Bases de données

TP n° 2 – Site Web en PHP et MySQL

19 février 2014

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.

1 Préliminaires

Terminer, si ce n'est déjà fait, le 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_real_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. Que se passe-t-il si vous tapez directement l'adresse qui correspond à votre fichier `insert.php` ?
   Le fait que `insert.php` est censé opérer sur des valeurs produites par le formulaire `ajout_nouveau_film.html` peut causer des problèmes. On peut distinguer selon que les variables `$_POST` ont été assignées ou non en utilisant la fonction PHP `isset()`. Utilisez cette fonction pour obtenir un script `insert.php` qui n'effectue aucune action tant que les paramètres n'ont pas été envoyés.
5. 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.

6. Que se passe-t-il si l'on demande la suppression d'un film qui porte le même nom qu'un autre film [1] ? 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.

   Il est possible de récupérer la dernière valeur insérée dans une colonne `AUTO_INCREMENT` (pour la connexion courante) en utilisant la fonction PHP `mysql_insert_id()`.

   Si vous avez besoin d'ajouter une colonne à une table, utlisez la commande SQL suivante :

   `ALTER TABLE` *`Table_name`* `ADD COLUMN` *`column_name column_definition`*

3 Jointure

1. On voudrait maintenant ajouter à notre base de données une liste d'acteurs pour chaque film. Pour chaque acteur, nous allons représenter également leur date de naissance.

   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. Dans une base de données avec plusieurs tables il est courant que les valeurs qui apparaissent dans différentes tables soient liées. Le type plus courant de relation entre attributs de différentes tables est la contrainte de "clef étrangère" (*foreign key*). Si `att1` est une clef primaire pour la table `Table1`, et `att2` est un attribut de la table `Table2`, il est possible de définir une contrainte de clef étrangère dans laquelle `att2` référence `att1`. Cela impose, pour qu'une instance de la base de données soit correcte, que chaque valeur de la colonne `att2` de `Table2` apparaisse également dans la colonne `att1` de `Table1`.

   `MySQL` permet de définir des contraintes de clef étrangère sur les tables – et vérifie que celles-là soient respectées – quand le *storage engine* InnoDB est actif (InnoDB est le *storage engine* par défaut sur le serveur MySQL du département). La syntaxe pour créer une contrainte de clef étrangère en phase de création des tables est la suivante :

   ```
   CREATE TABLE Table1 ( att1 INT PRIMARY KEY,...)

   CREATE TABLE Table2 ( att2 INT, ..., FOREIGN KEY (att2) REFERENCES Table1(att1) )
   ```

   Pour ajouter une contrainte de clef étrangère à des table existantes, utiliser la syntaxe :

   ```
   ALTER TABLE Table2 ADD FOREIGN KEY (att2) REFERENCES Table1(att1)
   ```

   Notez que les colonnes correspondantes à la contrainte dans la table et la table de référence doivent avoir le même type.

   Il est possible de définir des clefs étrangère constituées de plusieurs attributs en remplaçant, dans la syntaxe `FOREIGN KEY`, les attributs simples par une liste d'attributs séparés par une virgule.

   Définir les contraintes de clef étrangère appropriées dans la base de données des films. Avec des operations de `DELETE`, `INSERT`, `UPDATE`, essayer de violer les contraintes de clef étrangère définies. Vous aurez besoin de la syntaxe :

   ```
   INSERT INTO Table (colonne1,..., colonnek) VALUES ('valeur1',..., 'valeurk')

   UPDATE Table SET colonne1='valeur1', ..., colonnek='valeurk'
   WHERE colonne0='valeur0'

   DELETE FROM Table WHERE colonne1='valeur1'
   ```
4. 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
   ```

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.

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

5. 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.
6. Si nécessaire, modifier `supprimer.php` pour que la suppression d'un film ne viole pas les contraintes de clef étrangère.
7. Créer un formulaire d'ajout de nouvel acteur et le script PHP correspondant.
8. Rendre possible l'édition des données existantes.
9. Compléter l'application en embellissant les pages HTML avec du code CSS, etc.