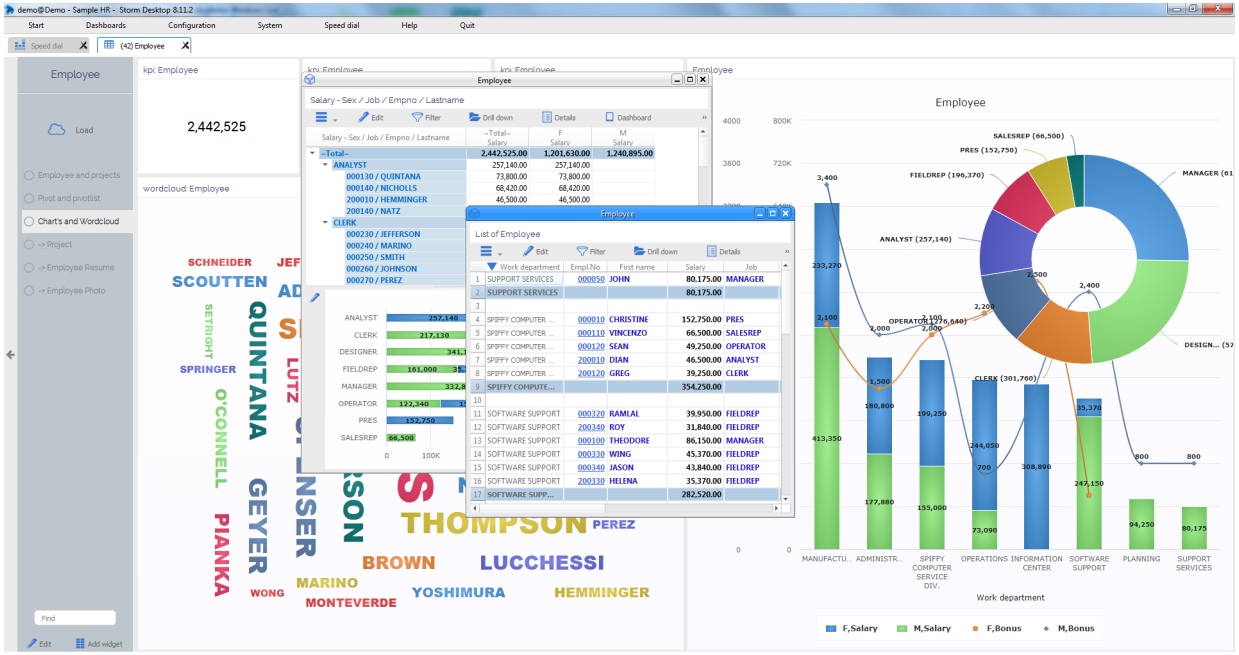
# User manual

v. 8.13.x



# Contents

**I User manual** **3**

**1 Using demo configuration** **4**
1.1 Login screen . . . 4
1.2 ”Demo - Sample HR” project selection . . . 5
1.3 Speed dial . . . 6
1.4 Start menu . . . 7
1.5 Dashboard overview . . . 8
1.6 Slave page . . . 10
1.7 Load . . . 13
1.8 Drill down . . . 14
1.9 List . . . 15
1.10 Pivot . . . 21
1.11 Pivotlist . . . 24
1.12 Chart . . . 25

**2 Editing the configuration** **26**
2.1 Common tasks . . . 26
2.1.1 Adding a page to a dashboard . . . 26
2.1.2 Adding a widget to a dashboard . . . 26
2.1.3 Editing a widget . . . 27
2.1.4 Adjusting the list . . . 27
2.1.5 Managing the tasks . . . 27
2.2 Editors’ reference . . . 27
2.3 User’s Editor reference . . . 30
2.3.1 Task . . . 30
2.3.2 Chart . . . 34
2.3.3 Chart types . . . 38
2.3.4 Geomap . . . 51
2.3.5 Heatmap . . . 54
2.3.6 Html App . . . 55
2.3.7 Image . . . 57
2.3.8 Kpi . . . 57
2.3.9 List . . . 58
2.3.10 Pivot . . . 60
2.3.11 Pivotlist . . . 64

2.3.12 Style . . . . . 65
2.3.13 Textapp . . . . . 67
2.3.14 Treemap . . . . . 68
2.3.15 Webpage . . . . . 70
2.3.16 Wordcloud . . . . . 70

# Part I

# User manual

# 1 Using demo configuration

## 1.1 Login screen

After launching Storm.exe, the login screen will appear. Here the user can select the project to launch and insert credentials.

Figure 1.1:
- "Projects" displays the list of projects to connect to it
- Click on desired project, and enter the user and password. Click on Sign In
- Further projects can be selected by clicking on *Open project* and browsing the folders
- Projects selected in this way will be visible in the list the next time Storm is launched
- If the user hover for a few seconds he can see the path of the project file, and other information such as owner and creation date
- To remove a project from the start menu, click "Remove selection" after right-cliking on it
- "Remove object" will delete the project file from the disk

## 1.2 ”Demo - Sample HR” project selection

The first time Storm is launched, five projects are available. Those are the five DEMO projects delivered in the package to illustrate Storm capabilities to adapt to different type of database:

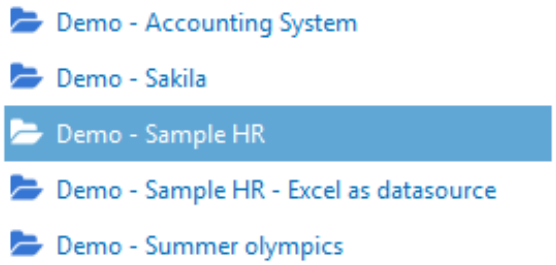



Figure 1.2:
• Accounting system: balance of a bank, will dashboard of clients, their transaction, and some general balance sheet
• Sakila: inventory and rental of a video shop chain
• Sample HR: the summary of a human resource department, with employees and managers, departments, projects, salary and bonus
• Sample HR - Excel as datasource: a smaller version of the previous config, but using an Excel file as datasource
• Summer olympics: olympic medals and events of all the modern olympiads until 2008

Select the project ”Demo - Sample HR” and log in with user ”demo”, leaving the password blank.

## 1.3 Speed dial

The Speed Dial is the first board shown after the start menu. lt displays a selection of dashboards, in order to allow quicker access to the most important data.

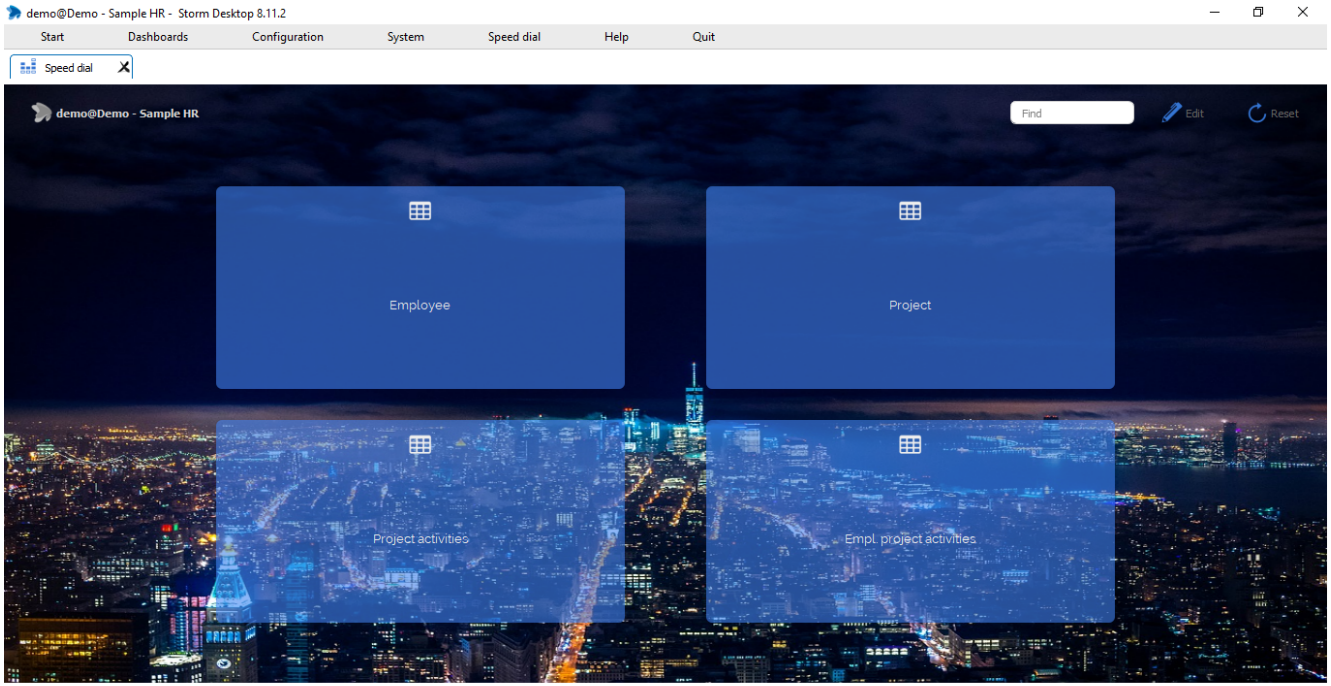



Figure 1.3:
• Find bar: filters out all dashboards not maching the input string
• Edit: change this selection by accessing Speed Dial editor
• Reset: reload the Speed Dial

## 1.4 Start menu

By clicking on the *Start* button in the upper left corner the start menu will open up, showing a list of dashboards.

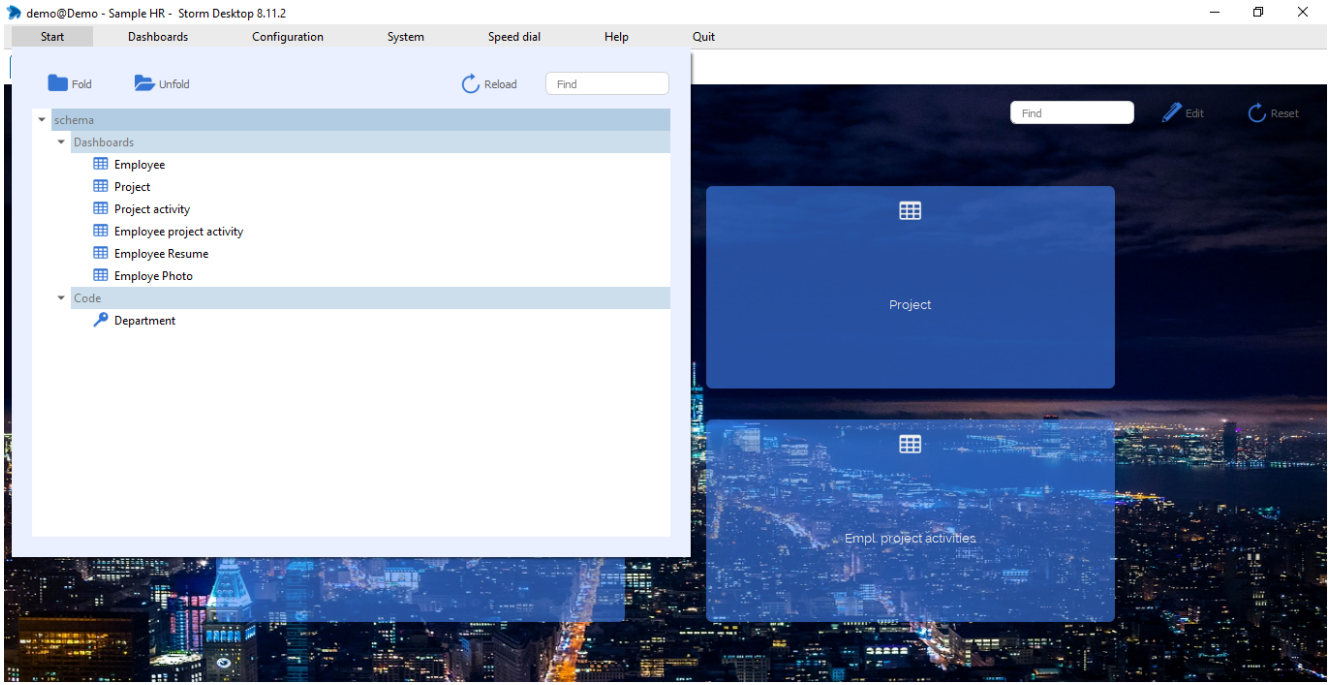



Figure 1.4:
- Fold: close all the nodes
- Unfold: open all the nodes
- Find bar: filters out all dashboards not maching the input string
- Reload: reload the config and the queries set to be launched at start up
- close selected node
- open selected node

This particular configuration has six dashboards (see 1.5) (although only 4 in the speed dial) , containing data about employees and projects, and one code table, which, for every department code, contains the associated department full name. The latter is used for code resolution (see 1.5).

## 1.5 Dashboard overview

A dashboard is main window or an extracted window which is made of:

- A side bar menu, with a Load menu 1.7 and a Table of content
- One or several pages displaying one or several Widgets. Each page name is listed in the Table of content

A dashboard load data from a main dataset and any number of slaves datasets. These data can be displayed in any number of pages by several types of widgets.

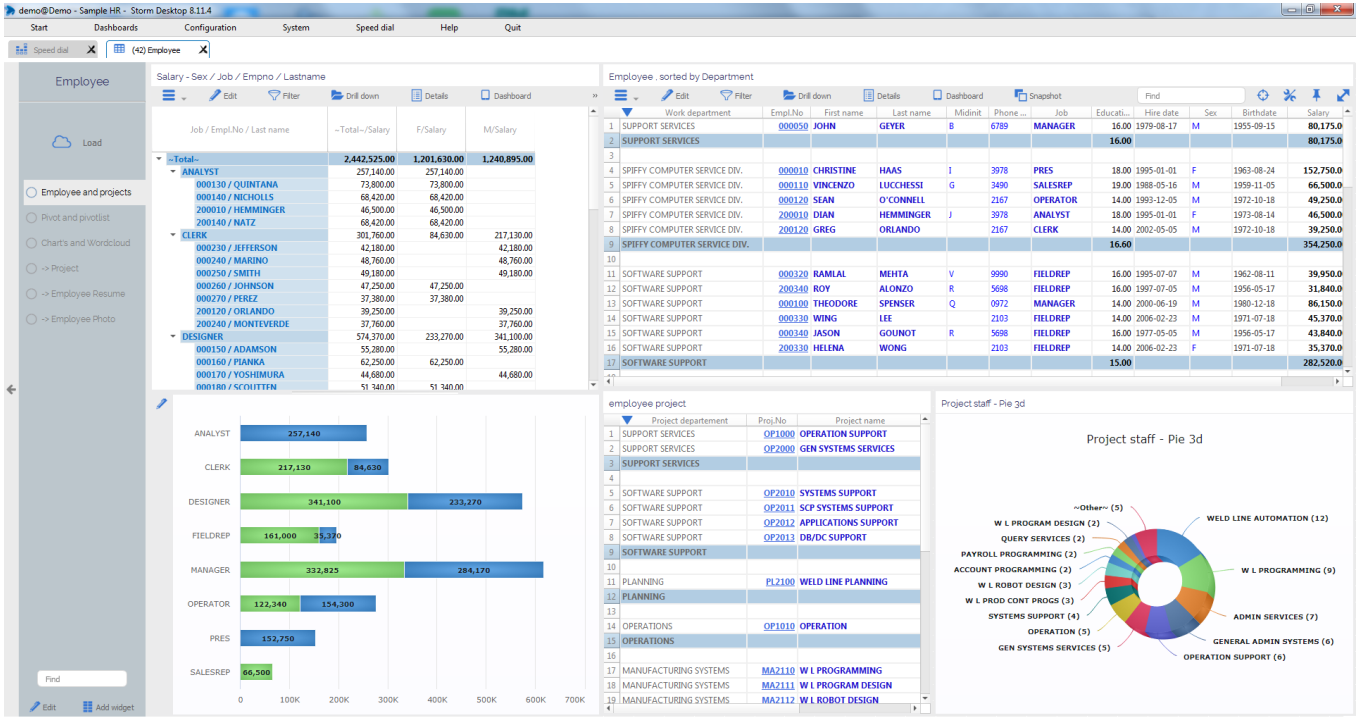



Figure 1.5:
Toolbar:
- hide the toolbar
- Edit: open dashboard context menu (see 2.1.1)
- Add widget: chose a dataset and add a widget
- Find bar: enlighten in green all data matching input string

Dashboard:
- right-clicking on an empty space: open dashboard context menu (see 2.1.1)
- enlarge the widget and fill the page
- a screenshot of the widget will be saved in the clipboard, and can be pasted in any environment that supports it

The first page "employees and projects" shows the most common widget of Storm: A Pivot (with a bar graph), two lists (one master one slave), and pie chart.

### Code resolution

In the Starmenu there are some tables labeles as "Code". They are used for code resolution.

**Concept: Code resolution**

A code is a pair: a symbol and a description of such symbol. In a context of countries, CH stands for Switzerland for example. Codes are often used in SQL databases to reduce the size of a table and are generally stored in a separate table. Storm is able to display the value to which a code refers without actually joining the tables in the query, thus saving resources. To do this, Storm caches the code table in memory and then uses it wherever required.

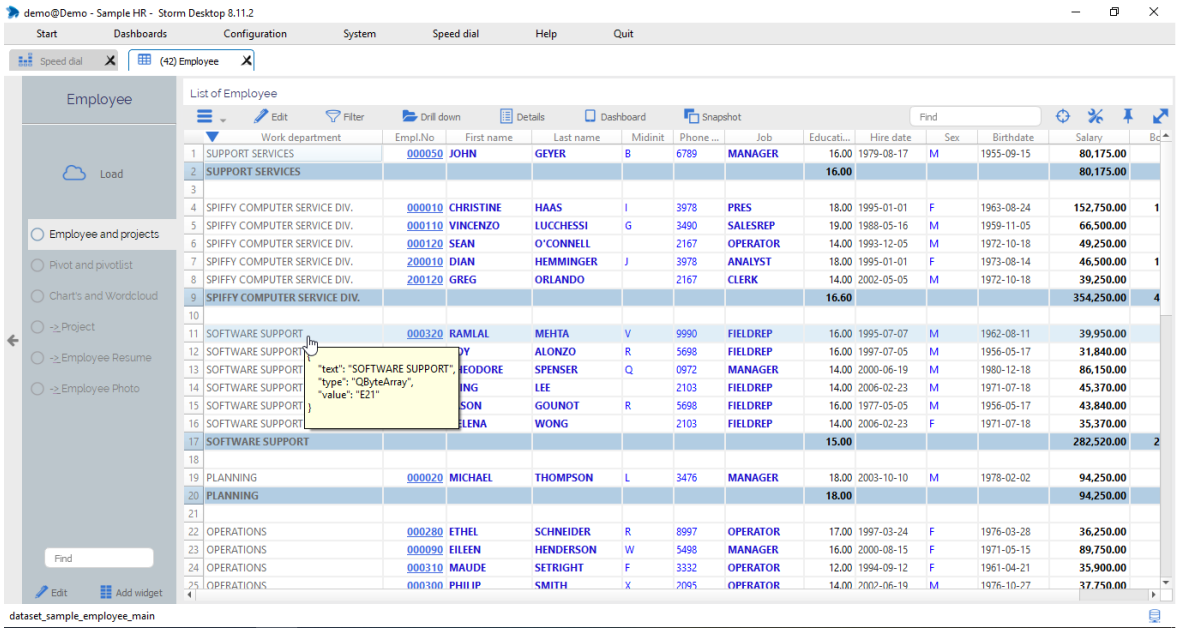



Figure 1.6: code E21 resolved as SOFTWARE SUPPORT

## 1.6 Slave page

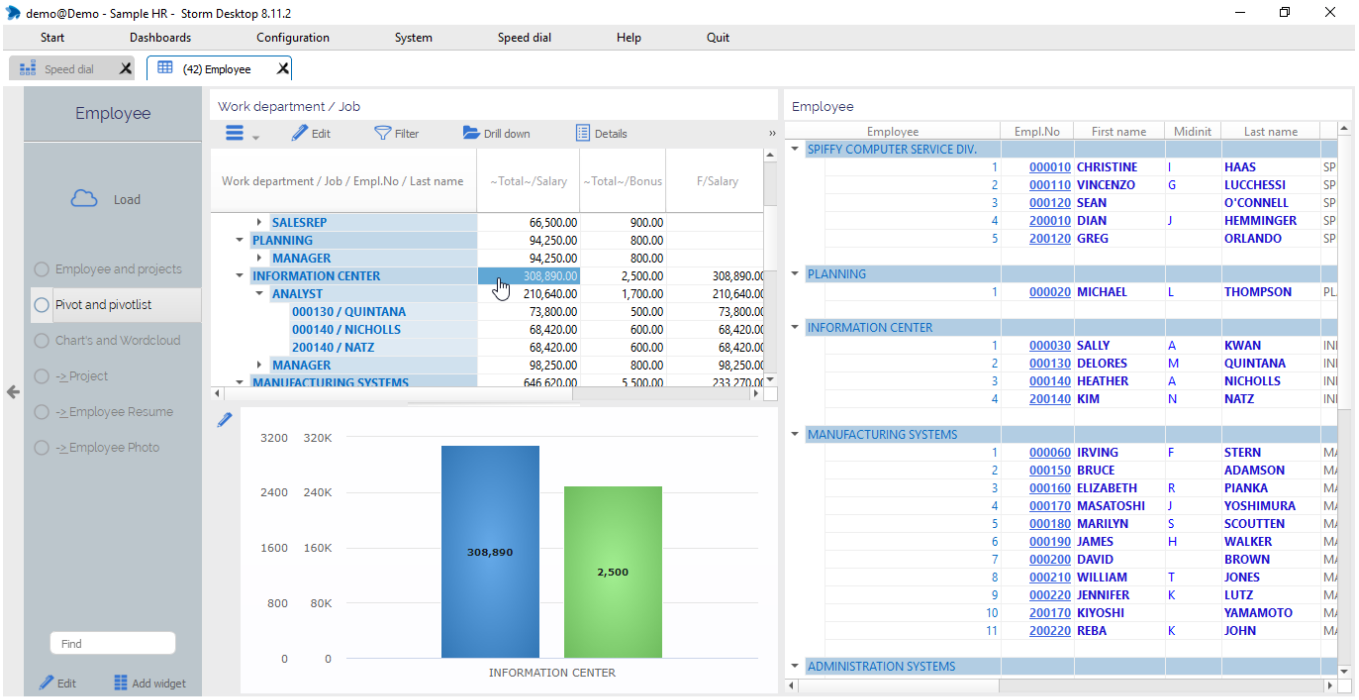
The last page "Employee photo "of the dashboard "Employees and projects" displays the list of the employees who have a photo in the human resource database This page contains a list which is tied to the previous list by a master-slave relation:

This mechanism works in the same way if the master dataset is displayed though a list, a total line of a list or a pivot: all data grouped in the same selectable object will be used to populate the slave.

**Concept: Slave widget**

A slave widget is a widget whose context is given by data from another widget. The relation between the two is called a *master-slave* relation. The relation is established by defining a link between the fields of the two data sources from which the widgets get the data. If a FK exists in the database, such link is created automatically by StormConnect when generating the configuration. Alternatively, it can be defined in the Key Maker (see section **??**) or using an editor (see subsection **??**). Once such link is defined, the slave widget will load only the records based on the values on the master widget.

