Data Visualization

Context

Data visualization is viewed by many disciplines as a modern equivalent of visual communication. It involves the creation and study of the visual representation of data, meaning «information that has been abstracted in some schematic form, including attributes or variables for the units of information». It is one of the steps in data analysis or data science.
The primary goal of data visualization is to communicate information clearly and efficiently via statistical graphics, plots and information graphics. Effective visualization helps users analyse and reason about data and evidence. It makes complex data more accessible, understandable and usable.
Data visualization is strategic. All its stake is to help to decision making because the future of company is on a line : data which would not be right could lead to bad decision.

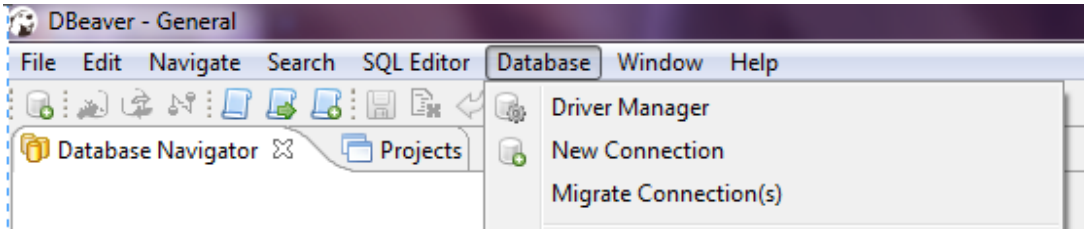
In our projet, the data visualization is very important because it can help clients to visualize their electrical consumption over the year of their home.

DBeaver and Tableau
Why
Tableau is a data visualization tool. Like we said, data visualization is very important nowadays so we decided to choose Tableau which is one of the more famous dataviz's tool.
With Tableau, The user is not dependent anymore. He can create himself his analyses and his displays to publish them then and spread them.

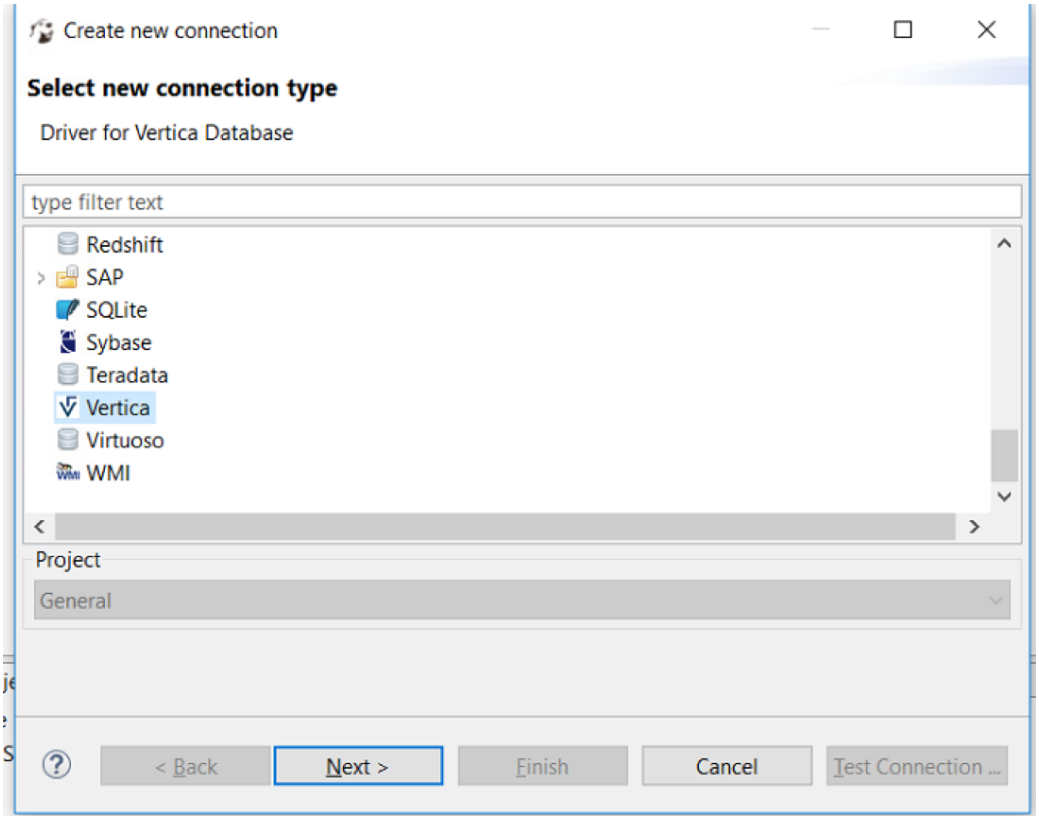
The DBeaver is an SQL client and a database administration tool that we used to connect to vertica.

To connect DBeaver to Vertica, you need to follow these:

In DBeaver, select Database > New Connection



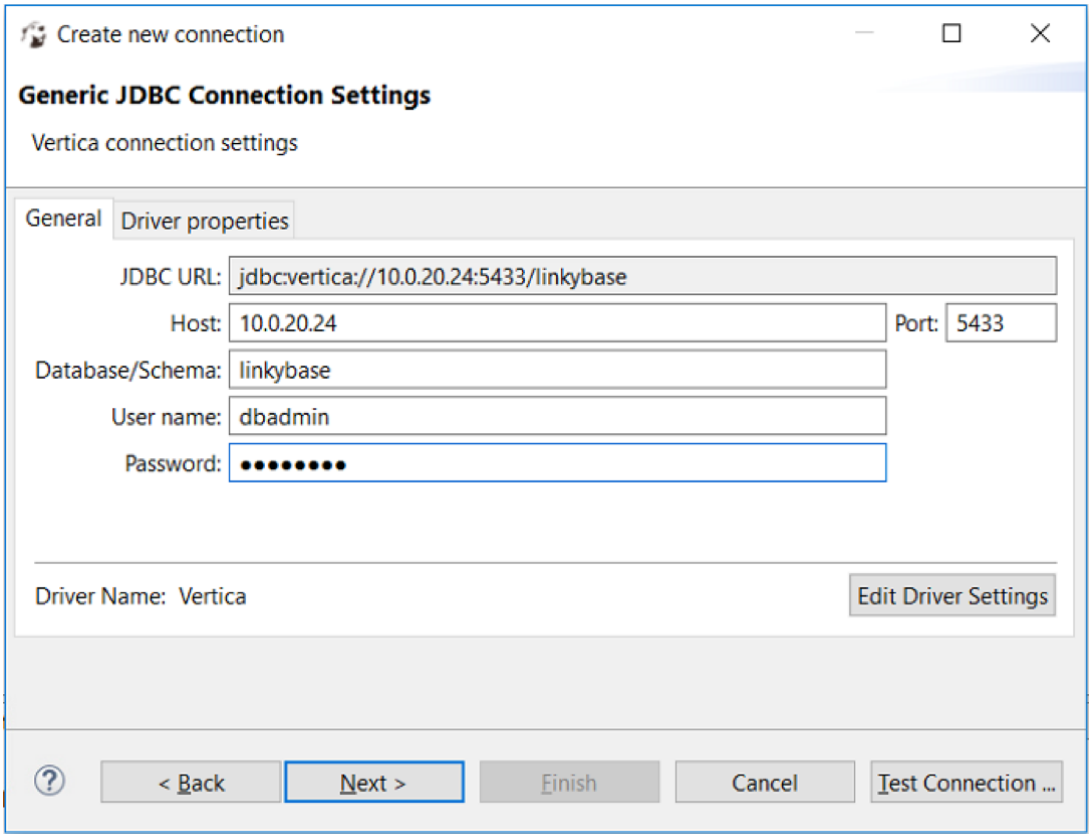
In the Create New Connection window, select Vertica and click Next.



