Practical Work 4 - Apache Superset

Prof. Dr. Laura E. Raileanu, Cédric Campos Carvalho, Elliot Ganty

17 novembre 2025

1 Introduction

1.1 Apache Superset

Apache Superset is an open-source data visualization platform that helps transform data into actionable insights. It is a powerful tool that can be used by Data Scientists or other business users to extract information about the company.

According to Inmon, Levins et Srivastava [1], data visualization is a process that consists of representing data in a graphical or visual manner. This makes it easier to understand the data and to identify structures and trends that would be difficult to see in raw data.

Apache Superset offers a wide range of visualization options, including charts, tables, maps, and dashboards. It also supports a variety of data sources, including relational databases, NoSQL databases, and cloud storage.

Data visualization is a powerful tool that can be used for many applications, including :

- Communicating results : Data visualization can help communicate the results of data analyses in a clear and concise way.
- Decision-making : Data visualization can help make better decisions by providing a better understanding of the data.
- Identifying new opportunities : Data visualization can help identify new opportunities and trends that might not otherwise be visible.

1.2 Data

The data comes from the AdventureWorks dataset. It comes from a sample provided by Microsoft as an example for its SQL products. We adapted and simplified this data so that it can be used as part of this lab. This dataset represents around 120,000 sales of bicycle items, both online and in stores, across several countries.

