RT 1A – Bases de données – TP4 Manipulation d'une base de données MYSQL à partir de PHP

Gilles Lebrun - IUT ifs, département RT

1) Les objectifs

- Interrogation d'une base de données MYSQL à partir de scripts PHP,
- Introduction au langage de programmation PHP et au langage de description HTML,
- Architecture client-serveur avec comme client un navigateur web et comme serveur un système LAMP.

2) Serveur LAMP

Pour réaliser l'interrogation de votre base de données prénom_nom (que vous avez créé au TP2) avec des scripts PHP, nous avons besoin d'un serveur LAMP (Linux, Apache, Mysql, Php). Nous allons utiliser le serveur spartacus (spartacus.iutc3.fr) de l'IUT d'ifs qui possède un serveur HTTP Apache avec le support de PHP et un serveur Mysql qui sont déjà configurés pour un système d'exploitation Linux.

Si vous voulez réaliser les mêmes manipulations chez vous, allez voir le lien suivant: <u>http://doc.ubuntu-fr.org/lamp</u> (la procédure est pour Ubuntu, cependant elle est très proche pour d'autres distributions de Linux). Vous pouvez également utiliser EasyPHP pour avoir un serveur WAMP (W pour Windows).

3) Configuration de votre compte personnel de l'IUT

Pour que le serveur Apache sur Spartacus puisse exploiter vos documents HTML et script PHP vous devez créer dans votre répertoire « home » un répertoire public_html. Donnez lui les droits 755 (tous les fichiers dans ce répertoire devront avoir au moins les droits 644). Créez ensuite dans ce répertoire un fichier index.html avec les données suivantes :

```
<html>
<head> <title> Ma page index.html </title> </head>
<body> <h1> Essai de mon public_html </h1> </body>
</html>
```

Nous ne rentrerons pas dans les détails du html (et encore moins du xhtml), vous devez juste savoir pour qu'un document soit visualisable sur votre navigateur avec un minimum de mise en forme, il doit être au format HTML. Le HTML est un langage de description (*HyperText Markup Langugage*) constitué d'un ensemble de TAGs. Un TAG est un mot clé entre les symboles < et >. Pour plus d'informations sur le HTML, voir les liens suivants :

- <u>http://www.la-grange.net/w3c/xhtml1/</u> (spécification traduite en français),
- <u>http://www.commentcamarche.net/contents/html/htmlbalise.php3</u> (les balises en HTML),
- <u>http://openweb.eu.org/articles/xhtml_une_heure</u> (introduction rapide au XHTML),

Vérifiez en tapant l'adresse <u>http://spartacus.iutc3.unicaen.fr/~nom.prenom</u> (pensez à changer nom et prénom par les vôtres) dans votre navigateur, que vous avez bien accès à la page de votre site personnel. Normalement une page d'authentification doit s'afficher, entrez votre identifiant et mot de passe correspondant à votre compte de l'IUT d'ifs. Si vous avez bien réalisé les opérations précédentes, vous devez voir apparaître le message: *Essai de mon public_html*.

4) Vérification du support de PHP par le serveur APACHE

Pour pouvoir exécuter des requêtes SQL, sur le serveur SPARTACUS, afin d'interroger la base de

données MYSQL et produire en retour un document HTML, il est nécessaire d'avoir un serveur web HTTP (Apache) avec PHP (sigle de PHP: Hypertext Preprocessor). PHP est un langage de script (donc un langage de programmation) permettant de produire des document web de façon dynamique. Pour vérifier la présence de PHP sur spartacus, écrivez le code ci-dessous dans un fichier que vous nommerez info.php (enregistrerez ce fichier dans votre répertoire public_html):

<?php phpinfo(); ?>

Vérifiez à partir de l'URL suivante: <u>http://spartacus.iutc3.unicaen.fr/</u> <u>~nom.prenom/info.php</u> que ce script PHP minimaliste est bien exécuté par le serveur Apache. Vous devez obtenir des informations précisant que c'est la version 5.2.0-8 de PHP qui est exploitée à partir d'un serveur Apache 2. En regardant plus bas, vous pouvez également observer que le support de MYSQL (version 5.0.32) est actif.

