

Bases de données

TP n° 1 – Introduction aux bases de données pour le Web

12 février 2014

Le but de ce TP est d'introduire différents outils et langages qui peuvent être employés pour gérer des bases de données sur le Web, et qui seront largement employés dans les TP suivants, à savoir : MySQL 5 et ses interfaces d'administration (ligne de commande, phpMyAdmin), les bases du langage SQL, HTML, PHP. Les différents exercices ci-dessous sont conçus pour être traités dans l'ordre.

1 L'outil de ligne de commande mysql

Nous allons réaliser au cours de ce premier TP une mini-application Web permettant de gérer une petite base de données de films. Dans un premier temps, nous allons supposer qu'à chaque film est associé un titre, un nom de réalisateur, et un pays.

1. Imaginer un schéma de base de données relationnelle adapté à cette application.
2. Nous allons créer la ou les tables correspondantes dans le serveur de bases de données MySQL 5. Pour cela, nous allons utiliser l'outil de ligne de commande `mysql`. Taper à l'invite du shell :

```
mysql -h tpbdd -u login -p
```

où `tpbdd` est le nom de la machine hébergeant le serveur MySQL et `login` votre identifiant de connexion à MySQL. Votre mot de passe MySQL vous sera alors demandé.

3. Vous pouvez taper à l'invite de commande MySQL des *ordres SQL* ou d'autres commandes spécifiques à cet outil. Les ordres et commandes se terminent en général par un caractère `< ; >` (pour un affichage en colonne) ou `< \G >` (pour un affichage en ligne). La commande `HELP` vous permettra d'obtenir de l'aide sur toute autre commande (on pourra aussi se référer à la documentation en-ligne de MySQL, voir <http://dev.mysql.com/doc/refman/5.5/en/>). Les mots-clés ne sont pas sensibles à la case, mais sont habituellement écrits en capitales.

Commençons par examiner la liste des *bases de données* auxquelles vous avez accès avec `SHOW DATABASES`. La base de données `information_schema` est une base virtuelle contenant des méta-informations sur les autres bases. Vous pouvez créer une base de données dont le nom est identique à votre login à l'aide de la commande `CREATE DATABASE nom.base`. Nous allons nous placer dans cette base de données avec `USE nom.base`.

4. Créer maintenant la ou les tables appropriées. Les commandes utiles sont les suivantes :

<code>SHOW TABLES</code>	affiche le nom des tables de la base courante
<code>DESCRIBE Nom_table</code>	affiche le schéma de la table <i>Nom_table</i>
<code>CREATE TABLE Nom_table (a INT, b VARCHAR(10))</code>	crée une table <i>Nom_table</i> à deux colonnes
<code>DROP TABLE Nom_table</code>	détruit une table (et son contenu)
<code>SELECT * FROM Nom_table</code>	affiche le contenu d'une table

Dans l'ordre SQL `CREATE TABLE`, le nom et type d'un attribut peuvent être suivis de certaines options :

- `NOT NULL` pour spécifier que l'attribut ne peut prendre la valeur spéciale `NULL`, dénotant une absence de valeur.
- `UNIQUE` pour spécifier que l'attribut est une *clef unique* de la table, c'est-à-dire que deux lignes de la table ne peuvent avoir la même valeur pour cet attribut.
- `PRIMARY KEY` pour spécifier que l'attribut est une *clef primaire* de la table, c'est-à-dire à la fois `UNIQUE` et `NOT NULL`.
- `AUTO_INCREMENT` pour spécifier que la valeur de cet attribut sera automatiquement généré comme identifiant unique auto incrémenté ; cette option ne peut s'appliquer qu'à une clef primaire.

Par convention, les noms de tables MySQL commencent par une majuscule. L'outil `mysql` dispose d'une complétion automatique avec la touche *Tabulation*.