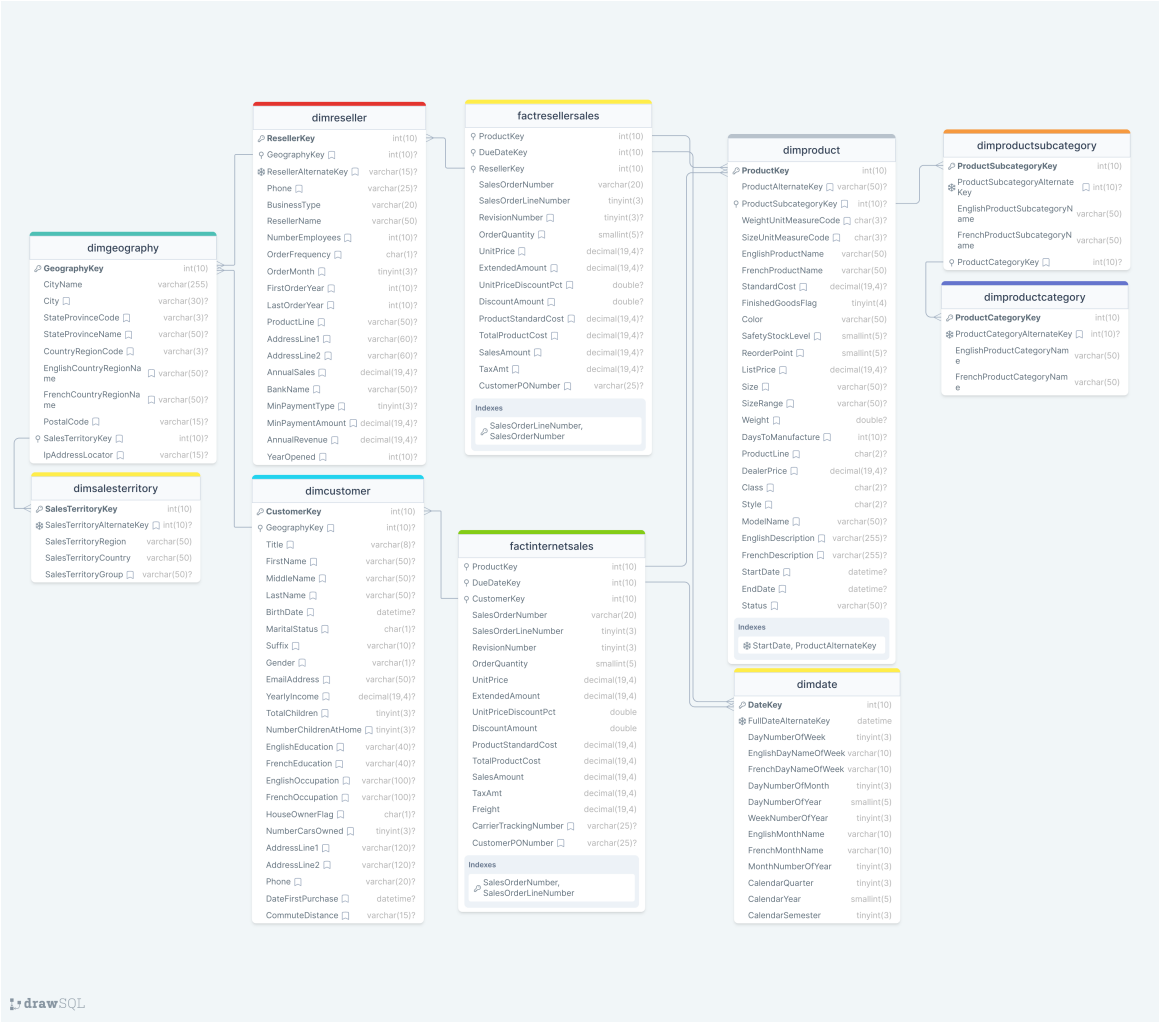
1.3 Submission

The submission is done on the Moodle page until **3 décembre 2025 à 23:59:59**, where you will provide a report containing the answers to the questions (including images) in **PDF** format, as well as the export of the *Apache Superset Dashboard* containing the various exercise visualizations in **ZIP** format.

2 Setup

Apache Superset offers several ways to install it (official page), for this lab, it is recommended to use the Docker method described in section 2.1. If you encounter many issues during the installation with Docker, other methods are available through the official documentation.

Figure 1 – Relationship diagram Adventure Works simplified.



2.1 Installation with Docker

For this lab, several files are provided to set up the Docker container containing all the necessary information, including the database for the lab. Once placed in the folder containing the installation files, simply launch Docker with the command :

```
1 docker compose up
```

The username is `admin` and the password is also `admin`.

2.2 Connection to the database

Once launched, *Apache Superset* is accessible via a browser on the port defined beforehand. After logging in, it contains several tabs, including :

- **Dashboards** : A list of dashboards, which are interactive data visualizations
- **Charts** : A list of charts created from datasets for the dashboards
- **Datasets** : A list of datasets, which are collections of raw data formed via SQL queries
- **SQL** : Provides an interface to execute SQL queries
- **Settings** : The settings that can be used to customize its operation, including database connections.

If needed, to add a connection to a database, go to the settings and select "*Database Connections*". First, you must make the downloaded "*aventures.sqlite3*" file accessible at a given path, such as "`/home/username/.superset`". In the top right corner, click "*+Database*" and select "*SQLite*". Then choose a name (Aventures) and enter the URI path. Following the example, the value is :

```
sqlite:////home/username/.superset/aventures.sqlite3
```

Once completed, click "*Finish*".

3 Exercises

3.1 Exercise 1

This exercise will allow you to become familiar with *Apache Superset* and will guide you step by step to obtain the desired result. The goal is to create a *Dashboard* and then display a *Chart* showing the number of French customers via an SQL query saved as a *Dataset*.

To perform the query, you can use the *SQL Lab* and then select the correct *Database* (Aventures). Now, execute the following command :

```
SELECT * FROM dimcustomer AS c INNER JOIN dimgeography AS g
ON c.GeographyKey = g.GeographyKey
WHERE g.FrenchCountryRegionName = "France"
```

This command performs a join on the `dimcustomer` and `dimgeography` tables, then filters by the country name to select customers from `France`. Feel free to look at the table schema (Figure 1 or through *SQL Lab*) to help you write the queries.

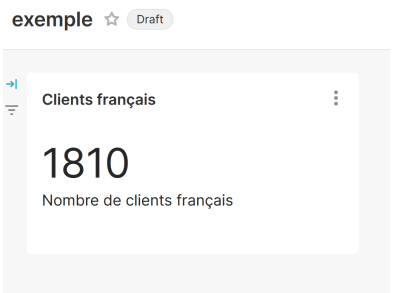
Finally, you can save the query as a *Dataset* using *Save>Save Dataset* once the query has been executed.

