

Mysql avec EasyPhp

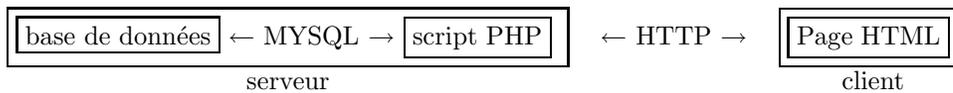
1^{er} mars 2006

Introduction

MYSQL dérive directement de **SQL** (Structured Query Language) qui est un langage de requêtes vers les bases de données relationnelles.

Il en reprend la syntaxe mais n'en a pas toute la puissance.

Le serveur de base de données **MySQL** est très souvent utilisé avec le langage de création de pages web dynamiques : **PHP**.



Les données manipulées par le site seront stockées dans une base de données **MySQL**. Le script **PHP**, se servira de commandes **MySQL** pour gérer et extraire des informations de la base de données afin d'engendrer les pages **HTML** qui seront interprétées par le navigateur.

Nous allons dans ce Tp, utiliser **EasyPHP** qui est une application permettant d'installer et de configurer automatiquement un environnement de travail complet sous Windows en regroupant un serveur Apache, un serveur **MySQL**, une interface graphique de gestion des bases de données **MySQL** : **PHPMyAdmin** et le langage **PHP**. On suppose donc que **EasyPHP** est installé sur votre machine.

Ces notes sont un résumé succinct des articles : utiliser **MySQL** et **PHP** de Frédéric Bouchery : http://www.phpinfo.net/articles/article_phpmysql.html et du tutoriel d'introduction de la doc **mysql** sur <http://www.nexen.net>.

Les liens vers les documentations **MySQL** et **PHP** se trouvent sur le site du cours.

1 MySQL

Pour pouvoir utiliser un serveur **MySQL**, il faut qu'il soit lancé. Cela se fait dans notre cas en lançant **EasyPHP** qui n'a d'autre effet que de démarrer les serveurs Apache et **MySQL**.

Vous pouvez ensuite dialoguer avec le serveur **MySQL**, soit en mode commande (dans une fenêtre invite de commande), soit via l'interface **PhpMyAdmin**. Nous décrivons ici le mode commande.

1.1 Connexion

```
shell> mysql -h hote -u utilisateur -p
Enter password: *****
```

***** représente votre mot de passe, entrez-le lorsque **mysql** affiche Enter password : . Si vous travaillez en local avec les paramètres par défaut, il n'y a pas de mot de passe. Si tout fonctionne, vous devrez voir quelques informations d'introduction suivies d'une invite de commande **mysql>**

```
shell> mysql -h localhost -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 459 to server version: 3.22.20a-log
```

Type 'help' for help.

```
mysql>
```

L'invite vous dit que mysql attend que vous entriez des commandes.

Après vous être connecté avec succès, vous pouvez vous déconnecter à tout moment en entrant QUIT après l'invite mysql>

1.2 Créer et sélectionner une base

```
mysql> CREATE DATABASE menagerie ;
```

La création d'une base de données ne la sélectionne pas pour l'utilisation ; vous devez le faire explicitement. Pour rendre menagerie la base courante, utilisez cette commande :

```
mysql> USE menagerie ;
```

Votre base a besoin d'être créée juste une fois, mais vous devez la sélectionner pour l'utiliser, chaque fois que vous débutez une session mysql .

1.3 Créer une table

Si une table a été sélectionnée, vous pouvez créer des tables. Par exemple, dans la base ménagerie :

```
mysql> CREATE TABLE animal (nom VARCHAR(20), maitre VARCHAR(20),
-> espece VARCHAR(20), sexe CHAR(1), naissance DATE, mort DATE);
```

VARCHAR est un bon choix pour les colonnes nom , maitre , et espece car leurs valeurs varient en longueur. La longueur de ces colonnes ne doit pas nécessairement être la même, et n'a pas besoin d'être forcément 20 . Vous pouvez choisir une taille entre 1 et 255 , celle qui vous semblera la plus raisonnable (si vous faites un mauvais choix et que vous vous apercevez plus tard que vous avez besoin d'un champ plus long, MySQL fournit la commande ALTER TABLE) .Le sexe des animaux peut être représenté de plusieurs façons, par exemple, "m" et "f" , ou bien "male" et "femelle" . Il est plus simple d'utiliser les caractères simples "m" et "f" .

1.4 Charger des données dans une table

Pour ajouter des données à une table, on peut utiliser les commandes LOAD DATA et INSERT. Nous ne décrivons ici que la commande INSERT.

