

# Bases de données, ENS Cachan & Ulm

## TP n° 2 – Site Web en PHP et MySQL

Pierre Senellart (pierre@senellart.com)

22 février 2008

Le but de ce TP est de poursuivre la découverte des langages de gestion de bases de données sur le Web, en particulier SQL et PHP.

### I Préliminaires

Terminer, si ce n'est déjà fait, les 4 premiers exercices du TP n° 1. Les exercices suivants vont en particulier détailler les différentes étapes nécessaires à la réalisation de l'exercice 5 du TP n° 1.

### 2 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 interprétée 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., &lt; 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<sup>1</sup> ? Utiliser une colonne supplémentaire `AUTO_INCREMENT` (cf. TP n° 1) si nécessaire. En pratique, à moins qu'une autre *clef primaire* naturelle existe (numéro de sécurité sociale, numéro d'immatriculation, etc.), on ajoutera dans la plupart des cas un tel *identifiant de n-uplet* aux tables créées, pour ce genre de circonstances.

### 3 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). On aura besoin d'une forme plus générale de l'ordre SQL `SELECT` :

```
SELECT Table1.Colonne1, Table2.Colonne2, Table2.Colonne3
FROM Table1, Table2
WHERE Table1.id=Table2.ref
ORDER BY Colonne1
```

Le préfixe du nom de table peut être omis dans le cas où il n'y a pas d'ambiguïté.

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 (`<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.

---

<sup>1</sup> Ce n'est pas une question sans fondement puisque par exemple trois films différents 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.