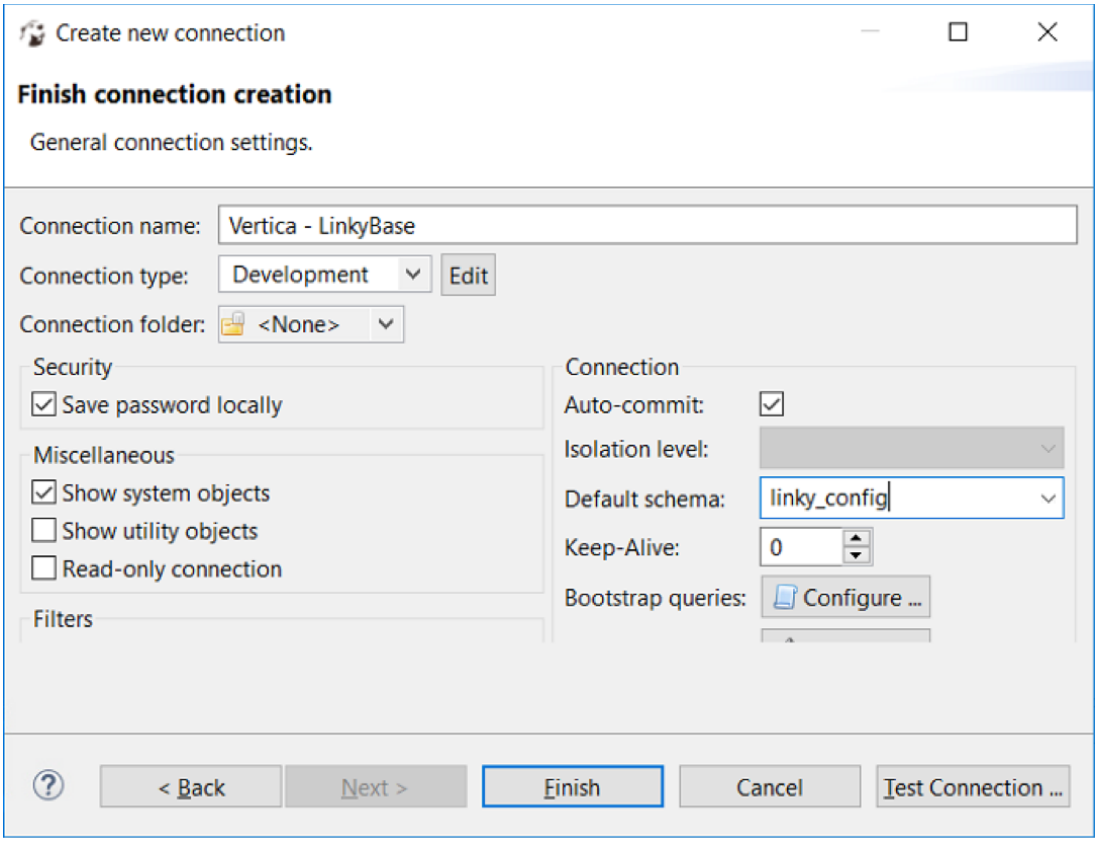
In the general tab, enter you database credentials.

Then, in Edit Driver Settings, select add file and select the Vertica JDBC file you downloaded.

