

TP PHP et MySQL

PHP et MySQL.

MICROSOFT

[Sélectionnez la date]

Créé par : Neila Ben Lakhal

TP PHP et MySQL

PHP et MySQL.

Rappel : Bases des interactions PHP-MySQL

La première chose à faire lorsque vous souhaitez accéder en PHP à une base de données est de vous connecter à un serveur de base de données qui héberge cette base, puis de vous connecter à la base proprement dite.

La connexion au serveur MySQL se fait via la fonction `mysql_connect()`, qui vous permet de préciser l'adresse du serveur, ainsi que le nom sous lequel vous vous connectez (login) et le mot de passe. Une fois connecté à un serveur de base de données, il est nécessaire d'ouvrir une base proprement dite. Pour ceci, utilisez la fonction `mysql_select_db()` en précisant le nom de la base en question.

Les identifiants de connexion ne sont pas nécessaires si on ne se connecte qu'à une seule base à la fois, ils permettent seulement de lever toute ambiguïté en cas de connexions multiples.

Vous avez maintenant accès à la base de données sélectionnées, avec les droits correspondant à l'identité sous laquelle vous vous êtes connectés au serveur. Cela signifie que vous ne pourrez pas forcément effectuer toutes les opérations que vous souhaitez sur la base.

Résumé des fonctions de connexion/déconnexion :

- `mysql_connect($server, $user, $password)` : permet de se connecter au serveur

`$server` en tant qu'utilisateur `$user` avec le mot de passe `$password`, retourne l'identifiant de connexion (noté ici `$id`) si succès, `FALSE` sinon.

- `mysql_select_db($base[, $id])` : permet de choisir la base `$base`, retourne `TRUE` en cas de succès, sinon `FALSE`.

- `mysql_close([$id])` : permet de fermer la connexion.

- `mysql_pconnect` : idem que `mysql_connect` sauf que la connexion est persistante, il n'y a donc pas besoin de rouvrir la connexion à chaque script qui travaille sur la même base.

Exemple :

```
// la fonction "die" permet d'arrêter le script et d'afficher un message.
if($id = mysql_connect('localhost', 'login', 'passwd')) {
    if($id_db = mysql_select_db('mabase')) {
        echo 'Succès de connexion.';
        /* code du script ... */
    } else {
        die('Echec de connexion à la base.');
```

```
    }
    mysql_close($id);
} else {
    die('Echec de connexion au serveur de base de données.');
```

Pour envoyer une requête à une base de donnée, on utilise la fonction : `mysql_query($str)` qui prend en paramètre une chaîne de caractères (ici, `$str`) contenant la requête SQL et qui retourne un identificateur de résultat ou `FALSE` si échec.

Pour effectuer une requête sur la base de données, on procède généralement comme suit :

Exemple :

```
// Construction d'une requête SQL permettant de récupérer le numéro
// de téléphone d'un client
$query = "SELECT telephone FROM client WHERE nom=\"'$nom'\"";
// exécution de la requête SQL
$result = mysql_query($query);
```

Attention, ici, les requêtes ne se terminent pas par un point virgule ";" et n'autorisent pas les `SELECT` imbriqués.

L'identificateur de résultat `$result` permet à d'autres fonctions d'extraire ligne par ligne les données retournées par le serveur.

Sous ORACLE, l'affichage des résultats d'une requête se fait ligne par ligne, sous MySQL, c'est identique. Une boucle permettra de recueillir chacune des lignes à partir de l'identifiant de résultat.

Une ligne contient (sauf cas particulier) plusieurs valeurs correspondant aux différents attributs retournés par la requête. Ainsi, une ligne de résultat pourra être sous la forme d'un tableau, d'un tableau associatif, ou d'un objet.

`mysql_fetch_row($result)` : retourne une ligne de résultat sous la forme d'un tableau. Les éléments du tableau étant les valeurs des attributs de la ligne. Retourne `FALSE` s'il n'y a plus aucune ligne.

