Laboratoire 4 - Apache Superset

Dr. Laura E. Raileanu, Cédric Campos Carvalho

29 novembre 2023

1 Introduction

1.1 Apache Superset

Apache Superset est une plateforme de visualisation de données open source qui aide à transformer les données en informations exploitables. C'est un outil puissant qui peut être utilisé par les Data Scientists ou autres utilisateurs professionnels pour soutirer de l'information sur l'entreprise.

Selon Inmon, Levins et Srivastava [1], la visualisation des données est un processus qui consiste à représenter les données de manière graphique ou visuelle. Cela permet de comprendre plus facilement les données et d'identifier des structures et des tendances qui seraient difficiles à voir dans les données brutes.

Apache Superset offre une large gamme d'options de visualisation, notamment des graphiques, des tableaux, des cartes et des tableaux de bord. Il prend également en charge une variété de sources de données, notamment les bases de données relationnelles, les bases de données NoSQL et le stockage cloud.

La visualisation des données est un outil puissant qui peut être utilisé pour de nombreuses applications, notamment :

- La communication des résultats : La visualisation des données peut aider à communiquer les résultats des analyses de données de manière claire et concise.
- La prise de décision : La visualisation des données peut aider à prendre de meilleures décisions en fournissant une meilleure compréhension des données.
- L'identification de nouvelles opportunités : La visualisation des données peut aider à identifier de nouvelles opportunités et tendances qui ne seraient peut-être pas visibles autrement.

