

# Chapitre 2 — Commandes SQL : DDL & DML

---

## 1. C'est quoi ?

Les commandes SQL permettent d'interagir avec la base de données :

- DDL (Data Definition Language) → pour créer/modifier la structure
  - DML (Data Manipulation Language) → pour ajouter/lire/modifier/supprimer les données
- 

## 2. Pourquoi on les utilise ?

- Pour **définir les tables** (structure, types, contraintes)
  - Pour **manipuler les données** : créer, consulter, modifier, supprimer
- 

## 3. Vocabulaire essentiel

Terme	Type	Rôle
CREATE TABLE	DDL	crée une nouvelle table
DROP TABLE	DDL	supprime une table
ALTER TABLE	DDL	modifie une table existante
INSERT INTO	DML	ajoute une ligne
SELECT	DML	lit des données
UPDATE	DML	modifie une ou plusieurs lignes
DELETE	DML	supprime une ou plusieurs lignes

---

## 4. Syntaxes de base (rappels)

```
-- Créer une table
CREATE TABLE machines (...);

-- Ajouter une ligne
INSERT INTO machines (name, sector, ...) VALUES (...);

-- Lire toutes les machines
SELECT * FROM machines;

-- Lire avec condition
SELECT * FROM machines WHERE in_service = 1;
```

```
-- Modifier une ligne
UPDATE machines SET sector = '1A' WHERE name = 'M123';

-- Supprimer une ligne
DELETE FROM machines WHERE name = 'M123';
```

## 5. Sécurité (dans Python)

- Utiliser les `?` pour injecter les valeurs → évite les injections SQL

```
cursor.execute("SELECT * FROM machines WHERE name = ?", (name,))
```

## 6. Transactions

- Chaque modification doit être **validée** par un `commit()`
- Si une erreur arrive → possibilité de faire `rollback()`

```
conn = sqlite3.connect(...)
cursor = conn.cursor()
cursor.execute("UPDATE ...")
conn.commit()
```

### ✓ Résultat :

Tu peux maintenant **créer, lire, modifier et supprimer** des données dans une base SQL avec un minimum de commandes pro, sécurisées et fiables.