Les types de données MySQL les plus utiles sont les suivants :

Type	Description	Exemple
INT	entier	42
DOUBLE	flottant en double précision	2.345e7
CHAR(<i>n</i>)	chaîne de caractères d'exactly <i>n</i> caractères	'HELLO'
VARCHAR(<i>n</i>)	chaîne de caractères d'au plus <i>n</i> caractères	'HELLO'
TEXT	long texte	'Hello world, how are you?'
BLOB	long contenu binaire	'éÀø.?'
DATE	date au format AAAA-MM-DD	'2008-02-15'
YEAR	année	2008
DATETIME	date et temps au format AAA-MM-DD HH:MM:SS	'2008-02-15 10:30:00'
TIME	heure au format HH:MM:SS	'10:30:00'

5. Insérer à la main un film dans la base de données, à l'aide de l'ordre SQL `INSERT INTO` qui peut s'employer en précisant ou non le nom des attributs :

```
INSERT INTO Toto (a,b) VALUES (42,'Hello')
INSERT INTO Toto VALUES (42,'Hello')
```

2 phpMyAdmin

Une autre interface d'administration est proposée par l'application Web phpMyAdmin, disponible à l'URL <http://tpbdd.dptinfo.ens-cachan.fr/monitoring/phpmyadmin/> (identifiant et mot de passe MySQL sont demandés à la connexion). Se familiariser avec cette interface, et l'utiliser pour rajouter quelques autres films.

3 Premier formulaire HTML

Vous placerez les documents produits dans le sous-répertoire `public_html` de votre répertoire Unix principal. Si ce répertoire n'existe pas, vous pouvez le créer avec la commande `mkdir public_html` depuis votre répertoire principal. Si votre login est `toto`, vous pourrez accéder à un document `titi.html` via un serveur Web à l'URL : <http://tpbdd.dptinfo.ens-cachan.fr/~toto/titi.html>. Pour vérifier qu'il n'y a pas d'erreurs dans l'écriture de fichiers HTML, en plus de regarder le résultat dans un ou plusieurs navigateurs, on pourra utiliser un *validateur* qui permet de vérifier la conformité d'un document. Un validateur pour HTML est disponible à l'URL <http://validator.w3.org/>. Vous pourrez vous référer à l'aide-mémoire HTML fourni, au tutorial HTML de w3schools <http://www.w3schools.com/html/> et à la recommandation du W3C pour HTML 4.01, <http://www.w3.org/TR/html401/> et XHTML 1.0, <http://www.w3.org/TR/xhtml1/>

Créer un document HTML `ajout_nouveau_film.html` contenant un *formulaire* qui servira à ajouter un nouveau film dans la base de données. Titre et nom de réalisateur pourront être saisis comme du texte libre, tandis que le pays pourra être proposé par une liste déroulante à quelques entrées.

4 PHP et MySQL

PHP est un langage interprété, adapté pour le développement de sites Web. Un script PHP est un document incluant du contenu littéral textuel (en général des morceaux de pages HTML) et des blocs d'instructions encadrés par les pseudo-balises `<?php` (ou `<?`) et `?>`. Un tel script est mis à la disposition du serveur Web et est interprété par celui-ci pour fournir à l'utilisateur une page Web (ou un autre document). Ainsi, le programme PHP suivant affiche la table de multiplication de 1 à `$M` sous forme d'un tableau HTML, où `$M` est récupéré comme paramètre de requête HTTP GET :

```
<?php $M=$_GET["M"]?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
  <head><title>Table de multiplication</title></head>
  <body>
    <h1>Table de multiplication</h1>
    <table>
      <caption>Table de multiplication de 1 à <?php echo $M ?></caption>
      <tr>
        <th></th>
```

```

    <?php for($i=1;$i<=$M;$i=$i+1) { echo "<th>$i</th>"; } ?>
</tr>
<?php
    for($i=1;$i<=$M;$i=$i+1) {
        echo "<tr><th>$i</th>";
        for($j=1;$j<=$M;$j=$j+1) {
            echo "<td>".($i*$j)."</td>";
        }
        echo "</tr>";
    }
    ?>
</table>
</body>
</html>

```

La syntaxe de PHP est très proche de celle de C, C++ ou Java. Nous indiquons ici les principales différences :

- N'est interprété que le code entre pseudo-balises PHP. Le contenu à l'extérieur de ces pseudo-balises est produit tel quel dans la page de résultat, comme si c'était une chaîne de caractères littérale en argument de l'opérateur `echo`. Ainsi, les deux lignes suivantes sont équivalentes :

```
Hello <?php echo "world" ?>!
```

```
Hello world!
```

Un raccourci est proposé par la pseudo-balise `<?=>`; celle-ci est équivalente à `<? echo`, donc les deux lignes suivantes sont équivalentes.

```
1 à <?php echo $M ?>
```

```
1 à <?=$M ?>
```

