and select **Columns**.

Here you can decide which columns you want to appear in the editor.



Filter lookup for right operand selection.



Filter editor customization.

Note that more complex combinations of filters can be defined using the Expression Wizard, which your ca find selecting the **Exp Wizard** icon.

In the following table the possible types of filters in the QbE are summarized. The use of subqueries in filters is explained later in *Advanced QbE functionalities* paragraph.

Possible combinations of filters in the QbE.

Filter type	Left operand	Operator	Right operand	Example
Basic	Entity. attribute	Any	value	Prod.family = ‘Food’
Basic	Entity. attribute	Any	Entity, attribute	Sales.sales > Sales.cost
Parametric	Entity. attribute	Any	[parameter]	Prod.family = [p_family]
Dynamic	Entity. attribute	Any	prompt	Prod.family = ?

Possible combinations of filters in the QbE.

Filter type	Left operand	Operator	Right operand	Example
Value list from subquery	Entity. attribute	In /not in	subquery	Sales.customer in subquery
Single value from subquery	subquery	< = >	value	Subquery > 0

Filters on Groups

By moving to the **Filters on Group** tab it is possible to define filters on aggregated measures.

Filters on groups are expressions of type:

Aggr. function + Left operand + Operator + [Aggr. function] + Right operand,

where the second [Aggr. function] is in this case optional. Example expressions could be, for instance, the filter “sum(sales) > 10000” or “sum(sales) > sum(costs)”.

Once you have selected the left operand, you can configure the filter using the proper setting values on columns. Columns are the same as those of the **Filters** tab, that is the ones just described in the previous section. There are, however, additional columns related to grouping functions. In particular, the two columns named **Function**, define he aggregation function to use on the left, or right, operand.

Query Preview

Once you are satisfied with your query or if you want to check the results, you can see the returned data by clicking the **Preview** button located in the top right corner of the panel. From there, you can go back to the **Designer** tab to modify the definition of the query or switch directly to the **Worksheet** designer to start building your graphical representation of the extracted data.

In case you have started the QbE editor directly from a model (that is, you have clicked on a model icon in the **My Data** > **Models** section) from here you can also click the **Save** button located in the top right corner of the page to save your query as a new dataset, reachable later from the **My Data**> **Dataset** section. Please note that this operation saves the *definition* of your query and not the snapshot of the resulting data. This means that every time you re-execute the saved dataset, a query on the database is performed to recover the updated data.

We highlight that when the save button is selected, a pop-up shows asking you to fill in the details, split in three tabs:

- **Generic**, in this tab you set basic information for your dataset like its **Label**, **Name**, **Description** and **Scope**. The available values for the scope are **Public** and **Private**. If you choose **Public**, the dataset will be visible to all other users otherwise it won't.
- **Persistence**, you have the chance to persist your dataset, i.e., to write it on the default database. Making a dataset persistent may be useful in case dataset calculation takes a considerable amount of time. Instead of recalculating the dataset each time the documents using it are executed, the dataset is calculated once and then retrieved from a table to improve performance. You can also decide to schedule the persistence operation: this means that the data stored will be update according to the frequency defined in the **scheduling** options.

Choose your scheduling option and save the dataset. Now the table where your data are stored will be persisted according to the settings provided.

- **Metadata** It recaps the metadata associated to the fields involved in your query.

Advanced QbE functionalities

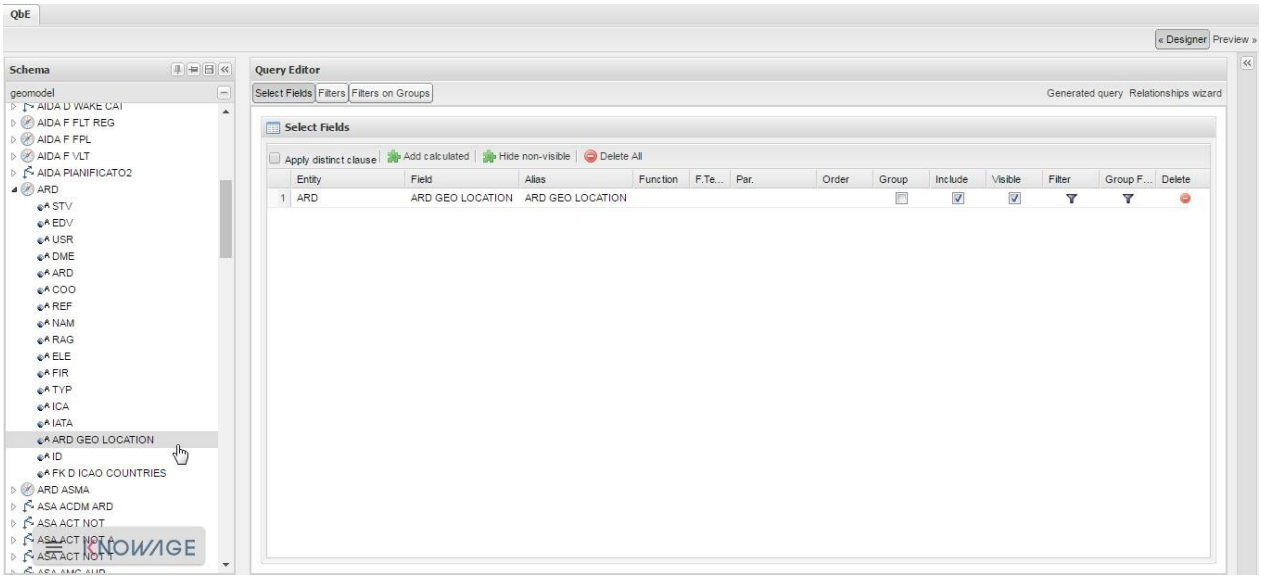
In this section we focus on advanced features, which can be comfortably managed by more expert users.

Spatial fields usage

The Qbe engine supports spatial queries through a set of operators (that return true or false) or a set of functions (these usually return a measure). This feature is although

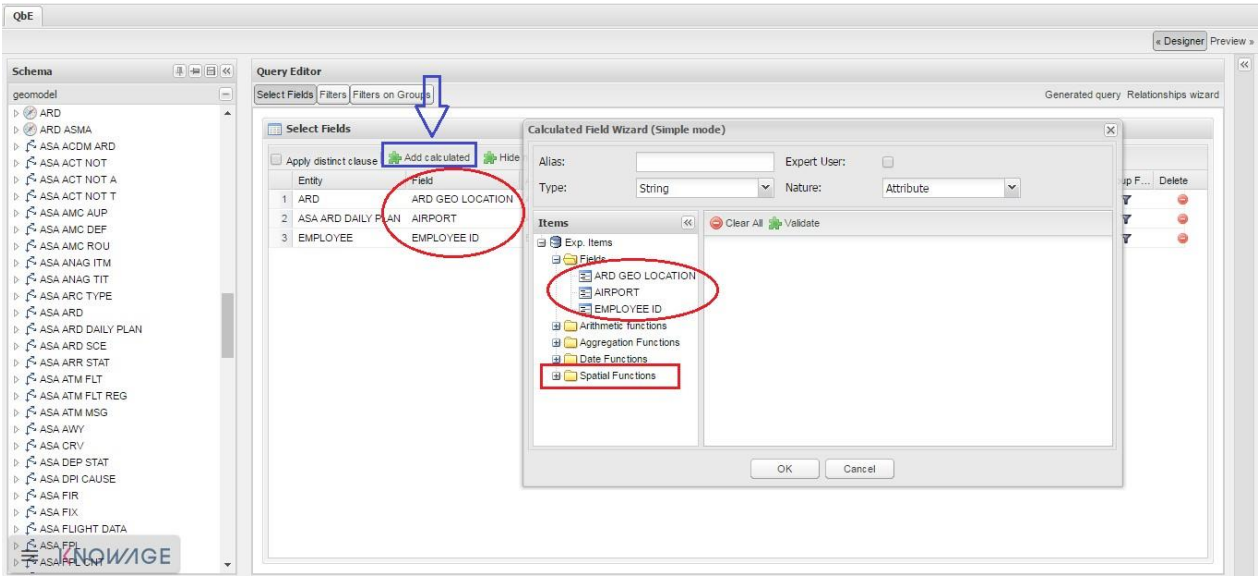
available only when the Location Intelligence (LI) license is possessed and when data are stored in Oracle 12c database. It also fundamental that the Business Model must be tagged as geographical model. You can refer to Meta Web Section to have details on how to set the geographical option using Knowage Meta.

We suppose that we have a BM with geographical dimensions enabled (by a technical user). In this case the dimensions which has spatial fields are marked with the compass icon . Once the spatial dimension is expanded the fields are listed. Here there is no tracking symbol to distinguish between geographical attributes and the “normal” one. Therefore, it is very important that the user is previously informed of which fields has geometrical properties.



QbE spatial dimensions.

After a first selection of fields, it is possible to add calculated fields. Click on the **Add calculated** option available on the query editor area as shown by the blue arrow in figure below. Note that a wizard opens: you can use this editor to insert a new field obtained through a finite sequence of operation on the selected fields. The circles of the next figure underline that the fields on which you can operate are the one previously selected via drag and drop (or by a simple click on the field).



Calculated field wizard with spatial filters.

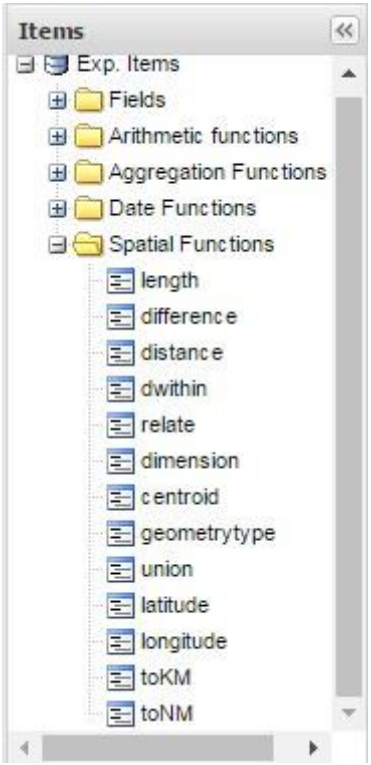
In addition, note that the **Items** panel provides all the applicable functions sorted by categories:

- arithmetic functions,
- aggregation functions,
- date functions,
- spatial functions.

Consider the Oracle function definition

It is important to refer to Oracle Documentation to know the arguments, in terms of type and number, of each function to assure the right functioning and do not occur in errors while running the Qbe document.

The latter are available only in the presence of a geographical Business Model and *must* be properly applied to spatial attributes or measures. Figure below shows the list of the available spatial functions while next table helps you to use them properly, supplying the corresponding Oracle function name and a link to grab more specific information about usage, number of arguments, type and output.



Spatial function list.

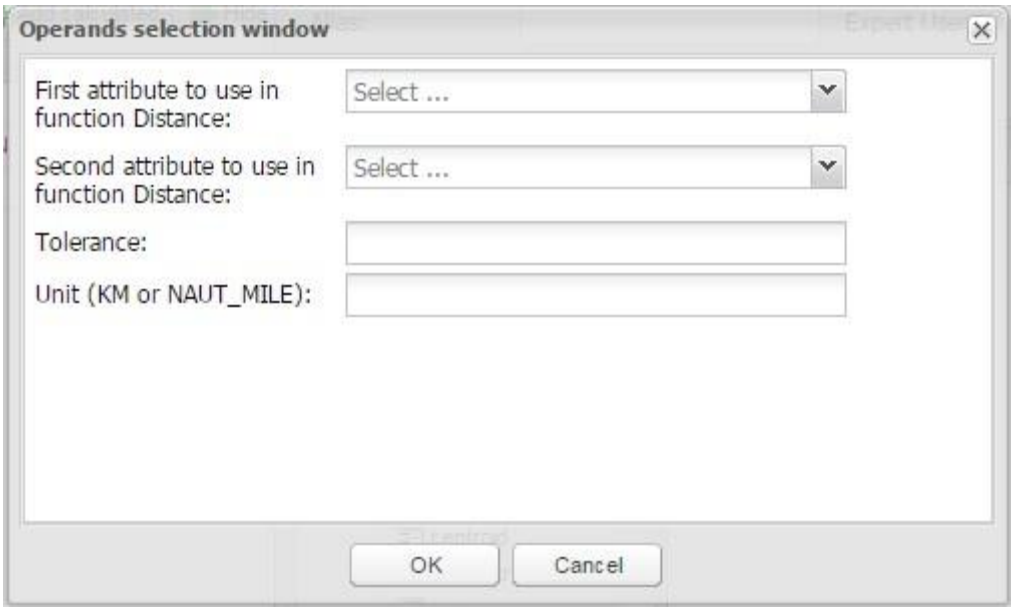
Link to Oracle spatial functions.

Function Name	Oracle Function	Link to Oracle web pages
distance	SDO_GEOM.SDO_ DISTANCE	https://docs.oracle. com/cd/B19306_01/appd ev. 102/b14255/sdo_objgeo m.htm#i857957
within	SDO_WITHIN_DIS TANCE	https://docs.oracle. com/cd/B19306_01/appd ev. 102/b14255/sdo_operat .htm#i77653
dimension	GET_DIMS	https://docs.oracle. com/cd/B10501_01/appd ev. 920/a96630/sdo_meth.h tm#BABDEBJA
difference	SDO_GEOM.SDO_ DIFFEREN CE	https://docs.oracle. com/cd/B19306_01/appd ev. 102/b14255/sdo_objgeo m.htm#i857512
centroid	SDO_GEOM.SDO_ CENTROID	https://docs.oracle. com/cd/B19306_01/appd ev. 102/b14255/sdo_objgeo m.htm#i860848
Geometry type	GET_GTYPE	https://docs.oracle. com/cd/B10501_01/appd ev. 920/a96630/sdo_meth.h tm#i866821

Link to Oracle spatial functions.

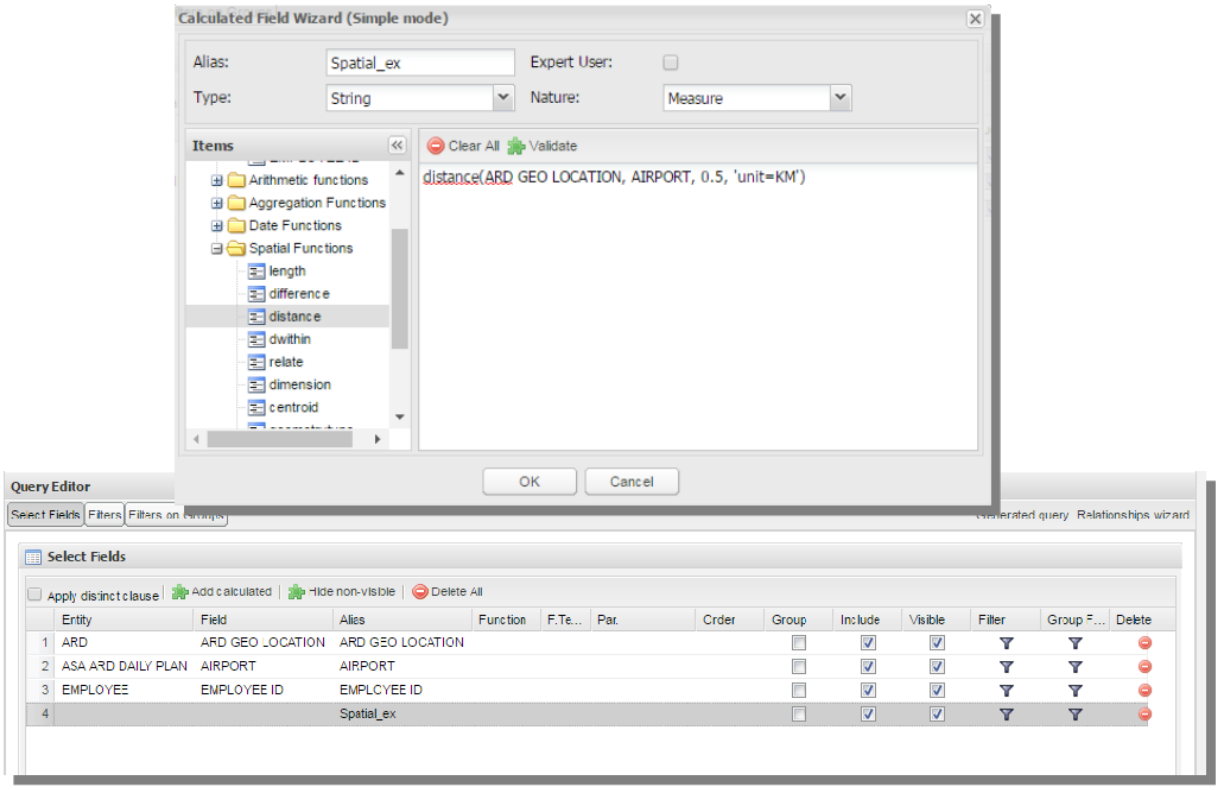
Function Name	Oracle Function	Link to Oracle web pages
union	SDO_GEOM.SDO_UNION	https://docs.oracle. com/cd/B19306_01/appd ev. 102/b14255/sdo_objgeo m.htm#i857624
length	SDO_GEOM.SDO_LENGTH	https://docs.oracle. com/cd/B19306_01/appd ev. 102/b14255/sdo_objgeo m.htm#i856257
relate	SDO_GEOM.RELATE	https://docs.oracle. com/cd/B19306_01/appd ev. m.htm#BGHCDIDG

To apply one function, click on the function name and the “Operands selection window” wizard opens. Figure below shows an example for the function “Distance”. Fill in all boxes since all fields are mandatory.



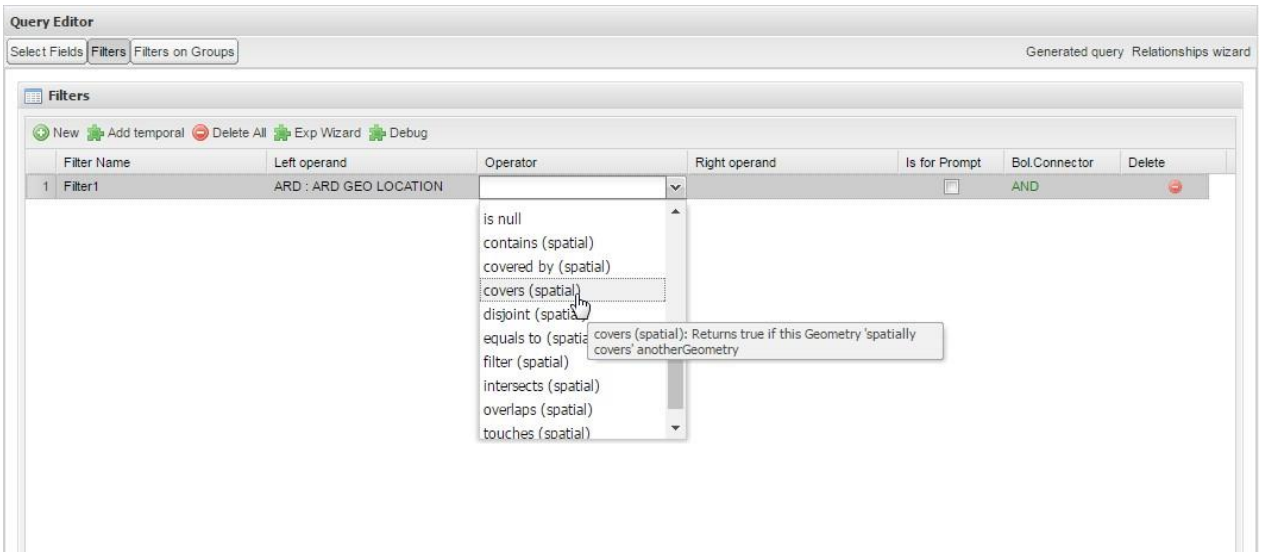
Operands selection window.

Finally, you can use spatial function to add a calculated field, as shown below.



Example of added calculated field using a spatial function.

As well as calculated fields it is possible to filter on spatial fields using specific geometric operators. Once again, we report in Figure below the available geometric operator (you can find them scrolling the panel to the bottom) and report the link to the Oracle web pages in the next table.



Spatial filters.

See the table below:

Link to Oracle filter functions.

Function Name	Oracle Function	Link to Oracle web pages
touches	SDO_TOUCH	https://docs.oracle. com/cd/B19306_01/appd ev. 102/b14255/sdo_operat .htm#BGEHHIGF
filter	SDO_FILTER	https://docs.oracle. com/cd/B19306_01/appd ev. 102/b14255/sdo_operat .htm#BJAFBCFC
contains	SDO_CONTAI NS	https://docs.oracle. com/cd/B19306_01/appd ev. 102/b14255/sdo_operat .htm#BGEHCFDH
covered by	SDO_COVERE DBY	https://docs.oracle. com/cd/B19306_01/appd ev. 102/b14255/sdo_operat .htm#BGEHEAEJ
inside	SDO_INSIDE	https://docs.oracle. com/cd/B19306_01/appd ev. 102/b14255/sdo_operat .htm#BGEFABDH
covers	SDO_COVERS	https://docs.oracle. com/cd/B19306_01/appd ev. 102/b14255/sdo_operat .htm#BGEGIJFB
overlaps	SDO_OVERL APS	https://docs.oracle. com/cd/B19306_01/appd ev. 102/b14255/sdo_operat .htm#BGEDACIF

Link to Oracle filter functions.

Function Name	Oracle Function	Link to Oracle web pages
equals to	SDO_EQUAL	https://docs.oracle. com/cd/B19306_01/appd ev. 102/b14255/sdo_operat .htm#BGEBCEJE
intersects	SDO_ANYINT ERACT	https://docs.oracle. com/cd/B19306_01/appd ev. 102/b14255/sdo_operat .htm#BGEJHDGD

Temporal dimension

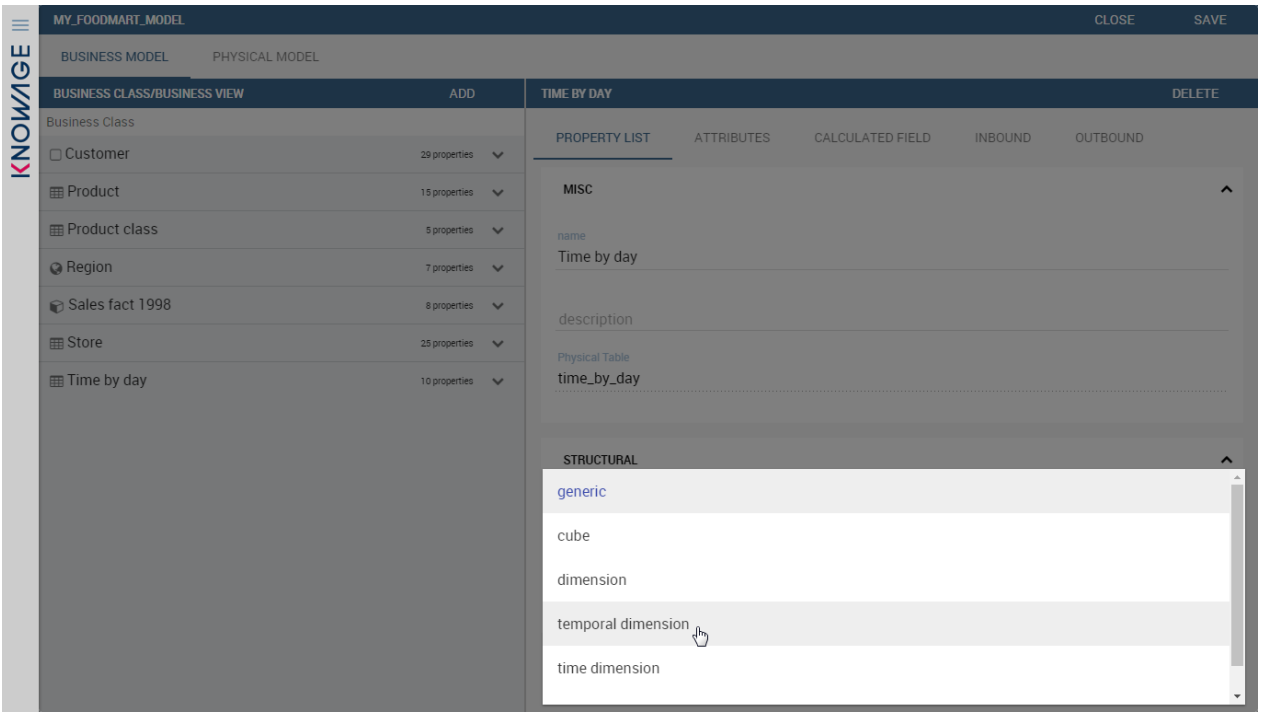
The Qbe engine on Knowage Server is endowed with some temporal functionalities that allow the final user to easily preform queries based on time.

We highlight that the new features are available only if the model has at least one temporal dimension. The latter must be defined while creating the model using Knowage Meta.

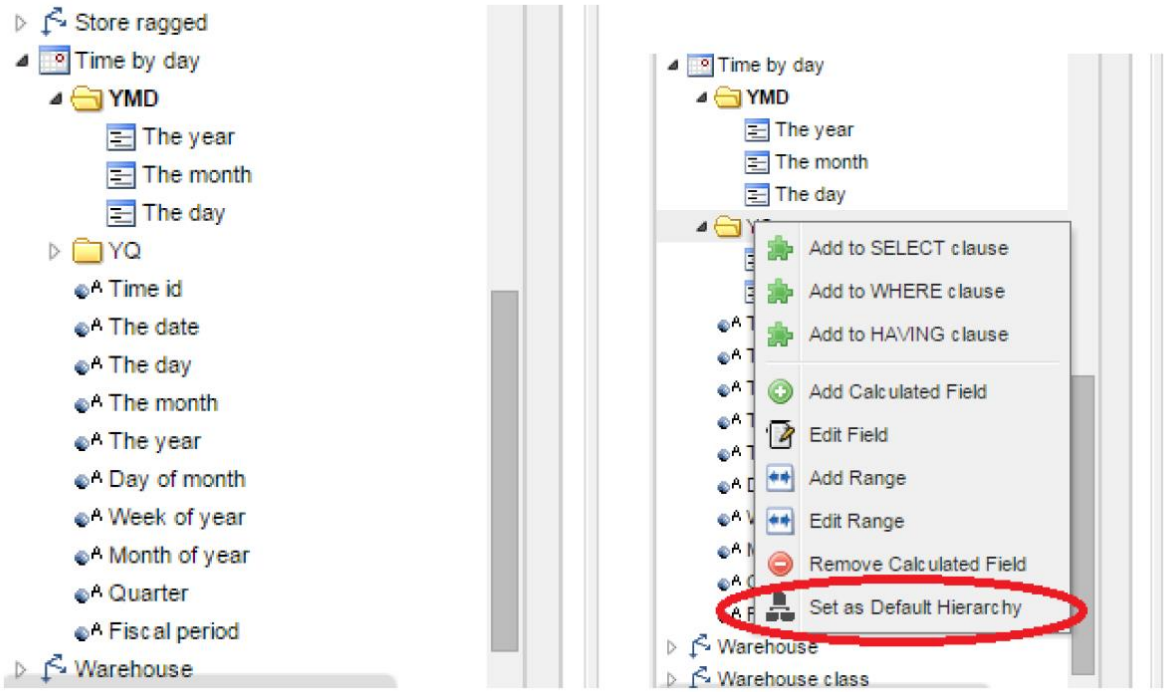
Define first the temporal dimension on Knowage Meta

To have a temporal dimension that can be used in the Qbe interface an expert user must enable it first on the model using Knowage Meta. Use the **property view** to set/change the type of the dimension as shown in the following figure. Refer to *Meta Web* chapter to learn how to use Knowage Meta.

The temporal dimension can have one or more hierarchies. Only one of these can stay active and that is the one used by the query code. Figure below shows that a temporal dimension can have one or more hierarchies. In the case of more hierarchies the user can see which the one set by default is just exploring the dimension: the bold highlighted hierarchy is the primary. On the other hand, the user can change the default choice by right-clicking on the target dimension hierarchy and selecting “*Set as Default Hierarchy*”.

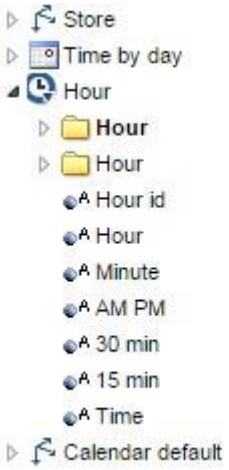


Temporal dimension definition on Meta.



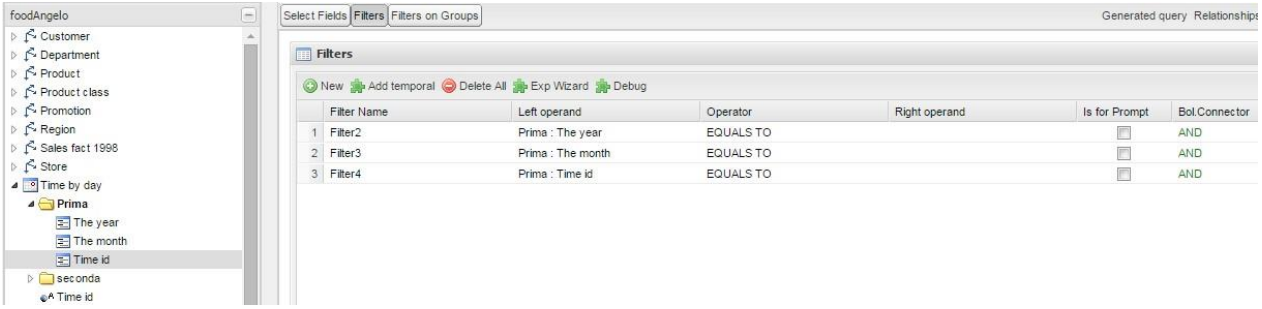
Temporal hierarchy visualization (Left). Changing hierarchies (Right).

Furthermore, there is the possibility to set a “time” dimension as Figure below displays.



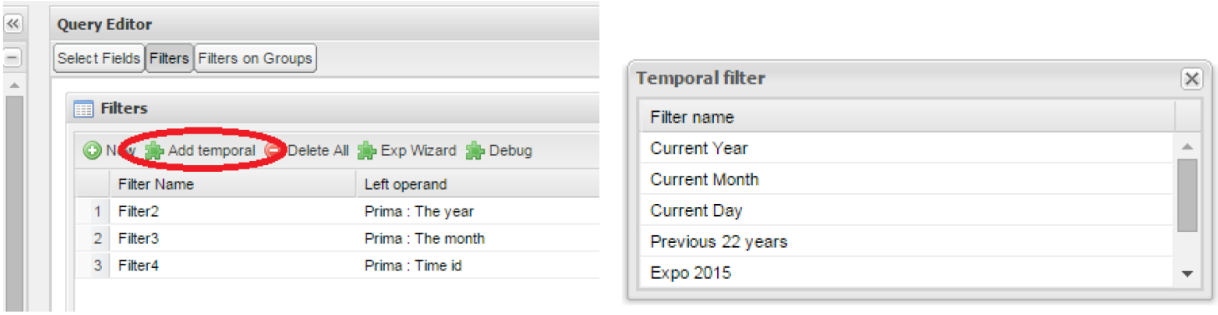
Time dimension.

The user can use the elements of each dimension as attributes in the “Select” instance. Note that if one drags and drops of element to be used as a filter also its parent nodes will be brought too. The following figure exhibits one example. Remember to assign a value to each parent node before you run the query.



Filter on an element means to filter also on its parent nodes.

Moreover, selecting the filters tab, you can use specific filters clicking on the button “Add Temporal” as shown in figure below (Left). The action opens the pop up displayed in next figure (Right).

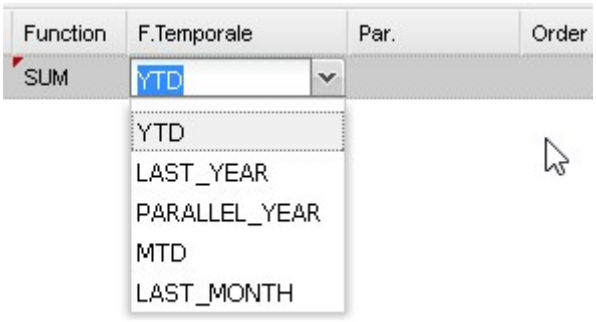


Add temporal filters (Left). List of available elements (Right).

In the list of available elements is made up of:

- filters defined by the admin through the Timespan GUI;
- system filters manageable through a table;
- the element “Current year”;
- the element “Current month”;
- the element “Current day”;
- the element “Last Period” for which you must indicate the number of years.

Inside the section “Select” you can use the temporal operators directly on attributes.



Apply operators directly on attributes.

For each function there is the possibility to assign a value to a parameter that indicates how long the function will act. We now describe the working principles of temporal functions.

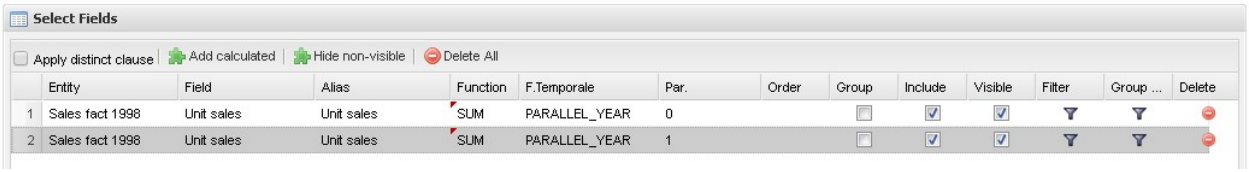
The PARALLEL_YEAR function.¶
This function allows to manage and study measures on parallel periods. For example, if one wants to analyze the product sales of the current year and, at the same time, those of the previous year. The following are some possible use cases:

- no temporal filter is set, and the temporal functions are applied directly on measures. In this case the current year is taken as default value. When the functions are applied on measures the user must apply them on ALL measures in order to have a coherent result.

1. In the case the user wants the sum of a measure relative to current year, he/she must drag and drop the measure in the “select fields” panel and launch the temporal function PARALLEL_YEAR passing 0 as value. See Figure below as example.

PARALLEL_YEAR example: sum of a measure referred to a specific time year.

2. In the case the user wants to compare the sales of 2016 with those of the previous year. He/she must drag twice the measure inside the “select fields” panel and indicate the temporal function “PARALLEL_YEAR” using 0 and 1 as value parameters.



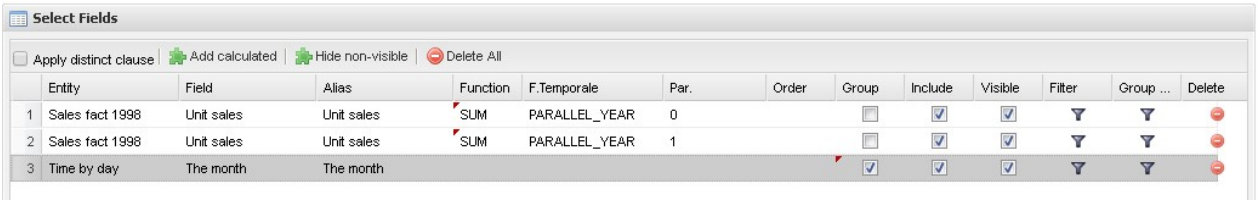
PARALLEL_YEAR example: comparing data with different time interval.

- Suppose now that the analysis requires to compare the unit sold from January to March of the current year with that of the same time interval of the previous one. In this instance the user must set the temporal filter which will be the point of reference.

PARALLEL_YEAR example: setting the temporal filter.

Remember that the temporal filter uses the “IN” operator.

- In the case one wants to compare the sales per month of the current year with the ones of the parallel year, the user should add the month field in the select clause (picking it up from the used temporal hierarchy) and group by it.



	Entity	Field	Alias	Function	F.Temporale	Par.	Order	Group	Include	Visible	Filter	Group ...	Delete
1	Sales fact 1998	Unit sales	Unit sales	SUM	PARALLEL_YEAR	0							
2	Sales fact 1998	Unit sales	Unit sales	SUM	PARALLEL_YEAR	1							
3	Time by day	The month	The month										

Comparing results with those of the parallel year.

An example of data visualization is given in the two figures below.

	Unit sales_PARALLEL_YEAR_0	Unit sales_PARALLEL_YEAR_1	The month
1	23,156.50	46,313.00	January
2	22,215.50	44,431.00	February
3	23,167.00	46,334.00	March
4	22,524.50	45,049.00	April
5	22,542.50	45,085.00	May
6	0.00	45,611.00	June
7	0.00	46,671.00	July
8	0.00	44,777.00	August
9	0.00	47,964.00	September
10	0.00	43,945.00	October
11	0.00	53,807.00	November

Comparing results with those of two parallel years.

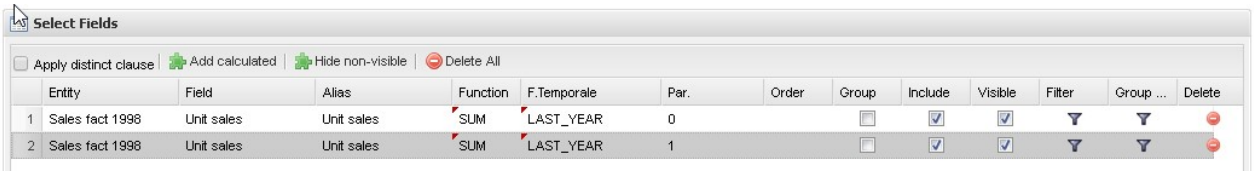
Unit sales_PARALLEL_YEAR_0	Unit sales_PARALLEL_YEAR_1	Unit sales_PARALLEL_YEAR_2	The month
23,156.50	46,313.00	66,161.28	January
22,215.50	44,431.00	63,472.71	February
23,167.00	46,334.00	66,191.28	March
0.00	45,049.00	64,355.57	April
0.00	45,085.00	64,406.99	May
0.00	45,611.00	65,158.42	June
0.00	46,671.00	66,672.71	July
0.00	44,777.00	63,966.99	August
0.00	47,964.00	68,519.84	September
0.00	43,945.00	62,778.43	October
0.00	53,807.00	76,866.97	November

Comparing results with those of three parallel years.

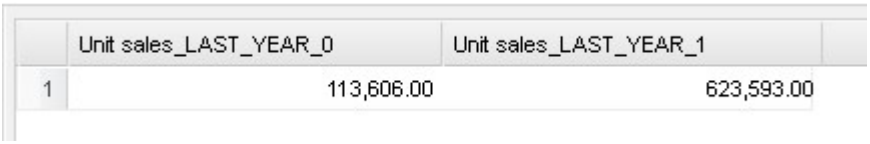
The LAST_YEAR function¶

This function allows the user to sum a measure referring to last period data. If the temporal filter isn't set, the engine takes the current year by default, otherwise the chosen one.

- In our example in the two figures below the period is the year. Here we compare last year sold products to the sum of those sold in last two years.

