

03

Information Visualization

Daniel Gonçalves, Sandra Gama

Pentaho Data Integration Tutorial



01

PENTaho DATA INTEGRATION: INTRODUCTION

PDI - Pentaho Data Integration (Kettle)

“Delivers powerful

Extraction,

Transformation

Loading

capabilities, using a groundbreaking, metadata-driven approach.”

Starting point: raw data file

Download the data file
(course page > labs' section)

Check your data (two files)!

Open with Excel: world population.csv

Check your data (two files)!

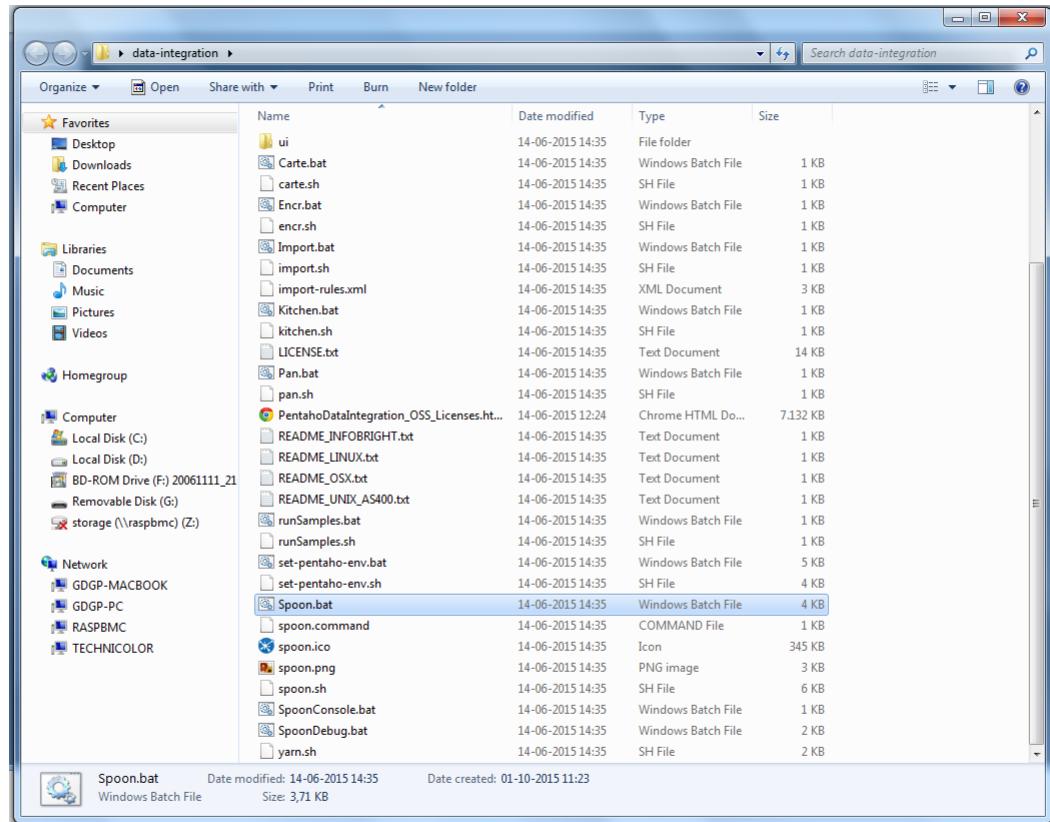
Open with Excel: GDP.csv

Let's transform it with Pentaho Data Integration

Download

<http://community.pentaho.com/projects/data-integration/>

Unzip (anywhere) & run!



Run
Spoon.bat

PDI – Spoon Interface

Spoon - Welcome!

File Edit View Action Tools Help

View Design

Steps

Welcome! file:///C:/Users/gdgp/Desktop/data-integration/docs/English/welcome/index.html

Perspective: Data Integration

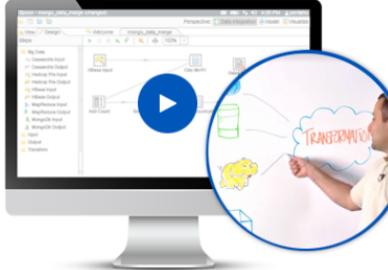
Pentaho Data Integration

Welcome Meet the Family Credits Many Reasons to Get Enterprise Edition

Get the Most From Pentaho

Let us help you become an ETL, Big Data Master.

Tutorials & Videos →



Take your ETL, Analytics & Big Data to the Next Level

Want to take your implementation to the next level? Experience for yourself our comprehensive data integration, visualization and analysis tools and create customizable reports and interactive dashboards.

Get Pentaho Enterprise Edition

Getting Started

New user? Want to know how to get started? Check out our documentation

Documentation

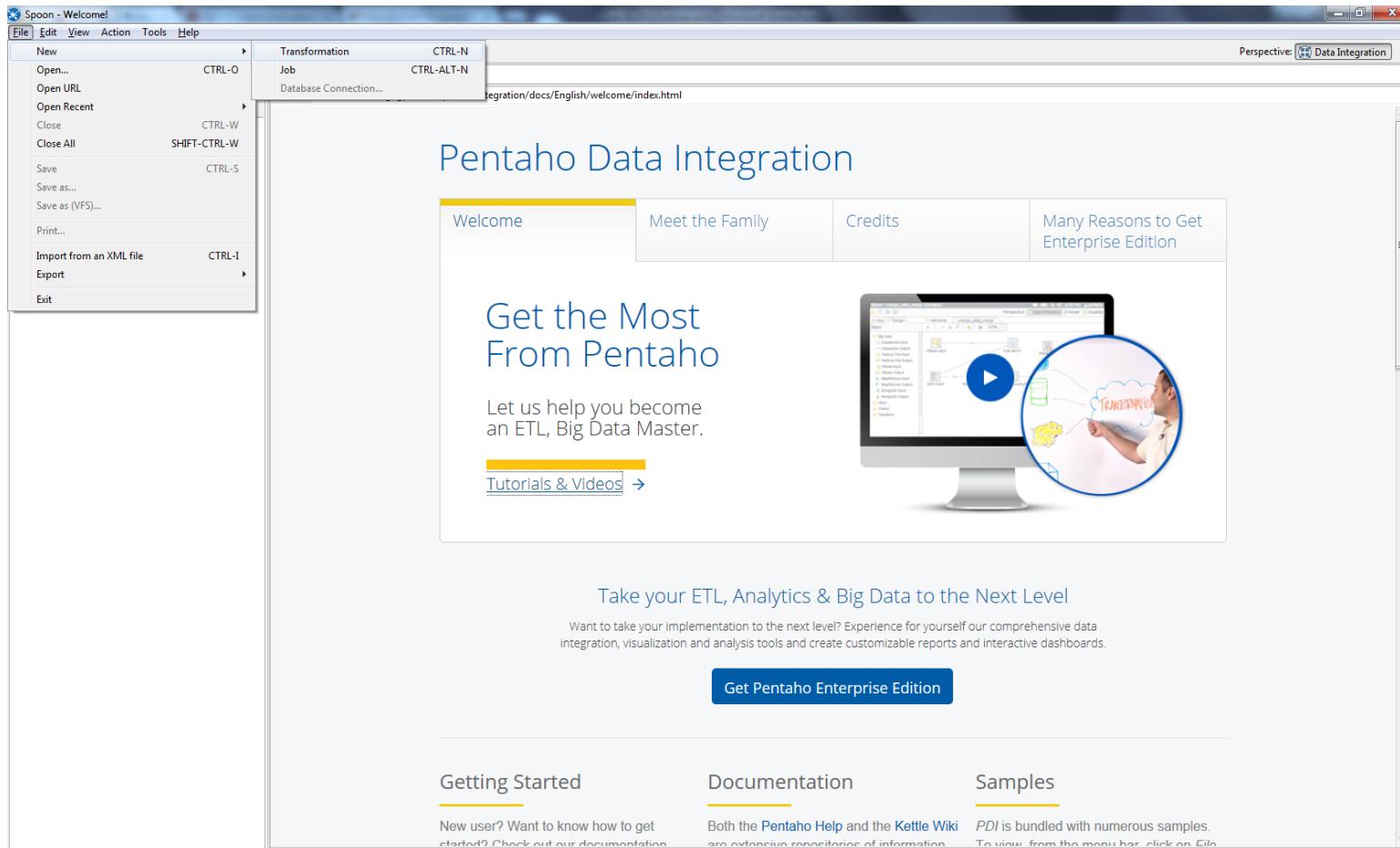
Both the [Pentaho Help](#) and the [Kettle Wiki](#) are extensive repositories of information

Samples

PDI is bundled with numerous samples. To view, from the menu bar, click on File

Create new Data Transformation

File > New > Transformation



The screenshot shows the Pentaho Spoon application window. The title bar reads "Spoon - Welcome!". The menu bar includes "File", "Edit", "View", "Action", "Tools", and "Help". The "File" menu is open, showing options like "New", "Open...", "Job", "Database Connection...", and "Exit". The "Transformation" option is highlighted. The main content area displays the "Pentaho Data Integration" welcome page. It features a navigation bar with "Welcome", "Meet the Family", "Credits", and "Many Reasons to Get Enterprise Edition". Below this is a section titled "Get the Most From Pentaho" with the subtext "Let us help you become an ETL, Big Data Master." A "Tutorials & Videos" button is present. To the right, there's a graphic of a computer monitor displaying the Spoon interface, with a circular callout highlighting a "TRANSFORMATION" node in the flow. At the bottom, a button says "Get Pentaho Enterprise Edition". The footer contains links for "Getting Started", "Documentation", and "Samples".

Pentaho Data Integration

Welcome Meet the Family Credits Many Reasons to Get Enterprise Edition

Get the Most From Pentaho

Let us help you become an ETL, Big Data Master.

Tutorials & Videos →

Take your ETL, Analytics & Big Data to the Next Level

Want to take your implementation to the next level? Experience for yourself our comprehensive data integration, visualization and analysis tools and create customizable reports and interactive dashboards.

Get Pentaho Enterprise Edition

Getting Started

New user? Want to know how to get started? Check out our documentation

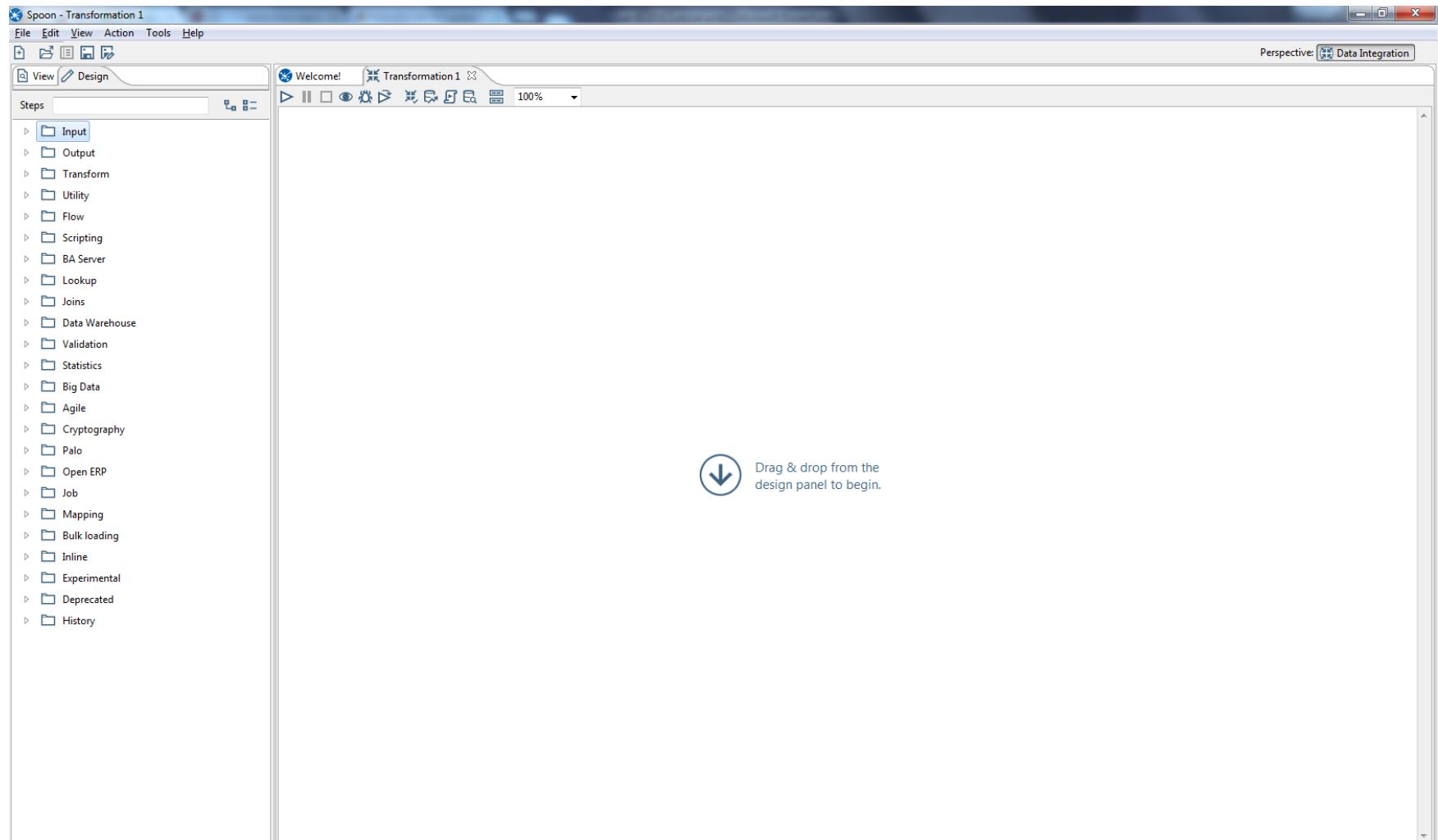
Documentation

Both the [Pentaho Help](#) and the [Kettle Wiki](#) are extensive repositories of information

