**RESULTATS DU TP DE DASSI SOPHIE**

**EXPLICATIONS**

**Données originales de la compétition Titanic**

Les données utilisées pour ce projet ont été récupérées sur kaggle au lien :

<https://www.kaggle.com/datasets/heptapod/titanic/data>.

Ces données relevées sur le naufrage Titanic ont été élaborées par Khashayar Baghizadeh Hosseini. Elles donnent des informations sur les survivants et les non-survivants de ce naufrage en donnant les détails concernant sur l’id, l’age, le sexe…

Cette base **bd\_titanic** comporte une seule table : **Passagers ;**

**Id Passenger :** identifiant des passagers

**Age :** contient l’âge des passagers à bord

**Fare :** Cette colonne indique le prix payé par chaque passager pour son billet à bord du Titanic.

**Sex :** Le sexe de chaque passager

**Sibsp :** Cette colonne indiquerait le nombre de frères, sœurs ou conjoints (époux/épouse) que chaque passager avait à bord du Titanic. Cela inclut à la fois les frères et sœurs biologiques et les conjoints.

**Survived :** contient des valeurs binaires : 1 pour survivant, 0 pour décédé

* **Pistes d’analyse des données**

Avec les données que comportent ce jeu de données nous pourront à travers les variables existantes chercher à :

* connaitre le nombre de survivants ;
* connaitre le taux de mortalité en fonction de l’age ;
* connaitre parmi les non-survivants l’age du plus jeune ;
* connaitre parmi les non-survivants l’age du plus vieux ;
* connaitre le sexe dominant parmi les non survivants.

**Mise en place d’un ETL en python qui permettra de charger les données dans notre base de données MySQL.**

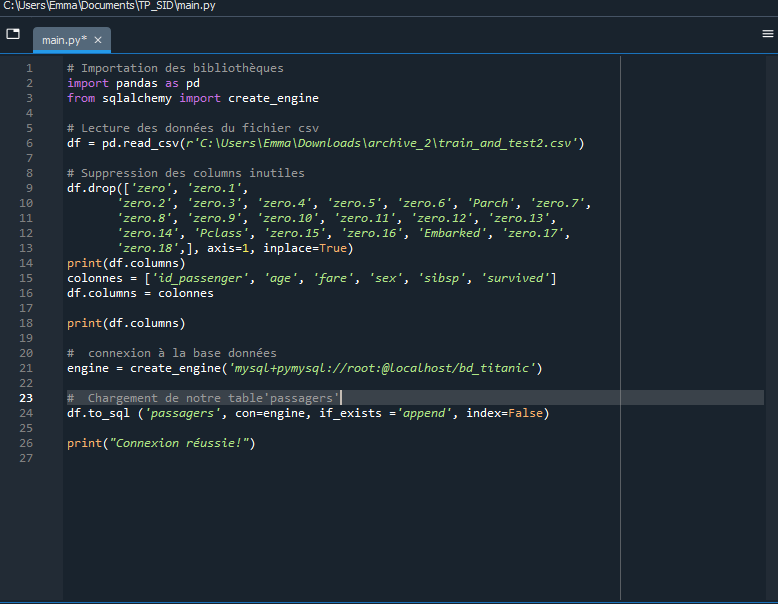


Figure illustrant le processus d’extraction et de transformation des données

**Voici une explication de chaque ligne de votre code** :

1. **`import pandas as pd`:** Cette ligne importe la bibliothèque Pandas en tant que "pd", ce qui nous permet d'utiliser les fonctionnalités de Pandas dans notre code en utilisant l'alias "pd".

2. **`from sqlalchemy import create\_engine**`: Cette ligne importe la fonction `create\_engine` de la bibliothèque SQLAlchemy. Cette fonction est utilisée pour créer la connexion à notre base de données SQL.

3.**`df=pd.read\_csv(r'C:\Users\Emma\Downloads\archive\_2\train\_and\_test2.csv')`:** Cette ligne lit les données à partir du fichier CSV situé à l'emplacement spécifié et les stocke dans le DataFrame Pandas appelé "df".