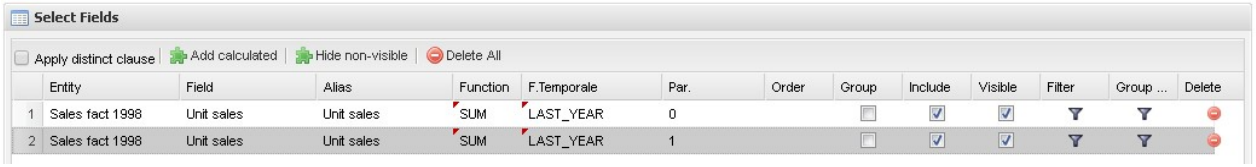


LAST_YEAR function.

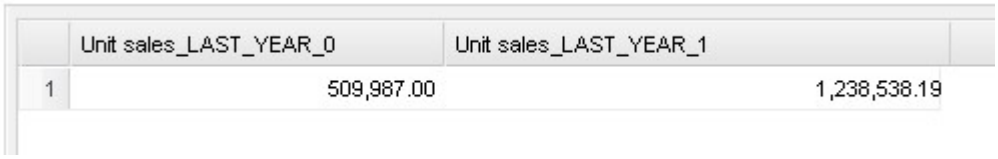


Comparing LAST_YEAR results.

- Referring to figures below give an example of how to define a time reference, for instance 2015. In this case I pass 2015 to the filter.

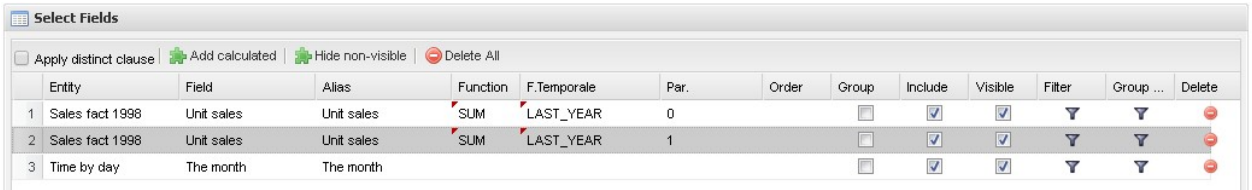


LAST_YEAR function example: changing the reference year.



LAST_YEAR function example: output of changing the reference year.

- In case the user wants to inspect the evolution of sales per month of the current year comparing them with those of last year plus the current. It is enough to add the month in the “selected fields” area and the measure “unit sold” where the LAST_YEAR function is set on 0 or 1. The two following figures show an example.



LAST_YEAR function example: last-year sold products compared to the last-two-year ones.

	Unit sales_LAST_YEAR_0	Unit sales_LAST_YEAR_1	The month
1	23,156.50	533,143.50	January
2	45,372.00	555,359.00	February
3	68,539.00	578,526.00	March
4	91,063.50	601,050.50	April
5	113,606.00	623,593.00	May
6	113,606.00	623,593.00	June
7	113,606.00	623,593.00	July
8	113,606.00	623,593.00	August
9	113,606.00	623,593.00	September
10	113,606.00	623,593.00	October
11	113,606.00	623,593.00	November

LAST_YEAR function example: output of last-year sold products compared to the last-two-year ones.

Note that the operator allows to visualize the sum of sales upon 2 years per month. In other words, LAST_YEAR (1) set to the month level starts the progression from the aggregated value of 2015 to which it adds the sales of 2016.

The LAST_MONTH function¶

This operator is very similar to the previous one. In this case the reference time period is the month. Remember that if the user does not specify the name of the referenced month the system will take the current one by default.

- The user wants to count the sales of last three months.

LAST_MONTH function example: setting time reference.

	Unit sales_LAST_MONTH_3
1	0.00

LAST_MONTH function example: setting time reference.

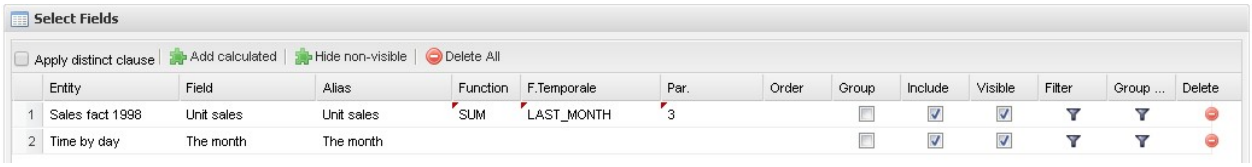
- Figure below shows how to aggregate data up to last three months per each month of the current year. Remember to add the month in the section “selected fields”. Therefore, inserting the month in the select clause the user obtains a projection on current year of sales of last 3 months per each month. Note that data are related to the current year, namely there is no shift to the passed one. Pay attention to the fact that if one month is missing the system does not notice it and return a sum relative to a bigger time period.

LAST_MONTH function example: sum up to last 3 months.

	Unit sales_LAST_MONTH_3	The month
1	23,156.50	January
2	45,372.00	February
3	68,539.00	March
4	91,063.50	April
5	90,449.50	May

LAST_MONTH function example: sum up to last 3 months output.

- The same query can be performed considering a specific year. In the following figures year 2015 has been selected.

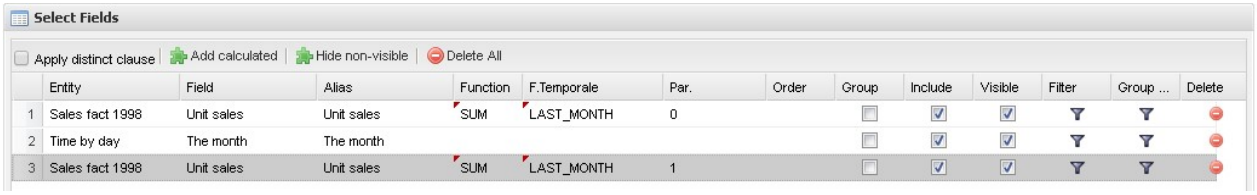


LAST_MONTH function example: sum up to last 3 months where year is 2015.

	The month	Unit sales_LAST_MONTH_3
1	January	46,313.00
2	February	90,744.00
3	March	137,078.00
4	April	182,127.00
5	May	180,899.00
6	June	182,079.00
7	July	182,416.00
8	August	182,144.00
9	September	185,023.00
10	October	183,357.00
11	November	190,493.00

LAST_MONTH function example: output when one sums up to last 3 months output where year is 2015.

- If the user wants to compare sales per month to those of the previous month summed to the current one. Results in the following figures reflect this selection.



LAST_MONTH function example: sales per month against the sum of current and previous month sales.

	The month	Unit sales_LAST_MONTH_0	Unit sales_LAST_MONTH_1
1	January	23,156.50	23,156.50
2	February	22,215.50	45,372.00
3	March	23,167.00	45,382.50
4	April	22,524.50	45,691.50
5	May	22,542.50	45,067.00

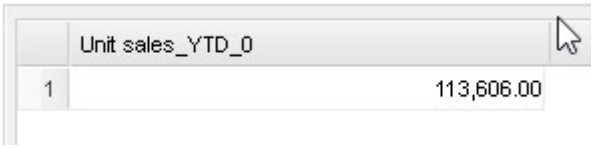
LAST_MONTH function example: results of sales per month against the sum of current and previous month sales.

The YTD function¶

This operator aggregates the measure of the first day of the year up to the execution date (currentDay). If the user sets temporal filters the YTD function must refer to the filter. The chosen day will be used as reference by the function. For example, if the user sets “15/03/2016” as reference day, the function sums starting from the first of January up to the 15th of March (2016). Observe that if the filter is monthly the engine will take the last day of the month, while if it is yearly the engine will take the whole year. If the user inserts a temporal element as aggregation function the measure must be aggregated progressively.

- Below shows the case in which the user wants to count the sales from the beginning of the year up to now.

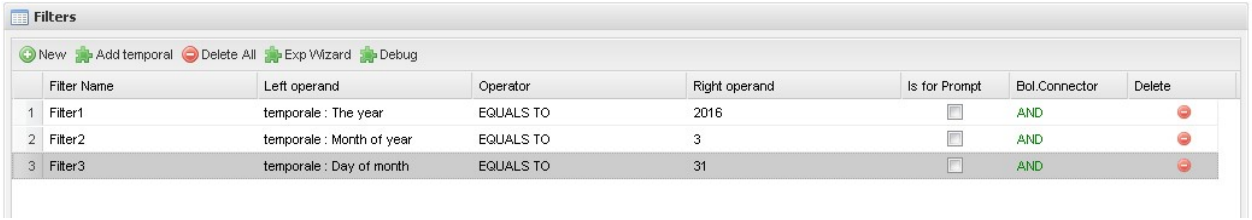
YTD function example: to count the sales from the beginning of the year up to now.



YTD function example: number of sales from the beginning of the year up to now.

- Below shows the case in which the user wants to count the sales from the beginning of the year up to the end of March.

YTD function example: to count the sales from the beginning of the year up to the end of March.



YTD function example: to count the sales from the beginning of the year up to the end of March.

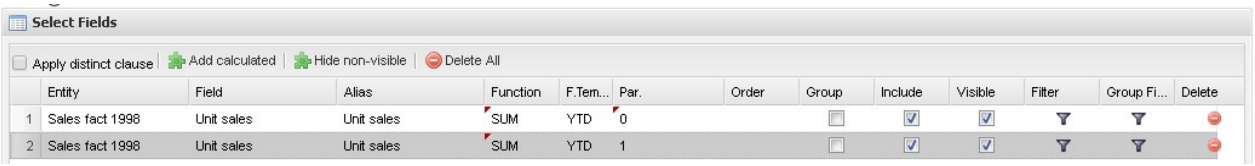
- The following figure refers to the case where the user wishes to sum 2015 sales considering the day in which the query is executed but of the previous year.

YTD function example: sum 2015 sales considering the day in which the query is executed but of the previous year.

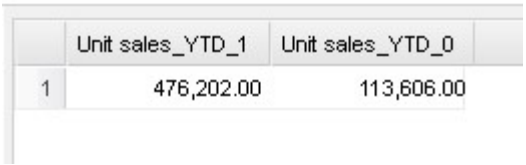
- The following figure refers instead to the case where the user wishes to sum 2015 sales of first 3 months of 2015.

YTD function example: sales summed up to the first 3 months of 2015.

- In the following figures the user is comparing the unit sold from the beginning of the year with those of the previous year. The engine considers the day of query execution as end of the time period.

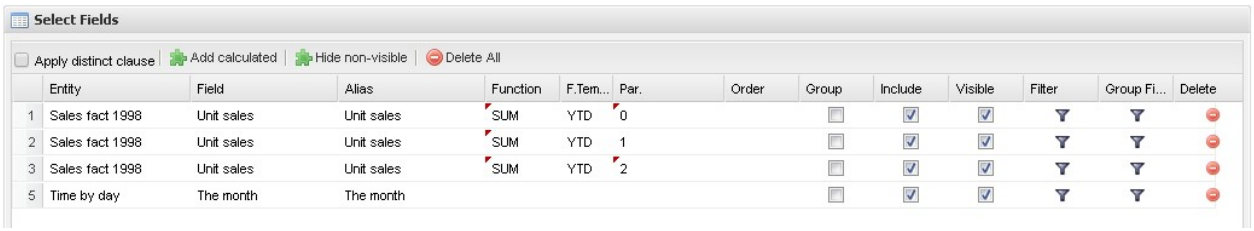


YTD function example: comparing the unit sold from the beginning of the year with those of the previous year.



YTD function example: output when comparing the unit sold from the beginning of the year with those of the previous year.

- Figures below shows the instance when the user wants to see the sum of unit sold each month after having added the month field in the select clause.



	Entity	Field	Alias	Function	F.Tem...	Par.	Order	Group	Include	Visible	Filter	Group Fi...	Delete
1	Sales fact 1998	Unit sales	Unit sales	SUM	YTD	0							
2	Sales fact 1998	Unit sales	Unit sales	SUM	YTD	1							
3	Sales fact 1998	Unit sales	Unit sales	SUM	YTD	2							
5	Time by day	The month	The month										

YTD function example: sum of unit sold each month after having added the month field in the select clause.

	Unit sales_YTD_1	Unit sales_YTD_0	The month	Unit sales_YTD_2
1	46,313.00	23,156.50	January	66,161.28
2	90,744.00	45,372.00	February	129,633.99
3	137,078.00	68,539.00	March	195,825.27
4	182,127.00	91,063.50	April	260,180.84
5	227,212.00	113,606.00	May	324,587.83
6	272,823.00	113,606.00	June	389,746.26
7	319,494.00	113,606.00	July	456,418.96
8	364,271.00	113,606.00	August	520,385.95
9	412,235.00	113,606.00	September	588,905.79
10	456,180.00	113,606.00	October	651,684.23
11	509,987.00	113,606.00	November	728,551.19

YTD function example: output when one sums unit sold each month after having added the month field in the select clause.

The MTD function¶

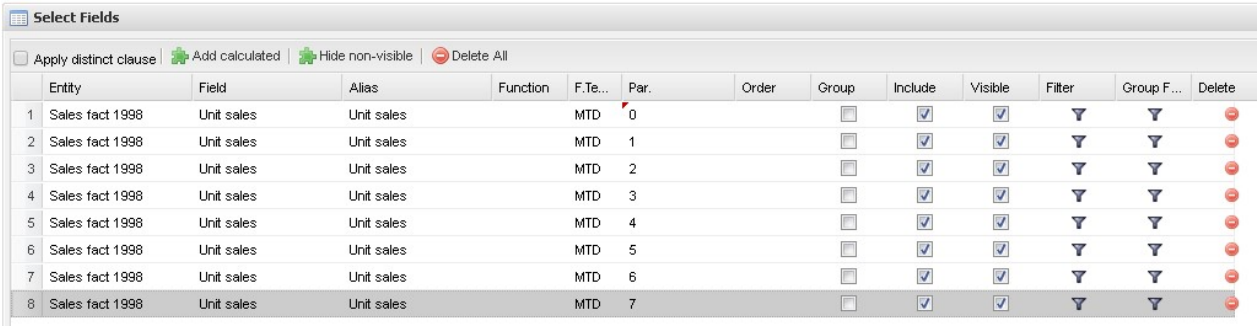
The MTD function follows the same logic as the YTD function but using the month.

- Figure below shows the case in which the user wants to check the unit sold during the current month.

	Entity	Field	Alias	Function	F.Temporale	Par.	Order	Group	Include	Visible	Filter	Group F...	Delete
1	Sales fact 1998	Unit sales	Unit sales	SUM	MTD	0							

MTD function example: check the unit sold during the current month.

- The following figures the user wants to check the aggregated sales of last 7 months, current (relative to the execution time) month included.


Select Fields

Apply distinct clause | Add calculated | Hide non-visible | Delete All

	Entity	Field	Alias	Function	F.Te...	Par.	Order	Group	Include	Visible	Filter	Group F...	Delete
1	Sales fact 1998	Unit sales	Unit sales		MTD	0							
2	Sales fact 1998	Unit sales	Unit sales		MTD	1							
3	Sales fact 1998	Unit sales	Unit sales		MTD	2							
4	Sales fact 1998	Unit sales	Unit sales		MTD	3							
5	Sales fact 1998	Unit sales	Unit sales		MTD	4							
6	Sales fact 1998	Unit sales	Unit sales		MTD	5							
7	Sales fact 1998	Unit sales	Unit sales		MTD	6							
8	Sales fact 1998	Unit sales	Unit sales		MTD	7							

MTD function example: aggregated sales of last 7 months.

	Unit sales_MTD_0	Unit sales_MTD_1	Unit sales_MTD_2	Unit sales_MTD_3	Unit sales_MTD_4	Unit sales_MTD_5	Unit sales_MTD_6	Unit sales_MTD_7
1	0.00	0.00	0.00	0.00	0.00	22,542.50	45,067.00	68,234.00

MTD function example: output of the aggregated sales of last 7 months.

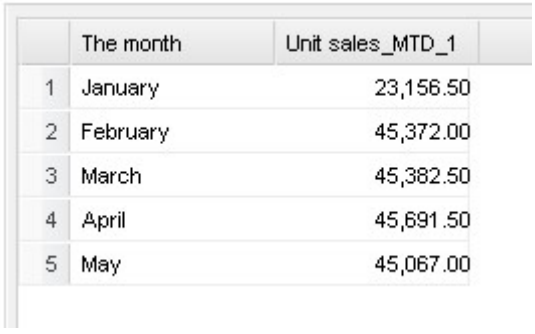
- Another case is shown in figures below where sales are aggregated on current month plus the previous one, relative to the current year (referring to the query execution time).


Select Fields

Apply distinct clause | Add calculated | Hide non-visible | Delete All

	Entity	Field	Alias	Function	F.Temporale	Par.	Order	Group	Include	Visible	Filter	Group F...	Delete
2	Sales fact 1998	Unit sales	Unit sales	SUM	MTD	1							
4	Time by day	The month	The month										

MTD function example: sales are aggregated on current month plus the previous one, relative to the current year.



	The month	Unit sales_MTD_1
1	January	23,156.50
2	February	45,372.00
3	March	45,382.50
4	April	45,691.50
5	May	45,067.00

MTD function example: output when sales are aggregated on current month plus the previous one, relative to the current year (referring to the query execution time) for the present year.

- Figures below shows a user that is summing the sales of current month

Select Fields

Apply distinct clause | Add calculated | Hide non-visible | Delete All

	Entity	Field	Alias	Function	F.Temporale	Par.	Order	Group	Include	Visible	Filter	Group F...	Delete
1	Sales fact 1998	Unit sales	Unit sales	SUM	MTD	0			☑	☑			
4	Time by day	The month	The month						☑	☑			

MTD function example: sales of current month for the present year.

	The month	Unit sales_MTD_0
1	January	23,156.50
2	February	22,215.50
3	March	23,167.00
4	April	22,524.50
5	May	22,542.50

MTD function example: output of the sales of current month for the present year.

- The following figures shows as a user can compare sales of aggregated months (up to the current) to the current one.

Select Fields

Apply distinct clause | Add calculated | Hide non-visible | Delete All

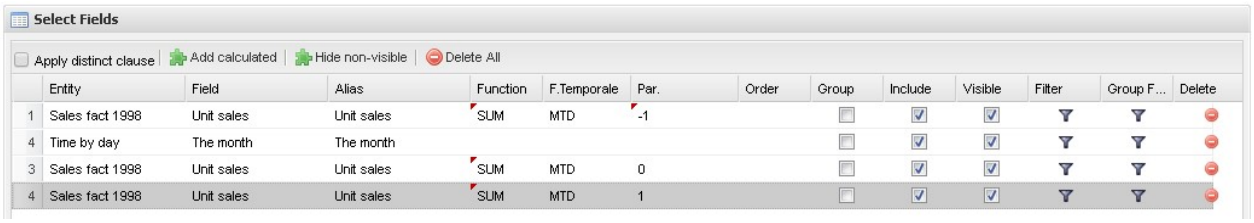
	Entity	Field	Alias	Function	F.Temporale	Par.	Order	Group	Include	Visible	Filter	Group F...	Delete
1	Sales fact 1998	Unit sales	Unit sales	SUM	MTD	1			☑	☑			
4	Time by day	The month	The month						☑	☑			
3	Sales fact 1998	Unit sales	Unit sales	SUM	MTD	0			☑	☑			

MTD function example: sales of current month for the present year.

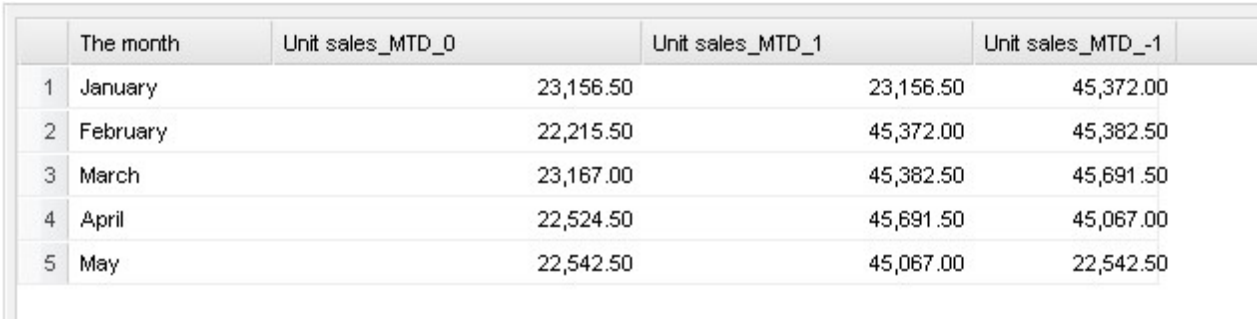
	The month	Unit sales_MTD_0	Unit sales_MTD_1
1	January	23,156.50	23,156.50
2	February	22,215.50	45,372.00
3	March	23,167.00	45,382.50
4	April	22,524.50	45,691.50
5	May	22,542.50	45,067.00

MTD function example: output of the sales of current month for the present year.

- Below shows a case very similar to the previous one. In this case the next month is added to the sum.



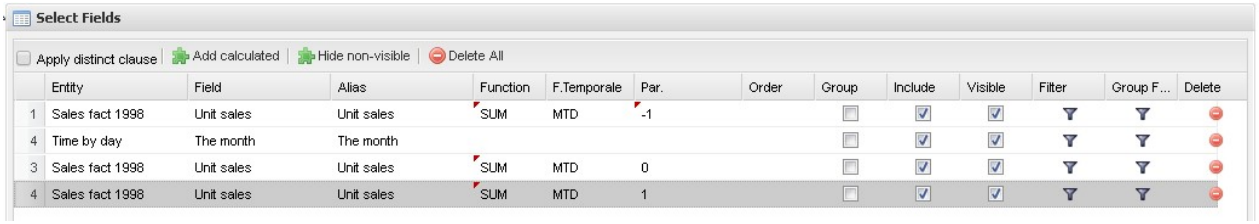
MTD function example: sum of sales of months up to now plus next month.



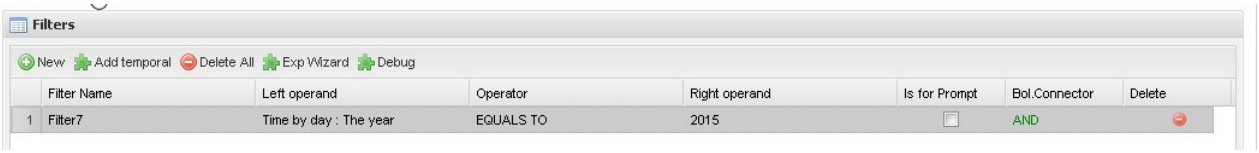
	The month	Unit sales_MTD_0	Unit sales_MTD_1	Unit sales_MTD_-1
1	January	23,156.50	23,156.50	45,372.00
2	February	22,215.50	45,372.00	45,382.50
3	March	23,167.00	45,382.50	45,691.50
4	April	22,524.50	45,691.50	45,067.00
5	May	22,542.50	45,067.00	22,542.50

MTD function example: output when one sums sales of months up to now plus next month.

- The following figures shows a case very similar to the previous one. In this case the reference year is specified through a filtering condition.



MTD function example: sum of sales of months up to now plus next month for a different year.



MTD function example: output when one sums sales of months up to now plus next month for a different year.

Catalogues¶

A hidden panel is activated once you click on the arrow on the right side of the QbE editor, right under the **Preview** button. This panel contains two elements:

- the catalogue of queries (at the top);

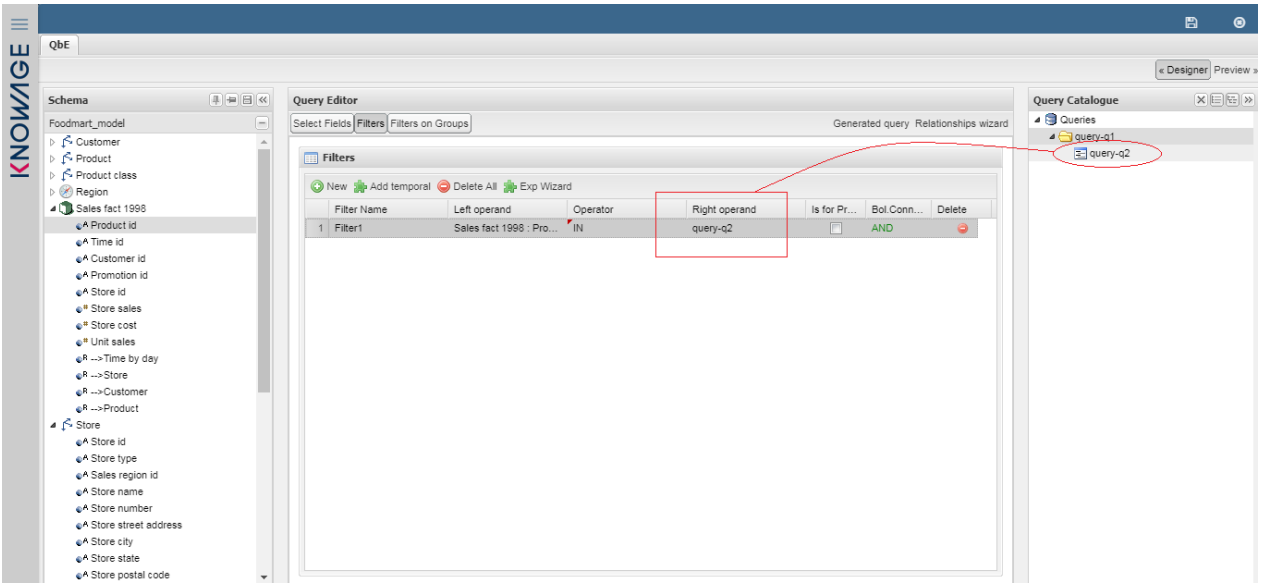
- the list of analytical drivers linked to the QbE document (bottom).

The catalogue of queries is the list of all queries defined in the QbE document, while the lower panel lists all analytical drivers linked to the QbE document.

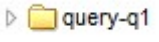
Queries catalogue and subqueries¶
Several queries can be built over the same QbE Datamart. The catalogue lists all saved queries on the current Datamart. The base query that we are creating in the query editor appears with a default name (query-q1): to rename it, simply double click on the query item in the catalogue tree.

To create a new query, click the icon  . The query appears in the catalogue at the same level as the base query. Using the query editor, you can create the query and save it.

The **QbE Engine** also supports the definition and usage of subqueries similarly to the SQL language. As a result, you can define a subquery and use it within a filter in association to the in/not in operator, as shown in Figure below. To create a new subquery, which can be used as a filter inside the main query, click on  . The query appears in the catalogue as a child node of the base query.



QbE query: use of a subquery in a filter.

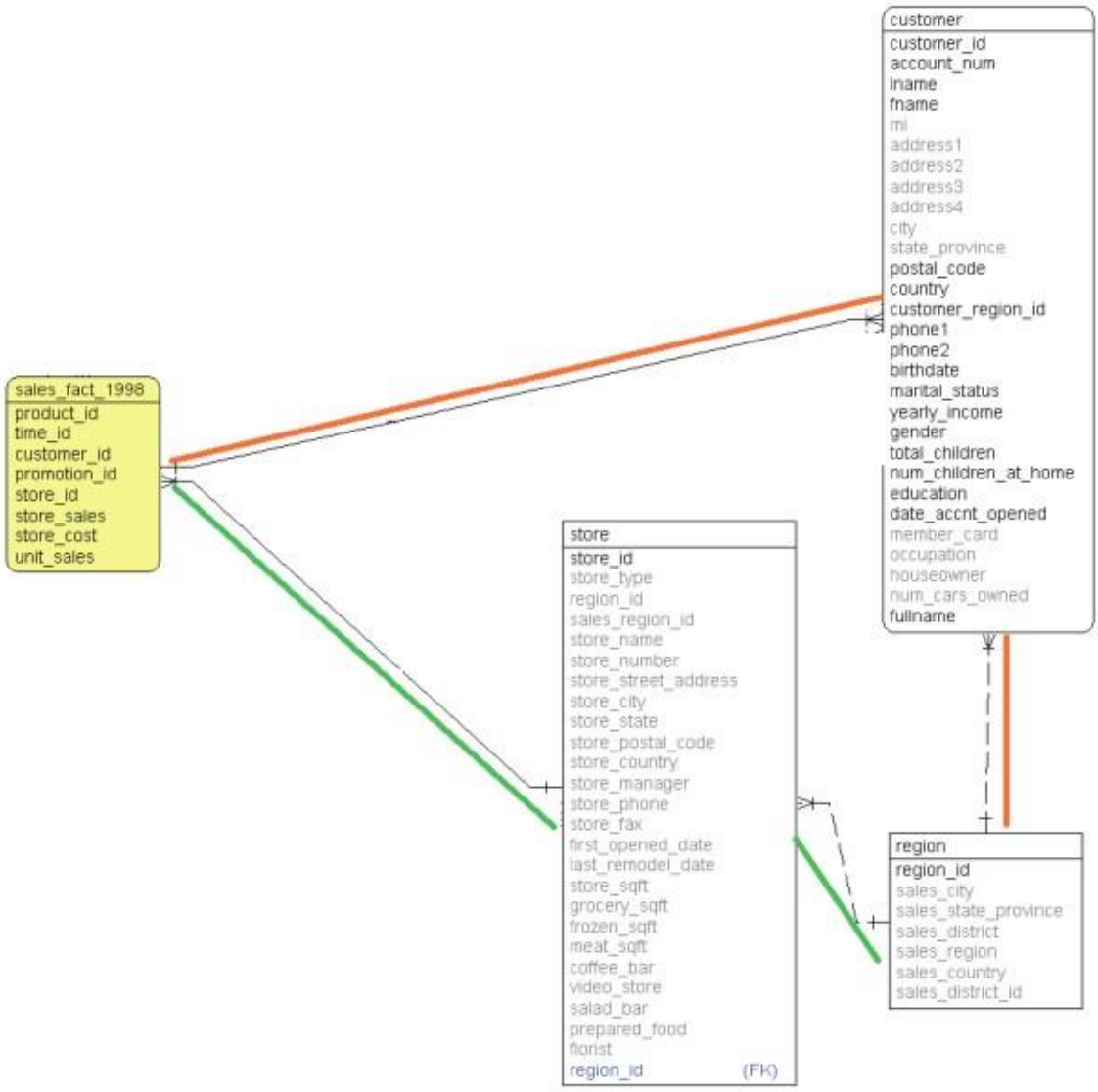
Once defined the main query and the filter that contains the subquery, go to the **Query Catalogue** panel and click on 

. The query appears in the catalogue as a child node of the base query.

To use the sub-query inside the main query, simply drag and drop it into the columns corresponding to the left or right operand of the filter and set the type of operand (**IN** or **NOT IN**). Now the subquery is used to provide values within the filter, in a similar way to SQL subqueries.

Multiple 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.



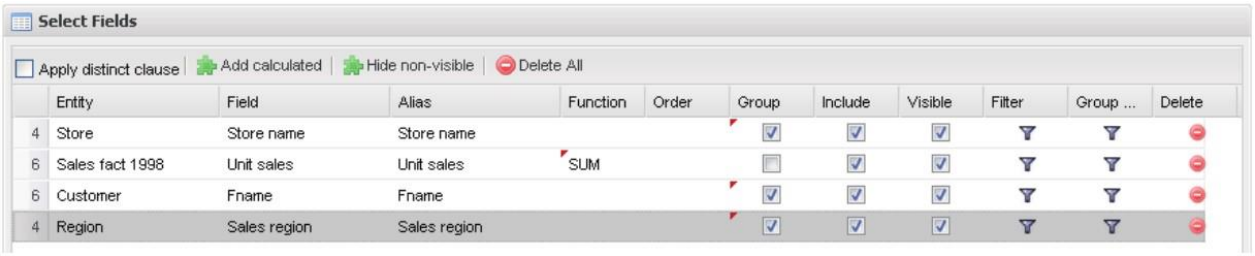
Relationships ambiguity - Schema.

Using the schema and data model represented in figure above, suppose you have a model with the following relationships:

- **Store** - **Region**;
- **Customer** - **Region**;
- **Sales Fact** - **Store**;
- **Sales Fact** - **Customer**.

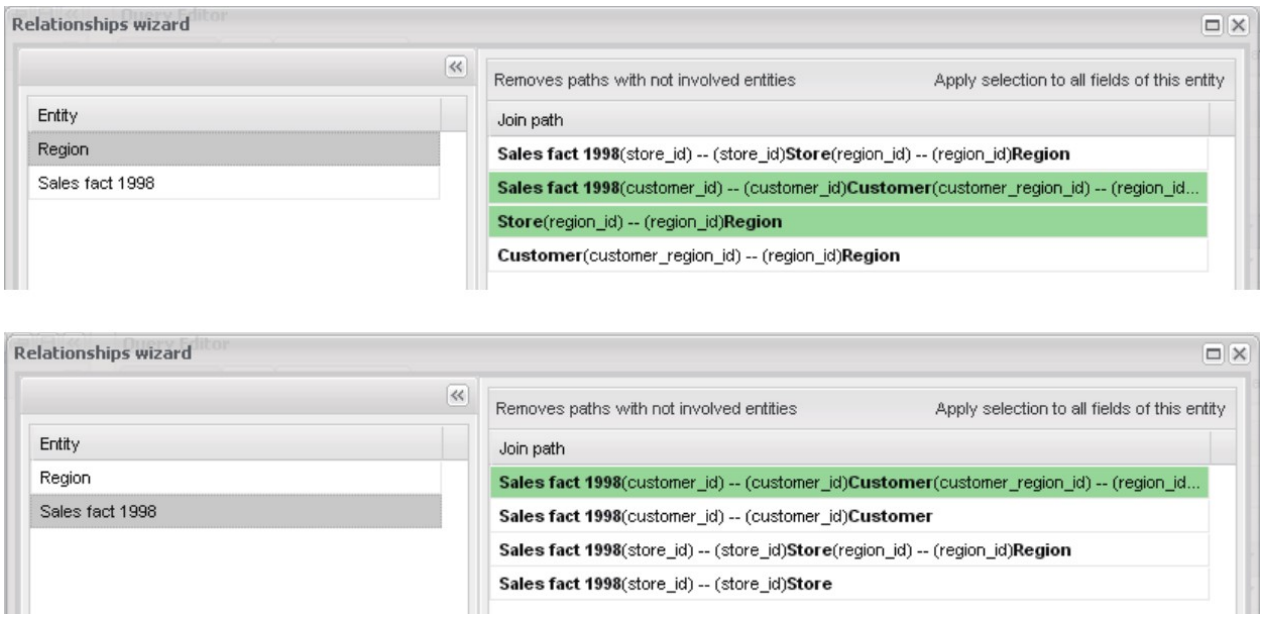
Ambiguity arises when attributes coming from the various tables are dragged and dropped into the query that is build in the QbE, as in Figure below. In this case, in order to identify the items sold by region, you may have one of the following join relationships:

- **Sales Fact - Customer - Region**,
- **Sales Fact - Store - Region**,



Relationships ambiguity - Query definition.

By clicking on the **Relationship Wizard** button in the top right corner of the query editor a pop-up window appears, where users can define the path as shown below.



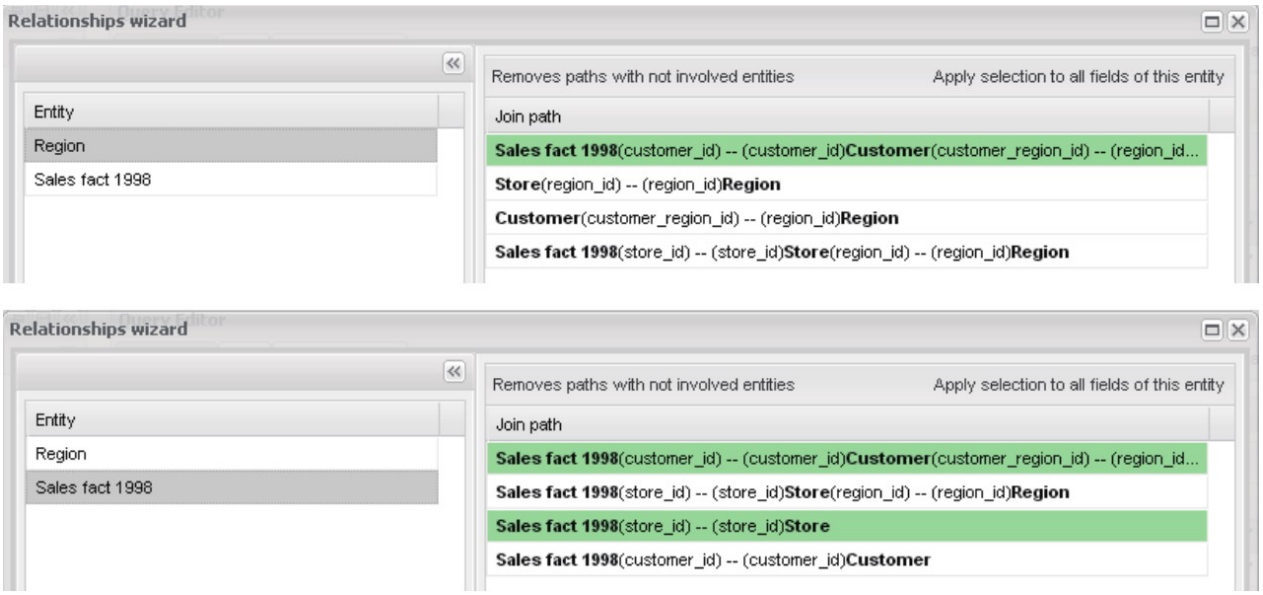
Relationship wizard.

The images of figure above 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, you can modify the relationship so as to eliminate ambiguity: for instance, if you wish to view the region related to a specific customer, first select the **Region** entity in the **Entity** panel on the left and double click the correct path in the panel on the right (the correct path and only the correct path has to be green-colored to be correctly selected).

Remember to repeat this operation for all the entities listed in the **Entity** panel: now select the **Sales Fact** table and the correct path. If a wrong path is selected (green background), double click on the corresponding row to de-select it. The new configuration is shown below.



Relationship Wizard - Choosing paths.

Once you are done, you can check the SQL code generated by the QbE query by clicking the Generated Query button. The relationship between Customer and Region is highlighted in bold, as shown below.

