



 WEB
APPLICATIONS

 ECOMMERCE
SITES

 ENTERPRISE
SYSTEMS

Serveur de Base de données MySQL

Mysql -> SQL (Structured Query Language)

Une base de données permet de stocker des informations.

On peut en ajouter, les modifier ou les supprimer.

Elles sont composées de tables contenant elles même des champs:

Structure

NomTable1
Id
Champs1
Champs2

Mode données

Id	Champs1	Champs2
1	Toto	Tata
2	Titi	Tutu
3	Tete	Tyty

Les tables et les champs

Une table porte un nom afin de pouvoir la pointer pour l'utiliser, on évite les espaces et les caractères spéciaux

La table est composée de champs qui correspondent aux entêtes d'un tableau.

Le premier champs est de manière conventionnelle appelé id pour identifiant.

Il a pour but d'identifier la donnée par un nombre unique. Pour gérer le côté unique, on applique un attribut d'auto-incrément qui donnera un nombre unique (1,2,3,4...).

Si l'on supprime la donnée du 2, elle ne sera plus jamais utilisé.

Les champs sont typés, c'est-à-dire, qu'il faut préciser le type de données que l'on va ajouter.

PhpMyAdmin

Pour éviter de manipuler les bases de données en mode texte, il existe des interfaces graphiques permettant son administration.

Le plus connu pour les serveurs MySQL est PhpMyAdmin, fourni par WAMP, MAMP. Une installation manuelle doit être faite sous LAMP.

PhpMyAdmin est comme un framework php donc il faut passer par le navigateur pour l'utiliser.



The image shows the phpMyAdmin login interface. At the top center is the phpMyAdmin logo, which includes a stylized sailboat icon. Below the logo, the text "Bienvenue dans phpMyAdmin" is displayed. The interface is divided into two main sections. The first section, titled "Langue - Language", contains a dropdown menu with "Français - French" selected. The second section, titled "Connexion", contains two input fields: "Utilisateur :" and "Mot de passe :". At the bottom right of the "Connexion" section is a button labeled "Exécuter".

Après s'être connecté car il s'agit d'un serveur, on arrive sur cette interface

The screenshot displays the phpMyAdmin web interface for a local MySQL server. The top navigation bar includes tabs for 'Bases de données', 'SQL', 'État', 'Utilisateurs', 'Exporter', 'Importer', 'Paramètres', 'Réplication', and 'Variables'. The main content area is divided into three panels:

- Paramètres généraux:** Contains a link to 'Modifier le mot de passe' and a dropdown menu for 'Interclassement pour la connexion au serveur' set to 'utf8_general_ci'.
- Paramètres d'affichage:** Contains a dropdown for 'Langue - Language' set to 'Français - French', a dropdown for 'Thème' set to 'pmahomme', and a dropdown for 'Taille du texte' set to '82%'. A link for 'Plus de paramètres' is also present.
- Server Information Panels (Right Side):**
 - Serveur de base de données:** Lists details such as 'Serveur: Localhost via UNIX socket', 'Type de serveur: MySQL', 'Version du serveur: 5.5.46-0ubuntu0.14.04.2 - (Ubuntu)', 'Version du protocole: 10', 'Utilisateur: root@localhost', and 'Jeu de caractères du serveur: UTF-8 Unicode (utf8)'.
 - Serveur web:** Lists 'Apache/2.4.7 (Ubuntu)', 'Version du client de base de données: libmysql - 5.5.46', and 'Extension PHP: mysqli'.
 - phpMyAdmin:** Lists 'Version: 4.0.10deb1', 'Documentation', 'Wiki', 'Site officiel', 'Contribuer', 'Obtenir de l'aide', and 'Liste des changements'.

The left sidebar shows a tree view of databases and schemas, including 'basesam', 'dronopale', 'information_schema', 'intersysteme', 'intranet', 'mysql', 'opaletrip', 'opway', 'performance_schema', 'phpmyadmin', and 'wphm'.