Samples

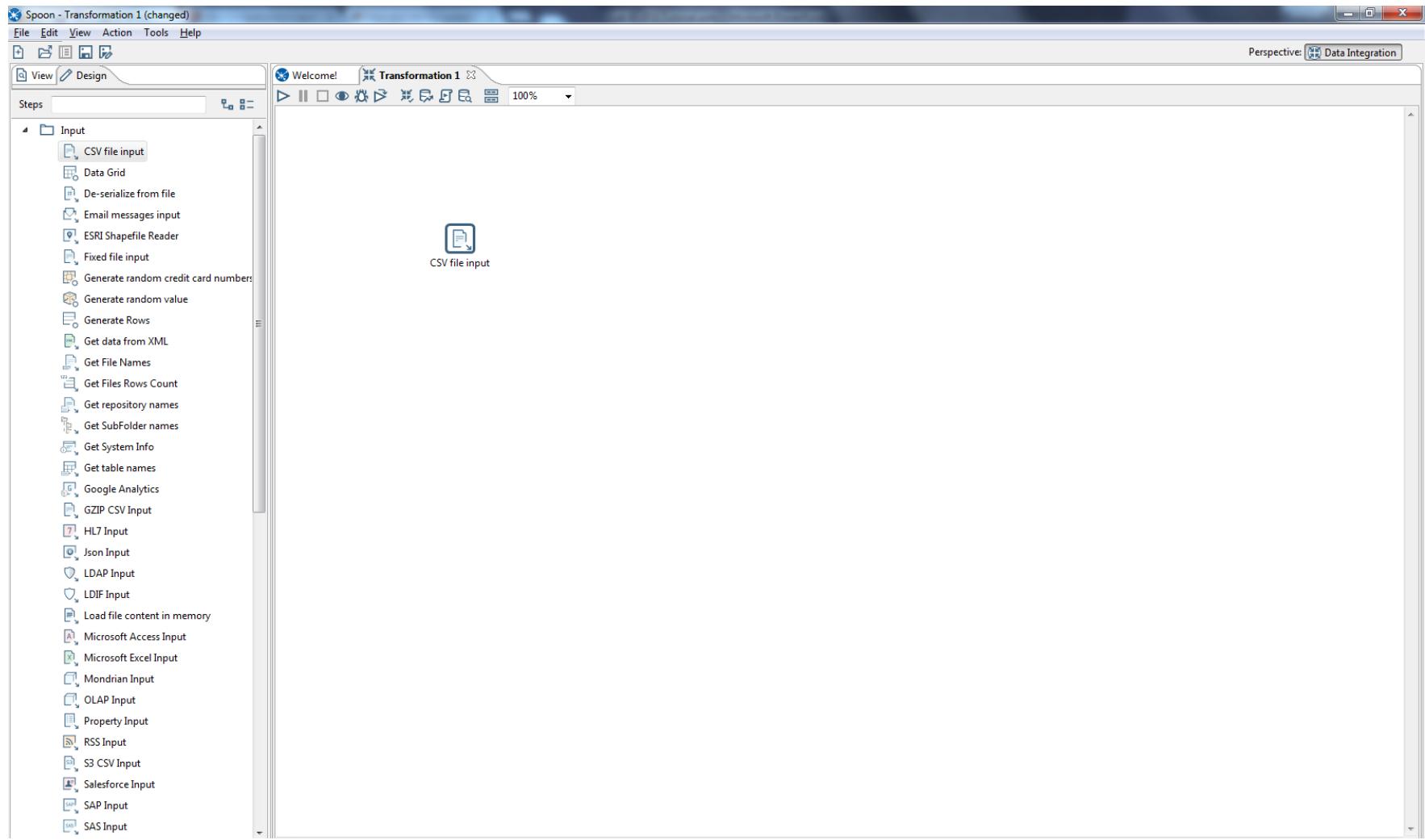
PDI is bundled with numerous samples. To view them from the menu bar, click on [File](#).

New transformation created



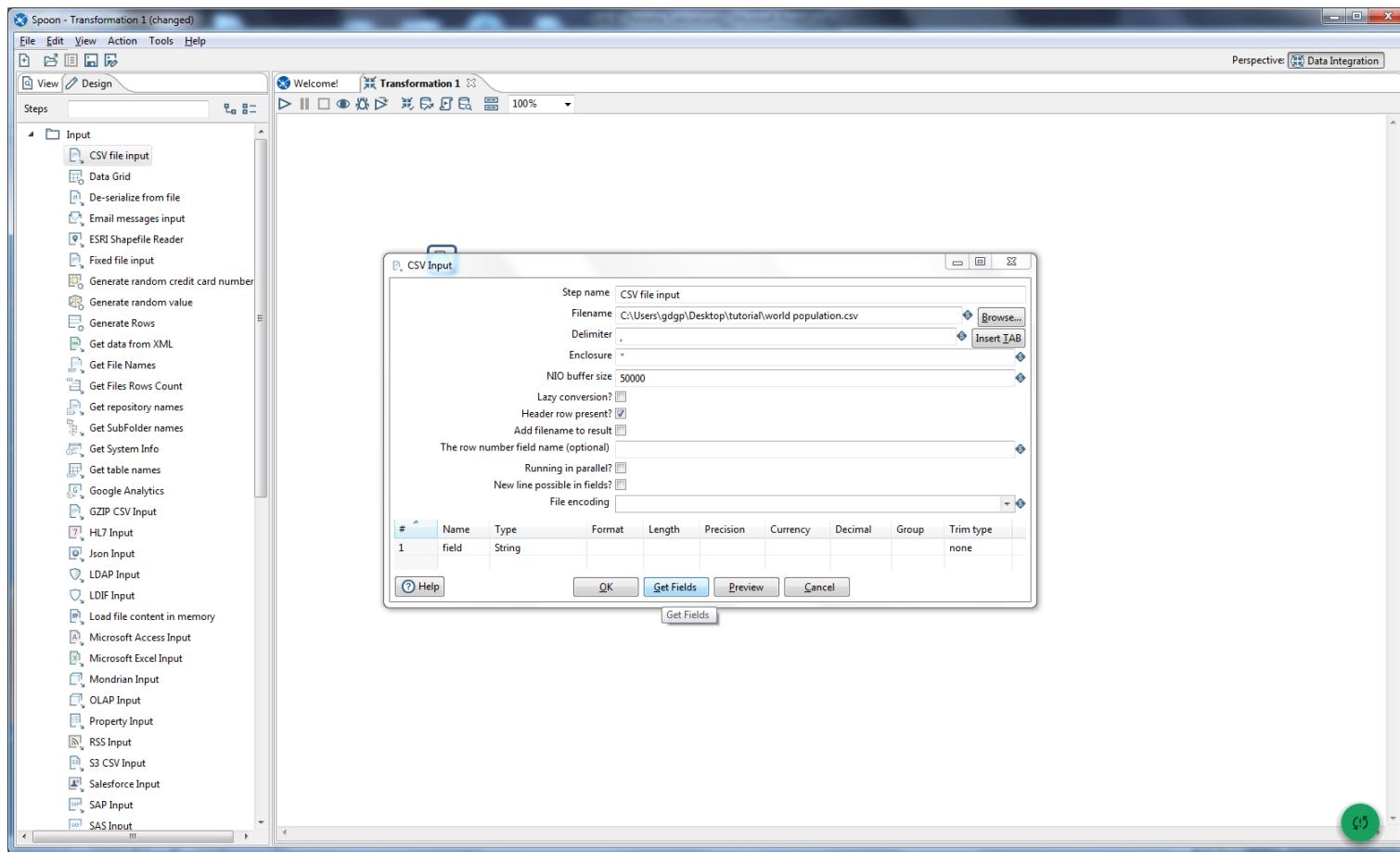
Now let's use our data file as input.

Input > CSV File input > drag & drop

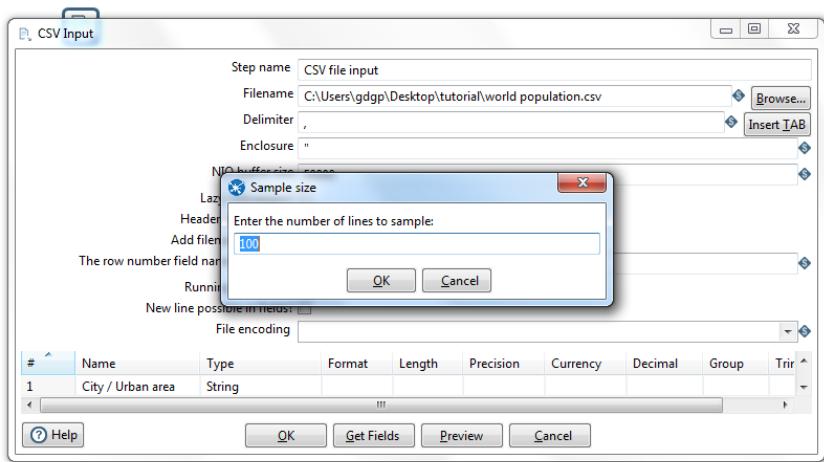


Right click to adjust properties

Define step name and filename (browse your data file)



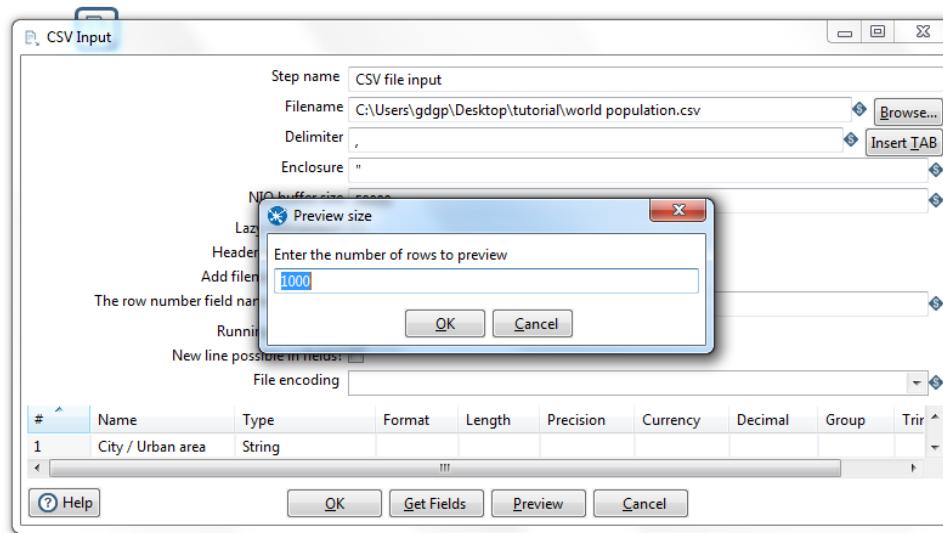
Get fields...



Scan results	
Here are the results of the document scan:	
Result after scanning 100 lines.	
Field nr. 1 :	
Field name	: City / Urban area
Field type	: String
Maximum length	: 22
Minimum value	: Ankara
Maximum value	: Washington
Nr of null values	: 0
Field nr. 2 :	
Field name	: Country
Field type	: String
Maximum length	: 12
Minimum value	: Argentina
Maximum value	: Vietnam
Nr of null values	: 0
Field nr. 3 :	
Field name	: Population
Field type	: Number
Estimated length	: 15
Estimated precision	: 0
Number format	: #,###,###.##
WARNING: More than 1 number format seems to match all sampled records:	
Number format	: #,###,###.##
Trim Type	: 0
Minimum value	: 2000000.0
Maximum value	: 3.32E7
Example	: #,###,###.##, number [2000000.0] gives 2000000.0
Number format	: #,###,###.##
Trim Type	: 3
Minimum value	: 2000000.0
Maximum value	: 3.32E7
Example	: #,###,###.##, number [2000000.0] gives 2000000.0
Nr of null values	: 0
Field nr. 4 :	
Field name	: Land area
Field type	: Number
Estimated length	: 15
Estimated precision	: 0
Number format	: #.#
WARNING: More than 1 number format seems to match all sampled records:	
Number format	: #.#
Trim Type	: 0
Minimum value	: 1.01
Maximum value	: 984.0
Example	: #.#, number [1.01] gives 1.01
Number format	: #.0
Trim Type	: 0
Minimum value	: 1.01
Maximum value	: 984.0
Example	: #.0, number [1.01] gives 1.01
Number format	: #.#
Trim Type	: 3
Minimum value	: 1.01
Maximum value	: 984.0
Example	: #.#, number [1.01] gives 1.01
Number format	: #.0
Trim Type	: 3
Minimum value	: 1.01
Maximum value	: 984.0
Example	: #.0, number [1.01] gives 1.01
Number format	: #.##
Trim Type	: 0

Now we have all the
fields

Preview...