5) Initiation rapide à PHP

Les instructions PHP sont comprises entre <?php et ?> (<?php peut être réduit en <? lorsqu'il n'y a pas d'ambiguïté). Comme dans un grand nombre de langage de programmation chaque instruction est séparée de la suivante par un point virgule. La fonction phpinfo() produit un document HTML (faire CTRL+U, dans Firefox, pour avoir le code HTML généré) donnant les détails de la version de PHP présente sur le serveur web. Les variables en PHP commence systématiquement par le symbole \$ et le typage est dynamique, c-à-d que vous n'avez pas à déclarer le type d'une variable comme en C (int x=5; en C devient x=5; en PHP). Comme en C, il est possible de réaliser des boucles et des tests avec les instructions for et if. L'écriture sur la sortie standard en PHP est réalisée avec l'instruction print (ou echo) qui est proche du fonctionnement du printf en C. Tous ce que vous apprenez actuellement en C est donc facilement transposable en PHP, sauf qu'au lieu que la sortie standard en C soit dirigée vers la console, elle est redirigée (grâce au protocole HTTP) vers votre navigateur. Pour vous convaincre de la similitude, voici le même programme écrit en C et en PHP. Saisissez la version PHP dans un fichier que vous nommerez simple.php et observez le résultat produit.

```
#include <stdio.h>
                                            <?php
int main()
                                              header("Content-Type: text/plain");
{
                                              print "[debut]\n\n";
 printf("[debut]\n\n");
                                               $i=1; $i_max=11;
  int i=1, i_max=11;
                                               $val=0.5;
  float val=0.5;
                                               $mess = "A l'itération ";
                                               for($i=0; $i< $i_max; $i++){</pre>
  char* mess="A l'itération %d La valeur
                                                 print $mess.$i." La valeur est $val
est %f\n";
  for (i=1; i<i_max;i++){</pre>
                                            n";
    printf(mess,i,val);
                                                 $val /= 2;
                                                 if ($i % 2 == 0) echo "\n";
    val/=2;
    if (i % 2 ==0) printf("\n");
                                               }
                                              print "[fin]\n";
  }
  printf("[fin]\n");
                                            ?>
  return 0;
```

L'instruction header ("Content-Type: text/plain"); permet de spécifier que le contenu transporté par HTTP est un simple fichier texte et non du HTML à partir de l'entête envoyée par Apache au navigateur. Sans cette ligne, le comportement par défaut de PHP est d'envoyer un entête indiquant que le contenu envoyé est du HTML (équivalent à un header ("Content-Type: text/html");). Vérifiez ce qui se passe si vous supprimez la ligne header... du fichier simple.hml, puis comparez le résultat affiché par votre navigateur au code reçu (affichage \rightarrow code source de la page).

Pour que le résultat produit par un script PHP soit lisible par un humain, il est obligatoire de prendre en compte la production de tag HTML. En particulier la production de tag
 pour obtenir des sauts de

lignes dans l'affichage du navigateur. Ces tags peuvent être produit par des print ou directement ajouté dans le fichier PHP en mélangeant code HTML et script PHP. L'exemple ci-dessous est une traduction de simple.php pour qu'il produisent du code HTML pour le navigateur. **Enregistrez ce code dans un fichier que vous nommerez simple_html.php et exécutez le:**

```
<html>
<head> <title> <?php print "Simple HTML" ?> </title> </head>
<body>
<?php print "[debut]<br/>>\n"; ?>
<hr/>
<?php
  $i=1;
  $i_max=11;
  $val=0.5;
  $mess = "A l'itération ";
  for($i=0; $i< $i_max; $i++){</pre>
    print $mess.$i." La valeur est $val<br/>>\n";
    $val /= 2;
    if ($i % 2 == 0) echo "<br/>\n";
  }
?>
<hr/> <?php print "[fin]<br/>\n"; ?>
</body> </html>
```

Cette brève introduction à PHP devrait vous permettre de comprendre les scripts PHP suivants relatifs à l'exploitation de MYSQL. Cependant, pour allez plus loin, dès maintenant, avec PHP voici quelques URL utiles :

- <u>http://www.manuelphp.com/</u> (surement le site en français indispensable à connaître),
- <u>http://www.php.net/</u> (le site PHP de référence),
- <u>http://www.phpfrance.com/</u> (une actualité fournie sur PHP).