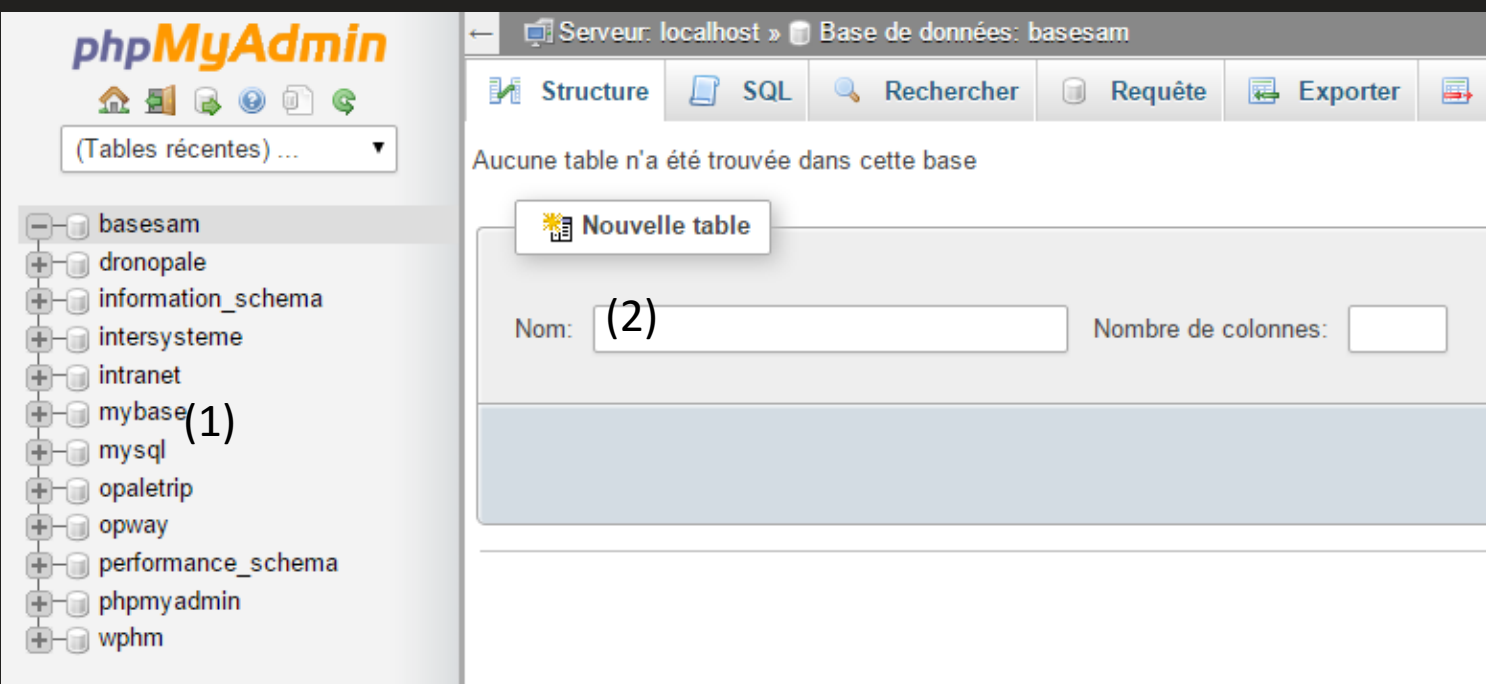
Créons notre première base:



The screenshot shows a web browser window with the address bar displaying "Serveur: localhost". The page title is "Bases de données". The navigation menu includes "Bases de données", "SQL", "État", "Utilisateurs", and "Exporter". The main content area is titled "Bases de données" and features a section "Créer une base de données" with a help icon. Below this, there is a text input field containing "mybase", a dropdown menu currently showing "utf8_general_ci", and a "Créer" button.

Nous allons l'appeler myBase et choisir l'interclassement utf8_general_ci qui définit le format des données.

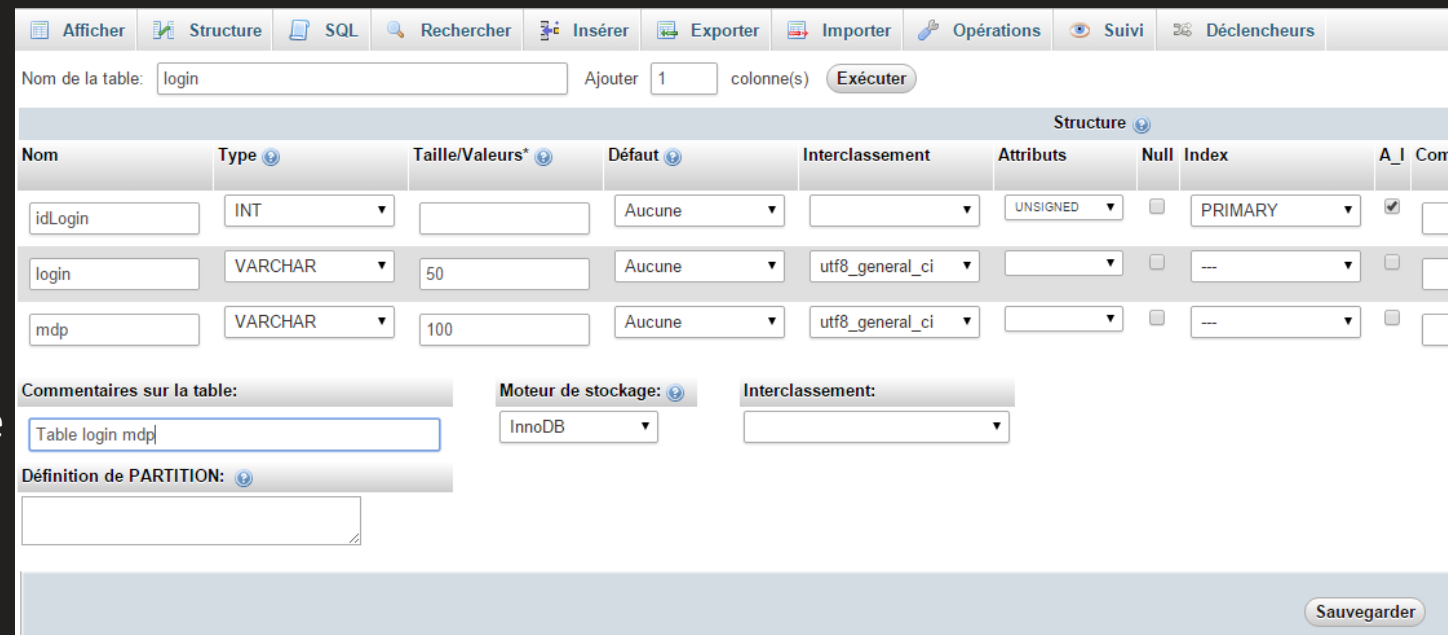
A vous!