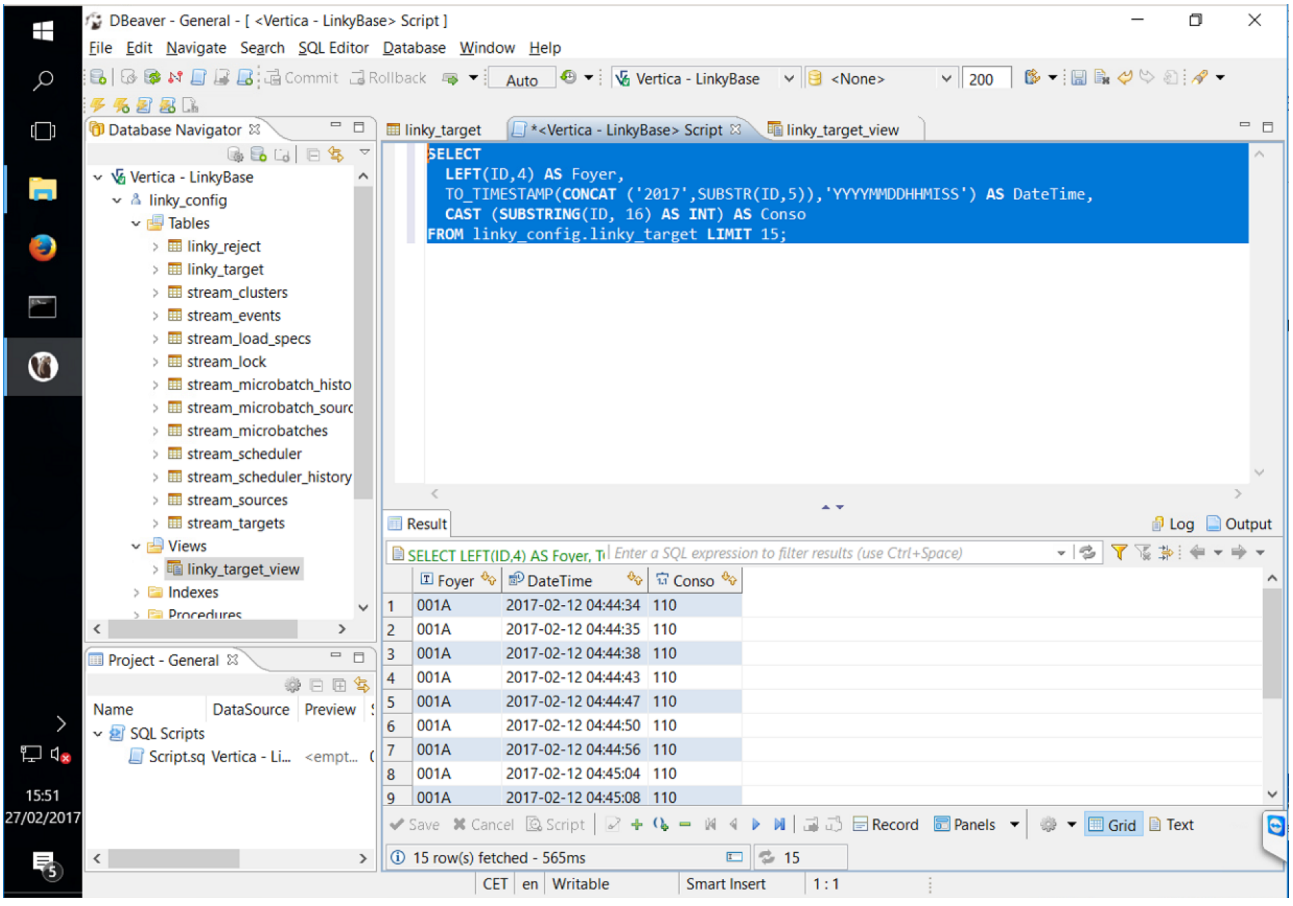
You can check if your connector is working by clicking Test Connection. If your connection to Vertica is successful, a message appears. Click on Next button next.

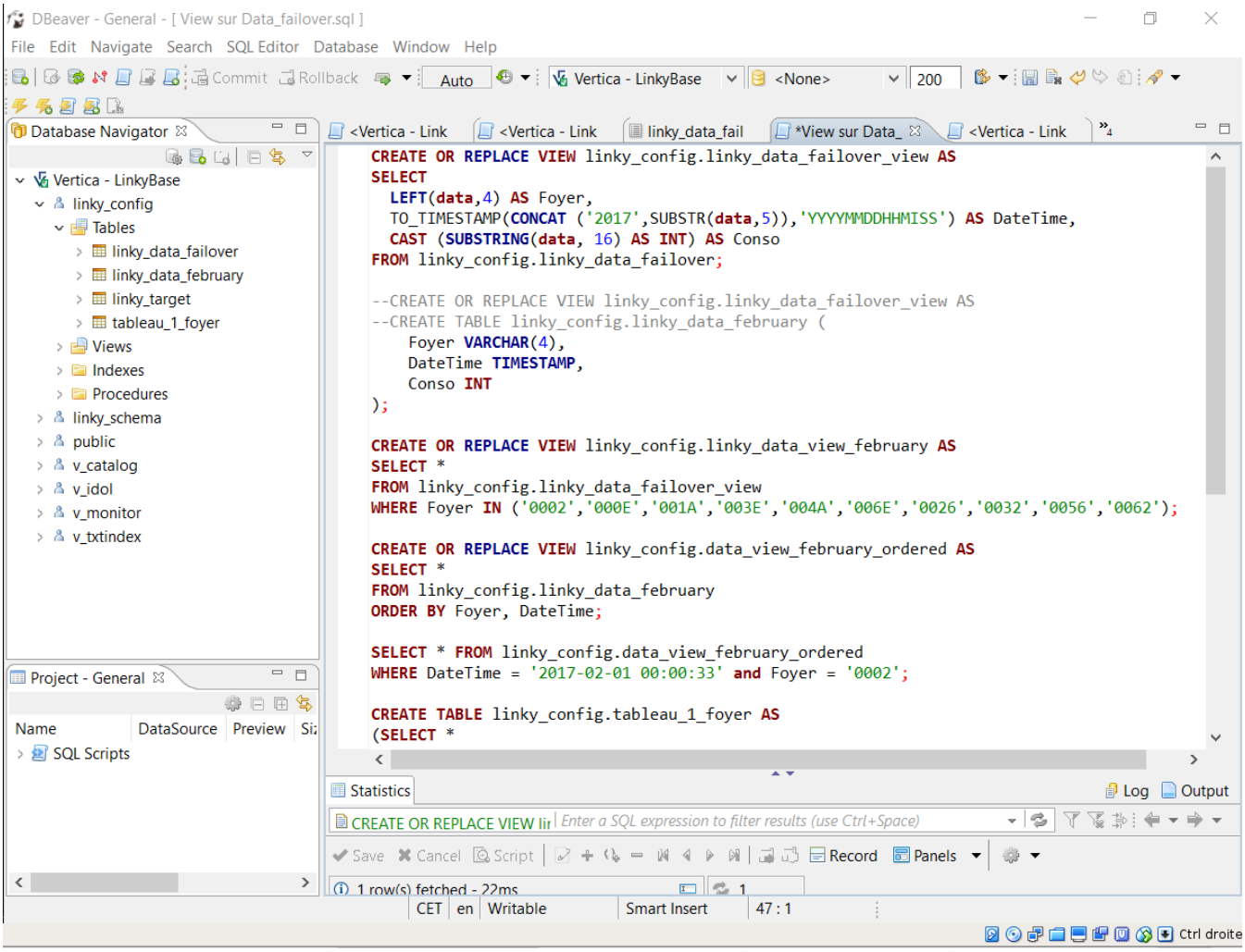


In Default schema, select linky_target because data are in linky_target.



From here, you can run queries and visually explore your Vertica database. When you are satisfy of your queries, you can create a view which can be used in Tableau.