6) Connexion à MYSQL à partir de PHP

Nous allons maintenant entrez dans le vif du sujet en écrivant un premier script PHP permettant la connexion à votre base de données. Le squelette de ce script est le suivant:

```
<!-- fichier connect_mysql.php permettant la connexion à votre base mysql-->
<html>
<head> <title> Connexion a MYSQL </title> </head>
<body>
<?php
     $Host = "spartacus"; // nom du serveur
     $User = "prenom_nom"; // nom de login pour mysql
     $Password = "votre_mot_de_passe"; // mot de passe
     // la connexion à la base mysql à travers la fonction PHP mysql_connect
     $connexion = mysql_connect($Host,$User,$Password);
     // la variable connexion contient un identifiant sur la connexion établie
     // pour les opérations suivantes sur cette base
     if ($connexion) {
           print " Connexion établie ";
     }
     else {
           print " Connexion impossible à $Host ";
     }
?>
</body>
</html>
```

Saisissez le code ci-dessous dans un fichier que vous nommerez mysql_connect.php. Pensez

à changer le contenu des variables \$User et \$Password afin qu'elles soient adaptées aux paramètres de votre configuration MYSQL. Puis testez la bonne exécution de votre script avec votre navigateur.

7) Réalisation d'une requête SQL à partir de PHP

Modifiez le script précédent, afin de le transformer comme ci-dessous. Nommez ce nouveau script premier_select.php. Premièrement, vous devez donner à la variable \$BDD le nom de votre base (il doit être identique à votre identifiant MYSQL). La fonction PHP mysql_select_db() est équivalente à un *use* dans l'environnement interactif de MYSQL. La requête SQL à exécuter sur cette base est codée comme une chaîne de caractère dans la variable \$requete_sql. Vérifiez que le *select* *... exécuté par ce script PHP extrait correctement les informations de la table LOGICIEL produite au TP3.

```
<!-- fichier premier_select.php permettant d'exécuter une requête select-->
<html>
<head> <title> Premier select </title> </head>
<body>
<?php
     $Host = "spartacus"; // nom du serveur
     $User = "prenom_nom"; // nom de login pour mysql
     $Password = "votre mot de passe"; // mot de passe
     $BDD = "prenom_nom"; // nom de la base
     $connexion = mysql_connect($Host,$User,$Password);
     if ($connexion) {
          print " Connexion établie avec
$User@$Host\n";
     }
     else {
          print " Connexion impossible à $Host \n";
     }
     mysql_select_db($BDD,$connexion); // sélection d'une base de données
     $requete_sql = "select * from LOGICIEL;"; // la requête SQL à exécuter
     $id_resultat = mysql_query($requete_sql,$connexion);
     // $id_result correspond à un identifiant sur le résultat de la requête
     // cet identifiant est nulle si la requête à échoué
     if (!$id_resultat) {
          print " Erreur d'exécution de la requête : $requete_sql !<br/>";
          print " Code : ".mysql_errno()."";
          print "Message : ".mysql_error()."";
     } else {
          print " ok ! \n";
          while($ligne = mysql_fetch_row($id_resultat)){
               foreach($ligne as $val) print $val."    ";
               print "<br/>\n";
          }
          print "\n";
     }
     print " FIN ! ";
?>
</body>
</html>
```

Quelques explications complémentaires sur les 4 fonctions PHP relatives à MYSQL: mysql_query, mysql_errno, mysql_error, mysql_fetch_row utilisées dans ce script PHP.

