

mettre **bd-init.zip** sur site

## 1 Problématique

Le modèle relationnel présenté dans le cours précédent est un modèle mathématique qu'il faut maintenant concrétiser sur machine.

Quels sont les outils permettant de construire une base de données ?

## 2 Organisation

### 2.1 Un logiciel

Un **Système de Gestion de Base de Données (SGBD)** est un logiciel permettant de manipuler les données d'une base de données.

Ils sont la plupart du temps basés sur un modèle client-serveur :

- la base de données se trouve sur un *serveur*,
- un *logiciel client* va interroger le serveur et transmettre la réponse que ce-dernier lui aura donné.



FIGURE 1 – Principaux SGBD

Un SGBD qui implémente le modèle relationnel est noté SGBDR.

Contexte historique :

- avant 1960 stockage sur bande magnétique → traitement des données était réalisé en lots (rapatrier données en mémoire, traitement, retour sur bande). L'arrivée des stockages à accès direct (disquette, disque dur) change la façon de traiter les données
- 1970 Codd : modèle relationnel. Remplace système hiérarchique mis en place entre 1960-1970 peut satisfaisant
- 1974 dvp chez IBM par Donald Chamberlin ; normalisé en 1986 par ISO (International Organization for Standardization)
- 1985 PostgreSQL : logiciel libre fondé sur une communauté mondiale de développeurs et d'entreprises
- 1995 MySQL : GNU General Public License (logiciel libre) ; le + utilisé au monde
- 2000 : SQLite : n'utilise pas système client-serveur. Au début devait être utilisé dans missiles embarqués. Logiciel libre.
- 2008 : Mysql → Sun pour 1 milliard \$

- 2009 : MariaDB : Mickael Widenius (My et Maria = nom de ses enfants) ; adopté par de grands groupes (Google, Wikipedia...)

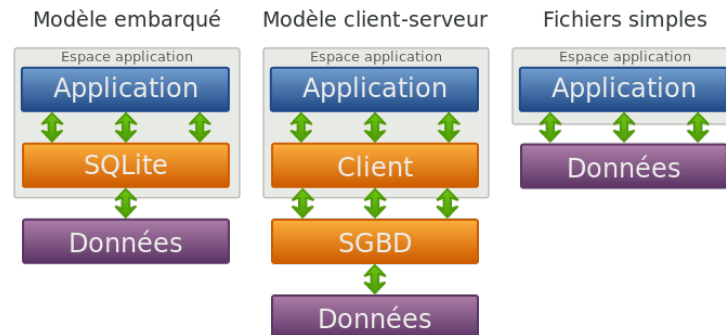


FIGURE 2 – Modèles d'accès aux données

## 2.2 Un langage

Les SGBD stockent et optimisent les données de manière efficace mais très complexe. Il n'est pas possible d'y accéder directement. Il faut effectuer des **requêtes** à l'aide d'un langage adapté.

### À retenir

Le **SQL (Structured Query Language)** est utilisé dans une écrasante majorité des SGBDR.

Pour la construction d'un site web dynamique, les pages web effectuent des requêtes SQL au serveur MySQL afin de récupérer des données.

Concurrent : QBE (Query By Example) utilisé dans Microsoft Access (SGBDR de type "fichier")  
modèle "fichier simple" : traitement de texte ; .doc, .odt = xml empaqueté

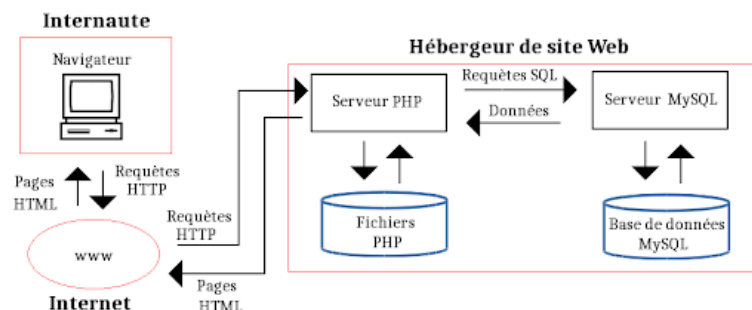


FIGURE 3 – Répartition des rôles

## 2.3 Application

Pour des contraintes d'installation, nous utiliserons dans un premier temps, un modèle SQLite.

