



Interfaçage avec une base de données

■ Principe

- PHP propose de nombreux outils permettant de **travailler** avec la plupart des SGBD
 - Oracle, Sybase, Microsoft SQL Server, PostgreSQL ou encore MySQL
- Php fournit un grand choix de fonctions permettant de **manipuler** les bases de données.
 - Quatre fonctions sont essentielles:
 - La fonction de **connexion** au serveur
 - La fonction de **choix de la base de données**
 - La fonction de **requête**
 - La fonction de **déconnexion**



Interfaçage avec une base de données

Les fonctions principales de manipulation pour mysql sont :

- La fonction de connexion au serveur :

mysql_connect()

mysql_pconnect()

- La fonction de choix de la base de données :

mysql_select_db()

- La fonction de requête :

mysql_query()

- La fonction de déconnexion :

mysql_close()



La connexion au serveur I

- Les variables qui vont permettre la connexion au serveur (Attention, le nom des variables est bien sûr libre)
 - **\$host** : l'hôte (ordinateur sur lequel le SGBD est installé)
 - **\$user** : le nom d'utilisateur de la BdD
 - **\$passwd** : le mot de passe de cet utilisateur

Exemple :

```
$host="localhost"
```

```
$user="admin"
```

```
$passwd="*****"
```



La connexion au serveur II

`$id_connexion = mysql_connect($host, $user, $passwd)`

- retourne une valeur de type integer (**`$id_connexion`**) servant d'**identificateur de connexion** lorsque la connexion a été établie
- **retourne false si une erreur** se produit lors de l'établissement de la connexion

Si **`$id_connexion`** est différent de false, la connexion est établie. Il peut être utile de bien tester cette connexion. Cela permettra de gérer proprement l'éventuelle indisponibilité du serveur de BdD.



La fonction die()

Termine le script courant

die(chaine)

die() termine l'exécution du script courant.

Elle n'a pas de valeur de retour mais elle utilisera chaine comme message de fin d'exécution. (emprunt à PERL)

Cette fonction est souvent utilisée pour les connexions aux bases de données, mais n'est pas du tout obligatoire.



Le script de connexion au serveur

```
<?php
$id_connexion = mysql_connect($host, $user, $passwd)
    or die("Impossible de se connecter");
echo "Connexion réussie";
?>
```

`$id_connexion` contient une référence à cette connexion sur la base. Cet identifiant pourra être présenté pour chaque commande mysql qu'on effectuera ensuite.

Cela peut être utile si votre programme se connecte à plusieurs bases de données différentes, ou à plusieurs serveurs différents. Cependant, on peut omettre sa gestion si on utilise une seule base, sur une seule machine.



Accès à une base de données

Suite à la connexion à un SGBD, il faut

- soit sélectionner la base de données si elle existe,
- soit la créer si elle n'existe pas.

- Sélection de la bases de données \$bdd :

```
mysql_select_db($bdd)
```

```
mysql_select_db($bdd,$id)
```

- Création de la bases de données \$bdd :

```
mysql_create_db ($bdd)
```

```
mysql_create_db ($bdd,$id)
```



Comment se connecter à une base

```
<?php
```

```
// Connexion au serveur
```

```
$id_connexion = mysql_connect($host, $user, $passwd)  
    or die("Impossible de se connecter");
```

```
// sélection de la base
```

```
mysql_select_db($bdd,$id_connexion)  
    or die("Connection à la base impossible");
```

```
?>
```

Ce fichier peut être inclus dans un script exploitant la base \$bdd à l'aide de **include()**

Avec cette procédure, le mot de passe est écrit "en dur" dans le fichier.
(mais en php, on ne voit pas le contenu à distance)

L'emploi d'un formulaire (avec la méthode POST) dans lequel on demande le login et le mot de passe évite de l'avoir écrit dans un fichier.



Comment se déconnecter

```
// Fermeture de la connexion
```

```
mysql_close( $id_connexion );
```

Cette commande sert finalement assez peu. En effet, si vous avez ouvert une connexion avec `mysql_connect()`, celle-ci est fermée automatiquement en fin de script.

Et si vous avez choisi une connexion persistante, il ne faut pas la fermer, sauf éventuellement en fin de session.



Les requêtes SQL

- Les requêtes doivent répondre à la syntaxe SQL en général
- Les requêtes SQL sont envoyées à la base de données par :

mysql_query()



Comment exécuter une requête

```
// Exécuter des requêtes SQL
$query = "SELECT * FROM my_table";
$result = mysql_query($query)
    or die("requête non conforme");
$query2= "INSERT INTO etud
    VALUES ("bob", "l'éponge", 2001/04/02)";
$result = mysql_query($query2, $id)
or die("requête non conforme");
```



Traitement des résultats d'une requête

\$result contient l'ensemble des enregistrements retournés par la requête. On dit que c'est un RecordSet.

