

TP n° 7

Introduction à MySQL, interface PHP/MySQL

Documentation Vous trouverez un manuel de référence à MySQL à l'url <https://dev.mysql.com/doc/refman/5.7/en/> Et bien sûr, il vous est conseillé de regarder les transparents du cours.

1 Introduction à MySQL

1 Préliminaires

2 Installer MySQL sur son ordinateur

Afin de ne pas perdre trop de temps avec l'installation, on va donner une méthode "user-friendly" pour installer MySQL¹. Téléchargez le paquet MySQL en cliquant sur ce lien (choisissez la version qui correspond à votre machine). Suivez les instructions d'installation. À la fin, on vous demande de choisir un mot de passe pour l'utilisateur root. Ce seront vos login et mot de passe pour MySQL. Le nom du serveur sera 127.0.0.1 (c'est le serveur local de votre machine).

3 Premières requêtes SQL

3.1 Connexion au SGBD

Placez vous dans votre dossier de travail, puis connectez vous à MySQL avec la commande suivante

```
mysql -u root -p
```

et rentrez votre mot de passe.

3.2 Création et interrogation de base

Rappel : Toutes les commandes en mysql se terminent par un point-virgule (;).

MySQL n'est pas sensible à la casse (minuscules/majuscules) en ce qui concerne les mots-clefs, mais pour des raisons de visibilité ceux-ci seront toujours écrits en majuscules. Attention : MySQL est sensible à la casse pour les noms de tables et d'attributs.

Créez une base de données avec la commande `CREATE DATABASE`, et déclarez que c'est cette base de données que vous utiliserez en utilisant la commande `USE`.

Exercice 1 — Contenu d'une base — Récupérez sur Moodle le fichier `create_tables.sql`. Vous pouvez regarder ce qu'il y a dedans : tout d'abord on supprime les tables si elles existaient déjà (`DROP TABLE...`) ; cela permettra de régénérer les tables. Ensuite, on crée les tables (`create TABLE...`). Puis on les remplit (`INSERT...`). Le fichier est écrit de manière plus élaboré que ce que vous verrez en cours cette année, ne vous inquiétez pas si vous ne comprenez pas tout.

— Connectez-vous à la base, si ce n'est pas encore fait, puis installez le fichier via :

```
source create_tables.sql
```

— Utilisez les commandes `DESCRIBE` et `SHOW` pour répondre aux questions suivantes :

Quelles sont les tables existantes ?

Nous ne nous intéresserons dans un premiers temps qu'aux tables `regions`, `departements` et `cities`. (Vous remarquerez que les données datent un peu : ce sont les anciennes régions et la Corse a encore deux départements). Quels sont les attributs de ces tables ?

1. Pour les puristes, en ligne de commande, sur une distribution Linux avec Aptitude, MySQL s'installe en tapant `sudo apt install mysql-server`. Vous pouvez également utiliser LAMP/WAMP/MAMP.

3.3 Interrogations simples de tables : SELECT

- Exercice 2**
1. Affichez le contenu de la table régions (toutes les colonnes et toutes les lignes).
 2. Affichez la liste des villes avec pour chacune, son nom et le code de son département (mais pas le id).
 3. Affichez la liste des villes (nom uniquement) de la Seine-Saint-Denis (code de département 93). Attention le code du département est une chaîne de caractères, n'oubliez pas les quotes.

3.4 Modifications des données : UPDATE, INSERT, DELETE

Exercice 3 Pour chacune de ces questions vérifiez bien que la modification a eu lieu. Par exemple, pour la première, comme le tableau des villes est grand le plus simple est de choisir une ville qui est en fin d'alphabet. Sinon il vous faudra repérer le code du département de la ville avant et vérifier en affichant juste les villes du département.

1. Supprimez la ville de votre choix.
2. Supprimez toutes les villes du département de votre choix.
3. Renommez la région Bretagne en "Breizh".
4. Ajoutez un nouveau département d'id 96, donnez-lui le nom que vous voulez et affectez-le à la région de votre choix.
5. Mettez dans ce nouveau département toutes les villes du Vaucluse (ou d'un autre département si vous avez déjà supprimé ces villes).
6. Ajoutez une nouvelle ville dans le département 96, donnez-lui le nom que vous voulez. Dans la table cities le premier attribut (id) est automatiquement calculé donc vous ne devez renseigner que les deux autres.

4 Interrogation de bases de données en PHP

On passe maintenant au PHP. Quittez MySQL en utilisant QUIT ;

La figure 1 donne la structure générale d'un fichier PHP qui interroge une base de données.

Exercice 4 Complétez le fichier villes.php de la figure 1, que vous pourrez télécharger sur moodle, pour qu'il affiche toutes les villes (nom et id) du département de votre choix. (Le département est codé en dur pour l'instant.)²

Attention : n'utilisez que des commandes qui commencent par mysqli_ et jamais mysql_.

Testez votre code. Il est possible que vous obteniez un message qui commence par

```
The server requested authentication method unknown to the client
```

Dans ce cas, reconnectez vous à MySQL, puis rentrez la commande suivante :

```
alter user 'root'@'localhost' identified with mysql_native_password by 'my_password';
```

où my_password est votre mot de passe.

5 Interrogation de bases de données en PHP avec un formulaire

Dans cette partie, nous voulons que l'utilisateur puisse choisir son département de manière interactive. Nous allons utiliser les formulaires et en profiter pour factoriser un peu le code. Le résultat final attendu est reproduit en figure 2. Nous allons procéder par étapes.