Selecting one or more records in the master dataset will automaticall filter the slave



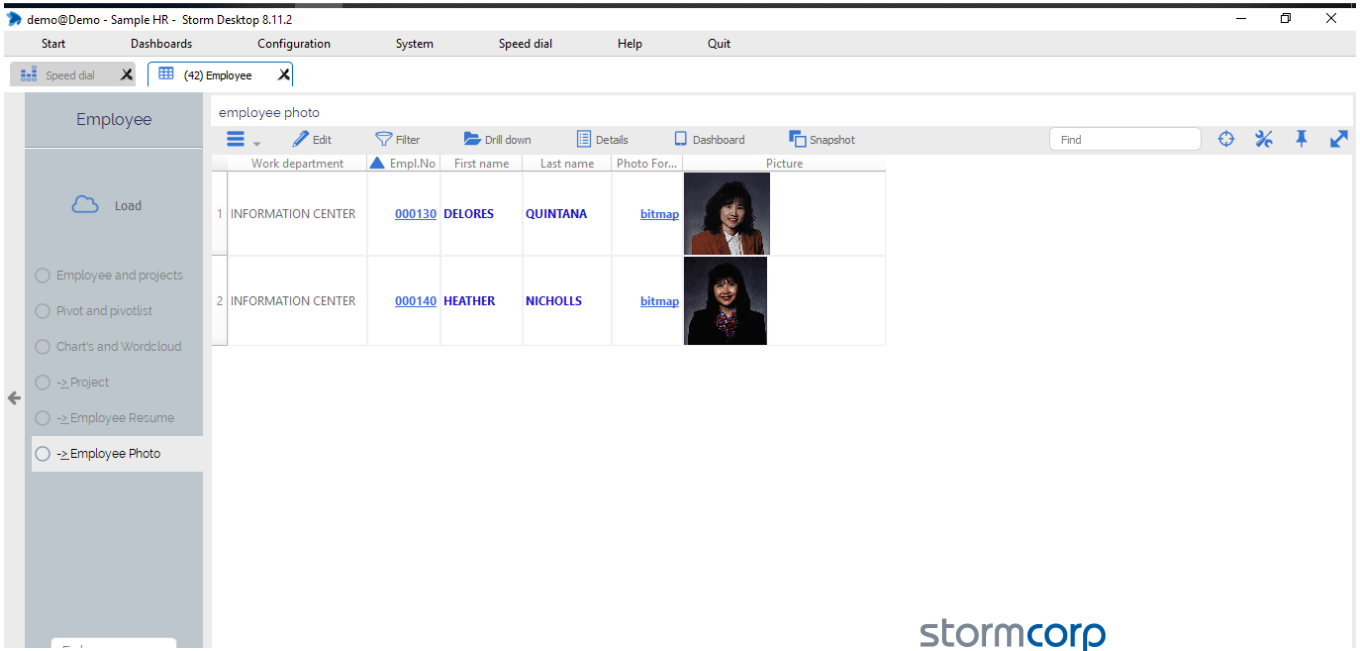




Figure 1.7:
• click on a record to have the slave instantly filtered basing on selection
• click on a total cell is equivalent to select all its grouped record. This includes:
  o subtotal lines in lists or pivotlists
  o total and subtotal of lines or columns in pivots
  o individual cells in pivot
• ctrl+click to select multiple records, cells or lines
• when nothing is selected all the records shown in the master list will be considered

stormcorp

By selecting one department (as example: [Information Center,Total]) all their employees are considered

Master-slave is a relationship between widget's dataset, not pages. A page can have multiple masters and slaves

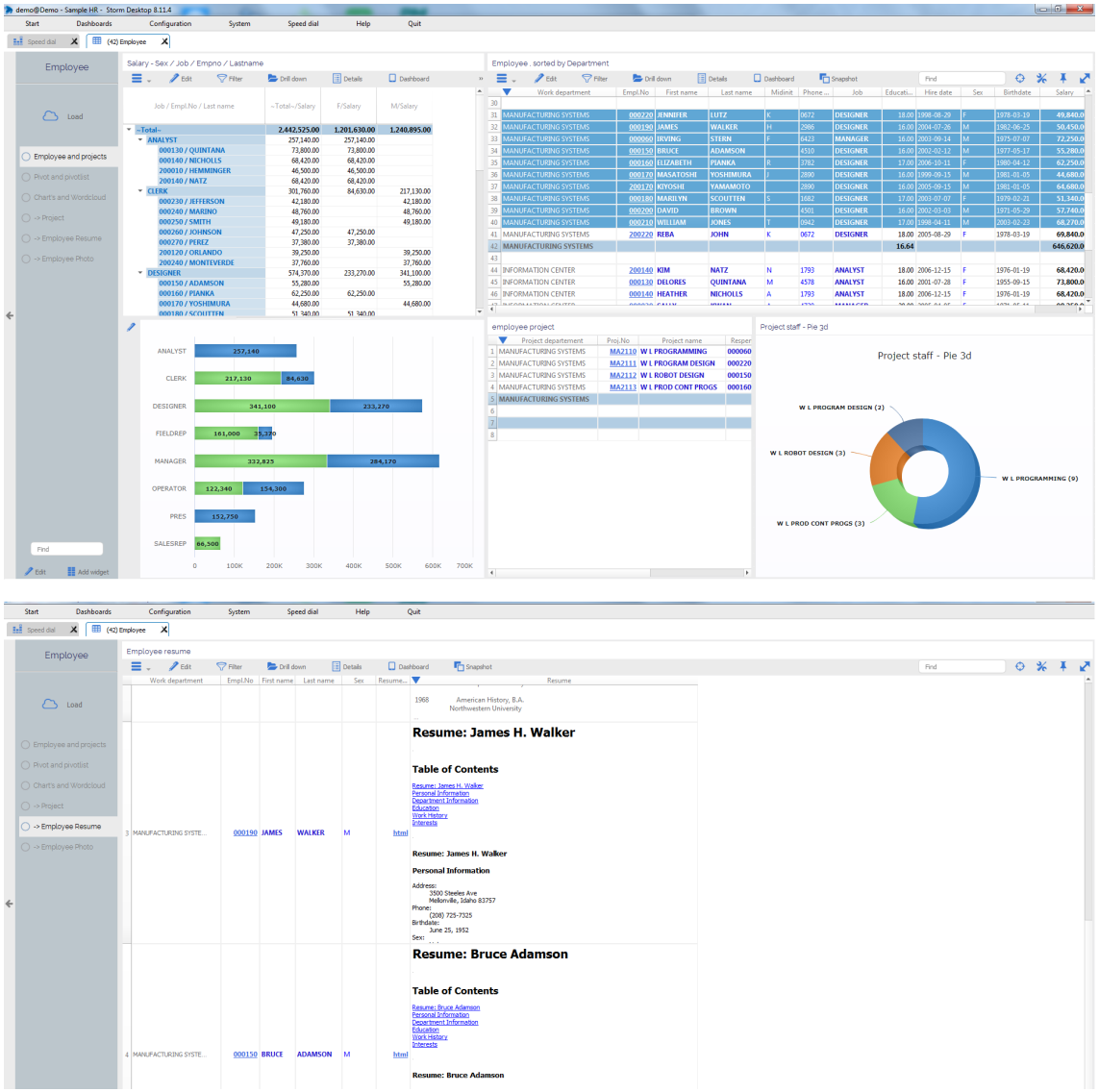



Figure 1.8:
• When more than one master widget is present, the last one seleter will be used to filter the slaves
• Graph of pivots are not slaves, they will show the data of their own selection.
• Slave widgets can be on the same page or on different page

A slave can be in a different page "−− > Project" or in the same page "chart: Project staff - Pie 3d".
Select first "Total Salary" in pivot, then "SPIFFY COMPUTER SERVICE DIV" in the list: the slaves will be filtered by this department, while the graph of the pivot will remain with previous selection.

## 1.7 Load

Open "Employee". This dashboard is set to load automatically the data all employees, displayed in several pages and grouped in several widgets. The user who wants to work with different data shall click on *Load*. The param menu will appear.

On the left side of the tab it is possible to see, add and remove the possible param. Note that Query Limit (1000) has a value by default. Any param can have a default value assigned

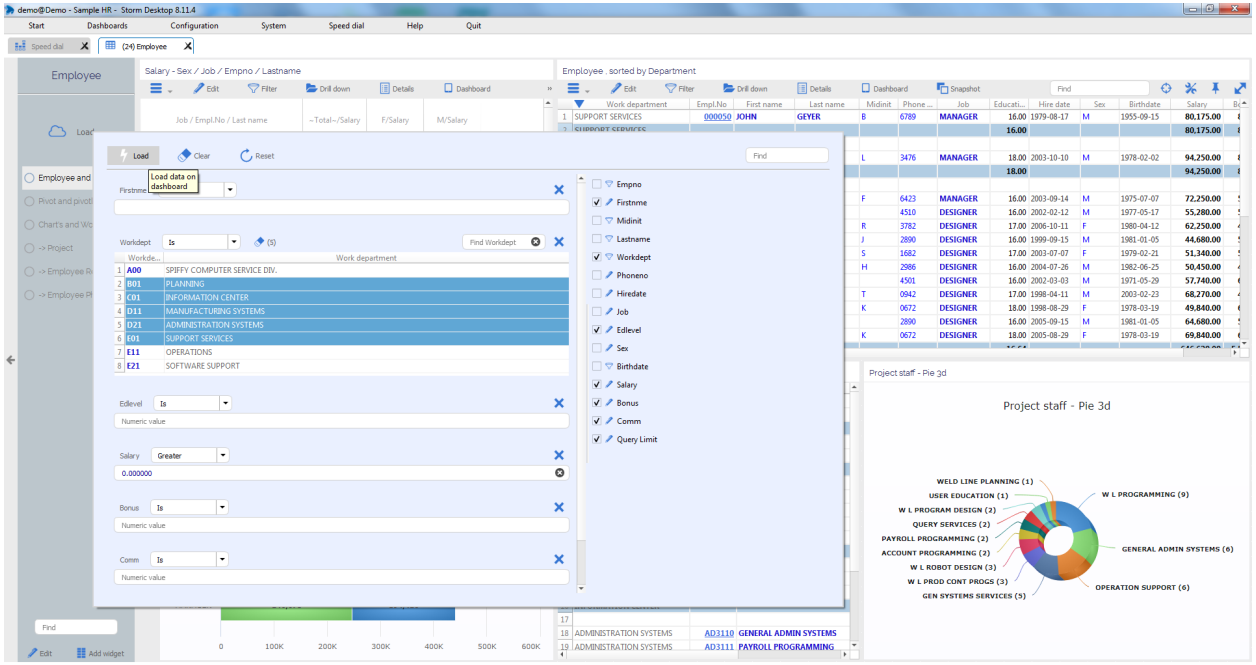



Figure 1.9:

- Load
  - o Default filters: set default values in param editor
  - o Scrolling: scroll the right bar to see more params
  - o Reset: reset all filters to default values
  - o Clear: remove all non mandtory filters
  - o Add / Remove filters
  - o Load: filter the query and load the data
- Parameters
  - o Filter condition: choose among many operators
  - o Search: hide values not matching your input
  - o Multi-selection: ctrl + click
  - o Right click menu: same as in a widget list

## 1.8 Drill down

After selecting any record (from any list, the pivot or pivotlist) and clicking on "drill down" the user can see the context tree. By double-clicking on "Project activities" a new dashboard will open, containing the data related to the activities included in the project for which only the selected employees are responsible. A single click only shows the records count. Alternatively it is possible to load all the records counts first, by clicking on "Load tree" and optionally open a selected dashboard by clicking on "Open node". While the process of selection works exactly as for the slave widget, the slave and the context tree are separate object: while the slave widget is loaded automatically the context tree is hierarchy-free, and the new page could have its own slaves and drill down.

**Concept: Context tree**

This is the menu that displays the data drill down possibilities from the current table/position.

Drill down works like master-slave, in terms of data, but opens a new dashboard and is not limited by two levels only

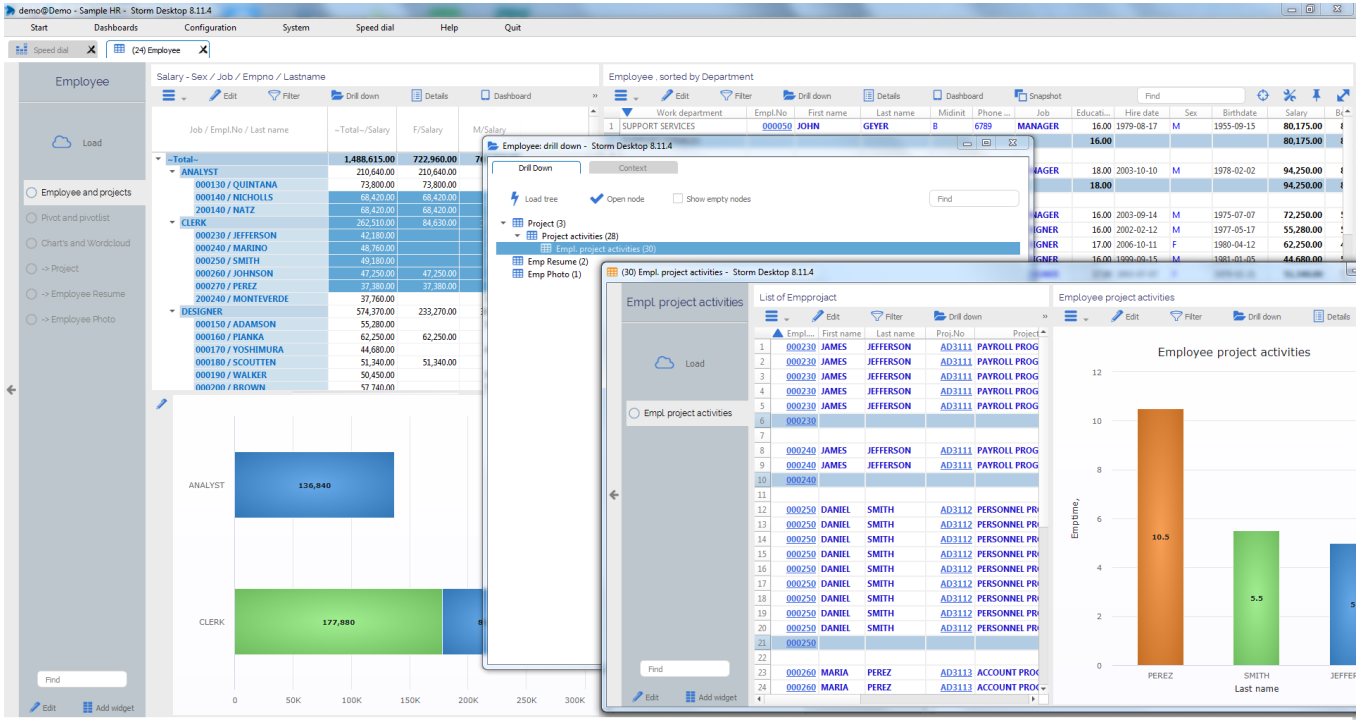



Figure 1.10:
1- Select one or more records, lines or cells
2- Click on "Drill down" to open the menu
• Click once on a node to calculate the number of records for that node
o Double-click on a node to open that dashboard
• Optionally, click on Load tree to calculate the number of records for each node of the tree

Drilling down in two steps, from "Employee" to "Project" and then from "Project" to "Project Activities", gives the same result than a one step drill down from "Employee" to "Project Activities"

## 1.9 List

A list is the basic widget: a sequence of columns containing records from database in a specified sorting.

**Concept: Widget**

A widget is a display of data. Lists, pivots, charts and maps take data from a query and display them in an interactive graphic interface. Each widget type has its own peculiarities, but many allows to filter data and select a group of records.

On the top right of the first page we can see the list of the employees. To focus on the list and work there properly the user should click on "maximize" button .

Select records to work with: click for one row, click and drag to select a range of rows, shift + click for multiple selection

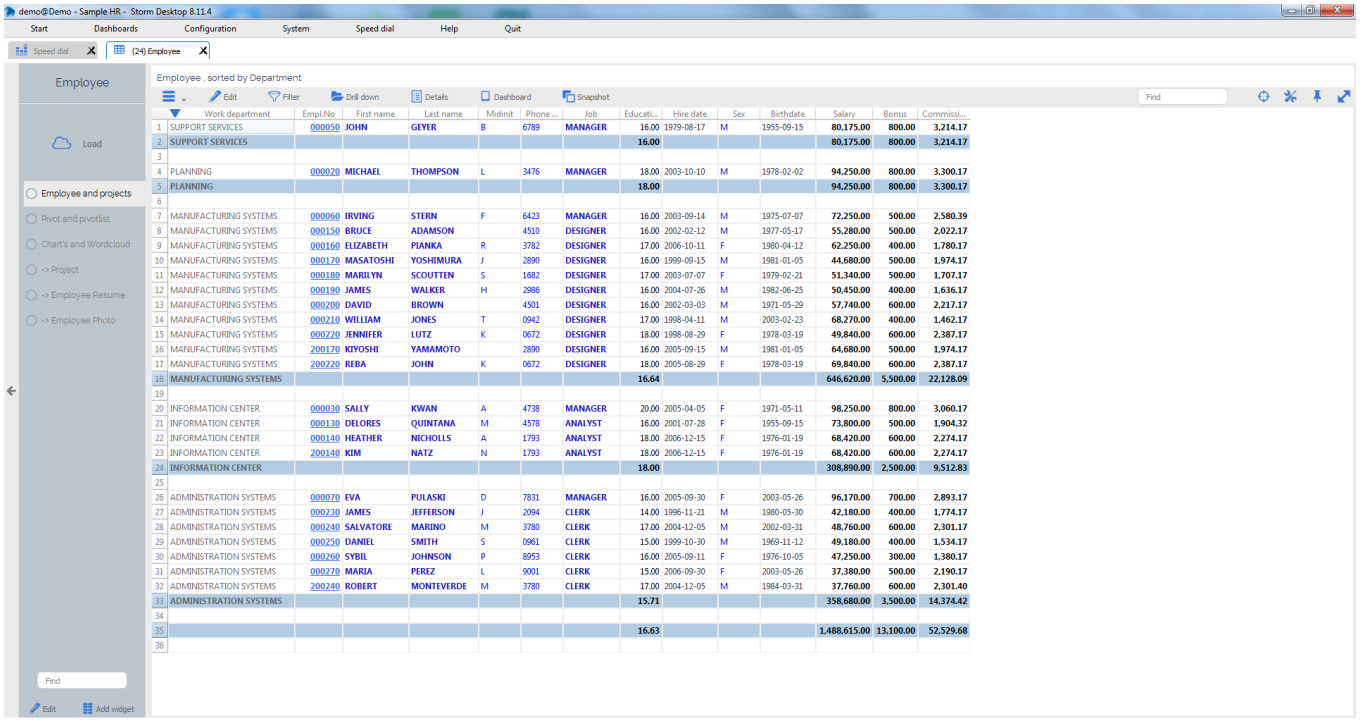



Figure 1.11: On the top of any widget there is a toolbar:

- Menu: select other widger
- Edit: modify current widget
- Filter: apply js filter to the data
- Drill-down: see 1.8
- Details: open datagrid with extra data
- Dashboard: restart the dashboard
- Snapshot: generate a widget with selected data
- Find bar: enlighten all cells matching input string
- Screenshot: take a screenshot
- Tools: hide the toolbar (reappers when hooverin on it)
- Pin: Block/unblock widget position and size
- Maximize: enlarge current widget to fill the window

When the list is bigger than the screen, two bars allow for horizontal and vertical scrolling

## List and widget menu

In a list it is possible to sort and group data according to any column or group of columns. This and other lists can be found under the *List* panel, in the *Widget Menu* located in the upper left corner of the list. By clicking on it and then selecting the desired list the data sorting will change accordingly.

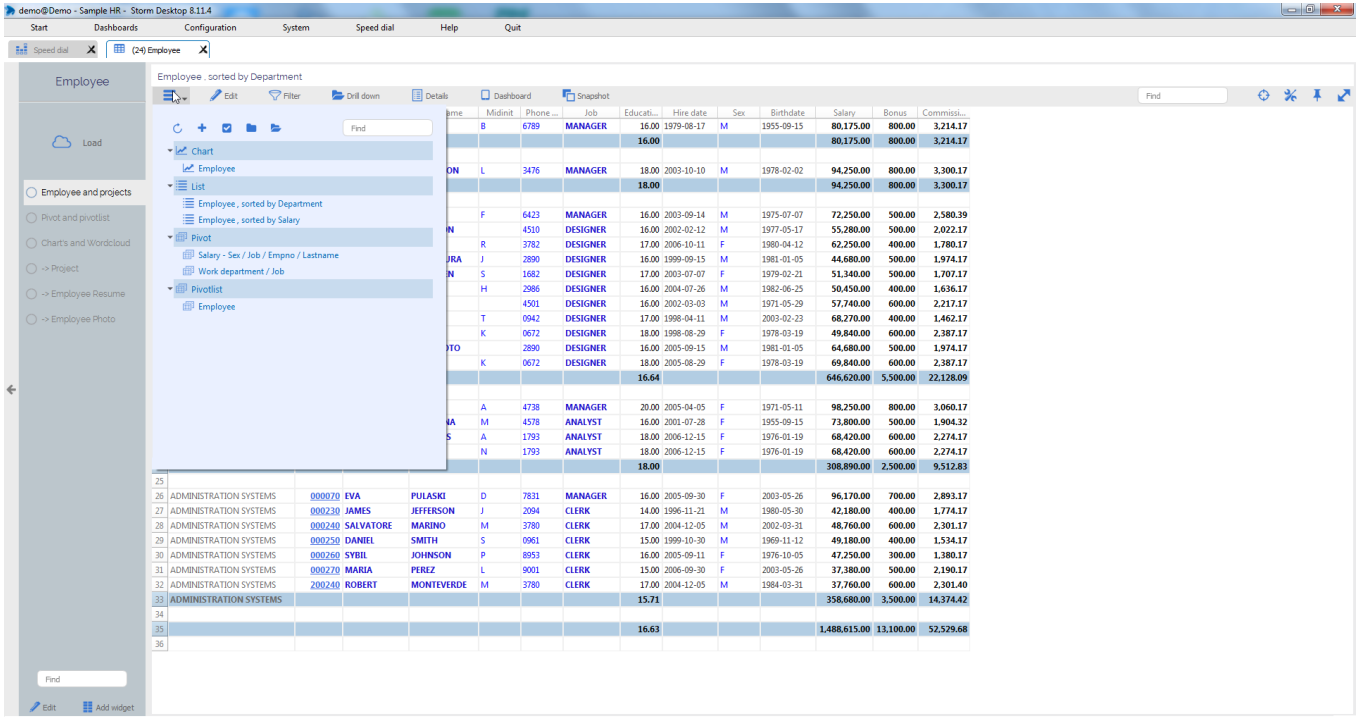
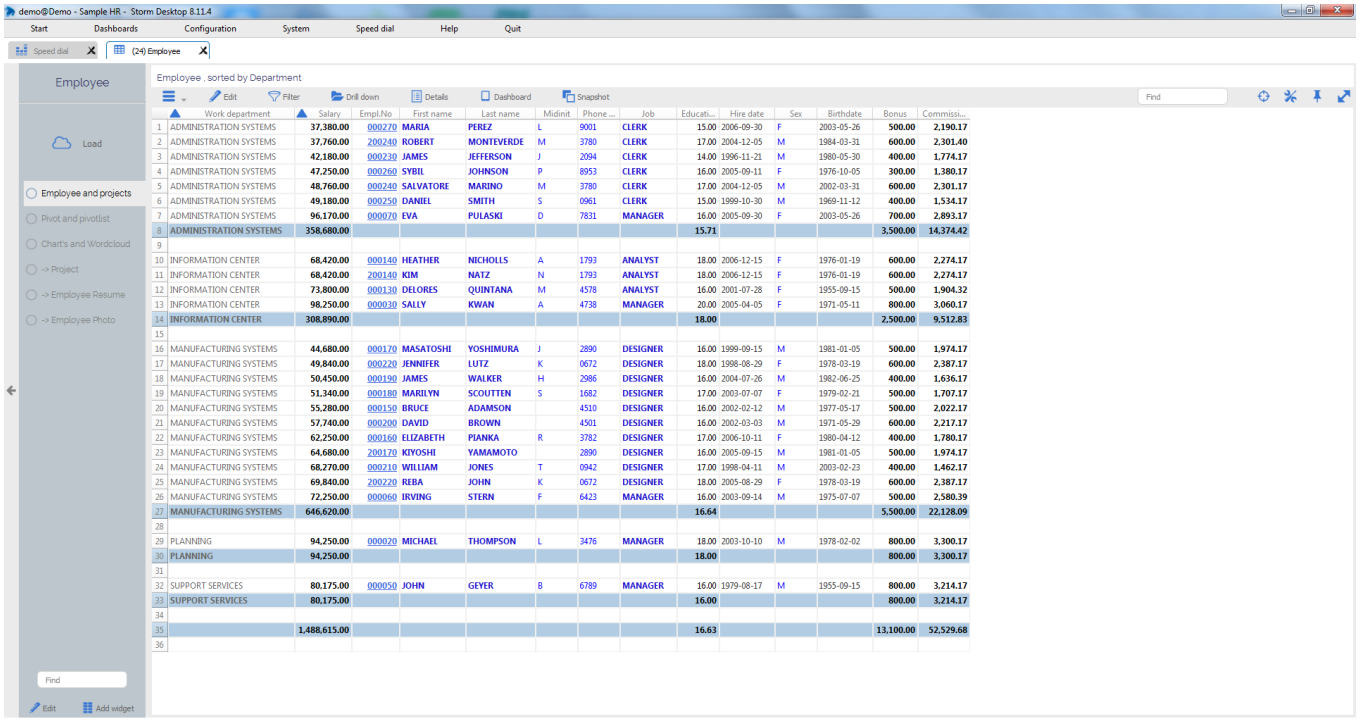



