

TP 13 – semaine du 26 mars 2007

Internet et Outils (IO2)

PHP et mysql

Le but de ce TP est d'apprendre à interfacier une base de données grâce à du PHP et à travers un formulaire XHTML.

1 Création de votre base de données

Exercice 1 — *Création de la base de données* Récupérez le dossier `tables.tar.gz`, que vous trouverez à l'adresse

`http://www.pps.jussieu.fr/~tasson/enseignement/IO2/tp13/tables.tar.gz`.

Au script, la création d'une base de donnée se fait par l'intermédiaire d'un formulaire, qui se trouve à l'adresse suivante

`http://pams.script.univ-paris7.fr/database/formulaire.html`.

Après y avoir entré votre login et votre mot de passe, ce formulaire créé tout seul une base de données vide, sur laquelle vous avez tous les droits, dont le nom est votre login et le mot de passe votre mot de passe.

Ensuite, toutes les modifications apportées à cette base seront faites par l'intermédiaire du client `mysql` de votre machine, après vous être connecté à votre base grâce à la commande :
`mysql -D login -h pams.script.jussieu.fr -u login -p`.

Exercice 2 — *Création des tables* Connectez vous au client `mysql`, et créez deux tables en respectant l'orthographe, la casse, et l'ordre de déclaration des champs :

- La table `Etudiant`, dont les champs sont les suivant :
 - `Num` de type entier
 - `Nom` de type chaîne de caractère de taille maximum 30
 - `Prenom` de type chaîne de caractère de taille maximum 30
 - `DateNaissance` de type date,
 - La clé primaire est le champ `Num`
- La table `Matiere`, dont les champs sont les suivant :
 - `Num` de type entier
 - `Nom` de type chaîne de caractère de taille maximum 30
 - `Lieu` de type chaîne de caractère de taille maximum 30
 - `Coeff` de type entier
 - `Enseignant` de type entier
 - `Description` de type chaîne de caractère de taille maximum 30
 - La clé primaire est le champ `Num`

Après avoir créé vos tables, remplissez les avec les fichiers que vous avez récupérés, en utilisant la commande `source nom_de_fichier` pour charger les fichiers `Etudiant.sql` et `Matiere.sql`.

2 Fichier de connexion à la base

Exercice 3 Toute connexion à une base de données doit contenir un nom d'utilisateur (`login`), un mot de passe (`votremotdepasse`), le nom du serveur (`pams.script.fr`) et le nom de la base de données (`login`) auxquels on souhaite se connecter. Créez un fichier `connect.php` contenant ces valeurs sous forme de définitions des constantes, `NOM`, `PASSE`, `SERVEUR` et `BASE`.

3 Recherche d'un étudiant

Exercice 4 — *formulaire XHTML* Cet exercice permet de rechercher un étudiant dans une base de donnée de l'université.

Créez un fichier `Formulaire1.html`, dont le titre est "formulaire de recherche d'un étudiant". Ce formulaire doit proposer deux types de recherche, soit par `nom`, soit par `date de naissance`.

- Le `nom` doit pouvoir être saisi dans une zone de texte, dont la valeur par défaut est le caractère `%`, qui remplace n'importe quelle chaîne de caractères.
- La `date de naissance` est comprise dans un intervalle de deux dates, indiquées dans deux zones de textes distinctes, sous la forme `aaaa-mm-jj`. La valeur par défaut du début de l'intervalle est `1936-01-01` et celle de la fin `2000-12-31`
- Enfin, l'utilisateur doit pouvoir choisir s'il veut combiner ou non les deux recherches : choix exclusif entre `ET` et `OU`.

Exercice 5 — *script PHP* Créez un fichier `exo1.php` qui, après s'être connecté à la base `login`, grâce à la commande `require("connect.php")` ; récupère la valeur des variables entrées dans le formulaire, et crée la requête `mysql` correspondant à la demande de l'utilisateur. Ce script php doit également afficher la nature de la demande, c'est à dire le nom et l'intervalle des dates de naissances.

Chaque étudiant renvoyé par `mysql` pourra être récupéré grâce à la fonction `mysql_fetch_object(resultat)`. Le script doit donc afficher, pour chaque étudiant sélectionné :

"L'étudiant numéro `numero` s'appelle `prenom nom`. Il s'agit d'un étudiant né en `date`."

Si la requête échoue, affichez un message d'erreur, et la nature de l'erreur renvoyée par `mysql`.

4 Mise à jour d'une base de données

Exercice 6 — *formulaire XHTML* Dans cet exercice, un formulaire doit permettre d'insérer ou de supprimer des éléments d'une base de données contenant les matières d'un parcours informatique.

Créez un fichier `Formulaire2.html`, dont le titre est "formulaire de mise à jour d'une base de données". Ce formulaire doit proposer :

- Le numéro de la matière, sous forme d'une zone de texte.
- Le nom de la matière, sous forme d'une zone de texte.
- Le lieu de la matière, `Jussieu`, `Halle aux farines`, sous forme d'un choix exclusif.
- Le coefficient de la matière, choix entre 1 et 5.
- Le numéro de l'enseignant, choix entre 1 et 9.
- Une brève description de la matière, sous forme d'une zone de texte.
- 3 boutons, `Insérer`, `Supprimer`, `Annuler`.

Exercice 7 — *script PHP* Créez un fichier `exo2.php` qui, après s'être connecté à la base `monitI02`, grâce à la commande `require("connect.php")` ; récupère la valeur des variables entrées dans le formulaire, et crée la requête `mysql` correspondant à la demande de l'utilisateur. Ce script `php` doit également afficher la nature de la demande.

Deux cas de figures se présentent :

- Si c'est une insertion :
 - Le script `php` doit vérifier que les données nécessaires à une insertion sont fournies, tous les champs sont obligatoires sauf la description de la matière. S'il manque une donnée, le script doit afficher un message d'erreur qui rappelle l'usage d'une insertion et sortir du programme.
 - Sinon, le script doit afficher le texte suivant "Insertion de la matière numero `numero`".
 - Après s'être connecté à la base de donnée, et avoir mis en place la requête `mysql` adéquate, deux cas se présentent :
 - si la requête fonctionne, le script doit afficher la nature de la requête qui a été effectuée ainsi que la valeur de chacun des attributs de la matière qui ont été ajoutés.
 - Si la requête n'a pas fonctionné, le script doit afficher la nature de la requête qui a échoué, ainsi que la raison de l'échec renvoyé par `mysql`, à l'aide de la fonction `mysql_error(connexion)`.
- Si c'est une suppression :
 - Le script `php` doit vérifier que les données nécessaires à une insertion sont fournies, c'est à dire le numéro de la matière. S'il n'est pas fourni, le script doit afficher un message d'erreur qui rappelle l'usage d'une suppression et sortir du programme.
 - Sinon, le script doit afficher le texte suivant "Insertion de la matière numéro `numero`".
 - Après s'être connecté à la base de donnée, le script doit stocker les attributs de la matière à supprimer à l'aide d'une requête `mysql` appropriée, afin de les afficher, plus tard si la requête de suppression fonctionne.
 - Après avoir mis en place la requête `mysql` de suppression d'une matière, si la requête fonctionne, le script doit afficher la nature de la requête qui a été effectuée ainsi que la valeur de chacun des attributs de la matière qui ont été ajoutés.
Si la requête n'a pas fonctionné, le script doit afficher la nature de la requête qui a échoué, ainsi que la raison de l'échec renvoyé par `mysql`.