PHP et mySQL

Code: php_mysql

Originaux

url: **http://tecfa.unige.ch/guides/tie/html/php-mysql/php-mysql.html**
url: **http://tecfa.unige.ch/guides/tie/pdf/files/php-mysql.pdf**

Auteurs et version

- Olivier Clavel - Daniel K. Schneider - Patrick Jermann - Vivian Synteta

- Version: 0.10 (modifié le 15/8/01 par DKS)

Prérequis

Module technique précédent: php-html
Module technique précédent: mysql-intro

Modules

Module technique suppl.: java-mysql

© TECFA 15/8/01

Objectifs

- PHP - MySQL basics :)

© TECFA 15/8/01

1. Table de matières détaillée

1. Table de matières détaillée 3
2. Principe de la connectivité PHP - bases de données 4
2.1 Un exemple complet documenté: 4
2.2 Fonctions PHP - MySQL 6
A.Se connecter à un serveur de bases de données 6
B.Sélectionner une base de données 6
C.Exécuter une requête SQL 7
D.Traitement des résultats 8
3. Une petite application PHP - mySQL : Le livre d'or. 13
3.1 Détails des fichiers et des ressources 13
3.2 Structure de la table comments 14
3.3 Détails de l'écriture des données 15
3.4 Détails de l'affichage de la table 17

2. Principe de la connectivité PHP - bases de données

- PHP permet d'interagir avec une base de données par l'intermédiaire de fonctions.
- Nous construisons les requêtes en écrivant un programme PHP.
- Nous affichons les résultats des requêtes avec HTML.

Fonctions PHP-mySQL

url: **http://tecfa.unige.ch/guides/php/php3/manual/ref.mysql.html**

2.1 Un exemple complet documenté:

url: **http://tecfa.unige.ch/guides/php/examples/mysql-demo/main.html**

url: **Les détails: http://tecfa.unige.ch/guides/php/examples/mysql-demo/**

- afficher un ensemble d'enregistrements (**dump_results.phps**)
- afficher un seul enregistrement
- ajouter un enregistrement (**new-entry.phps** et **insert-entry.phps**)
- éditer un enregistrement (**edit-entry.phps** et **replace-entry.phps**)
- effacer un enregistrement (**delete-entry.phps**)

Pour l'ajout d'enregistrements deux scripts sont nécessaires:

- new-entry.php produit un formulaire HTML vide

- insert-entry.php prend le contenu du formulaire, essaye de l'insérer dans la base de données et donne un feed-back à l'utilisateur.

Il en va de même pour l'édition d'un enregistrement:

- edit-entry.php produit un formulaire HTML contenant les valeurs précédemment enregistrées
- replace-entry.php tente de remplacer les anciennes valeurs avec celles que l'utilisateur a entré dans le formulaire et donne un feed-back à l'utilisateur.

2.2 Fonctions PHP - MySQL

A. Se connecter à un serveur de bases de données

- Avant de pouvoir accéder à une base de données, il faut établir une connexion avec le serveur qui l'héberge. On spécifie:
  - le nom de la machine sur laquelle est installé le serveur (host)
  - un nom d'utilisateur
  - le mot de passe correspondant

```
Syntaxe: mysql_pconnect
mysql_pconnect(host, username, password);
mysql_pconnect("tecfasun1.unige.ch","nobody","");
```

- 'nobody' est un utilisateur qui peut se connecter sans mot de passe mais qui a des permissions limitées.

B. Sélectionner une base de données

- Un serveur héberge plusieurs bases de données qui chacune contiennent des tables.

```
Syntaxe: mysql_select_db
mysql_select_db(dbname);
mysql_select_db("demo");
```

C. Exécuter une requête SQL

- PHP permet d'envoyer n'importe quelle requête SQL au serveur en utilisant la commande mysql_query.
- Le nom d'utilisateur et le mot de passe spécifiés en se connectant à la base de données déterminent si la requête peut être exécutée.
- La commande retourne un identificateur de résultat que l'on stocke dans une variable pour l'utiliser par la suite ($result).

```
Syntaxe: mysql_query
mysql_query(requête_SQL);
$result = mysql_query("SELECT * FROM demo1");
```

La variable `$result` est "boolean" et contient à ce moment

- 1 (TRUE) si l'opération a été effectuée
- 0 (FALSE) si il y a eu un problème.

qui est très utile pour le déboguage!

