A new Stratebi white paper

www.stratebi.com

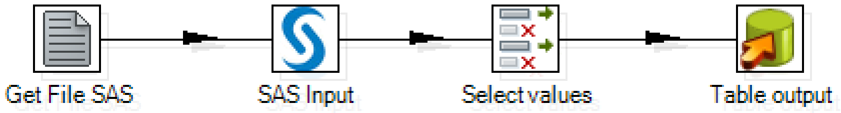
Aug 2013

Pentaho & SAS: Getting data from SAS and exploit it into Pentaho

In this post we try to unveil the capabilities of the new Pentaho Data Integration SAS Input step. This new feature was included in the latest stable version of PDI (4.4) and is very useful for those corporations which use SAS as corporative Business Analytics tool and want to exploit the information into Pentaho BI Suite.

This new add-on is an evidence of Pentaho strategy focused on expanding their products and tools with new capabilities

Our main goal in this document is reading a SAS file using PDI. In a second stage we will make use of the information read using AgileBI plug-in included in PDI.

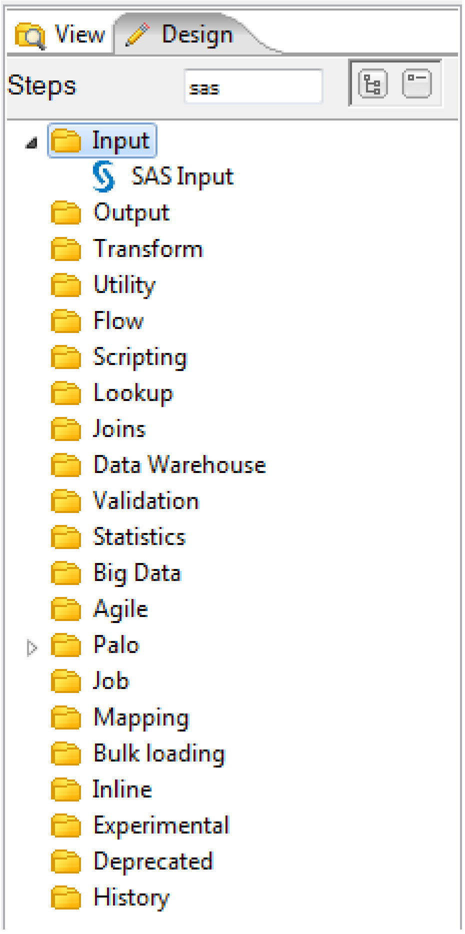


We are going to use as sample data the results of *"Estimating and modelling relative survival using SAS"* research carried out by Paul Dickman on 2004. Dickman's study includes Finnish patients diagnosed with colon carcinoma between 1975 and 1994. Here is the full document for detailed reference http://biostat3.net/download/sas/readme.pdf

Sample data contains:

- Age at diagnosis
- Date of diagnosis
- Date of exit
- Month of diagnosis
- Sex
- Clinical stage at diagnosis
- Anatomical subsite of tumour
- Survival time in completed months
- Survival time in completed years

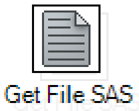
Due to the fact that we are working at data level this new feature is available a transformation step. Below is a screenshot with the new SAS Input step.

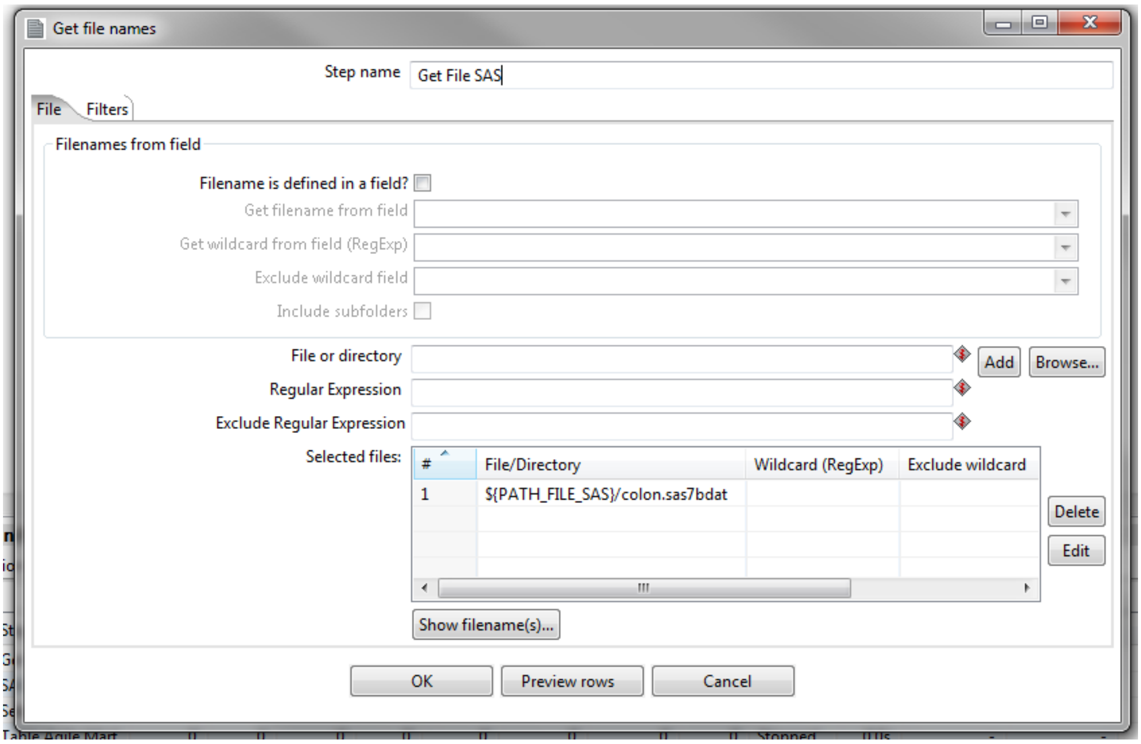


View
Design
Steps
sas
- Input
  - SAS Input
- Output
- Transform
- Utility
- Flow
- Scripting
- Lookup
- Joins
- Data Warehouse
- Validation
- Statistics
- Big Data
- Agile
- Palo
- Job
- Mapping
- Bulk loading
- Inline
- Experimental
- Deprecated
- History

Below are listed the Pentaho Data Integration components used in this document:

- **Get File Names:** This step is used to indicate the path and name of SAS source file.