- Les variables en PHP sont toujours préfixées d'un symbole `<< $ >>`. Elles n'ont pas besoin d'être déclarées, et sont non typées.
- Les chaînes de caractères littérales sont délimitées en PHP par des guillemets simples ou doubles. La différence entre ces deux possibilités est qu'une chaîne de caractères délimitée par guillemet double est sujette à *interpolation* : le nom d'une variable est remplacé par sa valeur. Ainsi, `echo "J'ai $age ans";` produit `J'ai 30 ans` si `$age` contient la valeur 30.
- La concaténation de chaînes de caractères s'exprime par l'opérateur `<< . >>`.
- En plus des valeurs scalaires (entiers, flottants, chaînes de caractères...), PHP supporte nativement deux types de valeur de tableaux : les tableaux indicés et les tableaux associatifs. Le parcours d'un tableau peut se faire avec une boucle `for` standard, ou avec l'instruction `foreach`. Pour un tableau indicé, les deux blocs suivants sont équivalents :

```
// Boucle foreach
```

```
foreach($tableau as $case) {
    echo $case . "\n";
}
```

```
// Boucle for
```

```
for($indice = 0; $indice < count($tableau); ++$indice) {
    echo $tableau[$indice] . "\n";
}
```

De même pour un tableau associatif :

```
// Boucle foreach
```

```
foreach($tableau as $clef => $valeur) {
    echo "$clef=$valeur\n";
}
```

```
// Boucle for
```

```
$clefs = array_keys($tableau);
for($indice = 0; $indice < count($clefs); ++$indice) {
    echo $clefs[$indice] . "=" . $tableau[$clefs[$indice]] . "\n";}

```

Vous pouvez vous référer à la documentation en ligne de PHP, sur <http://www.php.net/manual/>, en particulier pour connaître l'ensemble des fonctions de la (riche) bibliothèque standard.

`$_GET` et `$_POST` sont deux tableaux associatifs proposant respectivement les paramètres GET et POST de la requête HTTP; ils servent donc à récupérer les paramètres des formulaires.

L'accès à une base de données MySQL depuis PHP utilise les fonctions suivantes :

<code>mysql_connect(\$serveur,\$login,\$mdp)</code>	établit une connexion au serveur
<code>mysql_select_db(\$nom_base)</code>	sélectionne une base de données
<code>mysql_query(\$ordre_sql)</code>	exécute un ordre SQL
<code>mysql_fetch_array(\$resultat)</code>	retourne une ligne de résultat, sous forme de tableau indicé
<code>mysql_fetch_assoc(\$resultat)</code>	retourne une ligne de résultat, sous forme de tableau associatif
<code>mysql_error()</code>	renvoie le dernier message d'erreur

Une utilisation typique est ainsi :

```
if(!mysql_connect("localhost","dupont","a1b2c3d4")) {
    echo "<p>Desolé, connexion impossible</p>"; exit;
}
if(!mysql_select_db('ifips')) {
    echo "<p>Desolé, accès à la base impossible</p>"; exit;
}
$resultat= mysql_query("SELECT * FROM Personne");
if($resultat) {
    while($ligne=mysql_fetch_assoc($resultat)) {
        echo "<p>".$ligne["prenom"]." a ". $ligne["age"]." ans</p>";
    }
} else {
    echo "<p>Erreur dans l'exécution de la requête.</p>";
    echo "<p>Message de MySQL: ".mysql_error()."</p>";
}
```

1. Récupérer depuis l'adresse <http://tpbdd.dptinfo.ens-cachan.fr/~csirange/bd01> le script PHP affichant la table de multiplication et le tester (après l'avoir renommé `.php`).
2. Écrire un script `affichage.php` qui affiche sous la forme d'un tableau HTML le contenu de la table Films, trié par titre (pour faire ce tri, on peut rajouter la clause `ORDER BY colonne` après une requête `SELECT`).
3. Écrire un script `insert.php` comme script de traitement des données du formulaire `ajout_nouveau_film.html`, insérant le film correspondant dans la base de données.
4. Ajouter au tableau de `affichage.php` une colonne avec des boutons Supprimer ou des checkbox (on aura peut-être besoin d'utiliser des champs de formulaire de type `hidden`).
5. Écrire le script `supprimer.php` correspondant (il est possible qu'il soit nécessaire). On utilisera l'ordre SQL `DELETE` dont la syntaxe est la suivante :

```
DELETE FROM Toto WHERE a=3 OR b='foobar'
```