Exemple :

```
$requet = "SELECT * FROM users";
if($result = mysql_query($requet)) {
while($ligne = mysql_fetch_row($result)) {
$id = $ligne[0];
$name = $ligne[1];
$address = $ligne[2];
echo "$id - $name, $address <br>";
}
} else { echo 'Erreur de requête de base de données.'; }
```

Ici, on accède aux valeurs de la ligne par leur indice dans le tableau.

`mysql_fetch_array($result)` : retourne un tableau associatif. Les clés étant les noms des attributs et leurs valeurs associées leurs valeurs respectives. Retourne `FALSE` s'il n'y a plus aucune ligne.

Exemple :

```
$requet = "SELECT * FROM users";
if($result = mysql_query($requet)) {
while($ligne = mysql_fetch_array($result)) {
$id = $ligne["id"];
$name = $ligne["name"];
$address = $ligne["address"];
echo "$id - $name, $address <br>";
}
} else { echo 'Erreur de requête de base de données.'; }
```

Ici, on accède aux valeurs de la ligne par l'attribut dans le tableau associatif.

Quelques fonctions supplémentaires très utiles :

-mysql_free_result(\$result) : efface de la mémoire du serveur les lignes de résultat de la requête identifiées par \$requet. Très utile pour améliorer les performances du serveur.

-mysql_insert_id([\$id]) : retourne l'identifiant d'un attribut clé primaire

AUTO_INCREMENT de la dernière insertion.

-mysql_num_fields(\$result) : retourne le nombre d'attributs du résultat.

-mysql_num_rows(\$result) : retourne le nombre de lignes du résultat. Cela permet ainsi de remplacer le while par un for.

Remarque générale : pensez à bien tester la valeur de retour des fonctions (mysql_query et les autres).

l'interface PHPMyAdmin

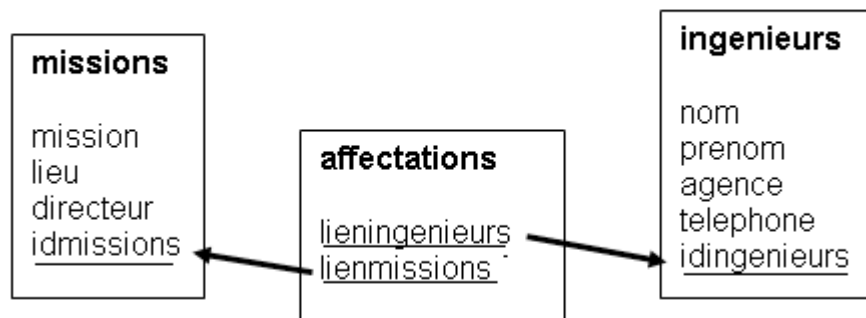
phpMyAdmin est une interface web d'accès au programme de bases de données MySQL, écrite en php. Pour y accéder avec le navigateur, ouvrez la page sur le serveur : <http://localhost/phpMyAdmin/>

CREATION D'UNE TABLE EN UTILISANT PHPMYADMIN:

La base de données sur laquelle nous allons travailler dans ce TP a pour but de gérer le suivi des missions confiées aux ingénieurs d'une SI. Pour une première approche du problème, l'entreprise a fourni les exemples de données suivants :

Dominique SZTUMSKY (agence de Lyon, tél 3027)					
Mission	lieu	directeur			
COJO	Lyon	BALDENT			
TGV	Lyon	BALDENT			
ARIANE	Toulouse	SRINOV			
35	Paris	FENDU			
MODULA	Paris	BALDENT			
Jérôme HAFIDI (agence de Renne, tél 2878)					
Mission	lieu	directeur			
COJO	Lyon	BALDENT			
ARIANE	Toulouse	SRINOV			
EXPO	Déville	MILDON			
MODULA	Paris	BALDENT			
Marc DUCOSTAUD (agence de La Défense, tél 4181)					
Mission	lieu	directeur			
COJO	Lyon	BALDENT			
TGV	Lyon	BALDENT			
BALTIQUE	Copenhague	MILDON			

La structure relationnelle proposée est constituée de trois tables MySql :



1. Connexion au serveur de bases de données

Accédez à l'interface de PhPMyadmin, accédez au menu privilège et créez un nouveau utilisateur « user » qui a pour mot de passe « user » et qui a tous les privilèges.