Get file names

Step name: Get File SAS

Filenames from field

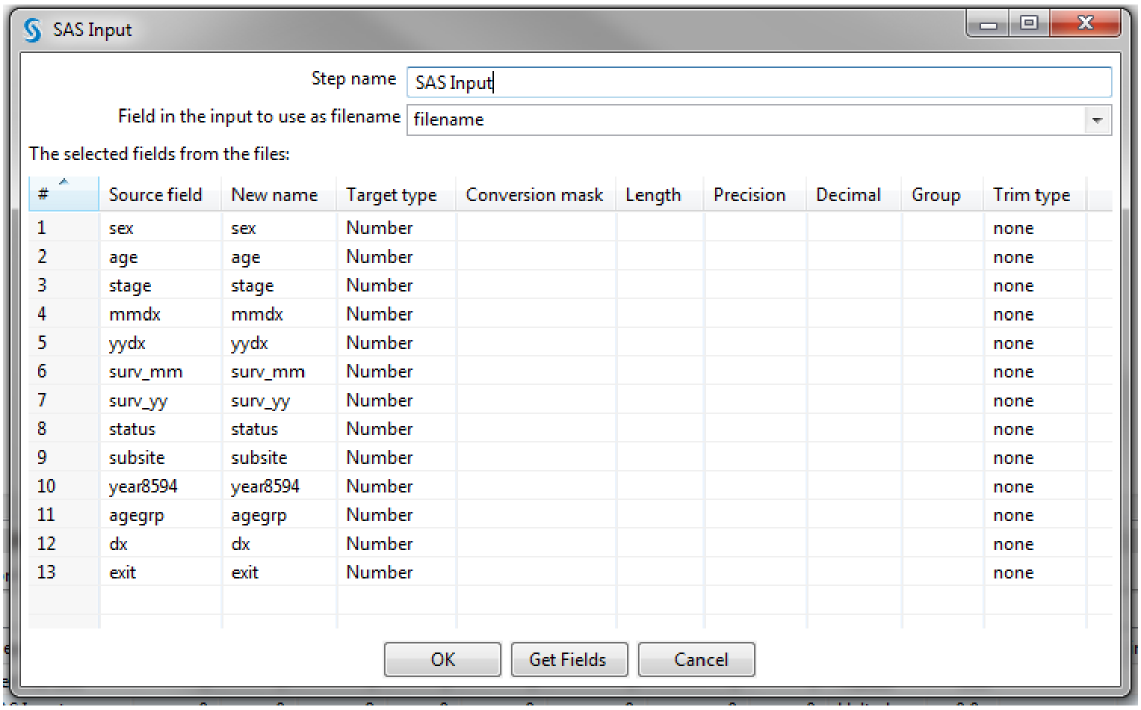
Filename is defined in a field? □
Get filename from field:
Get wildcard from field (RegExp):
Exclude wildcard field:
Include subfolders □

File or directory:
Regular Expression:
Exclude Regular Expression:
Selected files:

#	File/Directory	Wildcard (RegExp)	Exclude wildcard
1	${PATH_FILE_SAS}/colon.sas7bdat		

- **SAS Input:** At the initial stage we should click on Get Fields button to retrieve the name and properties of the fields included in the SAS file. Then if we consider it necessary we could change the properties identified by PDI.

SAS Input



SAS Input

Step name: SAS Input

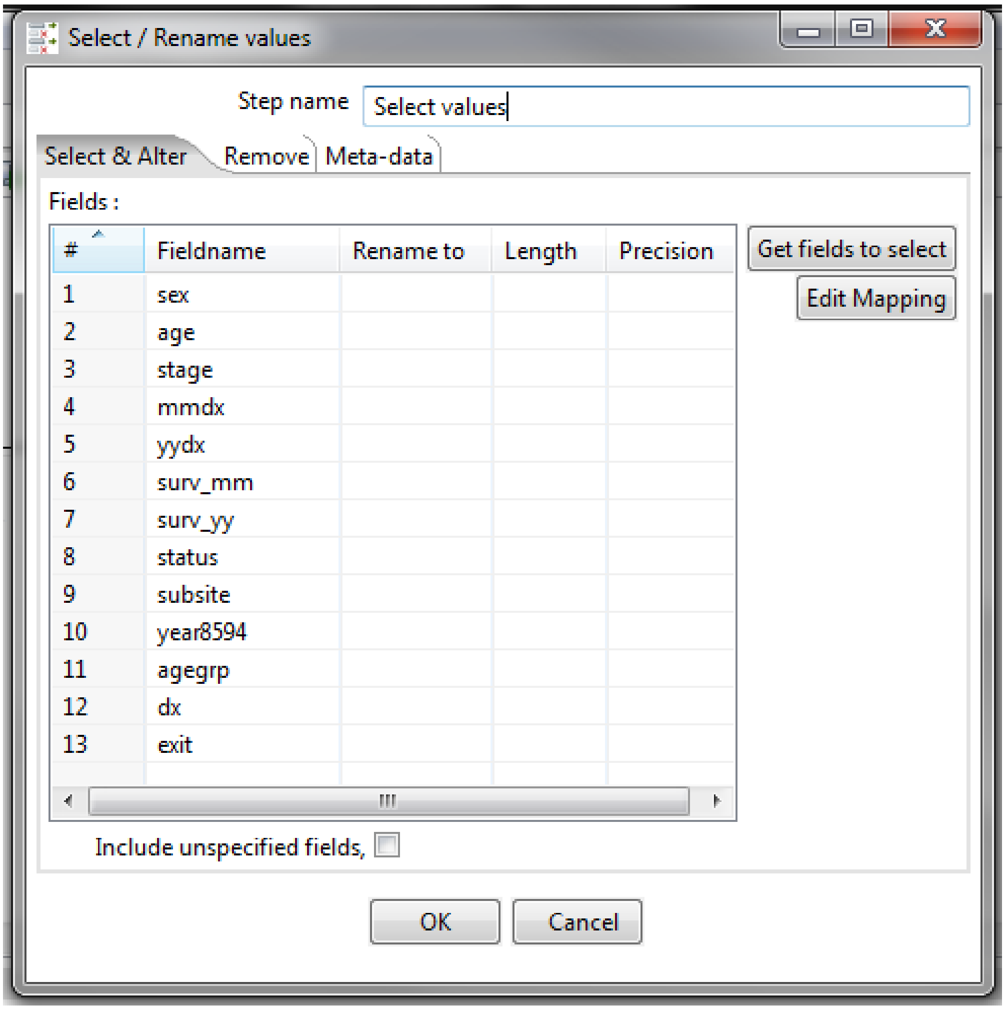
Field in the input to use as filename: filename

The selected fields from the files:

#	Source field	New name	Target type	Conversion mask	Length	Precision	Decimal	Group	Trim type
1	sex	sex	Number						none
2	age	age	Number						none
3	stage	stage	Number						none
4	mmdx	mmdx	Number						none
5	yydx	yydx	Number						none
6	surv_mm	surv_mm	Number						none
7	surv_yy	surv_yy	Number						none
8	status	status	Number						none
9	subsite	subsite	Number						none
10	year8594	year8594	Number						none
11	agegrp	agegrp	Number						none
12	dx	dx	Number						none
13	exit	exit	Number						none

- **Select values:** In this step we filter the fields from the stream.