4. `**df.drop([...], axis=1, inplace=True)`:** Cette ligne supprime les colonnes spécifiées du DataFrame "df". Les colonnes à supprimer sont indiquées dans la liste fournie.

5. `**print(df.columns)`:** Cette ligne imprime les noms des colonnes restantes dans le DataFrame "df" après avoir supprimé les colonnes inutiles.

6. **`colonnes = ['passengerid', 'age', 'fare', 'sex', 'sibsp', 'survived']`:** Cette ligne crée une liste appelée "colonnes" contenant les noms de colonnes souhaités pour le DataFrame.

7. `**df.columns = colonnes`:** Cette ligne attribue les noms de colonnes de la liste "colonnes" au DataFrame "df", remplaçant ainsi les noms de colonnes actuels par ceux spécifiés.

8. **`engine = create\_engine('mysql+pymysql://root:@localhost/bd\_titanic')`:** Cette ligne crée une connexion à la base de données MySQL appelée "bd\_titanic" sur le serveur local. Le nom d'utilisateur est "root" et il n'y a pas de mot de passe spécifié ici

9. **`df.to\_sql('passagers', con=engine, if\_exists='append', index=False)`:** Cette ligne écrit les données du DataFrame **"df"** dans une table MySQL appelée "passagers" dans la base de données spécifiée. L'argument **"if\_exists='append'"** indique que les données doivent être ajoutées à la table si elle existe déjà. L'argument **"index=False"** spécifie que l'index du DataFrame ne doit pas être inclus dans la table MySQL.

10. **`print("Connexion réussie!")`:** Cette ligne imprime un message indiquant que la connexion à la base de données a été établie avec succès. Cela confirme que les données ont été écrites avec succès dans la base de données.