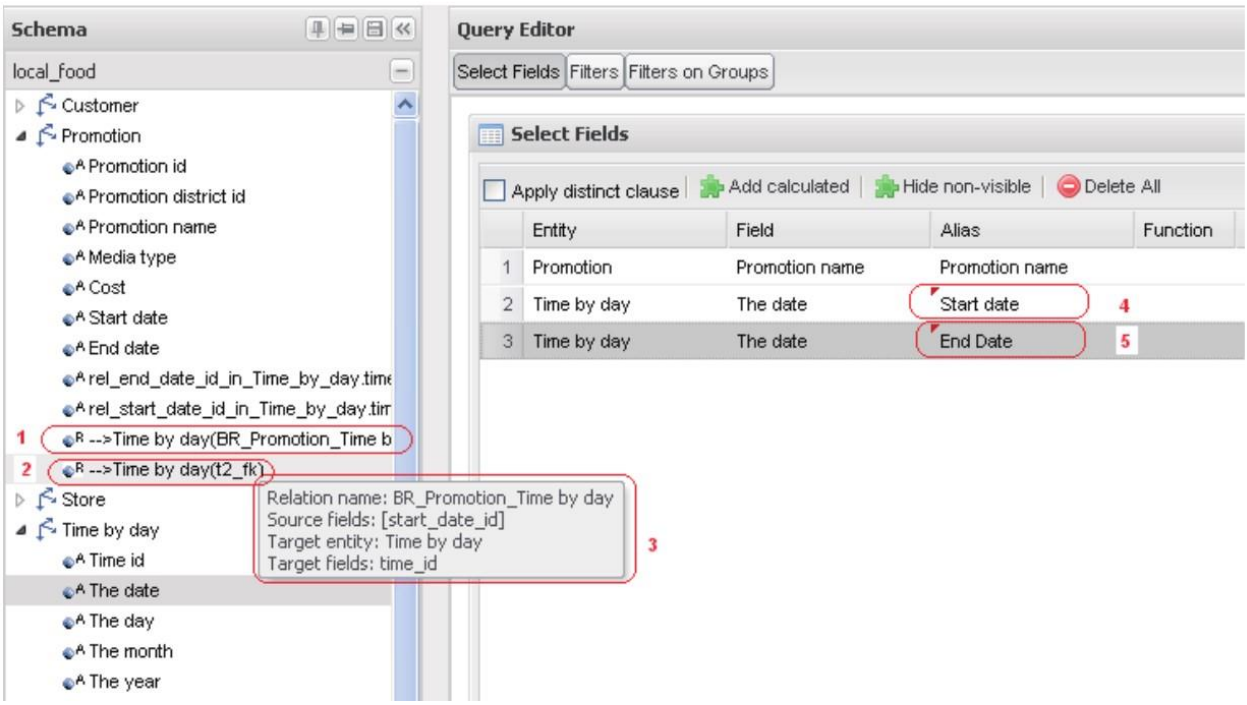
```
select
    customer0_.`fname` as `Fname`,
    store1_.`store_name` as `Store_name`,
    region2_.`sales_region` as `Sales_region`,
    sum(sales_fact3_.`unit_sales`) as `Unit_sales`
from
    `foodmart_key`.`customer` customer0_ cross
join
    `foodmart_key`.`store` store1_ cross
join
    `foodmart_key`.`region` 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`
```

Generated query.

Aliases and relationships

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. Suppose there is a double relationship between **Promotion** and **Time by Day** entities (see the figure below)).



Double relationships.

The two relationships concern the start date and end date of the promotion. As shown in the figure above, 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 (from the **Promotion** entity) and the **The Date** field (from the **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).

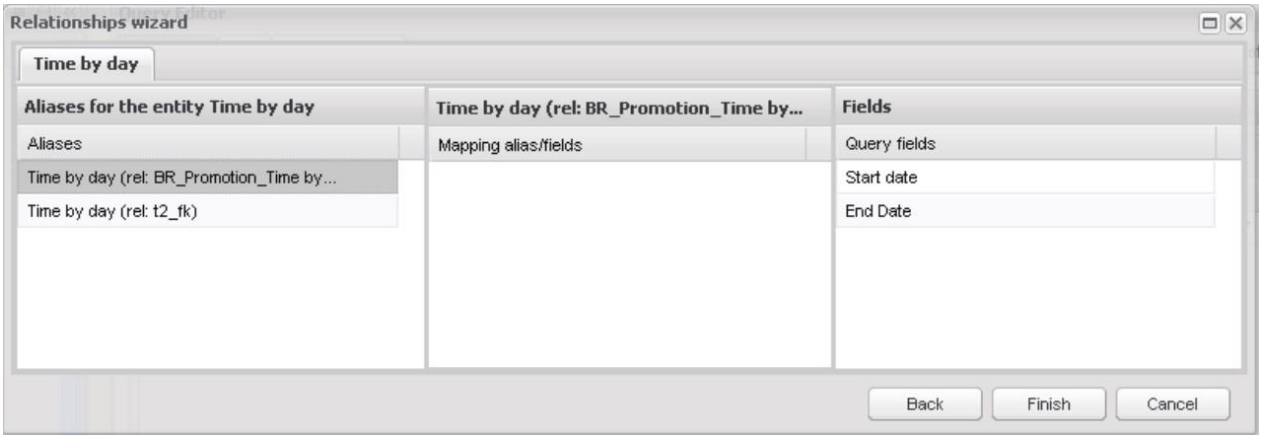
By executing the query, you will see that in the absence of specific indications, the system selects two relationships (path) at random.

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. If you wish to use both relationships (end date and start date), select both as shown below. The tooltip shows the complete path using an intuitive tree layout.

Relationship wizard - Double relationships (I).

Once the relationships are selected in both entities, click on **Apply**.

The window shown below will appear.



Relationship wizard - Double relationships (II).

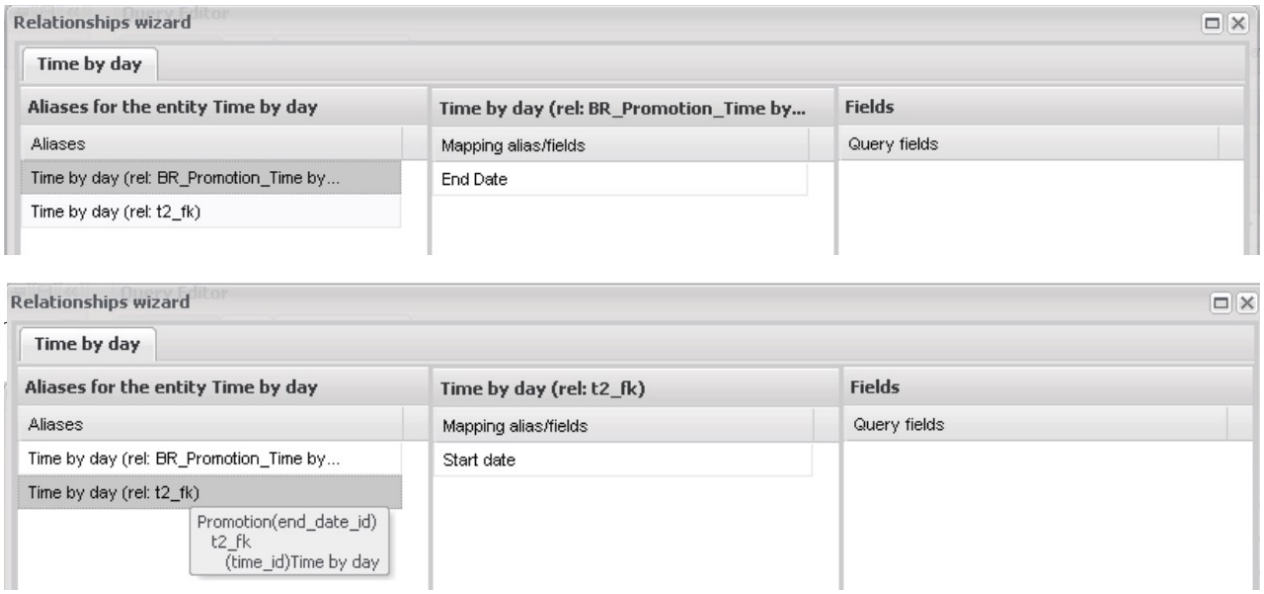
It includes three sections:

1. List of aliases: the first column on the left contains the different entity fields;
2. List of fields associated to the entities: here you can set the associations between aliases and entity fields;
3. List of fields: the first column on the right contains the aliases that you previously defined in the query and corresponds to the columns that you expect to be shown in the resulting table.

To distinguish the fields during the execution of the query, it is necessary to identify all the fields involved in the query (included in the third section List of fields) with the aliases of the entities that contain them (included in the first section List of aliases).

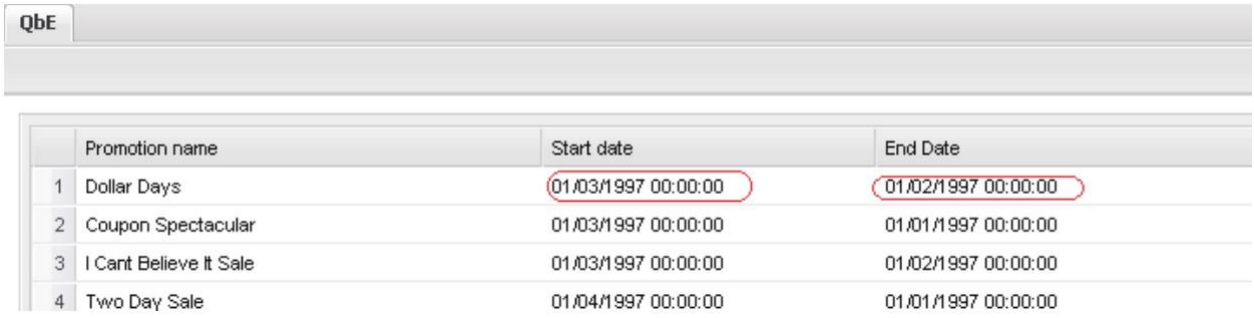
In this case, select the **Time by day (rel BR_Promotion_...)** entity in the first column, then drag and drop the **End Date** field from the third column to the one in the middle. Repeat the same with the **Time by day (rel: t2_fk)** entity and the **Start date** field.

The results are shown below. To check whether the association was correctly set, you can refer to the relationship specified in the tooltip.



Relationship wizard - Double relationships (III).

Click **Finish** and check the SQL code clicking **Generated Query**. Figure below shows the desired result.

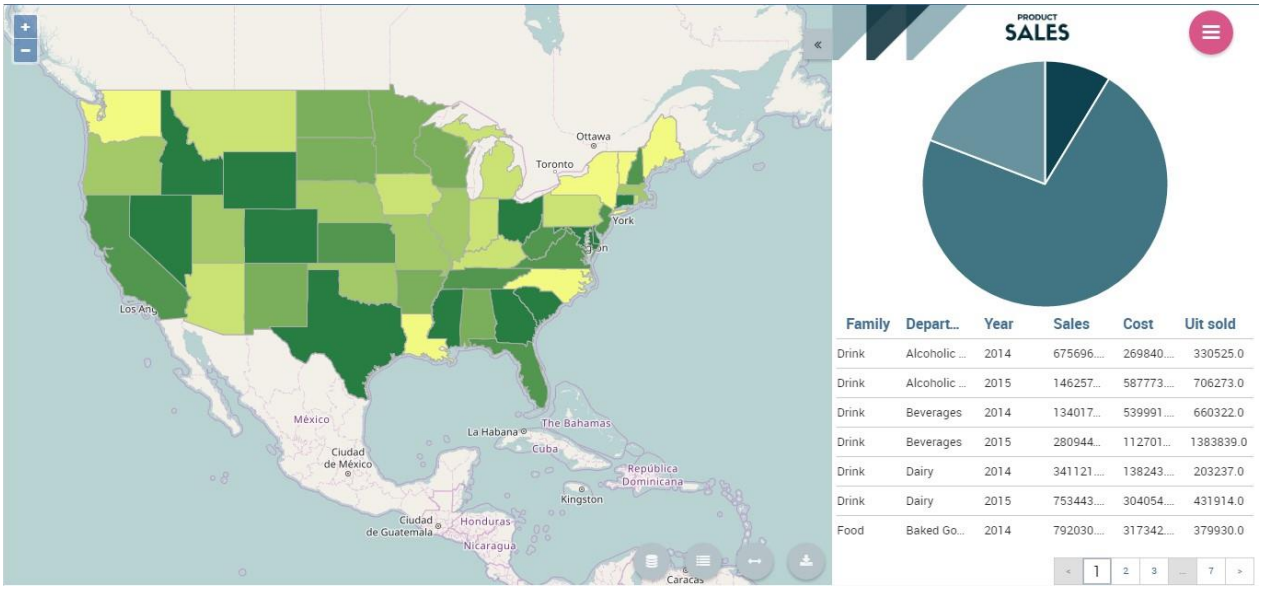


	Promotion name	Start date	End Date
1	Dollar Days	01/03/1997 00:00:00	01/02/1997 00:00:00
2	Coupon Spectacular	01/03/1997 00:00:00	01/01/1997 00:00:00
3	I Cant Believe It Sale	01/03/1997 00:00:00	01/02/1997 00:00:00
4	Two Day Sale	01/04/1997 00:00:00	01/01/1997 00:00:00

Double relationship preview.

Cockpit

Knowage allows end users to *self-build interactive cockpits* through an intuitive and interactive interface, with a few clicks and simple drag and drop. This allows you to compose your analytical documents with multiple widgets and define associations among them, so that clicking on one widget data are automatically updated in other widgets.



Cockpit document example.

It enables *data mash-up* to integrate enterprise data and externally sourced data.

Cockpit documents can be created and executed both by technical users and end users and are part of Knowage ad-hoc reporting system. A key aspect is that different widget can rely on different datasets and hence on different data sources. The only requirement needed to define associations between two or more datasets is the presence in each of them of one or more columns containing the same data.

My first Cockpit

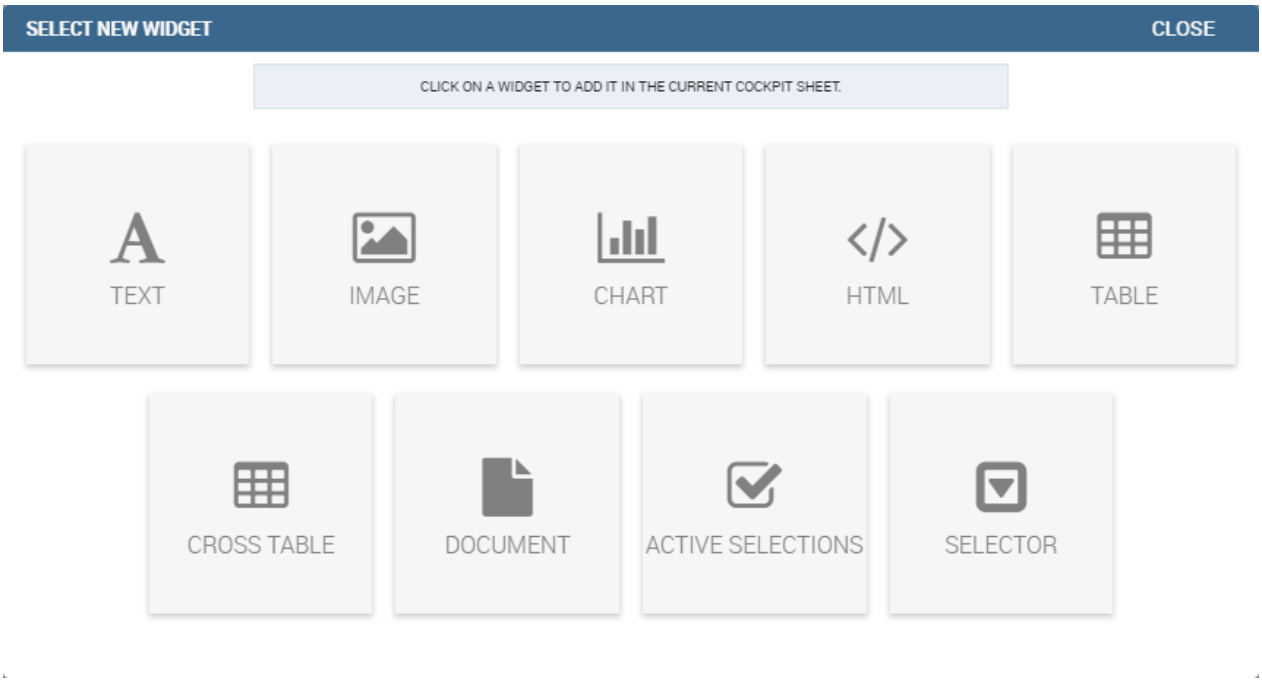
You can create your new Cockpit from the **Analysis** area of the **Workspace** by clicking on the “Plus” icon and selecting **Cockpits** if you enter Knowage Server as final user, while you can enter the document browser and start a new cockpit using the “Plus” icon if you enter Knowage Server as admin.

Let us see how to build a cockpit and how the interface is displayed within the server. Once opened, the cockpit interface is an empty page with a toolbar containing different options described in Table below.

Cockpit editor toolbar.

Icon	Name	Function
	**Cockpit menu**	Configuration menu of Cockpit.
	**Add widget**	It opens a window where you can create a new chart or table, add texts, images or Knowage documents.
	**General configuration**	It opens the window where you set the general cockpit options (name, label, show menu, etc.) and widget style (header, titles, borders, etc.).
	**Data configuration**	It opens a window where you can manage the dataset, the association between datasets and the refresh frequency.
	**Selections**	It adds a widget that manages selections.
	**Clear Cache**	It cleans temporary data.
	**Save as**	It opens the window to save the cockpit document as a new document.

By clicking the button **Add Widget** you can add a widget containing a **Text**, an **Image**, a **Chart**, a **Table**, a **Cross table**, a **Document**, the **Active selections** or the **Selector** to your cockpit, as shown below.

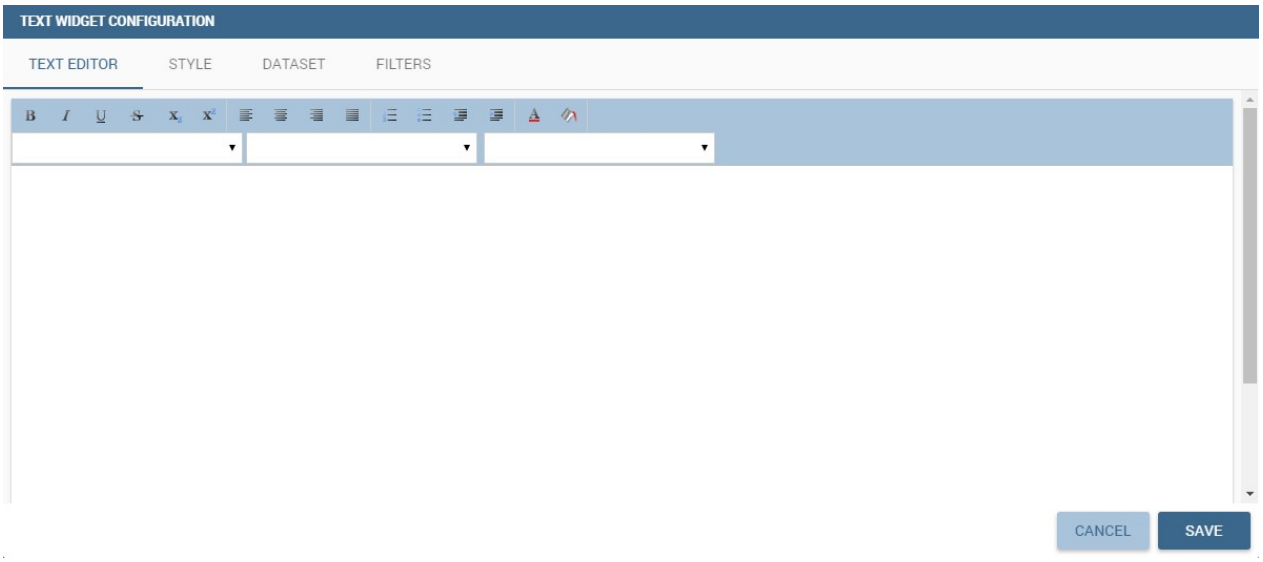


Widget Type.

In the following we go into details of each available widget.

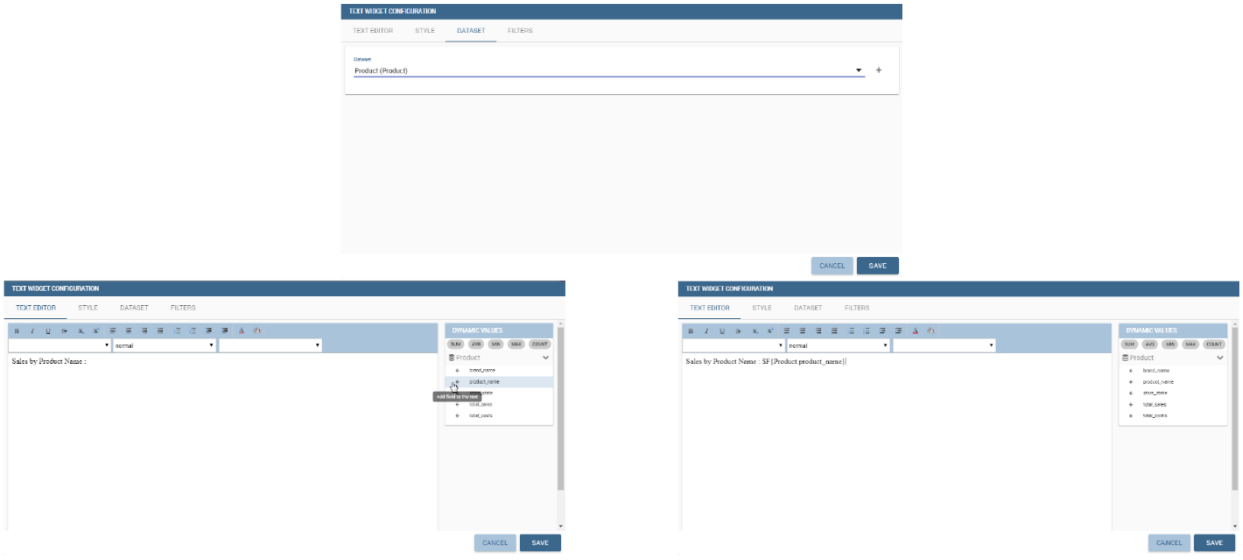
Text widget

By clicking the button Text Widget, you can add text to your cockpit. As shown in figure below, the widget editor opens and it is divided in three tabs: The **Text editor**, the **Style**, the **Dataset** and the **Filters** tab.



Text editor of text widget configuration.

On the “Text editor” tab you can type the desired text in center panel and customize it. Using the dataset tab, it is possible to associate dataset values to the text and read it real time at each execution. Move to the dataset tab to add a dataset to the widget. Then, going back to the Text editor tab, the user will find the dataset columns on the right side, as well as a set of functions to eventually apply to the fields. We summed up main steps in the Figure below. To add a function to a measure first select the desired function and then the field of numeric type.



Editing a dynamic text.

On the “Style” tab you can customize the text widget. We have provided all details about this tab in the Table widget. On the “Dataset” tab you can add more dataset to be used in the dynamic value. Finally, the “Filters” tab can be used to extract limited output from the dataset. We put details off to the table widget subsection.

Image widget

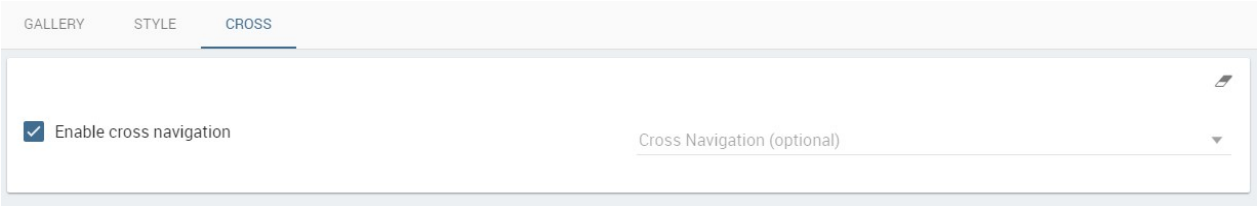
By clicking the button **Image Widget,** you can add images to your cockpit. As already seen the widget editor opens and it is divided in three sections.

On the **Gallery** tab you can upload an image, delete it or select one from the gallery. Refer to the following figure.

Gallery tab of Image Widget Configuration.

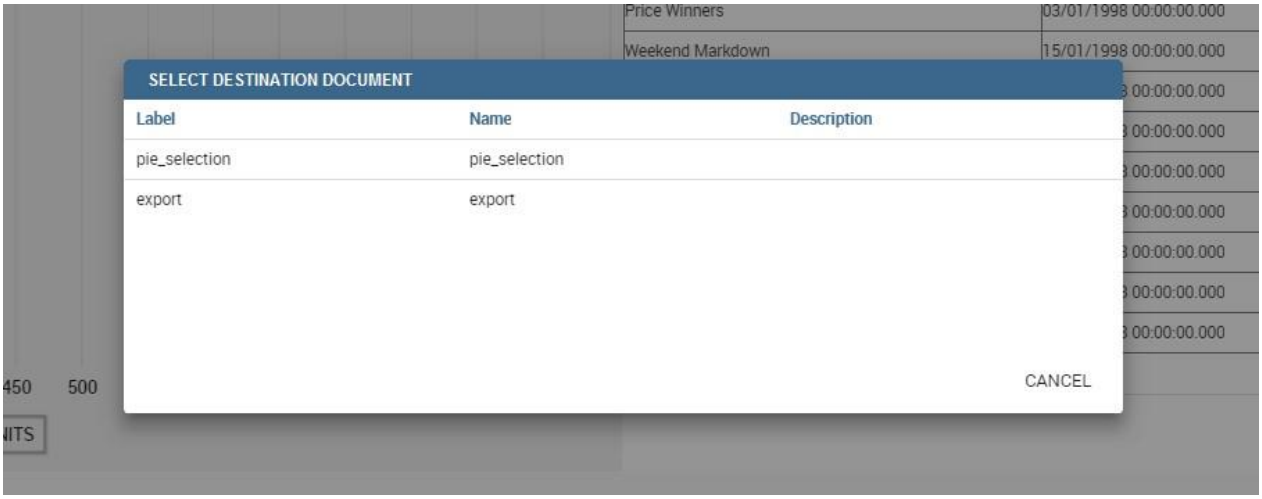
On the **Style** tab you can configure the style of your image widget with the different options offered by this tab. Many of them are defined in the table widget that you will find later.

On the **Cross** tab you can define navigation to another document, as shown in figure below.



Cross tab of Image Widget Configuration.

For this purpose, you must activate **Enable cross navigation** flag and select the destination document through the list of cross navigation definition. This last flag is optional. If you select a cross navigation definition, when you launch the cross navigation it will go to the document of arrival directly. If the cross-navigation definition is not defined, then when you launch the chart widget cross navigation will be shown a pop up (refer to figure below) with the list of cross navigation definition that exist for this cockpit.



Cross navigation multiple choices.

Chart widget

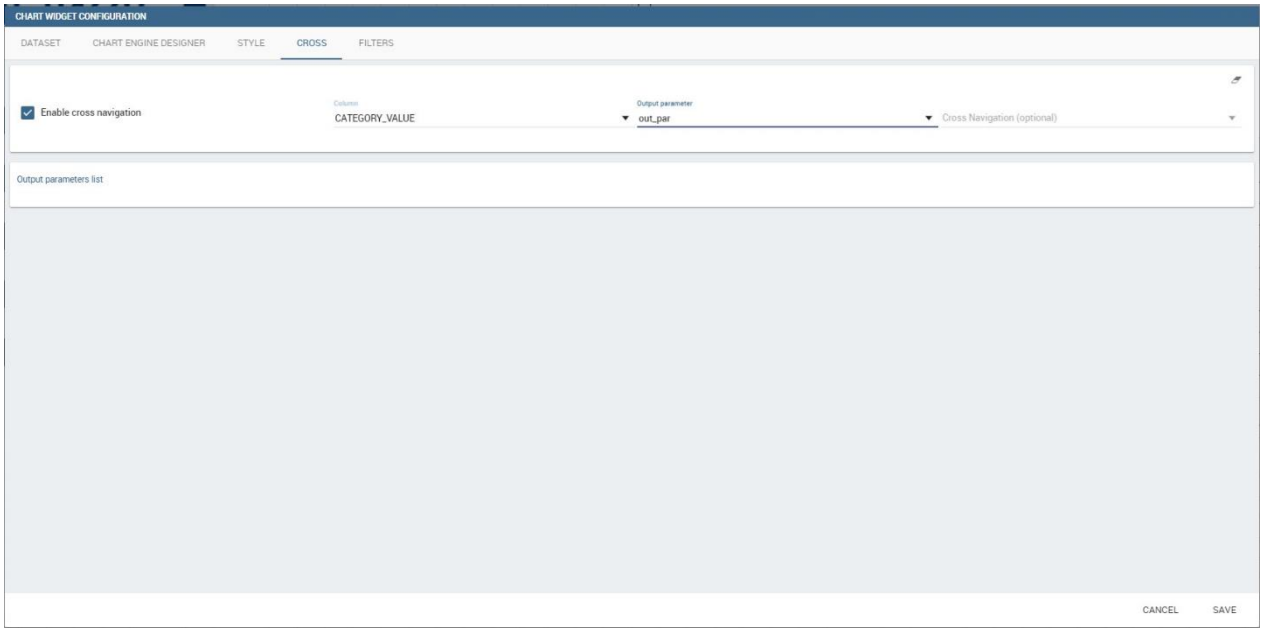
Charts are an essential representation of data, Knowage let you use many different charts type and configure them according to your needs. We have provided all details about charts type and configuration in Chart chapter.

We recall that also for chart widget it is possible to set cross navigation on elements.

As shown in next figure, it is mandatory to enable the cross-navigation feature by using the dedicate tab of chart editor GUI. It is mandatory to choose the column element to be

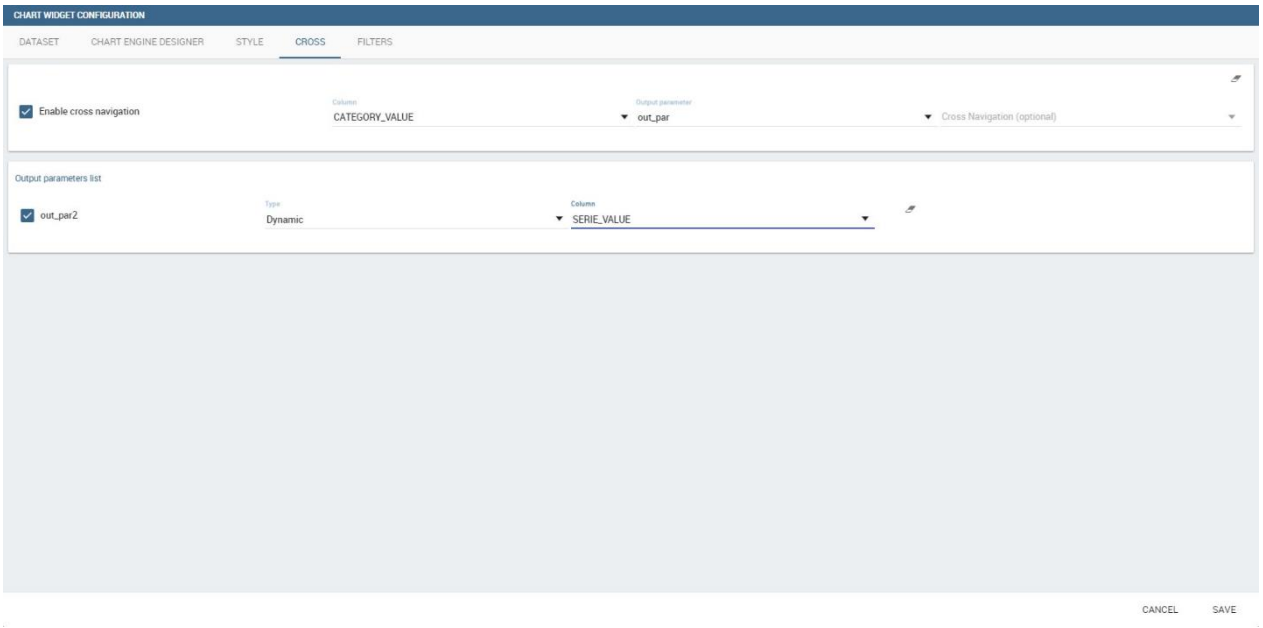
passed to the destination document and associate it to the right output parameter (previously added to the document using the detail interface).

The cross-navigation name can be left empty. In case multiple cross navigation definitions have been configured for the document, a pop up will be displayed, letting the user to choose which destination to reach (exactly as we saw earlier for Image widget in the last figure of that paragraph).



Cross navigation for chart widget.

In addition, if the navigation expects other parameters to be passed, use the bottom part of the page to add the additional parameters. The figure below shows an example.



Add all output parameters involved in the cross navigation.

Table widget

The **Widget table configuration** opens, and it guides you through the steps to configure the widget. The pop-up opens showing the **column** tab, as you can see from Figure below. In details, it is mandatory to select a dataset using the combo box (only if at least one dataset has been loaded using the **Data Configuration** feature) or clicking on the icon + available just aside the combo box line. You can page the table specifying the number of rows per sheet. Consequently, the user can set columns properties.

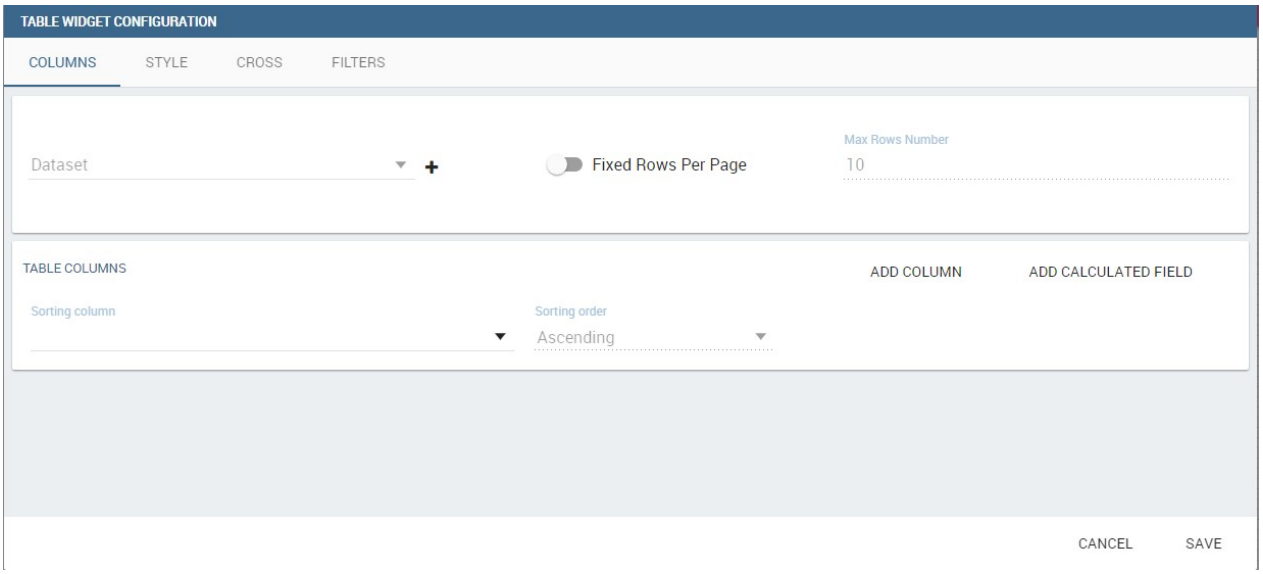
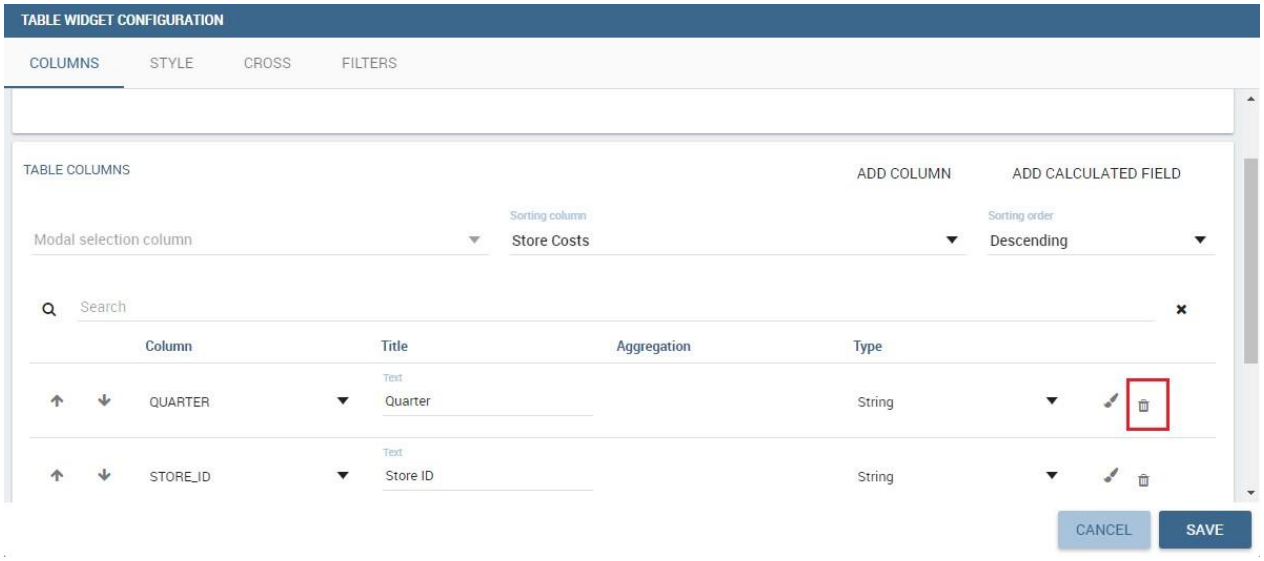


Table configuration window.

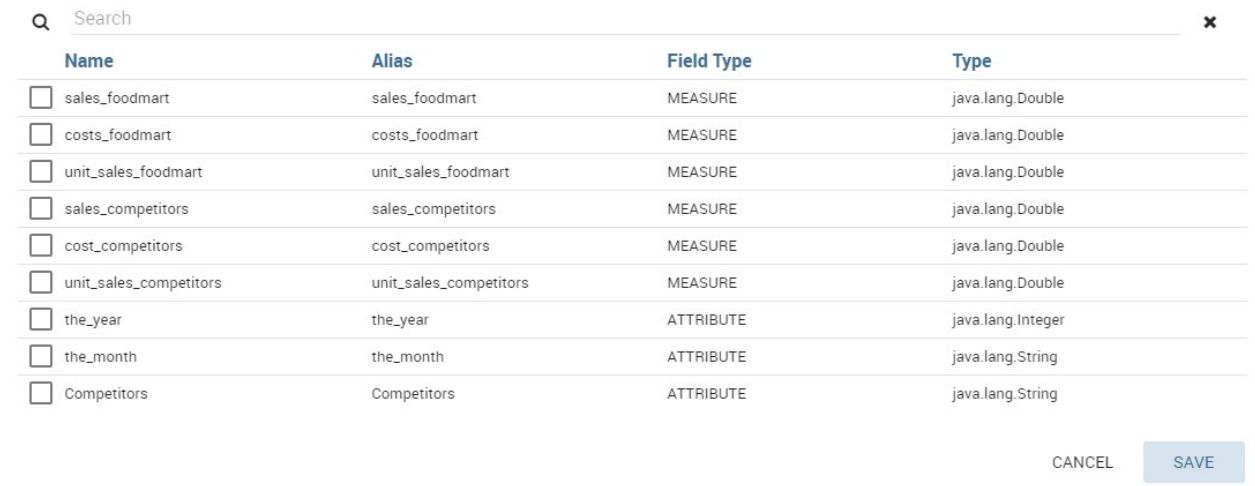
In fact, the column area is divided into two parts: on the left side you have columns ordering, on the right the user has the button to add a new column or a calculated field. As soon as the dataset is selected, you can indicate the sorting column or modal selection column. The modal selection serves to specify which value will be passed to other widgets (if interaction is enabled) when clicking on the cell at document execution time. You can specify this field by selecting a value from the combo box. In the same way, you indicate the sorting column and the order type that steers the rows ordering. You can select the field and the order from the dedicated combo boxes.