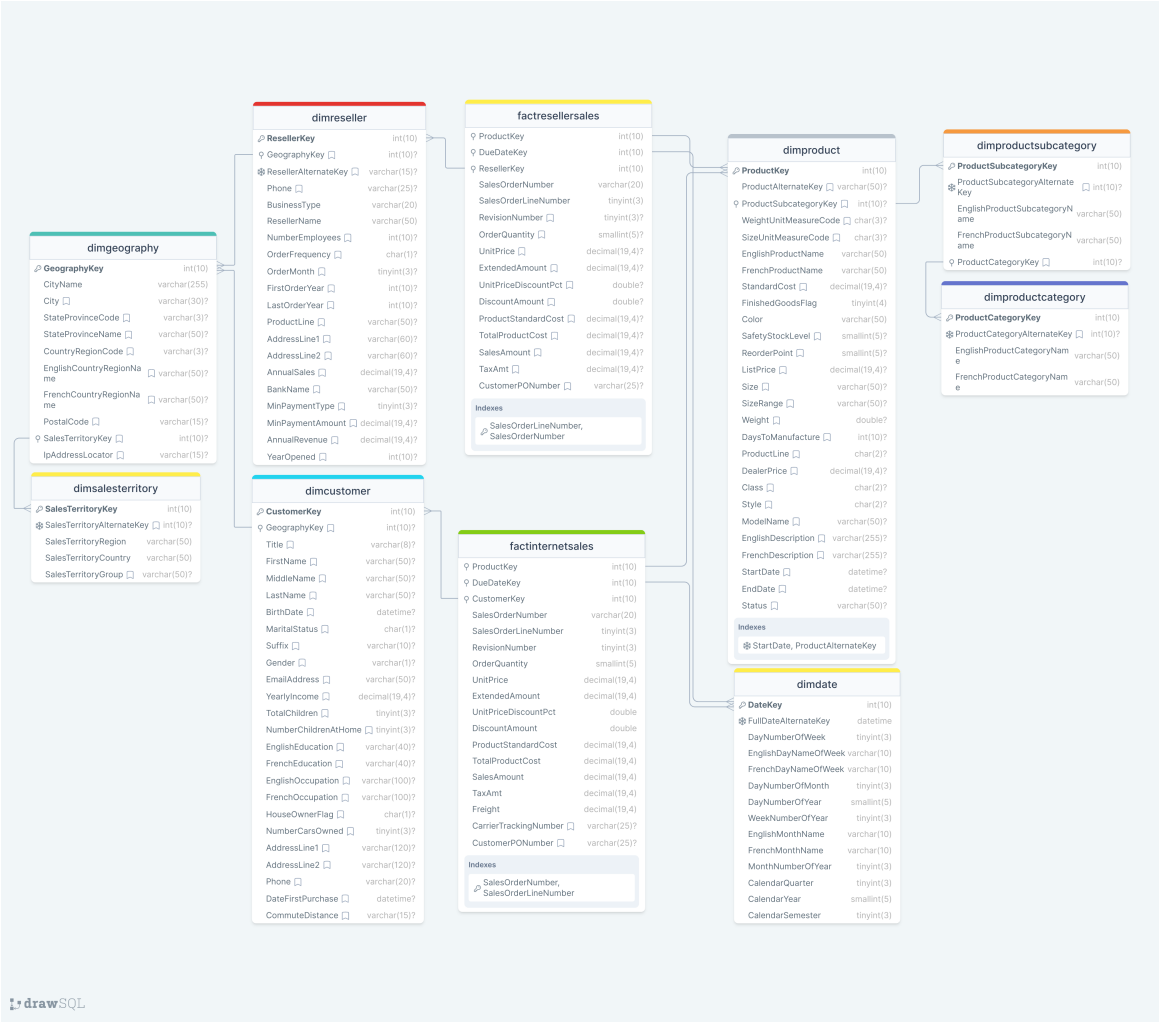
1.2 Les données

Les données proviennent du jeu de données AdventureWorks. Elles proviennent d'un échantillon fourni par Microsoft comme exemple pour ses produits SQL. Nous avons adapté et simplifié ces données afin que l'on puisse les utiliser dans le cadre de ce laboratoire. Ces données représentent environ 120'000 ventes d'articles de vélo, par internet et en magasin, dans plusieurs pays.

1.3 Rendu

Le rendu se fait sur la page Moodle jusqu'au **14 décembre 2023 à 23:59:59** ou vous mettrez à disposition un rapport contenant les réponses des questions (images y compris) sous format **PDF** et l'exportation du *Dashboard de Apache Superset* contenant les différentes visualisations des exercices sous format **ZIP**.

Figure 1 – Diagramme Adventure Works simplifié.



2 Préparation

*Apache Superset*offre plusieurs manières de s'installer (voir page officielle), pour ce laboratoire il est conseillé d'utiliser la méthode Docker décrite dans la section 2.1. Si vous rencontrez beaucoup de problèmes lors de l'installation avec Docker, une seconde méthode via Pypi est décirte à la section 2.2.

2.1 Installation via Docker

Pour ce laboratoire, on vous fournit plusieurs fichiers qui s'occuperont de mettre en place le *Docker container* contenant toutes les informations nécessaires dont nottamment la base de données pour le laboratoire. Une fois placé dans le dossier contenant les fichiers d'installation, simplement lancer le docker via la commande :

```
1 docker compose up
```

2.2 Installation via Pypi

Pour les utilisateurs de Windows, il vous est **fortement** conseillé d'utiliser WSL avec la distribution Ubuntu. L'installation a été testé avec :

— Ubuntu 22.04.2 LTS
— Python 3.10.6

La première étape consiste à installer toutes les dépendances nécessaires pour créer les environnements pythons et les compilateurs `GCC` et `G++` nécessaires pour *Apache Superset*.

```
sudo apt-get update
sudo apt-get install libpython3-dev
sudo apt-get install python3-venv
sudo apt-get install gcc
sudo apt-get install g++
```

Création de l'environnement python et activation :

```
python3 -m venv venv
. venv/bin/activate
```

Avant d'installer le package Python *Apache Superset*, il est conseillé de mettre à jour `pip` :

```
pip install --upgrade setuptools pip
pip install apache-superset
```

Pour des raisons de compatibilités, il est nécessaire de *downgrade* la version de `sqlparse` :

```
    pip install sqlparse==0.4.3
```

Il se peut que le package `marshmallow-enum` soit manquant, pour l'installer :

```
pip install marshmallow-enum
```

On peut maintenant passer à la configuration *Apache Superset*. En premier lieu, il est nécessaire de spécifier en variable d'environnement le nom du module à importer pour *Flask* (micro web framework en Python) :

```
export FLASK_APP=superset
```

Apache Superset a besoin de d'avoir une clé secrète générée aléatoirement dans un fichier de configuration python. La génération de la clé se fait grâce à la commande `openssl rand -base64 42`. Puis, il faut créer un fichier appelé `superset_config.py` et y ajouter le contenu suivant :

```
ROW_LIMIT=5000
SECRET_KEY = 'YOUR_OWN_RANDOM_GENERATED_SECRET_KEY'
```

Il ne reste plus qu'à ajouter la variable d'environnement pour spécifier le chemin au fichier de configuration [1] :

```
export SUPERSET_CONFIG_PATH=superset_config.py
```

Finalement, l'initialisation se fait avec les 3 commandes suivantes. La deuxième demande des informations pour le compte administrateur.

```
superset db upgrade
superset fab create-admin
superset init
```

1. Ces variables sont actives seulement pour le terminal où elles sont exécutées, pour les ajouter de façon permanente, regarder ce lien.