**Pour la creaction de la base de données nous exécuter des synthaxe MySQL :**

CREATE TABLE `passagers` (

`id\_passenger` int(11) NOT NULL,

`age` int(11) DEFAULT NULL,

`fare` float DEFAULT NULL,

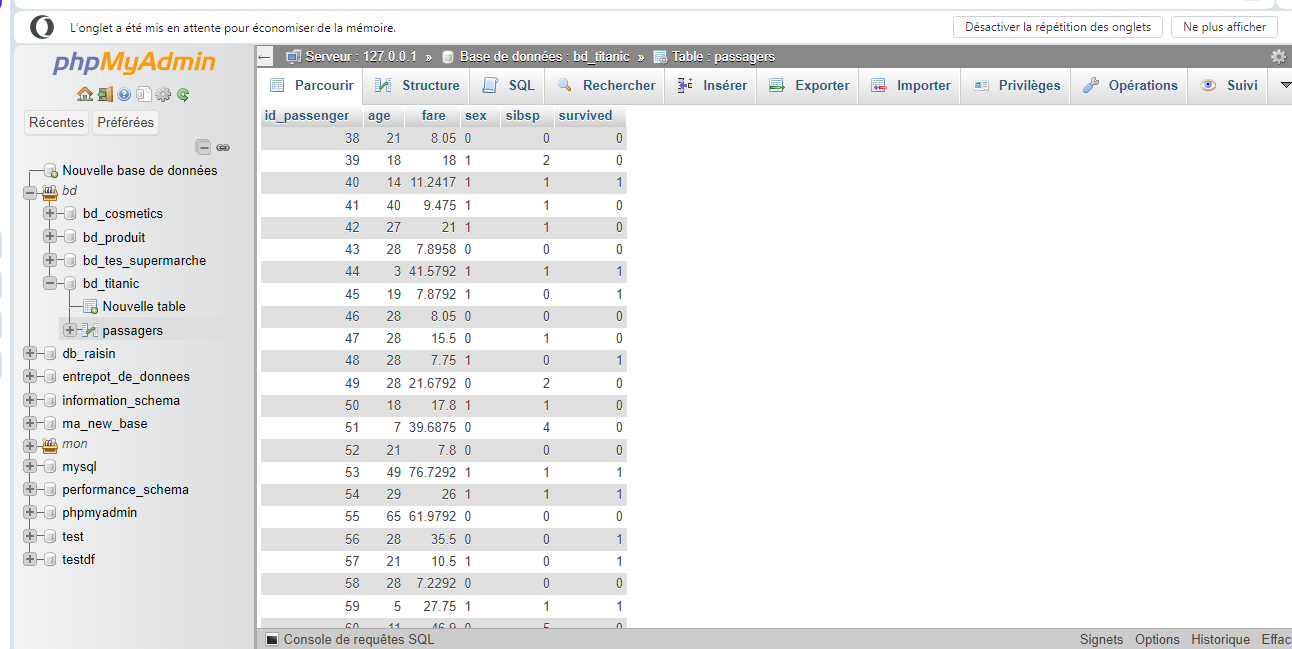
`sex` varchar(20) DEFAULT NULL,

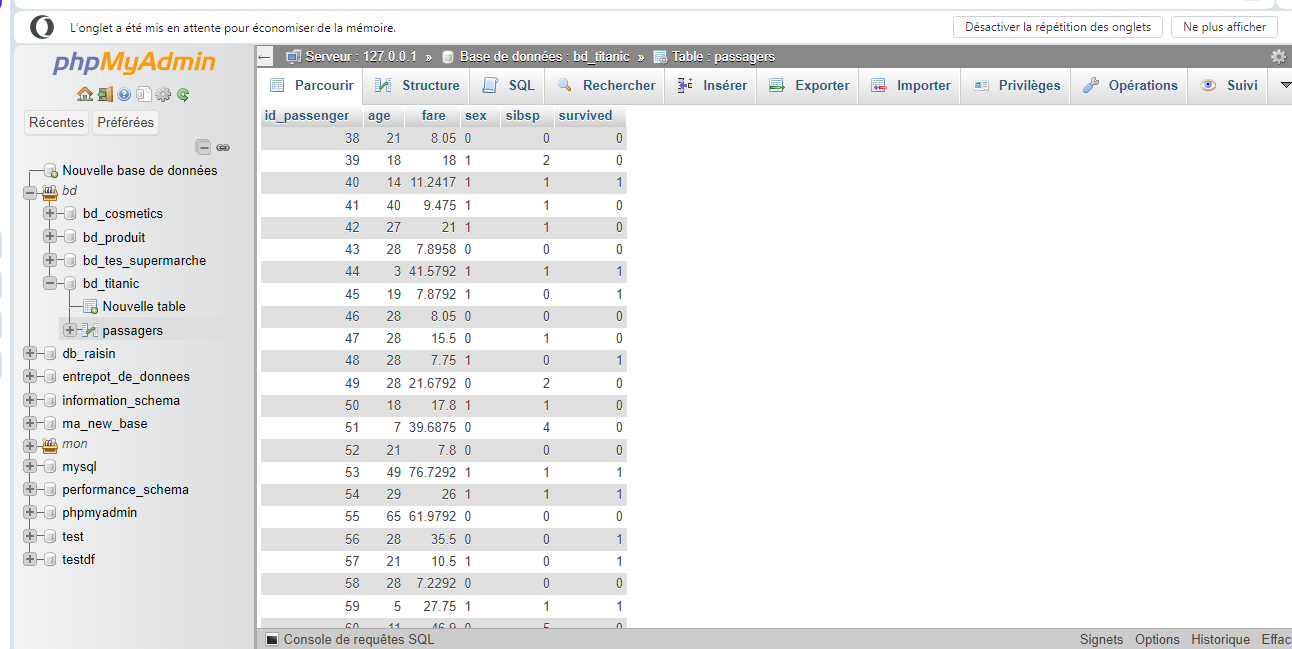
`sibsp` int(11) DEFAULT NULL,

`survived` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

COMMIT;

**Vues de la base de données dans MySQL**



* Une aperçue du processus de clonage, d’ajout, de commit et de push