Preview...

There you go!

Examine preview data

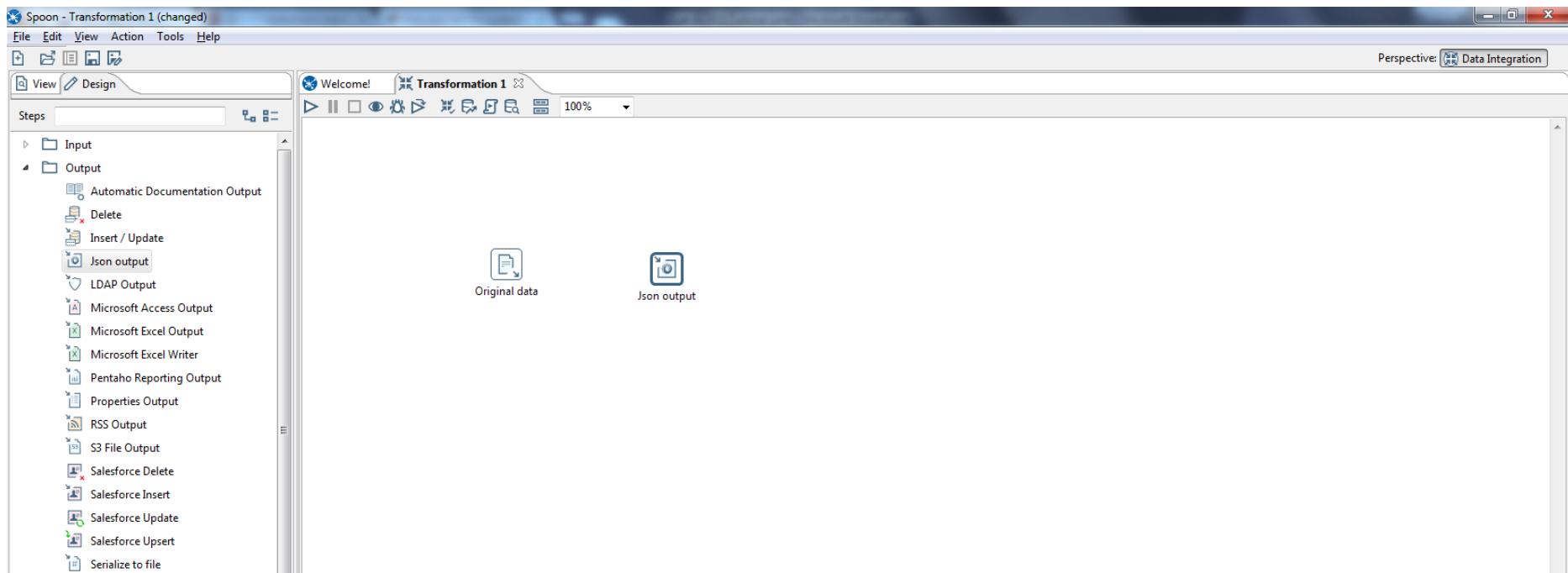
Rows of step: CSV file input (125 rows)

#	City / Urban area	Country	Population	Land area	Density
1	Tokyo/Yokohama	Japan	33,200,000	6,993	4,750
2	New York Metro	USA	17,800,000	8,683	2,050
3	Sao Paulo	Brazil	17,700,000	1,968	9,000
4	Seoul/Incheon	South Korea	17,500,000	1,049	16,700
5	Mexico City	Mexico	17,400,000	2,072	8,400
6	Osaka/Kobe/Kyoto	Japan	16,425,000	2,564	6,400
7	Manila	Philippines	14,750,000	1,399	10,550
8	Mumbai	India	14,350,000	484	29,650
9	Delhi	India	14,300,000	1,295	11,050
10	Jakarta	Indonesia	14,250,000	1,360	10,500
11	Lagos	Nigeria	13,400,000	738	18,150
12	Kolkata	India	12,700,000	531	23,900
13	Cairo	Egypt	12,200,000	1,295	9,400
14	Los Angeles	USA	11,789,000	4,320	2,750
15	Buenos Aires	Argentina	11,200,000	2,266	4,950
16	Rio de Janeiro	Brazil	10,800,000	1,580	6,850
17	Moscow	Russia	10,500,000	2,150	4,900
18	Shanghai	China	10,000,000	746	13,400
19	Karachi	Pakistan	9,800,000	518	18,900
20	Paris	France	9,645,000	2,723	3,550
21	Istanbul	Turkey	9,000,000	1,166	7,700
22	Nagoya	Japan	9,000,000	2,875	3,150
23	Beijing	China	8,614,000	748	11,500
24	Chicago	USA	8,308,000	5,498	1,500
25	London	UK	8,278,000	1,623	5,100
26	Shenzhen	China	8,000,000	466	17,150
27	Essen/Düsseldorf	Germany	7,350,000	2,642	2,800
28	Tehran	Iran	7,250,000	686	10,550
29	Bogota	Colombia	7,000,000	518	13,500
30	Lima	Peru	7,000,000	596	11,750
31	Bangkok	Thailand	6,500,000	1,010	6,450
32	Johannesburg/East Rand	South Africa	6,000,000	2,396	2,500
33	Chennai	India	5,950,000	414	14,350

What to do with this data?