- On utilise la fonction **mysql_fetch_row()**, qui découpe les lignes de résultats en colonnes (par exemple Nom,adresse,...) et les affecte à une **variable tableau** dans l'ordre où elles arrivent (exemple I).
- On utilise la fonction **mysql_fetch_assoc()** qui découpe les lignes de résultats en colonnes (par exemple Nom,adresse,...) et les affecte à une *variable tableau associatif* (exemple II).
- Exemple: une table appelée liens contenant le nom et l'URL de sites internet.
 - récupérer l'ensemble des enregistrements et de les afficher dans un tableau



C'est si simple ?

Mysql_query() renvoie un **RecordSet**.

C'est un jeu d'enregistrements

On peut se **déplacer** à l'intérieur, pour récupérer chaque tuple un par un (on dit qu'on fait un **fetch**, ce qui pourrait se traduire par « au suivant »)

Le RecordSet offre des **fonctions**, telles que *nombre de lignes*, *nombre de colonnes*, *listage des noms des colonnes*...



Exemple I

```
<html><head><title>Liens</title></head>
<body>
<table>
  <tr>
    <th>Nom du site</th>
    <th>URL</th>
  </tr>
<?php
// Déclaration des paramètres de connexion
$host = "la_machine";
$user = "votre_login";
$dbdd = "Nom_de_la_base_de_donnees";
$password = "Mot_de_passe";
// Connexion au serveur
mysql_connect($host, $user, $password) or die("erreur de connexion au serveur");
mysql_select_db($dbdd) or die("erreur de connexion a la base de donnees");
```



Exemple I suite

```
// Creation et envoi de la requete
$query = "SELECT nom,url FROM liens ORDER BY nom";
$result = mysql_query($query);
// Recuperation des resultats
while($row = mysql_fetch_row($result)){ //ici, on fetch un tableau indicé
    $nom = $row[0];
    $url = $row[1];
    echo "<tr>\n
        <td><a href=\"\$url\">$nom</a></td>\n
        <td>$url</td>\n
    </tr>\n";
}
// Déconnexion de la base de données
mysql_close();
?>
    </tr>
</table>
</body></html>
```



Exemple II

```
// Afficher des résultats en HTML
echo "<table>\n";
while ($ligne = mysql_fetch_assoc($result)) { //fetch en tableau
    associatif
    echo "\t<tr>\n";
    foreach ($ligne as $clef=>$val_col) {
        echo "\t\t<td>la $clef vaut $val_col</td>\n";
    }
    echo "\t</tr>\n";
}
echo "</table>\n";s
```



Interfaçage avec une base de données

- L'exploitation des données
 - Le décompte total des enregistrements ou des champs d'une table, peut être obtenu par l'intermédiaire de certaines fonctions applicables sur le recordset.

```
$nb_lignes = mysql_num_rows($id_resultat);
```

```
$nb_champs = mysql_num_fields($id_resultat);
```



Interfaçage avec une base de données

- L'exploitation des données

- A partir des enregistrements, chacun des champs devient **accessible** par des fonctions appropriées.
- Certaines fonctions sont capables **d'extraire** directement un champ déterminé

```
$valeur = mysql_result($id_resultat, $num_ligne, $num_col);
```



Fonctions mysql_

- mysql_close** -- Ferme la connexion MySQL
- mysql_connect** -- Ouvre une connexion à un serveur MySQL
- mysql_errno** -- Retourne le numéro d'erreur de la dernière commande MySQL
- mysql_error** -- Retourne le texte associé avec l'erreur générée lors de la dernière requête
- mysql_fetch_array** -- Retourne une ligne de résultat MySQL sous la forme d'un tableau associatif, d'un tableau indexé, ou les deux
- mysql_fetch_assoc** -- Lit une ligne de résultat MySQL dans un tableau associatif
- mysql_fetch_row** -- Retourne une ligne de résultat MySQL sous la forme d'un tableau
- mysql_num_fields** -- Retourne le nombre de champs d'un résultat MySQL
- mysql_pconnect** -- Ouvre une connexion persistante à un serveur MySQL
- mysql_query** -- Envoi une requête à un serveur MySQL
- mysql_select_db** -- Sélectionne une base de données MySQL

■ <http://www.nexen.net/docs/php/annotee/mysql.intro.php>

Construction d'une fonction générique

Comment écrire une fonction générique php/
mysql ?