2. n'oubliez pas de changer le mot de passe et le nom de la base !

```

<!DOCTYPE html>
<head>
  <title> Villes et villages des Vosges </title>
  <meta charset ="utf-8" />
</head>
<body>
  <?php
    /* connexion au serveur: */
    $connexion = mysqli_connect ('127.0.0.1',
      'root', 'my_password', 'ma_base' ) ;

    if (!$connexion) {
      echo 'Pas de connexion au serveur ' ; exit;
    }
    echo 'connexion réussie! <br/>';

    /* pour que les caractères reçus soient codés en utf-8. */
    mysqli_set_charset($connexion, 'utf8');

    /* requête: */
    $req = 'SELECT ....' ; /* A COMPLÉTER */

    $resultat = mysqli_query ($connexion, $req );
    if (!$resultat) {
      echo "requête incorrecte";
      echo mysqli_error($connexion);
    }
    else {
      while ($ligne=mysqli_fetch_assoc($resultat))
        //écrire ligne A COMPLÉTER
    }
    mysqli_close($connexion);
  ?>
</body >
</html >

```

FIGURE 1 – Structure d’un fichier PHP interrogeant une base de données.

Exercice 5 Faites un fichier php qui contient une fonction qui renvoie une connexion à la base et termine sinon (fonction exit). Faire ensuite un fichier php qui contient une fonction qui prend en argument un id de département et une connexion et affiche les villes de ce département. Faites enfin un fichier principal villes2.php qui fait la même chose que dans l’exercice précédent, mais en incluant ces fichiers avec require_once et en utilisant les fonctions de ces fichiers. Testez.

Exercice 6 Faites un nouveau fichier contenant une fonction qui va afficher le code HTML correspondant à la portion du formulaire qui va permettre de choisir le département. (Il s’agit d’une liste déroulante qui commence par <select . . , cf. cours html.) Dans villes2.php, créez un formulaire, appelez cette fonction au bon endroit. Le formulaire enverra ses données par POST à lui-même. Maintenant il vous reste dans villes2.php à vérifier si vous avez bien un paramètre POST pour le département et dans ce cas afficher les villes correspondantes. (Pour l’instant, on ne met pas de sous-titre du genre “Villes du département de Corrèze”).

Exercice 7 Maintenant, vous pouvez faire en sorte que la fonction qui affiche les villes récupère le nom du département par la requête adéquate et affiche son nom. (On ne s’embarrassera pas

villes du département de Corrèze

nom	id
Altiliac	31
Brive-la-Gaillarde	324
Espagnac	746
Lacelle	1030
Laguenne	1035
Marc-la-Tour	1190
Saint-Pantaléon-de-Larche	1967
Sioniac	2099
Tulle	2205
Varetz	2244

Requête:

Aude

▼

choisir

Autre requête:

Ain

▼

choisir

FIGURE 2 – À gauche avant que le formulaire ait été rempli, à droite, après.

du fait qu’il écrira “Villes du département de Jura” au lieu de “du Jura”). On fera aussi varier le sous-titre suivant qu’une requête a déjà été traitée ou non.

2 Enregistrer des utilisateurs à un forum

On va maintenant voir comment gérer des utilisateurs. Nous allons nous intéresser au premier enregistrement d'un nouvel utilisateur et à la page qui lui permettra de s'authentifier sur le site. Pour cela, nous allons utiliser la table `users` qui peut être créée avec le fichier `create_table_users.sql` fourni.

Nous vous conseillons, comme précédemment, de faire plusieurs fichiers avec des fonctions.

Exercice 8 Commencez par regarder à quoi ressemble cette table.

Exercice 9 Faites une page `nouveau.php` qui

- soit présentera un formulaire pour s'inscrire sur le site en demandant le nom et le prénom de l'utilisateur, le pseudo et le mot de passe qu'il veut utiliser. (Le pseudo servira de login ensuite.) Ce formulaire enverra les données par POST à lui-même.
- soit, si l'utilisateur a déjà fait sa demande testera si tout est correct (paramètres POST tous remplis) et insèrera le nouvel utilisateur dans la base. L'insertion se fait avec `mysqli_query` et retourne `false` en cas d'erreur. Si l'inscription est réussie, on affichera "inscription réussie", sinon, on affichera un message d'erreur et on réaffichera le formulaire.

Important : Le mot de passe devra être crypté avant d'être enregistré. Sur les machines du script les méthodes `password_hash` et `password_verify` ne fonctionnant pas, on utilisera à la place la fonction `md5` pour crypter le mot de passe et `===` pour tester l'égalité entre les deux chaînes cryptées. Par ailleurs, on utilisera `mysqli_real_escape_string` partout où c'est nécessaire.

Vérifiez en regardant directement dans la base que le mot de passe est bien crypté.

Exercice 10 Faites une page `accueil.php` qui

- soit présentera un formulaire pour se logger sur le site en demandant le pseudo et le mot de passe. Ce formulaire enverra les données par POST à lui-même. On affichera aussi un lien en bas du formulaire vers `nouveau.php` pour qu'un nouvel utilisateur puisse s'inscrire.
- soit, si l'utilisateur a déjà fait son login testera si tout est correct (champs tous remplis) et vérifiera que les données correspondantes sont dans la base.
Si le login est réussi, on affichera "login réussi", sinon, on affichera un message d'erreur et on réaffichera le formulaire.

Exercice 11 — Pour aller plus loin Au lieu d'afficher des messages en cas de réussite, on affichera un lien vers une autre page où grâce aux session on pourra afficher "Bienvenue BIDULE."