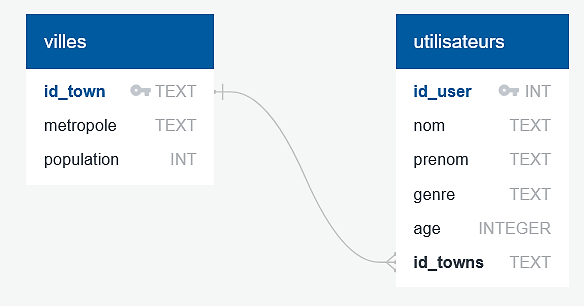
Tp\_2 - Requêtes Sql avec sqlite3 - python

Dans le Tp\_1 précédent, nous avons créé les tables ***villes*** et ***utilisateurs*** ci-dessous. Nous les avons remplis et avons exécuté des requêtes SQL de sélection afin d’en exploiter les données.

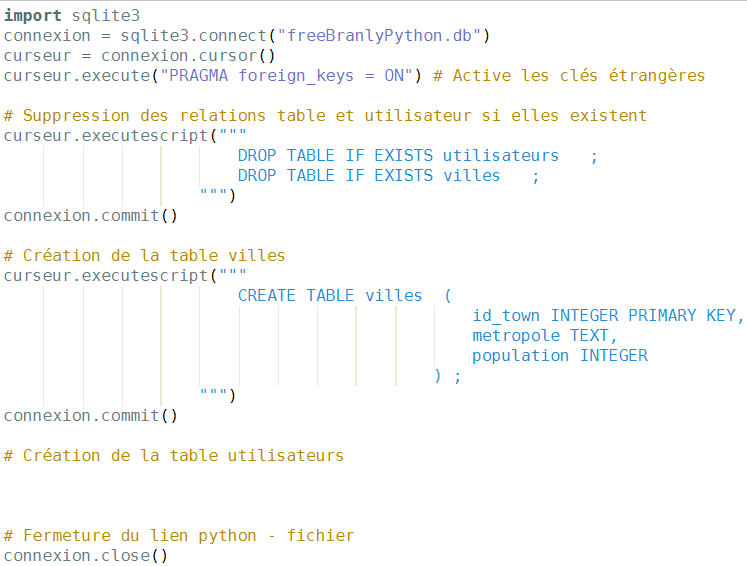
Ce travail avait été réalisé par l’intermédiaire du logiciel *DB Browser* qui permet de créer une base de données, d’exécuter des requêtes Sql et de visualiser le contenu des tables.

On se propose à présent, dans ce Tp\_2, de réaliser les mêmes opérations en exécutant les mêmes requêtes Sql cette fois-ci dans un code python. On utilisera ici uniquement *DB Browser*, pour vérifier l’état de la base de données créée.

⇨ Sur Pyzo, ouvrir un nouveau fichier nommé *freeBranlyPython.py* .

# **Création des relation *villes* et *utilisateurs* :**

⇨ Dans ce fichier, écrire et compléter le script python ci-dessous qui permet une fois complété, de créer les 2 tables *villes* et *utilisateurs*.



*Si le fichier n’existe pas, une base de données à ce nom est créée automatiquement*

***connexion*** est un objet lié à la base de données

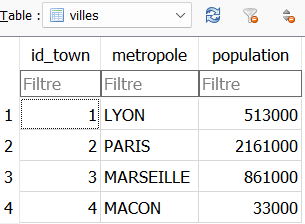
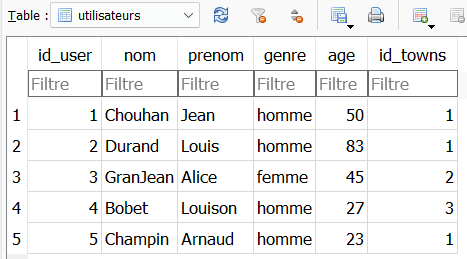
***curseur*** est un objet de la classe sqlite3

Comme se script crée les tables, si on les crées une seconde fois, cela provoque des erreurs. Pour les prévenir, on supprime ici les tables qui seront recréées ensuite.

*commit()* permet d’appliquer les modifications sur la bdd

# **Remplissage des relation *villes* et *utilisateurs* :**

⇨ Compléter encore ce fichier, afin d’insérer les enregistrements ci-dessous dans les tables ***villes*** et ***utilisateurs*** .

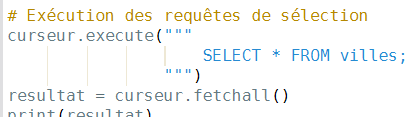
⇨ Vérifier avec le logiciel *DB Browser* que le contenu des tables est bien celui-donné ci-dessus.

A partir de ce point, on n’utilisera plus DB Browser et on pourra donc fermer ce logiciel. Toutes les requêtes de lecture données dans la suite seront réalisées par votre script python.

