

## Un SGBDR, c'est quoi ?

Avant d'ouvrir un outil et de créer des tables, on pose le vocabulaire : qu'est-ce qu'une base de données relationnelle, qui s'en occupe, et comment on s'y connecte ?

4TTR

 niveau

Jusqu'ici, tu as travaillé sur du papier : entités, relations, cardinalités, identifiants. C'est le **modèle**. La prochaine étape est de transformer ce modèle en une vraie base de données qui tourne sur ton ordinateur.

Avant de toucher à un outil, il faut comprendre **ce qui s'exécute** et **avec quoi on lui parle**.



## Du schéma aux tables


Dans une **base de données relationnelle**, les informations sont stockées dans des **tables**. Une table, c'est essentiellement un tableau avec :

- des **colonnes** – chacune correspond à un attribut (nom, prix, date...)
- des **lignes** – chacune représente une occurrence (un client précis, un ticket précis...)

La traduction depuis le MCD est directe :

MCD	Base de données
Entité ( CLIENT , TICKET ...)	Table
Attribut ( nom , prix ...)	Colonne
Une occurrence	Une ligne
Identifiant	Clé primaire (la colonne id )

Le **R** de SGBDR – *relationnel* – vient du fait que ces tables sont **reliées entre elles** par des colonnes spéciales appelées **clés étrangères**. Tu verras la mécanique exacte dans le prochain cours.

 Une base de données relationnelle, c'est donc un ensemble de tables qui se référencent mutuellement de façon contrôlée.



## Le logiciel qui gère tout ça : le SGBDR

Les tables ne flottent pas dans le vide. Elles sont gérées par un programme : un **SGBDR** – *Système de Gestion de Base de Données Relationnelle* (GB **RDBMS, Relational Database Management System**).

Son boulot :

- **stocker** les tables sur le disque
- **garantir** que les contraintes sont respectées (pas de doublons sur une clé primaire, pas de clé étrangère qui pointe vers le vide...)
- **exécuter** les requêtes qu'on lui envoie pour lire, ajouter, modifier ou supprimer des données
- **gérer** les accès concurrents (plusieurs personnes ou programmes en même temps)

## Les principaux SGBDR

SGBDR	Particularité
<b>MySQL</b>	Le plus répandu sur le web, racheté par Oracle
<b>MariaDB</b>	Fork libre de MySQL, presque 100 % compatible – c'est ce que Laragon installe par défaut
<b>PostgreSQL</b>	Très complet, réputé pour sa rigueur
<b>SQLite</b>	Pas un serveur : un simple fichier <code>.db</code> , idéal pour les petites apps
<b>SQL Server</b>	Solution Microsoft, courante en entreprise
<b>Oracle Database</b>	Historique, très présent en grande entreprise

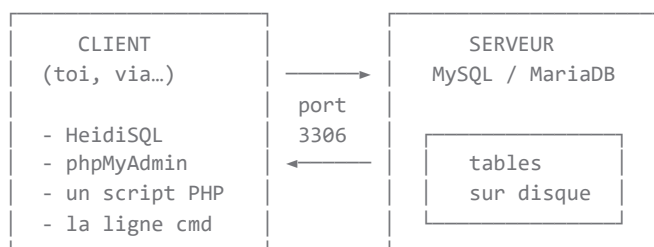
Dans ce cours, on utilise **MySQL / MariaDB** : c'est le SGBDR de loin le plus courant chez les hébergeurs web, et c'est celui que Laragon (ou XAMPP) a installé sur ta machine.

## Le modèle client / serveur

Voici le point qui surprend souvent au début.

Quand tu démarres Laragon, un programme se lance en arrière-plan : c'est le **serveur MySQL** (`mysqld`, le *d* pour *daemon*). Il tourne en permanence, écoute sur un **port réseau** (par défaut le `3306`) et attend qu'on lui parle.

Toi, tu ne lui parles **jamais directement**. Tu utilises un **client** – un autre programme qui se connecte au serveur et lui transmet tes ordres.



### Conséquences pratiques :

- Si le service MySQL de Laragon n'est pas démarré, **aucun** client ne pourra se connecter – même si HeidiSQL est ouvert.

- Plusieurs clients peuvent être connectés en même temps au même serveur, sur la même base.
- Le client peut être sur la même machine que le serveur (chez toi pour l'apprentissage) ou à l'autre bout du monde (sur un hébergement web).

## Les clients qu'on va utiliser

### HeidiSQL – le client desktop

Application Windows gratuite, interface graphique riche. On peut créer des bases, des tables, taper des requêtes SQL, voir et éditer les données comme dans un tableur. C'est l'outil de tous les jours sur ta machine.

### phpMyAdmin – le client web

Une application web (écrite en PHP) qui tourne sur le serveur web et expose une interface dans le navigateur. Quasiment **tous les hébergements mutualisés** la proposent – c'est souvent le seul moyen d'administrer une base distante sans installer de logiciel.



#### À RETENIR

: HeidiSQL et phpMyAdmin sont **deux clients différents qui parlent au même serveur MySQL**. Tout ce que tu crées avec l'un est visible avec l'autre – c'est la même base de données.

### Les autres clients (pour info)

- **La ligne de commande** `mysql` – installée avec Laragon, utile pour les scripts
- **Un programme** – n'importe quel langage (PHP, Python, Node.js, Java...) peut se connecter à MySQL via une bibliothèque dédiée. C'est ce qu'on fera plus tard quand on connectera un site web à sa base.

## La langue commune : SQL

Quel que soit le client, on parle au serveur dans le même langage : **SQL** (*Structured Query Language*).

Exemples – pas besoin de comprendre la syntaxe maintenant, c'est juste pour voir à quoi ça ressemble :

```
1  -- Créer une table
2  CREATE TABLE client (
3      id      INT NOT NULL AUTO_INCREMENT,
4      nom     VARCHAR(100),
5      prenom  VARCHAR(100),
```

```
6     PRIMARY KEY (id)
7 );
8
9 -- Ajouter une ligne
10 INSERT INTO client (nom, prenom) VALUES ('Dupont', 'Marie');
11
12 -- Lire les données
13 SELECT * FROM client WHERE nom = 'Dupont';
```

Les interfaces graphiques (HeidiSQL, phpMyAdmin) **gènèrent du SQL pour toi** quand tu cliques. Tu apprendras à écrire ce SQL directement dans les cours qui suivront – c'est une compétence indispensable dès qu'on quitte les interfaces toutes faites.

## À retenir

- Une **base de données relationnelle** est un ensemble de **tables** reliées par des **clés**.
- Le **SGBDR** est le logiciel qui stocke ces tables et qui exécute les requêtes – ici MySQL ou MariaDB.
- Le SGBDR fonctionne en mode **client / serveur** : un serveur tourne en permanence, des clients s'y connectent.
- **HeidiSQL** (desktop) et **phpMyAdmin** (web) sont deux clients différents pour le même serveur.
- Tous les clients parlent la même langue : **SQL**.

## Vérifications

1. Sur ta machine, qu'est-ce qui est *serveur* et qu'est-ce qui est *client* ?
2. Si on ferme HeidiSQL, est-ce que la base de données est perdue ? Pourquoi ?
3. Pourquoi un hébergement web propose-t-il phpMyAdmin plutôt que HeidiSQL ?
4. Cite trois SGBDR et donne un cas d'usage typique pour chacun.

## Suite

Maintenant que le décor est planté, on peut faire l'étape qui manque entre ton MCD papier et de vraies tables : traduire le schéma entité-association en **modèle relationnel**, avec ses clés primaires et ses clés étrangères.