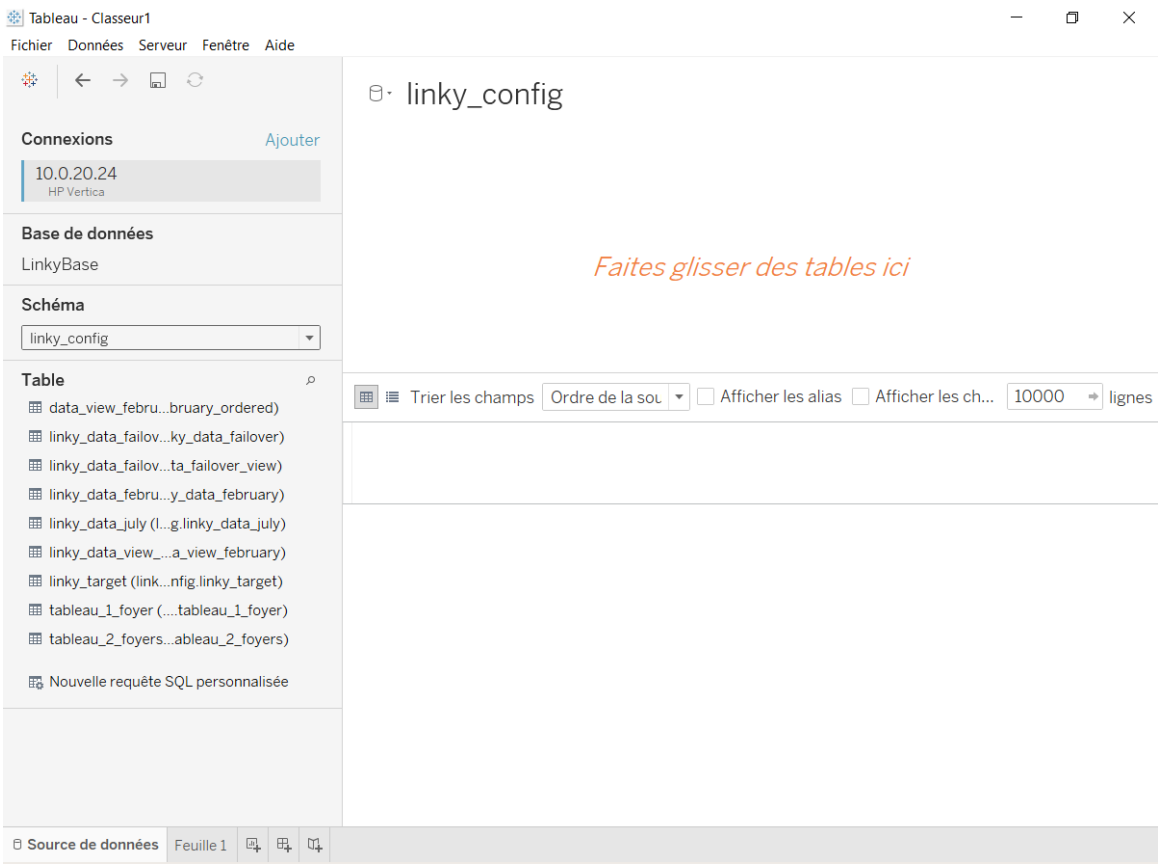
We want to show you how to use Tableau now that we made our queries on DBeaver.

Tableau uses ODBC to connect to Vertica. The ODBC drivers for Vertica are part of a Vertica client package.
You need to download and install the ODBC driver on Windows.

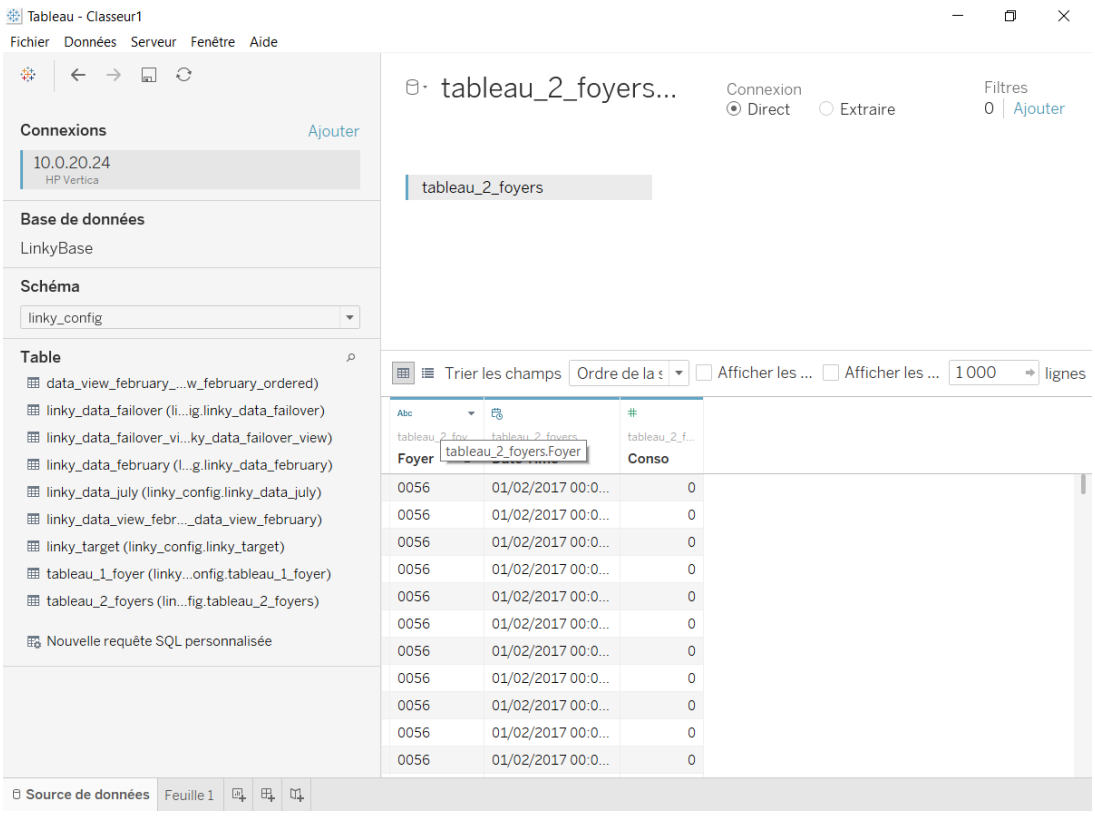
This is the home page when you open Tableau.

