

# PHP et MySQL

Code: act-php-mysql

## Originaux

[url: http://tecfa.unige.ch/guides/tie/html/act-php-mysql/act-php-mysql-FC.html](http://tecfa.unige.ch/guides/tie/html/act-php-mysql/act-php-mysql-FC.html)

[url: http://tecfa.unige.ch/guides/tie/pdf/files/act-php-mysql-FC.pdf](http://tecfa.unige.ch/guides/tie/pdf/files/act-php-mysql-FC.pdf)

## Auteurs et version

- [Olivier Clavel](#) - [Daniel K. Schneider](#) - [Patrick Jermann](#) - [Vivian Synteta](#)
- Version: 2.2 (modifié le 8/3/07 par DKS)

## Prérequis

*Module technique précédent:* [php-html](#)

*Module technique précédent:* [mysql-intro](#)

## Modules

*Module technique suppl.:* [java-mysql](#)

## Objectifs

- Comprendre les mécanismes de base de la connectivité PHP <-> MySQL.
- Connaitre les fonctions principales disponibles dans PHP pour communiquer avec un serveur de bases de données MySQL.
- Créer une petite application PHP/MySQL à partir d'un exemple (livre d'or).

## A faire

- PHP-MySQLi functions
- Parler de bibliothèques comme ADOdb

# 1. Table de matières détaillée

1. Table de matières détaillée .....	3
2. Principe de la connectivité PHP - bases de données.....	4
<b>2.1 Un exemple complet documenté:</b>	<b>4</b>
<b>2.2 Fonctions PHP - MySQL</b>	<b>6</b>
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
E.Gérer les erreurs (Warnings)	9
Exemple 2-1: Exemple gestion d'erreurs:	9
Exemple 2-2: Traitement indépendant du nom des champs.	
Génération d'une table HTML avec les enregistrements contenus dans un résultat	10
F.Traitement en utilisant le nom des champs	11
Exemple 2-3: Traitement utilisant le nom des champs.	
Génération d'une table HTML avec les enregistrements contenus dans un résultat	12
3. Une petite application PHP - mySQL : Le livre d'or.....	14
<b>3.1 Récupération des fichiers exemples</b>	<b>14</b>
<b>3.2 Structure de la table comments</b>	<b>15</b>
<b>3.3 Détails des fichiers.</b>	<b>16</b>

## 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 (username) + le mot de passe correspondant (password)

Syntaxe: `mysql_pconnect(host, username, password);`

```
$link = mysql_pconnect("localhost", "clavel", "secret");
```

- On utilise 'localhost' pour signifier qu'on veut se connecter au serveur local (là où se trouve le serveur Web qui exécute la page). Si le serveur se trouve ailleurs, il faut donner l'adresse complète. \$link contient la référence de la connexion ouverte (ou "FALSE" si la connexion ne s'est pas faite).

### B. Sélectionner une base de données

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

Syntaxe: `mysql_select_db(dbname, [linkID]);`

```
mysql_select_db("demo");
```

- Par défaut, la requête est envoyée sur le dernier lien ouvert. Si on a des connexions vers plusieurs serveurs, il faut le spécifier.

## 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(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.

Ce qui est très utile pour le débogage !

## 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)`

Exemple:

```
$result = mysql_query("SELECT * FROM demo1")
```

```
$row = mysql_fetch_row($result);
```

- prend un enregistrement dans le résultats.
- Cet enregistrement est un tableau (array simple) de valeurs qui correspond aux champs de la base de données.
- Au prochain appel cette instruction retourne la ligne suivante (voir exemple 2-2 "Traitement indépendant du nom des champs. Génération d'une table HTML avec les enregistrements contenus dans un résultat" [10])

### ***Nombre de champs dans un enregistrement***

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 retourner des "warnings" pour nous prévenir d'une erreur.
- Il existe deux fonctions PHP pour accéder aux erreurs:  
Syntaxe: `mysql_errno()`  
`mysql_errno(identificateur)`
- retourne 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)`
- retourne 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 2-1: Exemple gestion d'erreurs:

```
<?php
mysql_pconnect("nohost", "basuser", "wrongpass");
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 2-2: 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](http://tecfa.unige.ch/guides/php/examples/mysql-demo/dump_results_demo.php)

[url: /guides/php/examples/mysql-demo/dump\\_results\\_demo.phps](/guides/php/examples/mysql-demo/dump_results_demo.phps)

[url: /guides/php/examples/mysql-demo/dump\\_results\\_demo.source](/guides/php/examples/mysql-demo/dump_results_demo.source)

