

## TP n° 9

### Introduction à MySQL, interface PHP/MySQL

#### 1 Persistance par les Sessions (reprise du TP 8)

Le but de cet exercice est de voir la persistance des données par les sessions, si vous n'avez pas réussi à faire le TP précédent. Si vous avez fini l'exercice du TP 8, vous pouvez sauter cette partie du TP et commencer par les exercices sur SQL dont vous aurez besoin pour votre projet.

Les cookies nous ont donné une première façon de stocker les données entre deux visites d'un même utilisateur, mais ils ont de nombreux désavantages : ils ont une durée de vie limitée, on doit faire confiance au navigateur, et ils forcent à échanger une grande quantité de données à chaque requête (imaginez si vous aviez stocké une photo dans le profil de l'utilisateur !).

En plus, les cookies ne garantissent aucune sécurité : si vous voulez stocker une variable qui indique que l'utilisateur s'est connecté avec son identifiant et mot de passe, cette variable ne peut pas être stockée par le client car il pourrait la falsifier.

PHP offre une technique avancée, appelée "Sessions PHP" qui offre la persistance des données en faisant abstraction des cookies et des techniques de pseudo-persistance.

##### Exercice 1 — *Persistance par les sessions*

1. Créez une page `countSession.php` qui compte le nombre de visites grâce aux sessions.
2. Allez voir les cookies sauvegardés par votre navigateur, vous remarquerez que `pams.script` a enregistré un nouveau cookie nommé `PHPSESSID`. Effacez ce cookie et visitez à nouveau la page, que remarquez-vous ?

Ce n'est pas sorcier : les sessions PHP se servent des cookies pour permettre la persistance des données. Cependant le cookie `PHPSESSID` ne contient aucune des données que vous avez stockées dans le tableau `$_SESSION` : ce cookie ne contient rien d'autre que l'id de session. L'id de session est une chaîne de caractères aléatoire et unique créée par le serveur pour identifier le client ; lors du premier appel de `session_start()` les actions suivantes sont exécutées par PHP :

1. un nouvel id est généré ;
2. un fichier temporaire correspondant à l'id est créé **sur le serveur** ;
3. un cookie `PHPSESSID` contenant l'id de session est envoyé **au client**.

Pour s'entraîner plus, vous pouvez aussi reprendre l'Exercice 4 du TP 7 et adaptez les fichiers de l'exercice de façon à utiliser des sessions à la place des cookies.

## 2 Introduction à 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 Premières requêtes SQL

#### 1.1 Connexion au SGBD

Pour interroger la base de données en ligne de commande, nous allons utiliser l'interprète `mysql`. **Si vous utilisez les machines du script**, vous devez dans un premier temps vous être enregistré comme utilisateur et avoir créé votre propre base de données sur le serveur MySQL du script en allant sur <http://pams.script.univ-paris-diderot.fr/database/formulaire.html>. Notez bien le mot de passe fourni à cette étape : c'est votre mot de passe MySQL. Commencez alors par vous logger, en utilisant la commande

```
mysql -D votreLogin -h pams.script.univ-paris-diderot.fr -u votreLogin -p
```

puis rentrez votre mot de passe.

**Si vous utilisez votre ordinateur personnel**, nous allons mettre à profit le TP3, dans lequel vous avez installé XAMPP. Dans un premier temps, assurez-vous d'avoir bien lancé MySQL (des instructions par système d'exploitation sont disponibles dans le TP3).

Puis il faut vous connecter à MySQL :

Sous Linux, via la commande

```
/opt/lampp/bin/mysql -u root -p
```

Sous Mac, via la commande

```
/Applications/xampp/xamppfiles/bin/mysql -u root -p
```

Sous Windows, via la commande

```
C:\xampp\mysql\bin\mysql -u root -p
```

suivi dans les trois cas de votre mot de passe.

#### 1.2 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 ?

### 1.3 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 guillemets.

### 1.4 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.

## 2 Interrogation de bases de données en PHP

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

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

- Exercice 4**
- Complétez le fichier villes.php de la Figure ??, 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.)<sup>1</sup>

**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.

## 3 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 ??.

---

1. 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 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.

nom	id
Atiliac	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

Aude  choisir

Ain  choisir

5