- (1) Cliquez sur le nom de la base créée
- (2) Saisir un nom de table (login) et 3 colonnes

Le premier champs idLogin est de format INT(entier), UNSIGNED (non signé cad sans valeur négative), PRIMARY (Primaire) c'est-à-dire champs de référence pour les liens entre tables et en A_I (Auto increment) pour la gestion du côté unique.

Le deuxième login type VARCHAR(taille variable de caractères définit aux max par Taille/Valeurs) de taille 50 max. Et le dernier est mdp qui permettra de stocker le mot de passe (on l'encodera pour la sécurité)



Sauvegarder

Insérons des données (onglet Insérer) via PhpMyadmin:

The screenshot shows the 'Insérer' (Insert) tab in phpMyAdmin. At the top, there is a navigation bar with tabs: Afficher, Structure, SQL, Rechercher, Insérer, Exporter, Importer, Opérations, Suivi, and Déclencheurs. Below this, there are two identical data entry forms. Each form has a table with columns: Colonne, Type, Fonction, Null, and Valeur. The columns are: idLogin (int(10) unsigned), login (varchar(50)), and mdp (varchar(100)). Each column has a dropdown menu for 'Fonction' and a text input field for 'Valeur'. Below each form is an 'Exécuter' button. At the bottom of the page, there is a section with a dropdown menu set to 'Sauvegarder une nouvelle ligne', followed by 'et ensuite' and another dropdown menu set to 'Retourner à la page précédente'. Below this are 'Exécuter' and 'Réinitialiser' buttons. At the very bottom, it says 'Continuer l'insertion avec 2 lignes'.

Ajoutons les 2 comptes suivants:

- ✓ Login: simplon mdp:codeurKiFFeur
- ✓ Login:votrepseudo mdp:votremotdepasse

On laisse vide les 2 idLogin vident car ils seront gérés automatiquement avec l'A_I

Affichons les données saisies (onglet Afficher) via PhpMyadmin:

Seigneur: localhost » Base de données: mybase » Table: login"Table login mdp"

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations Suivi Déclencheurs

✓ Affichage des lignes 0 - 1 (total de 2, Traitement en 0.0005 sec)

```
SELECT *  
FROM `login`  
LIMIT 0 , 30
```

Profilage [En ligne] [Modifier] [Expliquer SQL] [Créer source PHP]

Afficher : Ligne de départ: Nombre de lignes: En-têtes à chaque ligne

Trier sur l'index:

+ Options

		idLogin	login	mdp
<input type="checkbox"/>	Modifier	Copier	Effacer	1 simplon codeurKiFFeur
<input type="checkbox"/>	Modifier	Copier	Effacer	2 manu codeurKiFFEur

Tout cocher Pour la sélection : Modifier Effacer Exporter

Afficher : Ligne de départ: Nombre de lignes: En-têtes à chaque ligne

Opérations sur les résultats de la requête

Version imprimable Version imprimable (avec textes complets) Exporter Afficher le graphique Créer une vue

Comment accéder à ses données dans nos pages pour les administrer:

Nous devons passer par PHP pour accéder à la base et à ses données:

- 🗄 Connexion au serveur et Sélection de la base
- 🗄 Requetes pour sélectionner, insérer, modifier ou supprimer des données
- 🗄 Exploitation des résultats principalement lors d'une sélection.