La première fonction **mysql_query** est essentielle, car elle permet l'exécution d'une requête SQL stockée dans une chaîne de caractères (ici \$requete_sql) par le serveur MYSQL. Cette fonction retourne un résultat (un identifiant pour être plus précis) permettant par la suite d'explorer la réponse à la requête SQL. Si l'identifiant est nulle, il y a eu une erreur. Dans ce cas, les fonctions mysql_errno et

mysql_error permettent d'obtenir respectivement le numéro de l'erreur et une description textuel de cette erreur.

Dans l'autre cas (pas d'erreur), la dernière fonction **mysql_fetch_row** permet d'obtenir une ligne du résultat de la requête SQL. Le premier appel à **mysql_fetch_row** retourne la première ligne. Chaque appel suivant à **mysql_fetch_row** retourne la ligne suivante du résultat de la requête SQL, ou *nulle* lorsqu'il n'y a plus d'autre ligne. Chaque résultat (ligne) retournée par la fonction **mysql_fetch_row** est organisée dans tableau à une dimension contenant *n* valeurs, où *n* correspond au nombre de champs (colonnes) produits par la requête SQL. L'instruction foreach permet de parcourir une à une les valeurs dans ce tableau.

Modifiez le script PHP précédent pour afficher le contenu globale de la table POSTE. Modifiez ensuite ce script pour obtenir pour chaque nom de poste le nom de la salle lui correspondant (pensez à faire une jointure entre la table POSTE et SALLE.

8) Formulaire et script PHP

Généralement, il est nécessaire de demander des informations à l'utilisateur avant de réaliser une requête SQL à partir d'un script PHP pour agir sur la base de données. Par exemple, l'insertion (*insert into* ...) d'une ligne dans une table fait typiquement partie de ce cas de figure.

Cette collecte d'informations est réalisée à partir d'un formulaire à remplir dans le navigateur. La réalisation d'un formulaire demande l'utilisation d'un ensemble de tags HTML qui ne seront pas décris précisément ici (mais explications de l'enseignant pendant la séance de TP). Le script insertion.php, ci-dessous, correspond à un exemple de formulaire qui va permettre de peupler la table LOGICIEL.

```
<?php // Script insertion.php
$titre_page=array("interrogation pour insertion", "Demande d'informations pour
insertion dans la table LOGICIEL");
include("debut.php"); ?>
<h2> Ajout d'un logiciel </h2>
<form action="insertion_exec.php" method="GET">
  numero de la clef : 
 <input type="text" name="clef" value="" size="3"/>  
  nom du logiciel : 
 <input type="text" name="nom_logiciel" value="" />  
  date d'installation : 
 <input type="text" name="date_install" value="" size="10"/> 
  version du logiciel : 
 <input type="text" name="version_logiciel" value="" />  
  type du logiciel : 
 <input type="text" name="type_logiciel" value="" />  
  nombre d'installations 
 <input type="text" name="nbr_install" value="" />  
<br/> <br/>
<input type="submit" value="Ajouter"/> &nbsp; &nbsp; &nbsp;
<input type="reset" value="Effacer tout"/>
<br/> <br/>
</form>
<? include("fin.php");?>
```

Les éléments essentiels de ce formulaire sont que l'appuie sur le bouton *Ajouter* demande l'exécution du script insertion_exec.php (valeur de action dans le tag <form>) et que les valeurs des champs textes remplies seront transmis dans l'url (method = GET dans le tag <form>). Les contenus des champs textes seront identifiés grâce à leurs noms (attribut name dans les différents tags <input type="text">).