When a dataset is added to a table widget, all its columns are listed below. If the user doesn't wish to show some of them, he can use the delete button available at the end of each column row, as shown below.



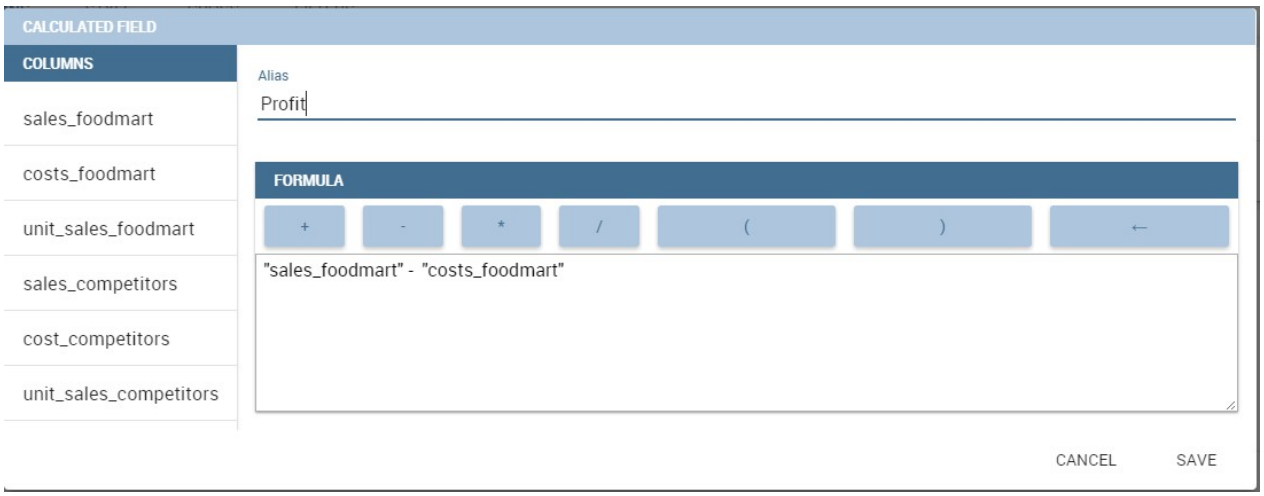
Delete a column.

In case of accidental cancellation or new table requirements, it is possible to re-add columns. In order to add a new column, you must click on the **Add Column** icon on the top right of the second box. Once opened you can select one or more columns. When you have finished selecting the desired columns you can click on save button and your new columns will appear in the field list. Refer to Figure below.



Add a new column.

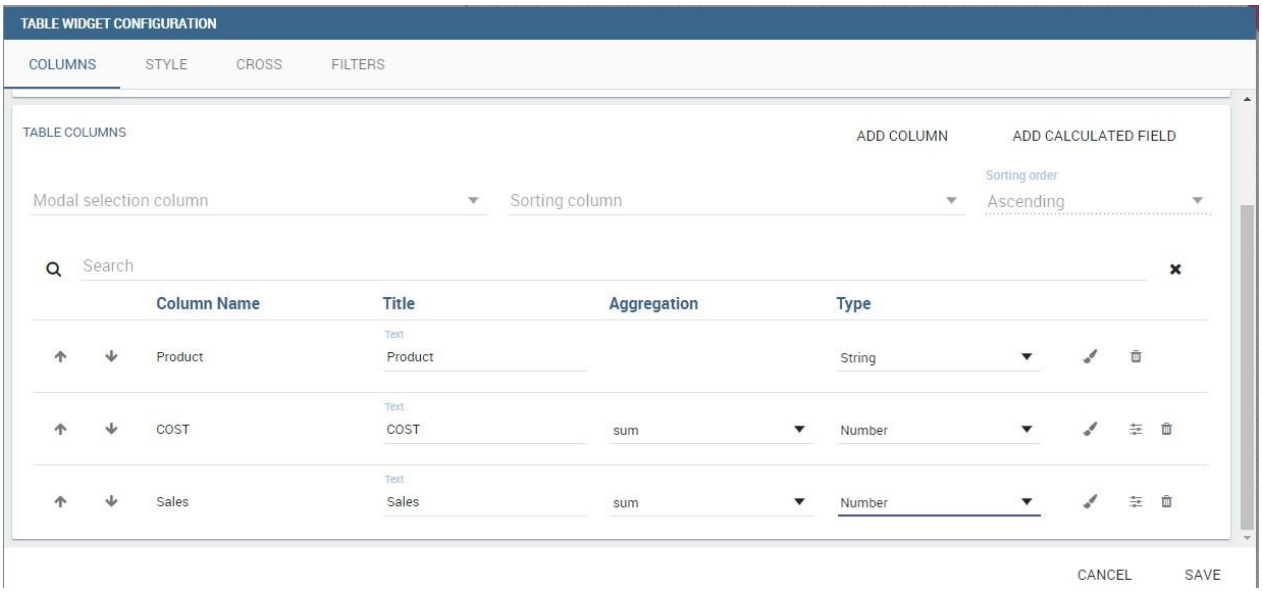
Likewise, to add a calculated field you must click on the **Add Calculated field** icon next to add column icon. Once opened the Calculated Field Wizard you must type an alias for your calculated field in the dedicated area at the top corner of the wizard. Then you can choose from the Items Tree the fields and the arithmetic function you want to use for building your expression. In the middle you can see the expression you have built. If you prefer you can create or modify the expression manually directly in the panel which is editable. When you are satisfied with your expression you can click on save button and your calculated field appears in the field list. We provide an example in the following figure.



Add a calculated field.

At the very bottom of the window, you can see the dataset fields listed and you also can sort columns displayed in the table, insert a column alias and customize it by adding font

and style configurations using the brush shaped icon, as you can see from figure below. Here you can find configuration features like the column size, max cell characters, hide on mobile option, etc.



Column settings.

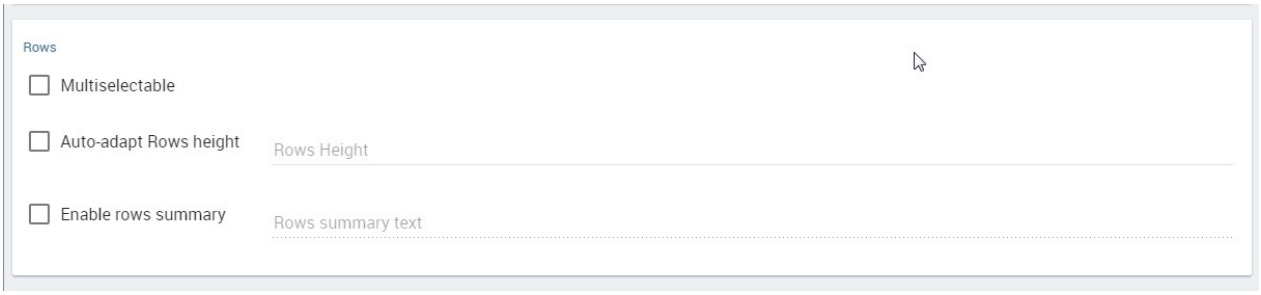
Note that here you can indicate the column type and the aggregation. To add an aggregation to a column you must control the type of data that column has. An aggregation can only be added if the column value is of “number” type. The different aggregation functions are *none (you also can not add any aggregation function)*, *Sum*, *Average*, *Maximum*, *Minimum*, *Count* and *Count distinct*.

The **Style** tab is where you can customize the table by using the different options of style. It is divided into eight parts:

- In the **Summary** section you can show the total of the column and customize it by typing the summary name and using font and style configurations. Refer to Figure below.

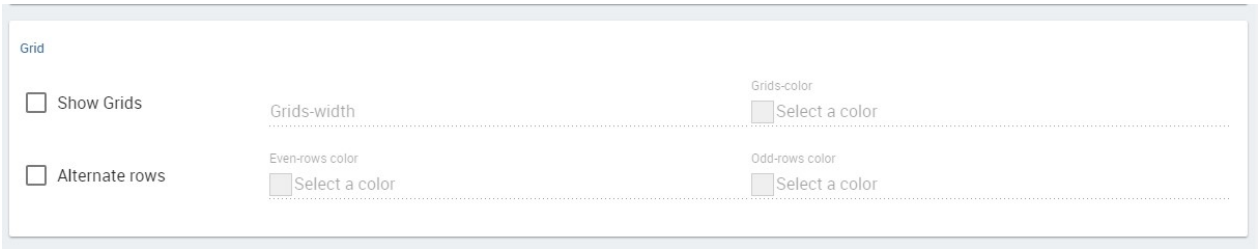
Summary section of the Style tab.

- In the **Rows** section you can set the table rows to be adapted in automatic or select a fixed height. You can also show the total of rows. While the multi-selectable option allows you to select multiple values and pass them to other cockpit widgets or other external documents. Refer to figure below.



Rows section of the Style tab.

- In the **Grid** section you can add borders to the table and add color to alternate rows. In this section you can find different options to customize them. Refer to figure below.

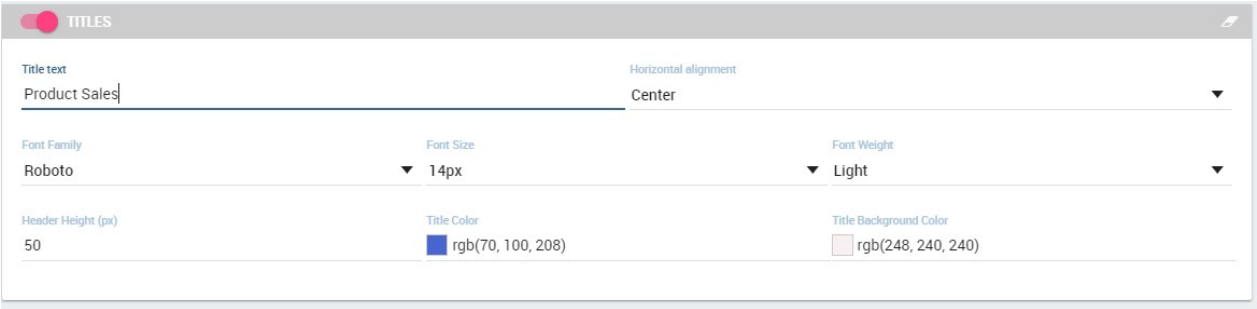


Grid section of the Style tab.

- In the **Header Style** section, you find the different options of style for the table header. Refer to Figure below.

Header style section of the Style tab.

- In the **Titles** section you can add the titles to the widget and modify the font size and weight. In this section you can also change the height of the widget title. Refer to Figure below.

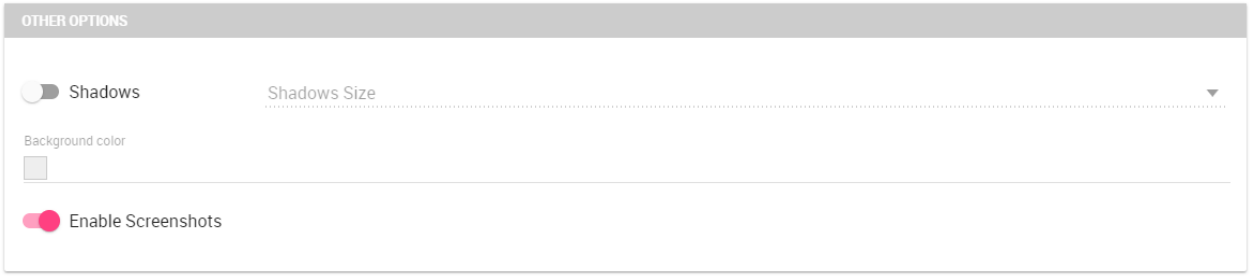


Titles section of the Style tab.

- In the **Borders** section you can add a border to the widget and customize it by using the colors, thickness and style. Refer to the following figure.

Borders section of the Style tab.

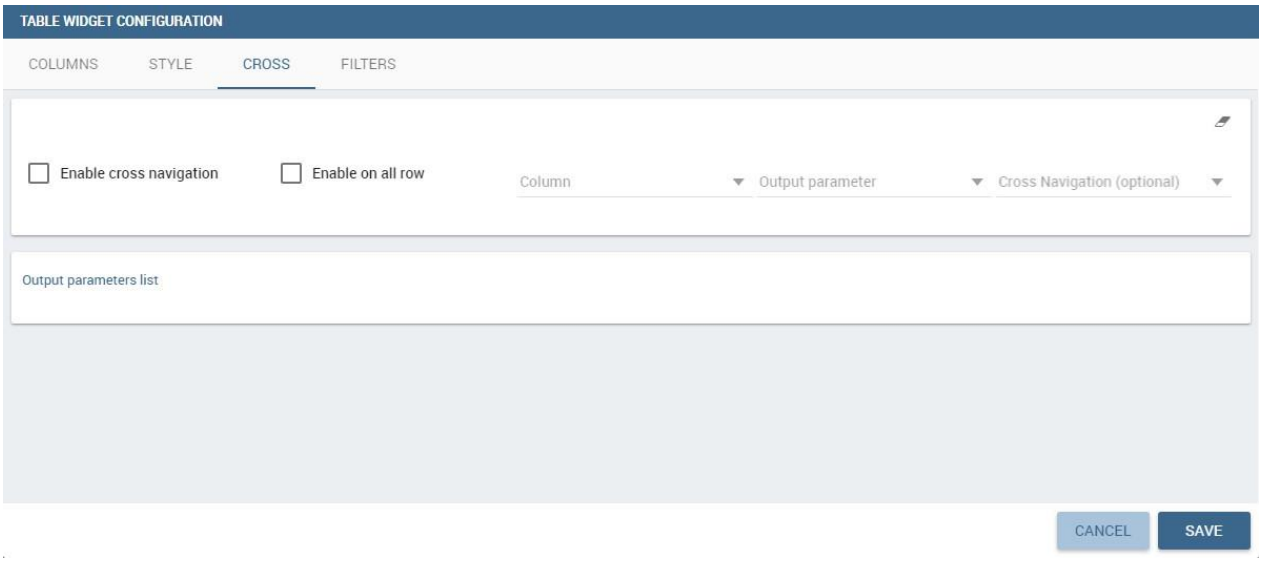
- In the **Other Options** section, you can add the shadows in the widget, you can set the background color of the widget and it is possible to disable or enable the screenshot option for that widget. Refer to the following figure.



Other Options section of the Style tab.

Once the table style settings have been implemented you can switch to the next tab. The “Cross” tab is where the navigation to other documents is defined. It is visible to final users but only configurable by a technical user (like an administrator).

Referring to figure below, we sum up how to add a cross navigation to the cockpit with the following bullet list:

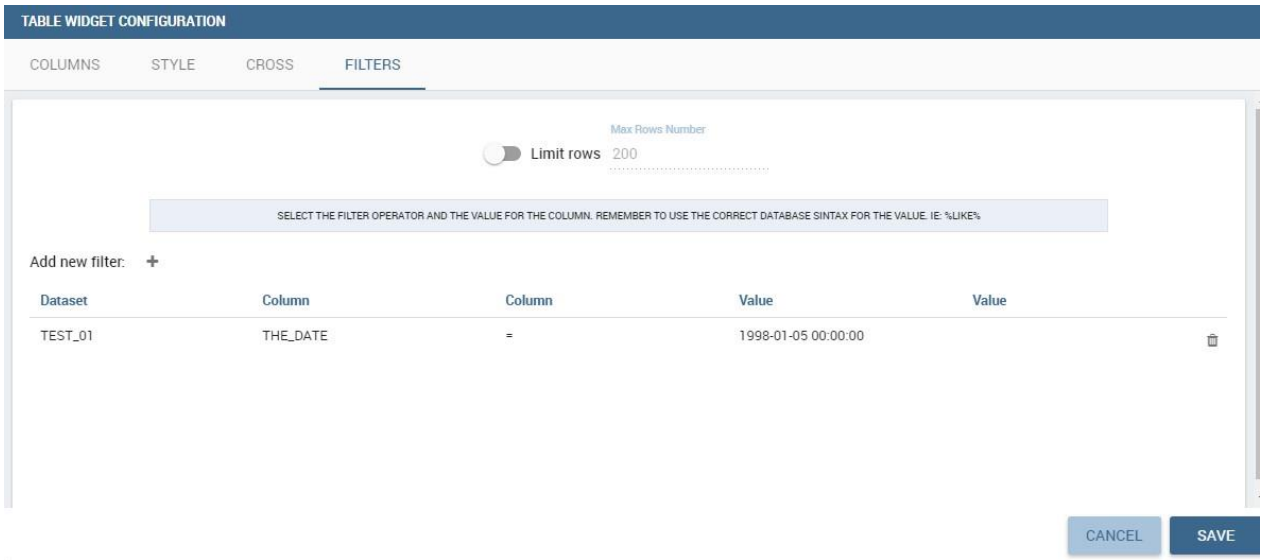


Cross tab of the table widget configuration.

- activate the cross-navigation flag;
- activate cross Enable on all row flag, if you want to be able to click on all the columns of the table;
- select the column whose value will be passed through output parameter to the document of arrival;
- select the output parameter that will pass the value to the document of arrival. This parameter type is defined in the document detail of the cockpit;

- select the destination document through the list of cross navigation definition. It is optional. If the Cross navigation is not selected then when you click to launch the cross navigation, a pop up will be open with all the cross navigations defined for that cockpit. If you select the Cross navigation and you click to launch the cross navigation, then it will go to the document of arrival directly.
- add all involved output parameters by adding them one by one in the bottom part of the GUI.

Finally, the “Filters” tab is where you can filter the table results by adding a limit to the rows or a condition in the columns. the following figure shows an example of how to set the limit rows or a condition on dataset columns.

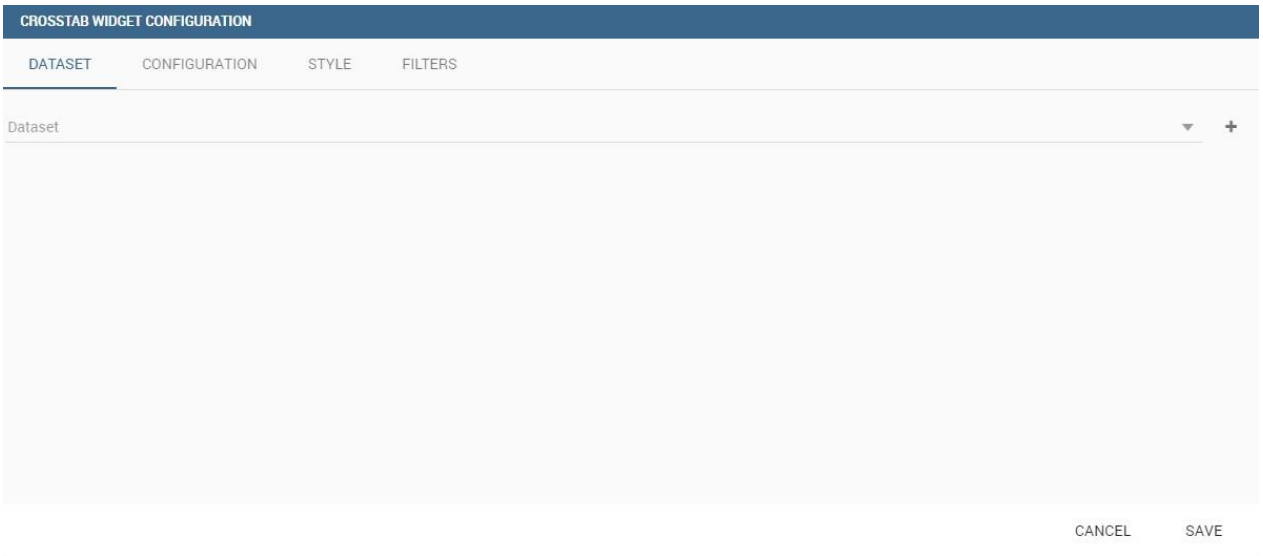


Filters tab of the table widget configuration.

Once you have finished setting the different configuration options of the table widget, then just click on “Save” and your new widget is displayed inside the cockpit.

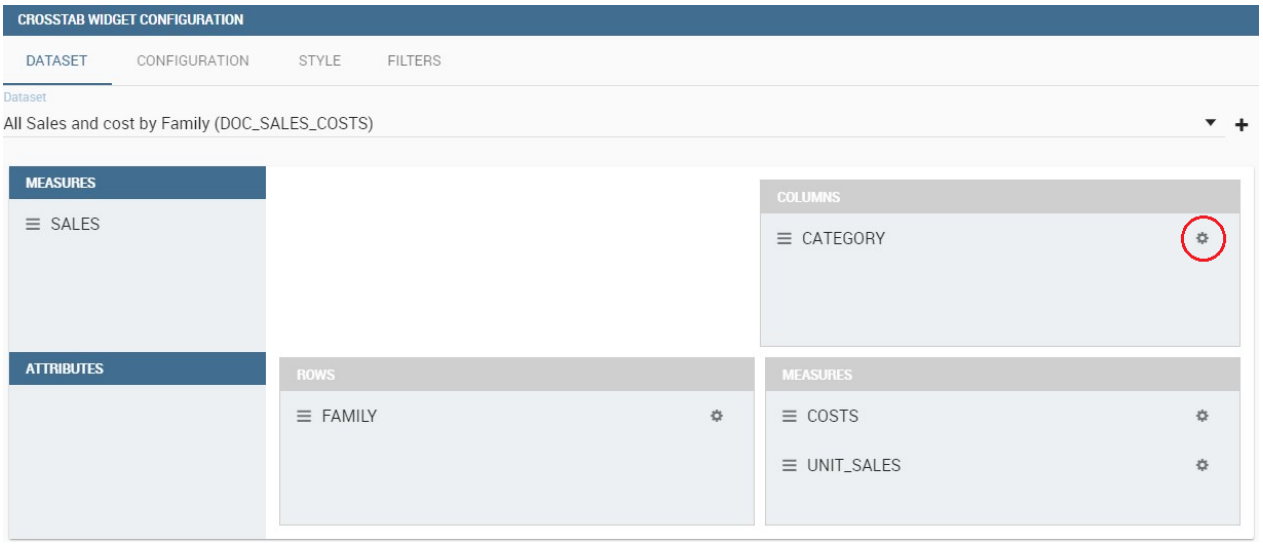
Cross Table widget

Similar configurations are available also for the Cross-Table widget. In this data visualization option, you still have the tabs: **Dataset** tab, **Configuration** tab, the **Style** tab and the **Filters** tab as you can see below.



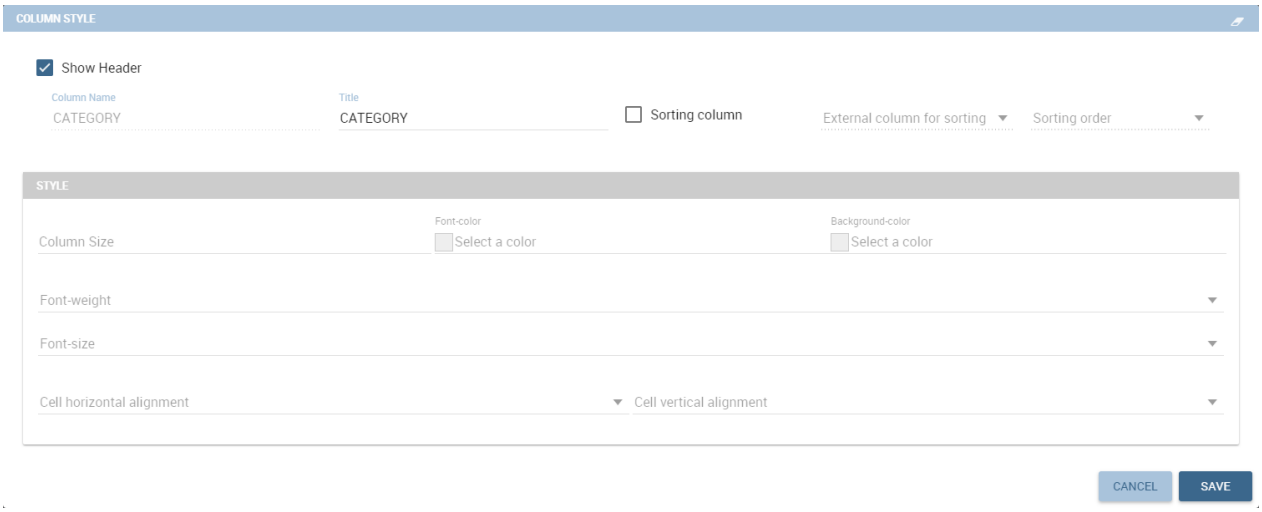
Dataset section of the crosstab widget configuration.

Using the “Dataset” tab the user can add the dataset to take values from. Consequently, it is necessary to select the fields you wish to appear as columns, those as row and measures to be exhibited in the pivot table. See figure below. Remember to set column and row fields as attributes, while measure fields as numbers.



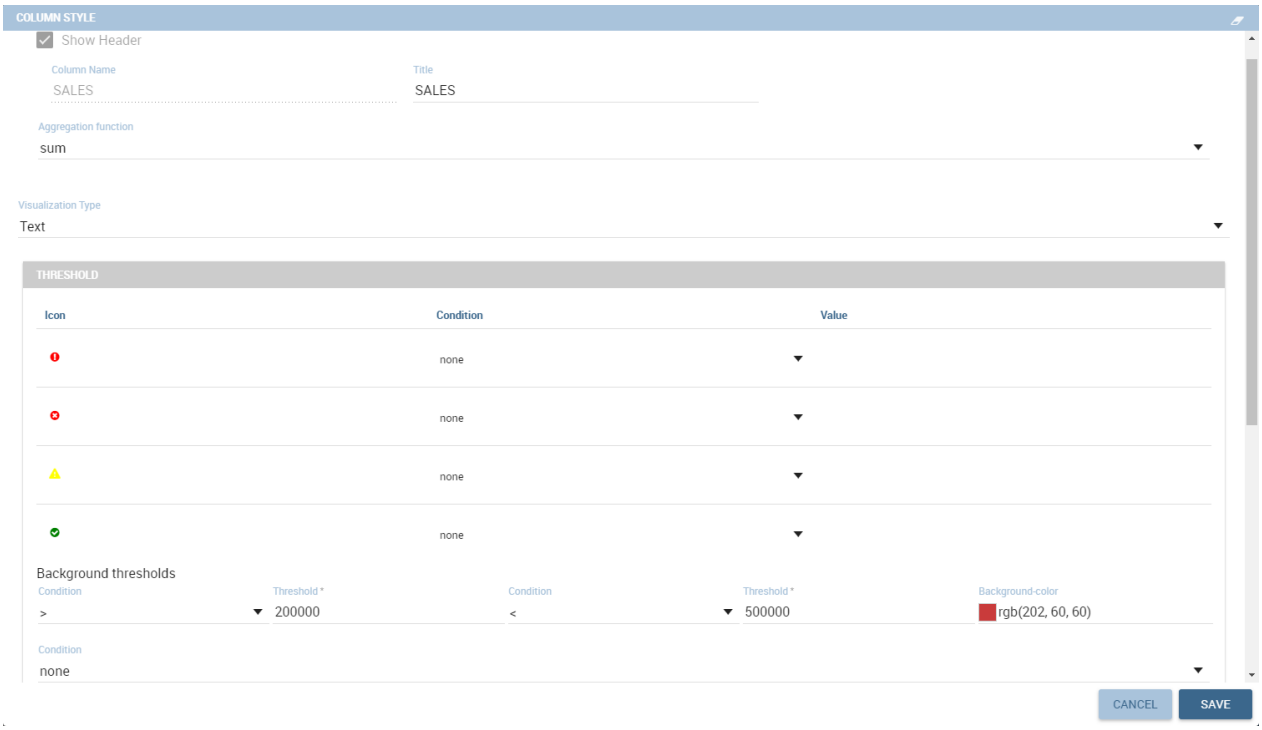
Selecting columns, rows and measures of the crosstab.

Once the columns, rows and measures have been selected the style of each column can be set by clicking on the cog settings icon. A popup will open with different options for the selected column. See figure below.



Column style popup.

It is possible to sort the crosstab according to the values of the selected column or, alternatively, according to columns not visible in the crosstab. It can also be set the style of the column, such as the font size, the font weight or the cell alignment. There is also the possibility to specify the size of the column in pixels (you can also use percent values, but it is better to use pixels).

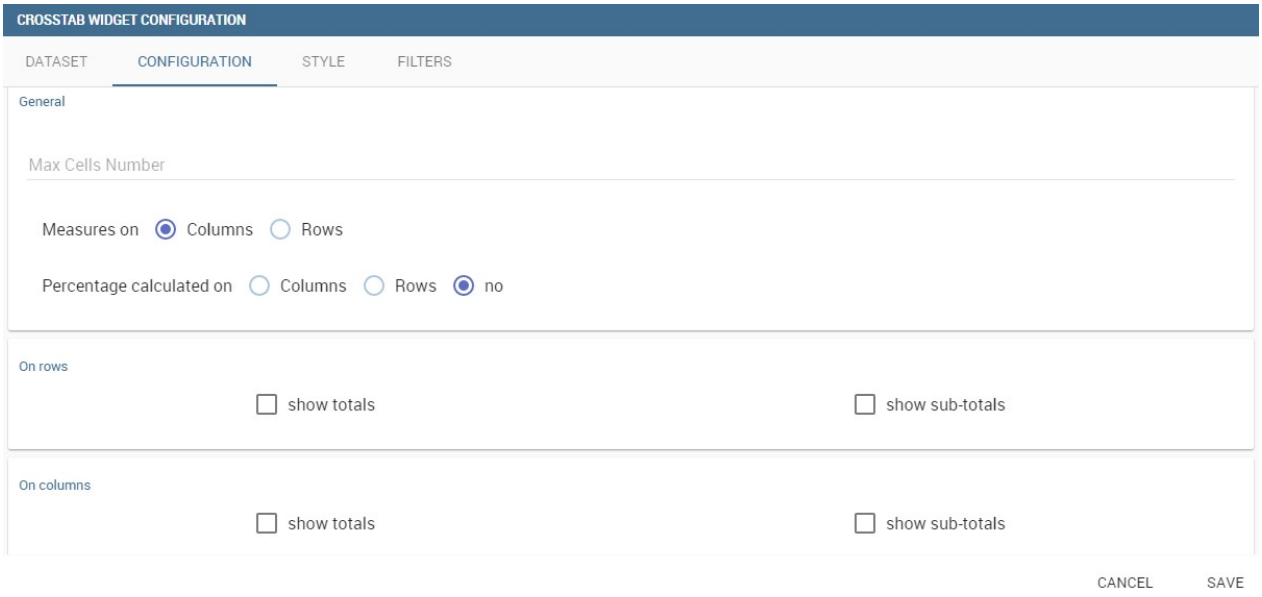


Measure column style.

In case the selected column is of type measure, as figure above shows, you can also manage threshold. It is possible to associate an icon or a specific background color to a measure's value or range.

Once the dataset has been properly configured, you can proceed to the "Configuration" tab.

The latter is made up of three sections: **General**, **On rows** and **On columns**, as Figure below shows.



Configuration tab interface.

In the "General" section you can set the following features:

- define the maximum cell number to show;
- decide to hook measures to columns or rows;
- decide to show percentages of measures related to columns or rows.

Thanks to the "On rows" feature, you can easily compute totals or subtotals on rows. Figure below exhibit an example.

	gender												
	F						M						Total
	occupation						occupation						Total
	Clerical	Management	Manual	Professional	Skilled Manual	SubTotal	Clerical	Management	Manual	Professional	Skilled Manual	SubTotal	Total
product family	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Sales
Drink	1,328.00	11,848.00	17,696.00	26,187.00	21,191.00	78,250.00	1,270.00	10,247.00	19,566.00	23,637.00	20,418.00	75,138.00	153,388.00
Food	9,817.00	97,784.00	150,429.00	216,781.00	171,188.00	645,999.00	11,982.00	86,671.00	160,463.00	198,064.00	173,250.00	630,430.00	1,276,429....
Non-Consumable	2,872.00	25,584.00	38,205.00	57,796.00	45,271.00	169,728.00	3,378.00	23,505.00	43,803.00	52,820.00	45,636.00	169,142.00	338,870.00

Computing totals and/or subtotals on rows.

Otherwise, thanks to the “On columns” feature, you can easily compute totals or subtotals on columns. Figure below exhibit an example.

		month_of_year											
		the_month	the_month	the_month	the_month	the_month	the_month	the_month	the_month	the_month	the_month	the_month	the_month
		January	February	March	April	May	June	July	August	September	October	November	December
gender	product family	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Sales
F	Drink	6,008.00	5,931.00	6,586.00	5,950.00	6,194.00	6,468.00	6,890.00	5,926.00	6,907.00	5,941.00	7,440.00	8,009.00
	Food	53,303.00	50,148.00	54,783.00	51,018.00	51,447.00	50,352.00	55,768.00	51,517.00	54,231.00	48,437.00	59,720.00	65,275.00
	Non-Consumable	13,449.00	13,572.00	14,453.00	13,253.00	13,490.00	13,063.00	14,488.00	13,632.00	14,032.00	13,512.00	15,768.00	17,016.00
	SubTotal	72,760.00	69,651.00	75,822.00	70,221.00	71,131.00	69,883.00	77,146.00	71,075.00	75,170.00	67,890.00	82,928.00	90,300.00
M	Drink	6,400.00	6,069.00	6,195.00	5,795.00	5,868.00	6,136.00	5,972.00	5,991.00	6,260.00	5,843.00	7,146.00	7,463.00
	Food	50,846.00	50,357.00	52,360.00	49,271.00	49,477.00	52,678.00	51,837.00	50,604.00	50,471.00	47,932.00	61,425.00	63,172.00
	Non-Consumable	13,953.00	12,838.00	14,216.00	12,749.00	13,546.00	13,980.00	14,576.00	13,570.00	13,639.00	13,408.00	16,089.00	16,578.00
	SubTotal	71,199.00	69,264.00	72,771.00	67,815.00	68,891.00	72,794.00	72,385.00	70,165.00	70,370.00	67,183.00	84,660.00	87,213.00
Total		143,959.00	138,915.00	148,593.00	138,036.00	140,022.00	142,677.00	149,531.00	141,240.00	145,540.00	135,073.00	167,588.00	177,513.00

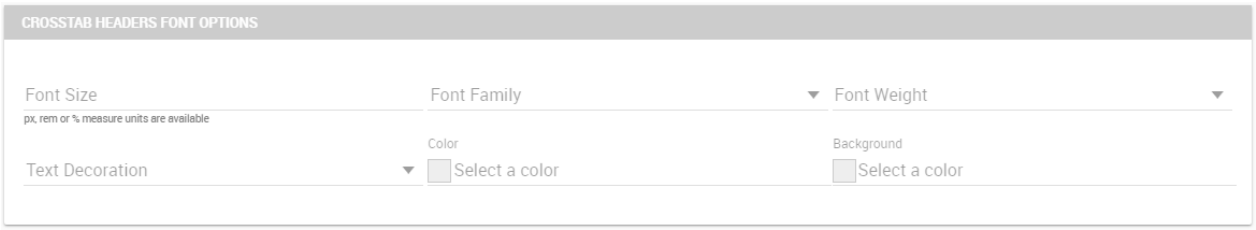
Computing totals and/or subtotals on columns.

Switching to the “Style” tab you can find the general style settings available for the crosstab.

- **Crosstab Font General Options** where font and font size are set;

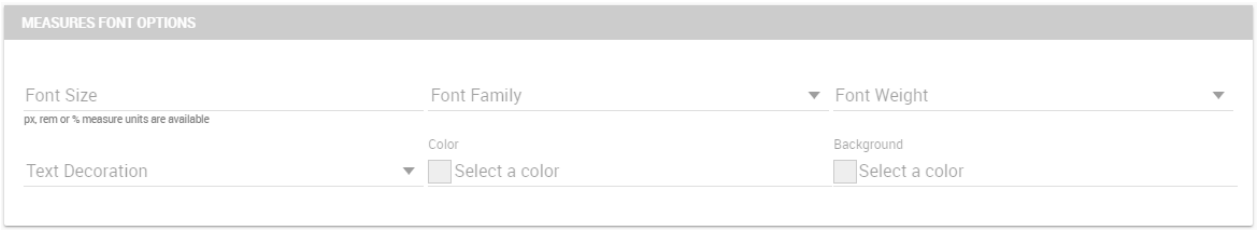
General style options for crosstab.

- **Crosstab Headers Font Options** where you can configure the header style settings as color, background, font, etc.



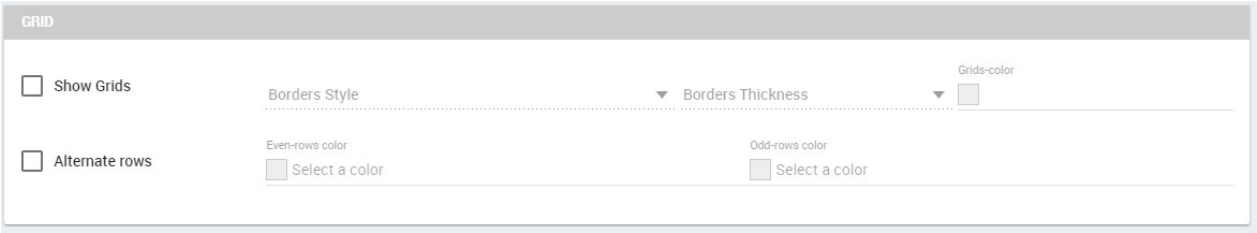
Crosstab Headers Font Options for crosstab.

- **Measures Font Options** where you can configure several style options for measures, such as color, background, font size, etc.



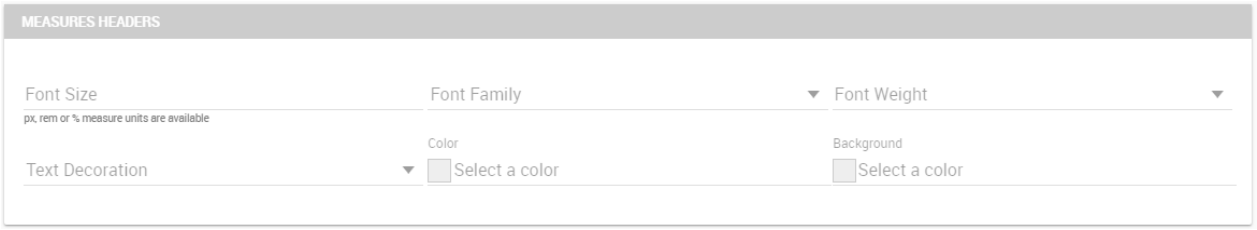
Measures Font Options for crosstab.