🗄 Connexion au serveur

mysqli_connect(addressServer, login, password, databasename);

Mysql est un serveur intégré dans LAMP, MAMP, Wamp, mais comme tout serveur nous aurons de nous y connecter via une adresse, un login et un mot de passe. On précisera la base sur laquelle nous allons intervenir.

Bon usage: souvent dans un fichier externe pour les 4 premières infos:

```
DEFINE(SERVER, "localhost");  
DEFINE(LOGIN, "simplon");  
DEFINE(MDP, "codeurKiFFeur");  
DEFINE(BASE, "myBase");
```

```
$connect=mysqli_connect(SERVER, LOGIN, MDP, BASE) or die("pb de connexion au serveur");
```

Requetes

Pour traiter les données, nous devons envoyer des requêtes au serveur. Pour cela nous utiliserons le SQL. Structured Query Language avec quelques spécificités pour MySQL par rapport au SQL Server de Microsoft.

La syntaxe Mysqli est la suivante: `mysqli_query($connect,"la requête");`

La première requête est la selection de données : SELECT

```
SELECT nomdeschamps FROM nomdelatable  
SELECT * FROM nomdelatable
```

On peut préciser les noms de champs séparés par une virgule ou utiliser le métacaractère * pour les sélectionner tous. Comme on peut sélectionner plusieurs tables.

```
SELECT champs1, champs2, champs3 FROM table1,table2
```

Afin de récupérer des données correspondantes à des critères, nous avons la clause where qui nous permet de filtrer les résultats:

```
SELECT * FROM nomdelatable where nomchamp=condition
```

Nous pouvons mixer plusieurs conditions en utilisant le 'et' (AND &&) et le 'ou' (OR ||).
On utilise des parenthèses pour utiliser des conditions complexes de filtrage.

Il existe aussi un élément qui permet de filtrer les données sélectionnées. Il s'agit de l'élément DISTINCT.
Son rôle est de garder uniquement les résultats sans les doublons.

Par exemple si on veut lister les villes dans un carnet d'adresses sans voir plusieurs fois la même ville.

```
SELECT DISCTINCT town FROM adressBook
```

EXO: sélectionner tous les login et mdp de votre table.

Résultat: Rien à l'écran.
Ben alors!!!!

Avant d'aller plus loin de le select, il serait bien de commencer à visualiser le résultat de notre requête. Comme nous travaillons avec des tables, nous allons récupérer le résultat dans un tableau. Nous allons décidé aussi sous quel format dans ce tableau nous allons récupérer les données. Nous allons les récupérer sous forme de tableau associatif (tableau[clé] => valeur). Un exemple est nécessaire:

```
$result=mysqli_query($connect, 'SELECT * FROM login');  
While($data=mysqli_fetch_assoc($result)){  
    echo "login :".$data['login']."<br/>";  
    echo "mdp: ".$data['mdp']."<br/><br/>";  
}
```

A vous de faire.

Pour des soucis d'écriture en français nous devons utiliser souvent :utf8_encode(\$data['mdp']) pour éviter les problèmes des caractères accentués.

Nous irons plus loin sur les requêtes de sélection lors du prochain cours.

Après savoir sélectionner, il est bien de savoir insérer de nouvelles données, nous n'allons pas passer par phpmyadmin à chaque fois. Pour cela nous avons la requête INSERT INTO

```
mysqli_query($con,"INSERT INTO Persons (FirstName,LastName,Age) VALUES ('Glenn','Quagmire',33)");
```

Lors d'une insertion dans une table contenant beaucoup de champs on peut faire comme cela:

```
mysqli_query($con,"INSERT INTO Persons VALUES ('','Glenn','Quagmire',33,'joij....')");
```

Les deux premières quotes dans le values permet de laisser le serveur gérer l'id avec l'auto_increment. Là aussi, nous irons un peu plus loin dans un future proche.

A vous de faire.

Nous avons aussi besoin de modifier des données sur un ou plusieurs champs et pour certaines fiches.
Pour cela, nous avons la requête

```
mysqli_query($con,"UPDATE table_name SET column1=value1,column2=value2,...  
WHERE some_column=some_value");
```

Si on veut juste modifier une fiche il faut penser à ajouter à la requête `limit 1` pour que la modification se fasse que sur la première fiche rencontrée et ne se poursuit pas.

Exo créez une requête qui modifie le login du compte que l'on vient d'ajouter

La dernière pour aujourd'hui, la suppression qui forcément est à manipuler avec précaution:

```
mysqli_query($con,"DELETE FROM table [WHERE conditions] [LIMIT]");
```

Si on veut supprimer une fiche il faut penser à ajouter à la requête limit 1 pour que la suppression se fasse que sur la première fiche rencontrée et ne se poursuit pas.

On utilise quasi tout le temps l'id dans le where pour identifier la fiche à supprimer.

Exo créez une requête qui supprimer le dernier login créé.