Dans sa forme la plus simple, vous spécifiez une valeur pour chaque colonne, dans l'ordre où les colonnes sont listées dans la requête CREATE TABLE . Supposons que Diane achète un nouvel hamster nommé Puffball. Vous pourriez ajouter ce nouvel enregistrement en utilisant un INSERT de la façon suivante :

```
mysql> INSERT INTO animal VALUES ('Puffball','Diane','hamster','f','1999-03-30',NULL);
```

Notez que les chaînes de caractères et les valeurs de dates sont spécifiées en tant que chaînes protégées par des guillemets. De plus, avec INSERT vous pouvez insérer la valeur NULL directement pour représenter une valeur manquante.

1.5 Récupérer des informations à partir d'une table

La commande SELECT est utilisée pour récupérer des informations à partir d'une table. La forme usuelle est :

```
SELECT quoiSelectionner
FROM quelTable
WHERE conditionsASatisfaire
```

quoiSelectionner indique ce que vous voulez voir. Cela peut être une liste de colonnes, ou * pour indiquer "toutes les colonnes". quelTable indique la table à partir de laquelle récupérer les données. La clause WHERE est optionnelle. Si elle est présente, conditionsASatisfaire spécifie les conditions que les lignes doivent satisfaire pour être sélectionnées.

Imaginons que nous ayons dans la table animal les données suivantes :

nom	maitre	espece	sexe	naissance	mort
Fluffy	Harold	chat	f	1993-02-04	NULL
Claws	Gwen	chat	m	1994-03-17	NULL
Buffy	Harold	chien	f	1989-05-13	NULL
Fang	Benny	chien	m	1990-08-27	NULL
Bowser	Diane	chien	m	1998-08-31	1995-07-29
Chirpy	Gwen	oiseau	f	1998-09-11	NULL
Whistler	Gwen	oiseau	NULL	1997-12-09	NULL
Slim	Benny	serpent	m	1996-04-29	NULL
Puffball	Diane	hamster	f	1999-03-30	NULL

mysql> SELECT * FROM animal ; affichera la figure précédente.

mysql> SELECT * FROM animal WHERE espece = "chien" AND sexe = "f" ; affichera les lignes de animal concernant les chiennes.

mysql> SELECT nom, naissance FROM animal ; affichera le nom et la date de naissance de tous les animeaux de la table animal.

mysql> SELECT DISTINCT naissance FROM animal ; affichera toutes les dates de naissance différentes de la table. (enlève les répétitions du résultat).

mysql> SELECT nom, naissance FROM animal ORDER BY naissance ; trie le résultat sur le champ naissance.

mysql> SELECT maitre, COUNT(*) FROM animal GROUP BY maitre ;

COUNT() compte le nombre de résultats non NULL. On extrait donc pour chaque maître, son nom et le nombre d'animeaux qu'il possède.

1.6 Utiliser plusieurs tables

```
mysql> CREATE TABLE evenement (nom VARCHAR(20), date DATE,
-> type VARCHAR(15), remarque VARCHAR(255));
```

Imagineons que cette table contienne les enregistrements suivants :

nom	date	type	remarque
Fluffy	1995-05-15	mise bas	4 chatons, 3 femelles, 1 mâles
Buffy	1993-06-23	mise bas	5 chiots, 2 femelles, 3 mâles
Buffy	1994-06-19	mise bas	3 chiots, 3 femelles
Chirpy	1999-03-21	vétérinaire	Redresser le bec
Slim	1997-08-03	vétérinaire	Cotes cassées

```

Bowser  1991-10-12  chenil
Fang    1991-10-12  chenil
Fang    1998-08-28  anniversaire  Don d'un nouvel objet de mastication
Claws   1998-03-17    anniversaire  Don d'un nouveau collier anti-puces
Whistler 1998-12-09   anniversaire  Premier anniversaire

```

Pour trouver l'âge de chaque animal lorsqu'il a mis bas, on a besoin de la table `animal` pour connaître la date de naissance de l'animal et de la table `evenement` pour connaître la date de l'accouchement. Il faut faire une jointure sur les champs noms des 2 tables. On utilisera la fonction `To_Days` qui retourne le nombre de jours depuis l'an 0 correspondant à la date. Voici la requête :

```

mysql> SELECT animal.nom,
-> (TO_DAYS(date) - TO_DAYS(naissance))/365 AS age,
-> remarque
-> FROM animal, evenement
-> WHERE animal.nom = evenement.nom AND type = "mise bas";

```

```

+-----+-----+-----+-----+
| nom    | age  | remarque                |
+-----+-----+-----+-----+
| Fluffy | 2.27 | 4 chatons, 3 femelles, 1 mâle |
| Buffy  | 4.12 | 5 chiots, 2 femelles, 3 mâles |
| Buffy  | 5.10 | 3 chiots, 3 femelles        |
+-----+-----+-----+-----+

