


|   |  |   |
|---|--|---|
|  | Sujet  | 6/12/2021   |
|   | Découverte SQL<br>Lycée Jules Haag – salle 262 |  |

## Connaissances théoriques

### Qu'est-ce qu'une base de données ?

Pour votre apprentissage de la structuration de données, il vous faut maîtriser un élément important : les bases de données.

Une base de données permet d'enregistrer des données de façon organisée et hiérarchisée.

Certes, vous connaissez les variables, mais celles-ci restent en mémoire **seulement le temps de la génération de la page**. Les bases de données constituent le meilleur moyen de stocker des données de façon simple et propre.

Les SGBD (**S**ystèmes de **G**estion de **B**ases de **D**onnées) sont les programmes qui se chargent du stockage de vos données.

Les plus connus sont :

- #
- **MySQL** : libre et gratuit, c'est probablement le SGBD le plus connu. Nous l'utiliserons dans cette partie ;
  - **MariaDB** : c'est un clone (on dit *fork*) de MySQL, que des gens ont voulu lancer depuis que MySQL a été racheté par... Oracle. MariaDB et MySQL sont donc quasiment le même SGBD.
  - **PostgreSQL** : libre et gratuit comme MySQL, avec plus de fonctionnalités mais un peu moins connu ;
  - **SQLite** : libre et gratuit, très léger mais très limité en fonctionnalités ;
  - **Oracle** : utilisé par les très grosses entreprises ; sans aucun doute un des SGBD les plus complets, mais il n'est pas libre et on le paie le plus souvent très cher ;
  - **Microsoft SQL Server** : le SGBD de Microsoft.

Dans cette activité, nous travaillerons avec MariaDB. Pour communiquer avec MariaDB, on utilise le langage SQL (*Structured Query Language*).

Voici un aperçu des requêtes que l'on peut faire en SQL à une base de données :

- ⇒ « Récupère-moi les 10 derniers news de mon site »
- ⇒ « Supprime le dernier message posté dans ce forum ».

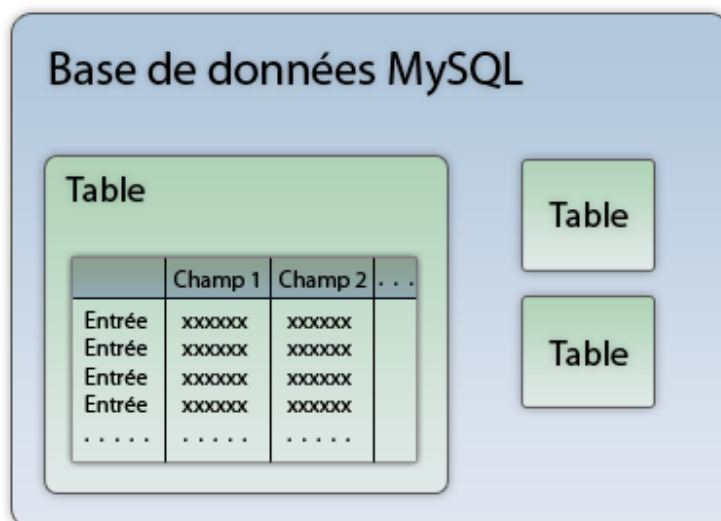
## Structure d'une base de données

Avec les bases de données, il faut utiliser un vocabulaire **précis**. Pour cela, on va se servir d'une image : celle d'une armoire.

- L'armoire est appelée **la base** dans le langage SQL. C'est le gros meuble dans lequel les secrétaires ont l'habitude de classer les informations.
- Dans une armoire, il y a plusieurs tiroirs. Un tiroir, en SQL, c'est ce qu'on appelle **une table**. Chaque tiroir contient des données différentes. Par exemple, on peut imaginer un tiroir qui contient les pseudonymes et infos sur vos visiteurs, un autre qui contient les messages postés sur votre forum...
- Mais que contient une table ? C'est là que sont enregistrées les données, sous la forme d'un tableau. Dans ce tableau, les colonnes sont appelées **des champs**, et les lignes sont appelées **des entrées**.