- Utilisation des commandes liées au recordset
- Quelques contraintes au niveau de la base de données
- On veut un tableau html visualisant les tuples d'une table, de façon générique, automatique, avec possibilité de modification et de suppression.



Construction générique

Utilisation possible : Caractère jocker de la projection en SQL : *

En Php, la fonction **mysql_field_name(\$recordset, NumColonne)** pourra retrouver le nom de l'attribut : Pratique pour créer les entêtes de colonnes.

Pour avoir le nombre de colonnes, **mysql_num_field(\$recordset)...**



Construction entête du tableau

Démarche : on fait un select *.

On interroge le recordset pour connaître le nombre de colonnes du résultat, et on interroge 1 à 1 le nom de cette colonne.

En PHP, on fabrique une table html, et on définit une ligne d'entête grace aux balises de tableau (`<thead><tr> <th>`)



Et on obtient...

```
function Entete($recordset) {  
    $nbchamps=mysql_num_fields($recordset)  
    or die("ce n'est pas une demande valide !");  
echo "<thead><tr>\n";  
for ($i=0;$i<$nbchamps;$i++) { // $i parcourt les attributs  
    echo "\t<th>";  
    echo mysql_field_name($recordset,$i);  
    echo "</th>\n";  
}  
echo "</tr></thead>\n";  
}
```



Le corps maintenant

Même raisonnement pour le corps du tableau, on va parcourir toutes les réponses contenues dans le recordset, et les ranger dans l'ordre des colonnes, afin de bien mettre en correspondance avec le titre de la colonne.



Le corps....

```
function Corps($recordset,) {  
    $nbchamps=mysql_num_fields($recordset);  
    $nbenr=mysql_num_rows($recordset);  
    echo "<tbody>\n";  
    for ($i=0;$i<$nbenr;$i++) {//Pour chaque ligne  
        echo "\t<tr>\n";$ligne=mysql_fetch_array($recordset);  
        for ($j=0;$j<$nbchamps;$j++) { //on affiche chaque champ  
            echo "\t\t<td>$ligne[$j]</td>\n";  
        }  
        echo "\t</tr>\n"; //fin de la ligne  
    }  
    echo "<tbody>\n";  
}
```

Appel de l'ensemble

```
function TableComplete($Question) {  
    $result=mysql_query($Question);  
    echo "<table>\n";  
    Entete($result);  
    Corps($result,);  
    echo "</table>\n";  
}
```

```
//exemple d'appel  
<p>Liste des étudiants</p>  
<?php  
TableComplete(« SELECT *  
FROM etudiants »);  
?>
```



Autres actions...

Libération de la mémoire : Si dans un script vous faites beaucoup de requêtes dont les résultats sont imposants, vous pouvez gérer la mémoire :

mysql_free_result();

Attention, en général, tout est libéré à la fin du script (même la connexion à la database est perdue)



Autres actions

mysql_data_seek(\$recordSet,\$row)

déplace le pointeur interne de résultat, dans le résultat associé à l'identifiant de résultat *\$recordSet* . Il le fait pointer à la ligne *\$row_number* .

Le prochain appel à une fonction MySQL de récupération de données, comme la fonction **mysql_fetch_assoc()** retournera cette ligne.

mysql_insert_id() —

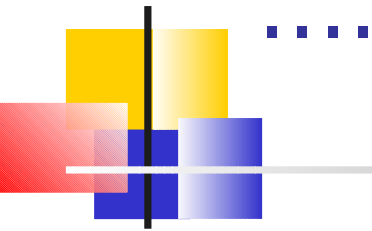
Retourne l'identifiant généré par la dernière requête INSERT MySQL.

Cela peut être pratique quand vous voulez connaître la valeur de la clé primaire qui a été affectée à l'enregistrement.



Quelques exemples

```
$insert = mysql_query("insert into archives
values(\{insertid}\",\{lestag[civilite]\"");
if (!$insert) {
    $message = mysql_error() . " Creation ARchives";
    $noProblem = false;
} //PEPIN
$numArchive = mysql_insert_id();
mysql_query("insert into archivesEDT select
$numArchive,Date,HeureDebut,NbHeures from presences,
sequences,profs where presences.NumSequence=
sequences.NumSequence and profs.NumProf=
sequences.NumProf and presences.NumStagiaire=
{lestag[NumStagiaire]");
```



//et on attaque le nettoyage..

```
$net0 = mysql_query("delete from prescriptions  
where NumStagiaire=$lestag[NumStagiaire]");
```

```
if (!$net0) {
```

```
    $message = mysql_error() . " PRESCRIPTIONS";
```

```
    $noProblem = false;
```

```
} //PEPIN
```