- Using the **Grid** section, you can mark (or not) grid borders, decide for border style, thickness and color. You can also alternate row indicating different colors.



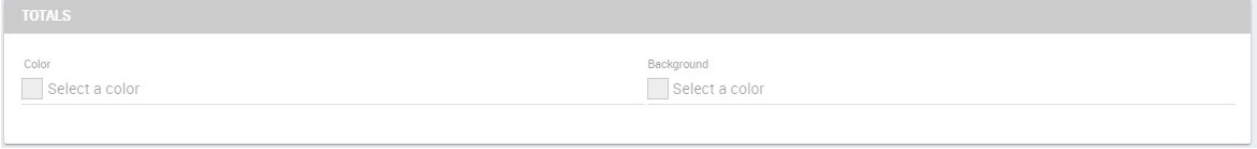
Grid Options for crosstab.

- In the **Measures Headers** section, you can configure different style option for measure headers, such as color, background, font size, etc.



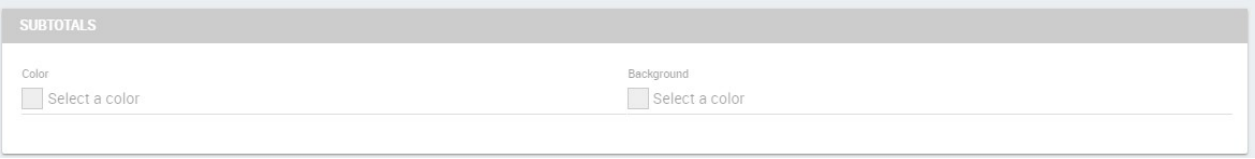
Measures Headers Option for crosstab.

- In the **Total** section you can set color and background of totals (if any).



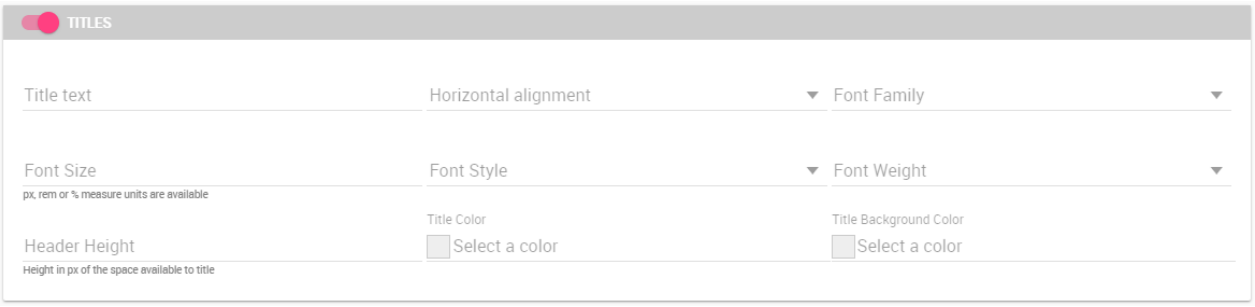
Color settings for Totals.

- In the **Subtotal** section you can set color and background of subtotals (if any).



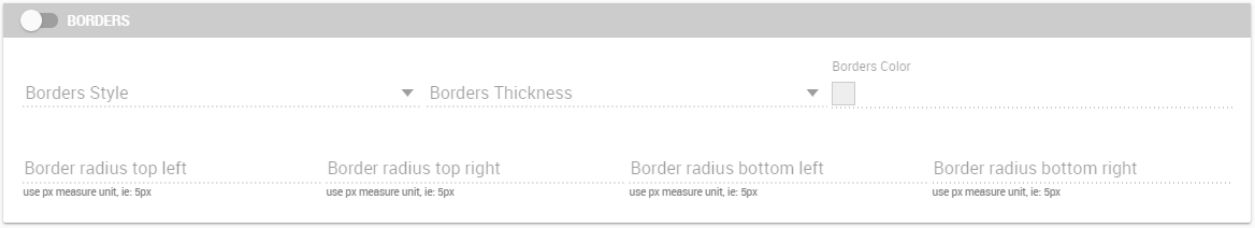
Color settings for Subtotals.

- In the **Titles** section you can add titles to widget and customize them using different styles.



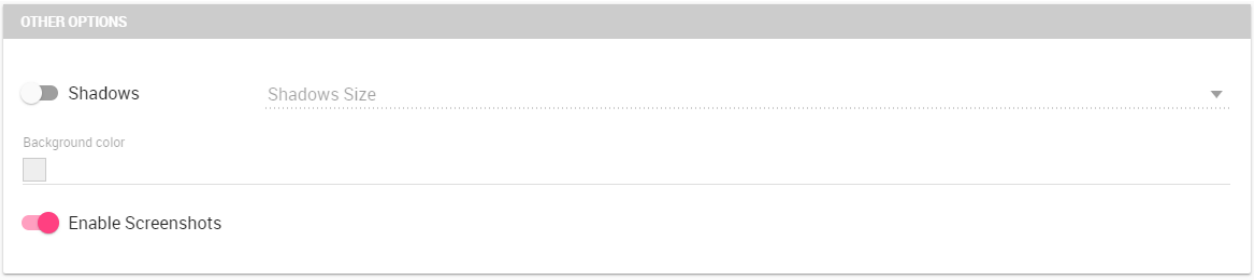
Title settings.

- In the **Borders** section you can add borders to widgets and customize them using different styles.



Border settings.

- In the **Other Options** section, you can add a shadow to widget layout and indicate its measure, color the widget background at convenience and it is possible to disable or enable the screenshot option for that widget.



Other Options for crosstab.

Once some or all (at least the mandatory) of the above-mentioned setting features have been set you can save, and the widget will be inserted into the cockpit area.

Document section

The Document widget allows to add an external document into the cockpit area. This widget supports documents like reports, graphs, maps, etc.

Use the Data configuration button to add a document source to the cockpit. Click on the “Plus” icon on the right half of the page to choose among all available documents.

The Document Widget configuration is divided into two parts: **Custom** tab and **Style** tab as you can see from Figure below.

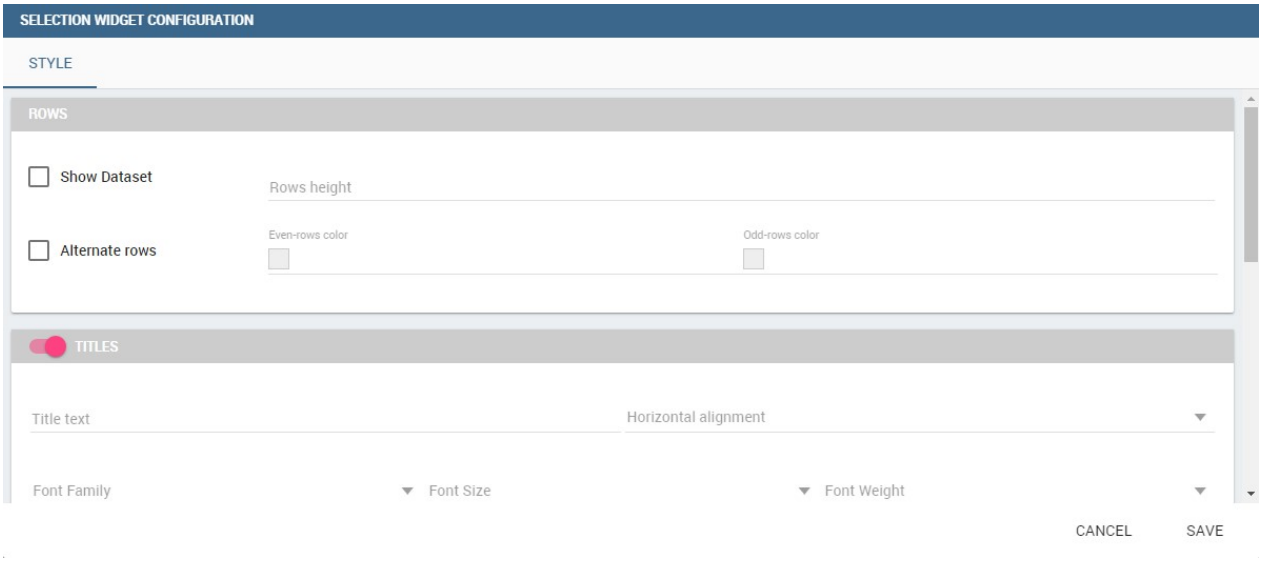
Custom tab of the Document widget.

The Custom tab is the place where the document is uploaded while the Style tab is where all style options are set.

Selection widget

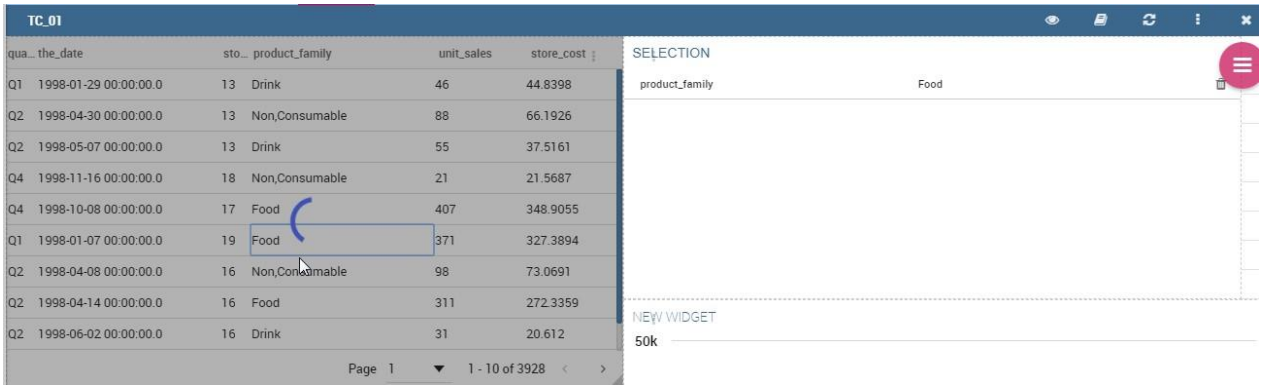
This widget is related to the association concept so in this subsection we give information on how to add and custom the **Selection Widget** into the cockpit area and its functioning, while we refer to the dedicated Document section for details on how to set (global) associations.

To enable the Selection widget, which means the possibility to have all associations listed and accessible on a widget, the user must open the “Selection” feature through the “Add widget” functionality and configure the demanded options. Figure below shows the “Selection widget configuration” interface.



Selection widget configuration.

The Selection Widget will display the elements selected by the user. Figure below shows an example.

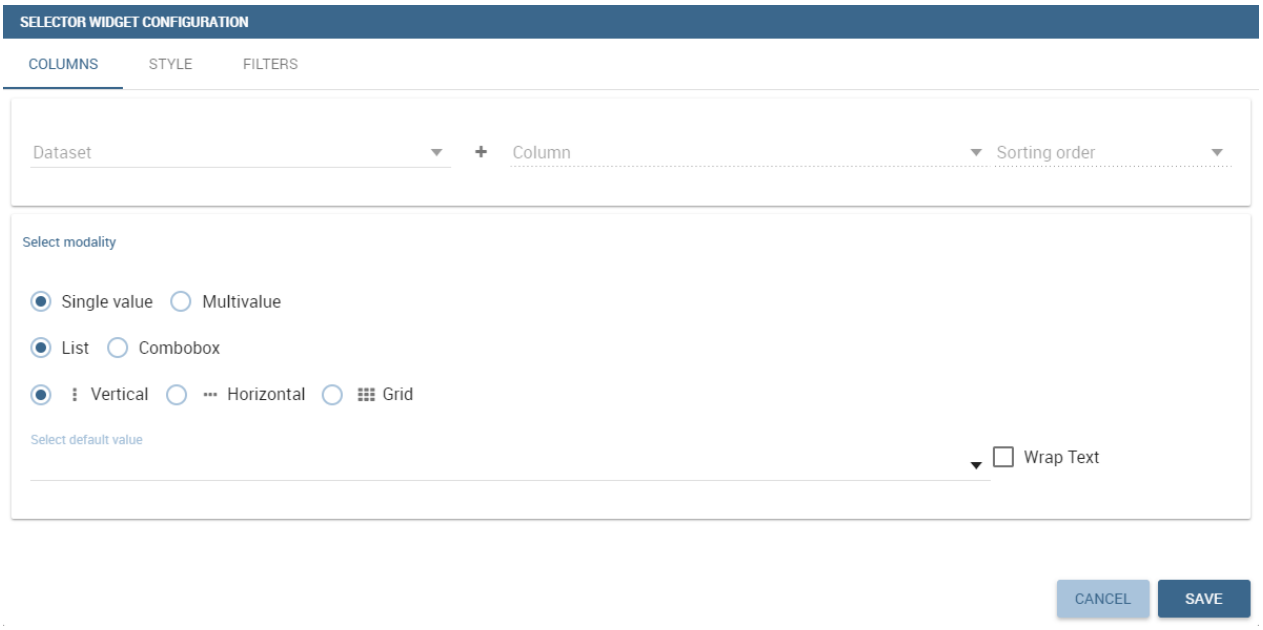


Selection widget outlook.

If global associations have been set, clicking on table, cross table or chart elements will update all corresponding widgets. Otherwise, only the widget on which selection has been made Selector Widget will be updated. In both cases the Selection widget will display the highlighted attribute values.

Selector Widget

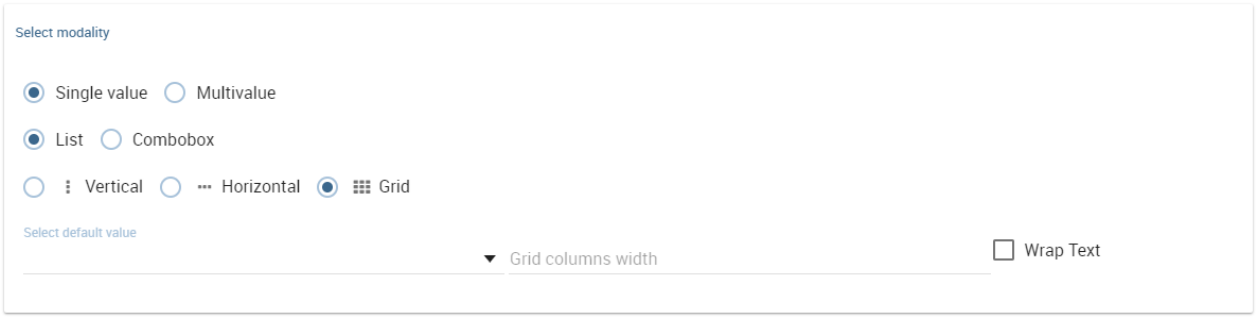
The **Selector Widget** is useful when an end user (a user with a USER tole type) wants to add a parameter to the document.



Selector widget outlook.

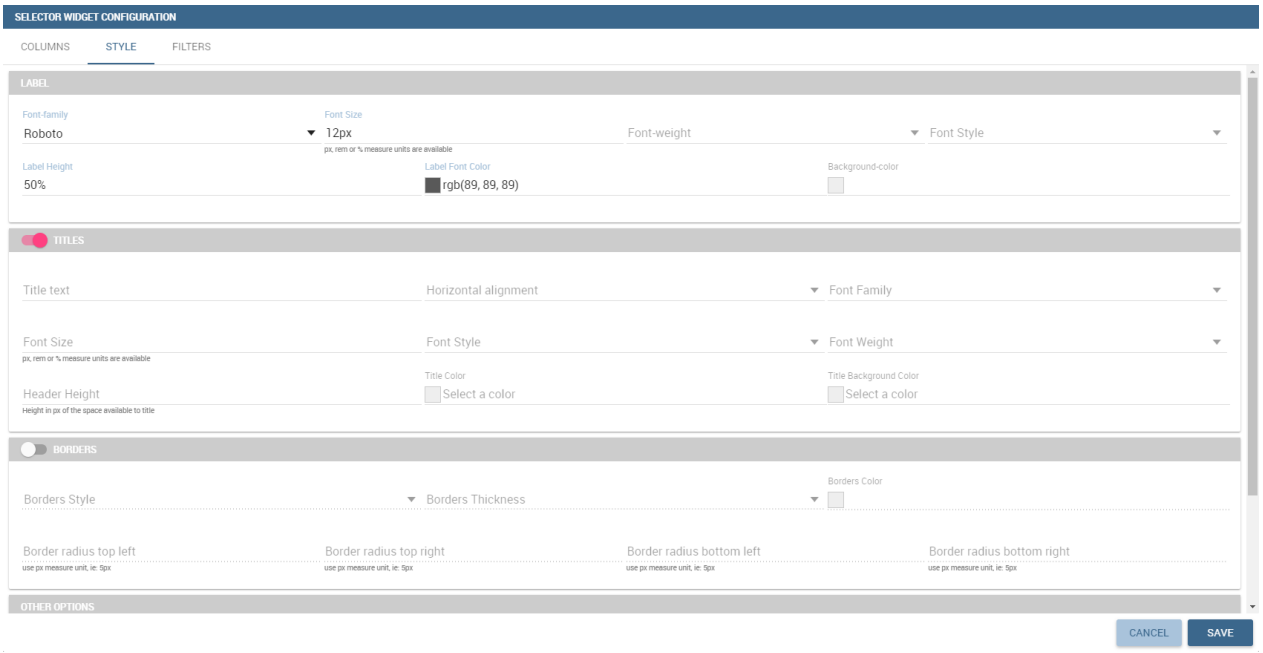
In detail, use the **Columns** tab to select the dataset and the dataset column on which you want to apply the filter. Then custom the **Select modality** options; for instance, choose between single or multivalue or to use a list or a combo box. Note that for the list option you can further choose among “vertical”, “horizontal” or “grid”. You can also decide to add a default value, chosen from main column’s first item, main column’s last item or to simply assign a static value. Finally, by clicking on the Wrap Text option it is possible to wrap the text shown in the selector; this option is useful when the categories to choose from are sting of long dimensions.

In the case of the selector of type list “grid” it is also possible to set the grid columns width.



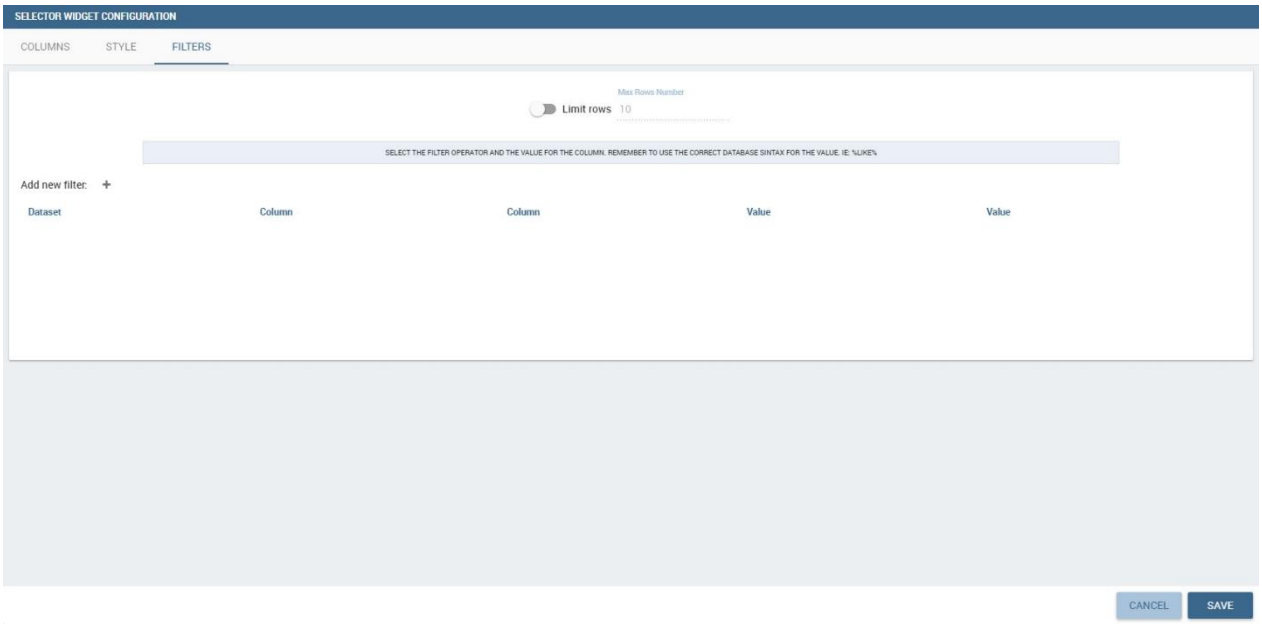
Grid columns width.

Move to the **Style** tab to set the widget style in terms of label, titles, borders, shadows and background color. Figure below shows a customization example.



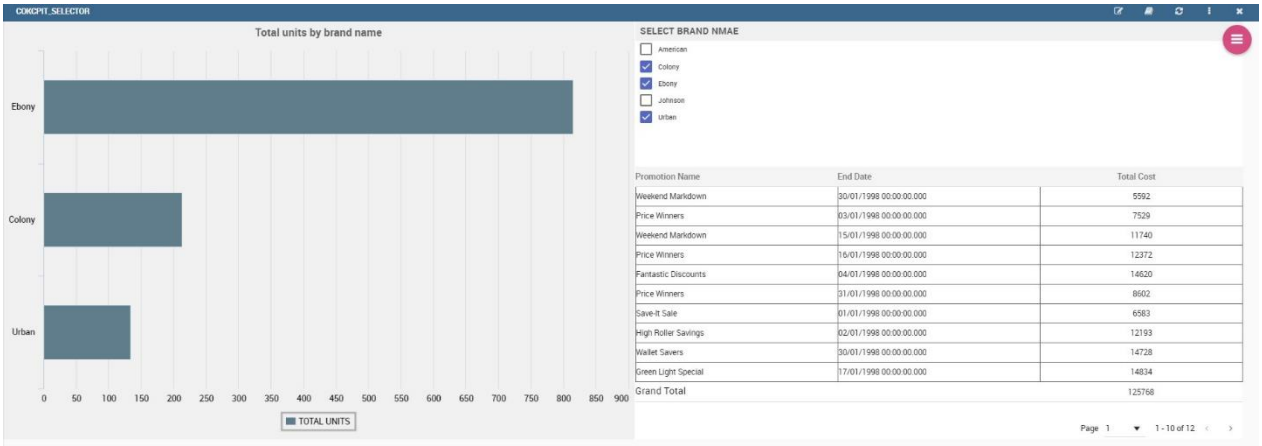
Selector widget configuration.

Finally use the **Filters** tab to handle pagination or filter on a dataset column.



Selector filters.

The Selector widget works effectively as a ready-to-use filter panel.



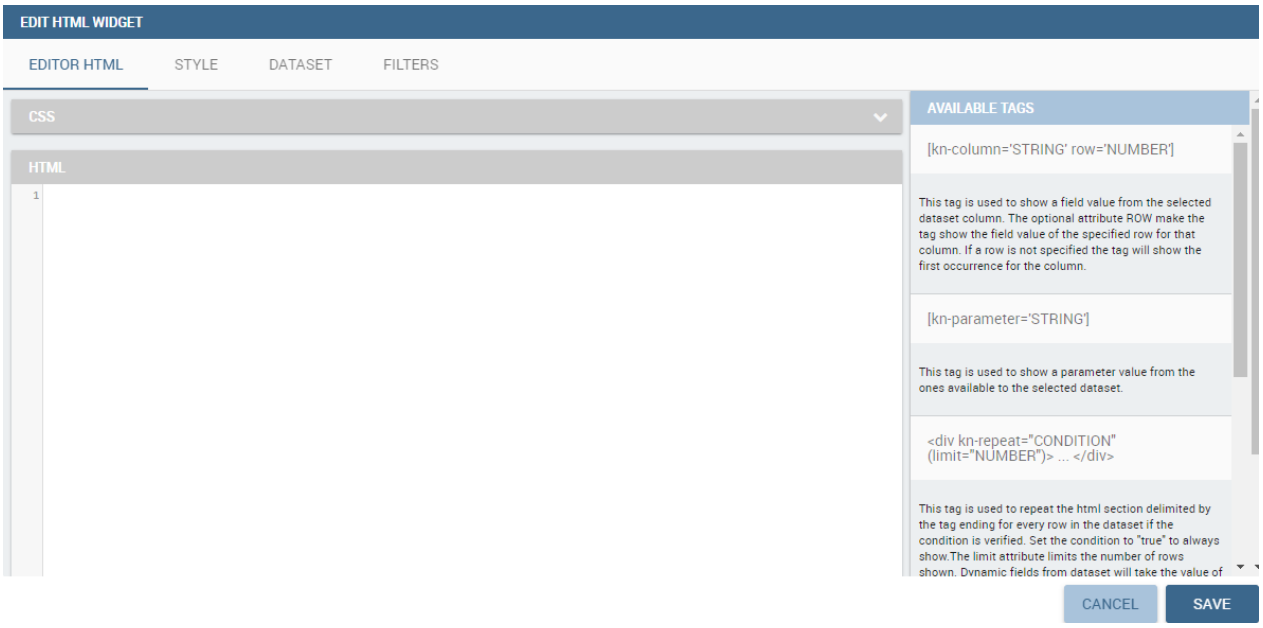
Selector widget execution example.

HTML Widget

The HTML widget allows to add customized HTML and CSS code to add very custom dynamic elements to the cockpit. This widget supports all HTML5 standard tags and CSS3 properties.

The Edit section of the widget is composed by tree tabs: the HTML editor, the style and the dataset. In the editor tab is possible to add the code that will be shown in the widget.

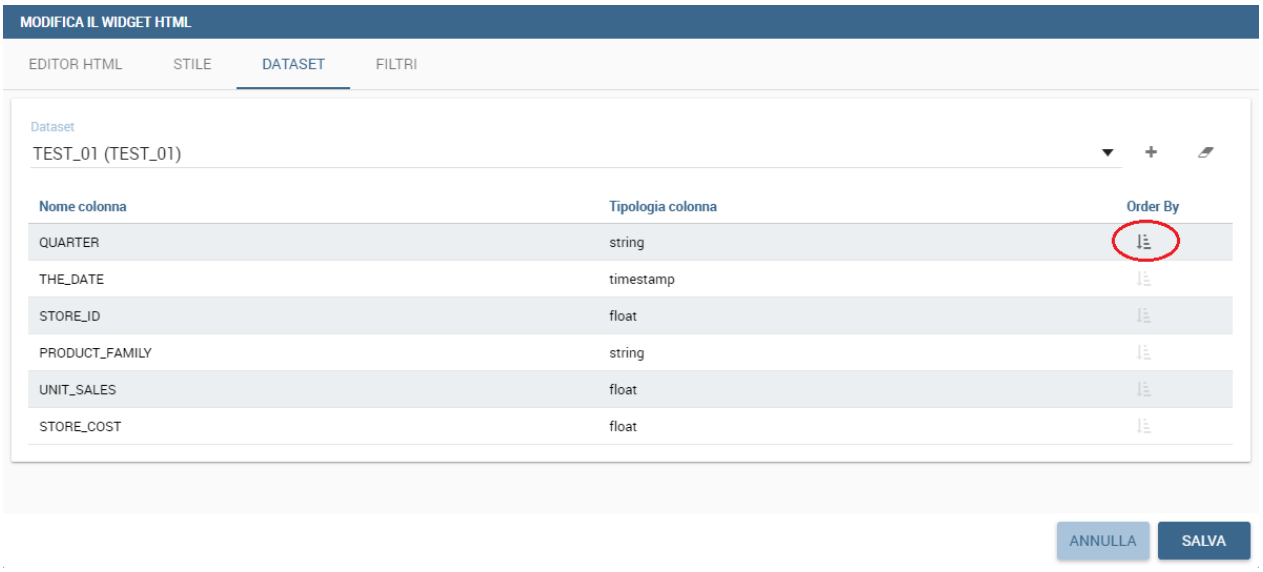
Clicking on the top expander section in the tab, the one named “CSS” also the CSS editor will be available.



HTML widget editor

In the right side of the editor is possible to take available tags to copy inside the code, those tags will be explained in detail in the following paragraphs. Is not possible to add custom JavaScript code inside the html editor, so the available tags are the tools to make the widget dynamic and to use the dataset data.

The Dataset tab allows the user to select a dataset to make the Widget dynamic and to bind it to dataset data. After choosing a dataset the list of available columns will be show. Those names will be useful inside the dynamic tags. Here it is also possible to order the dataset according to a column and to select the ordering type (ascending or descending).



Dataset selection

By clicking on the icon  of a specific column the dataset will be ordered by that column by default by ascending order. In order to select the descending ordering type, you must click another time on the icon (the icon will be now like this ).

Available Tags

```
[kn-column='COLUMN-NAME' row='COLUMN-ROW-NUMBER' aggregation='COLUMN-AGGREGATION' precision='COLUMN-DECIMALS']
```

The `kn-column` tag is the main dynamic HTML Widget tool, it allows to select a column name from the selected dataset and to print its value. The value of the kn-column attribute should be the name of the column value you want to read in execution.

The **row** attribute is optional and is a number type attribute. If no row is selected the first-row column value will be shown.

The **aggregation** attribute is optional and is a string type attribute. If inserted the value shown will be the aggregation of all column rows values. The available aggregations are: AVG|MIN|MAX|SUM|COUNT_DISTINCT|COUNT|DISTINCT COUNT.

The **precision** attribute is optional and is a number type attribute. If added and if the result value is a number, the decimal precision will be forced to the selected one.

```
[kn-parameter='PARAMETER-NAME']
```

The kn-parameter tag is the tool to show a dataset parameter inside the widget execution. The value of the kn-parameter attribute should be the name of the set attribute.

```
[kn-calc=(CODE-TO-EVALUATE) precision='VALUE-PRECISION']
```

The `kn-calc` tag is the tool to calculate expressions between different values on widget execution. Everything inside the brackets will be evaluated after the other tag's substitution, so will be possible to use other tags inside.

The **precision** attribute is optional and is a number type attribute. If added and if the result value is a number, the decimal precision will be forced to the selected one.

```
<div kn-repeat="true" limit="LIMIT-NUMBER"> ... REPEATED-CONTENT ... </div>
```

The `kn-repeat` attribute is available to every HTML5 tag and is a tool to repeat the element for every row of the selected dataset.

This attribute is naturally linked to `kn-column` tag. If inside a `kn-column` tag without a row attribute is present, the `kn-repeat` will show the column value for every row of the dataset.

Inside a `kn-repeat` is possible to use the specific tag `kn-repeat-index`, that will print the index of the repeated column row.

The **limit** attribute is optional and is a number type attribute. If added the number of rows repeated will be limited to the selected number.

```
<div kn-if="CODE-TO-EVALUATE"> ... </div>
```

The `kn-if` attribute is available to every HTML5 tag and is a way to conditionally show or hide an element based on some other value. The attribute content will be evaluated after the other tags' substitution, so will be possible to use other tags inside. If the

evaluation returns true the tag will be shown, otherwise it will be deleted from the execution.

```
<div kn-cross> ... </div>
```

The `kn-cross` attribute is available to every HTML5 tag and is a way to make the element interactive on click. This attribute generates an on click event on the element to open the cross-navigation set. If there is no cross navigation set this tag will not work.

```
<div kn-preview="DATASET-TO-SHOW"> ... </div>
```

The `kn-preview` attribute is available to every HTML5 tag and is a way to make the element interactive on click. This attribute generates an on click event on the element to open the dataset preview dialog. The attribute value will be the *dataset label* of the dataset that you want to open. If a dataset is not specified, the cockpit will use the one set for the widget. If no dataset has been set and the attribute has no value this tag will not work.

```
<div kn-selection-column="COLUMN-NAME" kn-selection-value="COLUMN-VALUE"> ... </div>
```

The `kn-selection-column` attribute is available to every HTML5 tag and is a way to make the element interactive on click. This attribute generates an on click event on the element to set the chosen column and value in the cockpit selections. The default will use as a selection the first-row value for the column.

The **kn-selection-value** attribute is optional and will add a specific value to the column selection.

Banned Tags

For Cross side scripting and security reasons some tags are removed on save by the security filter:

- `<button></button>`
- `<object></object>`
- `<script></script>`

If the tag is needed for some specific behaviour (i.e. the button default hover), please replicate it with CSS using a different allowed tag.

```
<?xml version="1.0" encoding="UTF-8"?>
<WHITELIST>
   <service baseurl="https://www.youtube.com" />
   <service relativepath="/knowage/themes/" />
</WHITELIST>
```

Like other widgets the “Style” tab and the “Filters” tab are available in order to set the general style options for the widget and to filter the results displayed in the HTML widget.

Widget properties

Once one or more (above mentioned) widgets have been implemented, the technical user has some more options exploring the icon available at the right top corner of the widget itself, as Figure below highlights.

Widget properties.

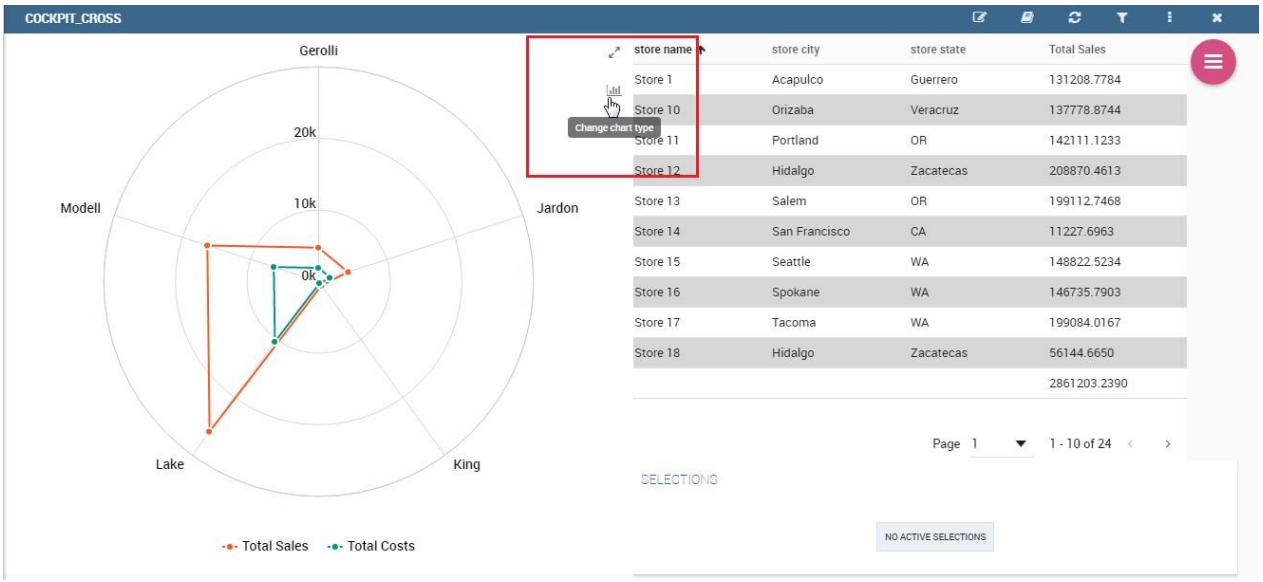
Here the user can:

- move the widget in the cockpit area at convenience;
- modify its dimension;
- delete it;

- activate the on-click interaction of the widget with the other ones;
- activate the updating of widget data due to the interaction with other widgets.

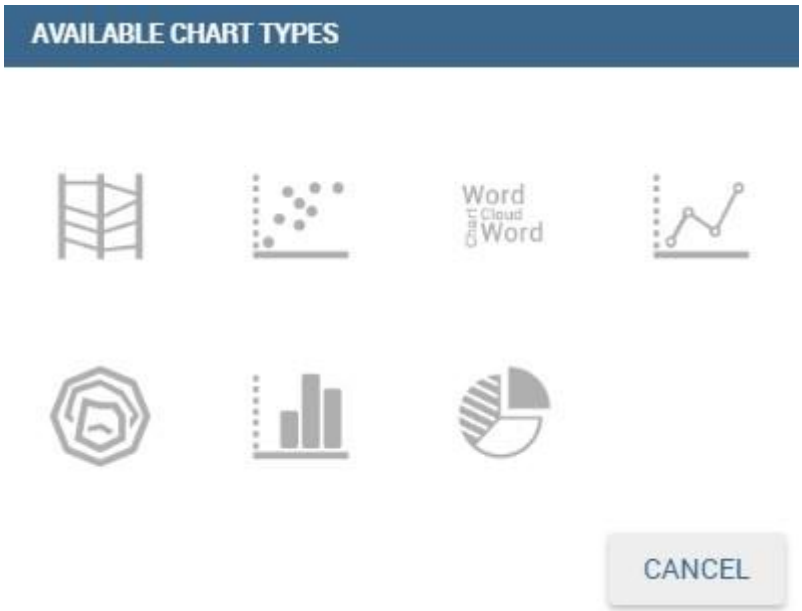
When executing the cockpit in visualization mode, the user has also some more options for widgets. For all widget the user can use the icon to expand the widget to all page and use the icon to reduce it again. There are also two new widget options: using the icon it is possible to capture the screenshot of the widget and clicking on the icon the data plotted on a chart or displayed in a table or crosstab are exported in an excel file.

Chart widget are endowed with an additional option that allows the user to change the chart type, as you can see in Figure below.



Change chart type button.

Referring to figure below, the available chart types are parallel, scatter, Wordcloud, line, radar, bar and pie.

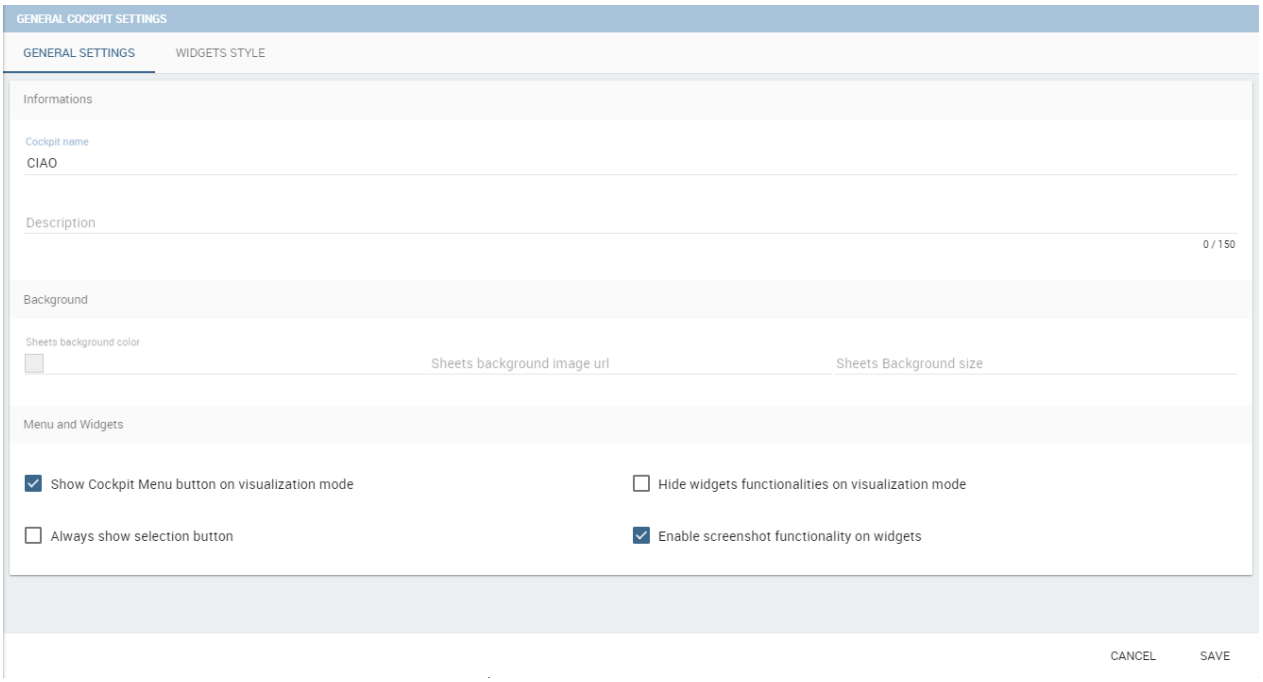


Available chart types.