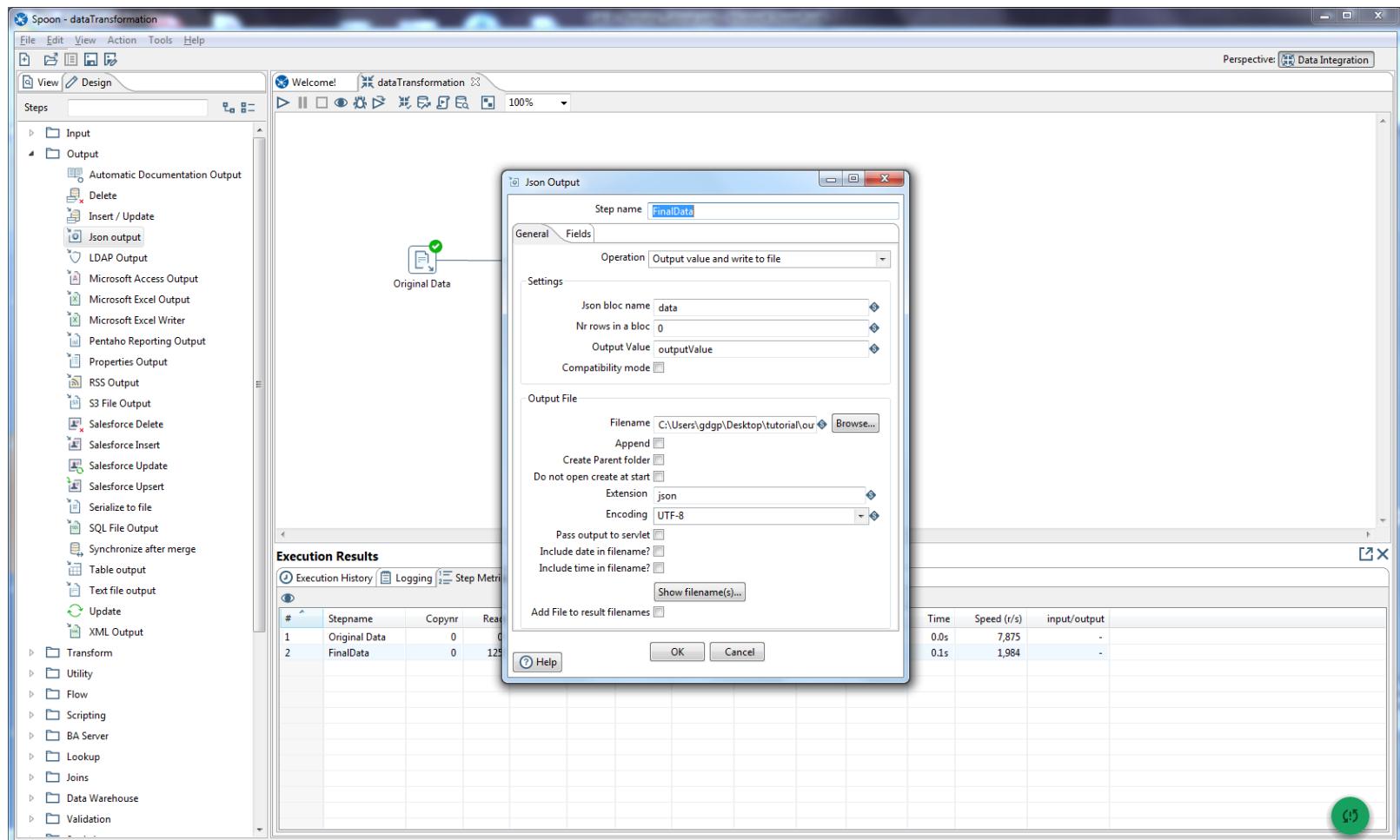
Let's write it to a .json file

Output > Json output > drag&drop



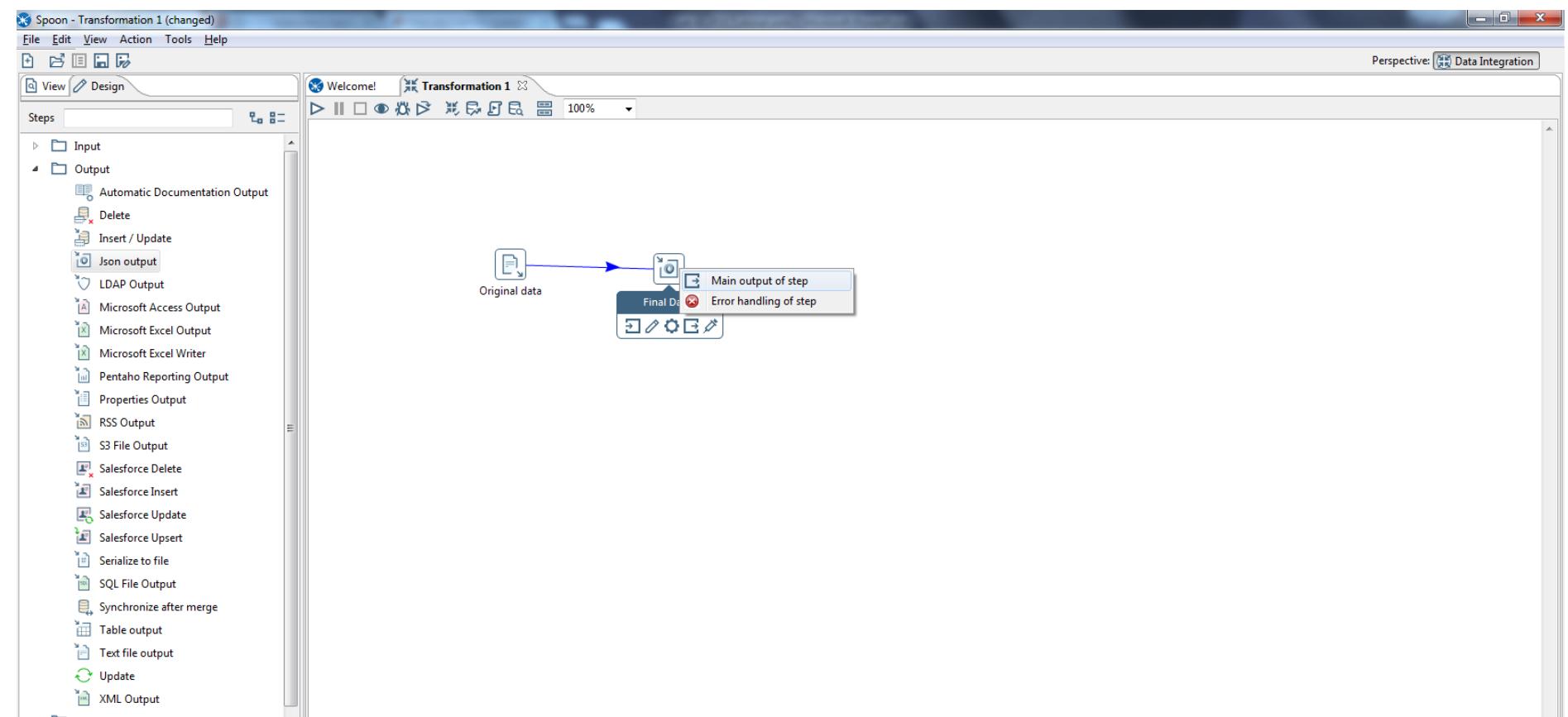
Let's configure the output

Configure step name, file name and extension



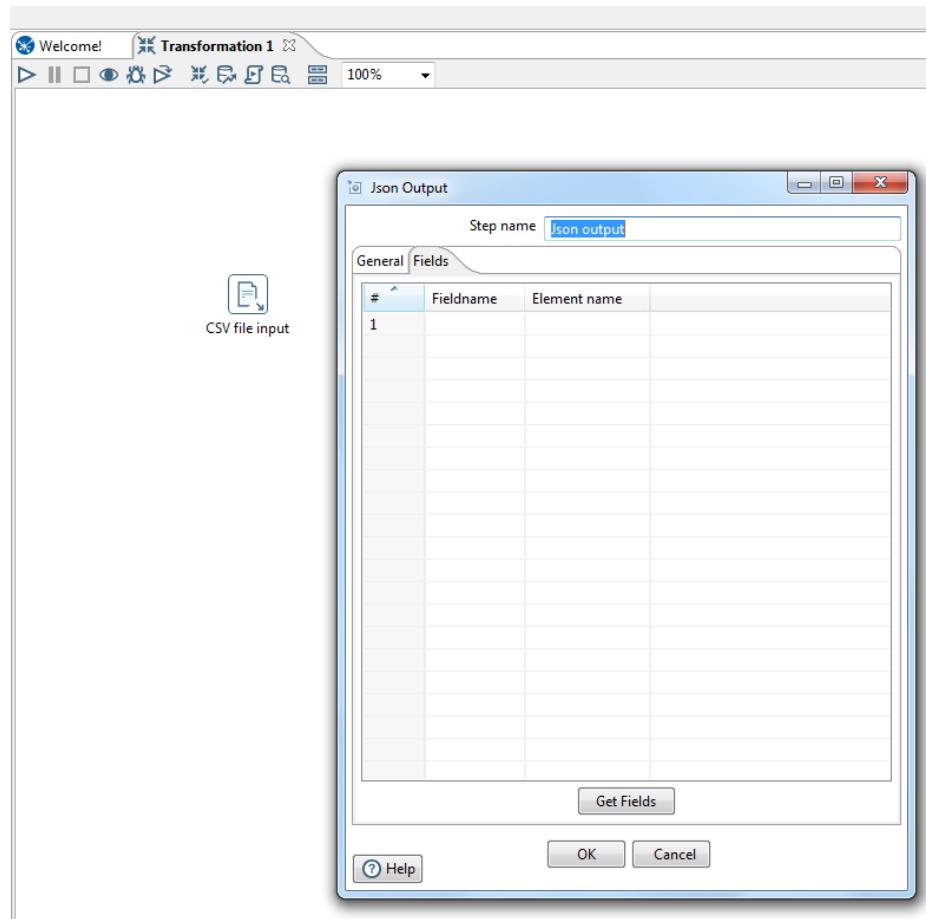
Connect input and output

Shift + click, drag, click



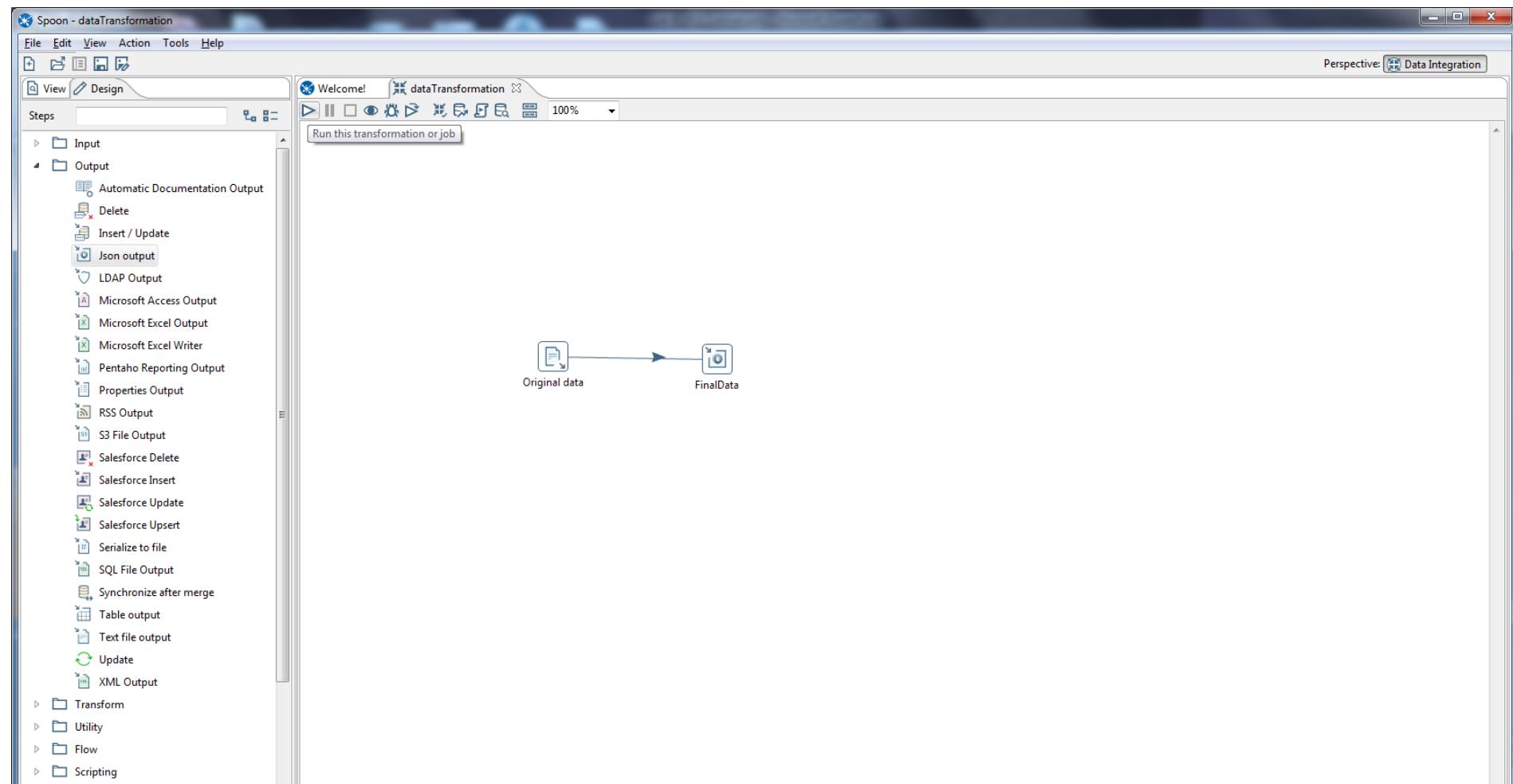
Let's configure the output

Fields > get fields



Run!

Save transformation, run and launch



Data transformation complete

Check execution results

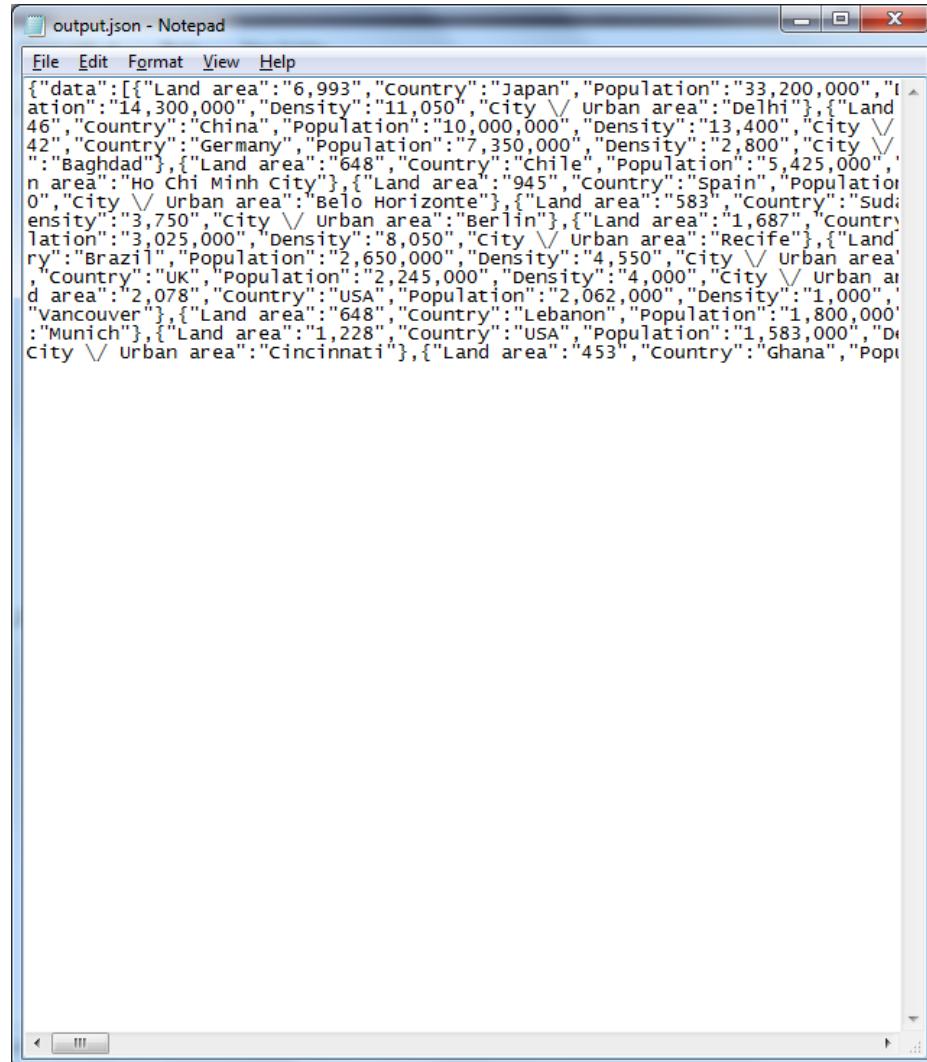
The screenshot shows the Apache Nifi Spoon interface for a data transformation named "dataTransformation".

The main workspace displays a single step: "Original Data" followed by a "FinalData" step, connected by a flow arrow.

The "Execution Results" panel at the bottom provides detailed metrics for each step:

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Original Data	0	0	125	126	0	0	0	0	Finished	0.0s	7,875	-
2	FinalData	0	125	1	0	1	0	0	0	Finished	0.1s	1,984	-

Now let's see the .json file



A screenshot of a Windows Notepad window titled "output.json - Notepad". The window displays a large amount of JSON data. The data is an array of objects under the key "data". Each object contains "Land area", "Country", "Population", "Latitude", "Longitude", "Density", and "city \ Urban area". The data includes entries for major cities like Delhi, Beijing, Berlin, and London, along with their respective countries and population densities.