Select values



Step name: Select values

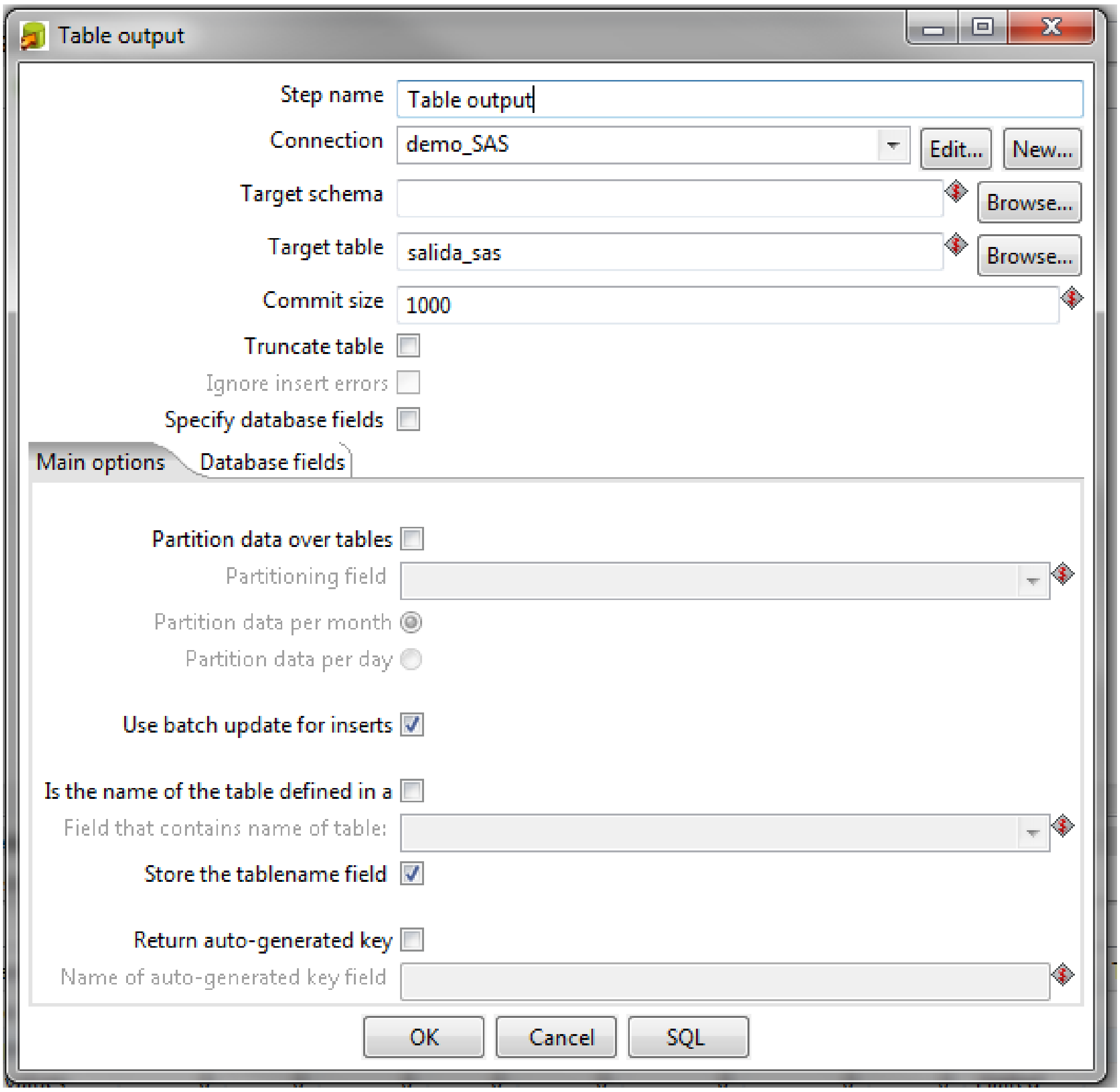
Fields :

#	Fieldname	Rename to	Length	Precision
1	sex			
2	age			
3	stage			
4	mmdx			
5	yydx			
6	surv_mm			
7	surv_yy			
8	status			
9	subsite			
10	year8594			
11	agegrp			
12	dx			
13	exit			

Include unspecified fields, □

- **Table output:** Finally we save the sample data into a MySQL table. If the table doesn’t exist PDI provides a SQL button which creates an auto-generated SQL code ready to create the table.

Table output



Step name: Table output

Connection: demo_SAS

Target schema:

Target table: salida_sas

Commit size: 1000

Truncate table □

Ignore insert errors □

Specify database fields □

Partition data over tables □

Partitioning field:

Partition data per month ●

Partition data per day ○

Use batch update for inserts ☑

Is the name of the table defined in a □

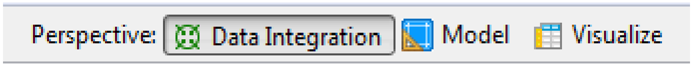
Field that contains name of table:

Store the tablename field ☑

Return auto-generated key □

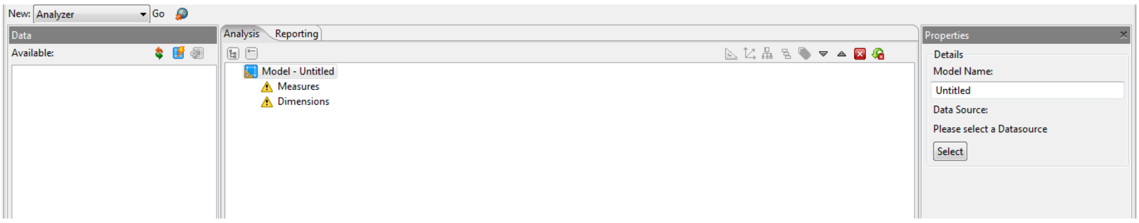
Name of auto-generated key field:

Once we have data stored in a table, we will easily visualize them using PDI. At the top right there is a Perspective selection toolbar. Up to now during the ETL design process the perspective selected was Data Integration option.