Figure 1.12: Available widgets are defined in the configuration. It is possible to select any type of widget

This list is "Employee sorted by salary". It determines both the columnset and the sorting

### List interactive sorting

Beside the predefined sorting selectable form the menu, sorting and subtotals can be configured manually by clicking on column headers:




| | Work department | Salary | Empl.No | First name | Last name | Midinit | Phone ... | Job | Educati... | Hire date | Sex | Birthdate | Bonus | Commissi... |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ADMINISTRATION SYSTEMS | 37,380.00 | 000270 | MARIA | PEREZ | L | 9001 | CLERK | 15.00 | 2006-09-30 | F | 2003-05-26 | 500.00 | 2,190.17 |
| 2 | ADMINISTRATION SYSTEMS | 37,760.00 | 200240 | ROBERT | MONTEVERDE | M | 3780 | CLERK | 17.00 | 2004-12-05 | M | 1984-03-31 | 600.00 | 2,301.40 |
| 3 | ADMINISTRATION SYSTEMS | 42,180.00 | 000230 | JAMES | JEFFERSON | J | 2094 | CLERK | 14.00 | 1996-11-21 | M | 1980-05-30 | 400.00 | 1,774.17 |
| 4 | ADMINISTRATION SYSTEMS | 47,250.00 | 000260 | SYBIL | JOHNSON | P | 8953 | CLERK | 16.00 | 2005-09-11 | F | 1976-10-05 | 300.00 | 1,380.17 |
| 5 | ADMINISTRATION SYSTEMS | 48,760.00 | 000240 | SALVATORE | MARINO | M | 3780 | CLERK | 17.00 | 2004-12-05 | M | 2002-03-31 | 600.00 | 2,301.17 |
| 6 | ADMINISTRATION SYSTEMS | 49,180.00 | 000250 | DANIEL | SMITH | S | 0961 | CLERK | 15.00 | 1999-10-30 | M | 1969-11-12 | 400.00 | 1,534.17 |
| 7 | ADMINISTRATION SYSTEMS | 96,170.00 | 000070 | EVA | PULASKI | D | 7831 | MANAGER | 16.00 | 2005-09-30 | F | 2003-05-26 | 700.00 | 2,893.17 |
| 8 | ADMINISTRATION SYSTEMS | 358,680.00 | | | | | | | 15.71 | | | | 3,500.00 | 14,374.42 |
| 9 | | | | | | | | | | | | | | |
| 10 | INFORMATION CENTER | 68,420.00 | 000140 | HEATHER | NICHOLLS | A | 1793 | ANALYST | 18.00 | 2006-12-15 | F | 1976-01-19 | 600.00 | 2,274.17 |
| 11 | INFORMATION CENTER | 68,420.00 | 200140 | KIM | NATZ | N | 1793 | ANALYST | 18.00 | 2006-12-15 | F | 1976-01-19 | 600.00 | 2,274.17 |
| 12 | INFORMATION CENTER | 73,800.00 | 000130 | DELORES | QUINTANA | M | 4578 | ANALYST | 16.00 | 2001-07-28 | F | 1955-09-15 | 500.00 | 1,904.32 |
| 13 | INFORMATION CENTER | 98,250.00 | 000030 | SALLY | KWAN | A | 4738 | MANAGER | 20.00 | 2005-04-05 | F | 1971-05-11 | 800.00 | 3,060.17 |
| 14 | INFORMATION CENTER | 308,890.00 | | | | | | | 18.00 | | | | 2,500.00 | 9,512.83 |
| 15 | | | | | | | | | | | | | | |
| 16 | MANUFACTURING SYSTEMS | 44,680.00 | 000170 | MASATOSHI | YOSHIMURA | J | 2890 | DESIGNER | 16.00 | 1999-09-15 | M | 1981-01-05 | 500.00 | 1,974.17 |
| 17 | MANUFACTURING SYSTEMS | 49,840.00 | 000220 | JENNIFER | LUTZ | K | 0672 | DESIGNER | 18.00 | 1998-08-29 | F | 1978-03-19 | 600.00 | 2,387.17 |
| 18 | MANUFACTURING SYSTEMS | 50,450.00 | 000190 | JAMES | WALKER | H | 2986 | DESIGNER | 16.00 | 2004-07-26 | M | 1982-06-25 | 400.00 | 1,636.17 |
| 19 | MANUFACTURING SYSTEMS | 51,340.00 | 000180 | MARILYN | SCOUTTEN | S | 1682 | DESIGNER | 17.00 | 2003-07-07 | F | 1979-02-21 | 500.00 | 1,707.17 |
| 20 | MANUFACTURING SYSTEMS | 55,280.00 | 000150 | BRUCE | ADAMSON | | 4510 | DESIGNER | 16.00 | 2002-02-12 | M | 1977-05-17 | 500.00 | 2,022.17 |
| 21 | MANUFACTURING SYSTEMS | 57,740.00 | 000200 | DAVID | BROWN | | 4501 | DESIGNER | 16.00 | 2002-03-03 | M | 1971-05-29 | 600.00 | 2,217.17 |
| 22 | MANUFACTURING SYSTEMS | 62,250.00 | 000160 | ELIZABETH | PIANKA | R | 3782 | DESIGNER | 17.00 | 2006-10-11 | F | 1980-04-12 | 400.00 | 1,780.17 |
| 23 | MANUFACTURING SYSTEMS | 64,680.00 | 200170 | KIYOSHI | YAMAMOTO | | 2890 | DESIGNER | 16.00 | 2005-09-15 | M | 1981-01-05 | 500.00 | 1,974.17 |
| 24 | MANUFACTURING SYSTEMS | 68,270.00 | 000210 | WILLIAM | JONES | T | 0942 | DESIGNER | 17.00 | 1998-04-11 | M | 2003-02-23 | 400.00 | 1,462.17 |
| 25 | MANUFACTURING SYSTEMS | 69,840.00 | 200220 | REBA | JOHN | K | 0672 | DESIGNER | 18.00 | 2005-08-29 | F | 1978-03-19 | 600.00 | 2,387.17 |
| 26 | MANUFACTURING SYSTEMS | 72,250.00 | 000060 | IRVING | STERN | F | 6423 | MANAGER | 16.00 | 2003-09-14 | M | 1975-07-07 | 500.00 | 2,580.39 |
| 27 | MANUFACTURING SYSTEMS | 646,620.00 | | | | | | | 16.64 | | | | 5,500.00 | 22,128.09 |
| 28 | | | | | | | | | | | | | | |
| 29 | PLANNING | 94,250.00 | 000020 | MICHAEL | THOMPSON | L | 3476 | MANAGER | 18.00 | 2003-10-10 | M | 1978-02-02 | 800.00 | 3,300.17 |
| 30 | PLANNING | 94,250.00 | | | | | | | 18.00 | | | | 800.00 | 3,300.17 |
| 31 | | | | | | | | | | | | | | |
| 32 | SUPPORT SERVICES | 80,175.00 | 000050 | JOHN | GEYER | B | 6789 | MANAGER | 16.00 | 1979-08-17 | M | 1955-09-15 | 800.00 | 3,214.17 |
| 33 | SUPPORT SERVICES | 80,175.00 | | | | | | | 16.00 | | | | 800.00 | 3,214.17 |
| 34 | | | | | | | | | | | | | | |
| 35 | | 1,488,615.00 | | | | | | | 16.63 | | | | 13,100.00 | 52,529.68 |
| 36 | | | | | | | | | | | | | | |

Figure 1.13:

- Click on a column header to sort by that column
  - o Click twice when selecting a column to reverse the sort
- Ctrl + click on a column to sort by that column after the previous sortings are applied.
  - o Columns involved in the sorting are highlighted by a triangle, whose verse depends on the verse of the sorting
- Shift + click on a column to sort by that column and calculate the subtotal.
  - o Ctrl+ shift + click to have both effects

Sorting on the left is obtained by first sorting by department with total, then sorting by job with total, then sorting by salary (decreasing) without total

## List right-click menu

Right click on the list to access its menu:

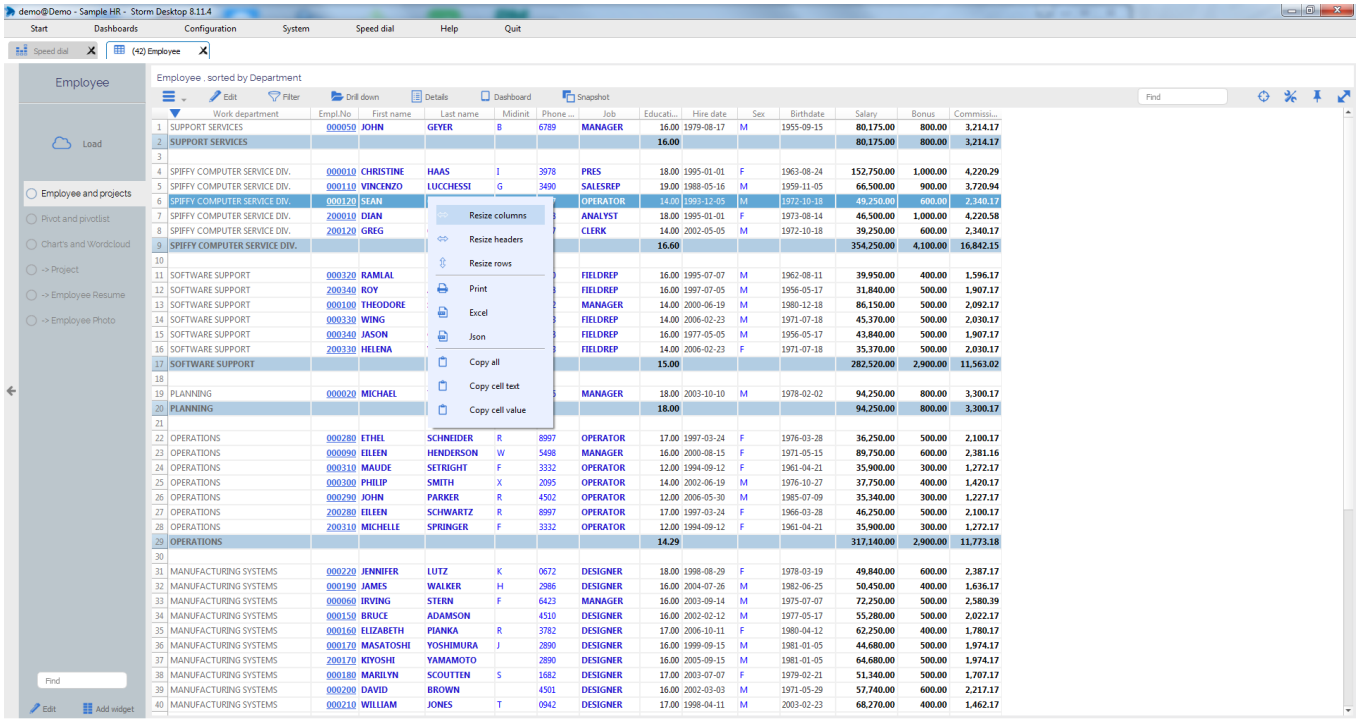



Figure 1.14:

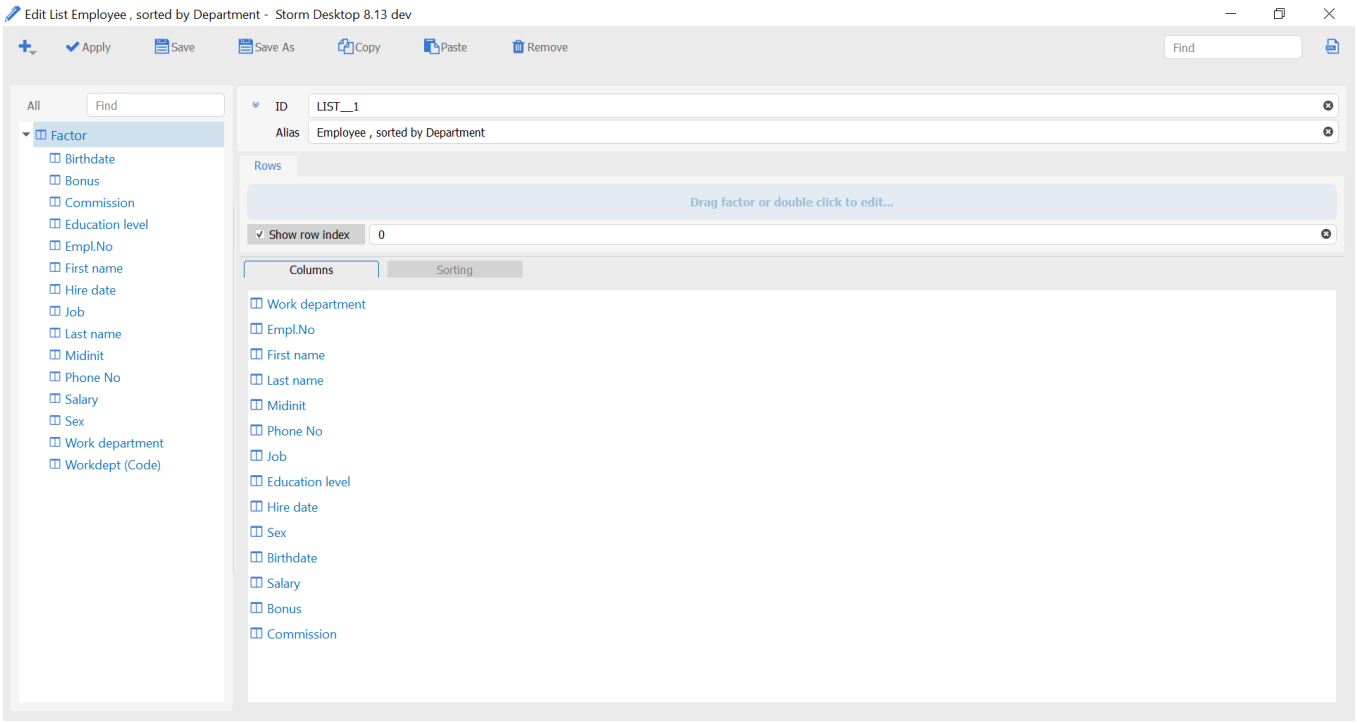
- Resize
  - Columns
  - Headers
  - Rows
- Save as
  - Pdf
  - Excel
  - Json
- Copy
  - All
  - Text
  - Cell value

Exported excel and json files can be re-opened by storm as datasource

### List edit columns

Click on Edit to open list editor: it consists of two tabs:

Column

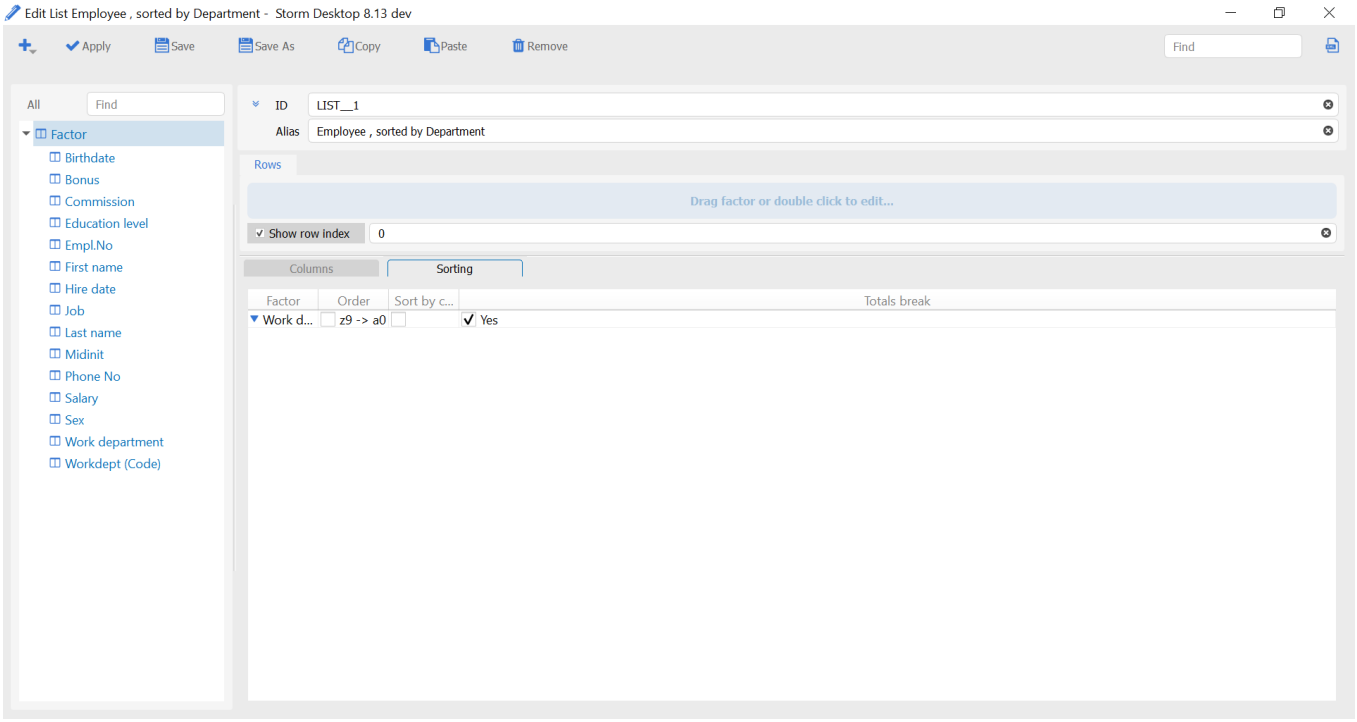



Setting Row height as 0 will leave it with default size

Figure 1.15:
• Drag and drop factors from left to the right to add them to the list
  o Drag and drop factors from left to *Row* to set a row header
• By default only a limited amount of columns is visible on the left menu: click on "All" to see more
• Use "Find" menu to filter out columns
• Row height is also editable in this menu

## List edit sorting

Sorting




Sort by department, decreasing, with subtotals

Figure 1.16:
• Drag and drop columns from left to the right to include them in the sorting
• Drag and drop columns inside the sorting to define hierarchy
• For each column there are 3 options:
  o Order
  o Sort by code (see 1.5 )
  o Calculate subtotal
• By default only a limited amount of columns is visible on the left menu: click on ”All” to see more
• Use ”Find” menu to filter out columns

## 1.10 Pivot

The pivot is a table where one (or more) numeric (but not only) columns, called "values" are totalized according to a specific js function "total" and grouped by some dimensions "rows and columns". On the rows it is possible to have several columns forming a descending level. Below the pivot there is the option to have a graph, synchronized to the pivot, that displays the same data. Hovering can reveal more information. Click on >> on the top right of the pivot, then on to enlarge the pivot and fill the page:

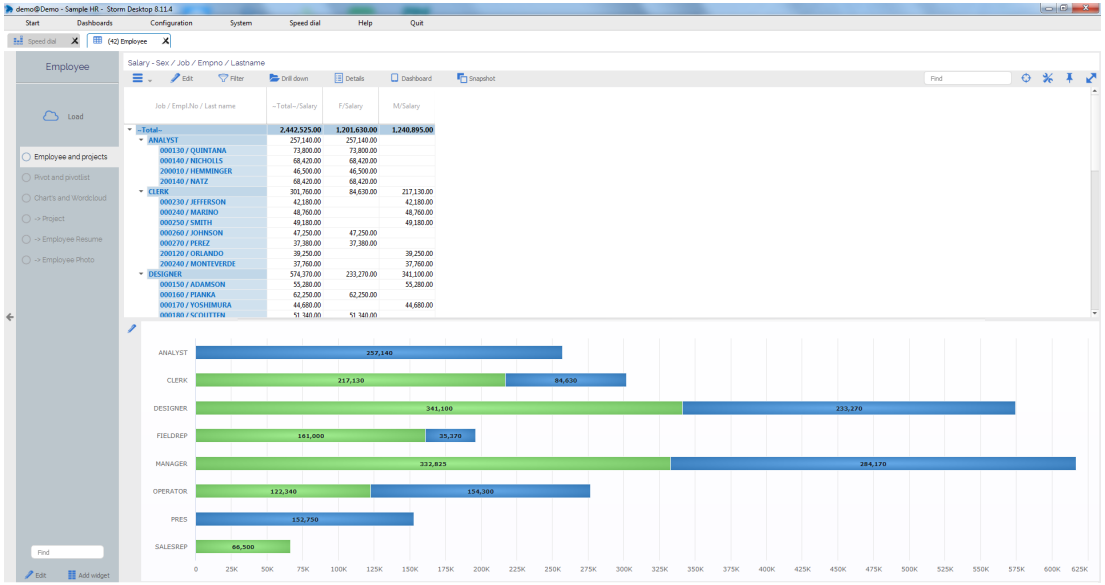



Figure 1.17:
- hide the toolbar
- enlarge the widget and fill the page
- a screenshot of the widget will be saved in the clipboard, and can be pasted in any environment that supports it.

This pivot shows a table counting the sum of the salary of the employees, grouped by job and gender. Below the pivot there is a graph, tied to the pivot, displaying the same data on bars, where data of men and women having the same job appear stacked together.

## Pivot nodes

A pivot is nothing more than a table with multiple levels. Every level can be opened or closed by clicking on the small black triangles. If a node is closed only a line containing the subtotal will be displayed.

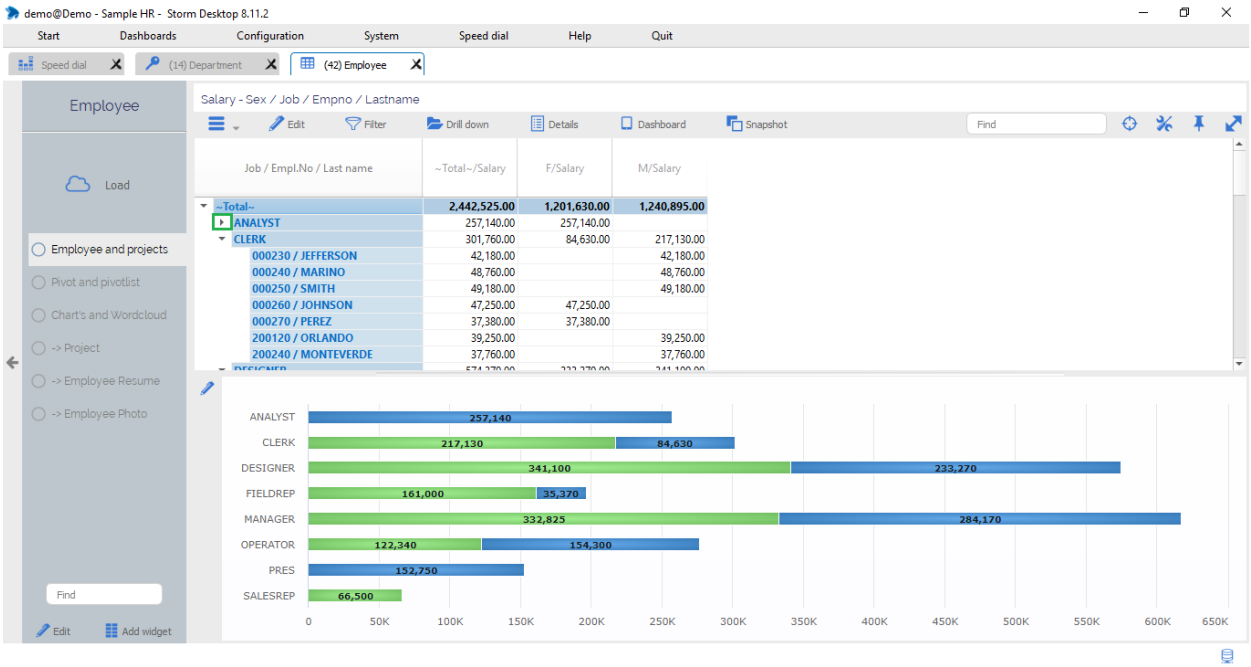



Figure 1.18:
- ▾ close selected node
- ▸ open selected node

### Pivot graph

The graph by default displays all the data. By clicking on some cell the graph will adapt to the selected data.