D. Traitement des résultats

- Après avoir soumis une requête à mySQL, la fonction mysql_query nous donne un identificateur de résultats ($result) qu'il faut décortiquer et afficher.
- Il existe de multiples façons d'accéder au résultat d'une requête. En voici deux.

Traitement indépendant du nom des champs

```
Syntaxe: mysql_fetch_row
mysql_fetch_row(identificateur)
$row = mysql_fetch_row($result);
```

- prend un enregistrement dans le résultats. Cet enregistrement est un vecteur de valeurs qui correspond aux champs de la base de données.

```
Syntaxe: mysql_num_fields
mysql_num_fields(identificateur)
$nb_champs = mysql_num_fields($result);
```

- donne le nombre de champs dans un enregistrement.

E. Gérer les erreurs (Warnings)

- Après avoir fait une opération mySQL, PHP peut nous rétourner des "warnings" pour nous prévenir d'une erreur.
- Il existe deux fonctions PHP à accéder aux erreurs:

`Syntaxe: mysql_errno()`

`mysql_errno(`*`identificateur`*`)`

- rétourne la valeur numérique d'erreur de la dernière opération mySQL ou zéro (0) s'il n'y pas des erreurs.

`Syntaxe: mysql_error()`

`mysql_error(`*`identificateur`*`)`

- rétourne la description d'erreur de la dernière opération mySQL ou un "string" vide "" dans le cas ou il n'y a pas des erreurs.

```
Exemple avec les deux fonctions:
<?php
 mysql_connect("marliesle");
 echo mysql_errno().": ".mysql_error()."<BR>";
 mysql_select_db("nonexistentdb");
 echo mysql_errno().": ".mysql_error()."<BR>";
 $conn = mysql_query("SELECT * FROM nonexistenttable");
 echo mysql_errno().": ".mysql_error()."<BR>";
?>
```


Exemple 2-1: Traitement indépendant du nom des champs. Génération d'une table HTML avec les enregistrements contenus dans un résultat

url: **http://tecfa.unige.ch/guides/php/examples/mysql-demo/dump_results_demo.php**
url: **http://tecfa.unige.ch/guides/php/examples/mysql-demo/dump_results_demo.phps**
url: **http://tecfa.unige.ch/guides/php/examples/mysql-demo/dump_results_demo.source**

```
<?
mysql_pconnect("tecfasun1.unige.ch","nobody","");
mysql_select_db("demo");
$result = mysql_query("SELECT * FROM demo1");
?>
<table  border="1">   <tr>
<?
while  ($row  =  mysql_fetch_row($result))  {
    echo  "<tr>";
    for  ($i=0;  $i<mysql_num_fields($result);  $i++)  {
      echo  "<td>";
      // test if this is the URL
      if ($i == 4) { echo  "<a href='$row[$i]'>$row[$i]</a>";  }
      else {  echo  "$row[$i]"; }
      echo  "</td>";
    } }
?>
</table>
```

Traitement en utilisant le nom des champs

mysql_num_rows

```
Syntaxe: mysql_num_rows(identificateur);
$nb_enregistrements = mysql_num_rows($result);
```

• Donne le nombre d'enregistrements contenus dans le résultat identifié par $result

mysql_result

```
Syntaxe: mysql_result(identificateur, index, champ);
$nom = mysql_result($result,0,'fullname');
```

• `index` désigne le numéro de l'enregistrement. L'indexation commence à 0 ! (zéro). A l'index 0 correspond le premier enregistrement.
• `champ` désigne le nom du champ que l'on veut récupérer.

Exemple 2-2: Traitement utilisant le nom des champs. Génération d'une table HTML avec les enregistrements contenus dans un résultat

url: **http://tecfa.unige.ch/guides/php/examples/mysql-demo/dump_results_demo2.php**
url: **http://tecfa.unige.ch/guides/php/examples/mysql-demo/dump_results_demo2.phps**
url: **http://tecfa.unige.ch/guides/php/examples/mysql-demo/dump_results_demo2.source**

```
<?
mysql_pconnect( "localhost", "nobody", "") or die( "Unable to connect to
SQL server");
mysql_select_db("demo") or die ( "Unable to select database");
$result = mysql_query( "select * from demo1");
?>
  <table border="1">
<?
$i = 0;
while ($i < mysql_num_rows($result)) {
  echo "<tr>";
  echo "<td>";
  echo mysql_result($result,$i,'id');
  echo "</td>";
  echo "<td>";
  echo mysql_result($result,$i,'fullname');
  echo "</td>";
  echo "<td>";
  echo mysql_result($result,$i,'love');
```

```
  echo  "</td>";
  echo  "<td>";
  echo mysql_result($result,$i,'sports');
  echo  "</td>";
  echo  "</tr>";
  $i++;
}
  echo  "</table>";
?>
```