By selecting Model option we could observe at a glance the data of our model in an OLAP view (Analysis tab), besides it is also available the option of building a report with a wizard (Reporting tab).

Analysis perspective



New: Analyzer

Model - Untitled
- Measures
- Dimensions

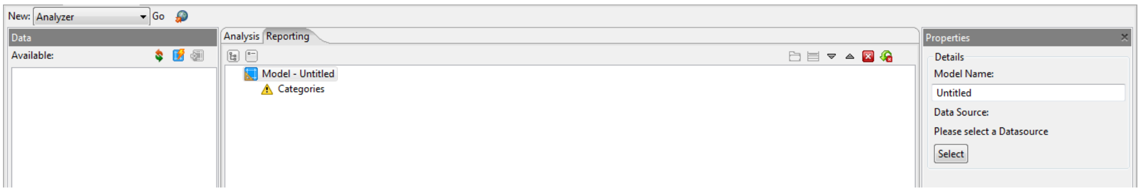
Details

Model Name: Untitled

Data Source:

Please select a Datasource

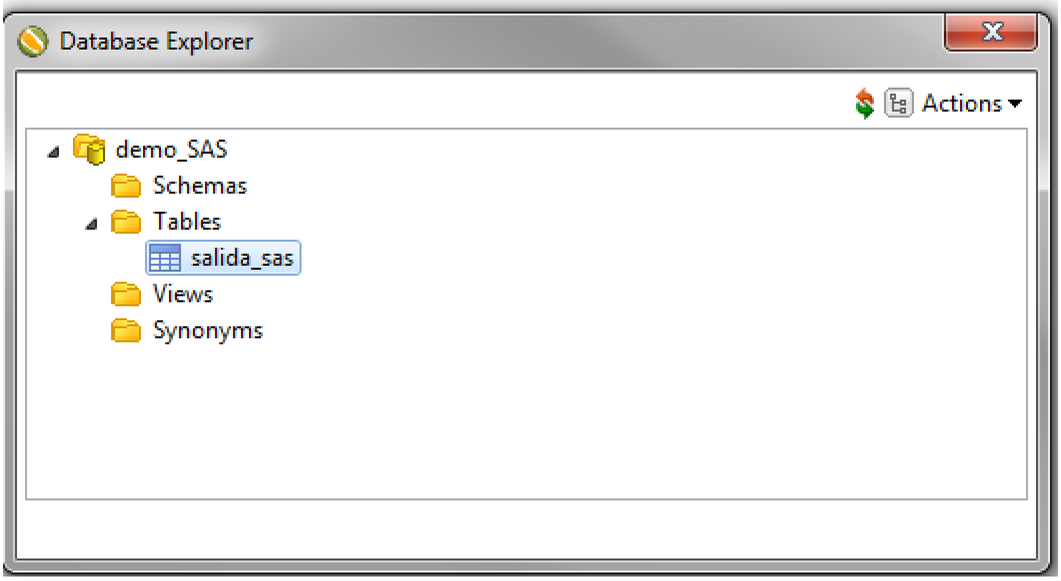
Reporting perspective



Model Name: Untitled

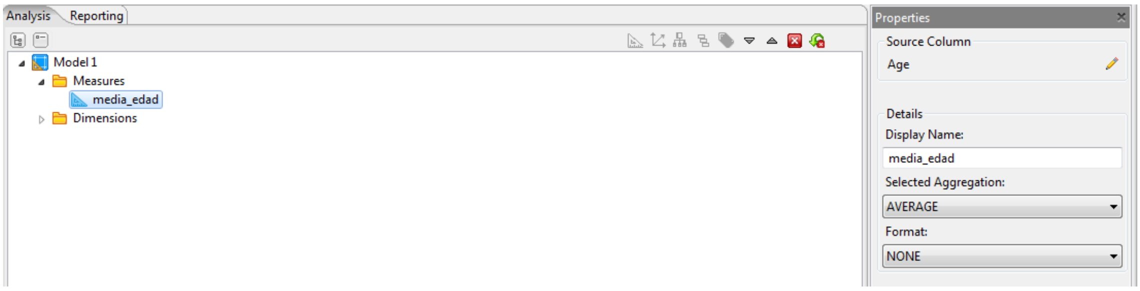
Data Source: Please select a Datasource

First we will assign a Model Name to our model and define a Datasource. Our data origin will be the MySQL table previously defined.



Next, having selected the Analysis scene we start designing a multidimensional structure. It is necessary to define at least one measure and one dimension to make use of the information in an OLAP view.

We define average age measure (media_edad) with an aggregation function of average. We select the field saved into the database table named age, as is evident with this indicator we could analyze the average age of the population.



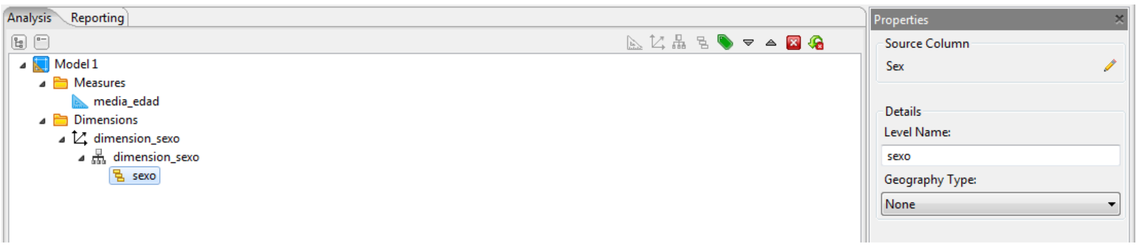
Source Column: Age

Display Name: media_edad

Selected Aggregation: AVERAGE

Format: NONE

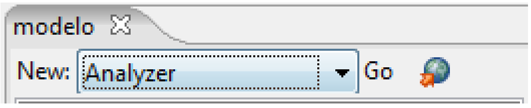
In a similar way we proceed to create a dimension containing the sex of each patient. This dimension will only include one hierarchy and one level called sex and the table field is also named sex in the database.