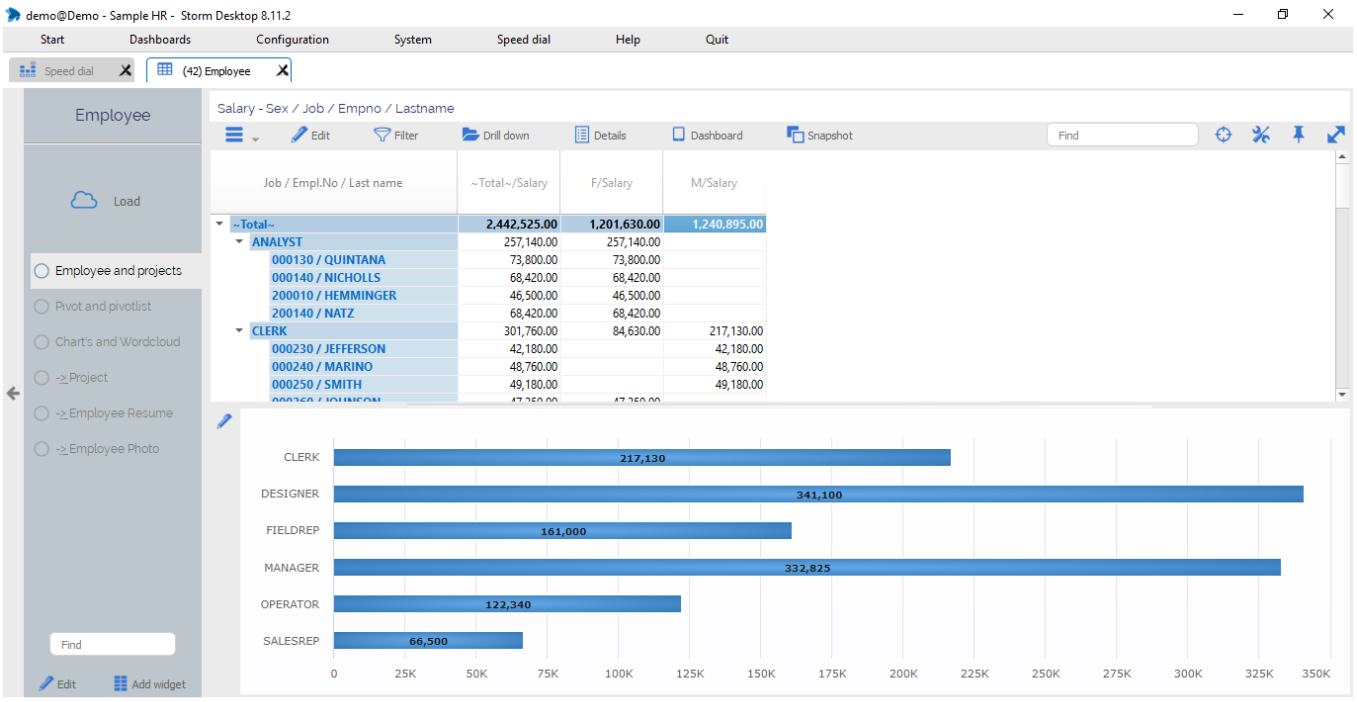



Figure 1.19: Multiple cells can be selected by holding left click and dragging or by ctrl + left click

This graph shows only the salary of the men, for all the jobs

## 1.11 Pivotlist

On the second page "Pivot and pivotlist" the data will appear grouped differently. The pivot on the left side furtherly regroups the employee by department, and next to the salary the bonus is displayed. On the right there is a pivotlist, where the lines contain all the SQL columns (as in a list) but are grouped by department (as they would be in a pivot).

A pivot is nothing more than a list with pivot-like levels on the rows. Every level can be opened or closed by clicking on the small black triangles. If a node is closed only a line containing the subtotals will be displayed.

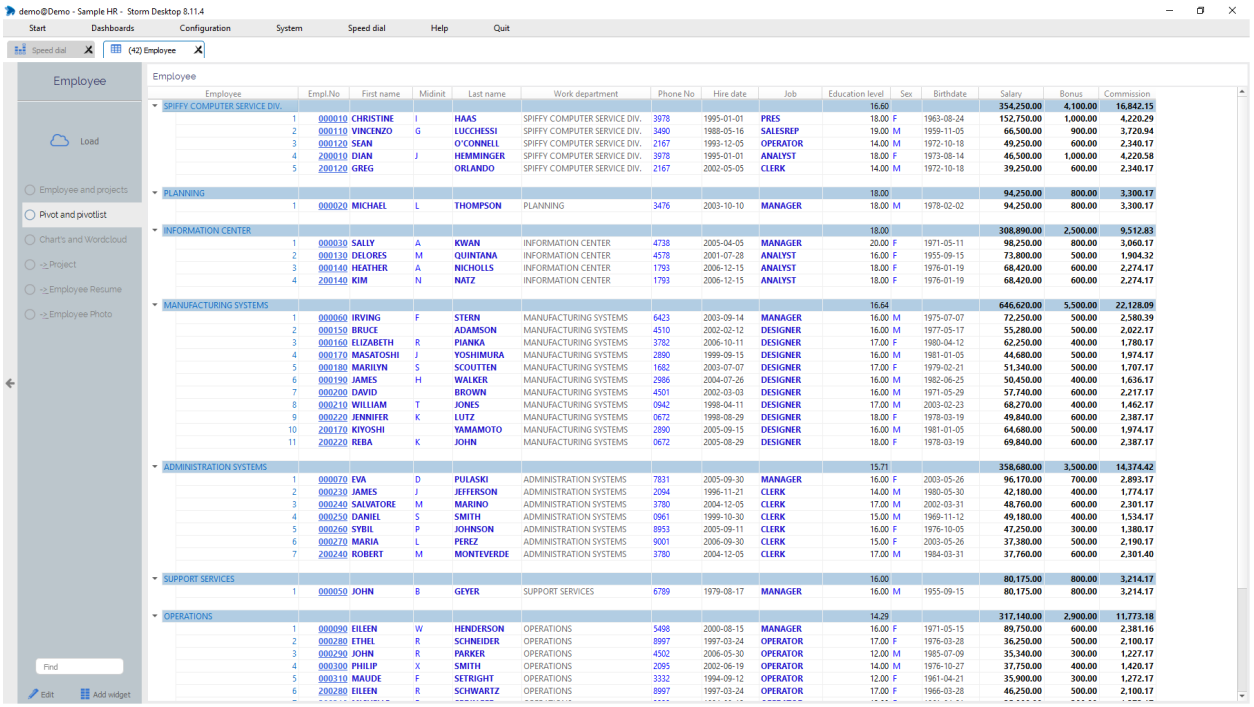



Figure 1.20:
- close selected node
- open selected node

"Salary", "Bonus" and "Commission" have the default total "sum" while "Education Level" has the "average" function.

## 1.12 Chart

A chart is a widget where one (or more) graphs are displayed. In each graph one numeric (but not only) column, called "value" is totalized according to a specific js function "total" and grouped by another column "dimension". Some types of graph allow an additional dimension, called x-axis, which is common among them. Hovering can reveal more information. Click on ⤢ on the top right of the chart to enlarge it and fill the page:

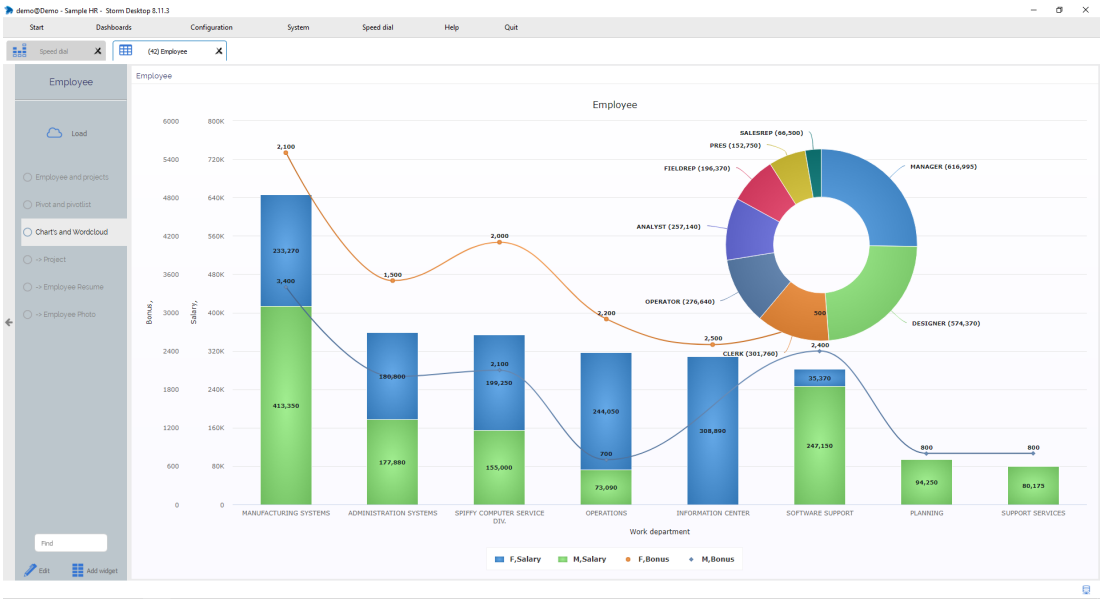



Figure 1.21:
- Many graphs on the same widget
- All two-dimensional graphs share the same axis
- The one-dimentional graphs are all independent from each other

This chart shows three graphs: the columns represent the sum of the salary of the employees, grouped by department and gender. The line represents the bonus, gruped by department and gender. The pie represents the salary, grouped by job. Note that the salary and the bonus share the same x-axis, but have a diffent scale on the y-axis.

# 2 Editing the configuration

StormDesktop allows to perform editing on any configuration, either the Demo contained in the download package or the ones generated by StormConnect. In this chapter the common configuration editing tasks will be treated first, while a reference for each editor is provided in the second section.

## 2.1 Common tasks

In this section some common configuration editing tasks are described. For the meaning of the various editors' sections refer to the 2.2 section.

### 2.1.1 Adding a page to a dashboard

To add a page choose the option *Add page* from the dashboard's *Edit* menu. Alternatively right click on the list of pages and select the *Add page* option.

### 2.1.2 Adding a widget to a dashboard

Open the dashboard's *Add widget* menu . The menu shows the available datasets: the master and all the slaves(see 1.6).

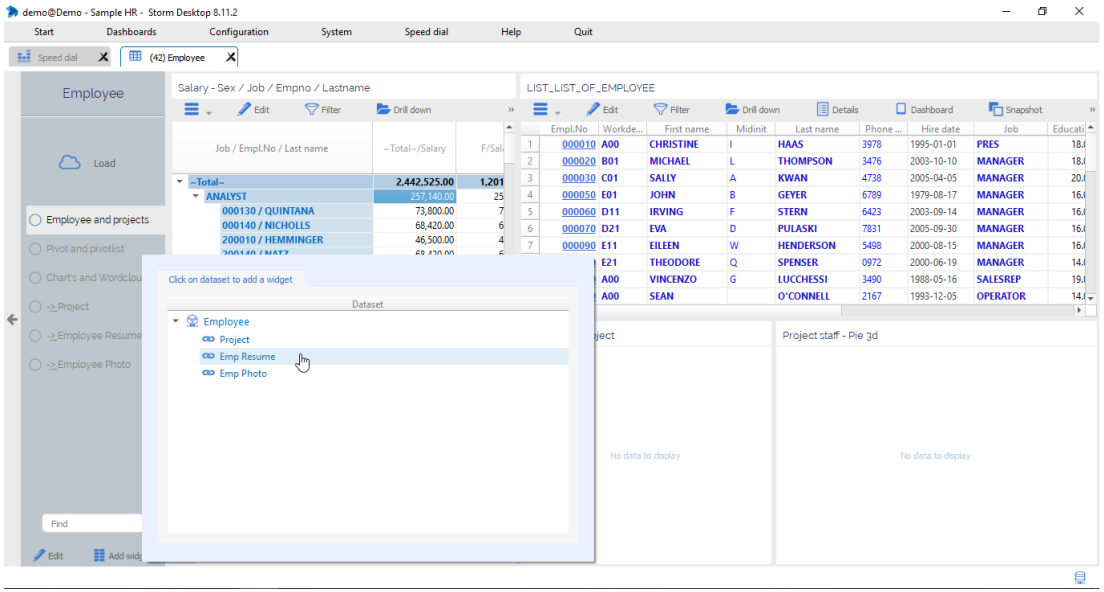



Figure 2.1:
- By right-clicking on the dataset the Dataset editor will open (see **??** )

By default the widget will be a list. however in can be easily switched to another type by clicking on the hamburger menu: .

### 2.1.3 Editing a widget

Each editable widget has an *Edit* button on the top. It opens the editing dialog for the widget. Dialogs vary across widgets so it is better to refer to the description of each particular editor in the next section.

Once finished editing, the user can either click on the *Apply* button to see the effects on the widget, or click on *Save* to make the changes permanent. Note that saving the a widget saves the definition of its' contents, *not* the properties of the container! Thus, if a user added a new widget to a dashboard, edited it and saved the widget, the change will not be saved to the dashboard and, the next time the dashboard is opened, the widget will not be present. What happened is that the definition of the widget now exists in the configuration, but to actually add and display this definition on the dashboard, it is necessary to save both the widget and the dashboard.

### 2.1.4 Adjusting the list

The sequence of columns displayed by a list widget is adjusted from the *Edit* menu of the list widget itself. Drag and drop columns from the left menu to add new columns to the columnset. Adjust the sequence on the right hand of the screen. It is also possible to adjust the sequence from the list widget itself by dragging column headers. Save once finished.

### 2.1.5 Managing the tasks

*Task* panel is used to see and create tasks. An task is some action done in background that will be reported to the user at its completion. On startup system will start the tasks and will start related timers. When the time has come, task's query/dataset/dashboard runs. When data is loaded the user will receive a report. Tasks' reports will be created in the folder set in *Project—>Report*. See 2.3.1

## 2.2 Editors' reference

Here we present an overview of each editor.

### Editors' header

All editors share the same header, shown in the figure.

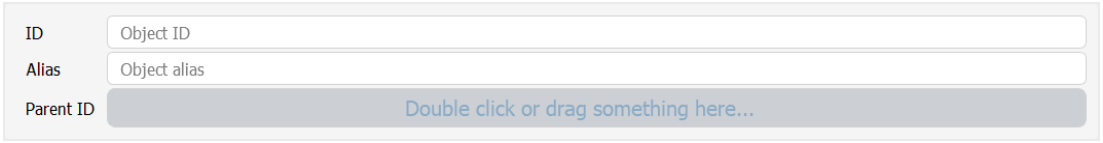


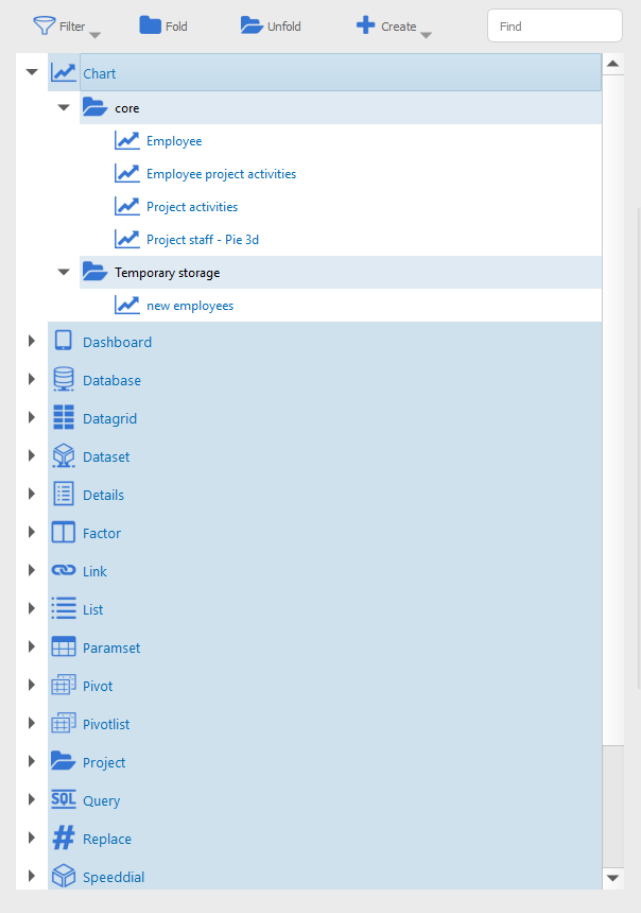

Figure 2.2:
- **Id:** the id of the definition. The creation timestamp is appended to the id.
- **Alias:** the displayed name of the definition.
- **Parent id:** the id of dataset to be used in. Effective only for widgets.

Often widget are directly associated to a dataset. Widget with the same parent id will appear in the same *Widget menu* ≡.

### Objects' selector

Another common element is the object selector, usually located on the left.

From here it is possible to drag and drop desired elements into the editor. The displayed categories vary across editors.




Nodes consist of object type and repository

Figure 2.3:
- Filter by object type
- Fold: close all the nodes
- Unfold: open all the nodes
- Find bar: filters out all dashboard not maching the input string
- Reload: reload the config including the Start Menu
- close selected node
- open selected node

Many editors which are divided into tabs, have a tab dedicated to JavaScript input called *Script*. While the behavior of this script may vary from element to element, the editor is the same, like shown below.

This editor can appear in tabs called "total", "expression" or "script"

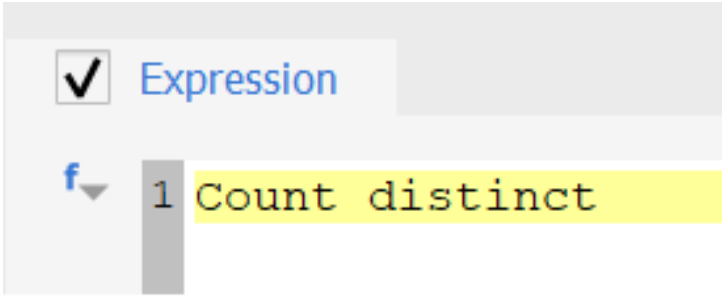



Figure 2.4:
•Script (checkbox):] sets the flag to use the provided JavaScript.
•**f** (menu): selects a JavaScript function from a list.
• **Text box** (box):] a JavaScript script can be written here directly. The last evaluated expression will be returned by the script.

Nodes consist of object type and repository

The following icons appears in many editors even if their position may vary from editor to editor:

Save: saves the definitions in the same object (update).

Save As: saves the definitions as a new object.

Apply: applies the definition without saving.

Clear: clears the definition.

: displays and copies to clipboard the xml code of the current object.

: menu for creating a new object.

: delete from disk the selected definition.

: reload all definitions, resets the filters if any.

>>: move object to the current editor.

Preview: shows a preview of the current object.

All fields of all editors are optional. The ones that are not, have always default values set.

## 2.3 User's Editor reference

### 2.3.1 Task

An task is some action done in background and then user is notified about the result. Editor consists of 4 tabs

Figure 2.5:
**Task:** Sets datasource, report and mail template
**Schedule:** Sets time day and frequency of the task
**Email:** Set mail server and destination
**Script:** Add a script

A notification mail will be sent when the task is completed. A more detailed report will appear in the folder sets in Project–>Report

### Task main tab

First tab *Task* sets the layout and content of the query

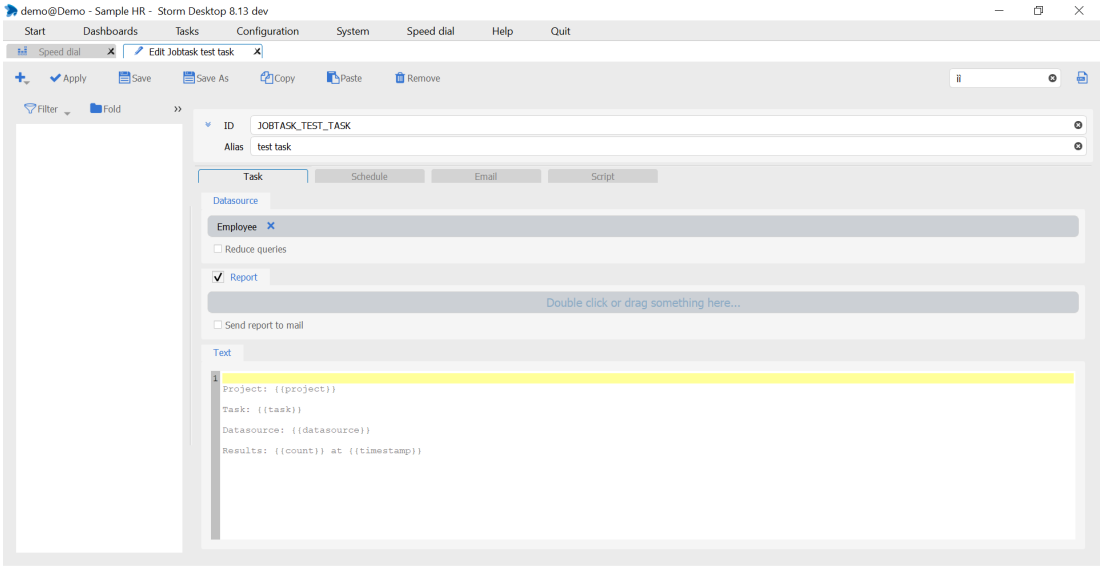



Figure 2.6:
**Datasource:** Drag and drop a dahboard, a dataset or a query from the selector on the left side to this panel
• Reduce queries: tick to use reduced instead of full version of the queries.
**Report:** Drag and drop a report template from selector on the left side to this panel
• Send report to mail: tick to attach the report.pdf in the mail
**Text:** Template of the mail. By default it contains projecm task, datasource, number of recordsa and timestamp

A notification mail will be sent when the task is completed. A more detailed report will appear in the folder sets in Project–>Report

### Tab schedule

Second tab *Schedule* sets the time day and frequency of the task

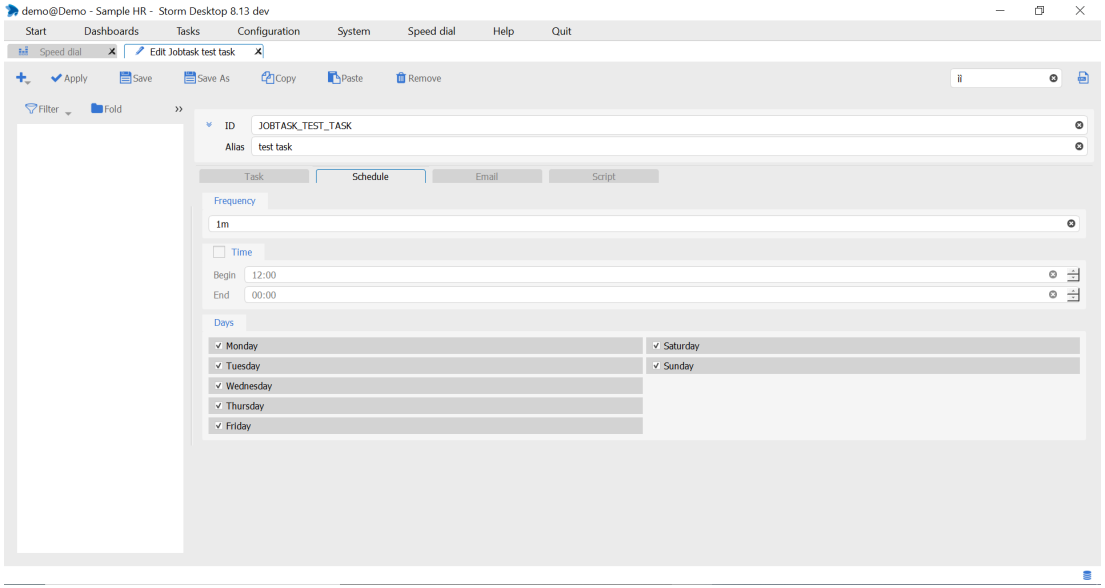



Figure 2.7:
**Scheduler:** Sets the schedule of the task
- Frequency: 30s, 5m, 1h etc.
- Time: if untick, the task will be launched on start
- Begin:
- End:
- Day selection:

Report sent by email will require a *mailsmtp* to be set up

### Tab Email

Third tab *Email* sets the mail destination and server

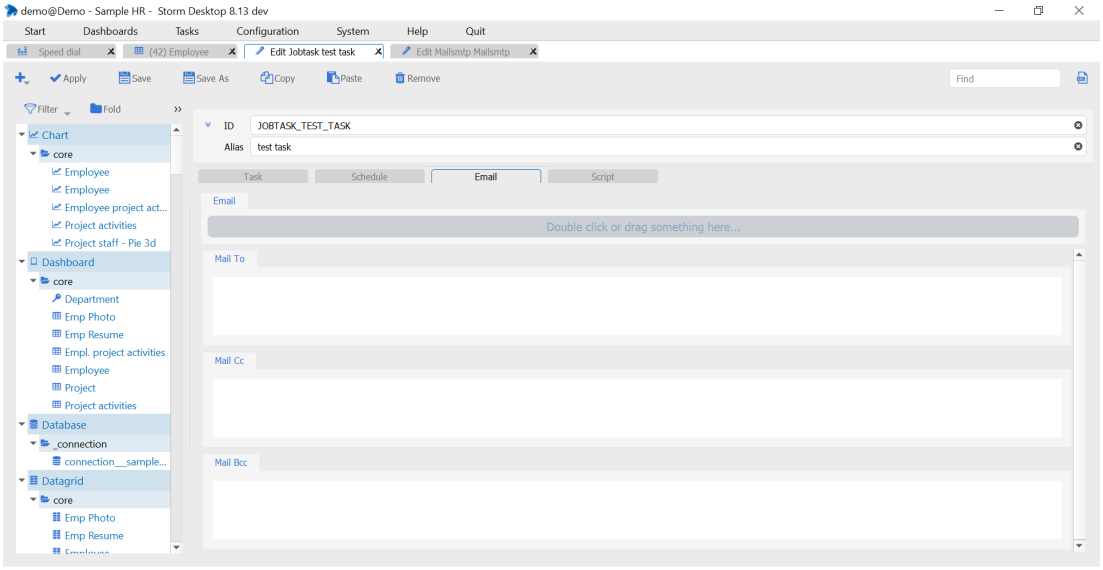



Figure 2.8:
- Email: drag and drop a mailsmtp from the left selector
- Mail To:
- Mail Cc:
- Mail Bcc:

Report sent by email will require a *mailsmtp* to be set up

### Tab Script

Here the user can define a script to be applied on the task. There is not a recommended selection of functions, it must be written from scratch by an expert user.