Il est désormais possible de lancer le serveur. Le port par défaut est 5000, si besoin il est possible de le changer avec l'option `-p N_PORT` :

```
superset run
```

2.3 Connexion à la base de données

Une fois lancé, *Apache Superset* est accessible via navigateur sur le port définit au préalable. Puis une fois connecté, il contient différents onglets, notamment :

- **Dashboards** : Une liste de tableaux de bord, qui sont des visualisations de données interactives
- **Charts** : Une liste de graphiques à partir de Datasets pour les tableaux de bords
- **Datasets** : Une liste d'ensemble de données, qui sont des collections de données brutes formées via requêtes SQL
- **SQL** : Offre une interface pour exécuter des requêtes SQL
- **Settings** : les paramètres qui peuvent être utilisés pour personnaliser son fonctionnement, notamment les connexions aux bases de données.

Si besoin, pour ajouter une connexion à une base de données, il faut accéder aux paramètres et sélectionner "*Database Connections*". En premier, il est nécessaire de rendre accessible le fichier téléchargé "*aventures.sqlite3*" sur un chemin, tel que "`/home/username/.superset`". En haut à droite, cliquer sur "*+Database*" et sélectionner "*SQLite*". Ensuite, il faut choisir un nom (Aventures) et entrer le chemin URI. Pour suivre l'exemple, la valeur est :

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

Une fois terminé, cliquer sur "*Finish*".

3 Exercices

3.1 Exercice 1

Cet exercice va vous permettre de vous familiariser avec *Apache Superset* et vous guidera étape par étape pour obtenir le résultat souhaité. L'objectif est de créer un *Dashboard* puis d'y afficher un *Chart* permettant d'afficher le nombre de clients français via une commande SQL enregistrée en tant que *Dataset*.

Pour effectuer la requête, vous pouvez vous aider via *SQL Lab* puis sélectionnez la bonne *Database* (Aventures). Maintenant, exécutez la commande suivante :

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

Cette commande effectue une jointure sur la table `dimcustomer` et `dimgeography`, puis filtre via le nom du pays pour sélectionner les clients de `France`. N'hésitez pas à regarder le schéma des tables (Figure 1 ou grâce à *SQL Lab*) pour vous aider à effectuer les requêtes.

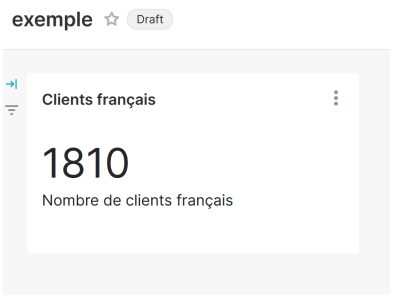
Finalement, vous pouvez enregistrer la requête en tant que *Dataset* avec *Save>Save Dataset* une fois la requête exécutée.

Et le comptage ? *Effectivement, le comptage n'est pas fait sur la commande, car elle se fait lors de l'agrégation dans les paramètres de la visualisation (Chart).*

Dans l'onglet *Charts*, nous allons maintenant créer un nouveau *Chart* en sélectionnant le type *Big Number* puis en sélectionnant le dataset créé avant. Par défaut, une métrique de comptage existe déjà, il est donc possible de la sélectionné via l'onglet *SAVED* une fois appuyé sur *METRIC*. Il est également possible de choisir d'autres méthodes d'aggrégations via l'onglet *SIMPLE* ou *CUSTOM SQL* (avec du SQL). On remarque aussi que le filtrage des pays aurait pu se faire au moment de la création de la visualisation via les paramètres *Filters* de la *Query*.

Maintenant, en appuyant sur *Create Chart*, le résultat s'affiche sur la droite. Lors de la sauvegarde, vous pouvez directement le sauvegarder dans un nouveau *Dashboard* en choisissant son nom et cliquant dessus. Vous pouvez ainsi visualiser le *Dashboard* et éditer les différents vues directement via le dashboard. Jetez un coup d'oeil à l'onglet *Customize* des *Charts* et modifiez le pour ajouter un *Subheader* et affichez le nombre au complet. Via l'édition du *Dashboard*, il est possible de modifier la taille des *Charts* et de les déplacer, mais également d'ajouter des éléments de *Layout*.