3. Une petite application PHP - mySQL : Le livre d'or.

url: http://tecfa.unige.ch/guides/tie/code/act-webm/comments.html

3.1 Détails des fichiers et des ressources

Cette petite application se compose de 3 fichiers qui ont les fonctions suivantes :

- *comments.html* affiche le formulaire et envoi les données à *comments-insert.php*
- *comments-insert.php* écrit les données du formulaire dans la base de donnée mySQL

```
Serveur : tecfasun5.unige.ch
base de donnée : mydb
table : comments
utilisateur : nobody
```

- *comments-list.php* affiche tous les enregistrements de la table *comments* dans un tableau.

3.2 Structure de la table comments

url: http://tecfa.unige.ch/guides/tie/code/act-webm/comments-table.txt

```
create table comments (
    id int(10) default '0' not null auto_increment,
    nom char(20) default '' not null,
    prenom char(20) default '',
    email char(50) default '' ,
    computer char(10),
    browser char(10),
    version char(10),
    comments char(200),
    primary key (id),
    key nom (nom)
);
```


3.3 Détails de l'écriture des données

url: http://tecfa.unige.ch/guides/tie/code/act-webm/comments-insert.phps

On commence par ce connecter à la base de donnée

```
$link = mysql_connect( "tecfasun5", "nobody", "")
or die ( "Unable to connect to SQL server");

mysql_select_db("mydb", $link)
or die ( "Unable to select database");
```

Ensuite, on construit la requête SQL à partir des données envoyées par le formulaire et on envoi la requête à mySQL

```
$query_string = "INSERT INTO comments (nom, prenom, email, computer,
browser, version, comments) VALUES ('$nom', '$prenom', '$email',
'$computer', '$browser', '$version', '$comments')";

$result = mysql_query ($query_string);
```

A ce stade, la variable `$result` contient

- 1 (TRUE) si l'opération a été effectuée
- 0 (FALSE) si il y a eu un problème.

Il faut faire un test sur cette variable pour donner un feedback à l'utilisateur

```
if ($result) {
    echo "<p>Vos données son bien enregistrées.";
      echo "<p>Vous pouvez aller voir tous les <a href='comments-
list.php'>commentaires</a>.\n";
  }
  else {
    echo "<p>Warning: ERROR writing on data base.</p>\n";
    echo "<p>Error returned by mySQL : " . mysql_error() . "\n";
  }
```

La fonction `mysql_error()` retourne la dernière erreur mySQL générée dans un string qu'on peut imprimer. Très utilile pour debugger !

En cas de problème, il est conseillé d'imprimer à l'écran la requête SQL pour pouvoir la vérifier

```
echo $query_string;
```

Dans de grosses applications avec beaucoup de données, il faut penser a libérer les ressources de la machine en fermant les connections ouvertes

```
mysql_close($link);
```


3.4 Détails de l'affichage de la table

url: http://tecfa.unige.ch/guides/tie/code/act-webm/comments-list.phps

Après avoir établi la connection (voir paragraphe suivant) il faut extraire les données de la table *comments*

```
$result = mysql_query( "select * from comments");
```

A ce stade, la variable $result contient :

- 0(false) si l'opération a posé un problème
- l'identificateur d'un "result set" si l'opération s'est déroulée correctement

Ce n'est pas fait dans cet exemple mais il est préférable de faire un test sur cette variable avant la suite des opérations.

```
if (!$result) {
    echo "ca marche pas :(";
    ...arreter le programme
} else {
    ...traiter les données
}
```

Pour présenter les données à l'écran, on les présente dans un tableau. Les lignes et les cellules sont crées dans une boucle qui va lire chaque enregistrement du "result set".

```
$i = 0;

while ($i < mysql_num_rows($result)) {
  echo  "<tr>";

  echo  "<td>";
  echo mysql_result($result,$i,'id');
  echo  "</td>";

  echo  "<td>";
  echo mysql_result($result,$i,'nom');
  echo  "</td>";

  echo  "<td>";
  echo mysql_result($result,$i,'prenom');
  echo  "</td>";

  ..... suite du traitement des champs

  echo "</tr>";

  $i++;
}
```