SAE 1.04 Création d'une base de données

Vous devez utiliser les connaissances acquises en BDD pour réaliser une application s'appuyant sur une base de données. Le travail principal consiste à récupérer des données officielles et à s'aider du langage SQL pour les stocker dans une base de données et de les explorer afin de produire des analyses.

Quelles sont les activités et productions de cette SAÉ ?

- Constituez-vous en groupes de 2 étudiants maximum
- Prendre connaissance du sujet et plus particulièrement de la structure de données fournies dans un fichier au format csv
- Le modèle de données
- Script création base de données
- Exploitation de la base de données et visualisation des résultats

La source de données que vous devez télécharger concerne le naufrage du Titanic (titanic_train.csv) https://up13.fr/?A7XN7MLb

Pour en savoir plus sur cet ensemble de données, vous pouvez lire https://www.kaggle.com/c/titanic/data

By Willy Stöwer, died on 31st May 1931 - Magazine Die Gartenlaube, en:Die Gartenlaube and de:Die Gartenlaube, Public Domain, Link

Le déroulement du projet :

Phase 1 : évaluation en séance de TP, semaine 20/09

- Intégration de PostgreSQL avec Jupyter Notebook
- Téléchargez le fichier `titanic_train.csv` et analysez sa structure (informations contenues dans fichier, objectifs visés)
- A partir de l'analyse qui aura été effectuée, produire le modèle de données (domaine des attributs, cardinalité, degrés). Définissez les tâches à faire (un tableau des tâches : phase 1, 2 et 3).

Phase 2 : évaluation en séance de TP, semaine 18/10

- Créez les tables nécessaires
  - Un mini rapport d'une demi page attendu sur la description des tables (les attributs, les clés.)
  - Un mini script permettant de créer les tables
  - Rapport et script en mode notebook (mode note)
- A l'aide de Python et SQL, alimentez chacune des tables à partir du contenu du fichier `titanic_train.csv`
  - Une démo

Phase 3 : évaluation en salle de TP, semaine 29/11

- Créez des requêtes SQL afin d'interroger la base de données et visualiser les résultats (amusez-vous)
  - Un mini rapport avec des visualisations et le script en mode notebook
  - Démo, présentation
  - Pour cette dernière démonstration, vous devez répondre à ces requêtes :
    - Combien de classes de passagers différentes y avait-il à bord du Titanic ?
    - Combien de passagers y avait-il dans chaque classe?
    - Combien de femmes et d'hommes y avait-il dans chaque classe?
    - Comptez le nombre et le pourcentage de survivants et de passagers morts.
    - Visualiser la répartition des passagers survivants et morts par classe
    - Visualiser la répartition des passagers survivants et des passagers décédés selon le sexe

Intégration du PostgreSQL avec Jupyter Notebook

Au cœur de tout système d'information se trouve un système de base de données. Il existe de nombreuses applications disponibles, cependant, pour ce projet SAE, nous utiliserons uniquement PostgreSQL. En outre, nous utiliserons également Jupyter, une application de navigateur basée sur Python qui permet de créer facilement, l'exécution et la réutilisation de requêtes SQL par le biais de notebook interactif. Ainsi il est possible d'utiliser dans le même notebook des requêtes SQL et du code python grâce à des extensions (seront décrites plus loin).

1 Postgresql

Visitez le site officiel de PostgreSQL (https://www.postgresql.org/download), où vous trouverez des instructions sur les étapes à suivre pour installer le système de base de données. Ci-dessous des informations synthétiques pour le système d'exploitation de votre choix.

Linux

>sudo apt install postgresql-12
sudo service postgresql start
Connectez-vous avec l'utilisateur "postgres" et lancez la ligne de commande PostgreSQL :
>sudo su - postgres
>psql
> \password (pour modifier le mot de passe)

Mac Os
Nous vous recommandons d'utiliser Homebrew pour l'installation et la maintenance des logiciels sur macOS (https://brew.sh/)

> brew install postgres
> brew services start postgresql
> psql postgres ou > psql postgres <login>
> \password (pour modifier le mot de passe)

Windows

Une bonne option consiste à télécharger le programme d'installation interactif d'EnterpriseDB.
https://www.enterprisedb.com/downloads/postgres-postgresql-downloads

Note : Pour le système d'exploitation Windows, rajouter le chemin d'accès aux commandes de Postgresql dans les variables d'environnement de Windows.

2 Jupyter

Cette étape n'est pas nécessaire si vous disposez déjà de Jupyter. Le moyen le plus simple est d'installer Jupyter à l'aide de pip/pip3 :
> pip3 install jupyter

Il existe un système automatisé d'installation de Jupyter ainsi que d'autres outils nommé Anaconda ; il est conseillé de l'installer et d'utiliser ses outils pour installer les différentes extensions citées plus bas.

Pour cela, vous pouvez visiter le site web officiel de Jupyter (jupyter.org/install). Vous pouvez également visitez https://www.anaconda.com/products/individual.

Note : Pour le système d'exploitation Windows, vous taperez les commandes d'installation des différentes extensions qui suivent en utilisant le shell d'Anaconda "*Anaconda PowerShell Prompt*".

3 ipython-sql

Créé par Catherine Devlin sur Github (https://github.com/catherinedevlin/ipython-sql), il permet l'utilisation de fonctions magiques SQL qui contiennent % et %%, vous permettant d'écrire du code de style SQL directement dans Jupyter Notebook.

>pip install ipython-sql

4 sqlalchemy

Créé à l'origine par Michael Bayer, sqlalchemy est présenté comme une "boîte à outils SQL et un mappeur objet-relationnel" pour Python. Il sera principalement utilisé pour stocker des requêtes SQL dans un dataframe pandas.

>pip install sqlalchemy

5 psycopg2

La troisième bibliothèque dépend du logiciel SQL que vous choisissez d'utiliser. Pour PostgreSQL, vous utiliserez psycopg2 :

>pip install psycopg2

6 Exemple

Créez un nouveau fichier notebook test.ipynb, puis vous pouvez créer une cellule pour chaque commande

```
%load_ext sql
```

```
%sql postgresql://user:pass@localhost/postgres
```

```
%%sql
CREATE TABLE personnes(nom varchar(10), age integer);
INSERT INTO personnes VALUES ('Gae'l, 30);
INSERT INTO personnes VALUES ('Nassim',22);
INSERT INTO personnes VALUES ('Clarisse', 22);
```

```
%%sql
SELECT * FROM personnes;
```

```
#commande python
result = %sql select * from personnes
```

```
print(result[1])
```

```
dataframe = result.DataFrame()
dataframe
```

Source
https://medium.com/analytics-vidhya/postgresql-integration-with-jupyter-notebook-deb97579a38d