### 2.3.2 Chart

The editor is split in multiple tabs.

**Tabs:**

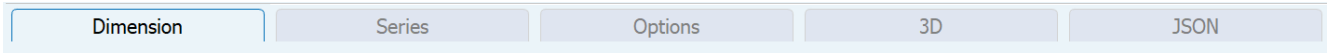



Figure 2.9:
• Dimension: set the x axis properties.
• Series: set the y and eventual further axises properties.
• Find bar: filters out all dashboard not maching the input string
• Options: miscellaneous chart options: title, legend, theme and alignment and others.
• 3D: 3d effect properties.
• JSON: accepts a fixed JSON representing a chart.

**Dimension tab:**

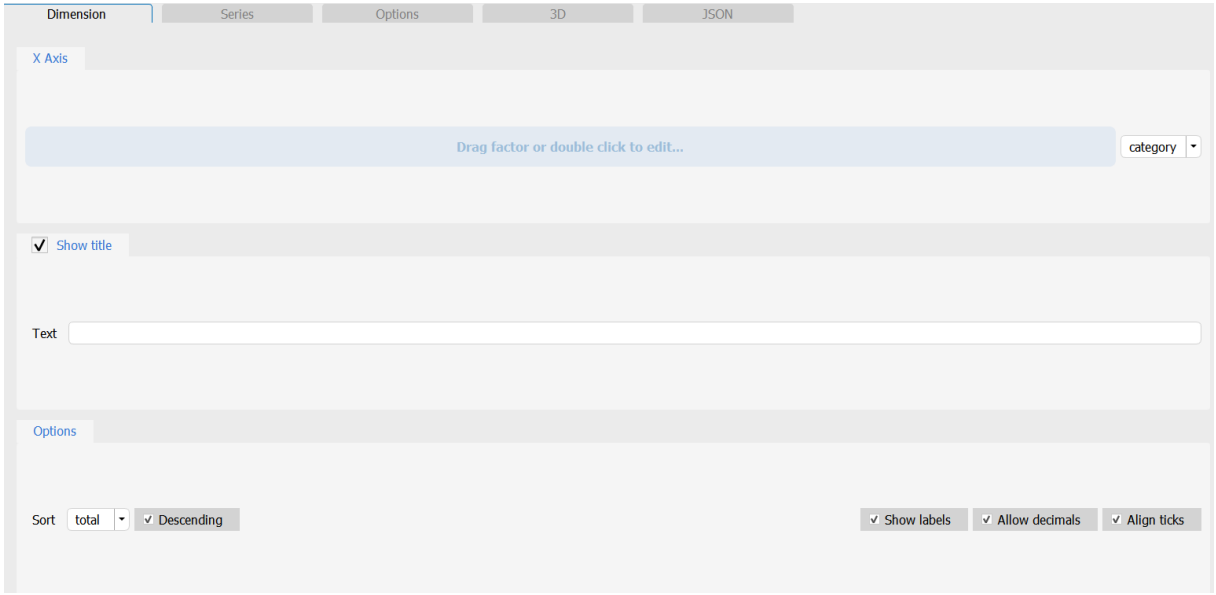



Figure 2.10:
**X Axis:** sets the column name(s) along which to distribute the chart's numeric values. The menu on the left is:
• category: each value will e interpreted as a category.
• datetime: each value will be interpreted as a datetime value and sorted accordingly.
**Show title:** Enable or disable the title.
• Text: Sets the title.
**Options:**
• Sorting: menu allows to choose between alpha, total or code.
• Descending: reverses the order of values on the axis.
• Show labels: displays the chart's labels.
• Allow decimals: shows decimal values on labels.
• Align ticks: when using multiple axis, the ticks of two or more opposite axes will automatically be aligned by adding ticks to the axis or axes with the least ticks.

**Series tab:**

Here the user defines how many charts are displayed, their type, the value axis and other properties.

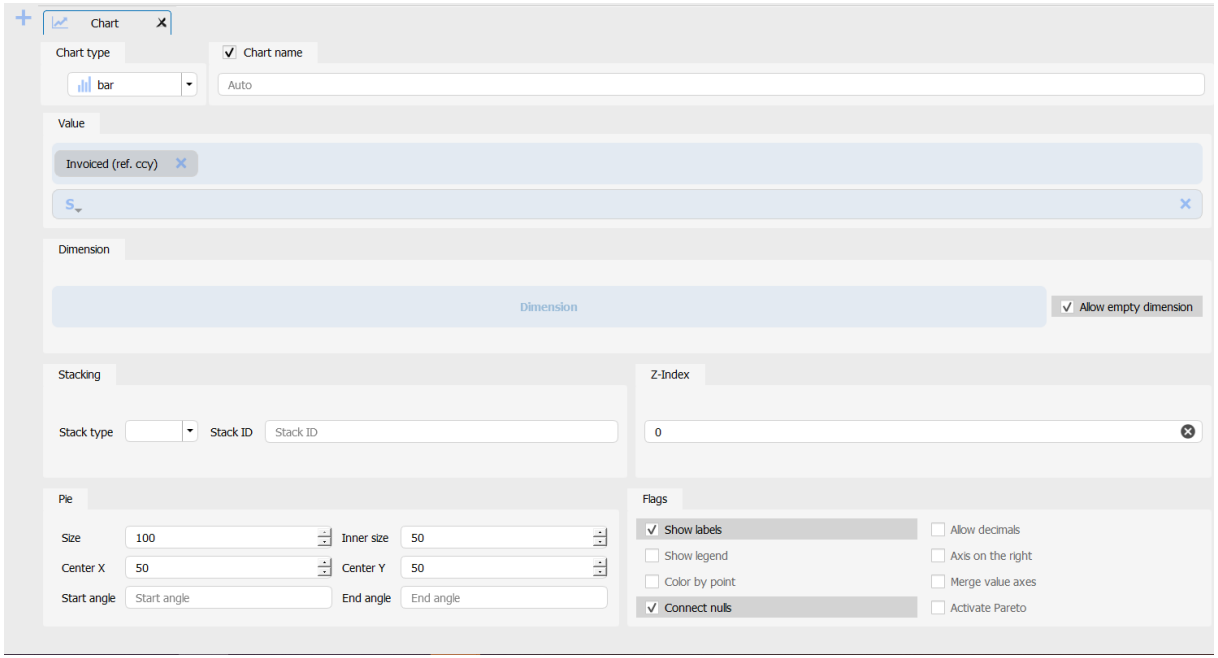



Figure 2.11:
**Chart type:** defines the chart type and an optional title.
**Value:** the value axis for the chart. Can be a column from the list on the left or an expression (**??** ).
• **f** (menu): selects a JavaScript function from a list.
**Dimension:** defines one or more dimensions for the chart. Can be a column from the list on the left or an expression (**??** ).
• Allow empty dimension: if checked, dimensions with false, null and zero values will be shown.
**Stacking:** choose the stack type. If none is selected, no stack is applied.
**Pie:** options for pie chart.
**z-index:** charts with higher z-index overlay charts with lower.
• Show labels: shows labels on graphs.
**Flags:**
• Allow decimals: shows decimal values on labels, same as in *dimension* tab.
• Show legend: shows/hides the legend.
• Color by point: colors items on the same point on the X-axis with the same color (including stacked columns).
• Merge value axes : When more than one value is selected, force them to use the same Y-axis.
• Axes on the right: Othewrise they're on the left.
• Connect null: Reccomended for line charts: when on a point on the x-axis there is no value, the chart istead of assuming the value is 0 in interpolates the 2 adiacent values.
• Activate Pareto: Reccomended for column charts sorted by descending total: draw the line of the Pareto sum and add a 0-100% axis.

**Options tab:**

Contains miscellaneous options for the chart, such as the title, the theme and personalized tooltips which are displayed when hovering the mouse over the graph.

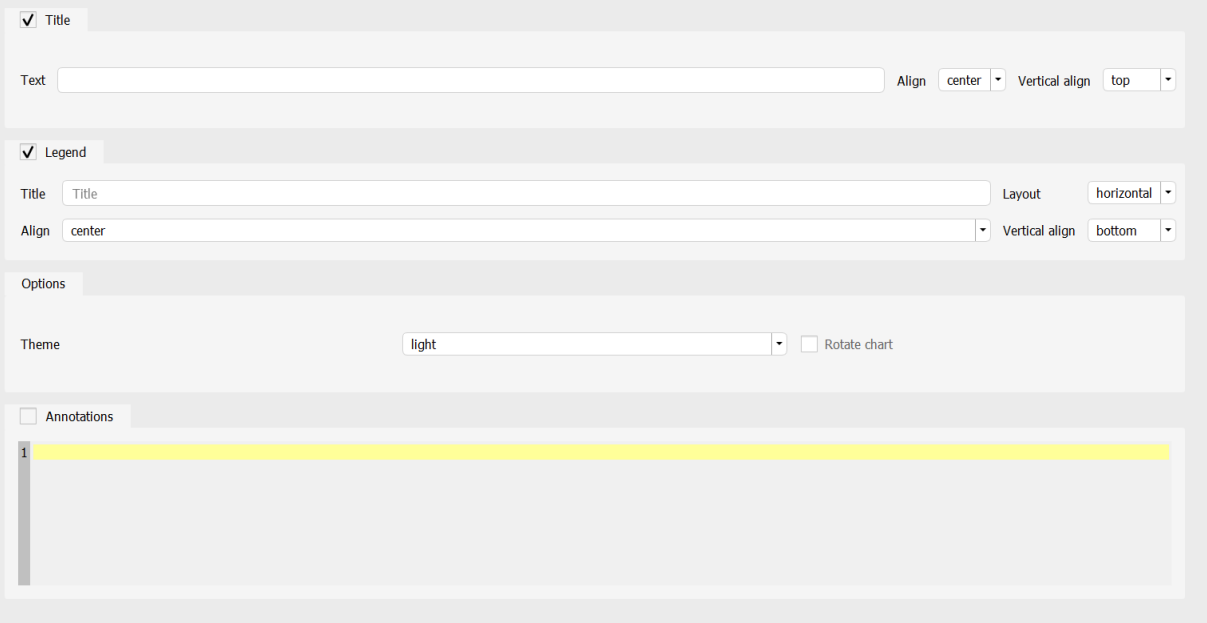



Figure 2.12:
**Title:** Enable, assign and align a title
**Legend:** Options relative to the chart's legend reside here. It is possible to give the legend a title, select the layout and the alignment.
**Theme:** Allows to change the theme between dark and light, and to rotate the chart.
**Annotations:** Enable, and assign an annotation that will appear on top of the chart.

**3D tab:**

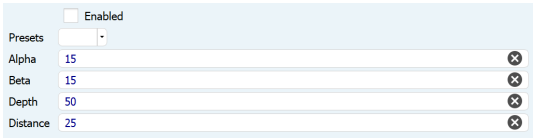



Figure 2.13

Contains options for the 3d effect of the graph.

**JSON tab:**

Figure 2.14

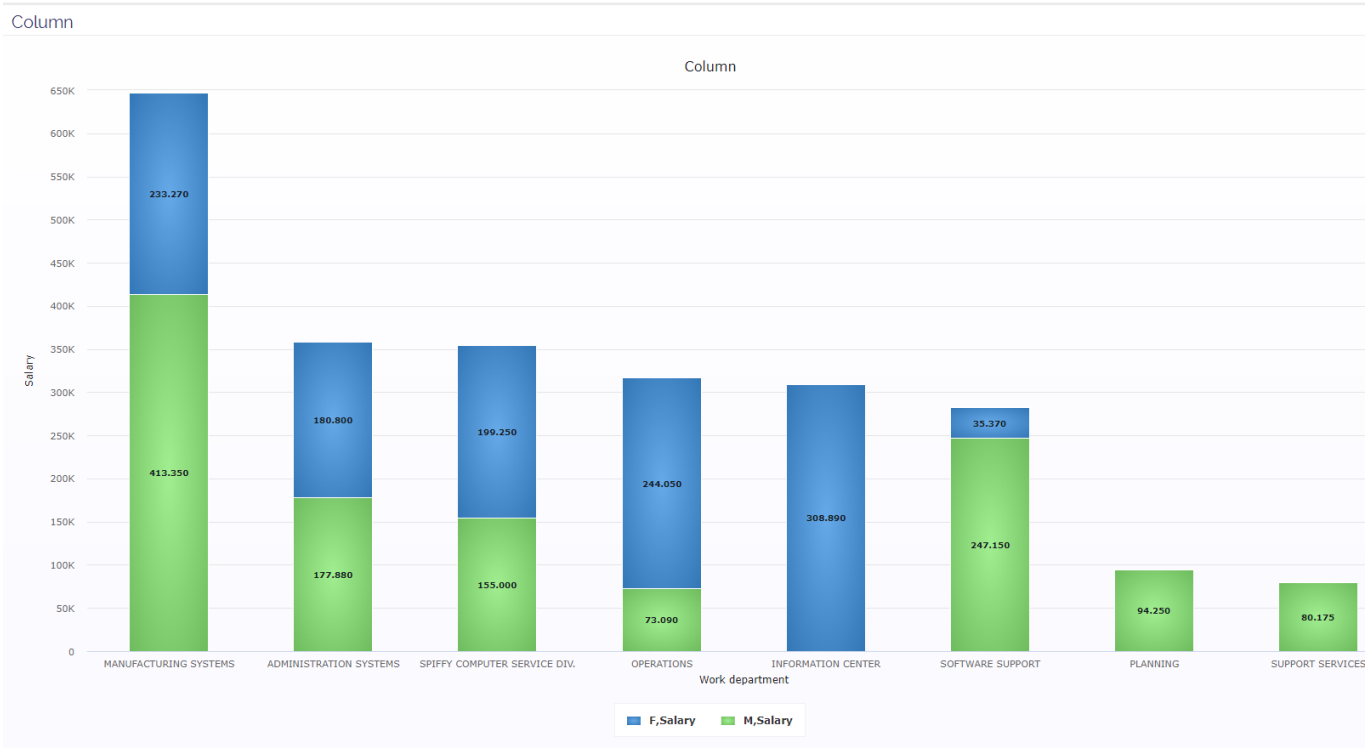
It is possible to paste here an existing JSON chart definition. For more details see section **??**

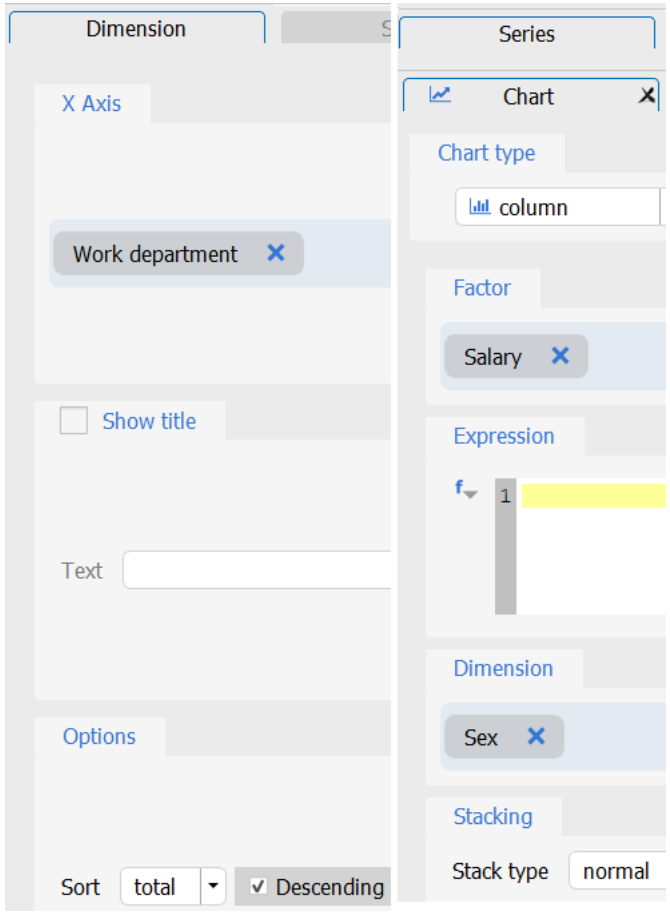
### 2.3.3 Chart types

In this section we show examples of several types of charts, selectable in chart editor–>series–>chart type.
Column
Bar
Funnel
Pyramid
Line
Scatter
Spline
Area
Areaspline
Waterfall
Packedbubble
Bubble

### Column

Column chart aggregates **Factor** grouped by **Dimension** and **x-axis**, then displays the data sorted on **x-axis** and colored by **Dimension**




Stacking is optional on Column charts

### Bar

Bar chart is identical to column, rotated by 90

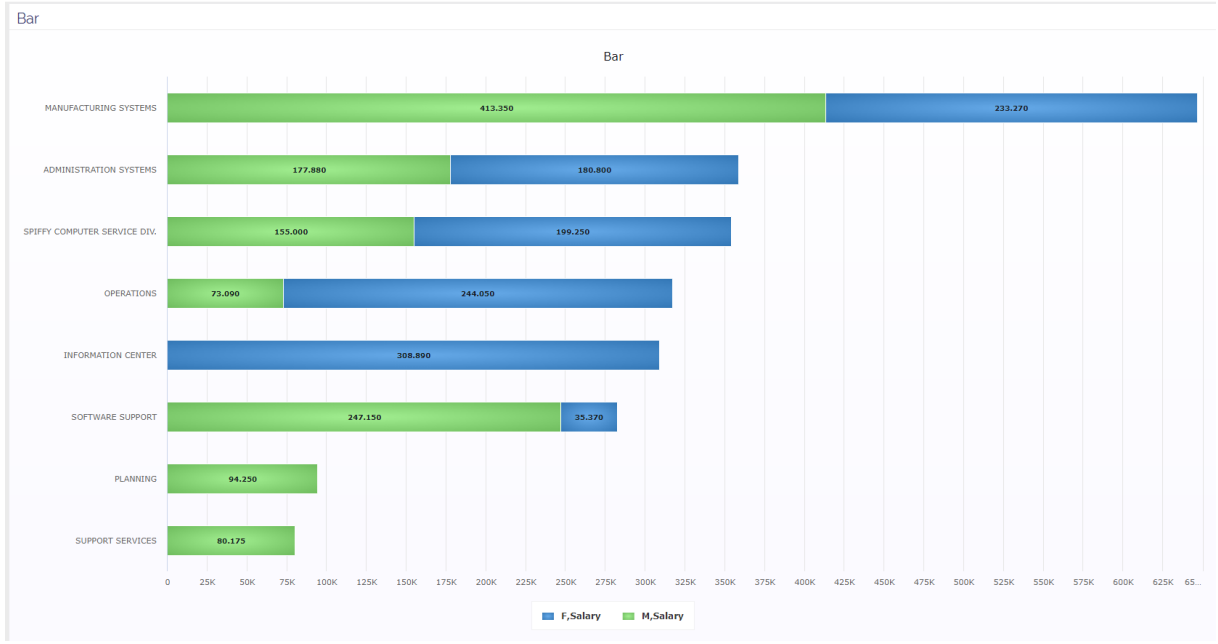
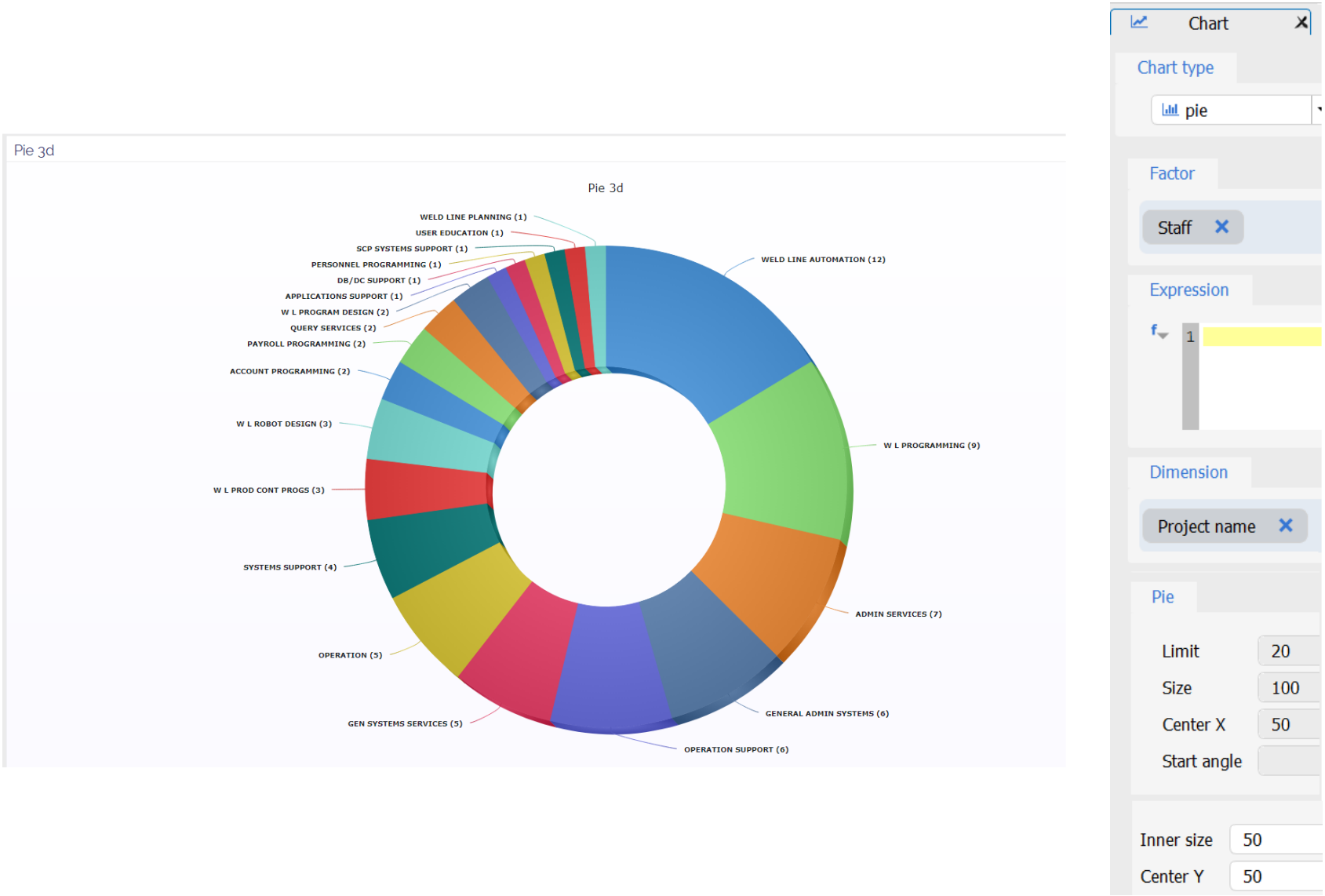



Figure 2.15

### Pie

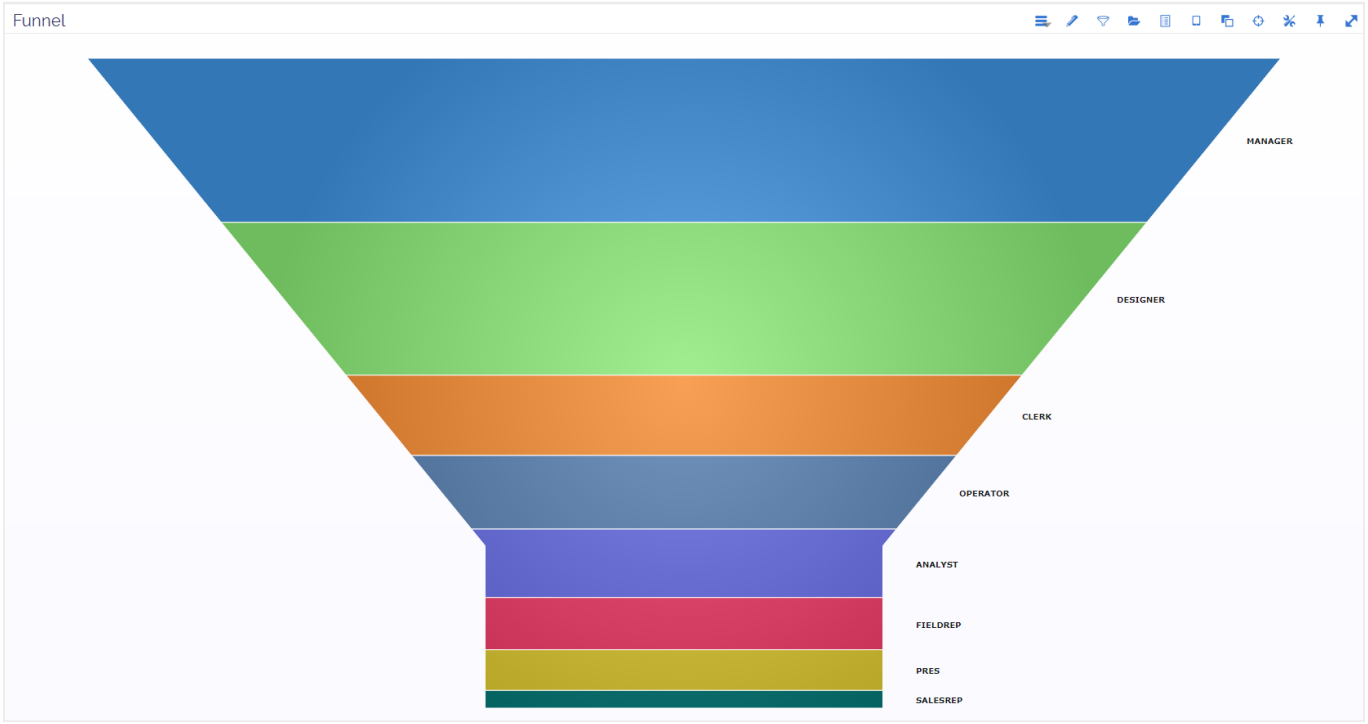
Pie chart aggregates **Factor** grouped by **Dimension** and displays the data in a classic pie chart

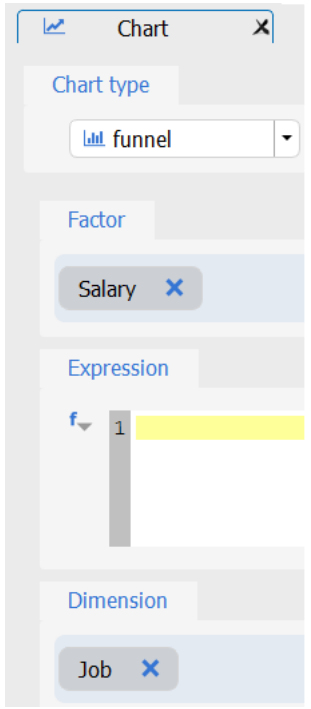



Pie charts contains several graphic options that allows to turn the pie into a donut

### Funnel

Funnel chart aggregates **Factor** grouped by **Dimension** and displays the data in decreasing sorting



### Pyramid

Pyramid is identical to funnel but with reversed order and a slightly different shape.