Pay attention though to the fact that when grouping functions have been used, the change chart type may not report the same level of aggregation. In fact, not all type of chart allows the grouping function. Refer to Chart types in detail to read more about each chart type configuration. Pay also attention when a two-series chart is changed with a single-series one. For instance, the parallel chart works only when (at least) two series have been set, while the Wordcloud works with only one series.

General configuration

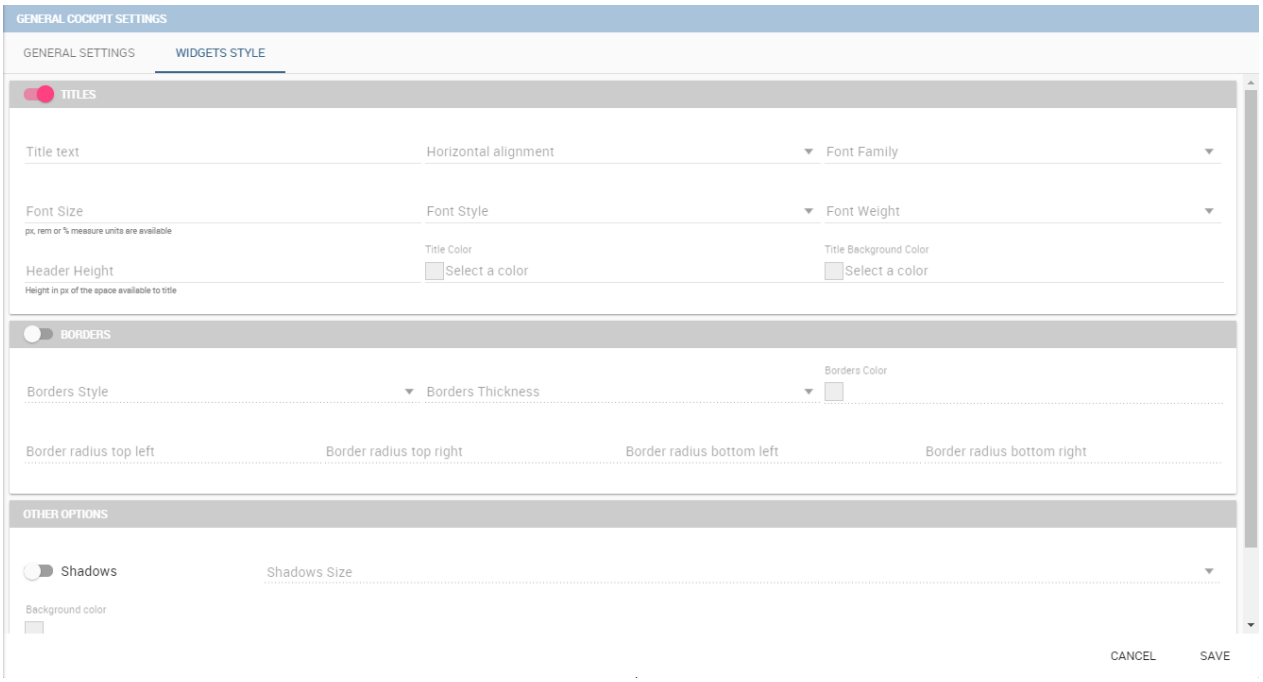
This option allows the user to manage all cockpit general settings that we are going to describe through images of the interface. Clicking on the **General configuration** button the window in figure below opens. This contains the **General Settings** tab and the **Widget Style** tab.



General configuration window.

Editing the fields of the first tab you can add or change the name and/or the description of your cockpit; moreover, here you can choose the sheet color or a background image and its size. In order to add a background image for the sheets, firstly you must add the image to the catalogue of the image widget and then copy the link of the image. It is also possible to decide to enable the menu and the widgets functionalities when the document runs in display mode or to disable the screenshot functionality for every widget.

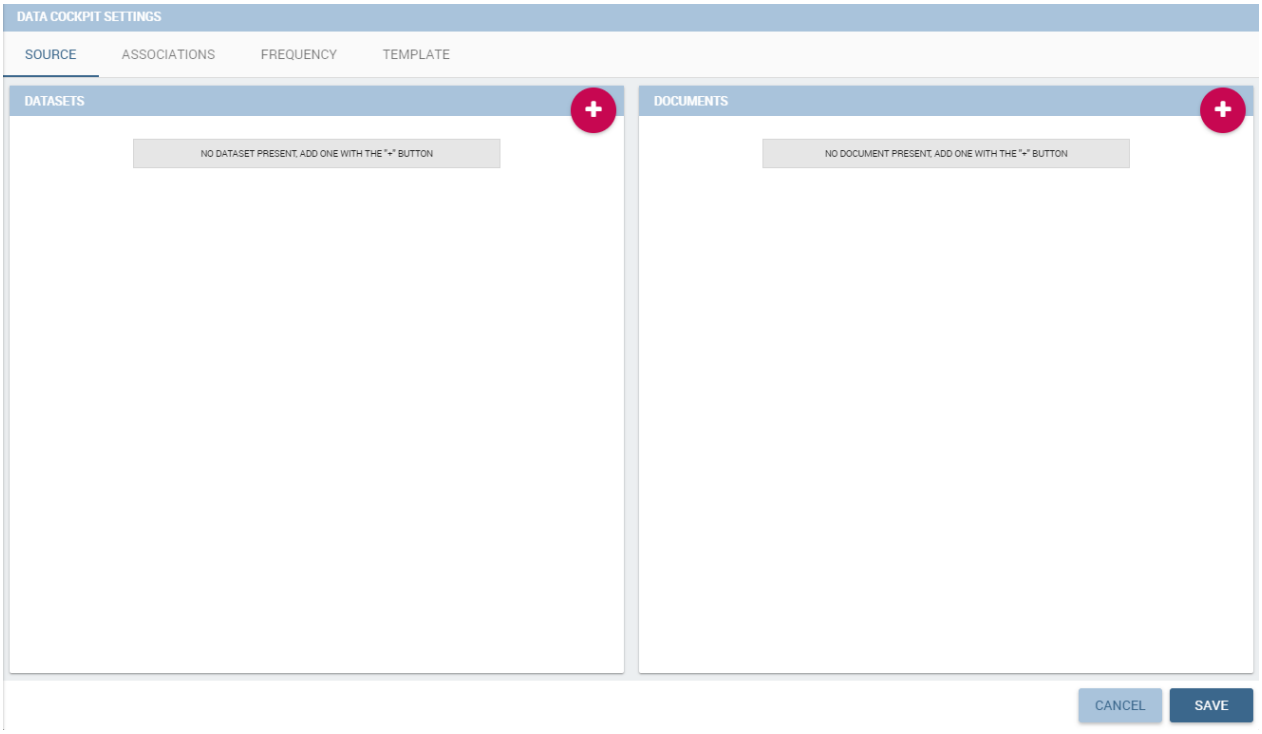
The second tab (Figure below allows to configure some style options of the cockpit, like borders, shadows, titles and background color.



Widget style tab.

Data configuration

This feature manages the data storage and usage. In fact, here there is the possibility to save data in cache, create associations between datasets, schedule the (data) refresh frequency and so on. Referring to the figure below, the feature is implemented through several tabs: The **Source** tab, the **Associations** tab, the **Frequency** and the **Template** tab.



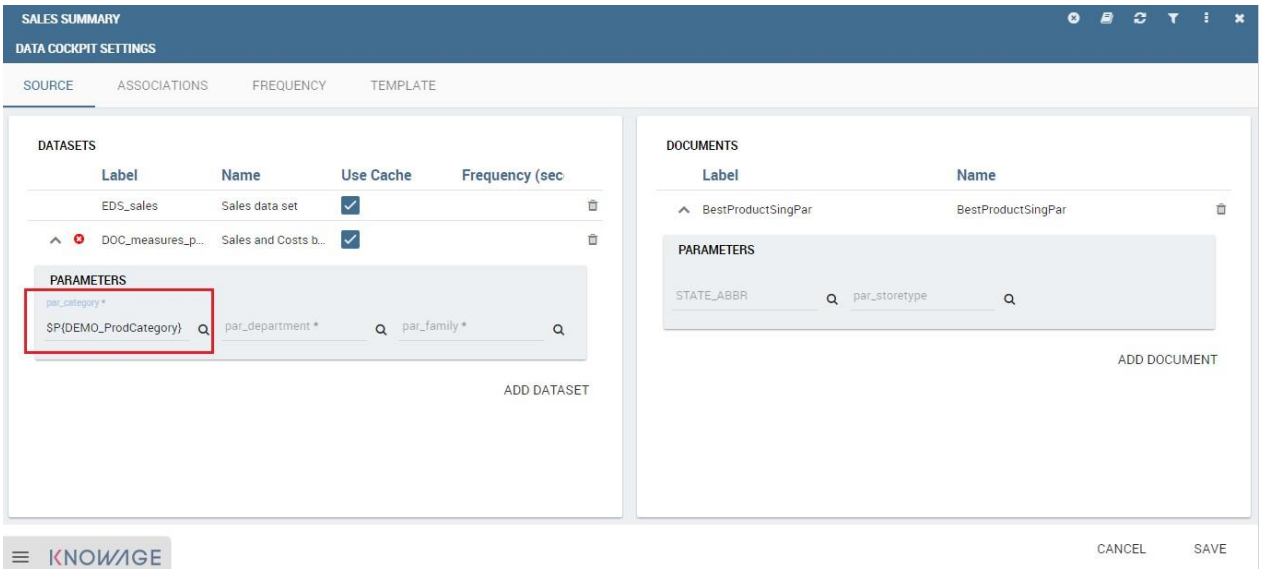
Data configuration window.

Source

The Source tab is split into two areas. On the left side the user can find the list of those dataset that are currently used by the cockpit. Here it is possible to add new dataset that will be passed to widgets. In other words, datasets inserted in this area will be listed in the dataset combo box of widgets like the Table, the Pivot Table and the Chart one. Note that the user can delete datasets as well.

Parametric sources management

If the user is adding a parametric dataset the window will exhibit them in an expandable box right below. It is also mandatory to give default values or to associate proper drivers to the document to secure its correct execution. By the way, a final user has no access to parametric dataset, and he/she cannot handle analytical drivers, therefore **parametric sources can be managed only by an admin user**. We stress that the user must also type the driver name in the field box as highlighted in Figure below. You can type it manually or use the look up just aside the parameter line.



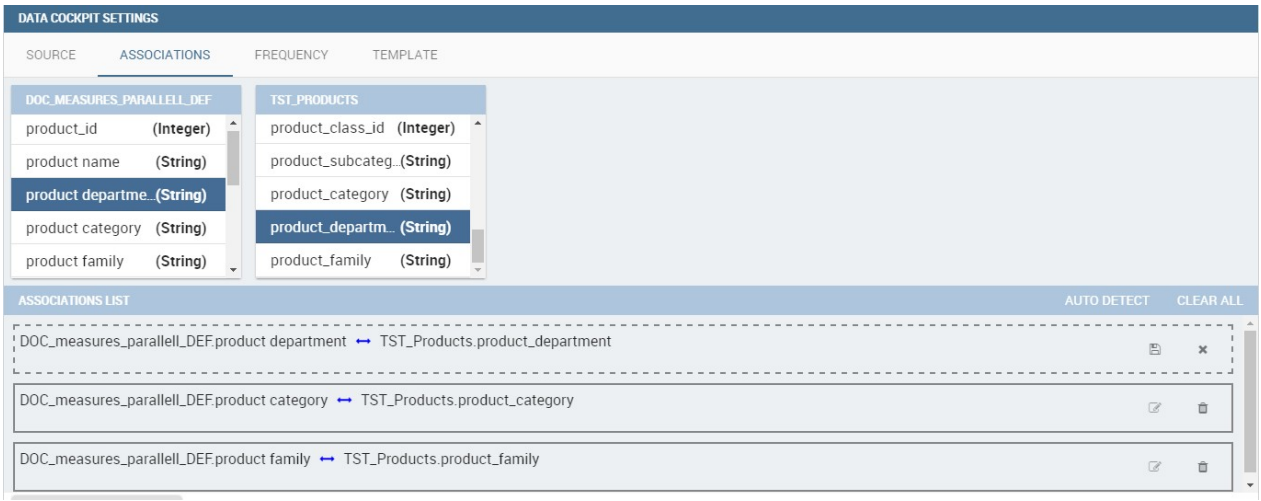
Dataset management.

On the right side of the window the user finds the list of external documents that can be added to the cockpit (through Document widgets), or as well as for the dataset case, of documents that are already in use in (previously set) Document widgets. In the occurrence of Associations parametric documents, parameter boxes are shown below. Note that it is mandatory to link them to analytical drivers (previously hooked to the document) or be assigned a fixed (default) value.

Associations

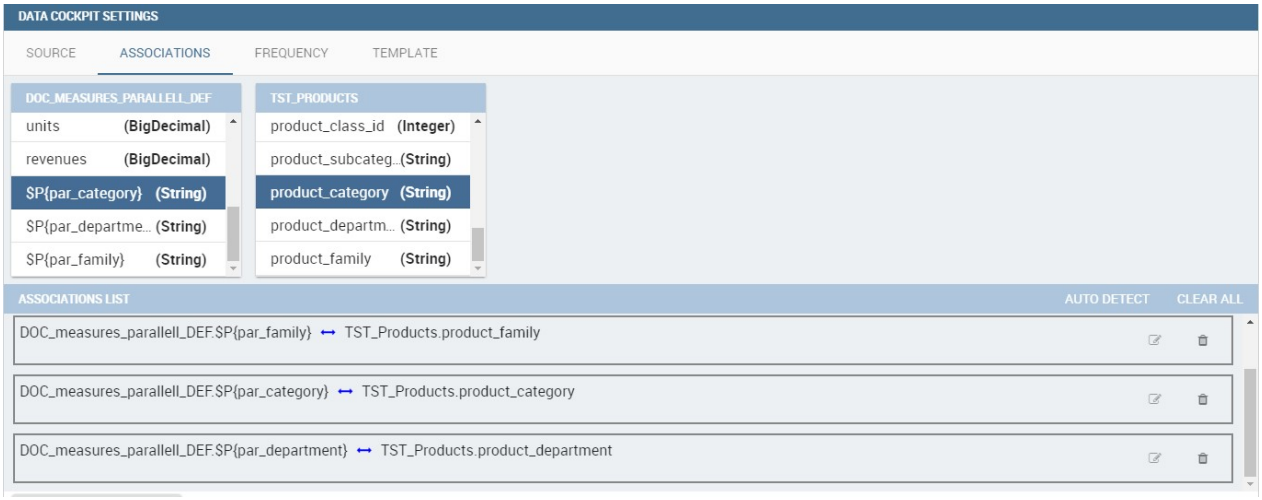
If your goal is to show data from a single dataset, it is not necessary to define any association. *Associations should be set within the designer when widgets are built on different datasets*. Associations can be set with the elements: dataset columns, dataset parameters and document parameters. Note that to implement an association the user must have at least one column. We show some examples in the following.

The following figure shows the association between two datasets. In this case the user must detect one field from the first dataset, the same field (in terms of values) in the other one. The relation will appear right below. Click on the save button to confirm the association. If the associations rely on multiple columns the user must add them one by one.



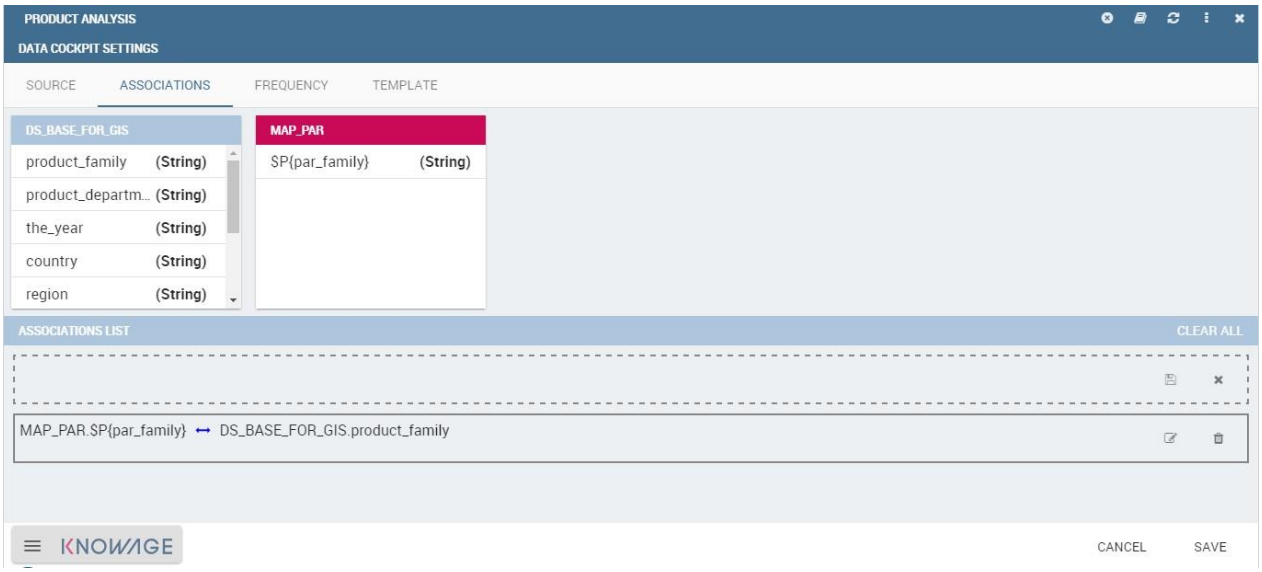
Associations between dataset columns.

The same procedure can be done in the case of dataset columns and dataset parameters, as shown below.



Associations between dataset column and dataset parameter.

Another example is supplied in Figure below. Here the association is performed between a dataset Frequency column and document parameter.

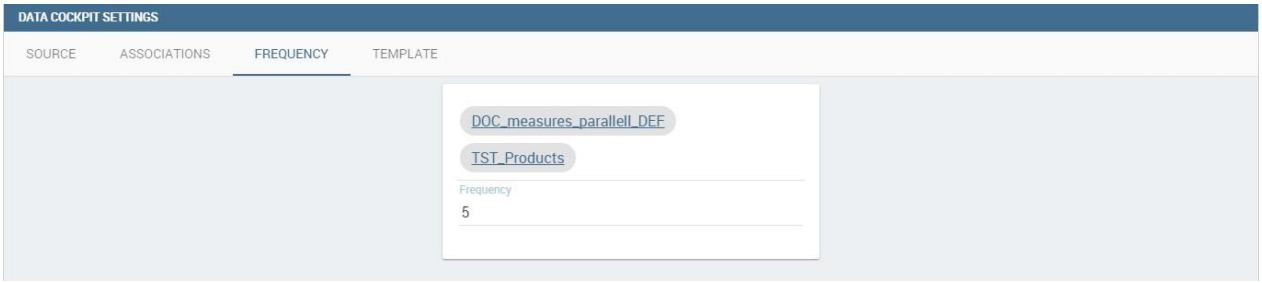


Associations between dataset column and document parameter.

Once you have defined the associations, as soon as you refresh one widget, all related widgets are refreshed simultaneously on data update.

Frequency

The Frequency tab defines a schedule over dataset involved in the associations. An example is supplied in the next figure. This means that associations are activated automatically, and data are reloaded according to this feature. Groups of real-time datasets that compose one or more associations can have different update frequencies. We stress that, in order to secure the right document execution, the group frequency do not affect the other ones and each group is reloaded at different times. In addition, real-time dataset that are not involved in any association can have their own frequency.



Frequency settings example.

Template

In this tab the user can find the json code (at the current stage of the work) which composes the template. Figure below shows an example.

Template example.

Selections

Adding the **Selections** to your widgets, namely the possibility to reload all widget data according to selection made through the click on a specific item of the cockpit (cell value, chart bar, etc.). Moreover, thanks to this functionality the user can reproduce the drill down feature that we introduced in Chapter of Chart. You can check which selections are active on your cockpit at anytime thanks to the **Selection** functionality. In Section 7.1 we already described how to add the “Selection” widget inside the cockpit

area. If the user does not wish for the widget to stay visible, selections can still be accessed and managed through the menu configuration bar. Clicking on the “Selection” menu icon you can enter the “Selections” window. Here all selections and associations are listed, as shown in Figure below. The “Delete” button is available just aside each row to let the user to remove that specific selections. Click on the “Cancel” button to exit the window.

Selection window.

Clear cache

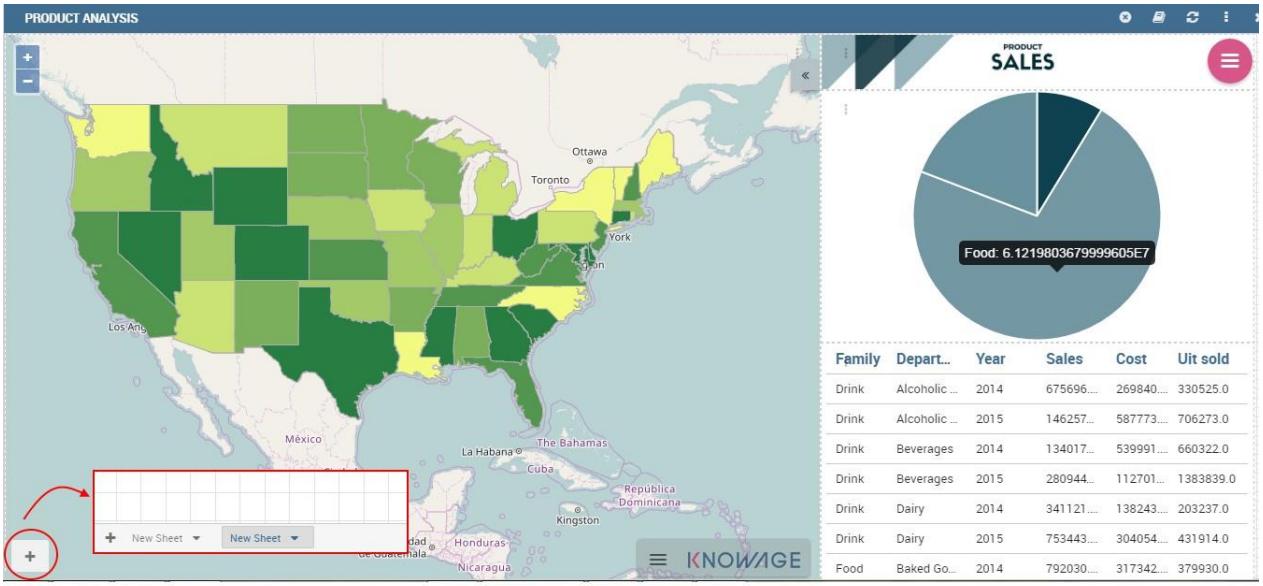
The **Clear cache** button lets you realign the data shown in your widget to the ones in your database. When you create your widget and associate your data fields, a photo of data is made and stored in temporary tables. This means that your cockpit will display the same data at each execution until you clean the chance by clicking on the dedicated button and execute the document again. Now your data are refreshed and updated to the one contained in your database at last execution time. As discussed before this button is available also in “Read only” modality.

Save

You can save the cockpit by clicking on the save button in the right-top corner. The document will be saved in the personal folder (technical users) or in the **My Analysis** section. We remember that it is possible to share the new cockpit with other users clicking on the dedicated icon. You can also choose in which folder, among the ones visible to your role, to place your shared document.

Multi-sheet functionality

Cockpit allows to manage data visualization splitting it in two or more sheets. In each layer the user can find and employ the features shown above. Indeed, it is possible to perform a new analysis (as highlighted in Figure below) selecting different datasets and gadgets.



Multi-sheet cockpit example.

A user can take advantage of the “move widget” functionality we saw in My first Cockpit to bring widget from one sheet to another.

Furthermore, it is possible, but not mandatory, to set associations between datasets underlying different sheets. The multi-sheet functionality is particularly useful to focus the analysis in a single spot and have a general overview over it in few clicks at the same time.

Chart

Charts are the most adopted method in presenting BI data since they allow an immediate perception of a phenomenon and are easily understandable. Focused on a visual impression more than a punctual lecture of values, they are specially suited to show trends and comparisons.

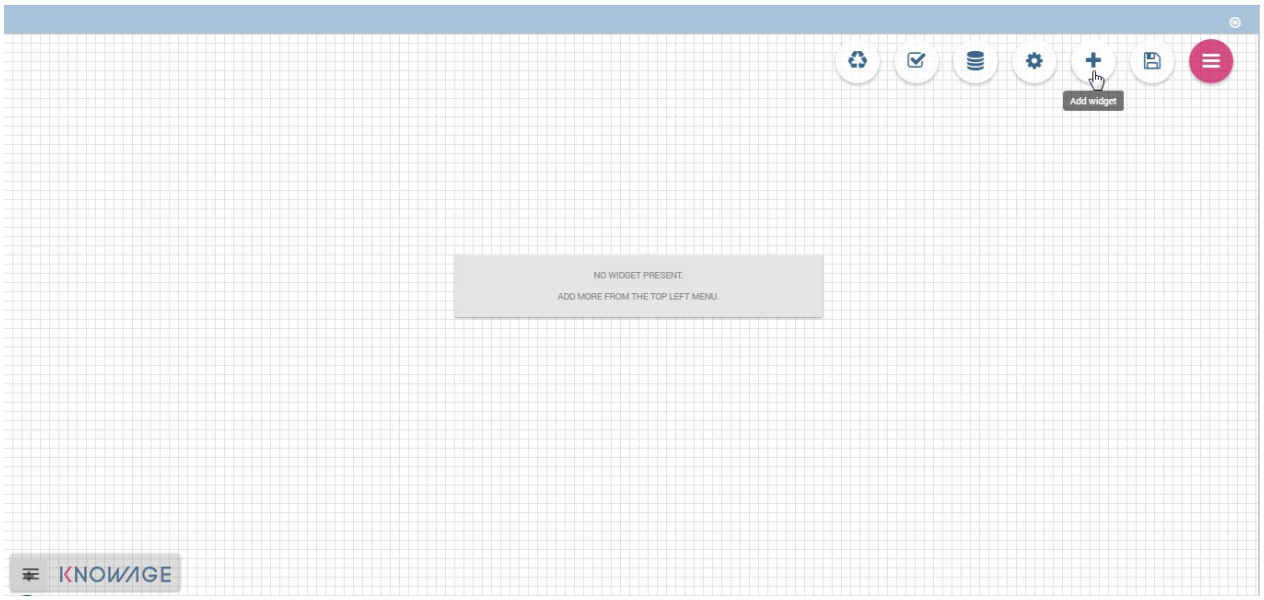
For these reasons, charts gain a pervasive level of usage and can be used by anyone to perform both synthetic and detailed analysis. Knowage provides a chart engine to create several types of charts, including:

- Bar
- Line
- Pie
- Sunburst
- Wordcloud
- Treemap
- Parallel
- Radar
- Scatter
- Heatmap
- Chord
- Gauge

My first Chart

Once you enter the Knowage environment as a final user, enter the **Analysis** area under the **Workspace** menu item, click on the **Create Analysis** icon and choose **Cockpit**.

Once opened, the cockpit interface is an empty page with a toolbar containing different options, the second of which is the **Add chart** feature.



Add a chart to a cockpit.

Cockpit

The Cockpit Engine allows the user to self-build interactive cockpits through an intuitive and dynamic interface. Read more in *Cockpit* chapter.

Clicking on the **Add Chart** icon, you will be asked to choose among some available widgets. Pick out the **Chart** one and let's now go into details on how to build a chart from scratch. The designer editor is divided into four principal tabs: **Dataset**, **Chart Engine Designer**, **Style**, **Cross** and **Filters**. As soon as the user clicks on the "Add Chart" button, he/she enters the "Dataset" tab editor. Here the user must select, using the "little plus" icon placed just aside the combo box line, one dataset. Then the user must switch to the "Chart Engine Designer" tab and choose a chart type among the available ones, as shown in figure below.

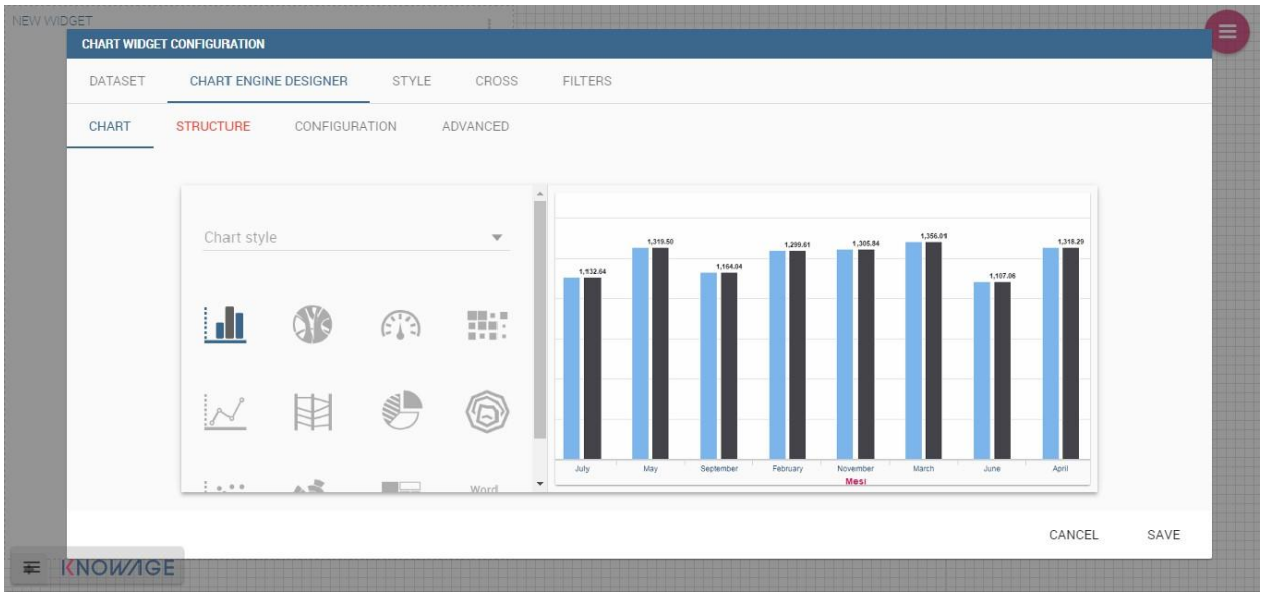


Chart editor.

After choosing the appropriate chart type you must go into the **Structure** page. Here it is possible to select the measures and the attributes chosen for the chart.

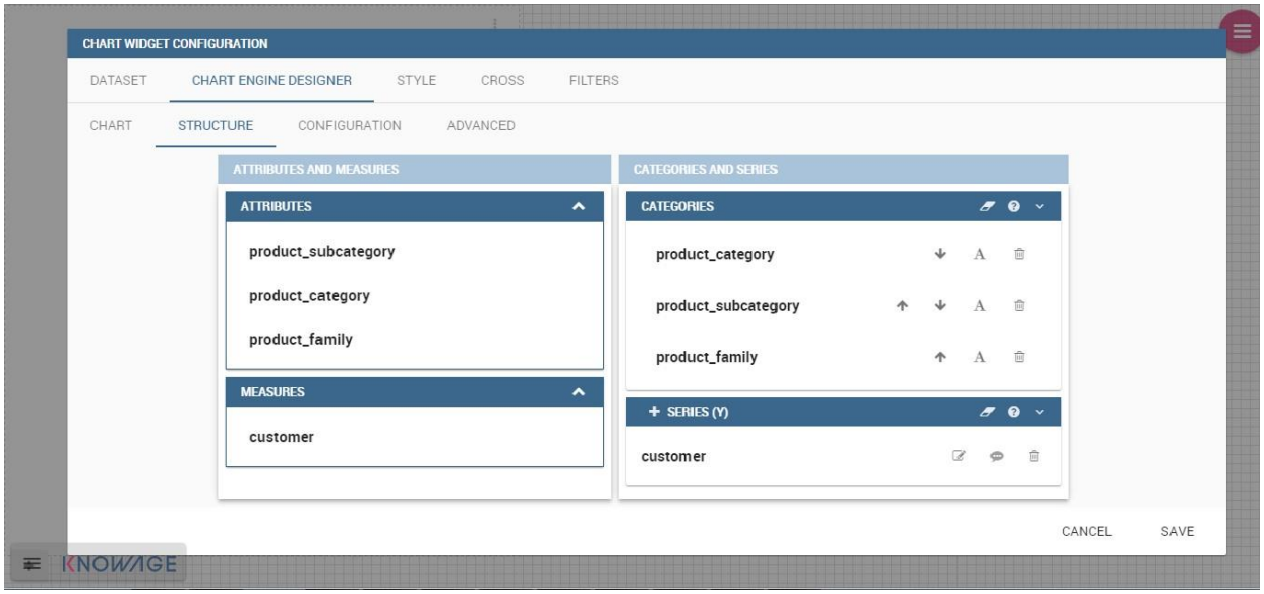


Chart structure.

Clicking on the **Configuration** page you will find eight different blocks as you can see in figure below.

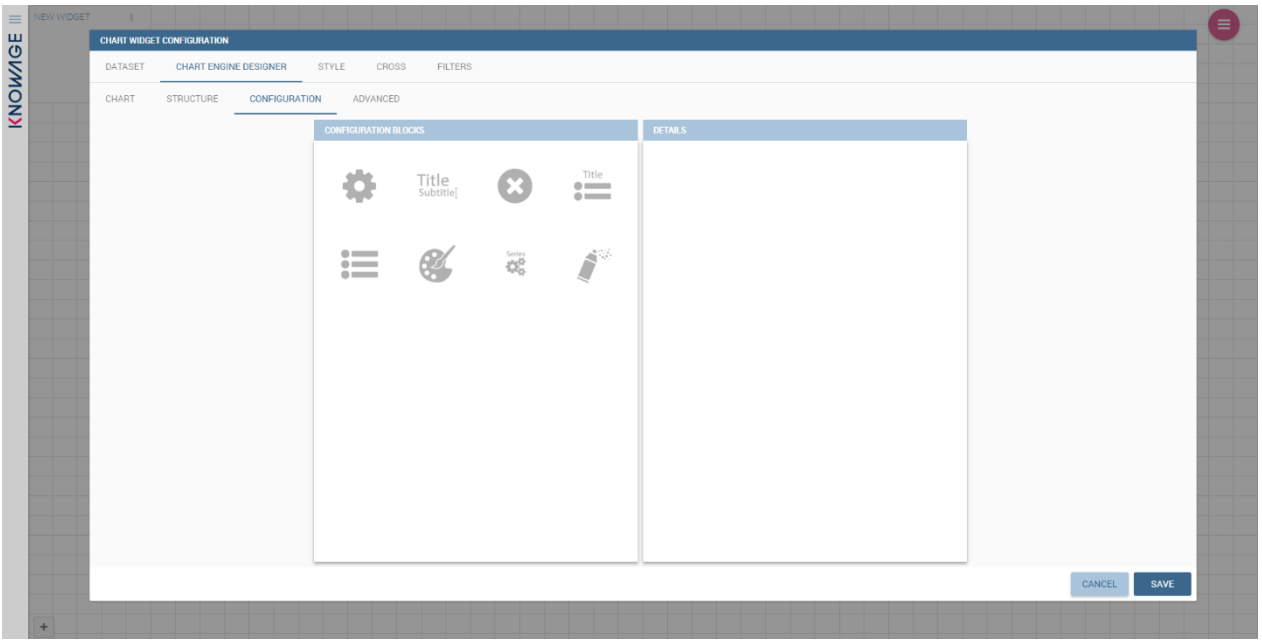


Chart configuration.

In detail these blocks concern:

- **Generic Details**, as the orientation of the chart, the family and the size font.
- **Title and Subtitle details**
- **No data message** where it is possible to put a message where the data are not founded.
- **Legend Title**
- **Legend Items**
- **Color Palette**
- **Advanced Series Configuration**
- **Custom Colors**

These eight blocks are common to all chart types; anyway, some chart types may have additional blocks.

The **Advanced** tab contains extra features, usually exploited by an expert user. Here the user can see all settable properties associated to the chart: it reflects the property tabs that an expert user should manually edit to generate a json template.

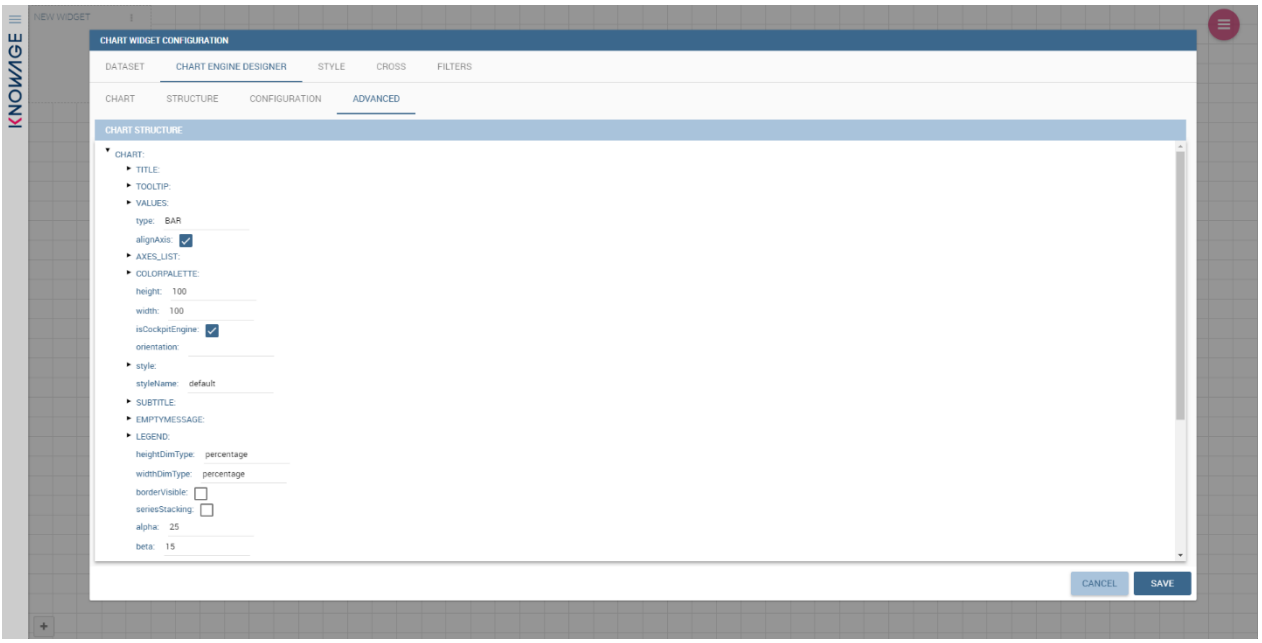


Chart Advanced Features.