Pour faciliter l'écriture du début et de la fin du document HTML produit par un script PHP deux fichiers annexes debut.php et fin.php sont utilisés (inclus dans le script). Ils ont les contenus suivants:

```
<!-- debut.php -->
<html>
<head> <title> <?php print $titre_page[0]?> </title> </head>
<body>
<h1> <?php print $titre_page[1]?> </h1>
<hr/><</pre>
```

```
<!-- fin.php -->
<hr/><hr/>
<?php
date_default_timezone_set('UTC');
echo date('l jS \of F Y h:i:s A');
?>
</head>
</html>
```

L'insertion d'un logiciel dans la table LOGICIEL à partir des valeurs du formulaire est réalisée grâce au script PHP ci-dessous. La récupération des contenus des champs textes (à partir de l'URL) est réalisée à partir des 5 instructions de la forme: **\$name = "log".\$_GET['name'];**. La requête SQL **insert into LOGICIEL values** ... est ensuite construite à partir des valeurs de ces 5 variables, puis exécutée. Un bouton (contenu dans un formulaire) permet de revenir au script insertion.php.

```
<?php // script insertion_exec.php</pre>
$titre page=array("insertion dans la table LOGICIEL","Insertion dans la table
LOGICIEL des données du formulaire");
// echo $_SERVER['REQUEST_URI']."<br/>br/>";
include("debut.php");
include("connexion.php");
$clef = "log".$_GET['clef'];
$nom_logiciel = $_GET['nom_logiciel'];
$date_install = $_GET['date_install'];
$version_logiciel = $_GET['version_logiciel'];
$type_logiciel = $_GET['type_logiciel'];
$nbr_install = $_GET['nbr_install'];
echo "données transmises : $clef - $nom_logiciel - $date_install -
$version_logiciel - $type_logiciel - $nbr_install <br/> <br/> <br/>";
$requete_sql = "insert into LOGICIEL values
(\"$clef\",\"$nom_logiciel\",\"$date_install\",\"$version_logiciel\",\"$type_logic
iel\",$nbr_install);";
echo "requete-sql= $requete_sql <br/> <br/>";
$id_resultat = mysql_query($requete_sql,$connexion);
if (!$id_resultat) {
     print " Erreur d'exécution de la requête : $requete_sql !<br/>";
     print " Code : ".mysql_errno()."";
     print " Message : ".mysql_error()."";
} else {
     print " insertion dans la table LOGICIEL réalisée
";
}
?>
<form action="insertion.php" method="GET">
<input type="submit" value="Retour à insertion"/>
</form>
<?php include("fin.php"); ?>
```

Le script connexion.php, qui est inclus dans insertion_exec.php, permet de centraliser les informations relatives à une connexion à la base MYSQL. Ce script est très proche du script connect_mysql.php, sauf qu'il inclut également la sélection de la base. Son contenu est le suivant:

```
<?php // script connexion.php
  $Host = "spartacus"; // nom du serveur
  $User = "gilles_lebrun"; // nom de login pour mysql
  $Password = "*******"; // mot de passe
  $BDD = "gilles_lebrun"; // nom de la base
  $connexion = mysql_connect($Host,$User,$Password);
  if ($connexion) {
      print " Connexion établie avec
  $User@$Host\n";
   }
   else {
      print " Connexion impossible à $Host \n";
   }
   mysql_select_db($BDD,$connexion); // sélection d'une base de données
  ?>
```

Saisissez les différents fichiers PHP précédents et vérifiez que l'insertion dans la table LOGICIEL est effective.

Produisez deux fichiers PHP : insertion_poste.php et insertion_poste_exec.php permettant de réaliser une opération similaire à la précédente, mais avec la table POSTE.

9) Allez plus loin en PHP

Les exercices suivants ne sont pas obligatoires, mais si vous avez suffisamment d'avance et/ou vous souhaitez en savoir un peu plus sur HTML, MYSQL et PHP, vous pouvez les réaliser en TP (et/ou chez vous).

