

Base de Données sous Python

Package sqlite3

Traitement des BDD

Le langage SQL peut être directement utilisé dans des logiciels comme « Sqlite browser » gratuit et disponible ici : <http://sqlitebrowser.org/>

Je vous recommande d'importer un fichier .csv en faisant :

- Création d'une nouvelle bdd et enregistrement
- Ne pas remplir de tables (Cancel)
- Fichier – Importer – Table vers un fichier csv – Afficher toutes les extensions – Ouvrir le fichier
- Choisir les bonnes options d'importation en visualisant le résultat
- Ecriture des requêtes dans l'onglet « Exécuter le SQL »

Forme générale de la manipulation des BdD sous Python

```
import sqlite3
BDD = sqlite3.connect('BDD.db')
cursor = BDD.cursor()
cursor.execute("..."")
BDD.commit()
BDD.close()
```

... sera la requête SQL

Afficher le dernier résultat de requête caché dans « cursor »

```
Resultats = cursor.fetchall() puis print(Resultats)
Resultat = cursor.fetchone() puis print(Resultat)
```

Gestion de la bdd	
Objet	Requête SQL
Création table	<code>CREATE TABLE Eleves(id INTEGER PRIMARY KEY,Prenom TEXT,Nom TEXT,Classe TEXT,Age INTERGER,Ville TEXT) ;</code>
Lister relations A écrire tel quel	<code>SELECT name FROM sqlite_master WHERE type='table';</code>
Lister attributs	<code>PRAGMA table_info(Eleves);</code>
Supprimer table	<code>DROP TABLE Eleves;</code>
Insertion avec objet	<pre>cursor.execute("""INSERT INTO Eleves VALUES (?,?,?,?,?,?) ;""",(1,"Donald","Knuth","TNSI",16,"Pertuis"))</pre> <p>Liste_Eleves = [(1,"Donald","Knuth","TNSI",16,"Pertuis"), ...] cursor.executemany("""INSERT INTO Eleves VALUES (?,?,?,?,?,?);""",Liste_Eleves)</p>
Modification d'un enregistrement	<code>UPDATE Eleves SET Nom = New_Nom, Prenom = New_Prenom WHERE id = 2 ;</code>
Suppression d'un enregistrement	<code>DELETE FROM Eleves WHERE id = 3 ;</code>

Requêtes usuelles	
	<code>SELECT ... FROM ... WHERE ...;</code>
Projection Attributs (colonnes) SELECT FROM	<pre>SELECT * FROM Eleves ; SELECT Nom,Prenom FROM Profs ; SELECT DISTINCT Age FROM Eleves ;</pre> <p><i>ORDER BY colonne (classement ascendant) DESC (pour descendant) LIMIT i ou FETCH FIRST 1 ROWS ONLY (pour ne récupérer que les i premiers résultats)</i></p>
Sélection/Restriction Enregistrements (lignes) WHERE	<pre>SELECT Nom,Prenom FROM Eleves WHERE Classe = "TNSI" ; SELECT DISTINCT Age FROM Eleves WHERE Classe LIKE "TNSI" OR Classe LIKE "1NSI" ; SELECT Nom FROM Eleves WHERE Age BETWEEN 16 AND 19 ;</pre>
Jointure Entre relations JOIN ON	<pre>SELECT Eleves.Prenom, Eleves.Nom, Profs.Salle FROM Eleves JOIN Profs ON Eleves.Classe = Profs.Classe ; SELECT DISTINCT Profs.Salle FROM Eleves JOIN Profs ON Eleves.Classe = Profs.Classe WHERE Eleves.Classe = "TNSI" ;</pre>

Minimum MIN()	SELECT MIN(Age) FROM Eleves ; SELECT Nom,MIN(Age) FROM Eleves ;
Maximum MAX()	SELECT MAX(Age) FROM Eleves ;
Somme SUM()	SELECT SUM(Age) FROM Eleves ;
Moyenne AVG()	SELECT AVG(Age) FROM Eleves ;
Comptage COUNT	SELECT COUNT(*) FROM Eleves ; SELECT COUNT(*) FROM Eleves WHERE Age = 16 ;
Utilisation du résultat sous Python	Res = cursor.fetchone() print(Res[0])

Création d'une relation sous Python à l'aide d'un résultat de requête
<pre># Réalisation d'une requête cursor.execute("""""") Res_OP = cursor.fetchall() # Nouvelle base de données cursor.execute("""CREATE TABLE T_OP(Ref Text,Prix Integer);""") cursor.executemany("""INSERT INTO T_OP VALUES (?,?);""",Res_OP) # Affichage du résultat cursor.execute("""SELECT * FROM T_OP;""") print(cursor.fetchall())</pre>