In the next subsections, the available functionalities of the Structure, the Configuration and the Advanced tabs are described in a more specific way.

Structure

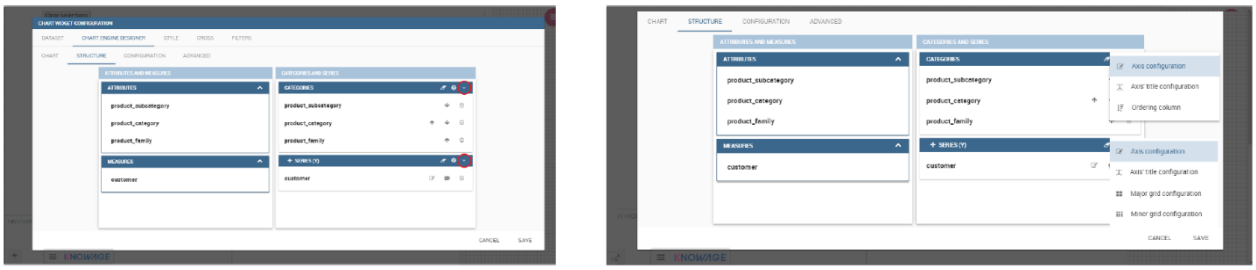
The “Structure” tab of the designer is the core of the Chart development. Here it is possible and mandatory to choose the measures and the attributes. When selected, the tab shows a two axes panel. The horizontal axis indicates the X-axis where you must choose one or more attributes. As well, the left axis is the Y-axis and here you must choose measures. You can also insert manually the axis title for both the X and the Y axis if the chart is configured to have axis titles.

Chart type changes may cause broke down

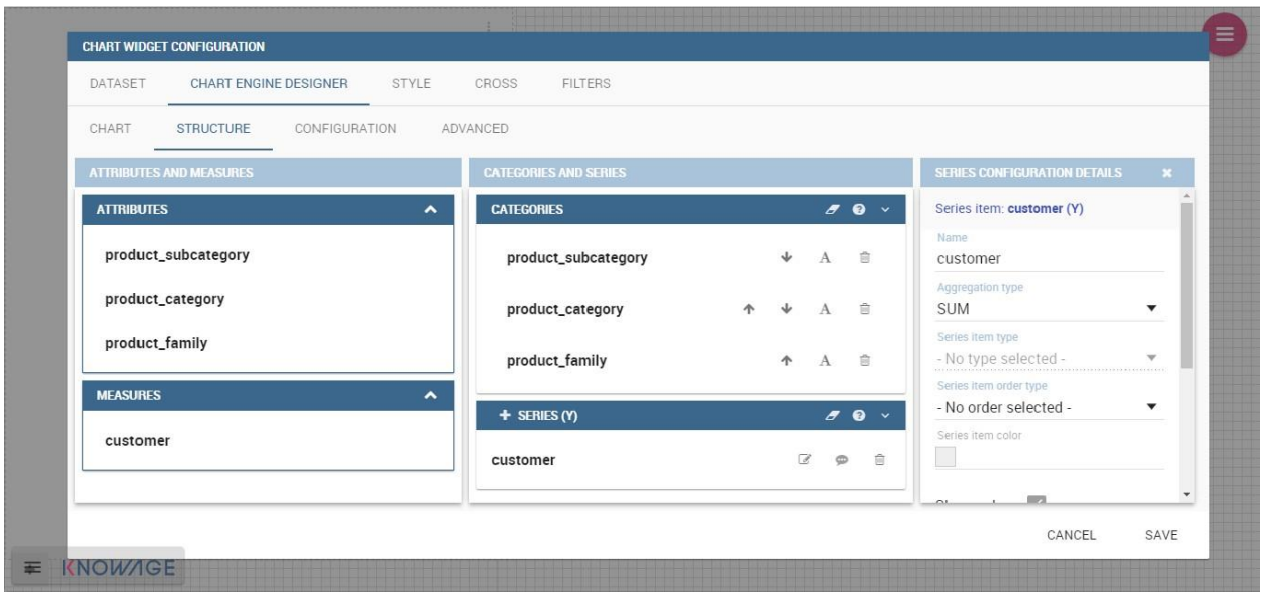
Before creating any chart, it is convenient to be sure of what kind of chart you want to develop. We stress that the user can change the chart type afterwards, but at the expense of a loss of just defined settings.

In this section it’s possible to customize the labels of the axis, title and grid style clicking on different buttons. With the arrow button, on the top of the Y-axis and X-axis, it’s possible to choose the axis configuration detail, the axis title configuration, the major and

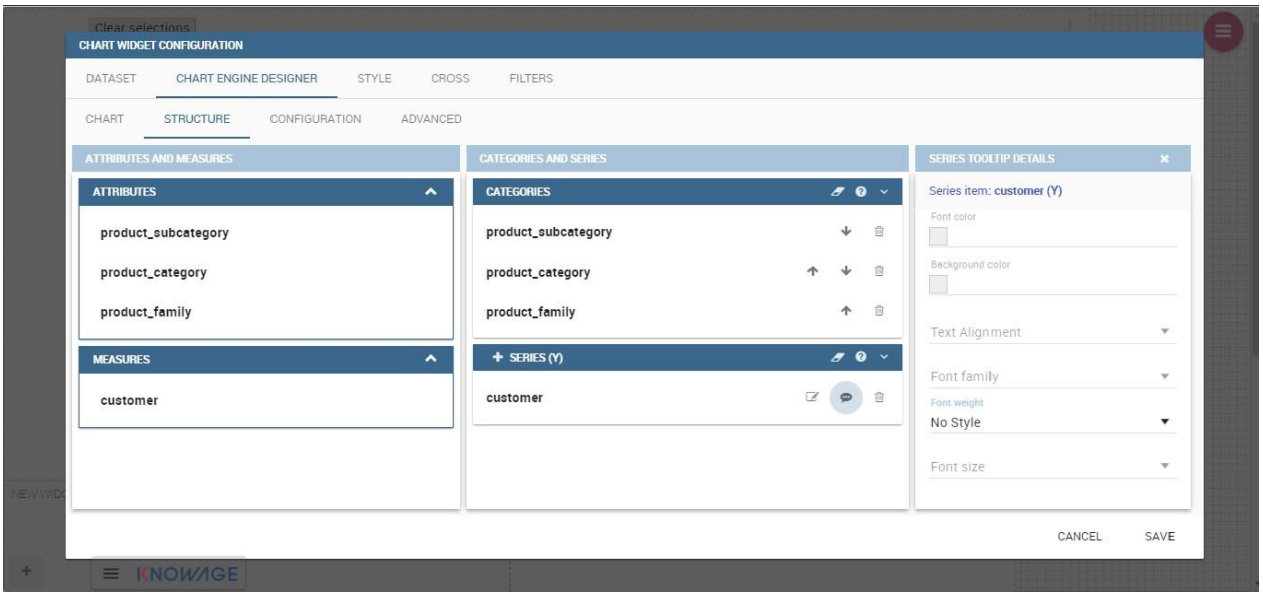
minor grid configuration (just for Y-axis) and ordering column (just for X-axis). With the pencil button opens a window on the right with the series configuration details where it's possible to choose the aggregation way, the order type of the series, if the data will be shown e so on. Finally, with the strip cartoon button you can choose the features of the tooltip (font color, text alignment, etc.). If the chart in place does not allow the customization of the axes the specific button will be disabled or not visible. The Figure below will show in detail the three buttons above explained:



From left to right: (a) Generic configuration axis (the specific arrow). (b) Generic configuration axis.



Series style configuration.

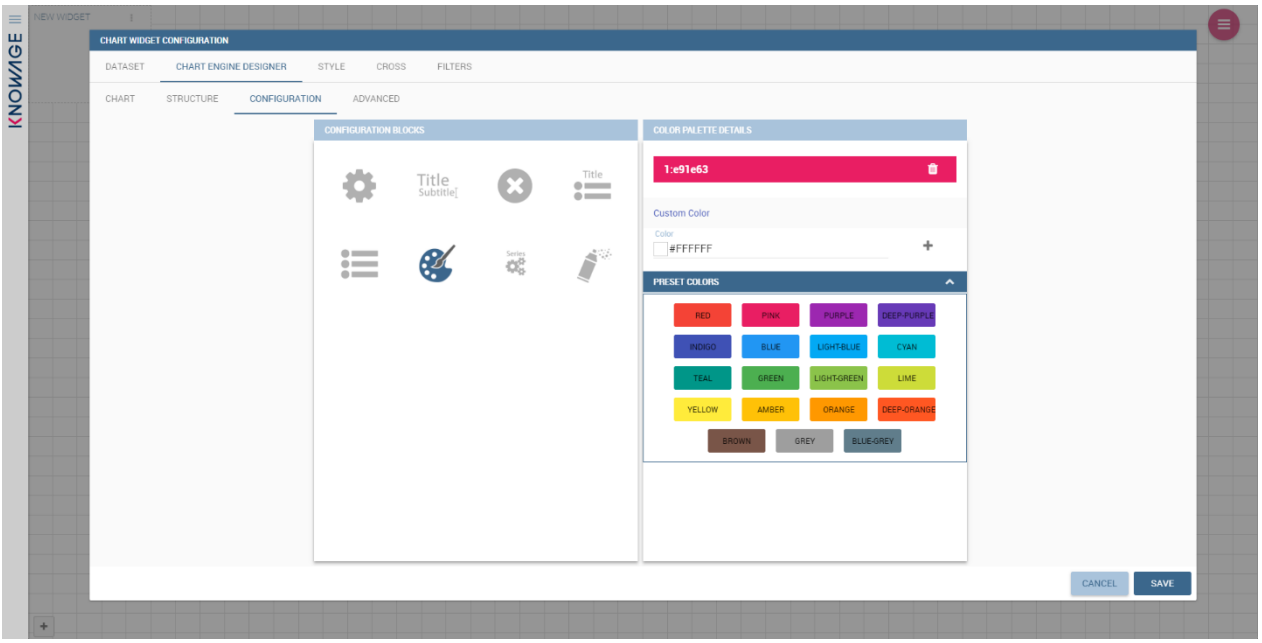


Series tooltip details.

Configuration

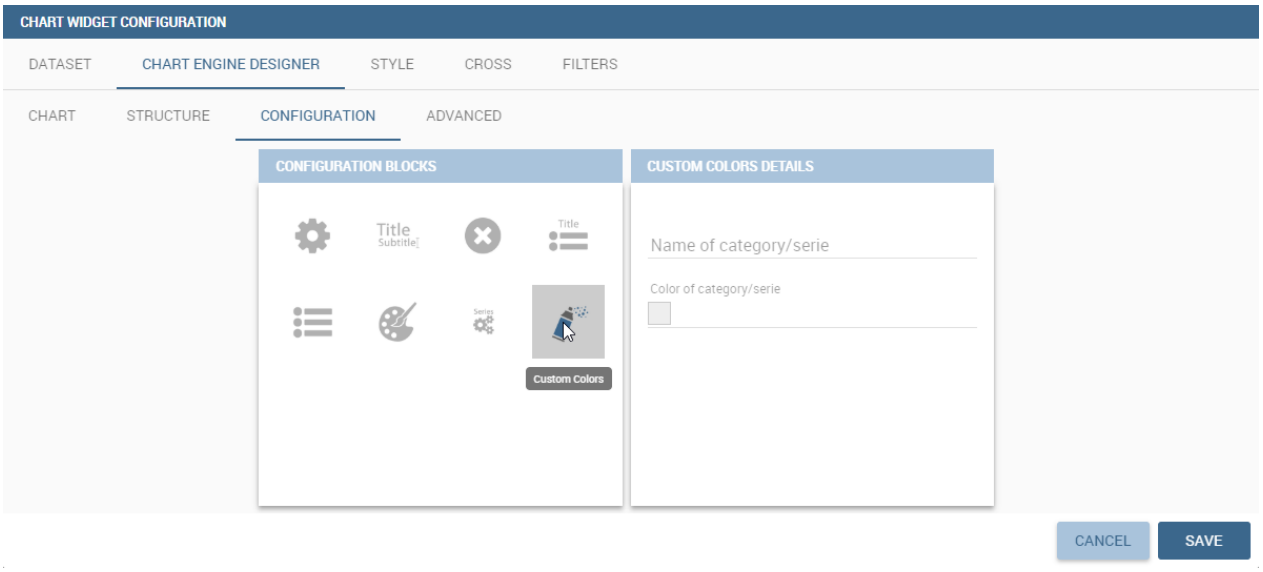
The **Configuration** section contains options to define the generic style of the chart. Here you can set the dimensions of the chart, the background color, insert the title and subtitle and define their style, choose the series palette, associate a specific color to a serie or category, add and configure the legend. The listed options are an example of what you can configure in the tab.

Note that for the color palette details you can use one already in the list or you can choose any color inserting the hex color code with the hashtag symbol. This is a very useful feature to customize the output.



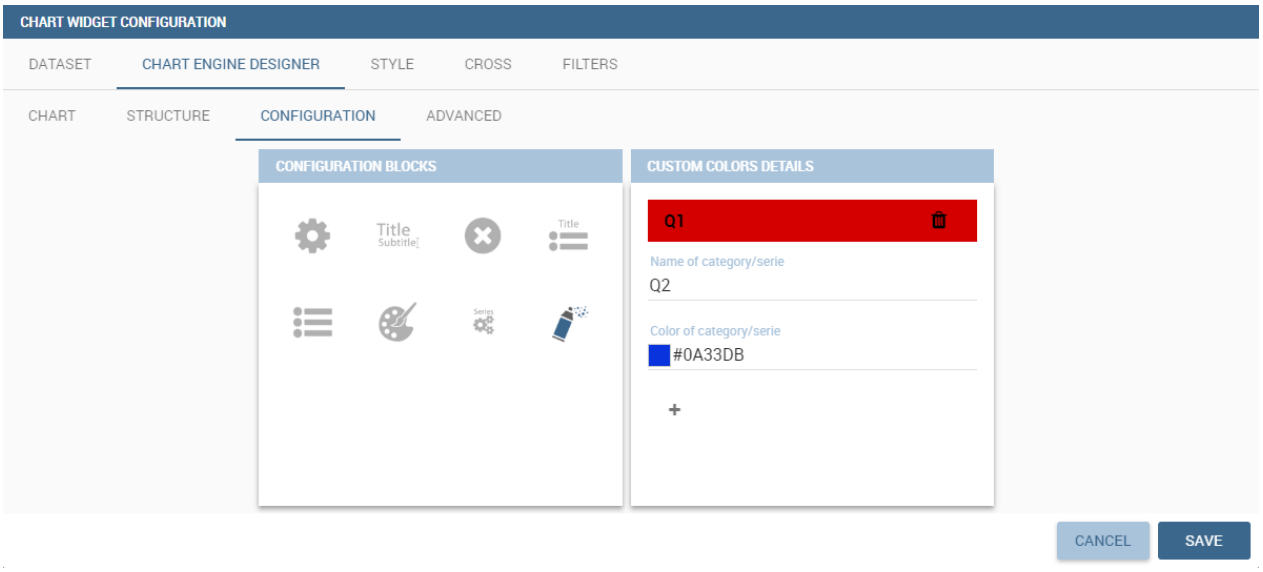
Color box editing.

In the 6.3 version, it has been introduced a new configuration option: The Custom Color.



Custom Colors details.

With this new option it is possible to assign a specific color to a category and/or serie or to a value of a category and/or serie. Look at the following figure for an example.



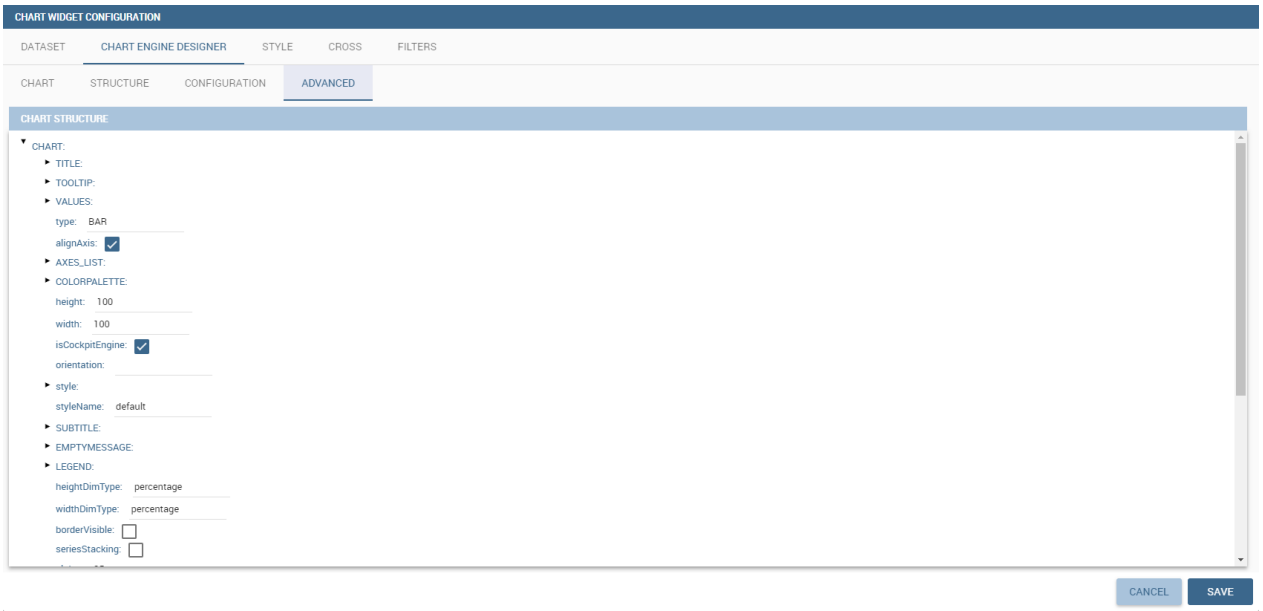
Custom Colors example.

To add a custom color simply write the category/serie value or name, select a color with the color piker and then click on the plus button. In the figure example it is assigned a color for each value of the ‘QUARTER’ category.

Indeed, the options available in this tab change according to the chart selected enabling different configurations. See Chart types in detail for a detailed description of the specific options of each chart.

Advanced options

The **Advanced** tab contains some advanced options to more customize the chart. Here it is possible, for example, to set the tooltip options, the widget dimensions, if the chart is stacking or not, the grouping type.



Advanced tab.

Down here are listed some of the most useful and new options.

The **dataLabels** option can be found under the path VALUES -> SERIE -> 0 or another serie -> dataLabels. The option is available only for measures. Here it is possible to set the labels style such as the color, font family or font weight.

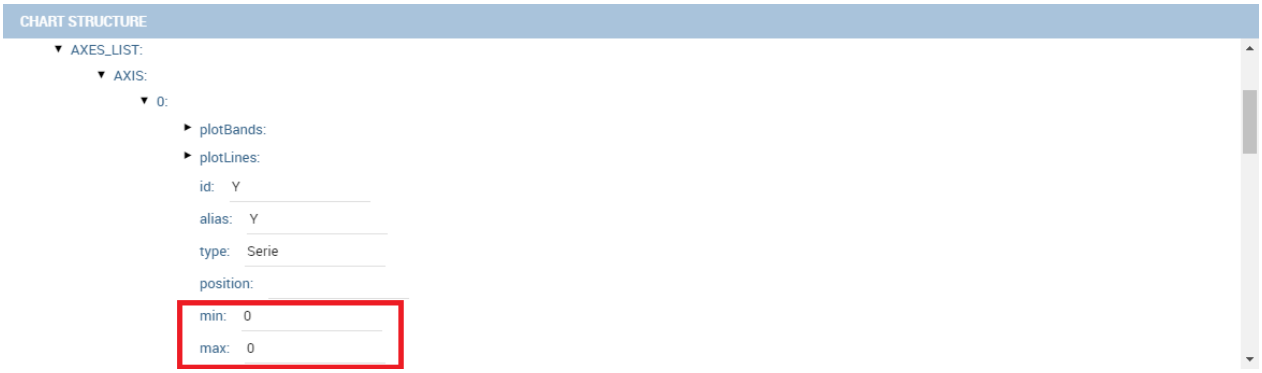
dataLabels option.

The **TOOLTIP** option allows to set the width and the radius of the tooltip's border.

The **plotBands** and **plotLines** options can be found under the path AXES_LIST -> AXIS -> 0 or another serie. With these options is possible to plot respectively bands and lines on the chart with fixed values and to set their style, like the line width and the line type or the band color.

plotBands option.

The **min** and **max** options are under the path AXES_LIST -> AXIS -> 0 or another serie. They are available only for series and allow to set the maximum and minimum axis value for the selected series' axis.



min and max options.

Chart types in detail

This section describes the different types of chart and how to create them within the **Chart Engine** of Knowage.

Traditional charts

Knowage allows you to create the so-called traditional charts like bar, line, pie, radar and scatter chart in a fancy way.

Each chart type is built on a specific dataset. Despite all, there are some general rules that can be applied to those "simpler" and common charts. The minimum requirement is to define/have a dataset with at least one attribute column and one measure column. Then you can select the type of chart you want to use from the **Chart** section;

meanwhile using the **Structure** section you can fill in the category box with one or more attributes (typically these will be place in the X-axis) and in the series box with one or more measures (typically placed as Y-axis' values). Refer to *Chart Structure* figure as example.

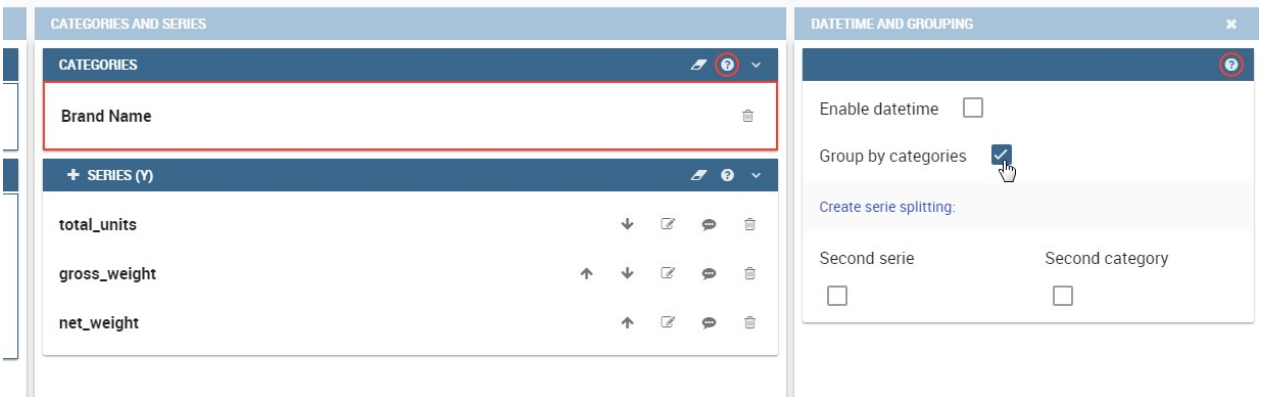
Once you have selected the attributes and measures you can edit the series style and axis style configurations as explained in My first Chart. Then go to **Configuration** to set the chart dimension, the title, the legend and to choose how to associate colors to series.

Some charts are endowed with datetime and grouping functions. It is possible to enable the grouping/splitting functions to **Bar** and **Line** charts.

The user can reach those functions just clicking on the "little arrow" located at the right end of category bar.

Datetime and grouping function.

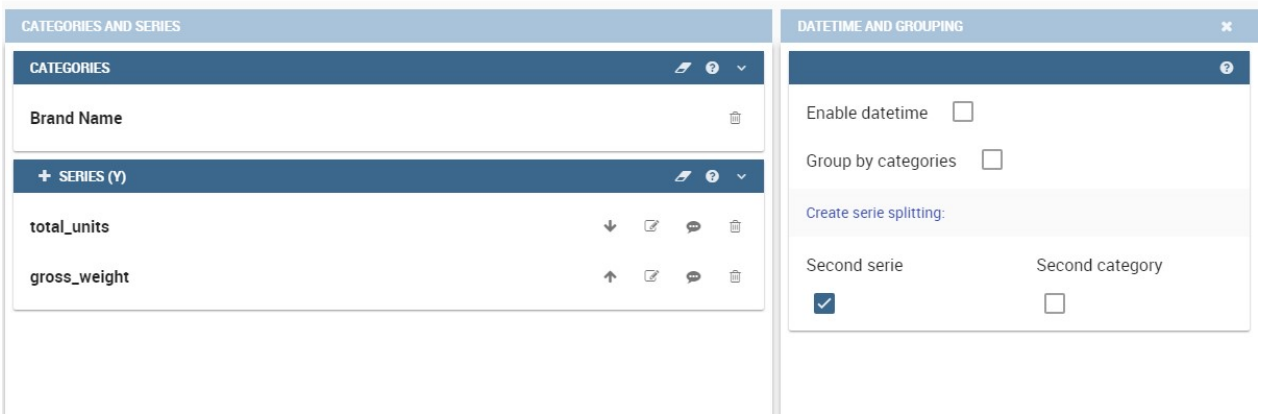
The grouping functions can be implemented only through specific categories and series configurations. As shown in figure below, the grouping function cannot be applied with just one attribute as category. To allow the function to be applied, the user must define two attributes as category fields.



Error alarm when enabling the grouping function.

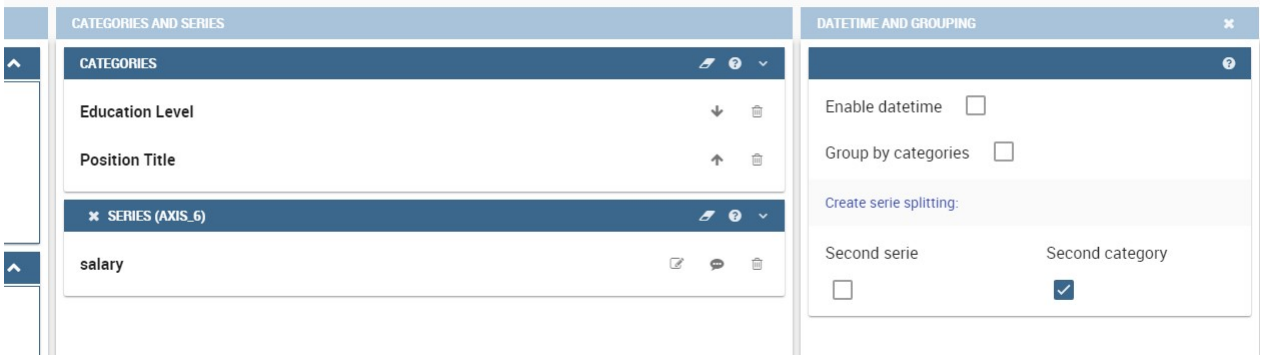
As well, the user can use the splitting functions to divide one series over the second one or over the second category.

To split the first series over the second one, remember that it is necessary to choose only one attribute as category field and two measures as series values. The following figure shows an example.



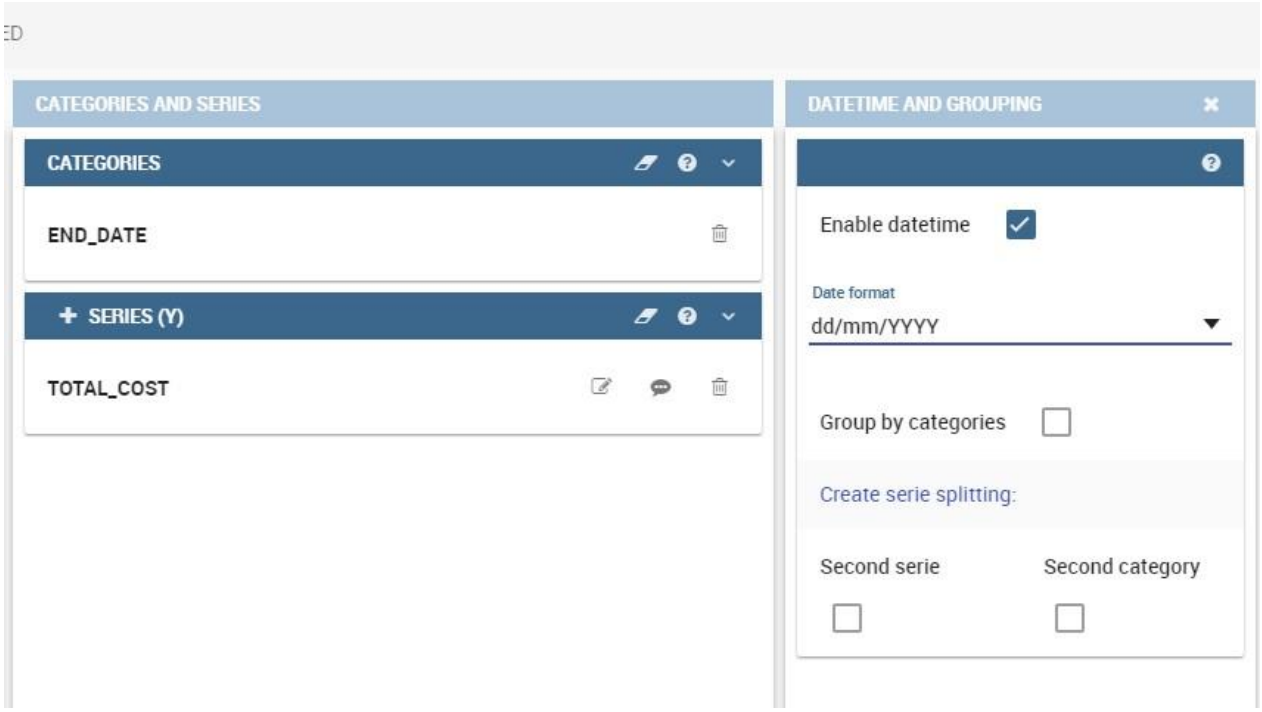
Split over second series.

Meanwhile to split a measure over second category it is mandatory to choose exactly two attributes as category field and only one measure as series value, as shown in figure below.



Split over second category.

Furthermore, in the occurrence the chart uses one datetime attribute as category field, the user can improve visualization applying the datetime function to custom date format.



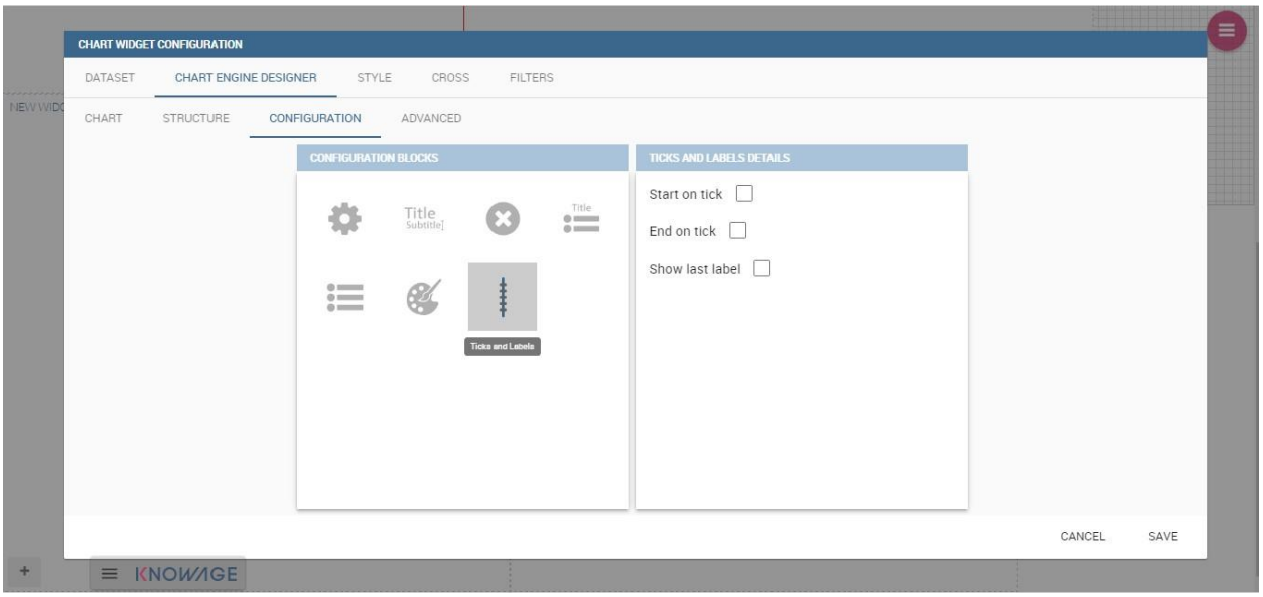
Datetime function usage.

For bar and line chart you can add more then one container for adding series in **Structure** section. In that case you will have in your chart more then one axis for series. In **Advanced** section you can specify to align these axes to 0 (zero) value. It is check box **align Axis** where checked means that axes will be aligned to 0, and unchecked means that they will not be aligned.

For pie chart inside **Advanced** section you can set configuration for your tooltip : to show/hide absolute value and/or percentage. Inside **tooltip** property of serie object you can find properties **showAbsValueTooltip** and **showPercentageTooltip**.

Scatter chart

A scatter chart is a graphical representation of scattering phenomenon of data. It is useful when the user wants to highlight the density of data upon certain spots to the detriment of readability of single points. If you select a scatter chart in the **Configuration** section, you will have Ticks and Labels Details instead of Advanced Series Configuration. Be careful to fill in the **Scatter configuration** with the **Zoom type**, as showed below.



Scatter Chart, ticks and labels details.

You must check if you want that the values in the Y-axis start (or end) in the first (last) tick or in the first (last) value of the dataset and if you want that the last label of the category axis should be showed.

Sunburst chart

The sunburst chart is a graph with a radial layout which depicts the hierarchical structure of data displaying a set of concentric rings. The circle in the center represents the root nodes, with the hierarchy moving outward from the center. The slices in the external rings are children of the slice in the inner circle which means they lie within the angular

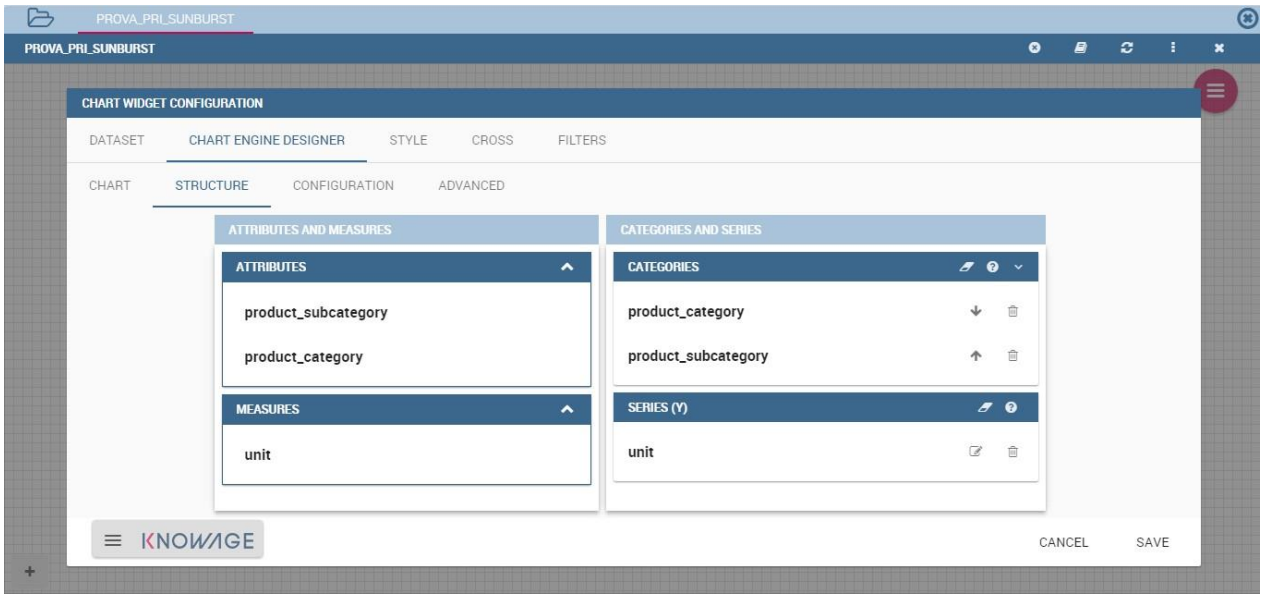
sweep of the inner circle. The area of each slice corresponds to the value of the node. Even if sunburst charts are not efficient space-wise, they enable the user to represent hierarchies in a more immediate and fascinating way.

To create a sunburst chart in Knowage you just must select a dataset with at least two attribute columns describing the hierarchy and at least a measure column that indicates the width of the slices. An example of dataset for the sunburst chart is showed in Table below.

Example of dataset for the sunburst chart.