On the left, click on HP Vertica.
As you can see, we connect on the server 4 to access Vertica (cf Architecture part)

This is the page you see once you are connected. You just have to drag and drop one table at the indicated position



Once you drag and drop your table, you can explore the data as you can see on the next picture. In our case, we wanted to visualize the home's id, the date and the home's consumption



For visualizing your data, click on Feuille 1 at the bottom of Tableau. You can now see the following page.

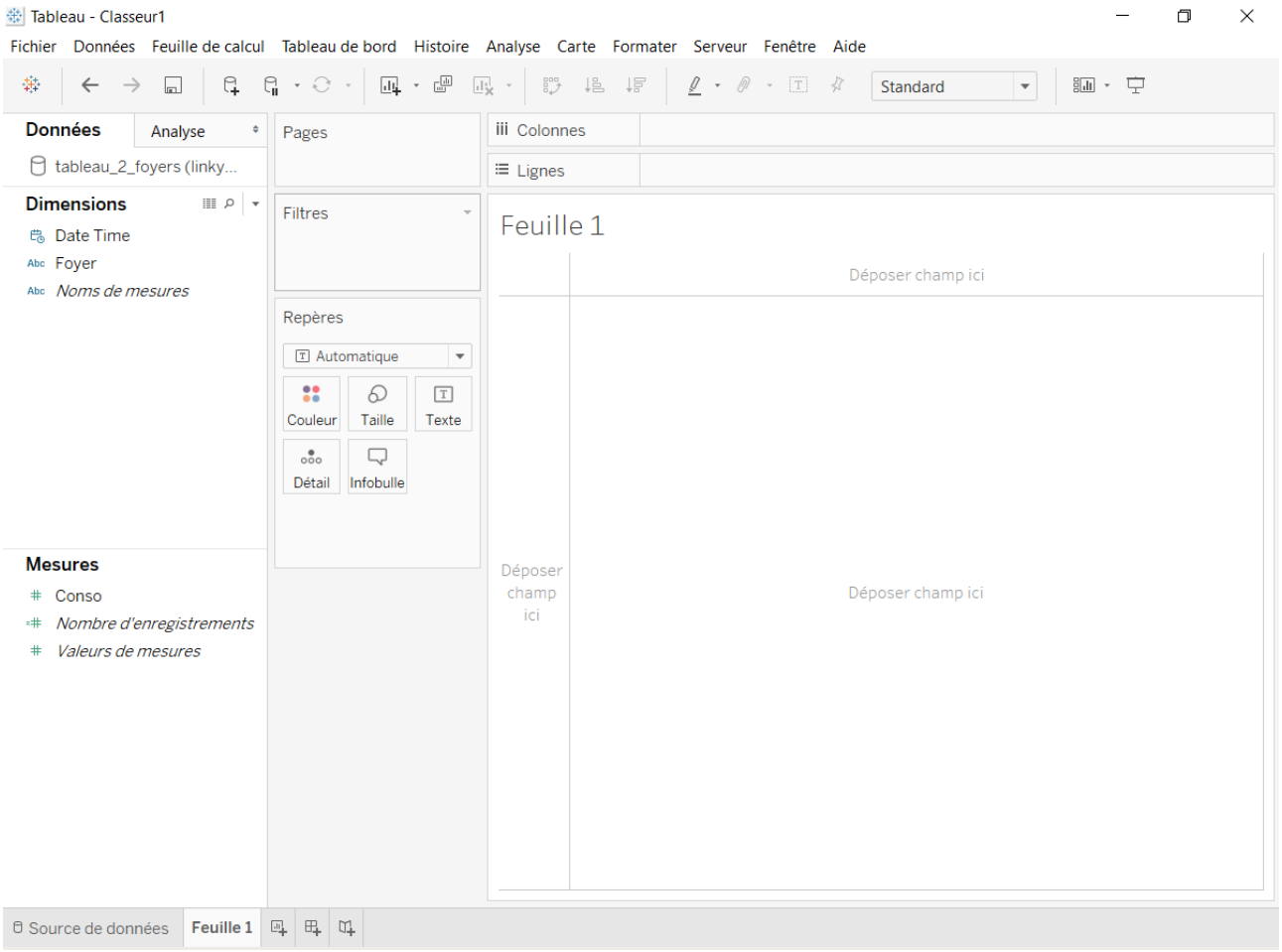
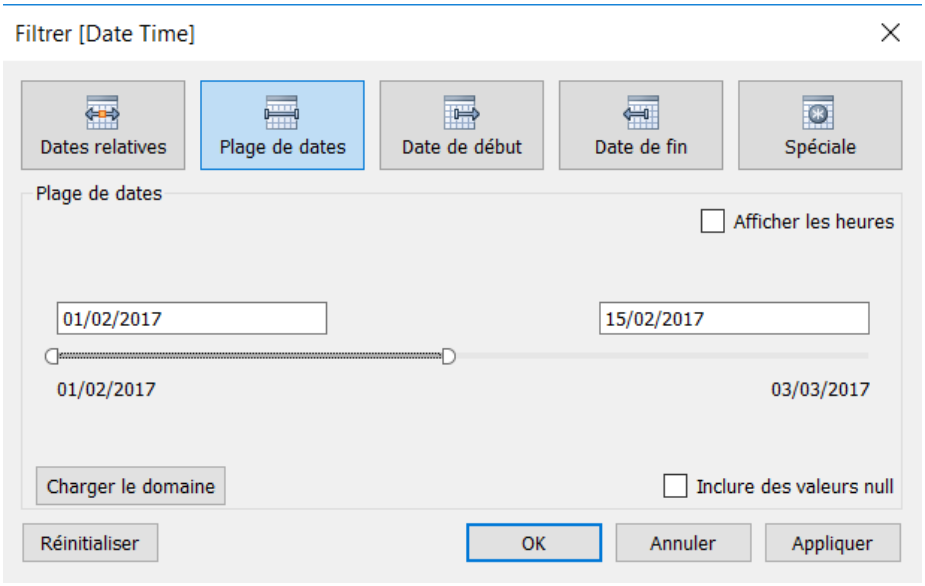
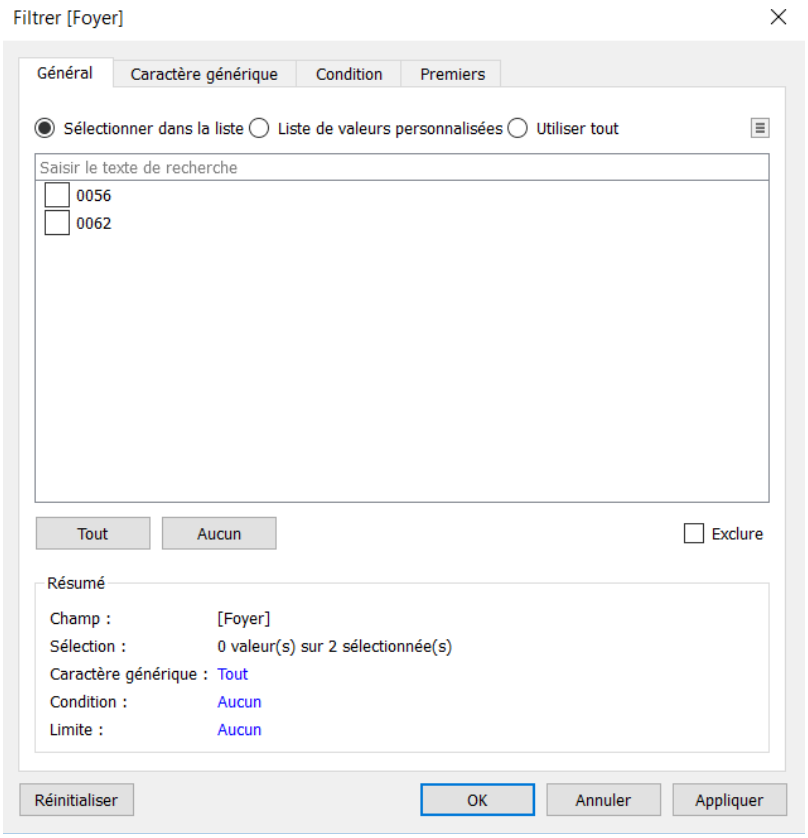