Figure 2 – Résultat du Dashboard suite à l'exercice 1



3.2 Exercice 2

1. Créez un *Dataset* pour obtenir la colonne du pays de tous les clients en anglais.
2. Créez un *Chart* de type *World Map* pour afficher le nombre de clients par pays sur la carte.

Sur votre rapport, mettez la requête SQL et une image du résultat de la visualisation depuis votre *Dashboard*.

3.3 Exercice 3

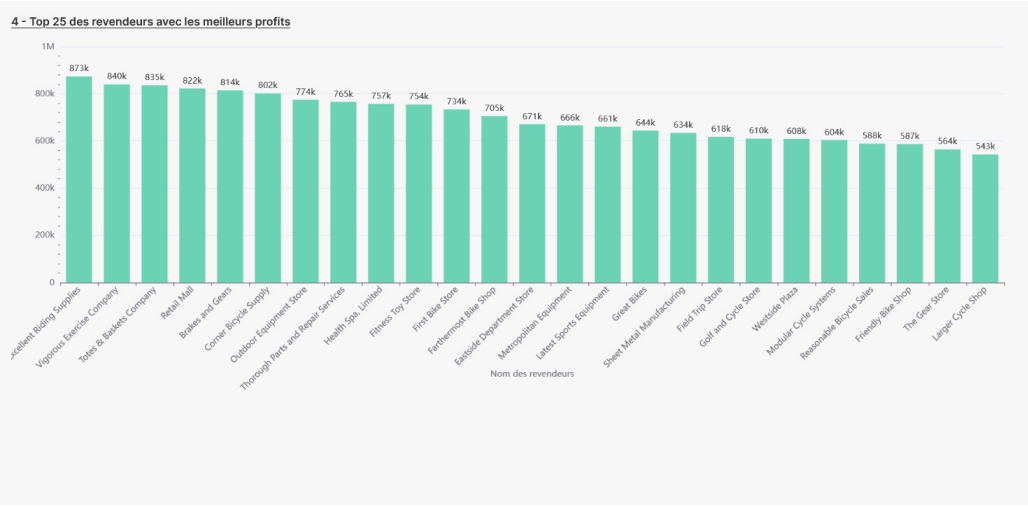
1. Créez un *Dataset* pour obtenir le nombre de ventes par catégories des produits anglais (champ *EnglishProductCategoryName*) faites par les revendeurs (table *factresellerssales*).
2. Créez un *Chart* de type *Sunburst Chart v2* pour afficher le nombre de ventes. (**N'oubliez pas de prendre en compte la quantité de la commande !)**
3. Modifiez la pour afficher le total au centre du diagramme.

Sur votre rapport, mettez la requête SQL et une image du résultat de la visualisation depuis votre *Dashboard*.

3.4 Exercice 4

1. Créez un *Dataset* pour obtenir le coût total des produits (champ *TotalProductCost*) des revendeurs (table *factresellerssales*).
2. Créez un *Chart* de type *Bar Chart* pour obtenir le même affichage que sur la Figure 3 (limite de 25 noms).

Figure 3 – Résultat souhaité pour exercice 4.



Sur votre rapport, mettez la requête SQL et une image du résultat de la visualisation depuis votre *Dashboard (ne remettez pas la même, elle sera vérifiée avec l'export du Dashboard)*.

3.5 Exercice 5

1. Créez un *Dataset* pour obtenir la quantité de ventes des revendeurs (table *factresellerssales*) avec la date des ventes.
2. Modifiez le *Dataset* pour rendre la colonne de la date temporale.
3. Créez un *Chart* de type *Time-series Line Chart* pour afficher le nombre de ventes au fil du temps. (**N'oubliez pas de prendre en compte la quantité de la commande !)**

Sur votre rapport, mettez la requête SQL et une image du résultat de la visualisation depuis votre *Dashboard*.

3.6 Exercice 6

1. Créez un *Dataset* pour obtenir la quantité des ventes des revendeurs (table *factresellerssales*), la date de vente et sa catégorie.
2. Modifiez le *Dataset* pour rendre la colonne de la date temporale.
3. Créez un *Chart* de type *Time-series Area Chart* pour afficher le nombre de ventes au fil du temps par catégories. (**N'oubliez pas de prendre en compte la quantité de la commande !)**
4. Modifiez le pour obtenir un affichage d'un graphe correct (avec légende et titres des axes, etc...)

