

Chapitre 1 — Modélisation relationnelle (SQL + SQLite)

1. C'est quoi ?

La **modélisation relationnelle** est la méthode pour structurer des données dans une base SQL sous forme de **tables liées entre elles** par des clés (primaires et étrangères).

2. Pourquoi on l'utilise ?

- Pour **organiser** les données proprement.
 - Pour **relier** des entités entre elles (ex : une machine a plusieurs logs).
 - Pour **garantir l'intégrité** des données (éviter les doublons, erreurs, liens cassés).
-

3. Comment ça fonctionne ?

Une base relationnelle = **plusieurs tables**, avec :

- des **champs** bien typés (texte, nombre, booléen)
 - une **clé primaire (PK)** = identifiant unique d'une ligne
 - une **clé étrangère (FK)** = lien vers une autre table
-

4. Vocabulaire essentiel

Terme	Rôle
TABLE	conteneur de données (équivalent d'un fichier Excel)
COLUMN	un champ dans la table (nom, type, etc.)
PRIMARY KEY	identifiant unique de chaque ligne
FOREIGN KEY	référence une ligne dans une autre table
NOT NULL	champ obligatoire (ne peut pas être vide)

5. Syntaxe et construction

```
CREATE TABLE IF NOT EXISTS machines (  
  name TEXT PRIMARY KEY,  
  sector TEXT NOT NULL,  
  serial_number TEXT NOT NULL,  
  year_of_acquisition INTEGER NOT NULL,  
  in_service BOOLEAN NOT NULL,  
  comment TEXT
```

```
);

CREATE TABLE IF NOT EXISTS logs (
  uuid TEXT PRIMARY KEY,
  machineName TEXT NOT NULL,
  type TEXT NOT NULL,
  project TEXT NOT NULL,
  timeStamp TEXT NOT NULL,
  userName TEXT NOT NULL,
  modifications TEXT,
  FOREIGN KEY(machineName) REFERENCES machines(name)
);
```

6. Représentation visuelle

```
MACHINES
-----
name (PK)
sector
serial_number
year_of_acquisition
in_service
comment

LOGS
----
uuid (PK)
machineName (FK → machines.name)
type
project
timeStamp
userName
modifications
```

✓ Résultat :

Deux tables **liées entre elles**, prêtes à recevoir des données cohérentes. Chaque log est attaché à une machine, et chaque machine est unique via sa clé `name`.