# **Requêtes de lecture :**

Pour récupérer dans une liste tous les enregistrements de la table villes, on exécute la requête Sql suivante :

On exécute la méthode *execute()* sur l’objet *curseur*



On exécute la méthode *fetchall()* sur l’objet *curseur*

La variable ***resultat*** qui est retournée par la méthode *fetchall()* est une liste de ***tuples*** :



Pour lire par exemple la population de Paris, on peut exécuter dans la console :

# 

Ainsi, si on demande tous les attributs de tous les enregistrements de *villes* , on obtient la liste :

Liste requête



**Les copies d’écrans demandées dans la suite seront insérées directement dans ce fichier .docx .**



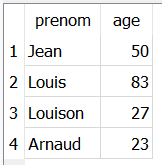
⇨ Donner ci-dessous une copie d’écran de la liste obtenue si on demande le nom des villes dont la population est supérieure à 500 000 habitants :

Liste requête



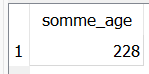
⇨ Donner une copie d’écran de la liste obtenue si on demande le nom des villes qui ont un « R » dans ce nom :

Liste requête

⇨ Donner une copie d’écran de la liste obtenue si on demande le prénom et l’âge des utilisateurs de sexe masculin :

Liste requête

# 

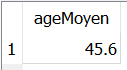
⇨ Donner une copie d’écran de la liste obtenue si on demande la somme de tous les âges.

Liste requête

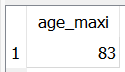


⇨ Donner une copie d’écran de la liste obtenue si on demande le nombre d’enregistrements de la table *utilisateurs*

Liste requête

⇨ Donner une copie d’écran de la liste obtenue si on demande la moyenne des âges des utilisateurs

Liste requête

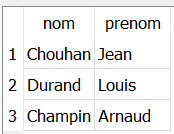


⇨ Donner une copie d’écran de la liste obtenue si on demande l’âge maximum des utilisateurs :

Liste requête

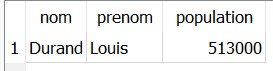
# 

Liste requête (2 requêtes : CREATE … et DROP TABLE ….)

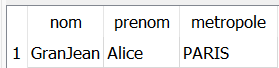


⇨ Donner une copie d’écran de la liste obtenue si on désire demande les noms et prénoms des clients qui habitent LYON.

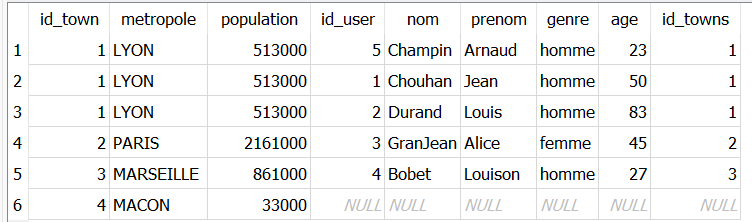
Liste requête

⇨ Donner une copie d’écran de la liste obtenue si on demande les nom, prenom et nombre d’habitants de la ville dans laquelle vit Durand :

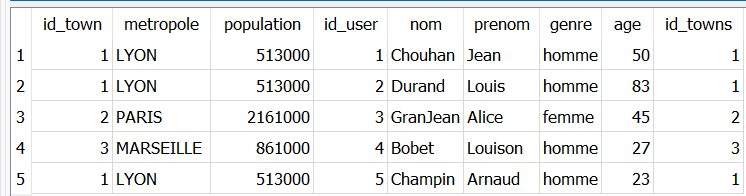
Liste requête

⇨ Donner une copie d’écran de la liste obtenue si on demande les nom, prénom et ville des utilisateurs qui vivent dans une ville de plus de 1 000 000 habitants :

Liste requête

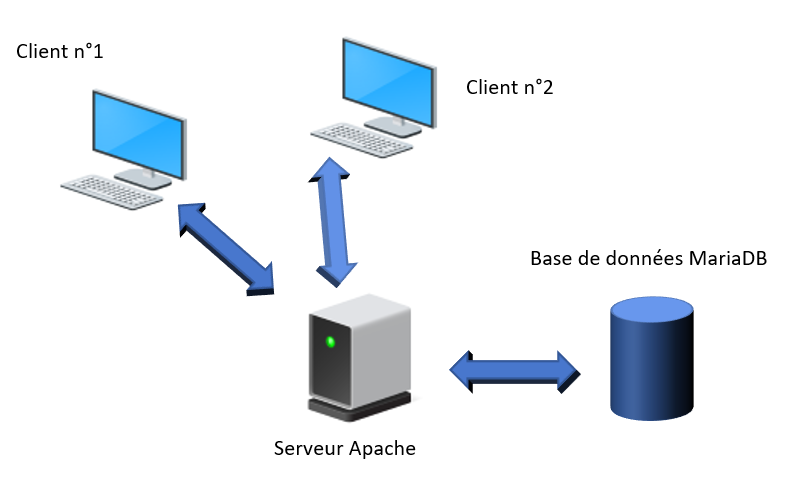
⇨ Donner une copie d’écran de la liste obtenue si on demande tous les attributs de la table *villes* avec en plus les attributs des *utilisateurs*, pour tous les enregistrements de *villes* :

Liste requête

⇨ Donner une copie d’écran de la liste obtenue si on demande tous les attributs de la table villes avec en plus les attributs des utilisateurs, pour seulement les enregistrements de *villes* répertoriés dans *utilisateurs* :

Requête

# **Conclusion :**

On a vu dans cette activité qu’il était possible de gérer une bdd directement avec un code python. C’est exactement ce qui se passe sur un serveur qui héberge un site internet. Ces serveurs peuvent être gérés en python ou le plus souvent avec d’autres langages (php, javascript, ruby, …). Les requêtes sql sont construites en fonction des demandes qui émanent du client. La base de données hébergée sur le serveur retourne alors rapidement les informations demandées pour construire ensuite une réponse (fichier html ou code JS) qui est aussitôt envoyée au client.

⇨ N’oubliez pas d’uploader les fichiers *freeBranlyPython.py et freeBranlyPython.docx* sur nsibranly.fr .