```
<?
mysql_pconnect("localhost","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>
```

## F. Traitement en utilisant le nom des champs

### *mysql\_num\_rows*

Syntaxe: `mysql_num_rows(identificateur);`

Exemple:

```
$nb_enregistrements = mysql_num_rows($result);
```

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

### *mysql\_result*

Syntaxe: `mysql_result(identificateur, index, champ);`

Exemple:

```
$nom = mysql_result($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.
- (voir slide suivant)

## Exemple 2-3: 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](http://tecfa.unige.ch/guides/php/examples/mysql-demo/dump_results_demo2.php)

[url: /guides/php/examples/mysql-demo/dump\\_results\\_demo2.phps](/guides/php/examples/mysql-demo/dump_results_demo2.phps)

[url: /guides/php/examples/mysql-demo/dump\\_results\\_demo2.source](/guides/php/examples/mysql-demo/dump_results_demo2.source)

```
<?php
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">
<?php
$i = 0;
while ($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.

### 3.1 Récupération des fichiers exemples

- Commencez par créer un sous répertoire `comments` dans un répertoire accessible au serveur web.
- Copiez les fichiers de l'exemple: `comments-list.source`, `comments-insert.source`, `comments.html`  
[url: http://tecfa.unige.ch/guides/tie/code/act-php-mysql-FC/](http://tecfa.unige.ch/guides/tie/code/act-php-mysql-FC/)
- Si vous avez un accès FTP ou local prenez les fichiers `*.source` et `*.html` et copiez les fichiers vers votre répertoire "comments".
- Sinon vous pouvez prendre les fichiers avec le navigateur web (click droit et "save as").

**Attention 1** : vous devez changer l'extension des fichiers `.source` en `.php` !!!

**Attention 2** : si vous utilisez IE, vous devez choisir le type "tout fichiers/all files" avant de sauver (sinon le fichier portera encore l'extension `.html`)

## 3.2 Structure de la table comments

[url: http://tecfa.unige.ch/guides/tie/code/act-php-mysql-FC/solution/comments-table.txt](http://tecfa.unige.ch/guides/tie/code/act-php-mysql-FC/solution/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)  
);
```

Utilisez cette définition pour créer votre table dans votre base de données sur [tecfa.unige.ch](http://tecfa.unige.ch) avec l'interface phpMyAdmin.

[url: http://tecfa.unige.ch/admin/phpMyAdmin/](http://tecfa.unige.ch/admin/phpMyAdmin/)

Vous pouvez soit envoyer directement la requête ci-dessus pour créer la table d'un seul coup, soit créer la table pas à pas en utilisant l'interface graphique.

### 3.3 Détails des fichiers.

Vous devez donc maintenant disposer de 3 fichiers dans votre répertoire. Ils composent une petite application et ils ont les fonctions suivantes :

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

Serveur : localhost (tecfa.unige.ch)

base de données : <votre\_login>

table : comments (ou celui que vous avez choisi)

utilisateur : <votre\_login>

mot de passe : <votre mot de passe>

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

**Attention** : le formulaire HTML est directement utilisable mais vous devrez modifier les 2 fichiers php pour que ça fonctionne. C'est la toute l'objectif de l'exercice. Tous les détails pour vous aider se trouvent dans les commentaires directement dans le fichier.

Commencez par le fichier `comments-insert.php` puis ensuite le fichier `comments-list.php`

**Attention** : pour tester, vous devez envoyer les fichiers sur le serveur avec FTP.

Pour vous faire une idée du résultat à obtenir, vous pouvez tester un exemple complet qui fonctionne ici :

[url: http://tecfa.unige.ch/guides/tie/code/act-php-mysql-FC/solution/comments.html](http://tecfa.unige.ch/guides/tie/code/act-php-mysql-FC/solution/comments.html)

.... Bon courage !! On est là pour vous aider :))