Dans la section 8 précédente, le choix du type de logiciel n'est pas contraint à ceux présents dans la table TYPE. Nous voulons remédier à cela en remplaçant le champ texte relié à type_logiciel par une liste déroulante contenant l'ensemble des types présent dans cette table. L'exemple ci-dessous, vous montre la syntaxe d'une liste déroulante en HTML. Modifiez insertion.php et insertion_exec.php pour prendre en compte la remarque précédente.

```
<html>
<head> <title> exemple avec balise select </title> </head>
<body>
<form action="test_balise_select.php" method="GET" name="test_select">
Ville :
  <select name="ville">
    <option value="Paris"> Paris et arrondissement </option>
    <option value="Caen" selected> Caen et Herouville St-Clair</option>
    <option value="Rennes"> Rennes </option>
  </select> <br/>
Logement :
  <select name="logement">
    <option value="F1"> Studio </option>
    <option value="F2"> Salle + Chambre </option>
    <option value="F2bis"> Salle + Salon + Chambre </option>
    <option value="F3"> Salle + 2 Chambres </option>
  </select><br/>
  <input type="submit" value="Transmettre" name="transmettre"/>
  <input type="submit" value="Envoyer" name="envoyer"/>
  <input type="reset" value="Effacer tout"/>
</form>
</body> </html>
```

La suppression d'une ligne d'une table n'a pas été réalisée à partir d'un script PHP dans les sections précédentes. Il suffit pour cela d'exécuter une requête SQL delete ... sur la bonne ligne d'une table. A partir du script suppression.php ci-dessous, produisez le script suppression_exec.php permettant de réaliser la suppression de la ligne sélectionnée dans la table LOGICIEL et TYPE. Dans

ce cas de la suppression d'un type, supprimez au préalable l'ensemble des logiciels de ce type.

```
<?php
$titre_page=array("Suppression dans LOGICIEL","Suppression de tuple dans la table
LOGICIEL");
// echo $_SERVER['REQUEST_URI']."<br/>br/>";
include("debut.php");
include("connexion.php");
$requete_sql = "select N_LOG, NOM_L, VERSION, NOM_TYPE from LOGICIEL, TYPE where
TYPE_L=TYP_LP;"; // la requête sql que l'on va demander à sql d'exécuter
$id_resultat = mysql_query($requete_sql,$connexion);
if (!$id_resultat) {
     print " Erreur d'exécution de la requête : $requete_sql !<br/>";
     print " Code : ".mysql_errno()."";
     print " Message : ".mysql_error()."";
} else {
     $couleur= array("#33FFFF","#33FF99");
     echo "<h2> Suppression d'un logiciel </h2>";
     echo "\n<form action=\"suppression_logiciel_exec.php\" method=\"GET\
name=\"sup1\" >\n";
     echo "<input type=\"hidden\" name=\"suppression\" value=\"logiciel\" /> \n";
     echo "\n";
     $cnt=0;
     while($ligne= mysql_fetch_object($id_resultat)){
       $n = $cnt % 2;
      echo " ";
      echo "  $ligne->NOM_L ($ligne->VERSION)   -   $ligne-
>NOM TYPE    \n";
      echo "<input type=\"submit\" value=\"supprimer\" name=\"$ligne->N_LOG\"/>
 \n";
      $cnt++;
     }
     echo " </form>\n";
$requete_sql = "select distinct TYP_LP, NOM_TYPE from TYPE;";
$id_resultat = mysql_query($requete_sql,$connexion);
if (!$id_resultat) {
     print " Erreur d'exécution de la requête : $requete_sql !<br/>";
     print " Code : ".mysql_errno()."";
     print "Message : ".mysql_error()."";
} else {
     $couleur= array("#22EEEE","#22EE88");
     echo "<h2> Suppression de tous les logiciels d\'un type donné </h2>";
     echo "\n<form action=\"suppression_logiciel_exec.php\" method=\"GET\"
name=\"sup2\" >\n";
     echo "<input type=\"hidden\" name=\"supression\" value=\"type\" /> \n";
     echo "\n";
     $cnt=0;
     while($ligne= mysql_fetch_object($id_resultat)){
       $n = $cnt % 2;
      echo " ";
      echo "  $ligne->NOM_TYPE    \n";
      echo "<input type=\"submit\" value=\"supprimer\" name=\"$ligne->TYPE_LP\"/>
 \n";
       $cnt++;
     }
     echo " </form>\n";
}
include("fin.php");
?>
```