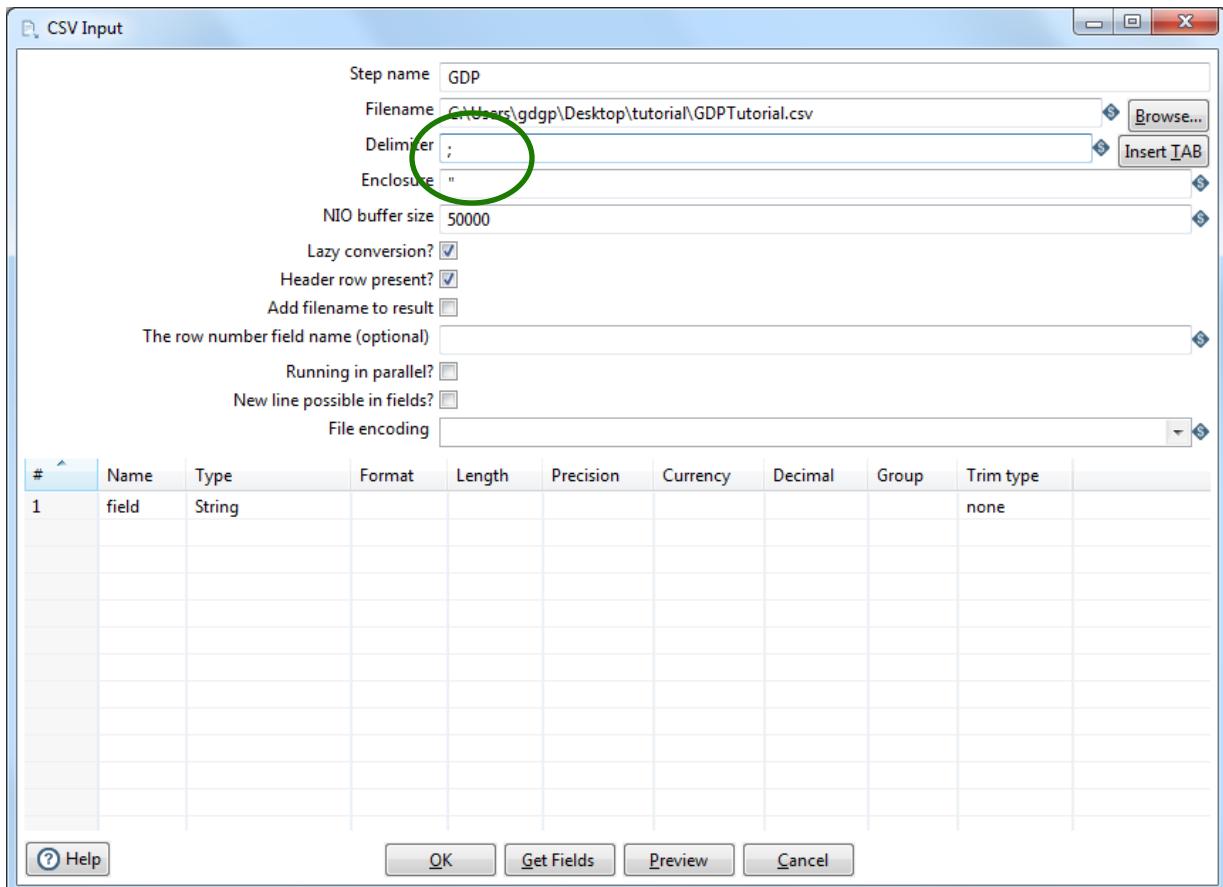
```
{"data": [{"Land area": "6,993", "Country": "Japan", "Population": "33,200,000", "Latitude": "35.7", "Longitude": "139.75", "Density": "1,050", "city \ Urban area": "Tokyo"}, {"Land area": "14,300,000", "Country": "India", "Population": "1,210,000,000", "Latitude": "23.0", "Longitude": "77.0", "Density": "11,050", "city \ Urban area": "Delhi"}, {"Land area": "46", "Country": "China", "Population": "10,000,000", "Latitude": "35.8", "Longitude": "104.2", "Density": "13,400", "city \ Urban area": "Beijing"}, {"Land area": "42", "Country": "Germany", "Population": "7,350,000", "Latitude": "48.2", "Longitude": "10.0", "Density": "2,800", "city \ Urban area": "Berlin"}, {"Land area": "648", "Country": "Iraq", "Population": "30,000,000", "Latitude": "33.0", "Longitude": "44.0", "Density": "1,000", "city \ Urban area": "Baghdad"}, {"Land area": "945", "Country": "Chile", "Population": "17,000,000", "Latitude": "-33.0", "Longitude": "-70.0", "Density": "5,425,000", "city \ Urban area": "Santiago"}, {"Land area": "945", "Country": "Spain", "Population": "46,000,000", "Latitude": "40.4", "Longitude": "-3.7", "Density": "1,000", "city \ Urban area": "Madrid"}, {"Land area": "583", "Country": "Brazil", "Population": "205,000,000", "Latitude": "-23.5", "Longitude": "-46.6", "Density": "3,750", "city \ Urban area": "Belo Horizonte"}, {"Land area": "1,687", "Country": "Sudan", "Population": "39,000,000", "Latitude": "15.5", "Longitude": "32.0", "Density": "3,025,000", "city \ Urban area": "Khartoum"}, {"Land area": "1,687", "Country": "Russia", "Population": "144,000,000", "Latitude": "55.8", "Longitude": "37.7", "Density": "8,050", "city \ Urban area": "Moscow"}, {"Land area": "1,687", "Country": "Nigeria", "Population": "190,000,000", "Latitude": "12.0", "Longitude": "3.0", "Density": "1,000", "city \ Urban area": "Lagos"}, {"Land area": "1,687", "Country": "Brazil", "Population": "2,650,000", "Latitude": "-23.5", "Longitude": "-44.2", "Density": "4,550", "city \ Urban area": "Recife"}, {"Land area": "1,687", "Country": "UK", "Population": "64,000,000", "Latitude": "51.5", "Longitude": "0.1", "Density": "4,000", "city \ Urban area": "London"}, {"Land area": "2,078", "Country": "USA", "Population": "328,000,000", "Latitude": "37.0", "Longitude": "-77.0", "Density": "2,245,000", "city \ Urban area": "New York"}, {"Land area": "2,078", "Country": "USA", "Population": "289,000,000", "Latitude": "34.0", "Longitude": "-118.0", "Density": "2,062,000", "city \ Urban area": "Los Angeles"}, {"Land area": "2,078", "Country": "USA", "Population": "265,000,000", "Latitude": "41.0", "Longitude": "-74.0", "Density": "1,000", "city \ Urban area": "Chicago"}, {"Land area": "2,078", "Country": "USA", "Population": "231,000,000", "Latitude": "43.0", "Longitude": "-77.0", "Density": "1,000", "city \ Urban area": "Houston"}, {"Land area": "2,078", "Country": "USA", "Population": "190,000,000", "Latitude": "37.0", "Longitude": "-122.0", "Density": "1,000", "city \ Urban area": "San Francisco"}, {"Land area": "2,078", "Country": "USA", "Population": "188,000,000", "Latitude": "41.0", "Longitude": "-71.0", "Density": "1,000", "city \ Urban area": "Boston"}, {"Land area": "2,078", "Country": "USA", "Population": "180,000,000", "Latitude": "33.0", "Longitude": "-71.0", "Density": "1,000", "city \ Urban area": "Philadelphia"}, {"Land area": "2,078", "Country": "USA", "Population": "170,000,000", "Latitude": "37.0", "Longitude": "-122.0", "Density": "1,000", "city \ Urban area": "Seattle"}, {"Land area": "2,078", "Country": "USA", "Population": "160,000,000", "Latitude": "34.0", "Longitude": "-118.0", "Density": "1,000", "city \ Urban area": "Phoenix"}, {"Land area": "2,078", "Country": "USA", "Population": "158,000,000", "Latitude": "33.0", "Longitude": "-82.0", "Density": "1,000", "city \ Urban area": "Cincinnati"}, {"Land area": "2,078", "Country": "Ghana", "Population": "27,000,000", "Latitude": "5.8", "Longitude": "0.0", "Density": "453", "city \ Urban area": "Accra"}]}
```

02

PENTaho DATA INTEGRATION MERGING TABLES

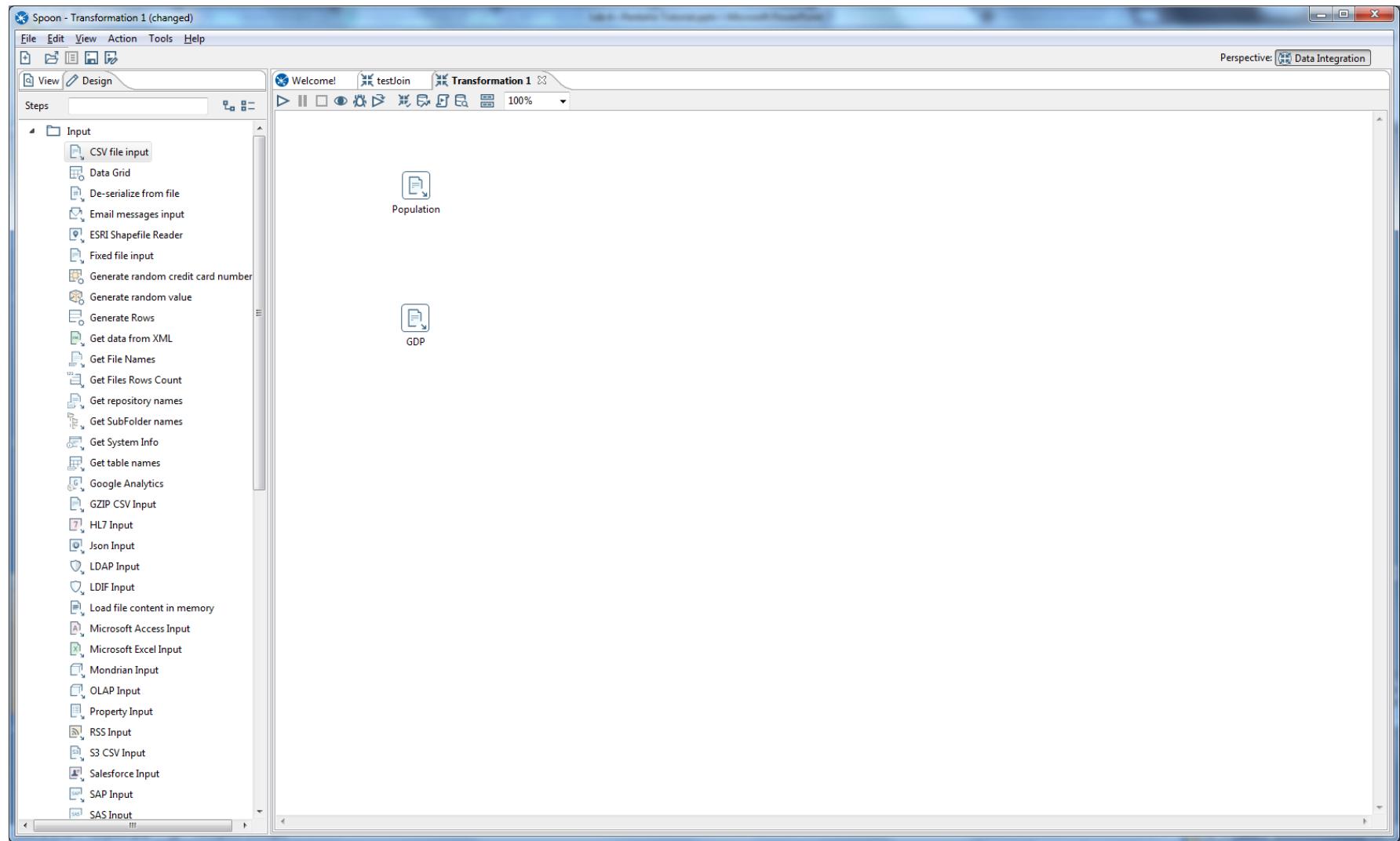
Now let's use both files as input.

Input > CSV File input > drag & drop
(both files)



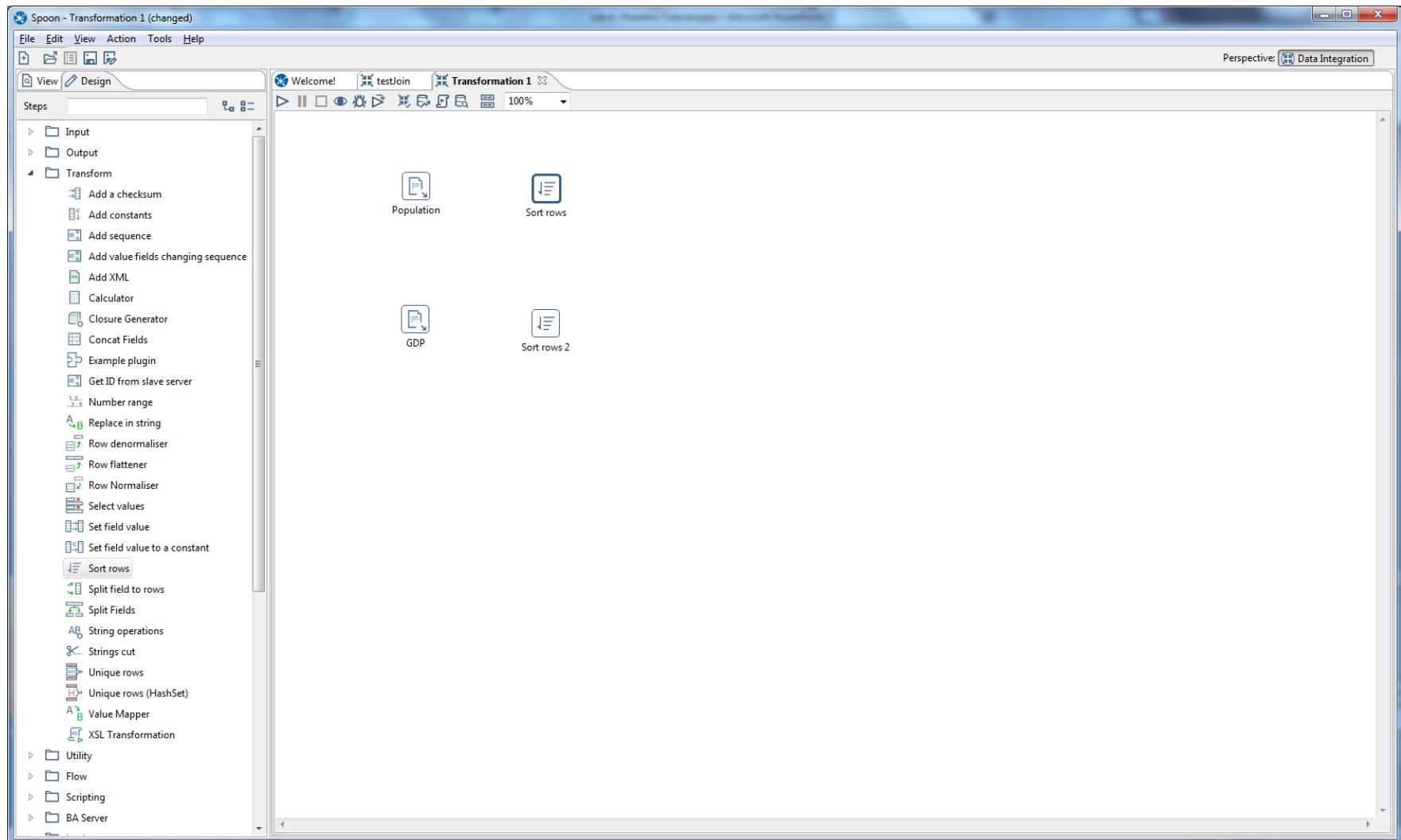
Attention:

Two input files.



