

## 1.5

## SQL et Python

NSI TERMINALE - JB DUTHOIT

nous allons ici travailler sur les requêtes SQL effectuées depuis un programme Python.

### 1.5.1 Connexion à une base de données et création de tables

Nous allons ici vous donner un code qui permet de se connecter à une base de données, ou de la créer si elle n'existe pas encore.

Pour plus de facilité, il est conseillé de créer un répertoire spécifique dans lequel on crée un fichier .py contenant :

```
# Importation de la bibliothèque
import sqlite3
#On se connecte ou on crée "bdd.db"
#On crée un objet de type connexion
conn = sqlite3.connect('bdd.db')
# On crée un objet de type cursor
# cur va permettre de manipuler la bdd
cur = conn.cursor()
#Création de la table
cur.execute("CREATE TABLE IF NOT EXISTS LIVRES(id INT,
titre TEXT, ann_publi INT)")
# Pour effectuer les requêtes
conn.commit()
# On ferme les objets
cur.close()
conn.close()
```

#### Remarque

| IF NOT EXISTS permet d'éviter une erreur si l'on tente de créer une table qui existe déjà.

☞ Vérifier que la base de données à bien été créée !

La méthode est toujours la même :

- Création d'un objet de type "connection"
- Création d'un objet de type "cursor"
- préparation d'une ou plusieurs requête(s) (méthode "execute" sur l'objet de type "cursor")
- Exécution réelle des requêtes (méthode "commit" sur l'objet de type "connection")
- fermeture de l'objet de type "cursor"
- Fermeture de l'objet de type "connection"

### 1.5.2 Ajouter des données

```
cur.execute("INSERT INTO LIVRES
(id,titre,ann_publi) VALUES(1,'1984',1949)")
```

☞ Vérifier que les données ont bien été ajoutées !

Il est possible de séparer la requête des données à ajouter :

#### ajout d'un enregistrement

```
import sqlite3
conn = sqlite3.connect('bdd.db')
cur = conn.cursor()
donnees = ('France',67,643)
cur.execute("CREATE TABLE IF NOT EXISTS PAYS(nom TEXT,population INT, superficie INT)")
cur.execute("INSERT INTO LIVRES VALUES(?,?,?)",donnees)
conn.commit()
cur.close()
conn.close()
```

#### ajout de plusieurs enregistrements

```
import sqlite3
conn = sqlite3.connect('bdd.db')
cur = conn.cursor()
donnees = [('France',67,643),('Irlande',5,70)]
cur.execute("CREATE TABLE IF NOT EXISTS PAYS(nom TEXT,population INT, superficie INT)")
cur.executemany("INSERT INTO PAYS VALUES(?,?,?)",donnees)
conn.commit()
cur.close()
conn.close()
```

### 1.5.3 Création d'un clé primaire avec incrémentation automatique

```
import sqlite3

conn = sqlite3.connect('bdd.db')
cur = conn.cursor()

donnees = [
    ('Baudelaire', 'Charles',1821, 'France'),
    ('Camus', 'Albert', 1913, 'Algérie')
]

cur.execute("CREATE TABLE IF NOT EXISTS AUTEURS
(id INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE,
 nom TEXT, prenom TEXT, ann_naiss TEXT, pays_naiss TEXT)")
cur.executemany("INSERT INTO AUTEURS(nom,prenom,ann_naiss,pays_naiss)
VALUES(?, ?, ?, ?)", donnees)
conn.commit()
cur.close()
```

```
conn.close()
```

☞ Vérifier que les données ont été ajoutées et que la clé primaire aussi, de façon automatique !

	id	nom	prenom	ann_naiss	pays_naiss
	Filtre	Filtre	Filtre	Filtre	Filtre
1	1	Baudelaire	Charles	1821	France
2	2	Camus	Albert	1913	Algérie



### Exercice 1.29

| Ajoutez une nouvelle données dans cette base AUTEURS

## 1.5.4 Modification de tables

Il est possible de modifier des données déjà présentes dans la table. A vous d'essayer !

## 1.5.5 Suppression d'une donnée

```
import sqlite3

conn = sqlite3.connect('bdd.db')
cur = conn.cursor()

suppr = ('Camus',)
cur.execute('DELETE FROM AUTEURS WHERE nom = ?', suppr)
conn.commit()

cur.close()
conn.close()
```

## 1.5.6 Utilisation de SELECT

```
import sqlite3
conn = sqlite3.connect('bdd.db')
cur = conn.cursor()
recherche = (50, 30)
cur.execute('SELECT nom FROM PAYS WHERE population < ? AND superficie > ?', recherche)
conn.commit()
liste = cur.fetchall()
cur.close()
conn.close()
```

### Remarque

| La méthode "fetchall" d'un objet de type "cursor" renvoie un tableau contenant des tuples

☞ Vérifier la valeur référencée par la variable liste.