```

Pour trouver les paires mâles / femelles par rapport à l'espèce, on fait une jointure de `animal` avec `animal`. On a besoin de donner des alias pour différencier les 2 tables :

```

mysql> SELECT p1.nom, p1.sexe, p2.nom, p2.sexe, p1.espece
-> FROM animal AS p1, animal AS p2
-> WHERE p1.espece = p2.espece AND p1.sexe = "f" AND p2.sexe = "m";

```

```

+-----+-----+-----+-----+-----+
| nom    | sexe | nom    | sexe | espece |
+-----+-----+-----+-----+-----+
| Fluffy | f    | Claws  | m    | chat   |
| Buffy  | f    | Fang   | m    | chien  |
| Buffy  | f    | Bowser | m    | chien  |
+-----+-----+-----+-----+-----+

```

2 PHPMYAdmin

EasyPHP offre un utilitaire : PHPMYAdmin qui donne une interface graphique pour MySQL. Nous l'utiliserons dans le TP.

3 MYSQL et PHP

L'utilisation de MySQL avec PHP s'effectue en 4 temps :

1. Connexion au serveur de données
2. Sélection de la base de données
3. Requête

4. Exploitation des requêtes

On peut ajouter à ça, un 5ème temps facultatif dans le cas d'une connexion simple : fermeture de la connexion.

3.1 Connexion au serveur de données

Pour se connecter au serveur de données, il existe 2 méthodes : Ouverture d'une connexion simple avec la fonction `mysql_connect` ou Ouverture d'une connexion persistante avec la fonction `mysql_pconnect`

La deuxième méthode diffère de la première par le fait que la connexion reste active après la fin du script, mais tous les hébergeurs n'autorisent pas cette possibilité.

Pour se connecter, il faut paramétrer l'adresse du serveur de données ainsi que votre nom d'utilisateur (" login " en anglais) et votre mot de passe (" password " en anglais). En retour, vous obtenez un identifiant. Si l'identifiant est à 0, une erreur s'est produite pendant la phase de connexion. Cette erreur peut venir du fait que l'on ne peut pas joindre le serveur (ex : panne réseau), que le nom du serveur n'est pas bon, que vous n'êtes pas autorisé à accéder à ce serveur ou que votre mot de passe n'est pas correct. Maintenant, si l'identifiant est différent de 0, tout c'est bien passé et vous êtes connecté au serveur de données.

Remarque : Vous pouvez garder cet identifiant dans une variable, mais si vous n'ouvrez pas d'autres connexions en parallèle, ce n'est pas utile.

Exemple : connexion à une base de données sur " Free " :

```
<?php

if( mysql_connect( 'sql.free.fr' , 'monchat.moustique' , 'miaou' ) > 0 )
    echo 'Connexion réussie !' ;
else
    echo 'Connexion impossible !' ;

?>
```

Exemple : connexion en local :

```
<?php

if( mysql_connect( 'localhost' , 'root') > 0 )
    echo 'Connexion réussie !' ;
else
    echo 'Connexion impossible !' ;

?>
```

3.2 Sélection de la base

En fait, les étapes sélection et requête peuvent être faites en même temps, mais il est plus simple si vous utilisez une seule base, de la sélectionner avant de commencer vos requêtes. Ainsi, toutes les requêtes à venir utiliseront cette base par défaut. Pour faire cette sélection, utilisez la fonction `mysql_select_db` et vous lui passez en paramètre, le nom de votre base. Si la connexion n'aboutit pas, la fonction vous retourne "False" (ou "0" si vous préférez) sinon elle retourne "True".

Remarque : Si cela vous chante, ou si vous avez ouvert plusieurs connexions en parallèle, vous pouvez ajouter un second paramètre qui est l'identifiant de la connexion. Si vous n'en donnez pas, la fonction utilise la dernière connexion ouverte.

Exemple :

Connexion à la base de donnée " ma_base "

```
<?php

if( mysql_select_db( 'ma_base' ) == True )
    echo 'Sélection de la base réussie' ;
else
    echo 'Sélection de la base impossible' ;

?>
```

3.3 Requête

On utilise MySQL pour accéder aux données. Pour envoyer Les requêtes MySQL , on peut utiliser 2 fonctions :

- mysql_query dans le cas où la base de données serait déjà sélectionnée.
- mysql_db_query dans le cas où l'on voudrait sélectionner la base en même temps.

Comme pour la connexion, cette fonction retourne un identifiant qui est à "0" lorsqu'une erreur s'est produite. Contrairement aux autres fonctions (connexion et sélection), cet identifiant est très important, il est utilisé pour retrouver les données rapatriées par une requête de sélection.

Lorsque l'on fait une requête de sélection, MySQL retourne les données et PHP les place en mémoire. L'identifiant permet donc de retrouver ces données en mémoire. En fin de script, la mémoire est libérée et les données sont donc perdues. Il est possible de libérer la mémoire, pour ne pas surcharger le serveur, avant la fin du script au moyen de la commande mysql_free_result.