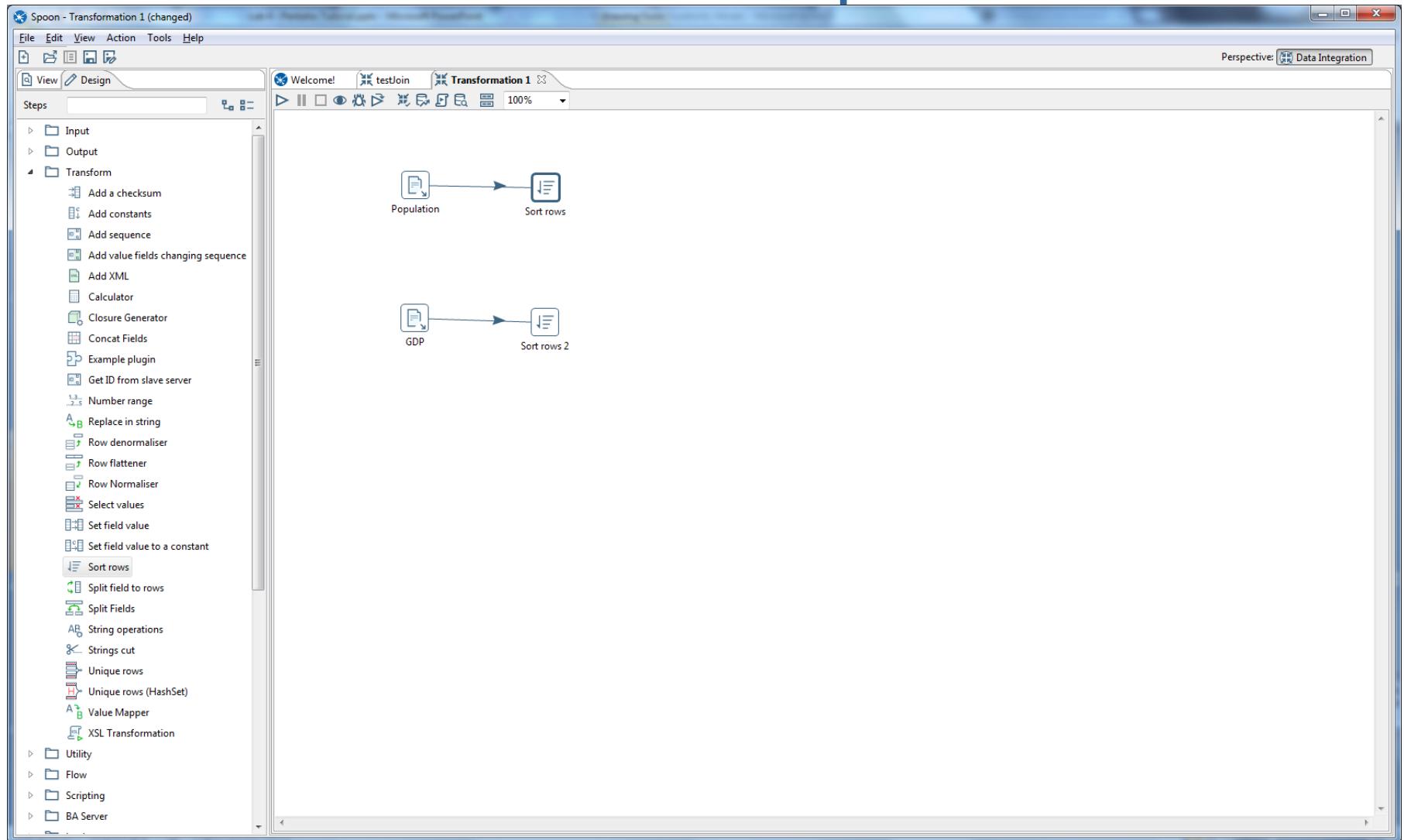
Now let's sort our data by country

Transform > sort rows > drag & drop



Now let's sort our data by country

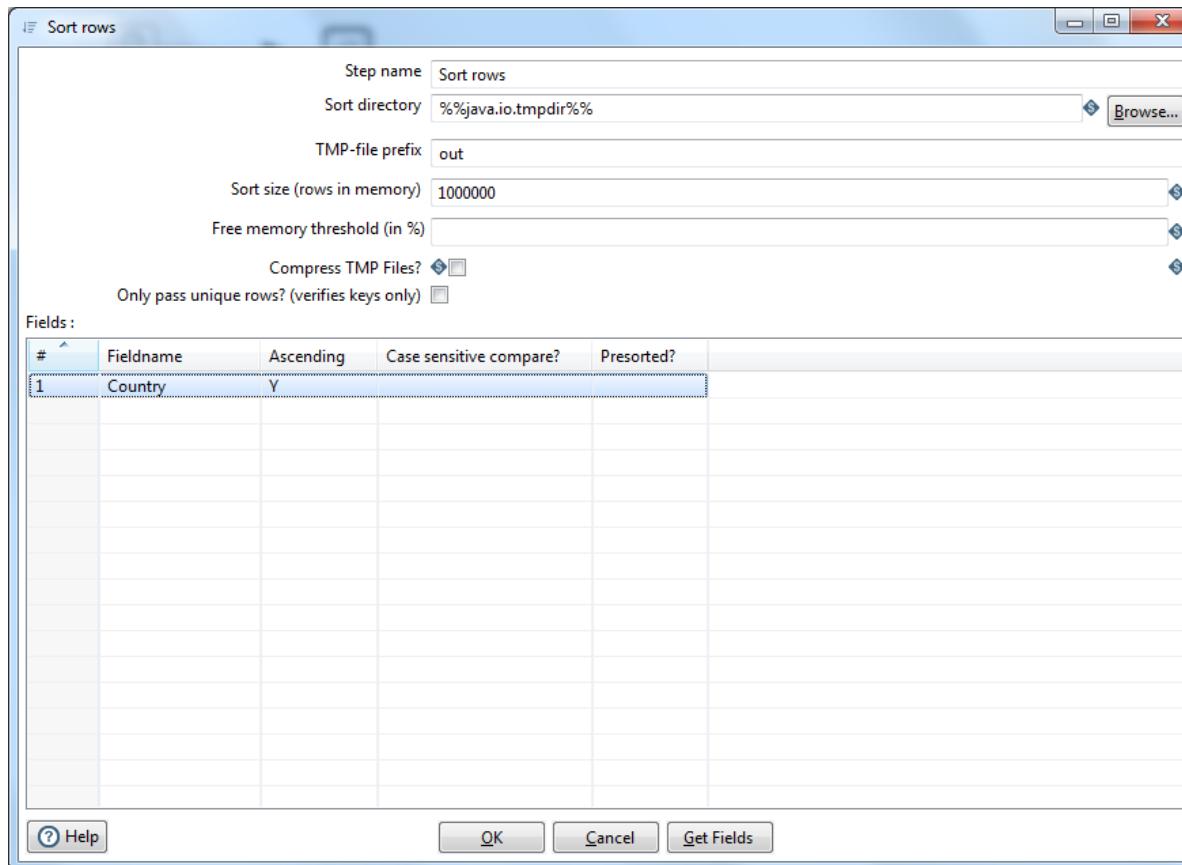
Create hops



Now let's sort our data by country

Configure sort steps (population):

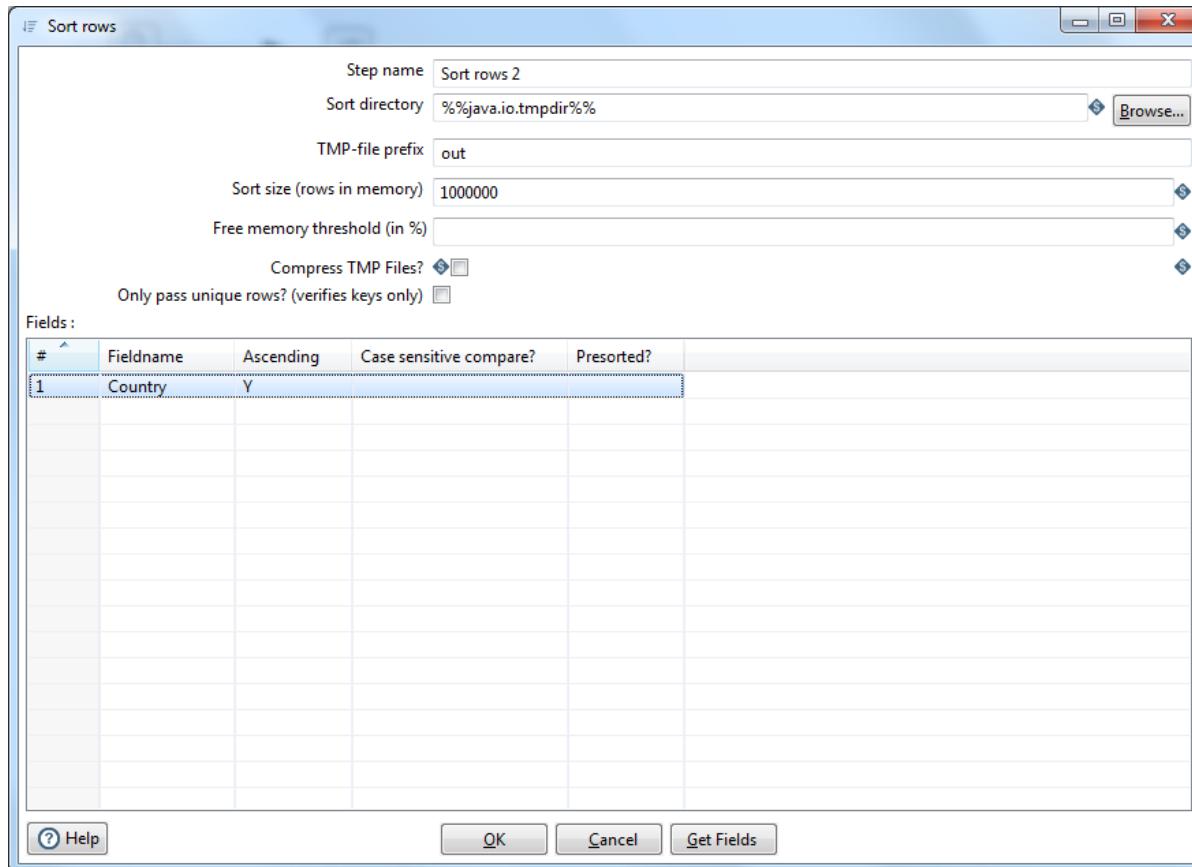
Double click > get fields > delete all but “country”



Now let's sort our data by country

Configure sort steps (GDP):

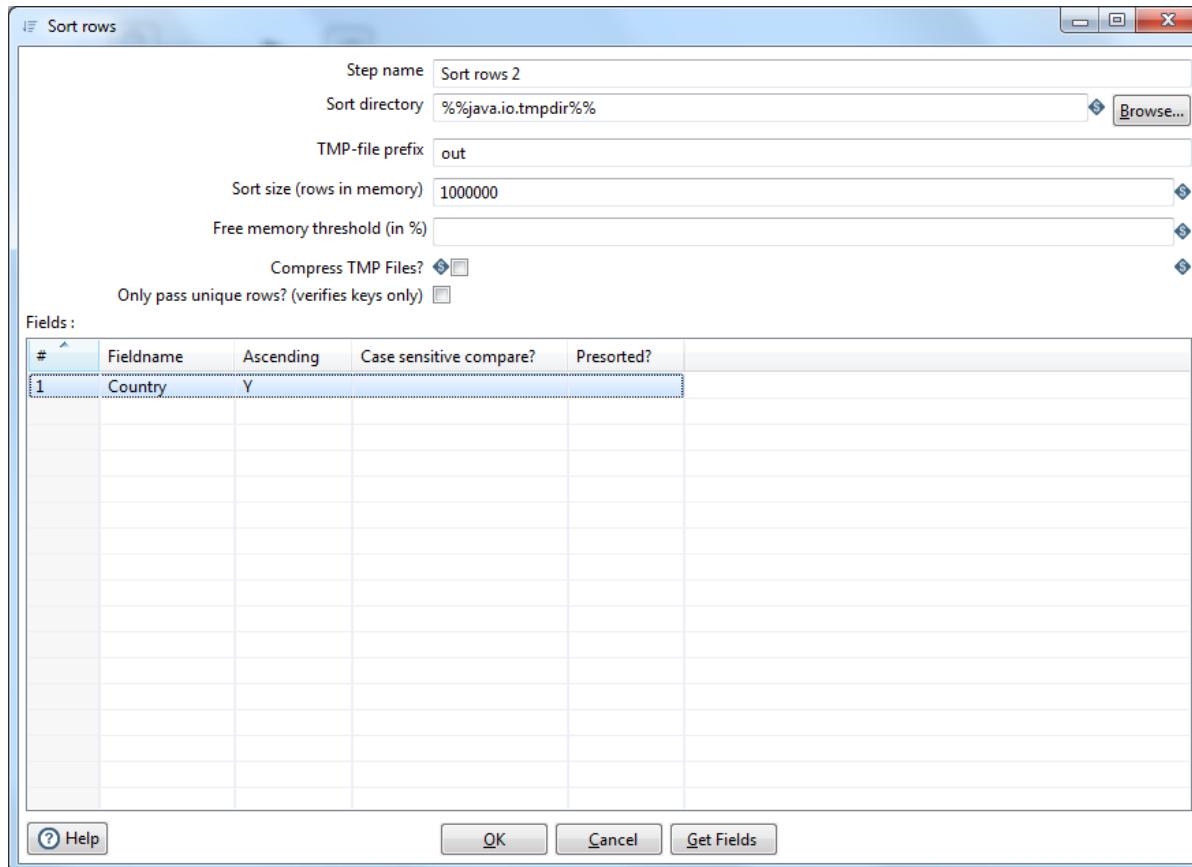
Double click > get fields > delete all but “country”



