

Chapitre 3 — SQLite + Python (sqlite3)

1. C'est quoi ?

`sqlite3` est le module natif de Python pour interagir avec une base SQLite. Il permet d'exécuter des commandes SQL en Python **sans serveur externe**.

2. Pourquoi on l'utilise ?

- Pour **gérer une base locale légère**, sans setup compliqué
 - Pour **connecter** Python à SQL facilement (CRUD, scripts, UI)
-

3. Comment ça fonctionne ?

- On importe le module : `import sqlite3`
 - On ouvre la base (ou on la crée si elle n'existe pas)
 - On crée un **curseur** SQL pour exécuter les requêtes
 - On exécute les requêtes avec `execute()`
 - On récupère les résultats avec `fetchall()` ou `fetchone()`
 - On valide les modifs avec `commit()`
-

4. Vocabulaire technique

Terme	Rôle
<code>connect(path)</code>	ouvre une base SQLite locale
<code>cursor()</code>	crée un exécuteur de requêtes SQL
<code>execute(req, val)</code>	envoie une commande SQL (sécurisée)
<code>fetchall()</code> / <code>fetchone()</code>	récupère les résultats
<code>commit()</code>	valide une opération
<code>rollback()</code>	annule en cas d'erreur
<code>with</code>	garantit la fermeture auto de la base

5. Exemple complet

```
import sqlite3

DB_PATH = "./data/machineMonitor.db"
```

```
with sqlite3.connect(DB_PATH) as conn:  
    cursor = conn.cursor()  
    cursor.execute("SELECT * FROM machines WHERE in_service = ?", (1,))  
    results = cursor.fetchall()
```

✓ Résultat :

Tu peux interagir avec une base SQL **directement en Python**, en toute sécurité, sans serveur externe, en mode fichier local.