**Activité 1 :**

1. Télécharger et extraire la version portable de *DB Browser for SQLite* depuis le site officiel <https://sqlitebrowser.org/dl/>
2. Télécharger et extraire la base *bd-init.zip* depuis le site <https://cviroulaud.github.io>
3. Ouvrir la base avec le browser.
4. Se concentrer d'abord sur l'onglet *Parcourir les données* et observer les tables existantes.

La table `sqlite_sequence` est interne au logiciel.

### 3 Contraintes d'intégrité

Le modèle relationnel établit des contraintes qui garantissent l'intégrité des données.

#### 3.1 Contrainte de domaine

Aux domaines abstraits du modèle relationnel correspondent les types de données du langage SQL.

Nom du type	Description
SMALLINT	Entier 16 bits signé
INT	Entier 32 bits signé
BIGINT	Entier 64 bit signé
REAL	Flottant 32 bits
CHAR(n)	Chaîne de n caractères exactement
VARCHAR(n)	Chaîne d'au plus n caractères
TEXT	Chaîne de taille quelconque
DATE	Date au format AAAA-MM-JJ
TIME	Heure au format hh:mm:ss
TIMESTAMP	Instant au format AAAA-MM-JJ hh:mm:ss

TINYINT = 1 octet = 0 à 255

les flottants sont des valeurs approchées

**DOUBLE PRECISION** : float 64

**DECIMAL** : représente nb à virgules de manière exacte

Il existe un **BOOLEAN** mais inégalement supporté par les SGBD (car pas obligatoire dans les spécifications SQL) ; pour faire un booléen : CHAR(1)

SMALLINT 2 octets =  $-2^{15}2^{15} - 1$  ; on peut préciser non signé en MySQL (pour id)'

**Activité 2 :**

1. Quelle est la valeur maximale que peut prendre un SMALLINT ?
2. Quelle sa taille en mémoire ?
3. Dans le browser, se rendre dans l'onglet *Structure de la base de données*.
4. Dérouler la table *Auteurs* et repérer les types de chaque attribut.

Le SGBD Sqlite simplifie les types (INTEGER, REAL, TEXT) en l'adaptant dynamiquement en fonction de la valeur stockée.

### 3.2 Contrainte d'entité

Chaque entité est identifiée de manière unique grâce à la *clé primaire*.

#### Activité 3 :

1. Dans le schéma de la table *Auteurs* comment identifie-t-on la clé primaire ?
2. Quel est le rôle du mot clé *AUTOINCREMENT* ?

### 3.3 Contrainte de référence

Afin de garantir la cohérence des données lors de modifications, on utilise une *clé étrangère*. C'est une référence à une clé primaire d'une autre relation.

#### Activité 4 :

1. Dérouler la table *Bandes\_dessinees*.
2. Rappeler les attributs qui sont des clés étrangères.
3. Glisser la souris sur le schéma de cette table. Quels mots clés sont utilisés pour créer une clé étrangère ?

### 3.4 Compléter la base de données

Afin de pouvoir gérer les emprunts de bandes dessinées, il faut créer les tables *Emprunteurs* et *Emprunts*.

**Activité 5 :** Depuis l'onglet *Exécuter le SQL*, créer ces deux tables, en prenant modèle sur les schémas des relations existantes.

#### Remarque

Le langage SQL est insensible à la casse. Nous pouvons écrire indifféremment *CREATE* ou *CreaTE*. Il est d'usage d'écrire les instructions SQL en majuscules.