Now let's sort our data by country

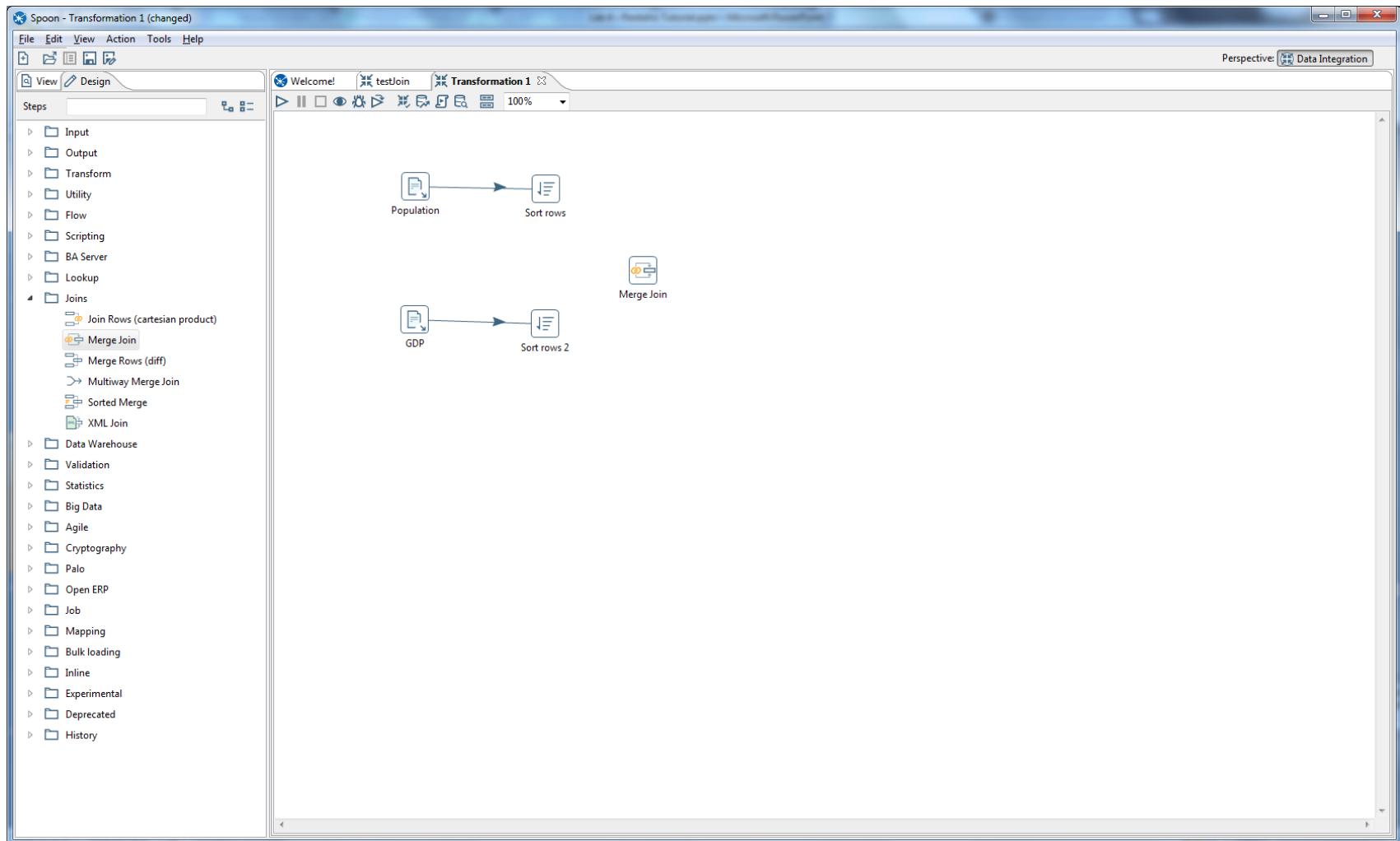
Configure sort steps (GDP):

Double click > get fields > delete all but “country”



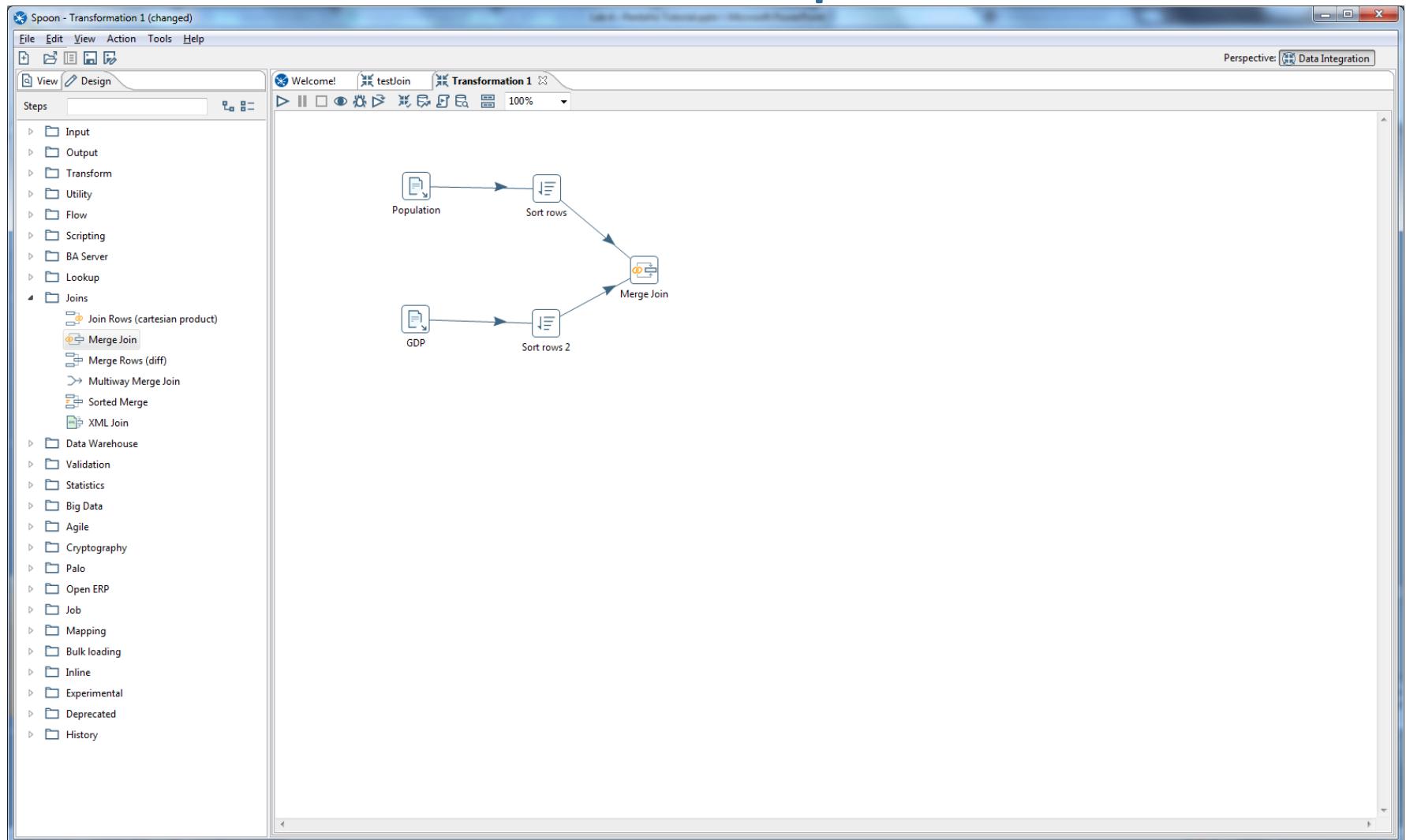
Merging by country...

Joins > Merge Join > drag&drop



Merging by country...

Create hops...

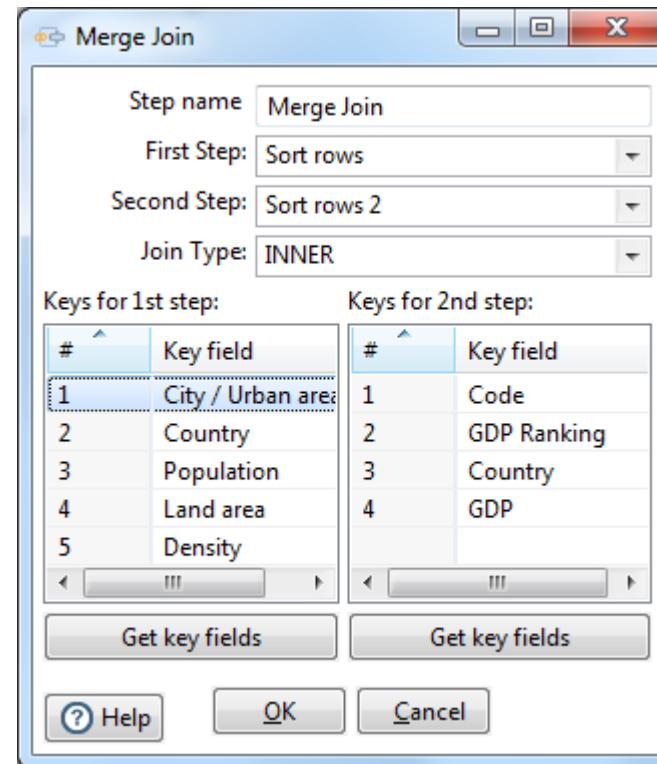


Merging by country...

Configure Merge Join step:

Double click > configure first & second step

Get key fields for both

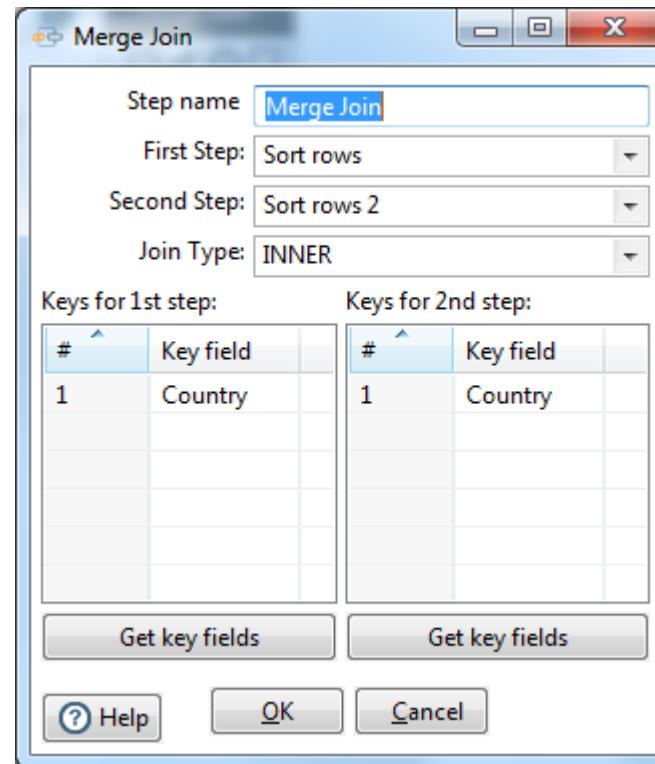


Merging by country...

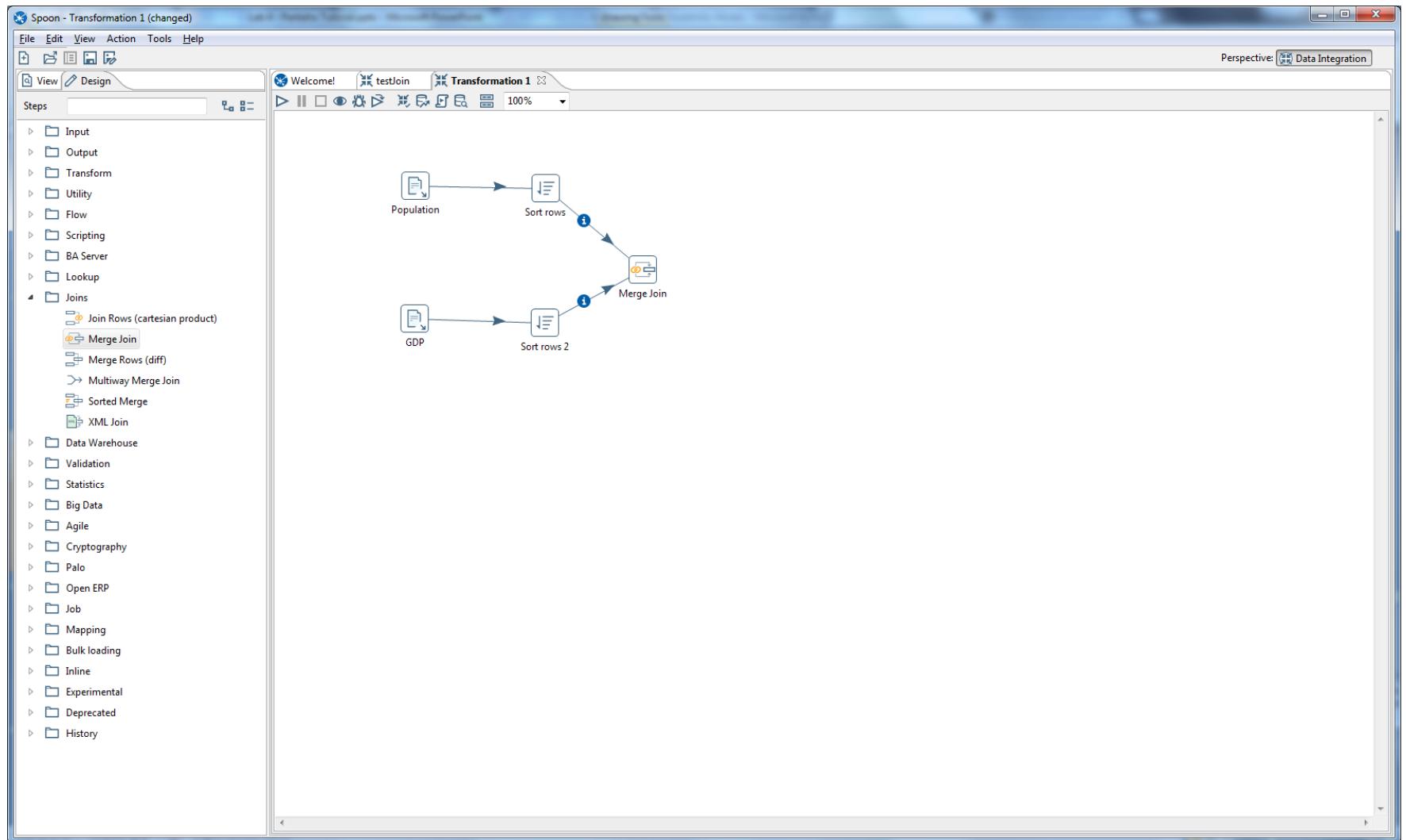
Configure Merge Join step:

Remove all keys except “Country”

Ok!