- Écrire le script **connect.php** qui définit les paramètres de connexion \$host, \$user, \$bdd et \$password correspondant respectivement au nom du serveur (localhost), à votre login, à votre base de données (test) et à votre mot de passe (user) et qui permet de se connecter au serveur local.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11
/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/HTML; charset=iso-8859-1" />
<title>connect.php</title>
</head>
<body>
<?php
//Declaration des parametres de connexion
$host="localhost";
$user="neila";
$bdd="test";
$password= "neila";
//Connexion au serveur
$connexion=@mysql_connect($host,$user,$password)
or die("Erreur de connexion au serveur");
//Connexion a la base
mysql_select_db($bdd,$connexion) or die("Erreur de connexion a la base");
?>
</body>
</html>
```

2. Création des tables

Avec un script php

- Écrire le script **creation_table_ingenieurs.php** qui permet de créer la table “ingenieurs” ci-dessus. La requête correspondante à la création de la table ingénieur sera par exemple de la forme :

```
CREATE TABLE ingenieurs (idingenieurs INT PRIMARY KEY, nom VARCHAR(20), prenom VARCHAR(30), agence VARCHAR(30), telephone VARCHAR(4))
```

Si vous avez bien fait l'exercice précédent, vous devez trouver la table “ingenieurs” dans votre base de données.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/HTML; charset=iso-8859-1" />
<title>creation_table_ingenieurs.php </title>
</head>
<body>
<?php
include("connect.php");
//Requete de creation de la table "ingenieurs"
$requete_ingenieurs="CREATE TABLE ingenieurs (idingenieurs INT PRIMARY KEY,
nom VARCHAR(20), prenom VARCHAR(30), agence VARCHAR(30), telephone VARCHAR(4))";
$resultat=mysql_query($requete_ingenieurs) ;
if ($resultat){
echo "Table ingenieurs creee";
}
else echo "Requete incorrecte pour creation de la table ingenieurs";
?>
</body>
</html>
```

En utilisant phpMyadmin

- Créer à présent les deux autres tables (“missions” et “affectations”) en utilisant l'interface graphique.

Pour cela, vous cliquez sur la base, phpMyAdmin vous propose alors de créer une nouvelle table en spécifiant son nom et le nombre de champs qu'elle contient. Lorsque ces champs sont remplis, cliquez sur “exécutez”. La page suivante vous permet de saisir les champs ainsi que leurs caractéristiques (ne pas oublier de définir les champs clés). Cliquez ensuite sur “Sauvegarder”. Vous obtenez alors une page sur laquelle vous retrouvez, entre autres, la requête qui a été exécutée.

- Créez de même la table “affectations” en prenant soin de définir les deux champs comme étant des clés.

3. Insérer des données dans une table via phpMyAdmin

La structure de vos tables étant créée, nous allons les remplir. Une première méthode consiste à utiliser tout simplement l'interface phpMyAdmin. Nous allons utiliser cette dernière afin de saisir les coordonnées des ingénieurs.

- En utilisant les données de l'entreprise, saisissez les coordonnées des trois ingénieurs. Pour cela, vous sélectionnez la table "ingénieurs" et cliquez sur le lien "insérer".

4. Insérer des données dans une table via un script php

Nous allons remplir la table "missions" sans utiliser l'interface phpMyAdmin.

Créer un script php qui va renvoyer directement les informations contenues dans un formulaire html au serveur Mysql. Coder le fichier **"formulaire_mission.html"**, qui contiendra le formulaire html de saisie d'une mission, et ensuite nous nous occuperons du fichier **"envoi_mission.php"**, qui sera exécuté lorsqu'on appuiera sur le bouton "Envoi !" du formulaire.