And the counting ? *Indeed, the counting is not done in the query, because it is performed during aggregation in the visualization (Chart) settings.*

In the *Charts* tab, we will now create a new *Chart* by selecting the *Big Number* type and then selecting the dataset created earlier. By default, a count metric already exists, so it can be selected via the *SAVED* tab once you click on *METRIC*. It is also possible to choose other aggregation methods via the *SIMPLE* or *CUSTOM SQL* tabs (using SQL). We also note that country filtering could have been done during the creation of the visualization through the *Filters* parameters of the *Query*.

Now, by clicking on *Create Chart*, the result appears on the right. When saving, you can directly save it into a new *Dashboard* by choosing its name and clicking on it. You can then view the *Dashboard* and edit the various views directly within the dashboard. Take a look at the *Customize* tab of the *Charts* and modify it to add a *Subheader* and display the full number. Through the *Dashboard* editing mode, it is possible to change the size of the *Charts* and move them around, as well as add *Layout* elements.

Figure 2 – Result of the Dashboard following the exercise 1



3.2 Exercise 2

1. Create a *Dataset* to obtain the country column of all customers in English.
2. Create a *Chart* of type *World Map* to display the number of customers per country on the map.

On your report, include the SQL query and an image of the visualization result from your *Dashboard*.

3.3 Exercise 3

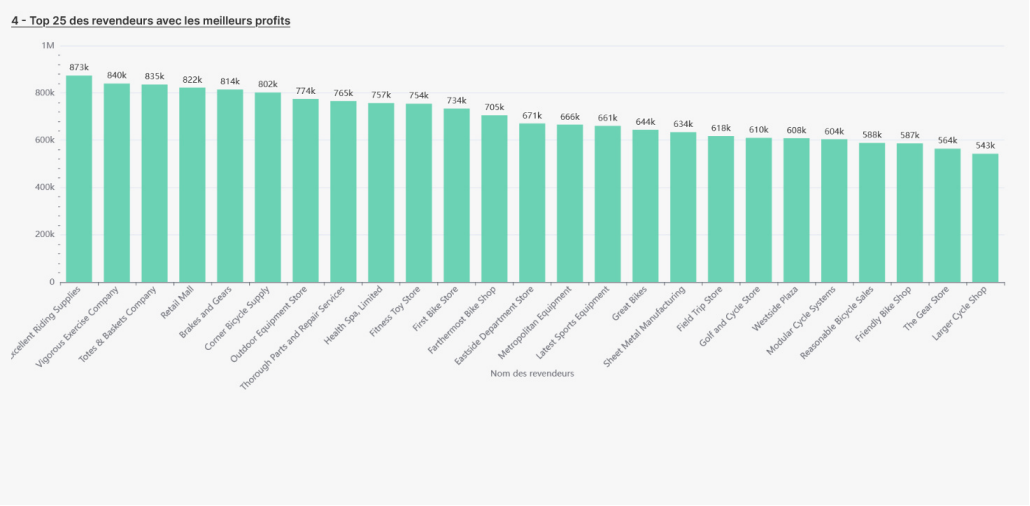
1. Create a *Dataset* to get the number of sales by product categories in English (field *EnglishProductCategoryName*) made by resellers (table *factresellerssales*).
2. Create a *Chart* of type *Sunburst Chart* to display the number of sales. (**Don't forget to take into account the order quantity !**)
3. Modify it to display the total in the center of the diagram.

On your report, include the SQL query and an image of the visualization result from your *Dashboard*.

3.4 Exercise 4

1. Create a *Dataset* to get the total cost of products (field *TotalProductCost*) from resellers (table *factresellerssales*).
2. Create a *Chart* of type *Bar Chart* to obtain the same display as in Figure 3 (limit of 25 names).

Figure 3 – Result wanted for exercise 4.



On your report, include the SQL query and an image of the visualization result from your *Dashboard. (Do not submit the same one, it will be checked with the Dashboard export).*

3.5 Exercise 5

1. Create a *Dataset* to get the quantity of reseller sales (table *factresellerssales*) with the sales date.
2. Modify the *Dataset* to make the date column temporal.
3. Create a *Chart* of type *Line Chart* to display the number of sales over time. (**Don't forget to take into account the order quantity !**)

On your report, include the SQL query and an image of the visualization result from your *Dashboard*.