Tableau is really simple and easy to use. You can drag and drop what you need to see into « colonnes » and « Lignes ».
You can also apply filters also by dragging and dropping into the « Filtre » section.

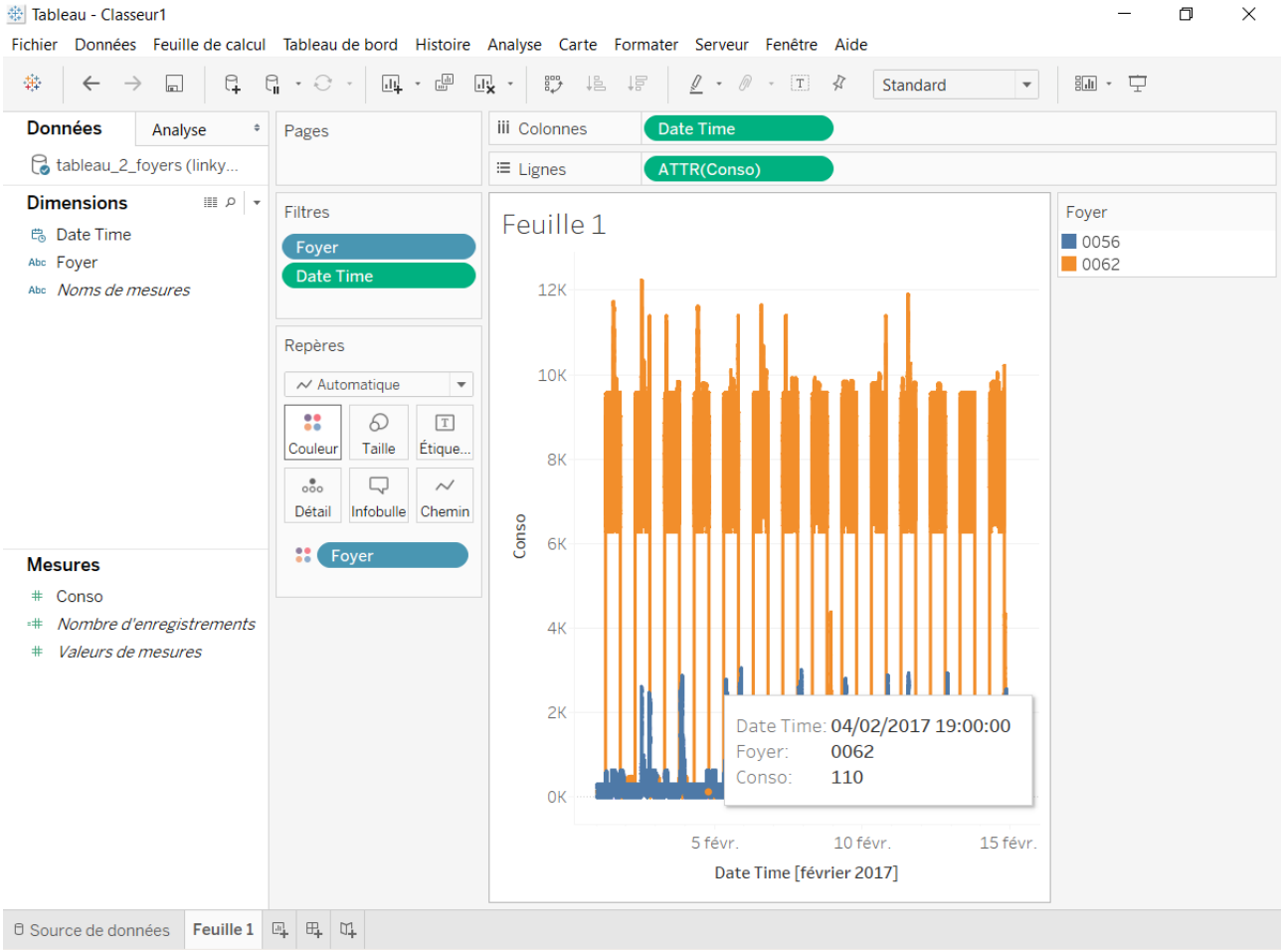
For example, we apply a filter on the date. We only want to select only the two first weeks of February.