Source Column
Sex

Details
Level Name: sexo
Geography Type: None

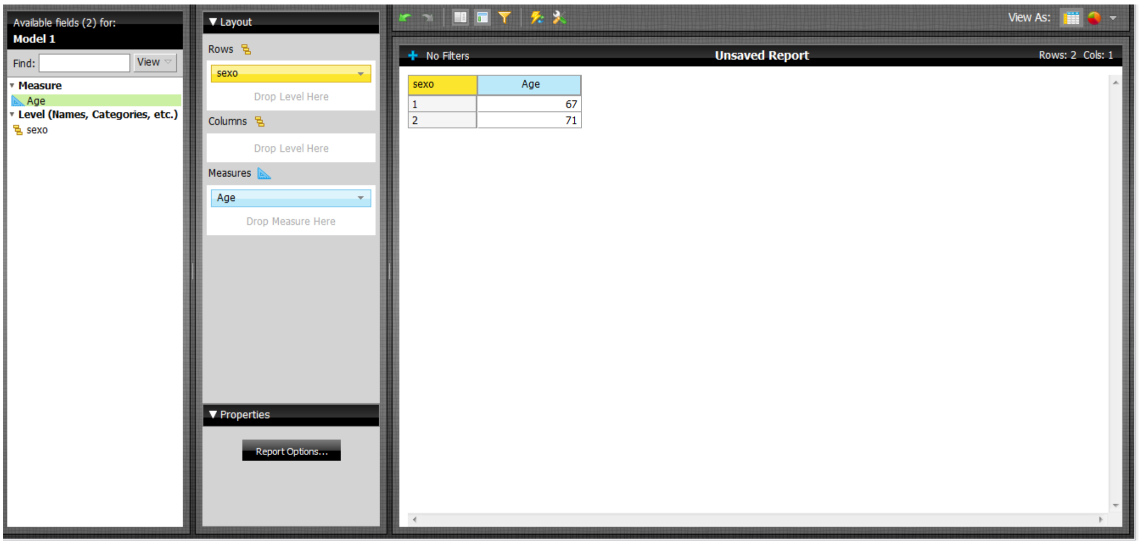
In order to visualize the OLAP structure we have developed we should only click on Go button to launch Visualize perspective.



New: Analyzer Go

Situated on this perspective we could drag and drop into rows and columns the measures and dimensions we have created in the previous stage. At the top right of the tool exists a *View as* toolbar to switch between table and chart format.

Table format

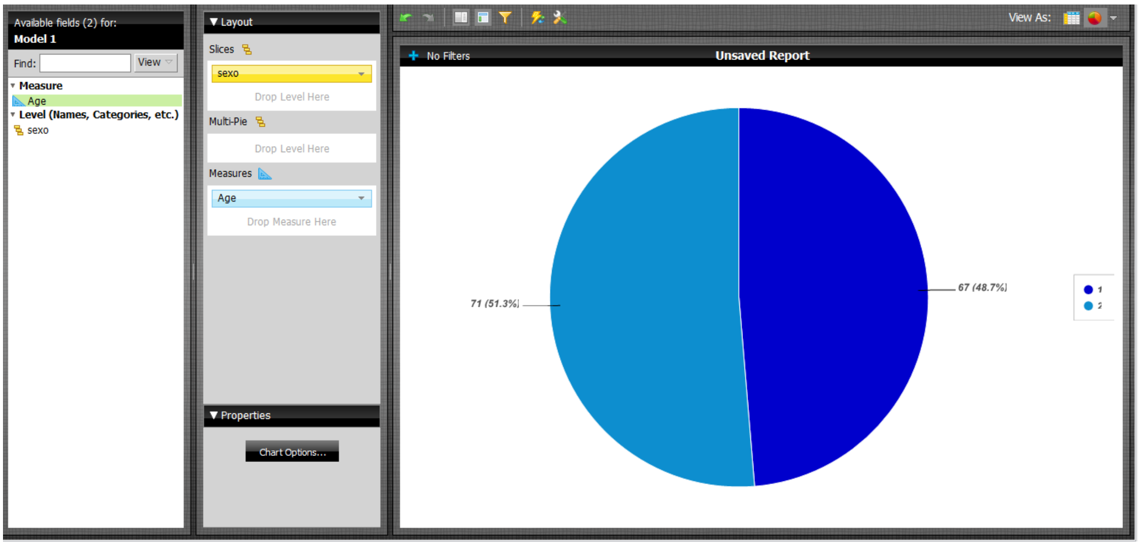


Rows: sexo
Drop Level Here
Columns
Drop Level Here
Measures: Age
Drop Measure Here
Properties

Rows: 2 Cols: 1

sexo	Age
1	67
2	71

Chart format



Available fields (2) for:
Model 1

Find: View

- Measure
  - Age
- Level (Names, Categories, etc.)
  - sexo

Layout

Slices: sexo

Drop Level Here

Multi-Pie: Drop Level Here

Measures: Age

Drop Measure Here

Properties

Chart Options...

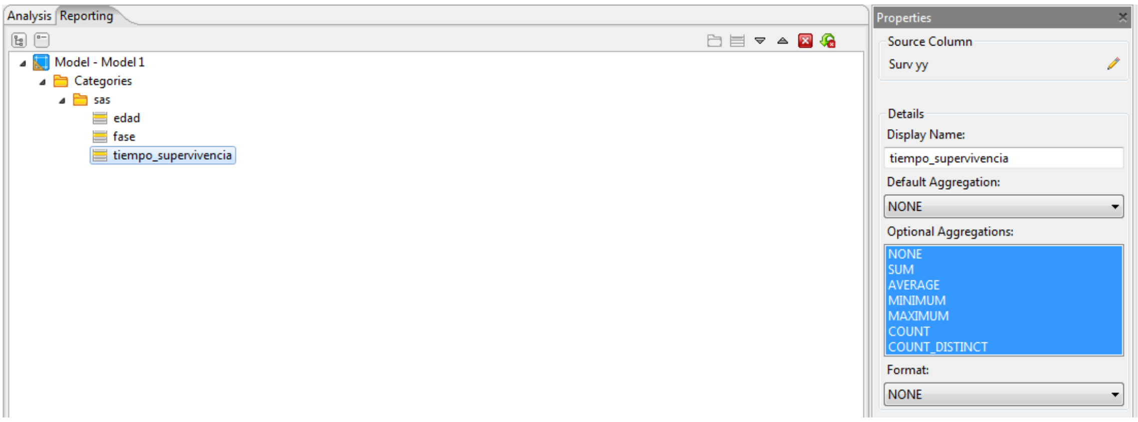
View As:

No Filters

Unsaved Report

Then, we move to define a reporting metadata structure. At first, it is required the creation of a category and select the fields that we want to have available in further stages. These are the fields chosen:

- Age (edad) → age field in database
- Clinical stage at diagnosis (fase) → stage field in database
- Survival time in completed years (tiempo_superviviencia) → surv_yy field in database



Analysis | Reporting

- Model - Model 1
  - Categories
    - sas
      - edad
      - fase
      - tiempo_supervivencia

Properties

Source Column: Surv yy

Details

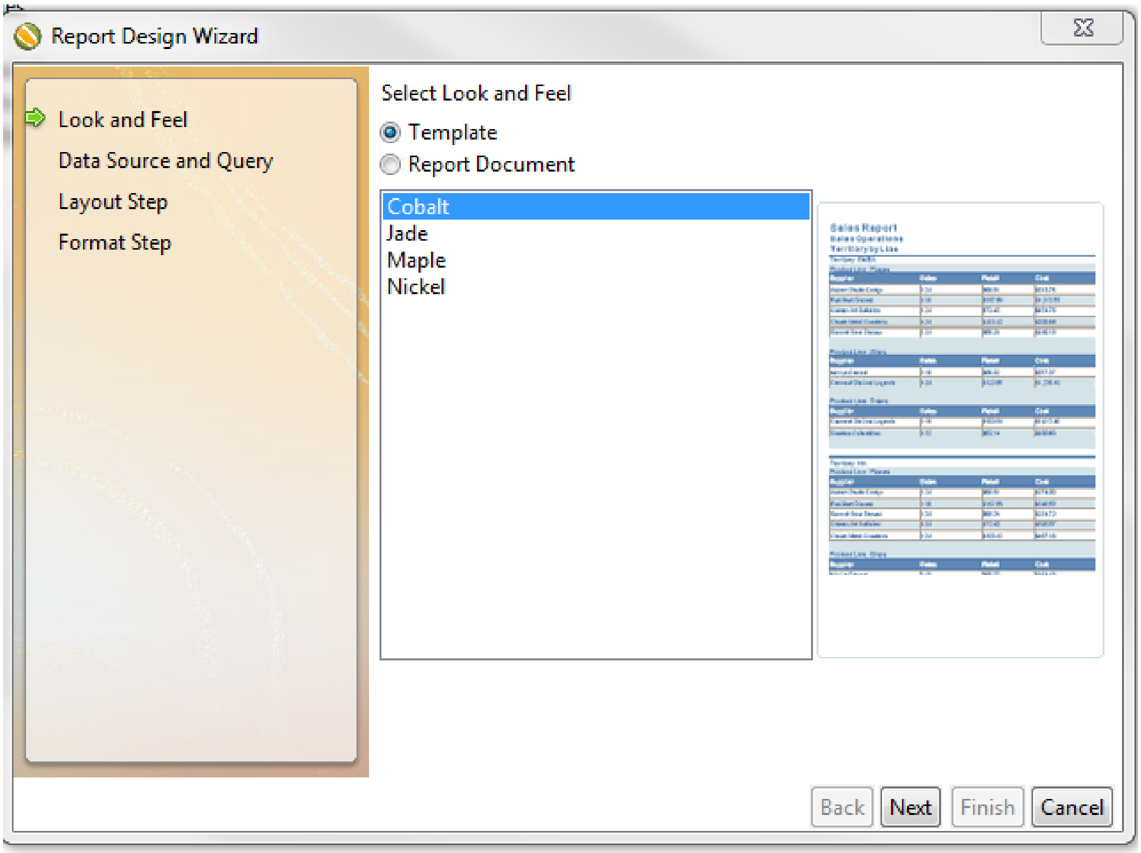
Display Name: tiempo_supervivencia

Default Aggregation: NONE

Optional Aggregations: NONE, SUM, AVERAGE, MINIMUM, MAXIMUM, COUNT, COUNT_DISTINCT

Format: NONE

In order to gain knowledge of the metadata we have just generated, we select Report Wizard and click on Go button. Automatically we will be directed to Visualize perspective, here we have to choose a template.



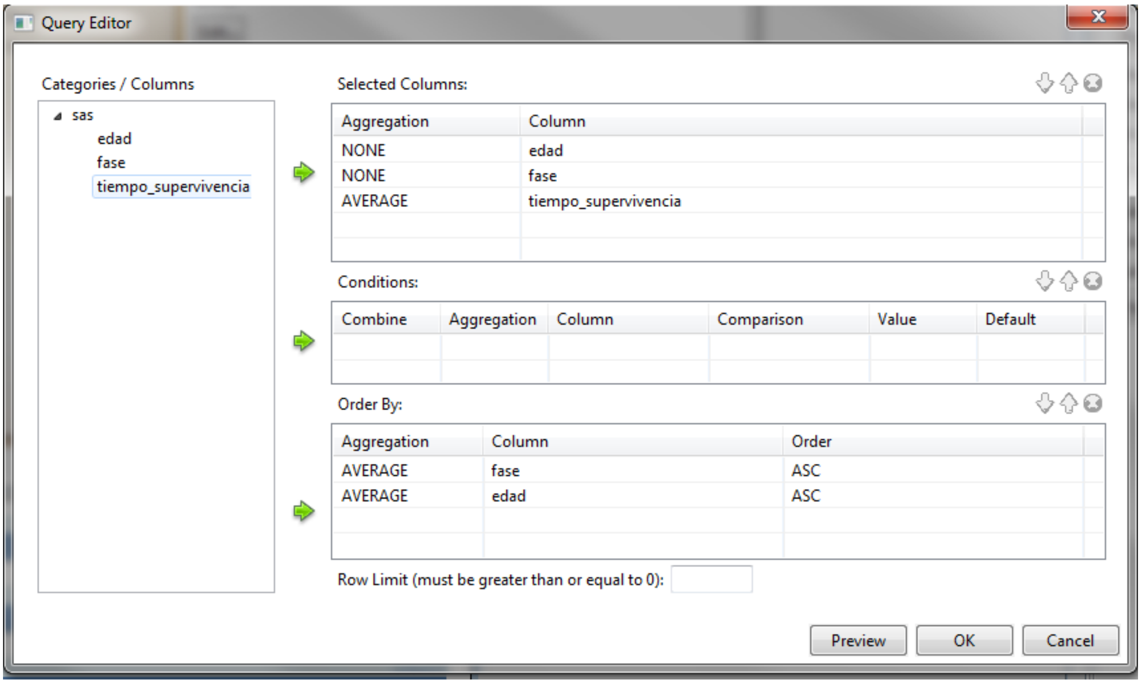
Select Look and Feel

● Template

○ Report Document

Cobalt
Jade
Maple
Nickel

Next, we select the fields we want to show in the report. We choose age (edad), clinical stage at diagnosis (fase) and the average of survival time in completed years (tiempo_supervivencia) sorted in ascending order by clinical stage and patient age.



