

TP 3 - Cartes de France en SQL

I Villes de France

Nous allons exploiter la base de données VillesFrance.

Lancer SQLiteStudio. Se connecter à la base de données VillesFrance.s3db, théoriquement présente dans le répertoire TP03. Explorer son contenu.

- Combien cette base de données contient-elle de villes ?
- Combien votre ville contient-elle d'habitants ?

On rappelle que la casse (les majuscules et minuscules) doit être respectée.

Utiliser Python pour se connecter à la base de données VillesFrance.

Rappel : Pour se placer dans le bon répertoire il est conseillé et parfois indispensable de doubler les anti-slash :

```
1 import os
2 os.chdir('chemin ???')
```

```
1 import os
2 #se placer dans le bon répertoire
3 os.chdir("????")
4
5 import sqlite3
6
7 # Connexion à la base :
8 cnx=sqlite3.connect("VillesFrance.s3db")
9
10 # Spécification de l'encodage :
11 cnx.text_factory=lambda x:str(x,"latin-1")
12
13 # Création d'un curseur :
14 curseur=cnx.cursor()
15
16 Req="Select * from villes"
17 # Envoi de la requête :
18 curseur.execute(Req)
19
20 # vider le curseur
21 data = curseur.fetchall()
22 print(data)
23
```

Attention ! Un curseur ne peut être parcouru qu'une seule fois. Sinon il faut relancer la requête.

Q1. Afficher la liste de toutes les villes de France de plus de 100 000 habitants, ainsi que la populations de chacune. Combien y a-t-il de telles villes ?

Q2. Afficher la liste de tous les départements de France ainsi que leur population.

Q3. Créer un graphique où chaque ville de plus de 5 000 habitants est matérialisée par un marqueur. Pour ceci utiliser les attributs `Longitude` et `Latitude`, puis l'instruction :

```
1 plot([x],[y],marker='.')
```

Utiliser un repère orthonormé.

Q4. Demander ensuite les villes de plus de 2000 habitants, voire toutes les villes de plus de 1000 habitants en s'armant de patience.

Le but de la suite est de tracer la carte de France autour.

II Les frontières

On utilise le fichier texte `Coordonnees_France.txt` contenu dans le dossier `Coordonnees_France.zip`. Commencer par décompresser le dossier puis parcourir le dossier. Un double-clic sur son icône depuis Windows montre son contenu.

Il s'agit des coordonnées des points frontières de la France métropolitaine. Ces coordonnées sont au format Lambert 93, système géodésique officiel depuis 2001.

`Coordonnees_France.txt`

```
849460.0 6524534.0
848984.9 6525112.0
849022.0 6527350.0
...
```

Q5. Ouvrir depuis Pyzo le fichier `Coordonnees_France.txt`. Afficher toutes ses lignes.

```
1 fichier=open(...)
2 for L in fichier:
3     print(L)
```

C'est très long! (CTRL I pour stopper, ou la croix sur fond rouge au-dessus du Shell).

Q6. Stocker les données contenues dans le fichier `Coordonnees_France.txt` dans deux listes X et Y de flottants, une pour la première colonne et l'autre pour la seconde. Utiliser la méthode `split()` pour découper et la fonction `float` pour convertir en flottant.

```
1 >>> P="Ceci est une phrase."
2 >>> P.split()
3     ...
4 >>> A="9.6"
5 >>> float(A)
6     ...
```

Q7. Afficher le graphique reliant les points dont les coordonnées ont été stockées dans les listes X et Y.

Tracer ce graphique en noir, dans un repère orthonormé.

Q8. Ajouter les villes de plus de 10 000 habitants en vert.

Q9. Ajouter les préfectures en rouge.

Q10. Demander à l'utilisateur un département, puis afficher toutes les villes de ce département en bleu.

III Compléments

Les exercices suivants peuvent être traités dans le désordre.

III.1 îles

En réalité la France n'est pas un seul bloc, elle possède aussi des îles.

Les fichiers `Coordonnees_France_n.txt` où n va de 1 à 28 contiennent toutes les parties de la France. Le premier contient la partie principale, les autres contiennent les îles.

Tracer à l'aide de ces fichiers la carte complète de France.

III.2 départements

Le fichier `Coordonnees_France_Dept.txt` contient la liste des points frontière des département français, avec également des lignes cut pour des coupures, permettant de tracer les îles.

Coordonnees_France_Dept.txt

Département 1	
849460	6524534
848984.9	6525112
849022	6527350
...	
851178	6525172
849460	6524534
Département 2	
781347	6953816
781338	6953428
...	
Cut	
699288	6901218
699398	6901248
...	

Q11. Tracer la carte des département français.

Q12. Ajouter les préfectures en rouge, les villes de plus de 10 000 habitants en vert, et toutes les villes d'un département choisi en bleu.

IV Bricorama

Ouvrir SQLiteStudio et se connecter à cette base de données, contenue dans le fichier `TP3-Bricorama.s3db` (menu *base de données, ouvrir*).

Cette méthode n'est pas habituelle : on copie rarement une base de données sur son propre ordinateur pour l'exploiter. Le principe est plutôt de se connecter à une base de données distante, laquelle est gérée par un serveur, qui en particulier surveille les différents accès et ordonne les modifications.