Sur votre rapport, mettez la requête SQL et une image du résultat de la visualisation depuis votre *Dashboard*.

3.7 Exercice 7

1. Créez un *Dataset* qui permet d'obtenir la somme cumulée des ventes via les revendeurs (table *factresellerssales*) au fil des jours [2]. Utilisez la date du champ *DueDateKey* pour déterminer le jour de vente. Les résultats doivent être triés par identifiant du client et par date (champ *FullDateAlternateKey*). (**N'oubliez pas de prendre en compte la quantité de la commande !)**)
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. N'affichez que les barres des cellules sur la somme cumulative.

Sur votre rapport, mettez la requête SQL et une image du résultat de la visualisation depuis votre *Dashboard*.

3.8 Exercice 8

1. Créez un *Dataset* qui permet d'obtenir la quantité des commandes pour chaque ventes des revendeurs (table *factresellerssales*) puis la date de livraison (champ *DueDateKey*).
2. Créez un *Chart* de type *Pivot Table* pour afficher les quantités des commandes où les lignes sont l'identifiant des clients et les colonnes sont les dates par mois des ventes.
3. Ajouter la somme totale des ventes par client.

2. Utilisez les Window Function de SQL (https://en.wikipedia.org/wiki/Window_function_(SQL))

4. Ajouter des conditions d'affichage pour changer les couleurs des celulles.
   (a) Plus grand que 7, utiliser la couleur `success`.
   (b) Entre 4 et 8 (bornes non-incluses), utiliser la couleur `alert`.
   (c) Pour le rester, utiliser la couleur `error`.

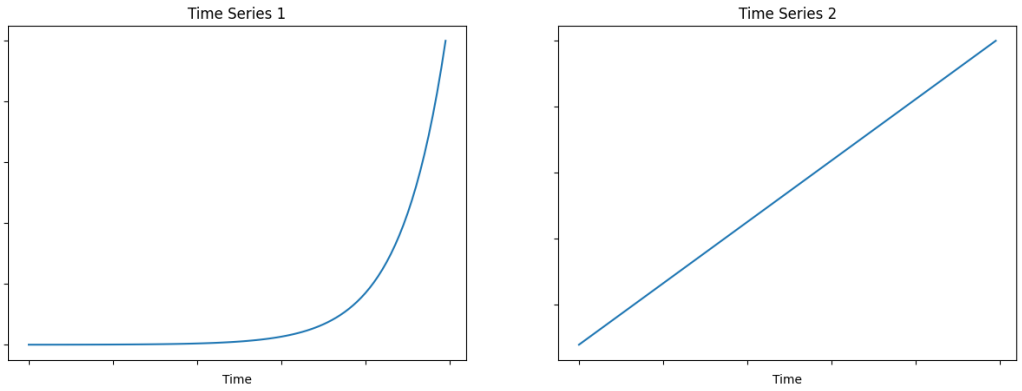
Sur votre rapport, mettez la requête SQL et une image du résultat de la visualisation depuis votre *Dashboard*.

3.9 Exercice 9

Veuillez répondre aux questions sur votre rapport.

1. Donner l'avantage premier d'effectuer une visualisation sur le Web comparé à un document (rapport imprimé par exemple). Puis décrire au moins deux exemples de cet avantage.
2. Expliquer ce qu'a indirectement apporté le modèle de communication de Shannon (1948) dans le domaine de visualisation des données.
3. Une échelle a été appliquée sur les séries de données temporelles du premier schéma de la Figure 4, résultant à l'affichage du deuxième schéma. Expliquer quelle est cette échelle et décrire quelle information elle donne par rapport au premier schéma.

Figure 4 – Schéma contenant des données temporelles en brut (à gauche) et dont on a appliqué une échelle résultant au deuxième graphe (à droite).



3.10 Exercice bonus

Tentez de modifier l'affichage du *Dashboard* pour ajouter des titres et déplacer/redimensionner les *Charts* pour un meilleur affichage. Puis ajoutez-y une dernière *Chart* sur laquelle vous êtes libres de faire ce que vous voulez.

Ne réutilisez pas un *Dataset* ou un même type de *Chart* que ceux des exercices précédents !

Sur votre rapport, mettez la requête SQL et une image du résultat de la visualisation depuis votre *Dashboard*.

Références

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