Selected Columns:

Aggregation	Column
NONE	edad
NONE	fase
AVERAGE	tiempo_supervivencia

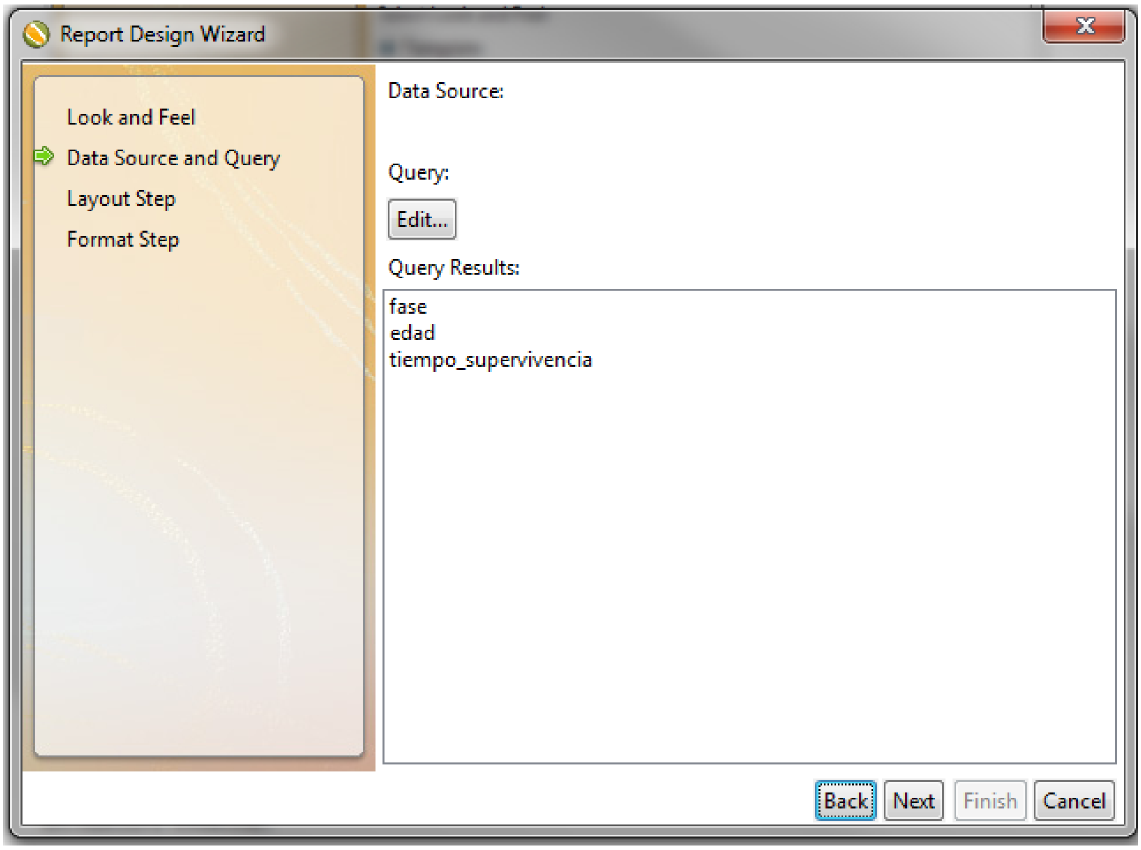
Conditions:

Combine	Aggregation	Column	Comparison	Value	Default

Order By:

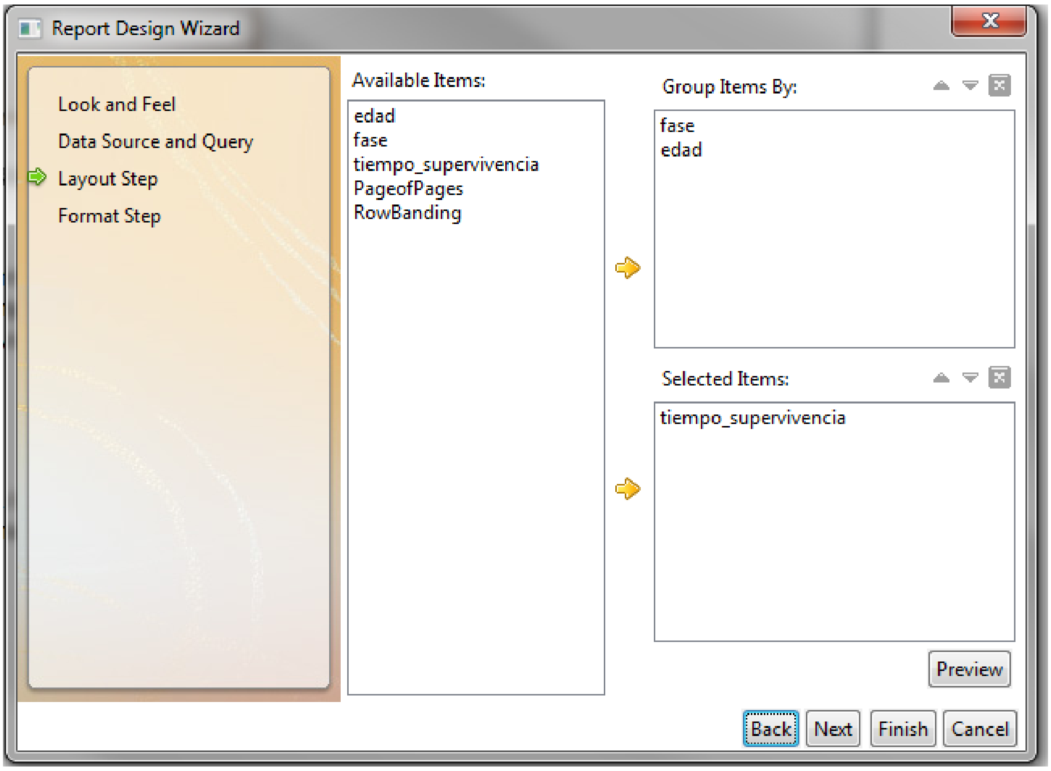
Aggregation	Column	Order
AVERAGE	fase	ASC
AVERAGE	edad	ASC

Row Limit (must be greater than or equal to 0):



fase
edad
tiempo_supervivencia

Then, we design the layout of the report. Our main objective is to show the results grouped by clinical stage and patient age.