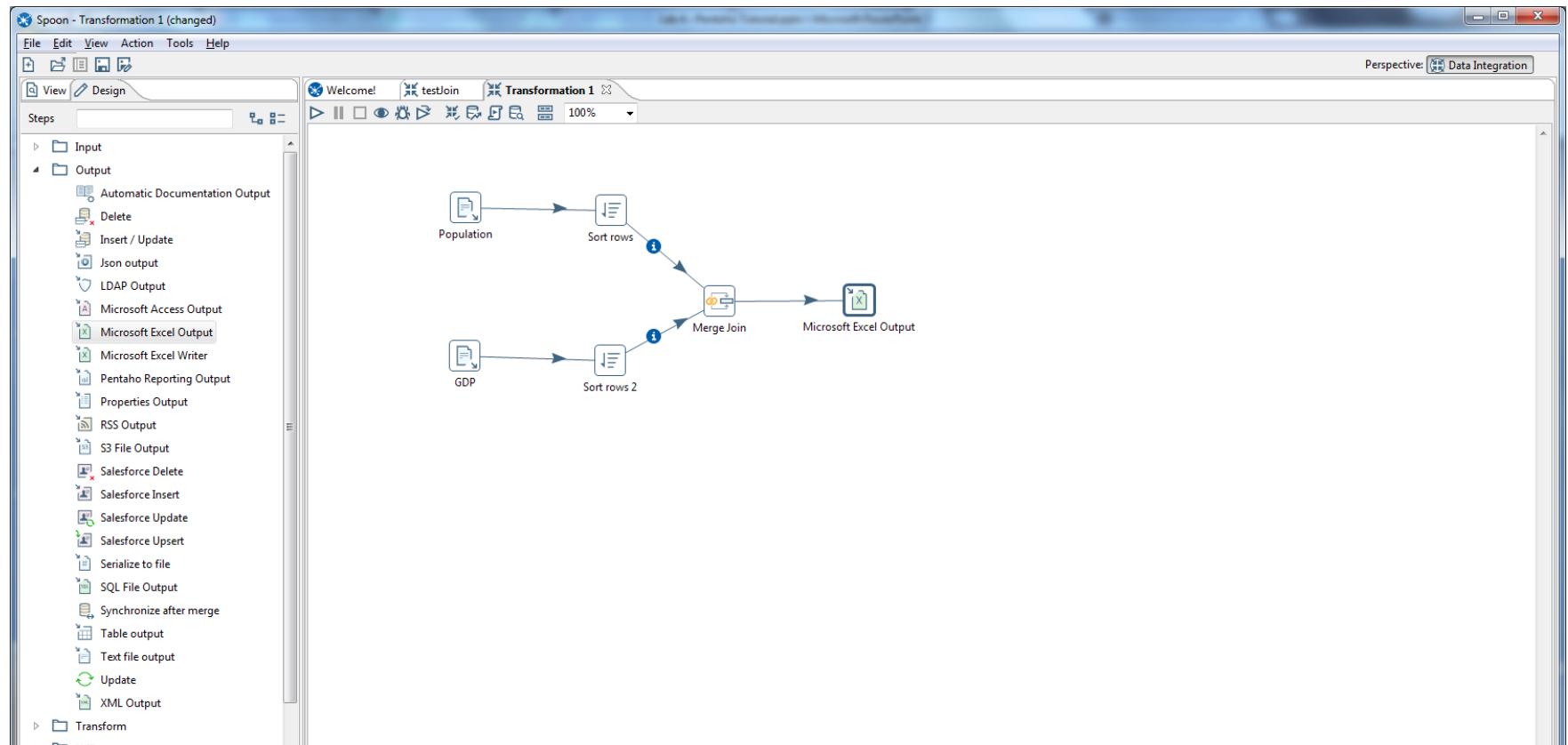
Almost there...



Let's create an Excel file to test

Output > Microsoft Excel Output > drag&drop

Create hop

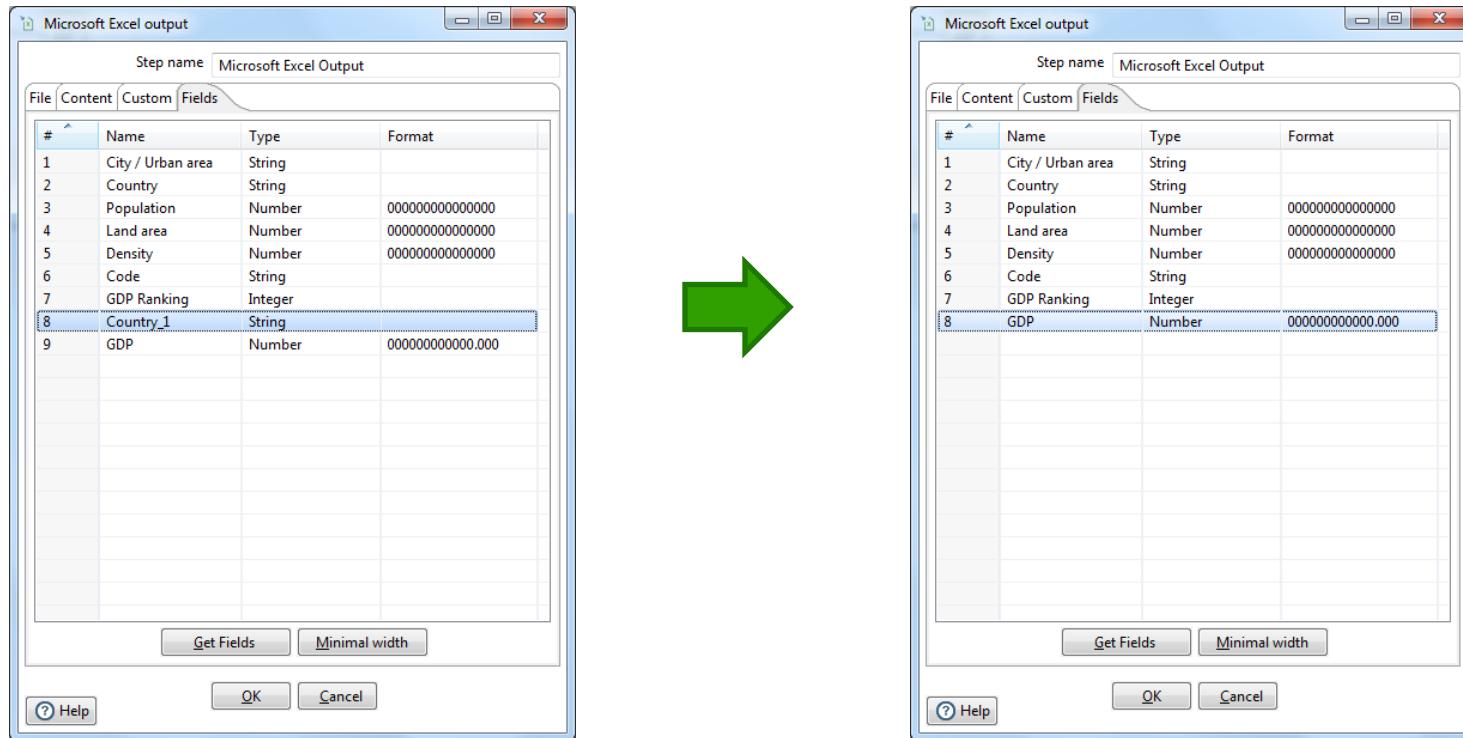


Let's create an Excel file to test

Configure Excel output:

Double click > configure filename, open “Fields” tab

Delete country_1 (the repeated field)



Let's create an Excel file to test

Run > Launch (save)

Open the Excel file:

03

PENTaho DATA INTEGRATION: EXERCISE: FILTERING

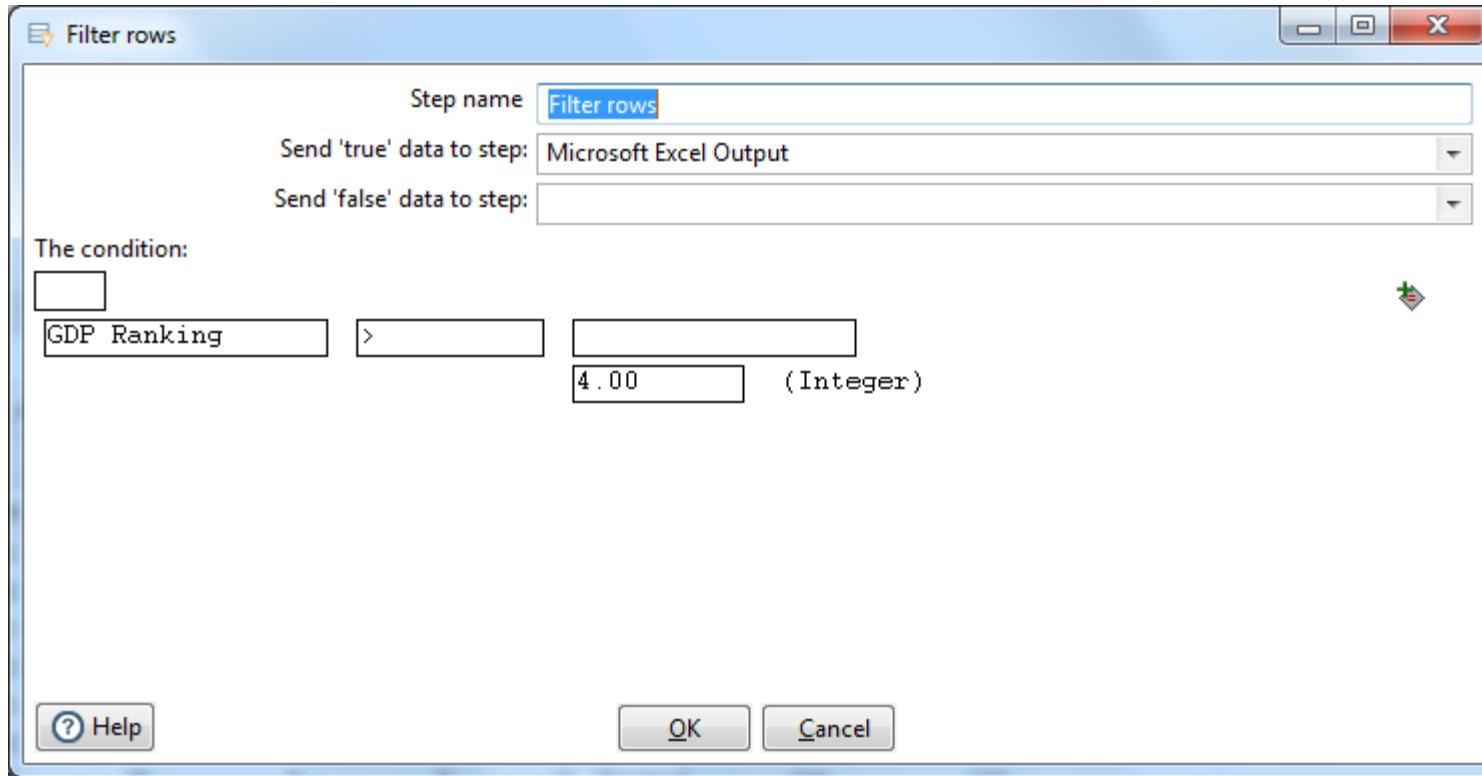
Remove the entries with GDP Ranking ≤ 4

Flow > Filter rows

Configure options

Don't forget to keep hops functional!

Remove the entries with GDP Ranking <= 4



04

NEXT LAB

Data > Information

Parse your original (raw) **data** into

INFORMATION

to use as input for your
visualization

QUESTIONS?