CATEGORY	SUBCATEGORY	UNIT
Baking Goods	Cooking Oil	349
Baking Goods	Sauces	109
Baking Goods	Spices	290
Baking Goods	Sugar	205
Bathroom Products	Conditioner	64
Bathroom Products	Mouthwash	159
Bathroom Products	Shampoo	254
Bathroom Products	Toilet Brushes	92
Bathroom Products	Toothbrushes	94

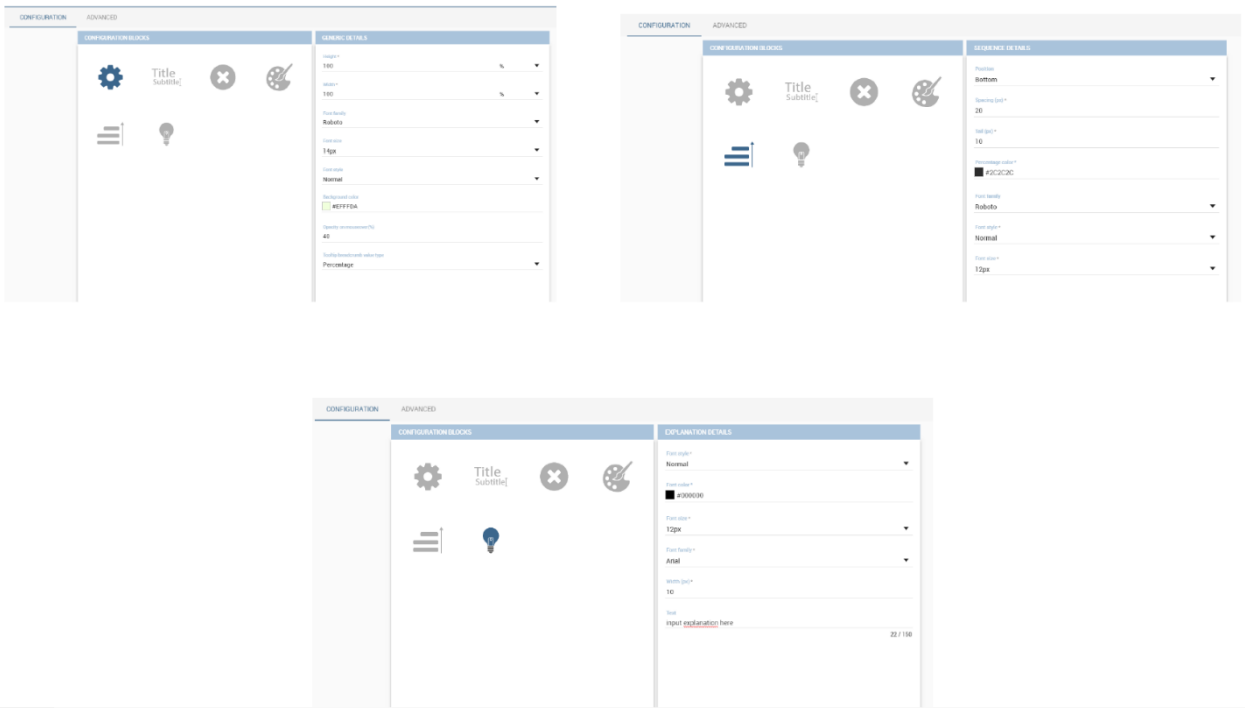
Once you selected the dataset and the type of chart, choose at least two attributes in the X-axis panel and a measure in the Y-axis panel as showed in the following figure.



Sunburst configuration.

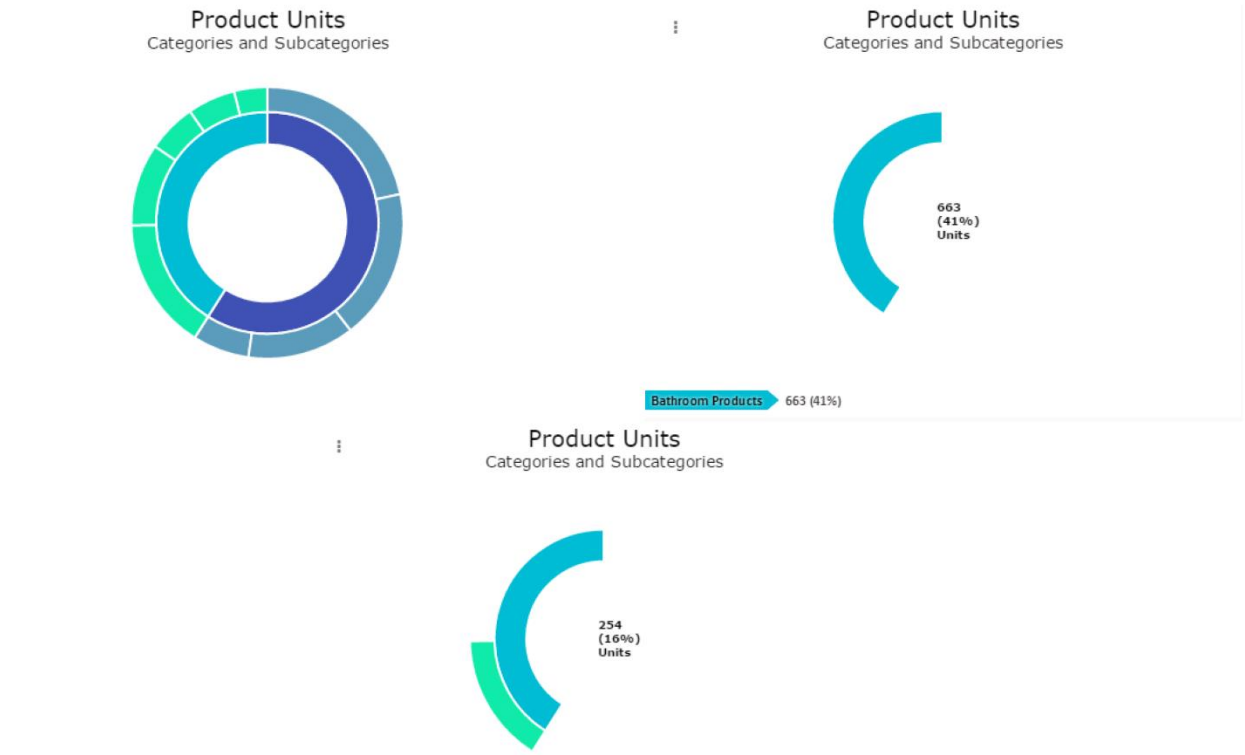
Then click on **Configuration**. As you can see the features are not the same as traditional chart. We give some tips on most important sunburst settings.

Using the **Generic** button, you can set the opacity on mouse movement and choose how to display the measure values: absolute, percentage or both. These two features allow the visualization of data just moving the mouse over the slice: the slice is highlighted, and values are shown in the center of the ring while the root-node path for the node selected is displayed on the left bottom corner of the page. To custom the root-node path, click on the **Sequence** icon and choose position, label tail size and text style. The tooltip is a mandatory field since it shows the value of the selected slice. Therefore, be sure to have filled it before saving by using the **Explanation detail** panel. Figure below sums up the three features.



Generic, Sequence and Explanation configuration

In Figure below you find the sunburst obtained with data of Table 20.



From left to right: (a) Sunburst. (b) Sunburst category.(c) Sunburst subcategory.

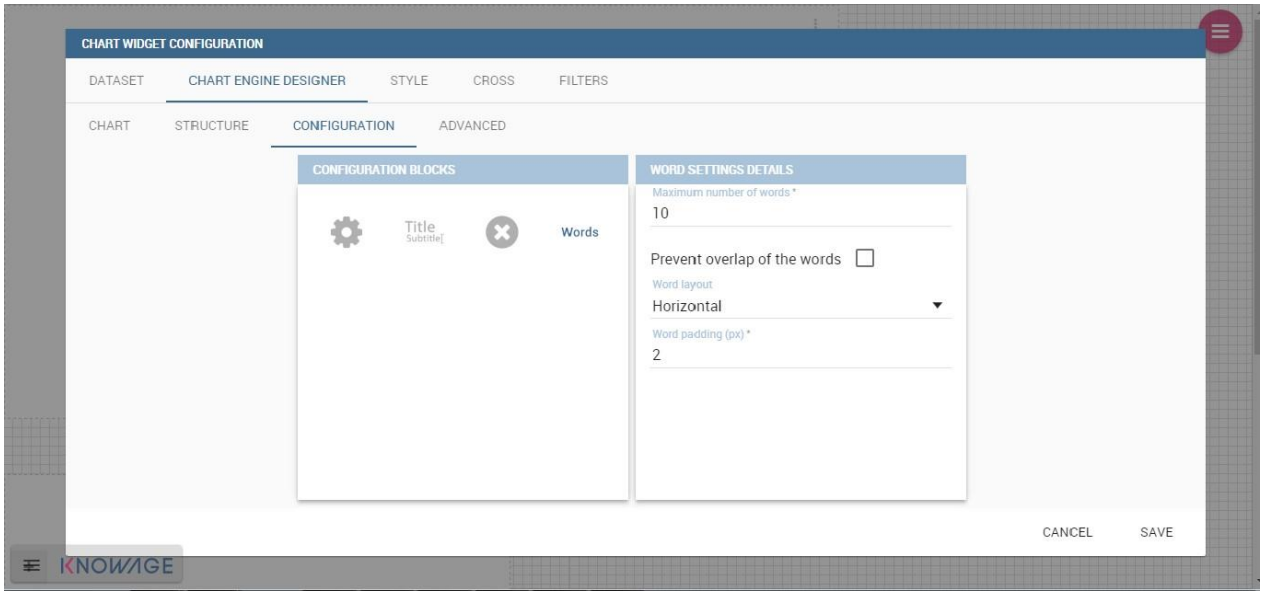
Inside **Advanced** section you can set value for scale that will increase/decrease your chart. You need to set numeric value for property **scale**.

Wordcloud chart

The Wordcloud chart is a graphic to visualize text data. The dimension of the words and colors depend on a specified weight or on the frequency of each word.

The dataset to create a Wordcloud should have at least a column with attributes and only one column with numerical data which represents the weight to assign to each attribute. Choose one attribute as category field (the Wordcloud accept only one attribute in the category box) and a measure as series field.

Switch to the **Configuration** section to set the generic configuration of the chart and to custom fields of the **Word settings details**. Here the use can decide if to resize the words accordingly to the measure retrieved in the dataset (**Series** option) or accordingly to the frequency of the attributes in the dataset (**Occurrences** option). Moreover, it is possible to set the maximum number of words that you want to display, the padding between the words, the word layout and whether you want to prevent overlap of the words as showed in Figure below.



Wordcloud chart specific configuration.

Treemap chart

The Treemap is a graphical representation of hierarchical data, which are displayed as nestled rectangles. Each branch of the tree is given by a rectangle, which is tiled with smaller rectangles representing sub-branches. The area of the rectangles is proportional to a measure specified by a numerical attribute. The Treemap is useful to display a large amount of hierarchical data in a small space.

To create a Treemap chart you must select a dataset as the one described for the sunburst chart in the Parallel chart.

Once you have selected the dataset, choose the Treemap chart type in the designer and then at least two attributes into the X-axis panel. The order of the attributes in the X-axis panel must reflects the order of the attributes in the hierarchy starting from the root to the top.

Finally, you can set generic configurations and colors palette in the **Configuration** tab and advanced configurations in **Advanced editor** tab.

In Figure below we show the Treemap resulting with data of our example

From left to right: (a) Treemap. (b) Treemap sub-branches.

Parallel chart

The parallel chart is a way to visualize high-dimensional geometry and various data. The axes of a multidimensional space are represented by parallel lines, usually equally spaced-out, and a point of the space is represented by a broken line with vertices on the

parallel axes. The position of the vertex on an axis correspond to the coordinate of the point in that axis.

To create a parallel chart, select a dataset with at least one attribute and two columns with numerical values. You can find an interesting example of dataset in the next table where we display some of its rows.

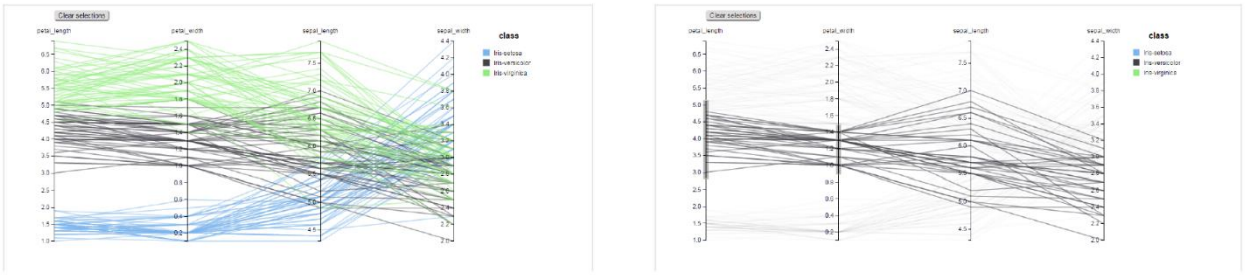
Example of dataset for the parallel chart.

ID	sepal_length	sepal_width	petal_length	petal_width	class
36	5.0	3.2	1.2	0.2	Iris-setosa
37	5.5	3.5	1.3	0.2	Iris-setosa
38	4.9	3.1	1.5	0.1	Iris-setosa
39	4.4	3.0	1.3	0.2	Iris-setosa
40	5.1	3.4	1.5	0.2	Iris-setosa
41	5.0	3.5	1.3	0.3	Iris-setosa
42	4.5	2.3	1.3	0.3	Iris-setosa
43	4.4	3.2	1.3	0.2	Iris-setosa
44	5.0	3.5	1.6	0.6	Iris-setosa
45	5.1	3.8	1.9	0.4	Iris-setosa
66	6.7	3.1	4.4	1.4	Iris-versicolor
67	5.6	3.0	4.5	1.5	Iris-versicolor
68	5.8	2.7	4.1	1.0	Iris-versicolor
69	6.2	2.2	4.5	1.5	Iris-versicolor
70	5.6	2.5	3.9	1.1	Iris-versicolor
71	5.9	3.2	4.8	1.8	Iris-versicolor
101	6.3	3.3	6.0	2.5	Iris-virginica

Example of dataset for the parallel chart.

ID	sepal_length	sepal_width	petal_length	petal_width	class
102	5.8	2.7	5.1	1.9	Iris-virginica
103	7.1	3.0	5.9	2.1	Iris-virginica
104	6.3	2.9	5.6	1.8	Iris-virginica
105	6.5	3.0	5.8	2.2	Iris-virginica
106	7.6	3.0	6.6	2.1	Iris-virginica
107	4.9	2.5	4.5	1.7	Iris-virginica
108	7.3	2.9	6.3	1.8	Iris-virginica

In this example three different classes of iris are studied. Combining the values of some sepal and petal width or lenght, we can find out which class we are looking at. In Figure below (a part) you can find the parallel chart made with the suggested dataset. While in next figure (b part) it is easy to see, thanks to selection, that all iris with petal length between 2,5 and 5.2 cm and petal width 0,9 and 1,5 cm belong to the iris-versicolor class.



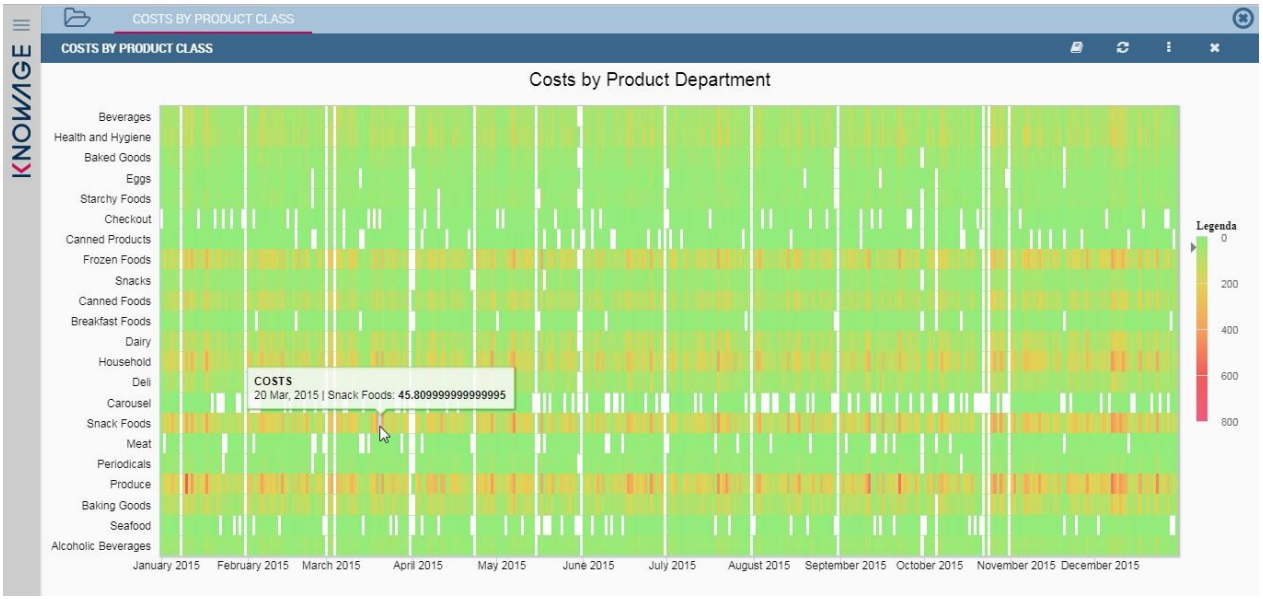
From left to right: (a) Parallel. (b) Parallel chart selection.

Therefore, select **parallel** as chart type using the designer interface, then choose one or more attributes in the X-axis panel and one or more measures in the Y-axis panel.

On the **Configuration** tab you can set the generic configuration for the chart and you must fill the **Series as filter column** filed under ”Limit configuration”.

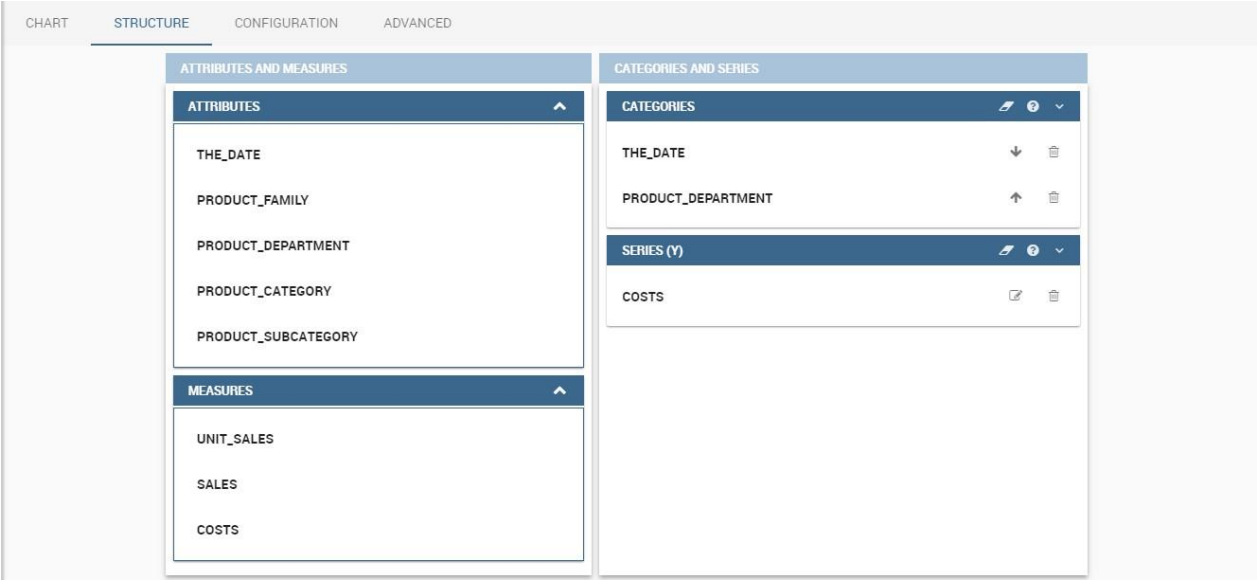
Heatmap chart

Heatmap chart uses a chromatic Cartesian coordinate system to represent a measure trend. Each point of the Cartesian system is identified by a couple of attributes. Note that one attribute must be a datetime one. Meanwhile, each couple corresponds to a measure that serves to highlight the spot with a certain color according to the chosen gradient. Figure below gives an example of how an heatmap chart looks like inside Knowage.



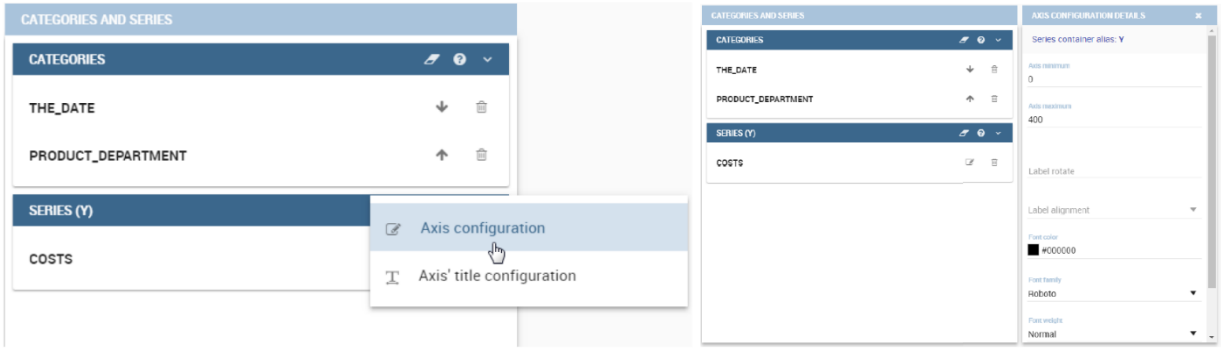
Heatmap example.

Before configuring a heatmap chart, be sure that your dataset returns at least two attributes, one of which **must** be a datetime one, and (at least) one measure. Once entered the chart designer, choose the “Heatmap” type and move to the “Structure” tab. Use the datetime attribute and another attribute as category fields and one measure as series fields. Figure below shows an example.



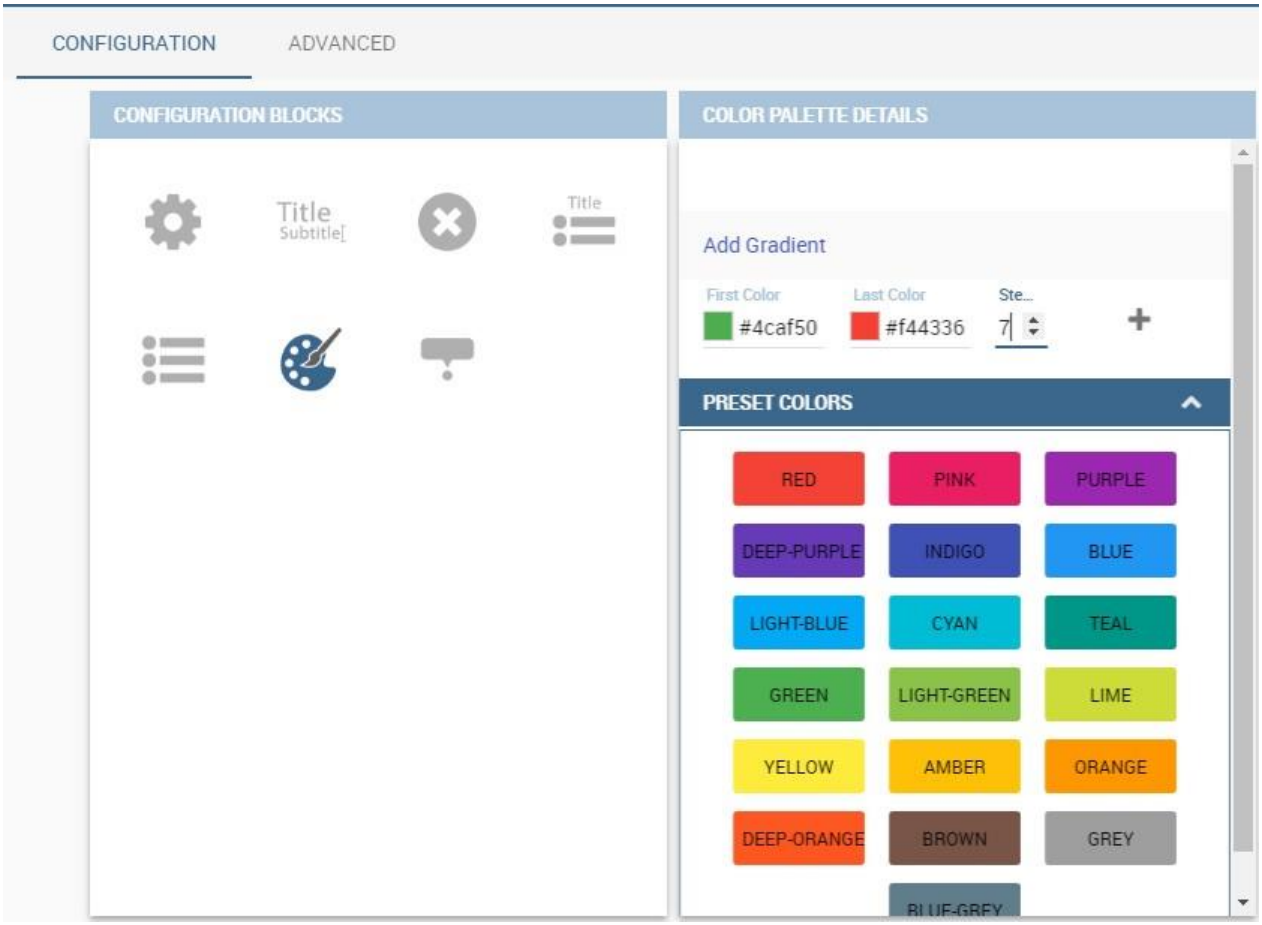
Configuring the attributes and the series for the heatmap chart.

Note that for series axis it is possible to specify the values' range by assigning a minimum and the maximum value, as shown in figure below. Otherwise, the engine will automatically link the axis scale to dataset results set.



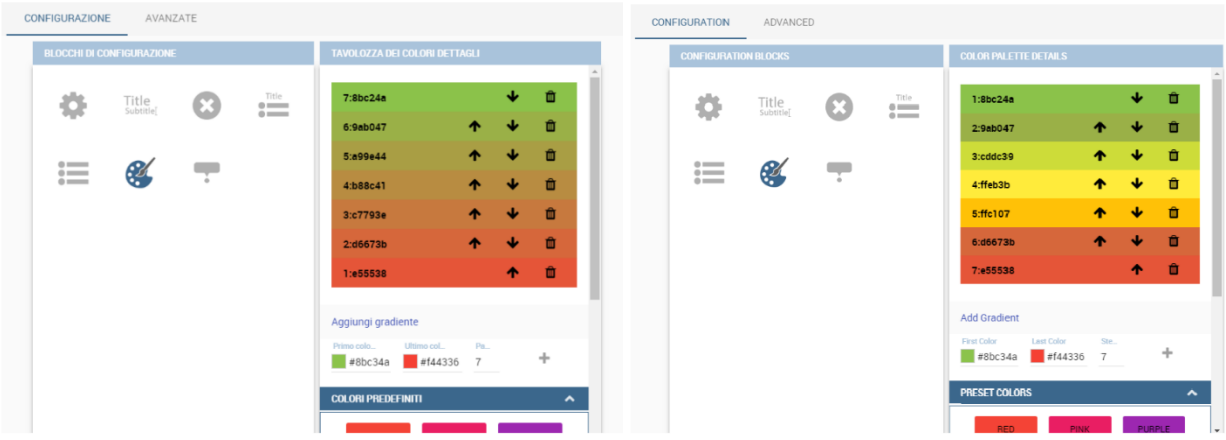
Configure min and max values for series.

The next step is to move to **Configuration** tab and select the **Color palette** icon. Here (figure below) the user must define the chromatic scale which will be associated to the measure values. The panel will demand the user to insert the first, the last color and the number of bands that will constitute the color scale.



Add gradient panel.

The engine will create a progressive color scale as shown in the left image of figure below. To custom the scale the user can use the Pre-set colors and use the arrow to move up and down Heatmap chart the added color or the user can increase the number of steps and then some intermediate color to leave more contrast between them.



Custom color scale.

Remember to edit both **Legend** and **Tooltip** configuration in the **Tooltip details** panel to improve the readability of the chart.

Chord chart

Chord diagram is a graph which allows to show relationship between entities and between data in a matrix. The entities can belong to a unique category while the arc be non-oriented or belong to two different categories. In this latter case, they have direct arcs. The data are arranged radially with arcs that represent the connection between points. The width of the arc connecting two points depends on the weight assigned to the edge connecting these two points. This graphic is useful when you want to represent a large number of data in a small space.

The chord diagram requires a dataset that have a column with numerical values. These represent the weight of the arc connecting two points. It also must have two columns with the entries for the entities to be connected in the diagram. These two columns must have the same set of values so that the engine can understand the relation between all the entities. If there is not a relation between two entities the weight of the arc is zero. Note that when you create a directed chord diagram with two different categories, all the relations between entities of the same category have a zero weight.

An example of dataset for the chord chart is represented in Table below:

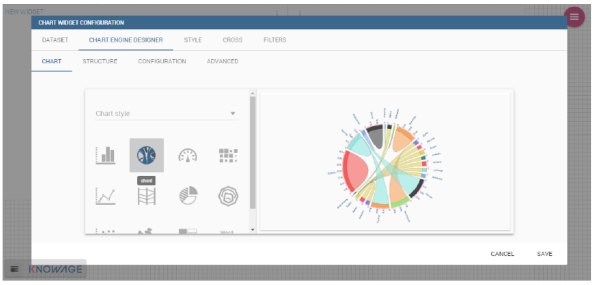
Example of dataset for the chord chart.

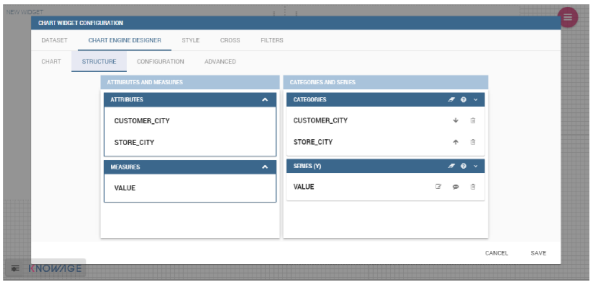
CUSTOMER_ CITY	STORE_ CITY	VALUE
Beaverton	Portland	4609.0000
Lake Oswego	Portland	4201.0000
Milwaukie	Portland	5736.0000
Oregon City	Portland	3052.0000
Portland	Portland	3984.0000

Example of dataset for the chord chart.

CUSTOMER_ CITY	STORE_ CITY	VALUE
W. Linn	Portland	3684.0000
Albany	Salem	5544.0000
Corvallis	Salem	8542.0000
Lebanon	Salem	8015.0000
Salem	Salem	6910.0000
Woodburn	Salem	6335.0000
Albany	Albany	0.0000
Beaverton	Beaverton	0.0000
Corvallis	Corvallis	0.0000
Lake Oswego	Lake Oswego	0.0000
Lebanon	Lebanon	0.0000
Milwaukie	Milwaukie	0.0000
Oregon City	Oregon City	0.0000
Portland	Portland	0.0000
Salem	Salem	0.0000
W. Linn	W. Linn	0.0000

Once you have selected the dataset open the designer and select chord chart type. Then choose the two entities in the X-axis panel and the value in the Y-axis panel as showed in figure below. Now you are ready to customize the chart setting the generic configuration and the palette on **Configuration**.





Chord configuration.

Gauge chart

Gauge chart uses needles to show information as a dial reading. It allows to visualize data in a way that resembles a real-life speedometer needle. The value of the needle is read on a colored data scale. Colors are used to provide additional performance context (typically green for good and red for bad). This chart type usually is used in dashboards to show key performance indicators or any measure having reference values.

For gauge chart you should have only series items, the one that gives you values for the chart. So, the defined dataset to be used should provide numerical data for the Y-axis for the gauge chart. After selecting the dataset go to the designer and select **gauge** in chart type combo box. Then choose one or more measure on the Y-axis panel on the **Structure**. Moreover, you must not forget to provide all data needed for the **Axis style configuration** of the Y-axis.

When you finished to set all the mandatory and optional parameters and configurations in the **Structure** tab you can select the **Configuration** tab and set the generic configuration of the chart.

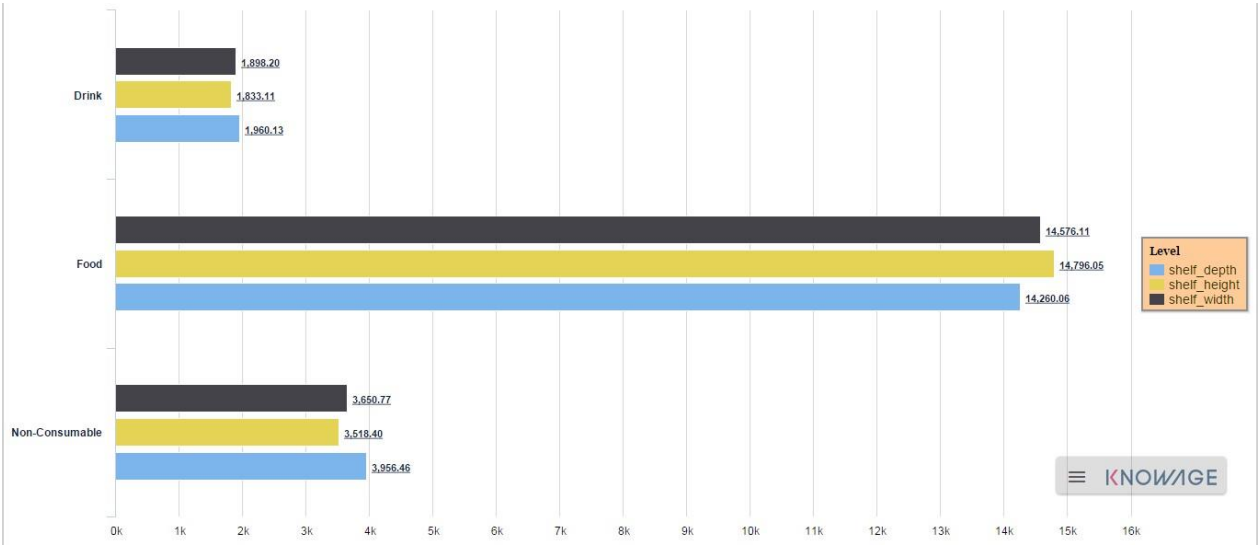
A short comment on chart drill down

Knowage **Chart Engine** allows you to drill down into categories. This means that the user can explore the details of each category as many times as configured. Indeed, to let the chart admits the drill down, it is necessary first that the chart in place allows it. Secondly the user must have dragged and dropped multiple attributes into the category axis in the **Configuration** tab. The order of the attributes in the X-axis panel determines the sequence in which the categories are going to be showed. When executing the chart, the label of the category is linkable, and it is possible to click on the label to drill down.

The chart that enables the drill down are:

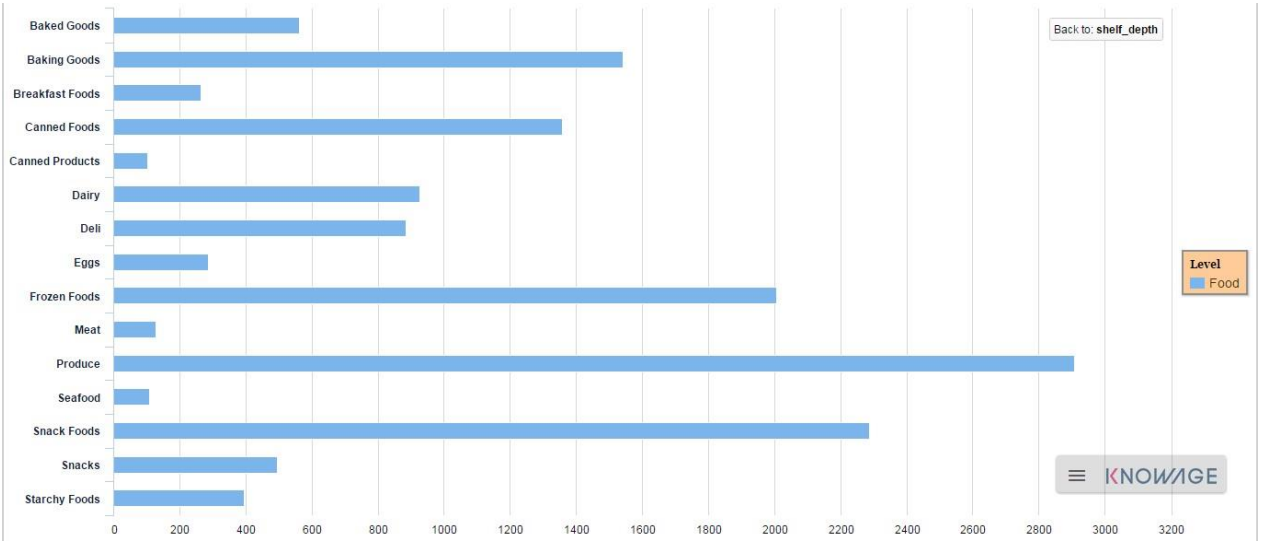
- Bar Chart
- Line Chart
- Pie Chart
- Treemap

To give an idea of the outcome, we take as instance the Bar Chart drill down. In the following example, the selected categories are four and called: `product_family`, `product_department`, `product_category` and `product_subcategory`. Once we open the document, we get as shown below:



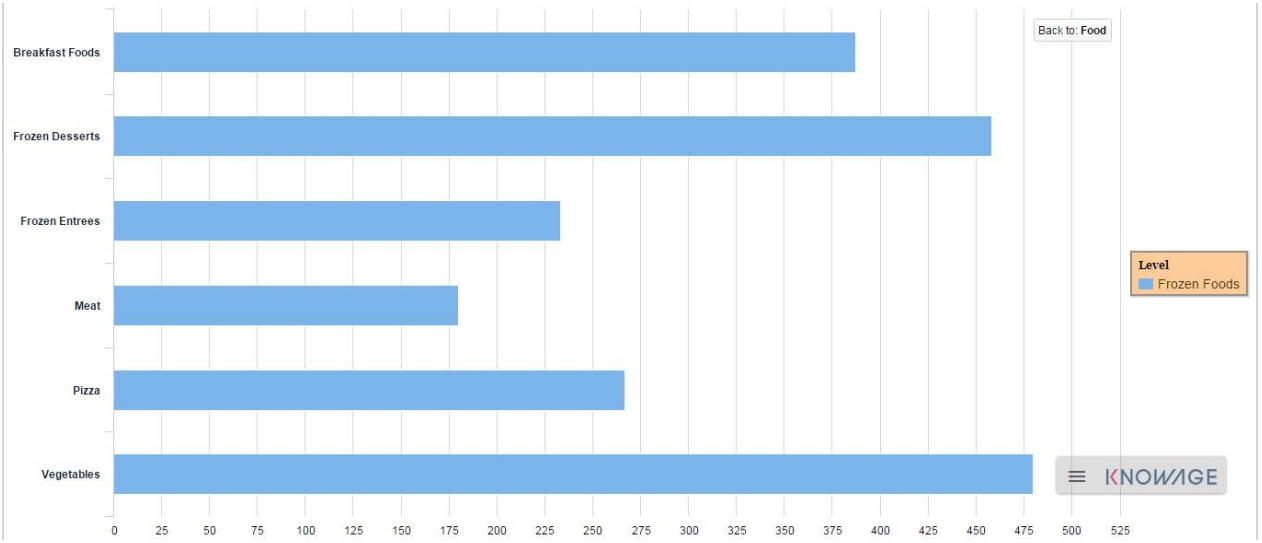
Drillable Bar Chart

When selecting `shelf_depth` measure of the Food category one gets (see next figure):



Drillable Bar Chart: first drill

Once again, we can select `Frozen food` subcategory and drill to a second sub-level as below:



Drillable Bar Chart: second drill

And so on to the fourth subcategory. Selecting the “Back to: …” icon available at the right corner of the graphic, the user can get back to the previous level. This efficient feature allows the user to have a deep insight of the analysis and draw important conclusions from it.