3.6 Exercise 6

1. Create a *Dataset* to get the quantity of reseller sales (table *factresellerssales*), the sales date and its category.
2. Modify the *Dataset* to make the date column temporal.
3. Create a *Chart* of type *Area Chart* to display the number of sales over time by categories. (**Don't forget to take into account the order quantity !**)
4. Modify it to obtain a correct graph display (with legend and axis titles, etc...)

On your report, include the SQL query and an image of the visualization result from your *Dashboard*.

3.7 Exercise 7

1. Create a *Dataset* that computes the cumulative sum of online sales (table *factinternetsales*) over days [1]. Use the date from the field *DueDateKey* to determine the sale day. The results must be sorted by customer ID and by date (field *FullDateAlternateKey*). (**Don't forget to take into account the order quantity !**))
2. Créez un *Chart* permettant d'afficher via un tableau le résultat de la requête en affichant seulement les colonnes de la date, l'identifiant du client et la somme cumulative.
3. Create a *Chart* displaying the result of the query with a table. Only show the date column, the user identification and the cumulative sum.
4. Only display the bars of the cells on the cumulative sum.

On your report, include the SQL query and an image of the visualization result from your *Dashboard*.

3.8 Exercise 8

1. Create a *Dataset* to get the quantity of orders for each online sale (table *factinternetsales*) and the delivery date (field *DueDateKey*).
2. Create a *Chart* of type *Pivot Table* to display the order quantities where the rows are the customer IDs and the columns are the sales dates by month.
3. Add the total sum of sales per customer.
4. Add display conditions to change the cell colors.

1. Use SQL Window Functions (https://en.wikipedia.org/wiki/Window_function_(SQL))

(a) Higher than 7, use the `success` color.
(b) Between 4 and 8 (bounds not included), use the `alert` color.
(c) For the rest, use the `error` color.

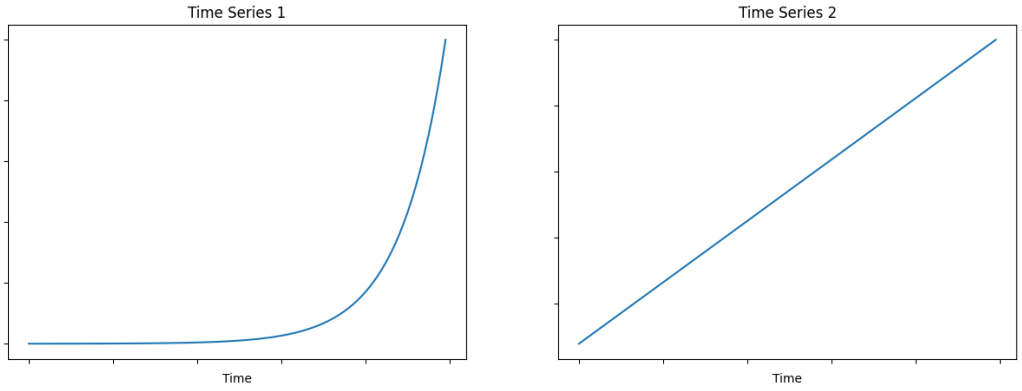
On your report, include the SQL query and an image of the visualization result from your *Dashboard*.

3.9 Exercise 9

Answer the following questions in your report.

1. Give the main advantage of performing a visualization on the Web compared to a document (printed report for example). Then describe at least two examples of this advantage.
2. Explain what the Shannon communication model (1948) has indirectly brought to the field of data visualization.
3. A scale has been applied to the time series data of the first diagram in Figure 4, resulting in the display of the second diagram. Explain what this scale is and describe what information it gives compared to the first diagram.

Figure 4 – Diagram containing raw time series data (left) and to which a scale has been applied resulting in the second graph (right).



3.10 Exercise bonus

Try to modify the *Dashboard* display to add titles and move/resize the *Charts* for better display. Then add a final *Chart* on which you are free to do whatever you want.

Do not reuse a *Dataset* or the same type of *Chart* as those from previous exercises !

On your report, include the SQL query and an image of the visualization result from your *Dashboard*.

References

[1] B. Inmon, M. Levins, and R. Srivastava. *Building the Data Lakehouse*. Technics Publications. ISBN: 9781634629683.