Exemple : On part du principe que la base de données est déjà sélectionnée et on veut avoir des informations sur le membre nommé "moustique" :

```
<?php

$requete = "SELECT * FROM membres WHERE pseudo = 'moustique' ";
$resultat = mysql_query( $requete );

?>
```

3.4 Exploitation des requêtes

Après l'exécution d'une requête de sélection, les données ne sont pas "affichées", elles sont simplement mises en mémoire, il faut donc aller les chercher enregistrement par enregistrement et les afficher avec un minimum de traitement. PHP gère un pointeur de résultat, c'est à dire qu'il repère un enregistrement parmi les autres, et lorsque l'on veut en lire un, c'est celui qui est pointé qui sera retourné. Lorsque vous utilisez une fonction de lecture, le pointeur est déplacé sur l'enregistrement suivant et ainsi de suite jusqu'à ce qu'il n'y en ait plus. Les fonctions qui retournent un enregistrement sont : mysql_fetch_row, mysql_fetch_array et mysql_fetch_object et prennent comme paramètre l'identifiant de la requête. Les 3 exemples suivants partent d'une requête "SELECT nom, prenom, date FROM membres."

mysql_fetch_row : Cette fonction retourne un enregistrement sous la forme d'un tableau simple.

```
<?php

$enregistrement = mysql_fetch_row( $resultat );
// Affiche le champ - nom -
echo $enregistrement[0] . '<br>';
// Affiche le champ - prenom -
echo $enregistrement[1] . '<br>';
// Affiche le champ - date -
```

```

    echo $enregistrement[2] . '<br>';
?>

```

mysql_fetch_array : Cette fonction retourne un enregistrement sous la forme d'un tableau associatif.

```

<?php

    $enregistrement = mysql_fetch_array( $resultat );
    // Affiche le champ - prenom -
    echo $enregistrement['prenom'] . '<br>';
    // Affiche le champ - nom -
    echo $enregistrement['nom'] . '<br>';
    // Affiche le champ - date -
    echo $enregistrement['date'] . '<br>';

?>

```

mysql_fetch_object : Cette fonction retourne un enregistrement sous forme d'une structure (objet).

```

<?php

    $enregistrement = mysql_fetch_object( $resultat );
    // Affiche le champ - date -
    echo $enregistrement->date . '<br>';
    // Affiche le champ - nom -
    echo $enregistrement->nom . '<br>';
    // Affiche le champ - prenom -
    echo $enregistrement->prenom . '<br>';

?>

```

Si il n'y a pas ou plus d'enregistrement à lire, ces fonctions retournent "false." Enfin, si vous voulez savoir combien d'enregistrements ont été retournés par la sélection, vous pouvez utiliser la commande **mysql_num_rows** qui prend comme paramètre l'identifiant de la requête.

3.5 Fermeture de la connexion

Vous pouvez fermer la connexion au moyen de la fonction **mysql_close**, mais il est bon de savoir que cette opération sera faite lorsque le script se terminera.

3.6 Exemple de synthèse

Voici un exemple d'utilisation de MySQL avec PHP. Ce script PHP se connecte sur la base **menagerie**, sélectionne les noms et date de naissance des animaux pour les afficher dans une page html.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
  <head>
    <title>ManipZoo</title>
    <meta http-equiv="Content-Type" content="text/HTML; charset=iso-8859-1" />
  </head>

```

```

<body>

<?php
$mysqlserver = "localhost";
$mysqllogin = "root";
// $mysqlpassword = 'secret';

$connexion= mysql_connect( $mysqlserver , $mysqllogin ) ;
// possiblement un 3 argt = le mot de passe
// retourne 0 si erreur.

if (!$connexion){
echo "<p>connexion impossible au serveur </p> \n";
exit;
}
if (!mysql_select_db("menagerie",$connexion)){
// celle ci fait la connexion a une base
echo "<p>acces a la base refuse </p>\n";
exit;
}

echo "<p>connexion a la base menagerie reussie<p>";
$requete = "SELECT nom, naissance FROM animal ORDER BY naissance";
// Execution de la requête
$resultat = mysql_query($requete, $connexion);
if (!$resultat){
echo "<p>erreur a l'execution de la requete$requete <br />";
echo "message de mysql : ".mysql_error($connexion);
echo "</p>";
exit;
}

echo "<h3> affichage du resultat de la requete </h3>\n";
// Récupération du nombre d'enregistrements
$nombre_animaux = mysql_num_rows( $resultat );
echo "<p>. $nombre_animaux . " </p> \n ";
echo " <ul>\n";
// boucle d'affichage
while( $row = mysql_fetch_array( $resultat ) ) {
echo "<li>";
echo $row['nom'] . ' ' . $row['naissance'] ;
echo "</li>\n";
}

echo "</ul>\n";

?>

</body>
</html>

```