Bases de données (INF225), TELECOM ParisTech

TP PHP & MySQL

Bogdan Cautis (bogdan.cautis@telecom-paristech.fr)
Pierre Senellart (pierre.senellart@telecom-paristech.fr)

6 février 2009

Le but de ce TP est de découvrir l'utilisation d'un SGBD dans le cadre du développement d'une application Web. À la place d'Oracle, nous utiliserons le SGBD MySQL pour ce TP, ce qui nous permettra d'en constater les similarités et différences (le langage SQL est utilisé dans Oracle et dans MySQL, mais il y a parfois de petites différences). On utilisera également le langage de programmation PHP.

1 L'outil de ligne de commande `mysql`

Nous allons réaliser au cours de 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 le ou les tables correspondantes dans le serveur de bases de données MySQL. Pour cela, nous allons utiliser l'outil de ligne de commande `mysql` (c'est l'équivalent de l'outil `sqlplus` pour Oracle). Taper à l'invite du shell :

   ```
   /usr/pkg/bin/mysql -h mysql.infres.enst.fr -P 3307 -u login -p login
   ```

   où `login` votre identifiant de connexion à MySQL (à mettre deux fois sur la ligne ci-dessus, une fois pour le nom d'utilisateur, une fois pour le nom de base). 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/5.0/en/`).

   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. Nous allons nous placer dans l'autre bases de données à votre disposition avec `USE` *`nom_base`*.
4. Créer maintenant la ou les tables appropriées, avec l'ordre SQL standard de création de table. La commande `DESCRIBE` *`Nom_table`* permet d'afficher le schéma d'une table créée. Noter que le type de données Oracle `VARCHAR2(`*`n`*`)` n'existe pas, il est remplacé en MySQL par le type `VARCHAR(`*`n`*`)`
5. Insérer à la main quelques films dans la base de données.

2 Premier formulaire HTML

Vous placerez les documents produits dans le sous-répertoire `public_html` de votre répertoire Unix principal. Si votre login Unix est `pierre`, vous pourrez accéder à un document `index.html` via un serveur Web à l'URL : `http://www.infres.enst.fr/~pierre/index.html`. S'assurer que le serveur Web a bien un droit de lecture sur votre répertoire principale et le sous-répertoire `public_html` à l'aide de la commande :

```
chmod a+rx ~ ~/public_html
```

Il peut être nécessaire de réutiliser cette commande à chaque fois que vous créez un nouveau fichier, en fonction de la configuration.

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.

Si vous êtes débutant en HTML, ou si vous avez besoin de vous rafraîchir la mémoire, référez-vous rapidement au cours lié depuis la page Web du cours : `http://pierre.senellart.com/enseignement/2008-2009/inf225/`

3 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 (c'est-à-dire indiqué sous la forme `?M=30` à la fin de l'URL) :

```
<?php $M=$_GET["M"]?>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <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 son 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 caractère 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 $case."\n";
}
// Boucle for
foreach($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);
foreach($indice=0;$indice<count($clefs);++$indice) {
  echo $clefs[$indice]."=".$tableau[$clefs[$indice]]."\n";
}
```

Vous pouvez vous référer à la documentation en ligne de PHP, sur `http://php.net/`, 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 :