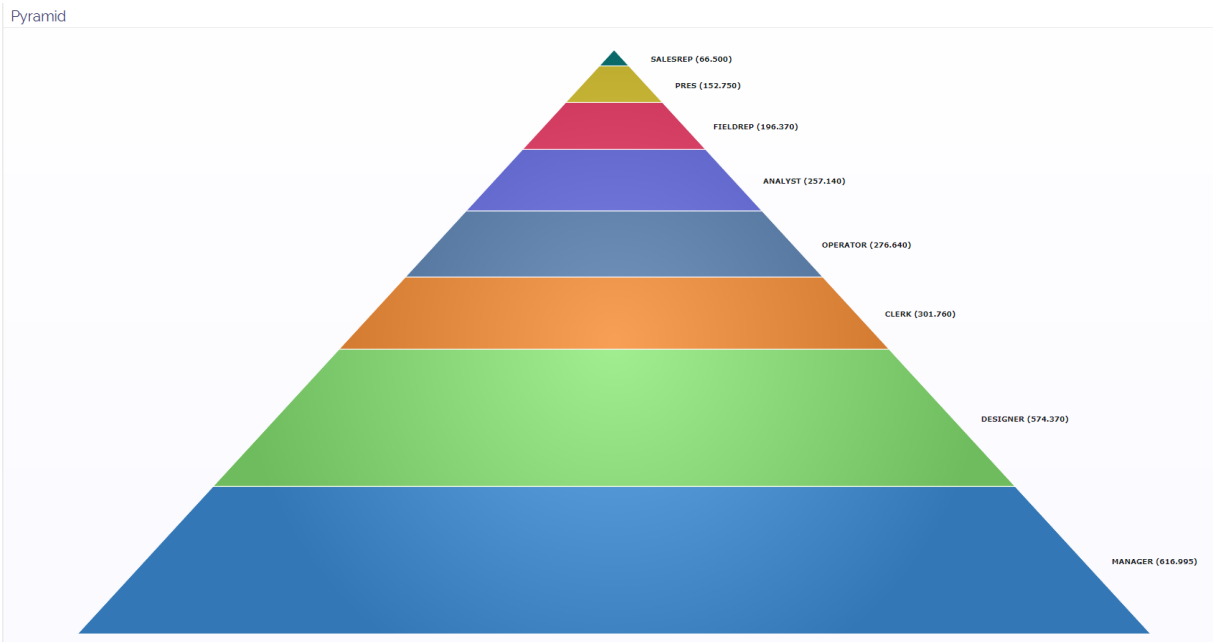
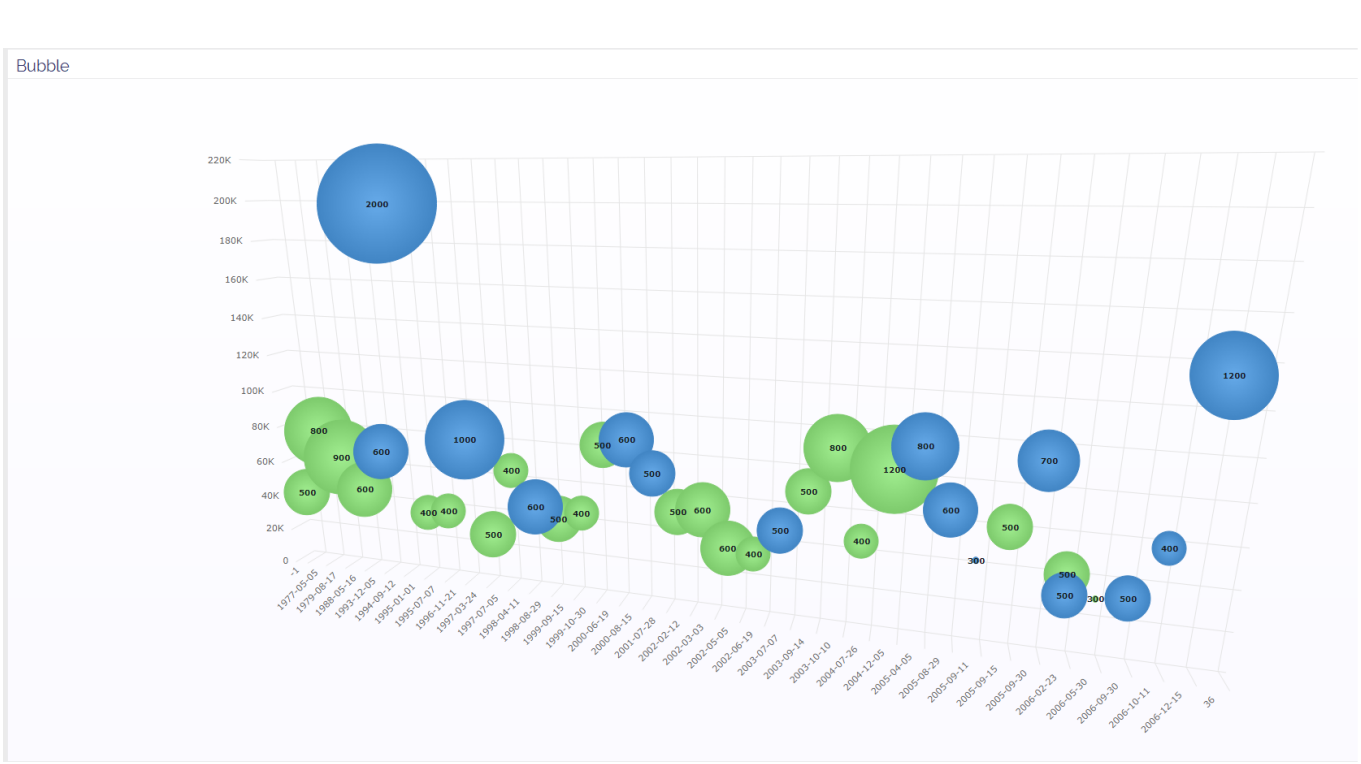
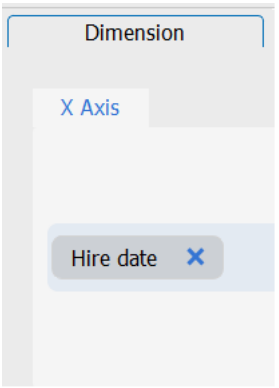


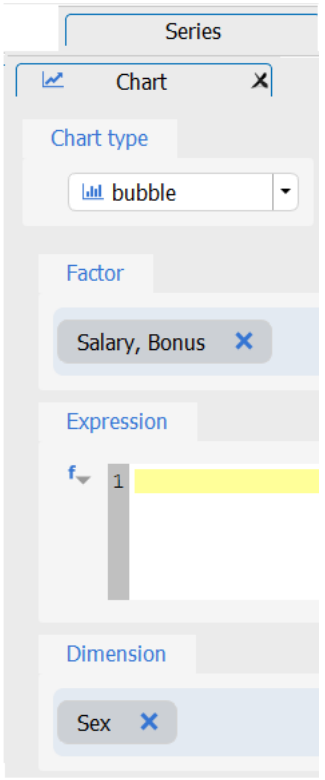

Figure 2.16

### Bubble

Bubble chart aggregates **Factor** grouped by **Dimension** and **x-axis**, then displays the data sorted by **x-axis** and colored by **Dimension**



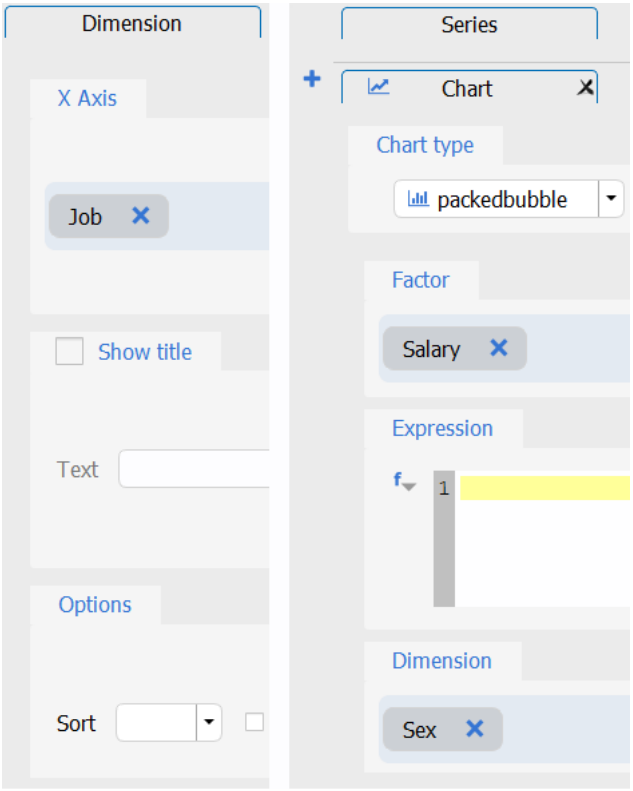




Two **Factors** are required, the first one will be represented by the vertical axis while the others will be displayed on the graph though the bubble size.

### Packedbubble

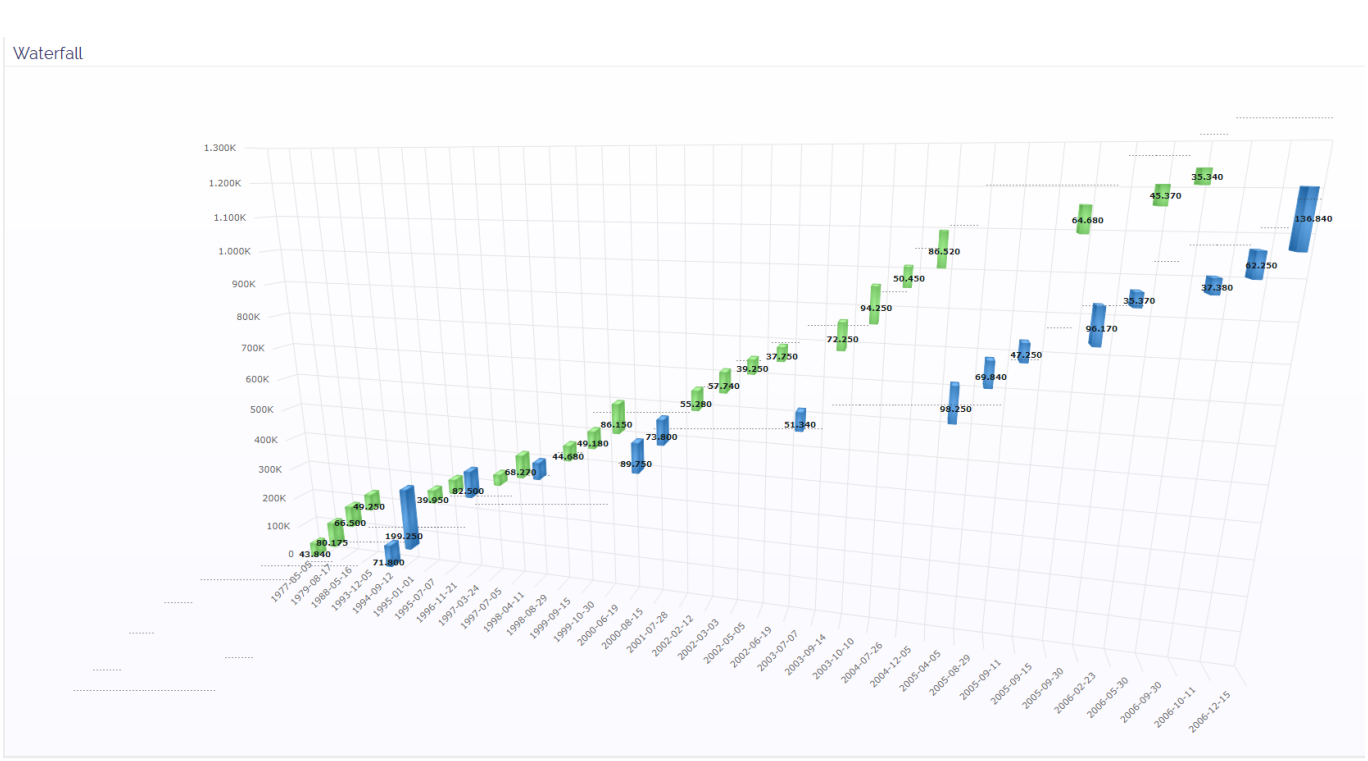
Packetbubble chart aggregates **Factor** grouped by **Dimension x-axis**

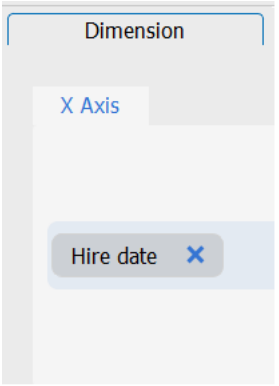


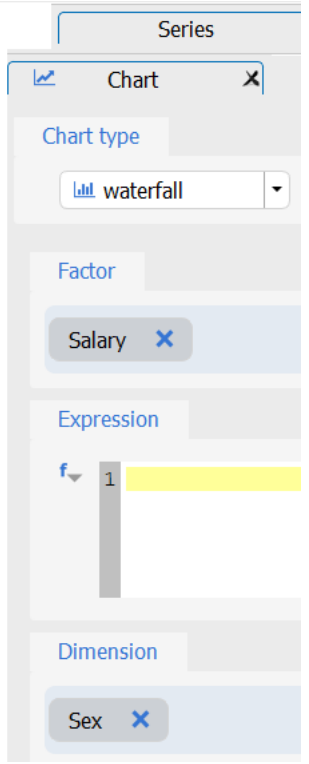

**x-axis** will be written on the bubble, **Dimension** determines the color and **Factor** the size of the bubbles

### Waterfall

Waterfall chart aggregates **Factor** grouped by **Dimension** and **x-axis**, then displays the data sorted by **x-axis** and colored by **Dimension**



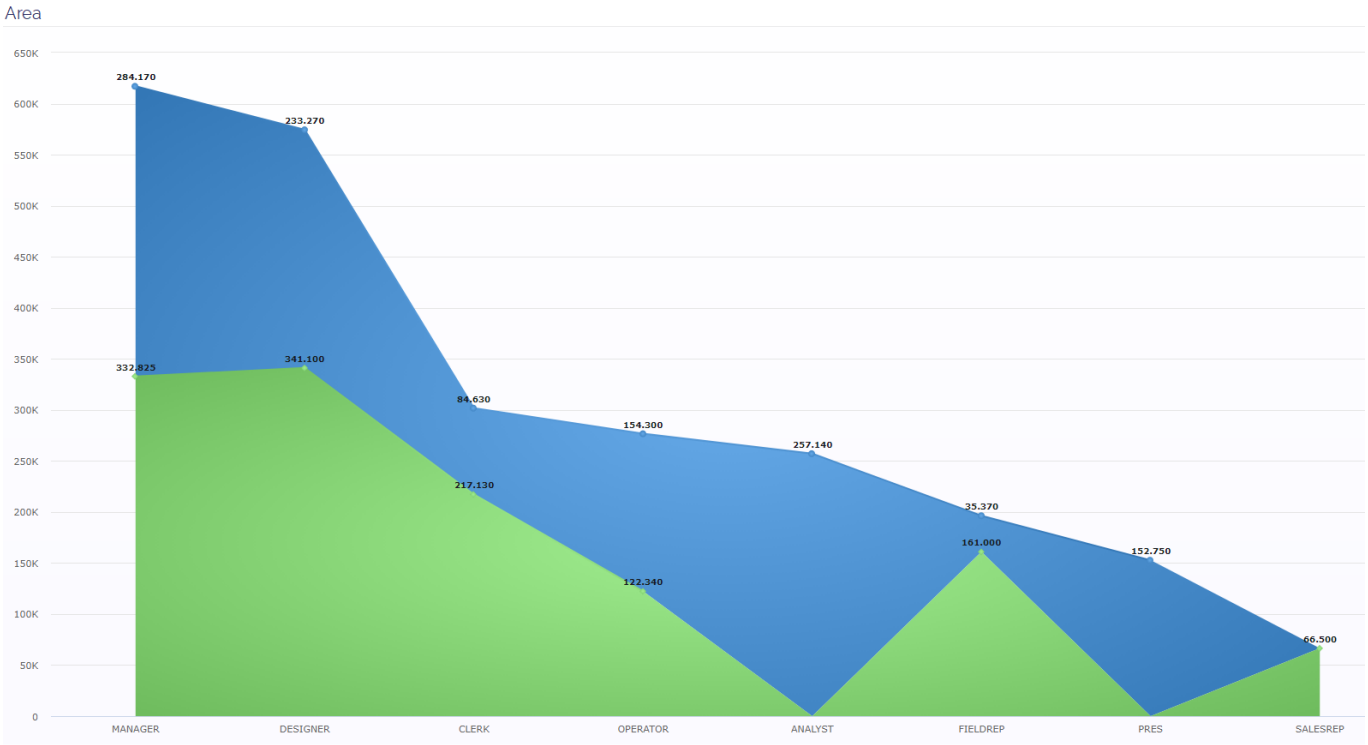



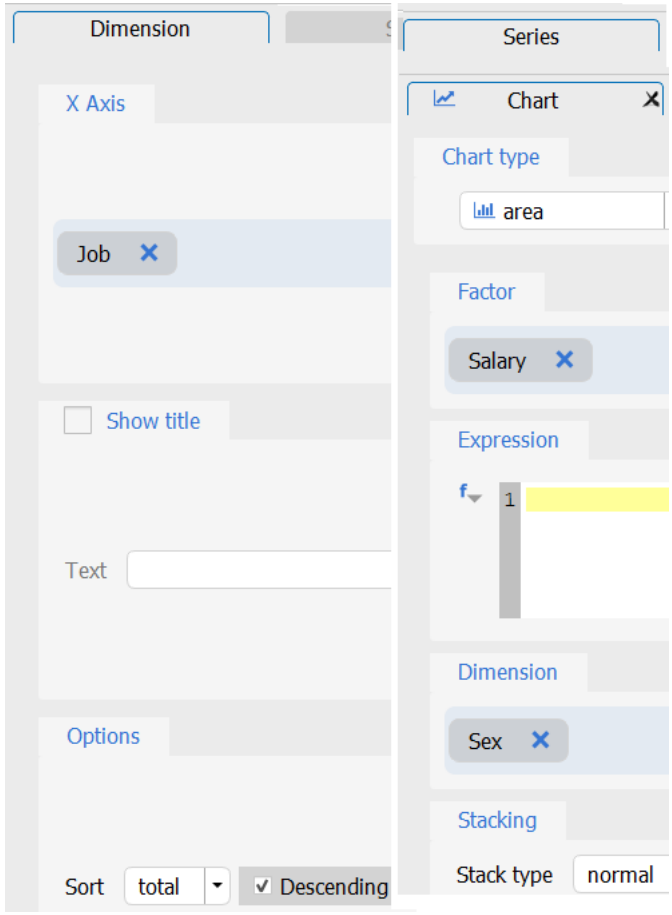




**Factor** determines the size of each column, while their position on the vertical axis is cumulative with all the data of the same **Dimension**

### Area

Area chart aggregates **Factor** grouped by **Dimension** and **x-axis**, then displays the data sorted on **x-axis** and colored by **Dimension**




Stacking is not mandatory but highly recommended on Area charts

### Areaspline

Areaspline is identical to area but with smooth curves.