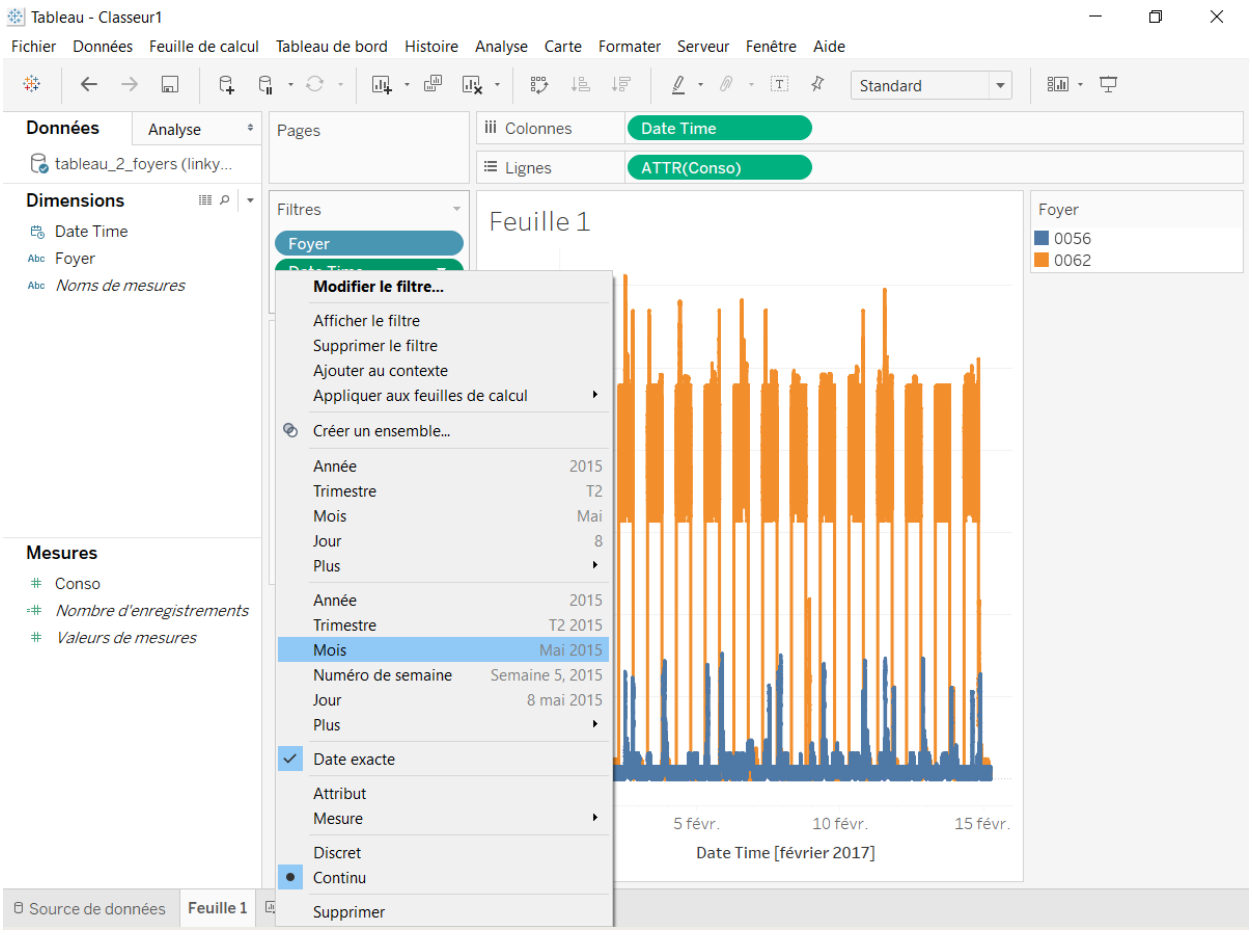
We can also apply a filter on the home to select.



Here is the result of all the filter we apply on our data. you can see the electrical consumption of the two home we selected in our filter at the date we chose.
By default, the colors of the two home is identical but we can also apply a filter on the colors.

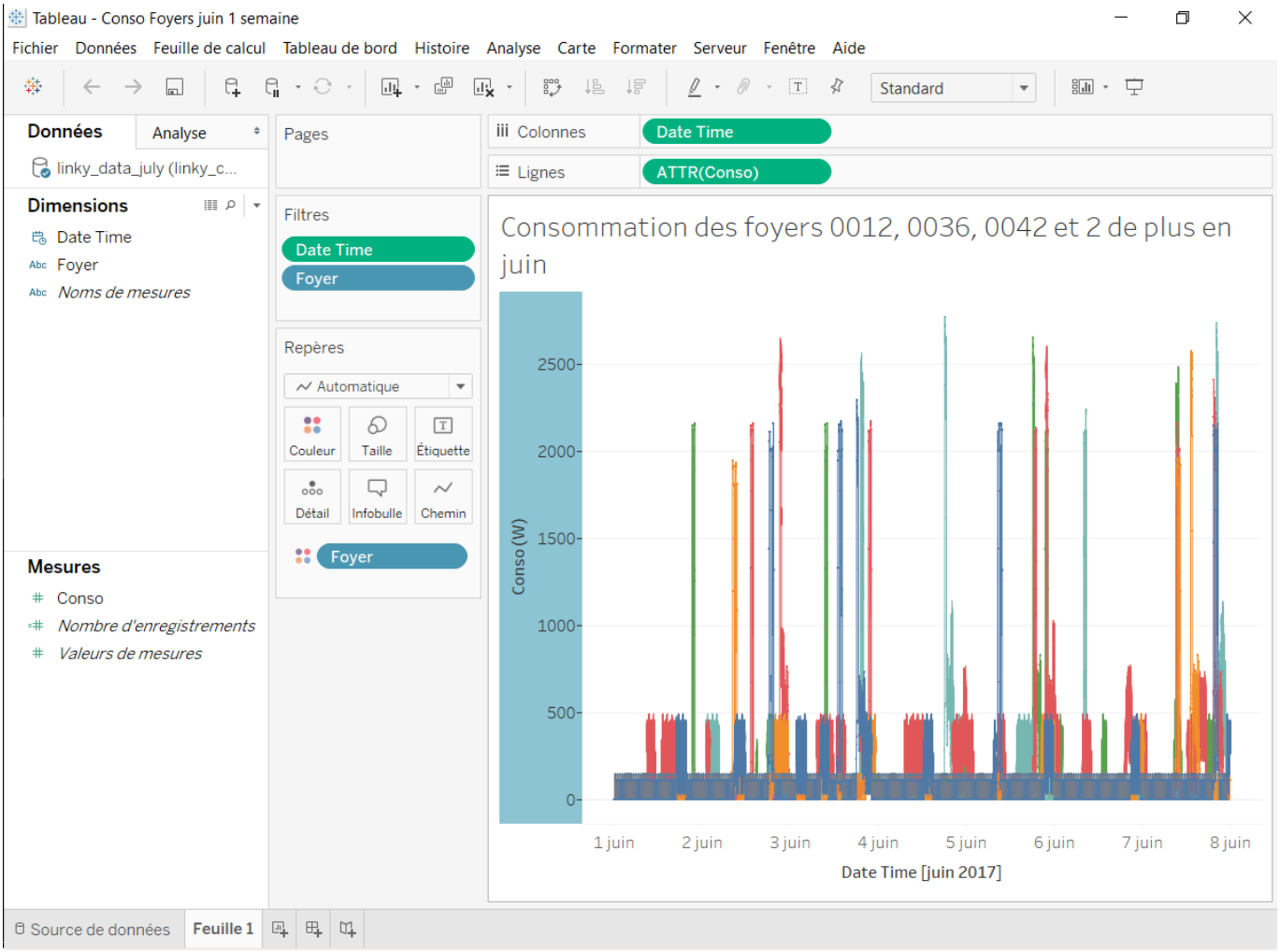


You can modify your filters whenever you want by clicking on the filter you want to change.