La base de données **Bricorama** gère les ventes en ligne d'une entreprise de commerce de matériel de bricolage.

Voici son **schéma relationnel** :

Clients

textttibut	NCli	Nom	Adresse	Ville	Compte	Entreprise
Type	INTEGER	VARCHAR	VARCHAR	VARCHAR	DECIMAL	BOOLEAN

Commandes

textttibut	NCom	DateCom	NClient
Type	INTEGER	DATE	INTEGER

Details

textttibut	NCommande	NProduit	Quantité
Type	INTEGER	VARCHAR(5)	INTEGER

Produits

textttibut	NPro	Libellé	Prix	Stock
Type	VARCHAR(5)	VARCHAR	DECIMAL	INTEGER

IV.1 Quelques exemples d'enregistrements

Clients

NCli	Nom	Adresse	Ville	Compte	Entreprise
0	Jean Martin	18 rue du Férétra	Toulouse	1 200,00	<input type="checkbox"/>
1	Albert Dupond	3 rue Henri II	Paris	-350,00	<input type="checkbox"/>
3	Durand SA	56 avenue de la Berezina	Albi	0,00	<input checked="" type="checkbox"/>

Produits

NPro	Libelle	Prix	Stock
CS262	Chevron Sapin 200x6x2	75	45
CS264	Chevron Sapin 200x6x4	120	2690
PA45	Pointe acier 45	105	580
PA60	Pointe acier 60	95	134
PO214	Tournevis Plat 4mm	11	855

Commandes

NCom	DateCom	NClient
2	30/12/2011	0
3	11/12/2012	1
4	23/10/2013	1

Details

NCommande	NProduit	Quantite
2	PA60	20
3	PA45	25
3	PA60	30

IV.2 Exos

Le commentaire en SQL est indiqué par le double tiret : -- .

- Q13.** Afficher la liste de toutes les entreprises clientes.
- Q14.** Afficher la liste de tous les produits dont le prix est inférieur à 10 euros, sans la quantité de produits en stock.
- Q15.** Compter le nombre de commandes.
- Q16.** Compter le nombre de commandes par client.
- Q17.** Compter le nombre de produits commandés.
- Q18.** Donner, pour chaque référence de produit, la quantité moyenne commandée en un envoi.
- Q19.** Afficher les numéros des clients ayant passé une commande un jour.

Les ordonner (instruction `ORDER BY...` en fin de requête).

Je rappelle qu'il faut ajouter `DISTINCT` après le `SELECT` pour rendre chaque occurrence unique.

Q20. Afficher les numéros des clients qui n'ont jamais passé de commande.

Pour ceci sélectionner la liste des numéros des clients en renommant `Ncli` en `NClient` puis utiliser `EXCEPT`.

Q21. Créer la jointure naturelle des tables `Clients` et `Commandes`.

Q22. Donner la liste (nom, adresse complète, entreprise) des clients de Toulouse ayant passé au moins une commande.

Q23. Compter le nombre de commandes passées depuis chaque ville.

Q24. Créer la jointure des tables `Produits` et `Details`.

Q25. Poursuivre la requête pour obtenir la table contenant les numéros de référence et les libellés des produits, et pour chaque produit le nombre total d'exemplaires commandés. Par renommage on appellera `NComTotal` ce nombre.

Q26. Obtenir les références, les libellés et la quantité commandée des produits commandés en plus de 100 exemplaires.

Q27. Déterminer, pour chaque produit (en indiquant sa référence et son libellé), le nombre de fois où il a été commandé, la quantité minimale, la quantité moyenne et la quantité maximale commandée.

On souhaite dresser les factures des clients. Pour ceci on veut obtenir une table contenant la liste des clients qui ont effectué une ou plusieurs commandes, leur adresse, et la somme qu'ils doivent.

Q28. Créer une requête renvoyant la jointure des trois tables autres que `Clients`, en ne gardant que les colonnes nécessaires.

Q29. Modifier la requête précédente en remplaçant les colonnes `Prix` et `Quantité` par une unique colonne contenant les multiplications `Prix*Quantité`.

Q30. Modifier encore la requête précédente de façon à ne garder que les numéros des clients et la somme qu'ils doivent, sous le nom `PrixTotal`.

Q31. Ajouter la table `Clients` à la jointure. Afficher les noms et adresses complètes des clients, ainsi que la colonne `PrixTotal`.

Q32. Modifier la requête précédente, de façon à tenir compte de la valeur du compte de chaque client, cad soustraire la valeur `Compte` à la valeur `PrixTotal` et l'appeler `APayer`.

Q33. Ne garder que les enregistrements pour lesquels la somme à payer est positive.

On doit obtenir 34 lignes

IV.3 Connexion sous Python

Utiliser le module `sqlite3` pour créer une connexion à une base de données et envoyer des requêtes, comme pour l'exercice des villes de France.

Q34. Créer la liste de toutes les villes contenant un adhérent au site `Bricorama`.

Q35. Afficher, pour chaque ville, le nom de la ville, le nombre de clients, la liste des noms des clients et le nombre de commandes pour chaque client.

```
1 Toulouse : 18 clients
2     Jean Martin : 9 commandes
3     ...
4 Paris : 4 clients
5     Albert Dupont : 5 commandes
6     ...
```