Une table est donc représentée sous la forme d'un tableau ; par exemple, le tableau suivant vous montre à quoi peut ressembler le contenu d'une table appelée « Visiteurs ».

| Numéro | Pseudonyme    | E-mail                        | Âge |
|--------|---------------|-------------------------------|-----|
| 1      | Kryptonik     | kryptonik@free.fr             | 24  |
| 2      | Serial_Killer | serialkiller@unitedgamers.com | 16  |
| 3      | M@teo21       | top_secret@siteduzero.com     | 18  |
| 4      | Bibou         | bibou557@laposte.net          | 29  |
| ...    | ...           | ...                           | ... |



## *Où sont les bases de données ?*

Pour qu'une base de données puisse exister, elle doit se situer sur un serveur capable de faire tourner un service SQL (par exemple MySQL). Comme pour le PHP, ce serveur peut être simulé localement (WAMP, XAMP, MAMP).

## *Gestion des bases de données*

Pour gérer les bases de données, on peut au choix :

- Le faire avec des lignes de commande (via la console)
- Faire appel à un programme informatique (script Linux par exemple)
- Exécuter les requêtes depuis un programme (PHP, Python, etc.).
- Utiliser une interface d'administration de bases de données (PhpMyAdmin)

Dans cette activité, nous utiliserons en premier l'interface d'administration phpMyAdmin pour créer les bases de données puis nous verrons comment les utiliser avec les requêtes en Python.

---

## Instructions de l'activité

---

### Créer une table

Ouvrir phpMyAdmin :

- Ouvrir « <http://localhost> » sur un navigateur WEB => **SI LA BASE DE DONNEES EST LOCALE** puis cliquer sur phpMyAdmin  
Ou bien taper l'adresse du serveur suivi de </phpmyadmin> => **SI LA BASE DE DONNEES EST SUR UN SERVEUR.**

Remarque : Sur WAMP, le nom d'utilisateur par défaut est 'root', et le mot de passe est root également. Si ça ne fonctionne pas, essayer 'root' et ' ' (rien).

Pour le serveur SQL, utiliser les identifiants fournis.

Sur la gauche, vous pouvez voir les bases de données déjà existantes sur votre serveur WEB. Par défaut, des tables sont créées (mysql, performance\_schema, information\_schema et sys).

**Il ne faut surtout pas les toucher.**

Nous allons créer une table permettant de lister les utilisateurs d'un site ainsi que leurs droits (super utilisateur, ou simple utilisateur).

- Une base de données nommée « prenom.nom » a déjà été créée.
- Cliquer sur la base de données à gauche. Il s'affiche « *Aucune table n'a été trouvée dans cette base de données* ».
- Dans l'encart « *nouvelle table* », rentrer les informations pour créer une nouvelle table : Nom : « *utilisateurs* », nombre de colonnes : 3.

La table est immédiatement créée : il faut maintenant indiquer le nom des champs, et les données qu'ils peuvent contenir.

- Créer les trois champs suivants dans la colonne « *Nom* » :
  - ⇒ **Id** : comme bien souvent, vous allez devoir créer un champ appelé id. C'est le numéro d'identification. Grâce à lui, toutes vos entrées seront numérotées, ce qui est bien pratique
  - ⇒ **Login** : ce champ contiendra l'identifiant d'un utilisateur
  - ⇒ **Password** : enfin, ce champ contiendra le mot de passe de l'utilisateur

Pour chacun des champs, phpMyAdmin demande plusieurs informations (ne rien faire ici, juste lire) :