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

Une utilisation typique est ainsi :

```
if(!mysql_connect("mysql.infres.enst.fr:3307","login","password")) {
  echo "<p>Desolé, connexion impossible</p>"; exit;
}
if(!mysql_select_db("database")) {
  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 la page Web du TP (`http://pierre.senellart.com/enseignement/2008-2009/inf225/`) le script PHP affichant la table de multiplication et le tester.
2. Écrire un script `affichage.php` qui affiche sous la forme d'un tableau HTML le contenu de la table Films, trié par titre.
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 formulaires contenant des boutons Supprimer (on aura besoin d'utiliser des champs de formulaire de type `hidden`).
5. Écrire le script `supprimer.php` correspondant.

4 Sécurité et redirections

1. Tenter d'ajouter avec le formulaire réalisé un film dont le titre comprend :
   – une apostrophe (p. ex., *L'Auberge Espagnole*) ;
   – des chevrons (p. ex., *Bienvenue à <Gattaca>*).
   Que se passe-t-il dans chacun de ces deux cas ?
2. L'apostrophe est un caractère spécial pour MySQL, qui délimite les chaînes de caractères (SQL). Comme un ordre SQL est vu par PHP comme une simple chaîne de caractères (PHP), il est crucial d'échapper (c'est-à-dire, précéder d'un backslash) les apostrophes contenus dans des variables PHP destinées à être utilisées à l'intérieur d'une chaîne de caractères MySQL. La fonction `mysql_escape_string` fait cela (la fonction `stripslashes` fait l'opération inverse au cas où celle-ci est nécessaire).
   Ne pas faire attention à cela peut non seulement causer des bugs, mais aussi des problèmes de sécurité. Quel sera, ainsi, le comportement de la requête suivante :

   ```
   mysql_query("SELECT * FROM Users WHERE login='$login' AND password='$password'");
   ```

si `$password` contient « `' OR 1=1 --` » (« `--` » introduit des commentaires en SQL) ? Ce problème de sécurité est connu sous le nom d'injection de code SQL.

Ajouter partout où c'est nécessaire cette protection, tester.

3. Les chevrons, de même que l'esperluette (&), sont des caractères spéciaux en HTML. Ainsi, une chaîne de caractères affichée par un simple `echo` contenant ces caractères va être interprété comme du code HTML, ce qui peut poser des problèmes d'affichage, voire des problèmes de sécurité en cas de code actif (en particulier, scripts JavaScript), connus sous le nom de *XSS* ou *cross-site scripting*. La fonction `htmlspecialchars` permet de remplacer ces caractères par les entités correspondantes (p. ex., `<` pour <). Dans le cas où le texte produit est à l'intérieur d'un *attribut* HTML, il faut aussi protéger les guillemets avec :

   ```
   echo htmlspecialchars($variable,ENT_QUOTES);
   ```

   Ajouter partout où c'est nécessaire cette protection, tester.

4. Bien que beaucoup de scripts PHP aient pour rôle de produire une page HTML qui sera affichée dans un navigateur, certains d'entre eux se contentent de réaliser une action (insertion, suppression) avant de passer la main à un autre script. Ce comportement peut-être obtenu avec la fonction PHP `header` qui modifie les en-têtes HTTP envoyés au client Web ; ainsi,

   ```
   header("Location: suite.php");
   ```

   demande une redirection vers `suite.php`. Attention : une telle redirection (de même que toute manipulation des en-têtes HTTP) n'est possible que si rien n'a encore été écrit sur la page (pas de blanc, pas de déclaration de type de document HTML, etc.).

   Ajouter une telle redirection depuis `insert.php` et `supprimer.php` vers `affichage.php`, dans le cas où les opérations se sont déroulées sans encombre.

5. Que se passe-t-il si l'on demande la suppression d'un film qui porte le même nom qu'un autre film[1] ? On pourra ajouter une colonne supplémentaire déclarée comme `AUTO_INCREMENT` si nécessaire.

5 Jointure

1. On voudrait maintenant ajouter à notre base de données une liste d'acteurs pour chaque film. Revoir le schéma de la base de données pour permettre ceci. Éviter au maximum la redondance d'informations !
2. Ajouter à la main quelques acteurs apparaissant dans les films déjà rentrés.
3. Modifier le script `affichage.php` pour afficher dans une colonne supplémentaire du tableau HTML la liste des acteurs (par exemple, séparés par des virgules).
4. Modifier le formulaire d'ajout et `insert.php` pour demander la liste des acteurs apparaissant dans le film. La liste des acteurs possibles pourra être présentée sous forme de liste à choix multiple (c'est-à-dire, avec `<select multiple="multiple">`). Il sera nécessaire de faire du formulaire d'ajout un script PHP pour récupérer la liste des acteurs.
5. Créer un formulaire d'ajout de nouvel acteur et le script PHP correspondant.
6. Compléter l'application en permettant la modification des données existantes, en embellissant les pages HTML avec du code CSS, etc.

[1] Ce n'est pas une question sans fondement puisque par exemple trois films différent portent le titre *La vie est belle*. On peut même imaginer des cas où deux films différents ont le même titre, le même nom de réalisateur et le même pays.