- **formulaire_mission.html** : ce formulaire doit proposer tous les champs de texte nécessaires : idmissions, mission, lieu et directeur. Pour permettre de lancer la page "envoi_mission.php", on place un bouton de type submit. Le formulaire enverra les données vers "envoi_mission.php" par la méthode "POST". Attention, mettez les bons "name" pour vos champs de texte, c'est grâce à ces noms que nous auront la possibilité de récupérer les données.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/HTML; charset=iso-8859-1" />
<title>formulaire_mission.html</title>
</head>
<body>
<h1>Formulaire de saisie d'une mission</h1>
<form method="post" action="envoi_mission.php"/>
<fieldset><legend>Ajout Mission</legend>
<p> Numero d'identification : <input type="text" name="idmissions" size="7"/></p>
<p> Mission : <input type="text" name="mission" size="27"/></p>
<p> Lieu : <input type="text" name="lieu" size="30"/></p>
<p> Directeur : <input type="text" name="directeur" size="27"/></p>
<p><input type="submit" value="envoi!"/></p>
</fieldset>
</form>
</body>
</html>
```

- **envoi_mission.php** : ce script récupère les informations saisies dans le formulaire, les affiche, se connecte à la base de données et insère les données dans la base.

Vous pouvez mettre un lien dans le script envoi_mission.php afin de retourner directement au formulaire afin de faciliter la saisie d'une nouvelle mission.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/HTML; charset=iso-8859-1" />
<title>envoi_mission.php</title>
</head>
<body>
<?php
//Recuperation des donnees
$idmissions=$_POST["idmissions"];
$mission=$_POST["mission"];
$lieu=$_POST["lieu"];
$directeur=$_POST["directeur"];
echo "<p>Recapitulatif des donnees saisies </p>";
echo "<ul>";
echo "<li> Identification mission : ".$idmissions."</li>";
echo "<li> Mission : ".$mission."</li>";
echo "<li> Lieu : ".$lieu."</li>";
echo "<li> Directeur : ".$directeur."</li>";
echo "</ul>";
//Connexion a la base
include("connect.php");
//Requete d'ajout a la table "missions"
$requete="INSERT INTO 'missions' ( 'idmissions' , 'mission', 'lieu' , 'directeur')
VALUES (".$idmissions", ".$mission", ".$lieu", ".$directeur");";
mysql_query($requete);
echo "<p>Donnees ajoutees...</p>";
echo "<p>Cliquez <a href='formulaire_mission.html'>ici</a> pour ajouter une autre
mission.</p>";
?>
</body>
</html>

```

5. Récupérer les données d'une table via une page web.

- Écrire les scripts php respectivement appelés **affiche_ingenieurs.php** et **affiche_missions.php** qui permettent d'afficher dans une page html sous forme de tableaux le contenu des tables correspondantes.

```

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/HTML; charset=iso-8859-1" />
<title>Affichage de la table ingénieur</title></head>
<body>
<?php
//Connexion a la base
include("connect.php");
//Creation et envoie de la requete
$requete="SELECT idingenieurs, nom, prenom, agence, telephone FROM ingenieurs";
$resultat=mysql_query($requete);
echo "<table border=1>";
echo "<tr bgcolor = #ffffaa>";
<th>Id</th>

```



```

<th>Nom</th>
<th>Prénom</th>
<th>Agence</th>
<th>Tél</th>
</tr>";
//Recuperation des resultats
while ($ligne=mysql_fetch_row($resultat)){
$idingenieurs=$ligne[0];
$nom=$ligne[1];
$prenom=$ligne[2];
$agence=$ligne[3];
$telephone=$ligne[4];
echo "<tr>
<td> $idingenieurs </td>
<td> $nom </td>
<td> $prenom </td>
<td> $agence </td>
<td> $telephone </td>
</tr>";
}
?>
</body>
</html>

```