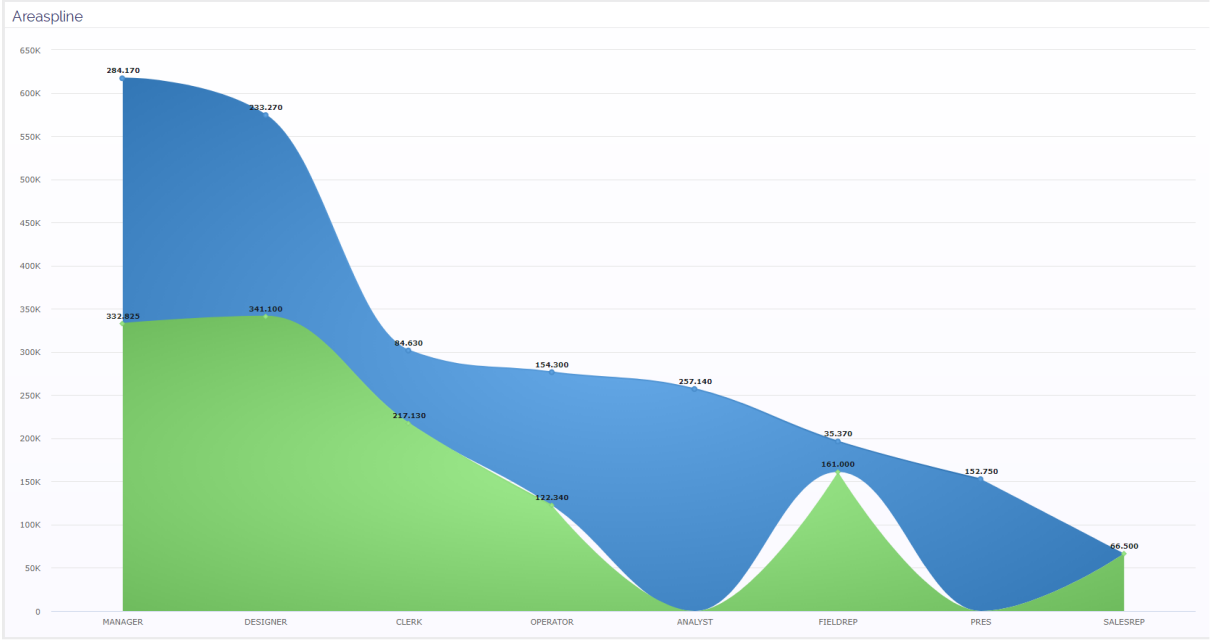
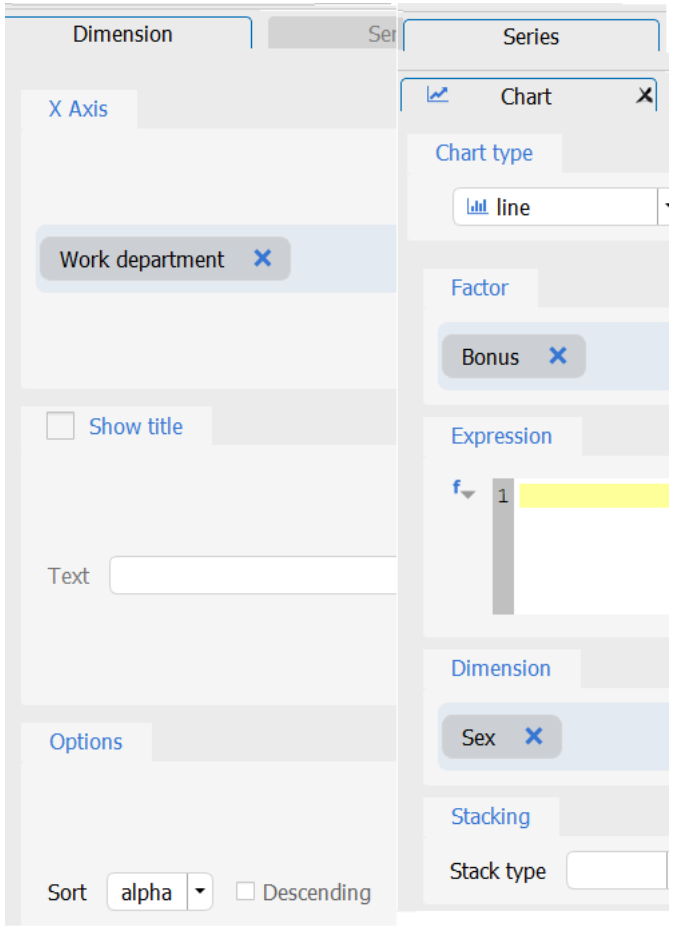



Figure 2.17

### Line

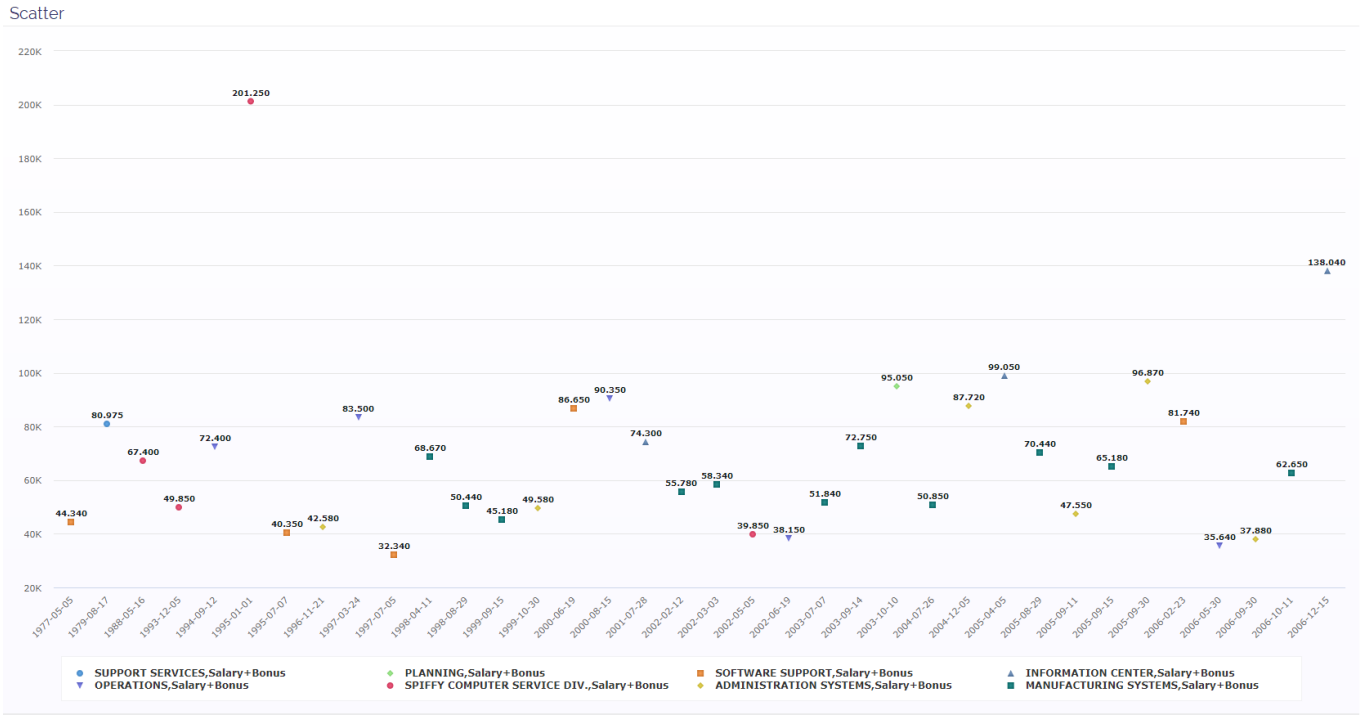
Line chart aggregates **Factor** grouped by **Dimension** and **x-axis**, then displays the data sorted on **x-axis** and colored by **Dimension**

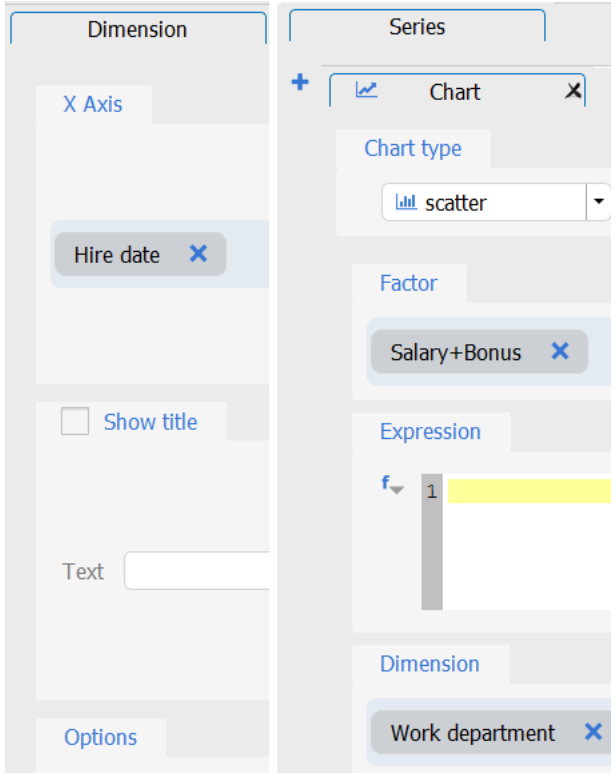



Line chart has obvious similarities and differences with Area chart, so stacking is possible but discouraged

### Scatter

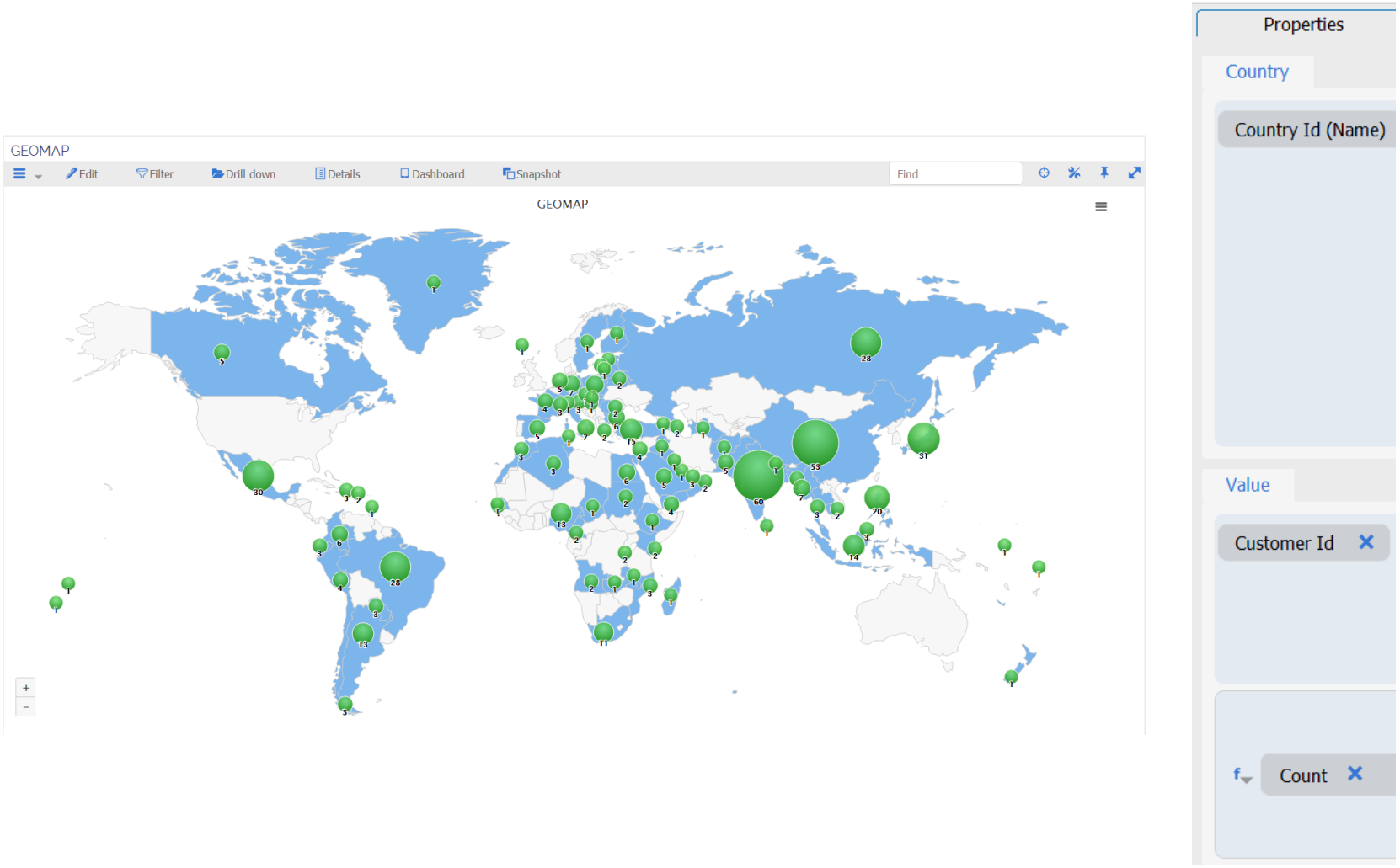
Scatter chart aggregates **Factor** grouped by **Dimension** and **x-axis**, then displays the data sorted on **x-axis** and colored by **Dimension**



### 2.3.4 Geomap

Geomap aggregates **Value** grouped by **Country** and **x-axis**, then displays the data on a world map




Line chart has obvious similarities and differences with Area chart, so stacking is possible but discouraged

Geomap has two tabs, Properties and JSON

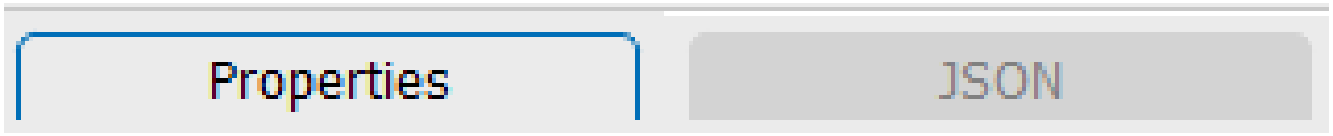



Figure 2.18

**Properties tab:**

Figure 2.19:
**Country:** defines column determining the country.
**Value:** defines the value. Higher vill be the value, darker the color of its cell.
**f (Menu):** sets the total function.

**JSON tab:**

It is possible to paste here an existing JSON geomap definition

### 2.3.5 Heatmap

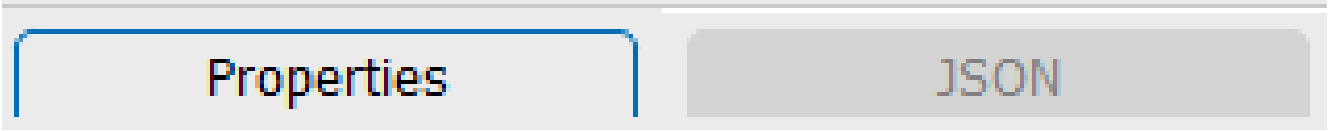



Figure 2.20

**Properties tab:**

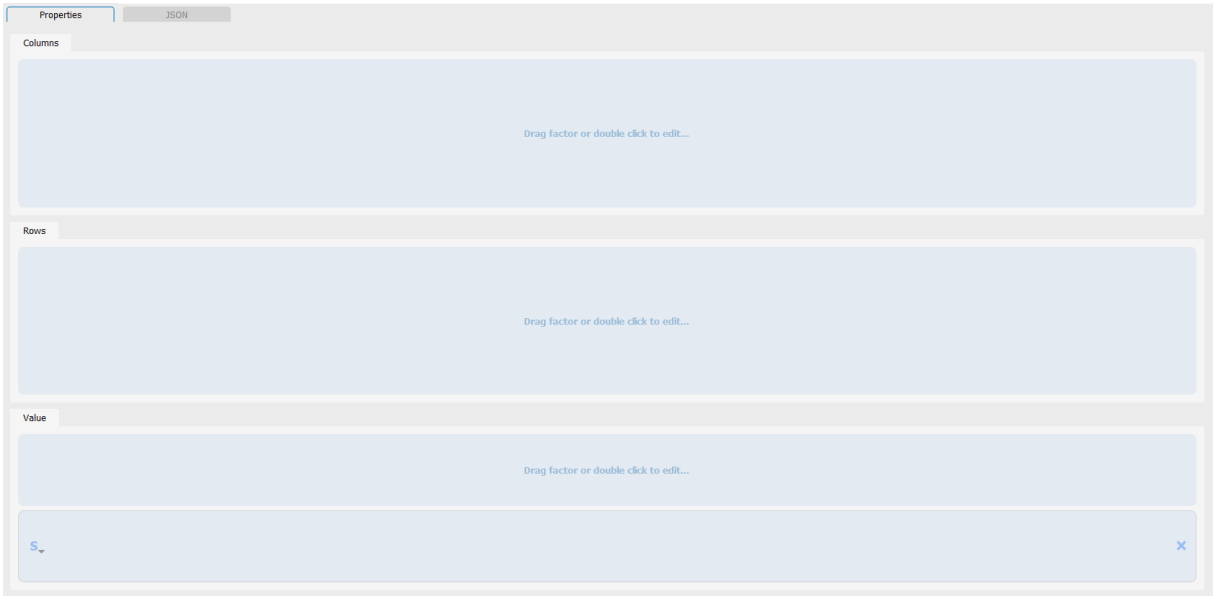



Figure 2.21:
**Column:** defines column dimension for the heatmap. Only one dimension is supported.
**Row:** defines row dimension for the heatmap. Only one dimension is supported.
**Value:** defines the value. Higher vill be the value, darker the color of its cell.
**Script:** sets the total function.

**JSON tab:**

It is possible to paste here an existing JSON heatmap definition

### 2.3.6 Html App

The editor is composed of two tabs: Html, where the desired html code is written, and Script, the usual tab accepting a JavaScript as input. The *Open file* button allows to browse for an existing html definition. Once loaded the path to this file will be displayed in the *Path* field. StormDesktop comes with a number of built in html definitions found in the `./storm.data/html` folder.

Html can display values from a dataset or from the result of the script. The html expects a JSON, that is the JavaScript must return a JSON object. A dataset also is read as a list of JSON objects. This list is accessible through the `$storm.context.table` interface. Values are then referenced by their name in the JSON encased in `${}`. In the case of a dataset, the JSON reference is the column name. Note that if the dataset contains multiple rows, a random one will be taken by the widget. See example code and result:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="fonts/css/font-awesome.min.css">
    <link href="style.css" rel="stylesheet">
    <script src="jquery-3.3.1.min.js"></script>
    <script src="script.js"></script>
  </head>
  <body>
       <div class="storm-widget-one">
         <div class="storm-widget-number">
           <span>${number}</span>
           <i class="fa fa-sort" aria-hidden="true"></i>
         </div>
         <div class="storm-widget-label">
           <span>${text} World</span>
         </div>
       </div>
  </body>
</html>
```

```
{
  'number' : 129,
  'text': 'Hello'
}
```

Result:

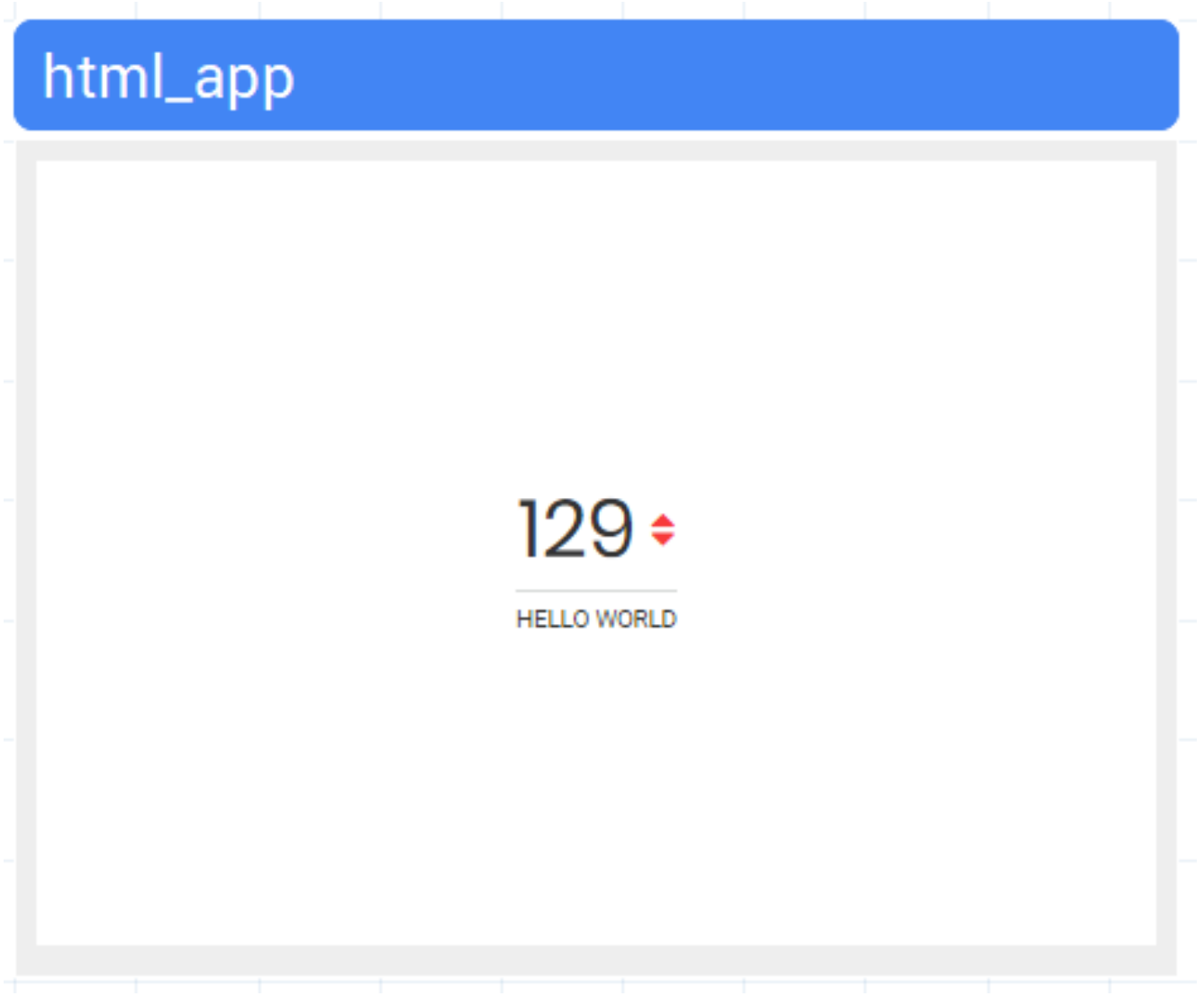



Figure 2.22

Example JavaScript taking values from the dataset:

```
let v = $storm.context.table;
console.warn(typeof(v));
// object
console.warn(JSON.stringify(v[0]));
// {"city#city":"A Corua (La Corua)",
// "city#city_id":1,
// "city#country_id":87,
// "city#last_update":"2006-02-15T03:45:25.000Z"}
v[0];
```

Thus values from this dataset can be referenced using for example `${city#city}`.

### 2.3.7 Image

Load an image from disk.

**File:** select a file from disk.

### 2.3.8 Kpi

The Key Performance Indicator calculates one aggregate value and displays it with many graphic options Consists of two tabs: *Value*, concerning the data, and *Padding*, concerning the appearence:

#### Value tab

Figure 2.23:
**Value:** defines the value.
**Expression:** sets the total function. If empty it uses the total set in column.

#### Padding tab

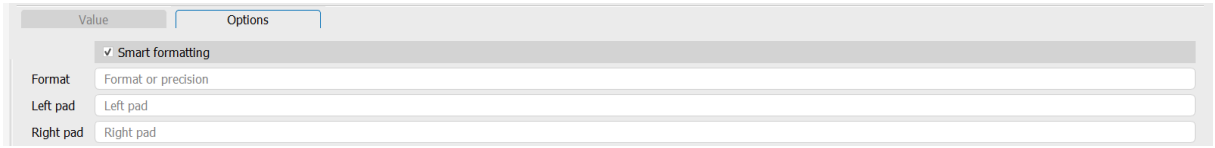



Figure 2.24:
**Smart formatting:** Let the engine deal with the formatting.
**Format:** sets the number of decimals shown.
**Left pad:** defines the distance from the left side.
**Right pad:** defines he distance from the right side

### 2.3.9 List

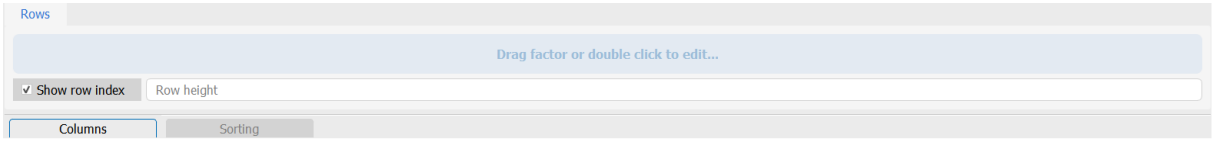



Figure 2.25:
**Rows:** set desired factor as row header, replacing the raw index.
**Show row index:** display a row count.
**Row height:** Sets row height: 0 sets to default.
**Columns:** 2.26 allows to build a set of columnset by dragging and droping from the factor selector on the left.
**Sorting:** 2.27 allows to define the sortings by dragging and dropping from the factor selector on the left..

See below the details of the two tabs

**Columns tab:**

Allows to build a set of columnset by dragging and droping from the factor selector on the left.

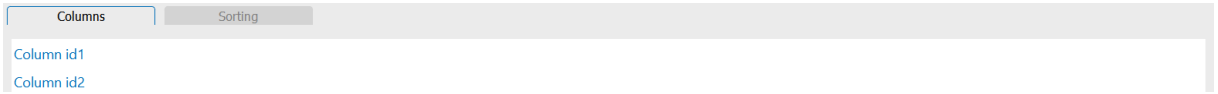



Figure 2.26:
**Expression:** allows to define and expression.
**Enable columnset:** activate panel below.
**Columnset:** Allows to build a columnset by drag and drop from the factor selector on the left or from the expression environment on top.

**Sorting tab:**

Allows to define the sortings by dragging and dropping from the factor selector on the left.

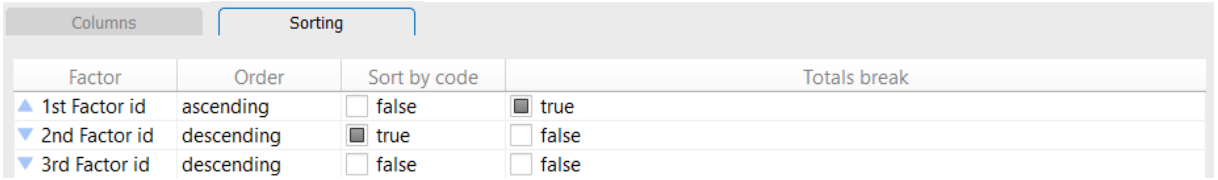



Figure 2.27:
**Factor:** drag and drop the factors from the selector on the left. When records have the same value for the upper factor, lower factors define the sorting among them.
**Order:** ascending or descending.
**Sort by code:** sort according the the code instead of the display.
**Totals break:** Caculate or not the subtotal for the subgroup defined by that factor.

### 2.3.10 Pivot

The editor is split in the following tabs

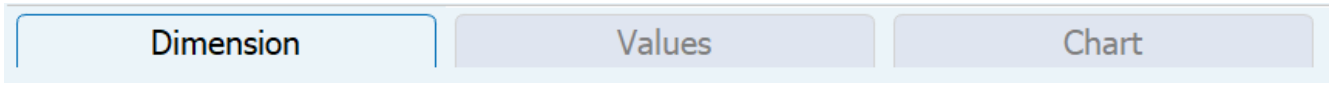



Figure 2.28

**Dimension:** select the pivot dimensions.

**Values:** select the matrix value(s) column.

**Chart:** set the options for the pivot chart.

**Dimension tab:**

This tab sets the two dimensions (rows and column) of the pivot:

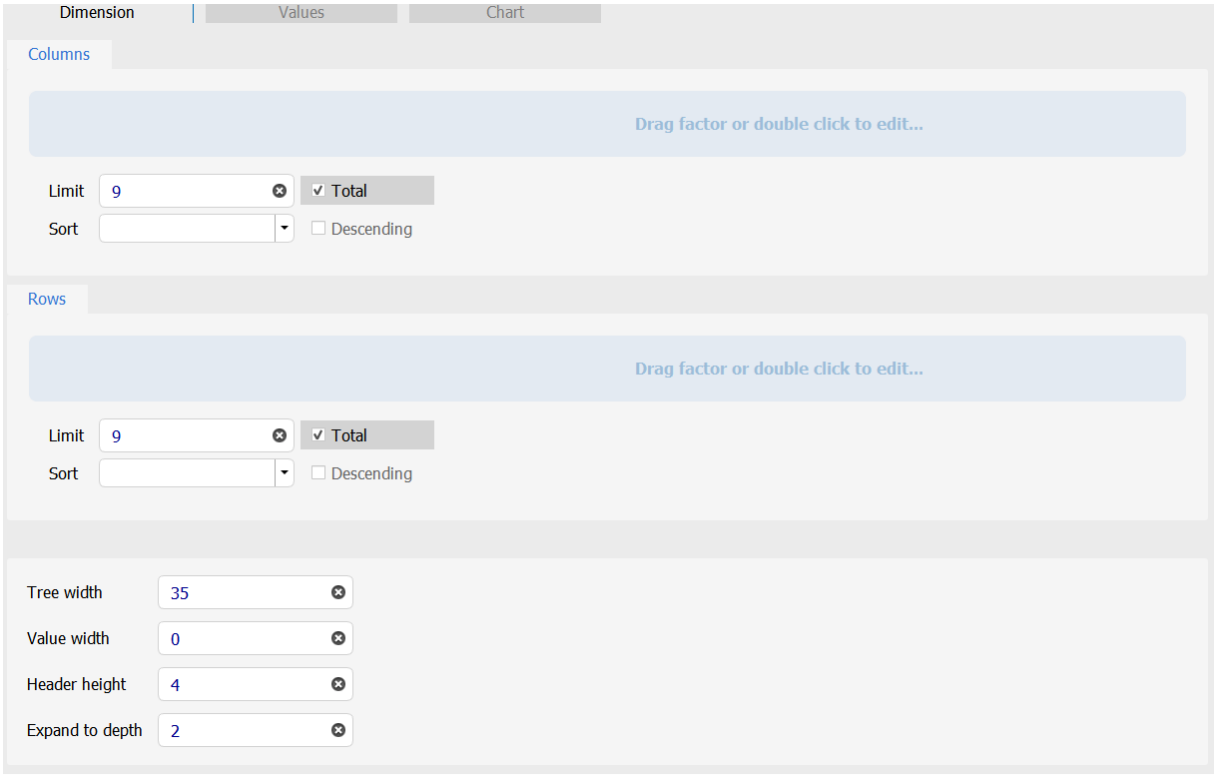



Figure 2.29:
**Columns:** Factors defining the columns of the pivot.
• *Total* checkbox controls whether totals are calculated.
• *Sort* menu defines sorting (total, absolute value, subtotals, code or alphabetical).
• *Descending* revers the order.
• *Limit* If more columns are present, then only this number of columns will be shown while the rest is collapsed into a *other* total.
**Rows:** Factors defining the rowss of the pivot. Multiple factors will create additional level
• *Total* checkbox controls whether totals are calculated.
• *Sort* menu defines sorting (total, absolute value, subtotals, code or alphabetical).
• *Descending* revers the order.
• *Limit* If more columns are present, then only this number of columns will be shown while the rest is collapsed into a *other* total.
**Options:**
• *Tree width* sets the width of the pivot.
• *Value width* sets the width of each cell.
• *Header height* sets the heigh of the headers.
• *Expand to depth* sets how many level are opened by default.

**Values tab:**

This tab consists of any number of Value subtabs, all indepentent from each other:

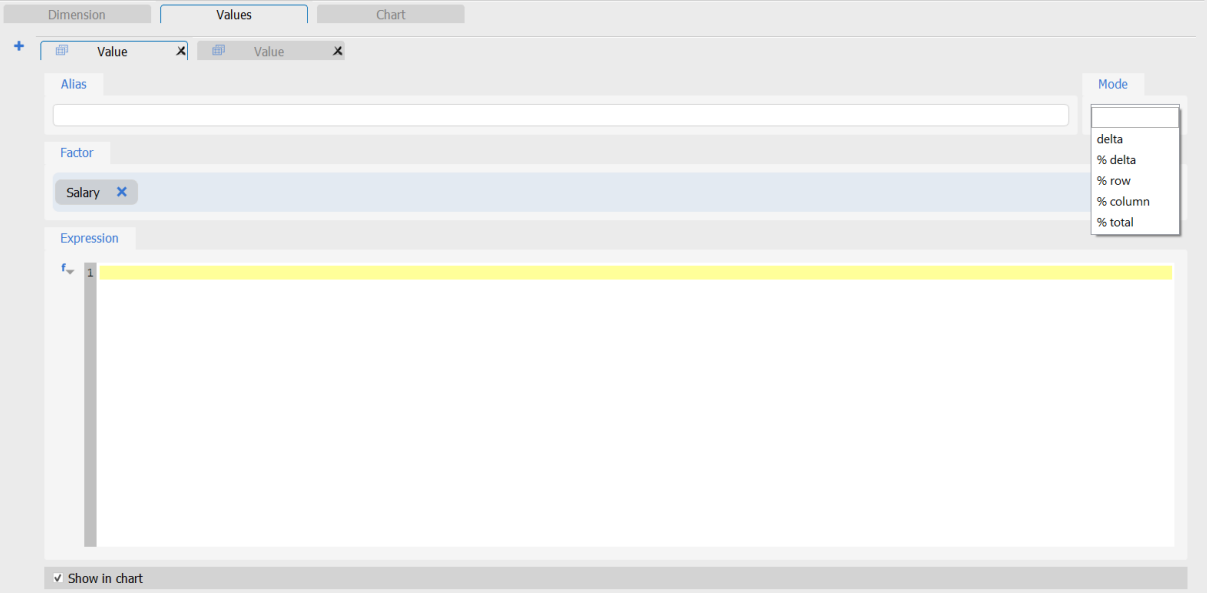



Figure 2.30:
**Factor:** select the value (column or expression).
**Mode:** choose the total type. If none is selected, standard aggregation is used.
**Expression** (menu): selects a JavaScript function from the list or allow the user to write its own script.
**Show in chart:** Show this value in the pivot's graph.

**Chart tab:**

This tab sets general graph options:

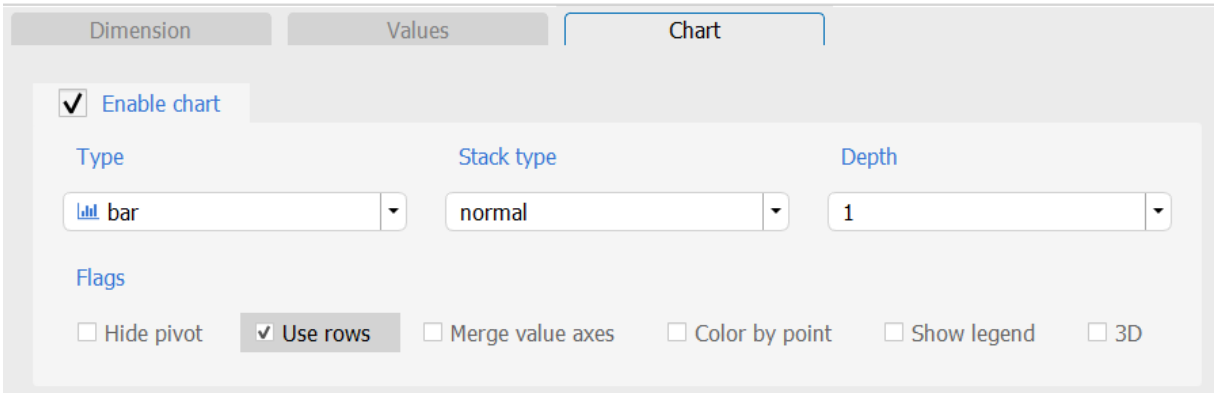



Figure 2.31:
**Enable chart (checkbox):** enables the chart.
• Type (menu): selects the chart type.
• Stack type (menu): select between disabling the stack, enabling simple stack or normalised stack.
• Depth (menu): select how many levels of rows shall be displayed in the chart.
• Hide pivot (checkbox): don't display the pivot chart.
• Use rows (checkbox): uses rows (instead of columns) as first dimension of the chart.
• Merge value axes (checkbox): When more than one value is selected, force them to use the same Y-axis.
• Color by point (checkbox): Colors items on the same point on the X-axis with the same color (including stacked columns).
• Show legend (checkbox): enables the legend
• 3d (checkbox): enables the 3d effect. Applies to column, bar and pie charts.

### 2.3.11 Pivotlist

The editor consist in one main tab and two subtabs:

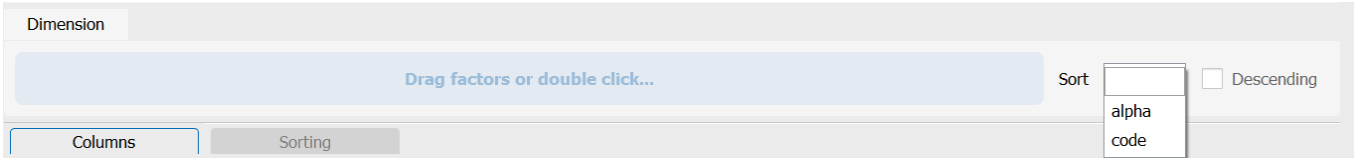



Figure 2.32:
•Dimension: select the pivot dimensions.
•Sort: select the type of sorting.
•Descending: reverse the sorting.

**Columns tab:**

The editor consist in one main tab and two subtabs:

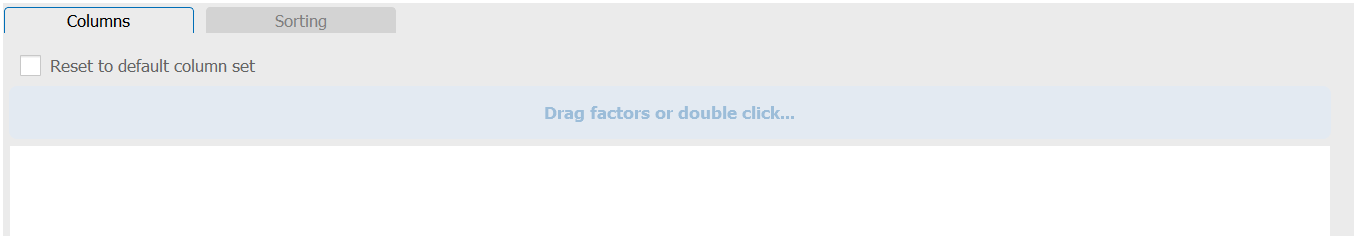



Figure 2.33:
•Columns Selects one or more columns to be used as values for the pivot.

Accepted values are the columns from the left menu and/or expressions (**??**).
•Reset to default columnset: Allows to reset to the columnset defined in the appropriate tab of dataset (see **??**).

**Sorting tab:**

The editor consist in one main tab and two subtabs:

Figure 2.34:
•Columns Selects one or more columns to be used as values for the pivot.

Accepted values are the columns from the left menu and/or expressions (**??**).
•Reset to default columnset: Allows to reset to the columnset defined in the appropriate tab of dataset (see **??**).

### 2.3.12 Style

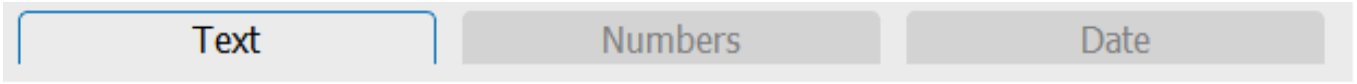



Figure 2.35:
- Text: text style options.
- Numbers: number formatting options.
- Date: date formatting options.

**Text tab:**

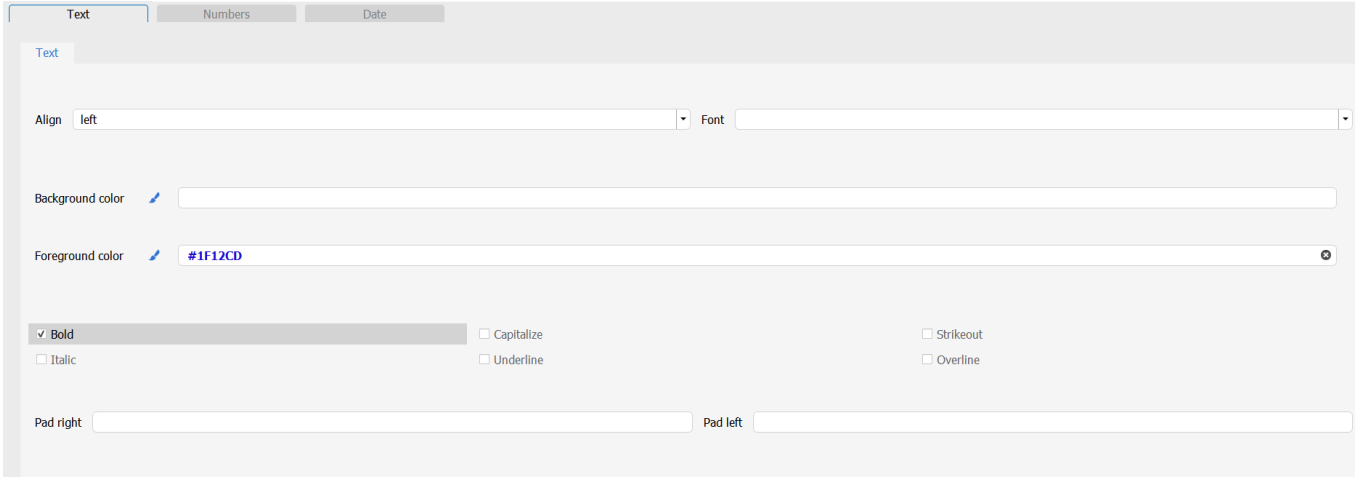



Figure 2.36:
- Font: selects font.
- Align: selects alignment.
- Background color: background color.
- Foreground color: foreground color.
- Padding: sets right and left padding
- Text decoration: standard text style options.

**Numbers tab:**

Figure 2.37:
• Format number: add dots and commas to numbers. Precision is the number of decimal digits.
• Zero is visible: shows or hide zero values.
• Precision: Sets is the number of decimal digits.
• Positive numbers color: sets color for positive values.
• Negative numbers color: sets color for negative values.

**Date tab:**

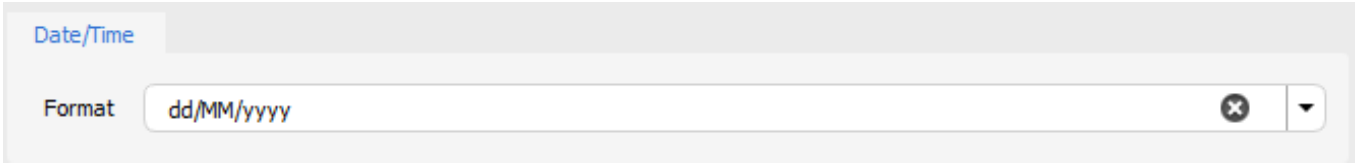
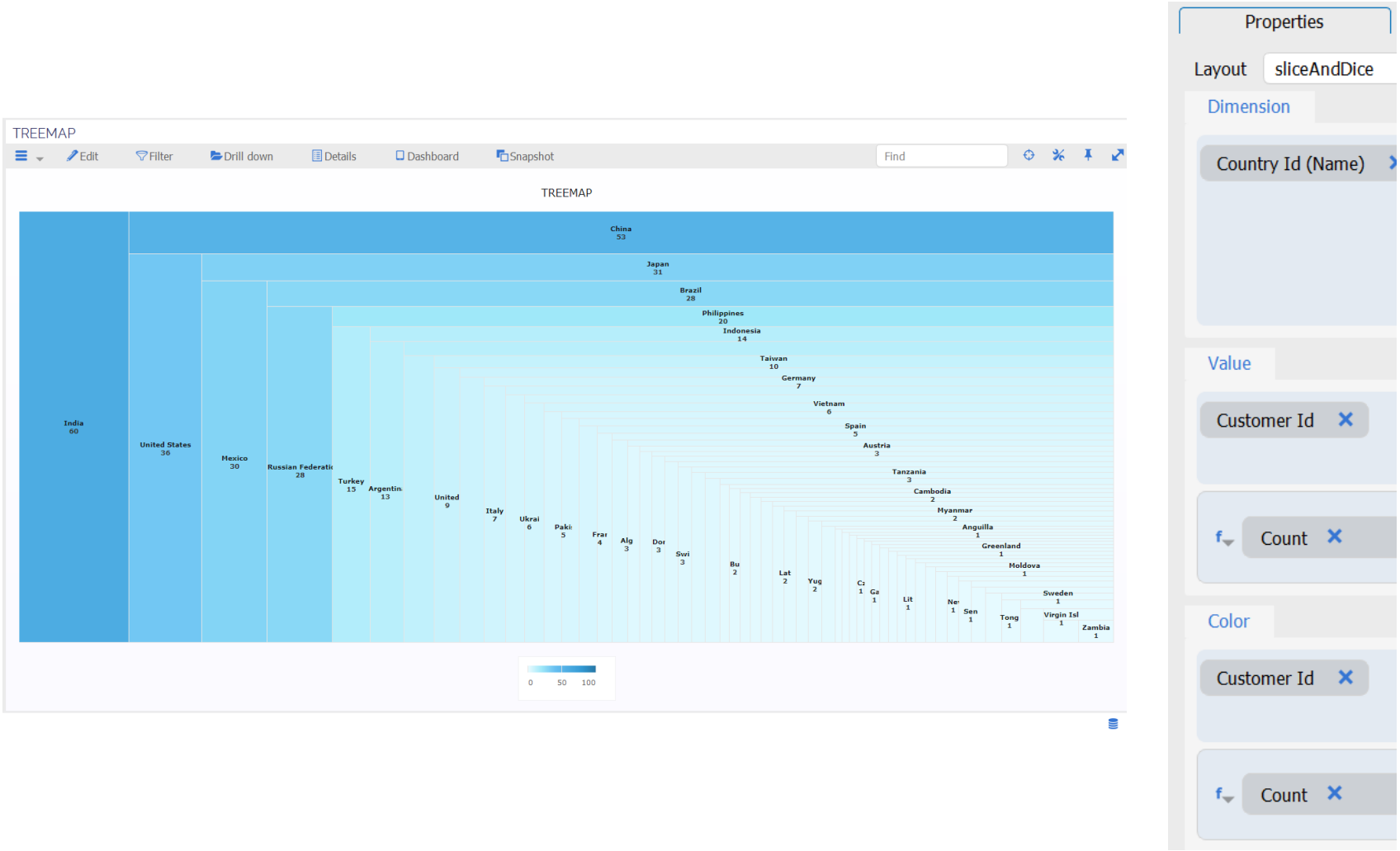



Figure 2.38:
• Format : Sets the date format. If blank, default format is determined by User Locale.

### 2.3.13 Textapp

### 2.3.14 Treemap

Treemap aggregates **Value** and **Color** grouped by **Dimension**




Colors will determine the tonality of the cells while Value will be written in them

Treemap editor consists of two tabs:

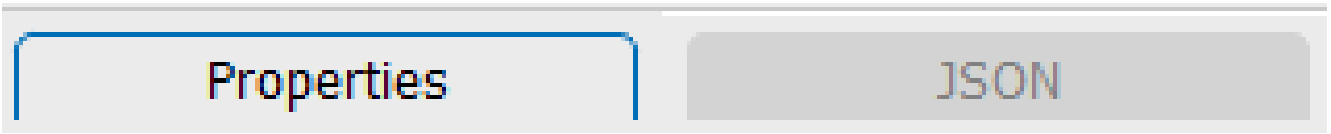



Figure 2.39

**Properties tab:**

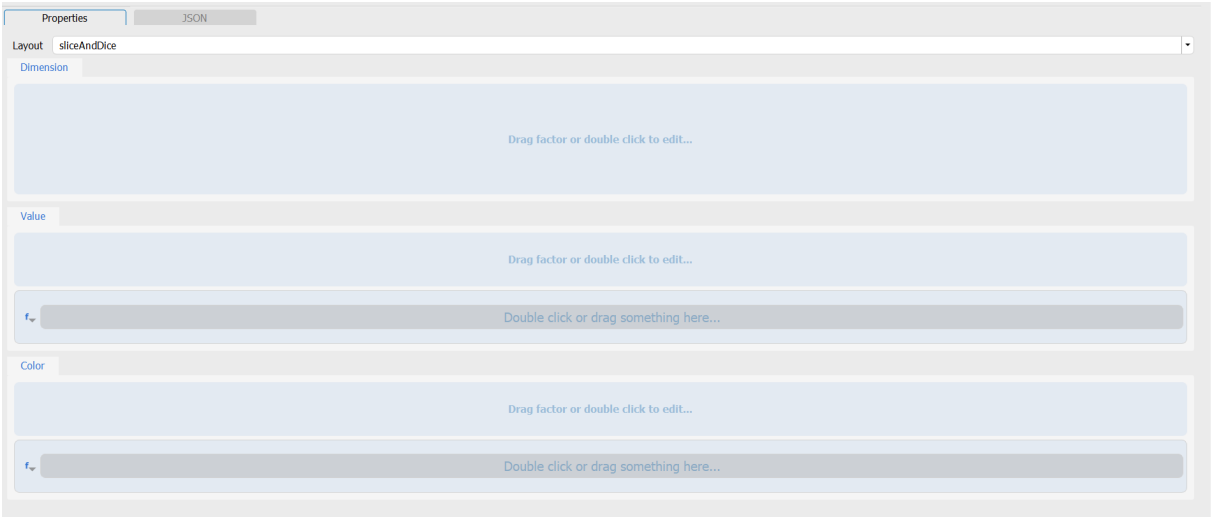



Figure 2.40:
**Layout (menu):** choose among several shapes of treemap.
**Dimension:** defines column dimension for the heatmap. Only one dimension is supported.
**Value:** defines the main value. It will be written inside the cell
f: select a JavaScript function from the list or write one.
**Color:** defines the secondary value. Higher vill be the value, darker the color of its cell.
f: select a JavaScript function from the list or write one.

**JSON tab:**

It is possible to paste here an existing JSON treemap definition

### 2.3.15 Webpage

### 2.3.16 Wordcloud

**Value:** defines the value.

**Dimension:** defines one or more dimensions for the worldcloud. More dimension will appear as *dimX/dimY*

**Script:** sets the total function.