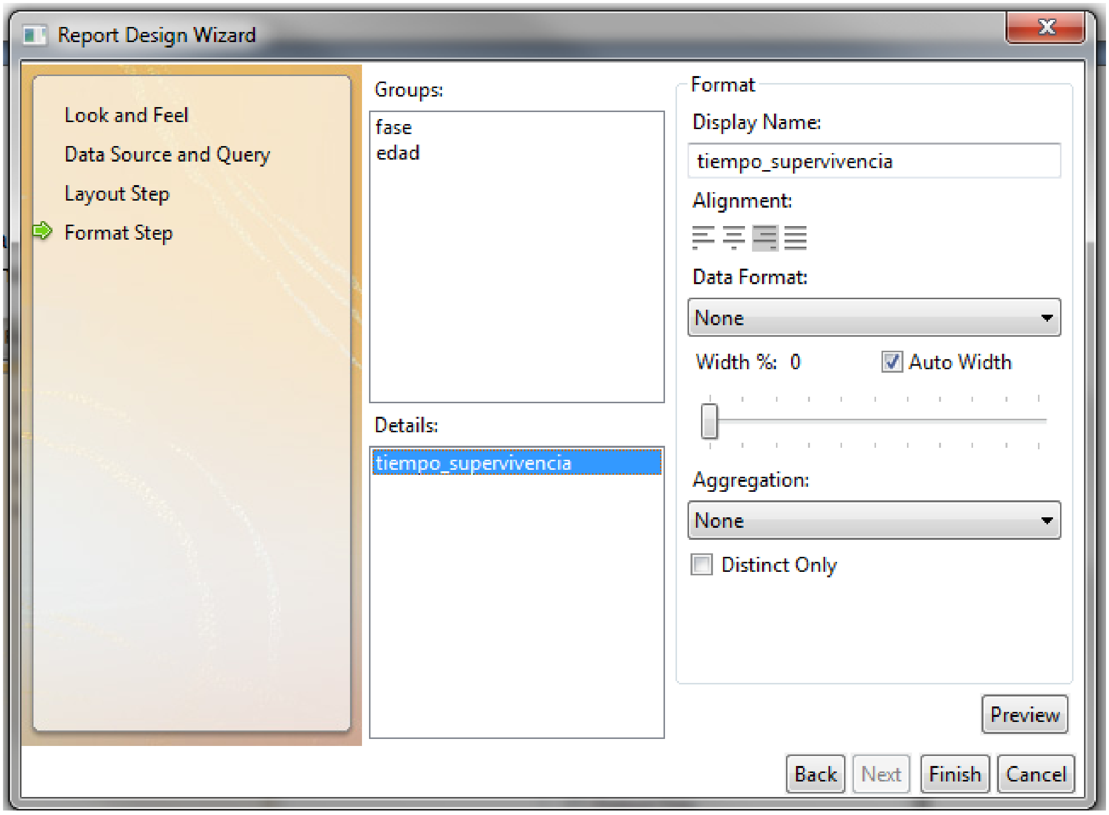


Available Items:
edad
fase
tiempo_supervivencia
PageofPages
RowBanding

Group Items By:
fase
edad

Selected Items:
tiempo_supervivencia

After that we could change format properties such as alignment, width, format strings …



Groups:
fase
edad

Details:
tiempo_supervivencia

Format

Display Name: tiempo_supervivencia

Alignment:

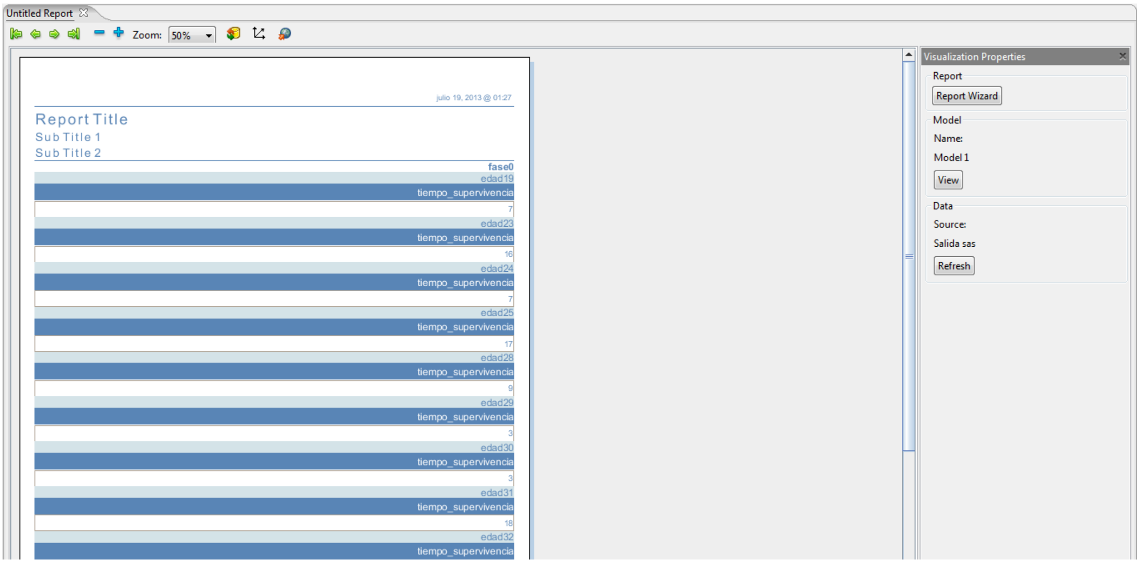
Data Format: None

Width %: 0 ☑ Auto Width

Aggregation: None

□ Distinct Only

Finally, we have our report finished in less than 5 minutes. It is possible to edit the report by pushing Report Wizard button located in the Visualization Properties palette located at the right side of the tool.



julio 19, 2013 @ 01:27

Report Title
Sub Title 1
Sub Title 2

fase0
edad19
tiempo_supervivencia
7
edad23
tiempo_supervivencia
16
edad24
tiempo_supervivencia
7
edad25
tiempo_supervivencia
17
edad28
tiempo_supervivencia
9
edad29
tiempo_supervivencia
3
edad30
tiempo_supervivencia
3
edad31
tiempo_supervivencia
18
edad32
tiempo_supervivencia

Report

Model
Name:
Model 1

Data
Source:
Salida sas

This powerful tool allows us to save the metadata structure previously generated with .xmi extension available to reuse it in future.