This is an other example of visualization of our data.

You can see the electrical consumption of several home during one week in June.



HPE asked us to realize a proof of project for EDF which is installing new intelligent electric meter in all the country in order to recover the data without sending a person retrieve them.
The data are stored on servers. What EDF wants to do is to analyse all the data to be able to do prediction for the future and also proposing a consumption's visualization to their client.
EDF aim to be able to suggest their clients a solution to reduce their invoice.
As you can see, we have a pretty good visualization of the data of several home for a week and we think that these results are satisfying and can be showed to client asking about their home's consumption.

Amelioration

In order to improve our project in the futur, we think that use Spark will allow to improve our performance.

Apache Spark is an open source cluster-computing framework. It provides an interface for programming entire clusters with implicit data parallelism and fault-tolerance.

Spark realizes a reading of the data at the level of the cluster, makes all the necessary operations for analysis, then writes the results at the same level. In spite of the fact that he spells with the languages Scala, Java and Python, it makes best use its capacities with his native language, Scala.

Therefore, where MapReduce of Hadoop works by stage, Spark can work on all of the data at the same time. Spark is thus until ten times faster for the treatment(processing) and until hundred times faster to make the analysis in memory.

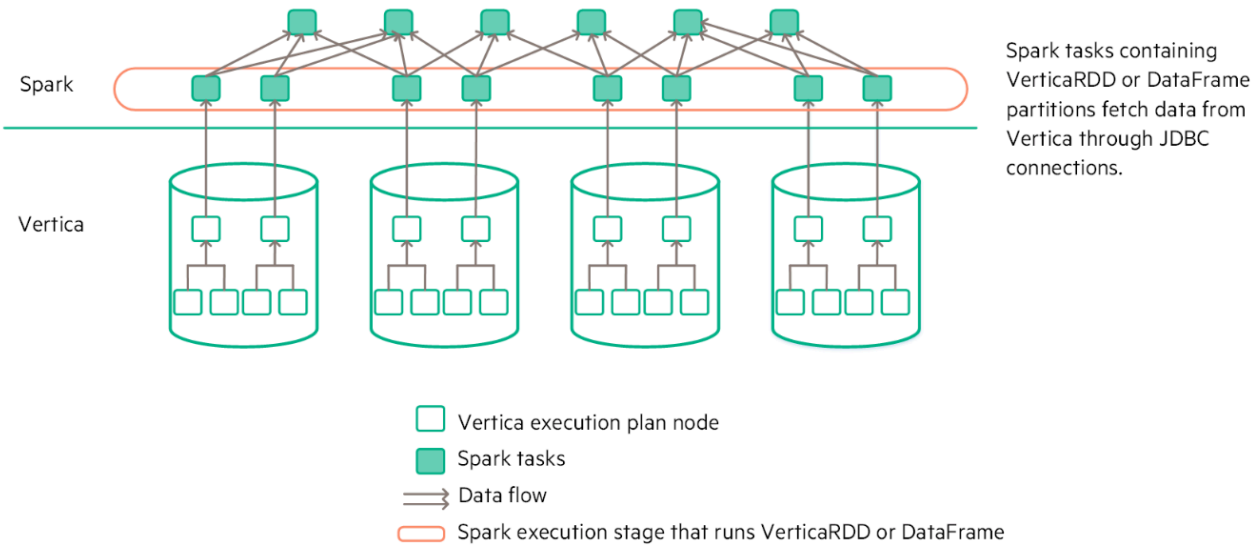
Spark executes all the operations of data analysis in memory and in real time. He does lean on records only when his memory is not sufficient anymore. On the contrary, with Hadoop the data are written on the record after each of the operations. This work in memory allows to reduce the latent periods between treatments, which explains such a speed.

However, Spark does not arrange a management system of file. It is necessary to supply with one (examples: Hadoop Distributed File System/Informix/Cassandra/OpenStack Swift/Amazon S313). It is advised to use him with Hadoop who remains at present the best global solution of storage thanks to its tools of administration, safety and monitoring more advanced.

In case of breakdown or of failure of the system: the objects of data are stored in what we call resilient distributed datasets (RDD) distribute on the cluster of data allowing the complete recovery of data.

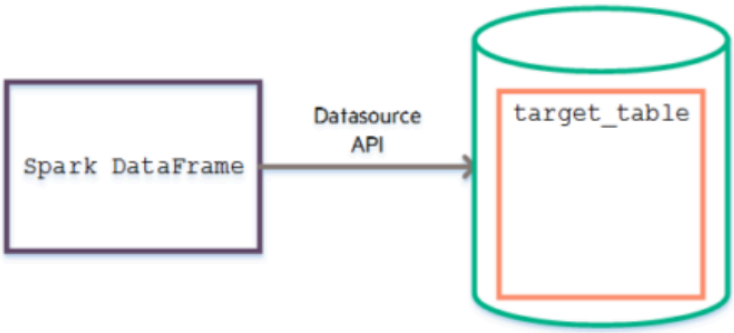
A RDD is a collection of data calculated from a source and preserved in memory lively (as long as the capacity allows it). One of the advantages brought by RDDs is in his capacity to keep enough information on the way a partition RDD was produced. In case of loss of a partition he is thus capable of recalculating it.

How Vertica and Spark work together:



Using the HPE Vertica Connector for Apache Spark, you can:

- Move large volumes of data from Spark DataFrames to Vertica tables using parallel read and write from HDFS.



- Save Spark data to Vertica with the DefaultSource API.
- Move data from Vertica to a Spark RDD or DataFrame.