Rapide révision MySQL

- SELECT
- FROM
- WHERE
    - OR/AND
    - IN
    - LIKE
- GROUP BY
- ORDER BY
- INNER JOIN
    - ON
- Variables de calculs ‘AS’
- Fonction de calculs
    - SUM
    - AVG
    - MIN
    - MAX
    - COUNT

Explication de PDO

→ new pdo('mysql:host=localhost;dbname=classicmodels', 'root', 'troiswa');

→ pdo→ Exec(‘SET NAMES UTF8’)

Explication des requêtes

→$query = pdo→ prepare(string query) // Une query s’écrit toujours entre guillemets lorsqu’on l'exécute dans php.

→ query->execute;

→ $query→ fetchAll(PDO::FETCH_ASSOC);

→ tableau 2 dimensions

[0 =>

[champ 1
champ 2
champ 3
…],

1=>

[champ 1
champ 2
champ 3
…],

var_dump();

Envoi de variables en <a href> et récupérations grâce à $_GET[“variable”] :

Lorsque l’utilisateur clique sur un lien, PHP à la possibilité d’envoyer une variable via l’url (ex : pour identifier un utilisateur spécifique ou une **commande** spécifique)

Cette variable est déclarée et ajoutée à l’URL par la déclaration suivante : ?var=

Cette déclaration intervient toujours après le nom de l’url spécifiée

Exemple :
$user;
// J’ai une variable user contenant le nom de mon utilisateur récupéré via une requête SQL.

<a href="profile.php?username=<?php echo $user?>>Profile

// Affiche un lien vers la page profile.php en ajoutant une variable username a l’url, pour générer le profil correspondant à l’utilisateur

Note :

“<?php echo” peut s’écrire plus rapidement comme ceci : ”<?=”

Récupération
PHP peut bien évidemment **récupérer** les variables envoyés via l’url par un lien. On utilisera pour ceci la variable $_GET[“variable”]
A contrario de $_POST[“champ”] qui lui est utilisé pour récupérer un champ directement posté par l’utilisateur dans un input, $_GET[“variable”] récupère une variable envoyé depuis un lien, souvent une valeur récupérée par une requête SQL pour afficher des informations relatives à la base de données.

Mon lien pointe vers profile.php et envoie une variable ?username=$user, $user étant une requête SQL contenant le nom de mon utilisateur

Voici comment je récupère cette variable envoyé dans profile.php :

profile.php :
$user = $_GET[“userName”];

// je déclare une variable $user (peu importe le nom) dans profile.php et je l’assigne à la valeur de $_GET[“username”] “username” étant le nom exact (sensible à la casse) que j’ai spécifié après le “?” dans mon envoi de variable via URL.

Si je veux afficher le contenu de ma variable via un echo :

echo $user;

// “Morgan”

Profile.php doit m’afficher les informations relatives à l’utilisateur que j’ai spécifié dans la variable d’url ($_GET)

J’aimerai faire une requête SQL dans laquelle je ne récupère que les informations de mon utilisateur actuel (Morgan),
J’ai le droit de concaténer un variable php dans une requete SQL mais ceci est fastidieux, lent, et sources de beaucoup d’erreurs lors de l’utilisation des guillemets :

```
'SELECT
      username
      age
      registerDate
   FROM users
   WHERE username=\" . $user . '\' ';
```

Difficile à comprendre et à écrire, notez la présence d'antislashs (caractère d'échappement) pour permettre à la requête de rester entre guillemets, immonde. J'appelle cela le quoting hell.

La bonne méthode pour insérer une variable récupérée en $_GET dans une query SQL et d'ajouter un ? dans le champ de la requête que l'on ne connaît pas encore est la suivante :

```
'SELECT
      username
      age
      registerDate
   FROM users
   WHERE username=?';
```

On voit ici que la clause WHERE de la requête est en attente d'une valeur grâce au ?. Au moment de l'exécution de la requête, je peux lui envoyer les paramètres qu'il lui manque :

```
$query->execute(array($user));

// Ici le "?" deviendra "Morgan" car la variable $user a été envoyé via l'url et que j'envoie à ma requête lors de son exécution
```

Notez que l'on envoie la variable au sein d'un array lors du execute, c'est la marche à suivre pour envoyer des paramètres à une requête.

Enoncé :

Tableau des commandes

Le principe de ce TP réside dans l'utilisation d'HTML pour afficher des résultats de requêtes SQL.
Il est composé de 2 parties fonctionnelles :

- Un tableau HTML de toutes les commandes de la base de données classicmodels
- Un tableau HTML affichant la commande en détails → client, montant, etc lorsque l'on clique sur un numéro de commande du tableau de commandes.

Le TP tableau de commande contient 4 fichiers :

- Index.php → contiendra la connexion à MySQL + la requête SQL à exécuter pour pouvoir générer le tableau HTML
- Index.phtml → contiendra le tableau HTML généré dynamiquement
- order-form.php → contiendra la connexion à MySQL + la requête SQL à exécuter pour afficher la commande en détails
- order-form.phtml → contiendra le tableau HTML affichant les détails de la commande

Enoncé Partie 1 :
Afficher un tableau HTML généré dynamiquement comprenant les informations suivantes concernant toutes les commandes de la base de données classic_models :

- Numéro de commande
- Date de commande
- Date de livraison
- Statut de la commande

Toutes les informations sont rangées par date de commande.
Toutes les informations doivent être récupérées au moyen de l'objet PDO et d'une requête SQL.
Lorsque l'on clique sur un numéro de commande cela doit envoyer vers la page order-detail.php

Partie 2 :

order-detail.php doit afficher toutes les informations relatives au client qui a passé **la commande choisie**
Il doit indiquer également la liste des produits achetés sous forme de tableau pour **la commande choisie**
A côté de chaque produit devra également être indiqué
→ le prix unitaire, la quantité achetée, le prix total de toute la quantité achetée
Tout en bas du tableau à droite, indiquer :
→ le montant total HT (does that ring a bell)

→ le montant de la TVA
→ le montant total TTC

Un lien “Retour à l’accueil” tout en haut à gauche de la page devra être affiché et doit nous permettre de revenir sur la liste des commandes. (notre page principal)

Tips & Tricks :

- Rappelez vous que le TP est découpé en 2 parties, concentrez vous déjà sur le php de la partie 1.
- PHP & HTML fonctionnent en couple :
  Imaginez votre phtml comme la suite directe de votre code PHP, d’abord je récupère les informations SQL dans mon php, puis j’include mon phtml et j’utilise dans mon phtml les informations récupérées dans mon php.
- Essayez dans un 1er temps de compartimenter le travail, commencez par reussir une requête SQL en php et affichez la avec var_dump().
- Vous pouvez utiliser la fonction number_format() pour formater un nombre

Voici quelques liens et quelques références pour le TP

http://php.net/manual/fr/pdo.exec.php
http://php.net/manual/fr/pdo.prepare.php
http://php.net/manual/fr/pdostatement.execute.php
http://php.net/manual/fr/pdostatement.fetchall.php
http://php.net/manual/fr/function.number-format.php

Bonne chance :DDDDDDDDD