- ✓ **Type** : le type de données que va stocker le champ (nombre entier, texte, date...). Le lien suivant explique en détail chaque type : [https://www.w3schools.com/sql/sql\\_datatypes.asp](https://www.w3schools.com/sql/sql_datatypes.asp)
- ✓ **Taille/Valeurs** : permet d'indiquer la taille maximale du champ, utile pour le type VARCHAR notamment, afin de limiter le nombre de caractères autorisés
- ✓ **Valeur par défaut** : comme son nom l'indique, permet de définir une valeur par défaut si le champ n'est pas rempli par l'utilisateur.
- ✓ **Interclassement** : l'interclassement est l'encodage utilisé pour la page. Il faut choisir un encodage permettant d'utiliser les caractères spéciaux de la langue française, utf8\_unicode\_ci qui permet de prendre en compte les ligatures (Œ), les accents, etc.
- ✓ **Attributs** : permet de choisir une option pour le type choisi par exemple.
- ✓ **Null** : permet d'autoriser ou non la valeur NULL c'est-à-dire rien comme valeur. Un si cette case est cochée, l'utilisateur pourrait ne rien rentrer dans un champ et valider.
- ✓ **Index** : active l'indexation du champ. Cela signifie que votre champ sera adapté aux recherches. Le plus souvent, on utilise l'index PRIMARY sur les champs de type id, puis rien dans les autres.
- ✓ **AUTO\_INCREMENT** : permet au champ de s'incrémenter tout seul à chaque nouvelle entrée. On l'utilise fréquemment sur les champs de type id.

### En pratique, seules les colonnes « Type, Taille/Valeurs, Index et AUTO\_INCREMENT » sont renseignées.

Remarque :

- ⇒ **Toute table doit posséder un champ qui joue le rôle de *clé primaire*.** La clé primaire permet d'identifier de manière unique une entrée dans la table. En général, on utilise le champ id comme clé primaire, comme on vient de le faire.

A gauche de votre écran, la table que vous venez de créer devient visible. Si vous cliquez dessus, elle s'affiche à droite de l'écran. Il est possible de rajouter, modifier ou supprimer des champs à tout moment.

- Jeter un œil aux onglets du haut : « structure », Parcourir », « SQL », etc. Cela vous amènera vers différentes pages d'options qui vous seront utiles plus tard. Terminer cette excursion par l'onglet « *insérer* »

Sur cette page, vous devriez pouvoir créer une entrée dans votre table facilement.

- Créer une nouvelle entrée avec les valeurs suivantes : login = 'root' ; password = 'password'. (Attention, il ne faut pas donner de valeur à id qui s'incrémente automatiquement. SQL va donc lui donner une valeur par défaut de départ (1) et

l'incrémenter à chaque nouvelle entrée). Une fois les informations entrées, cliquer sur exécuter.

- Recommencer cette manipulation afin d'ajouter deux nouveaux utilisateurs : 'user1' ; 'password1' et 'user2' ; 'password2'.
- Rajouter un champ « superUser » dans votre table qui ne peut prendre que trois valeurs différentes (0 ou 1 ; aide : [https://www.w3schools.com/sql/sql\\_datatypes.asp](https://www.w3schools.com/sql/sql_datatypes.asp)). Ce champ pourrait être utilisé de la manière suivante :
  - S'il est à 1 : l'utilisateur a des droits plus importants que simple utilisateur
  - S'il est à 0 : l'utilisateur reste simple utilisateur

Faire une simulation en autorisant les droits à un ou plusieurs utilisateurs

⇒ Appeler le formateur pour vérifier

- Ajouter à votre base de données une seconde table nommée 'IMC'. Puis configurer cette table en y ajoutant les champs : 'nom' ; 'prenom' ; 'taille' ; 'poids' ; 'genre' ; 'IMC'. En réfléchissant au type de données (: [https://www.w3schools.com/sql/sql\\_datatypes.asp](https://www.w3schools.com/sql/sql_datatypes.asp)).
- De même que précédemment, faire quelques entrées pour tester la table (simuler les données de deux ou trois personnes dans la base).

⇒ Appeler le formateur pour vérifier