

# Cours Web n°7

## MySQL et PHP

Sandrine-Dominique Gouraud (gouraud@lri.fr)  
Pierre Senellart (pierre@senellart.com)



Semaine du 7 novembre 2005

# Plan du cours

- 1 Introduction aux bases de données
- 2 PHPMyAdmin
- 3 Requêtes SQL
- 4 Accéder à MySQL depuis PHP
- 5 Compléments utiles de PHP
- 6 Références
- 7 Application

- **SGBD** : **S**ystème de **G**estion de **B**ases de **D**onnées
- Fournit des méthodes **efficaces** pour gérer des données qui répondent à une structure (un **schéma**) précis.
- Rechercher des données : **requêtes**
- Ajouter, supprimer, modifier des données : **mises à jour**
- Traite de manière rapide de grandes quantités de données.
- De nombreux produits commerciaux (Oracle, Microsoft SQLServer, IBM DB-2, Microsoft Access... ) et libres (MySQL, PostGreSQL)

- Modèle le plus répandu et le plus classique.
- Les données sont organisées en des **tables**, chacune des colonnes représentant un **attribut** des données.

## Exemple

Prénom	Nom	Date de naissance
Jean	Dupont	1967-08-07
Pascale	Dupuis	1981-09-12
Alfred	Lambert	NULL

- Chaque attribut (colonne) est **typé**.
- SQL (**S**tructured **Q**uery **L**anguage) : langage standard de requête et de mise à jour des données (petites variantes suivant les SGBD).

INT : entier (42)

REAL : nombre en virgule flottante (3.14159)

VARCHAR(N) : chaîne de caractères ayant au plus N caractères; les valeurs sont délimitées par des apostrophes ('Ceci est une chaîne').

TEXT : long texte

DATE : date (2005-11-08)

TIME : temps (09:30:00)

- **NULL** : valeur spéciale
- Dénote l'absence de valeur.
- Différent de 0, de "..."
- Une comparaison normale ( $=, <>$ ) avec NULL renvoie toujours FAUX.
- IS NULL, IS NOT NULL peuvent être utilisées pour tester une valeur.
- Chacune des colonnes doit être déclarée comme acceptant ou non la valeur NULL.

# Plan du cours

- 1 Introduction aux bases de données
- 2 PHPMyAdmin**
- 3 Requêtes SQL
- 4 Accéder à MySQL depuis PHP
- 5 Compléments utiles de PHP
- 6 Références
- 7 Application

- **Interface Web**, écrite en PHP, de gestion d'une base de données MySQL
- Chaque opération réalisée est traduite en un ordre SQL, qui est affiché.

## Fonctionnalités

- création de table
- insertion de lignes
- suppression de lignes
- chargement d'un fichier
- affichage du contenu de la table
- exécution de requêtes SQL
- suppression de table

# Plan du cours

- 1 Introduction aux bases de données
- 2 PHPMyAdmin
- 3 Requêtes SQL**
- 4 Accéder à MySQL depuis PHP
- 5 Compléments utiles de PHP
- 6 Références
- 7 Application

- Une fois, la table créée, on peut :
  - ▶ Insérer des données avec une requête **INSERT**
  - ▶ Sélectionner des données avec une requête **SELECT**
  - ▶ Retirer des données avec une requête **DELETE**
  - ▶ Mettre à jour des données avec une requête **UPDATE**
- Remarque : Les commandes de suppression et de mises à jour sont des variantes du **SELECT**.

- Pour insérer des données *Tuple* dans la table *Table*, on utilise une des requêtes suivantes :

```
INSERT INTO Table  
VALUES Tuple
```

```
INSERT INTO Table nuplet_d_Attributs  
VALUES nuplet_de_valeurs
```

- Dans le dernier cas, les attributs sans valeurs seront initiés à NULL.

## Exemple

```
INSERT INTO Films  
VALUES ('Tootsie',1982,'US')
```

```
INSERT INTO Films (Titre,Annee)  
VALUES ('Wallace et Gromit le mystère du lapin-garou',2005)
```

- Pour sélectionner des données *nuplet\_d\_Attributs* dans la table *Table* selon une condition *Cond*, on utilise la requête suivante :

```
SELECT nuplet_d_Attributs  
FROM Table  
WHERE Cond
```

### Exemple

```
SELECT Titre  
FROM Films  
WHERE Annee > 1980
```

## SELECT

- \* pour sélectionner tous les attributs
- liste d'attributs séparés par une virgule

## FROM

- liste de tables séparés par une virgule

## WHERE

- clause optionnelle
- condition basée sur les opérateurs : AND, OR, LIKE, =, <>, >, >=, etc.

## ORDER BY

- clause optionnelle
- noms de champs séparés par une virgule

### Exemple

```
SELECT *  
FROM Films  
WHERE Pays='UK' OR Pays='US'  
ORDER BY Annee
```

- Pour retirer les données de la table *Table* qui correspondent à la condition *Cond*, on utilise la requête suivante :

**DELETE FROM** *Table* **WHERE** *Cond*

- Pour mettre à jour les données de la table *Table* qui correspondent à la condition *Cond*, on utilise la requête suivante :

**UPDATE** *Table* **SET** *attribut = nouvelle valeur* **WHERE** *Cond*

## Exemple

```
DELETE FROM Films  
WHERE Annee>2005
```

```
UPDATE Films  
SET Pays='RU'  
WHERE Pays='UK'
```

# Plan du cours

- 1 Introduction aux bases de données
- 2 PHPMyAdmin
- 3 Requêtes SQL
- 4 Accéder à MySQL depuis PHP**
- 5 Compléments utiles de PHP
- 6 Références
- 7 Application

`mysql_pconnect("localhost", "login", "password")` permet d'établir une connexion avec MySQL pour un compte utilisateur *"login"*, *"password"* et un serveur donné *"localhost"*. Cette fonction renvoie une valeur qui sera ensuite utilisée pour communiquer avec le serveur.

`mysql_select_db("base", $connexion)` permet de se placer dans le contexte de la base de données *"base"* du serveur *\$connexion*.

`mysql_query("requête", $connexion)` permet d'exécuter la requête *"requête"* sur le serveur *\$connexion*. Cette fonction renvoie une variable représentant le résultat de la requête.

`mysql_fetch_objet($resultat)` permet de récupérer une des lignes du résultat *\$resultat*, et positionne le curseur sur la ligne suivante. La ligne est représentée sous forme d'un objet *objet* et on accède aux attributs de cet objet via *objet->attribut*

`mysql_error()` Renvoie le message de la dernière erreur rencontrée

`mysql_close($connexion)` ferme la connexion *\$connexion* avec MySQL

## Exemple

```
<?php
$connexion=mysql_pconnect("localhost","dupont","a1b2c3d4");
if(!$connexion)
{ echo "Desolé, connexion à localhost impossible"; exit;}
if(!mysql_select_db('ifips',$connexion))
{ echo "Desolé, accès à la base ifips impossible"; exit;}
$resultat= mysql_query("SELECT * FROM Films",$connexion);
if($resultat){
    while($film=mysql_fetch_object($resultat)){
        echo "$film->Titre est paru en $film->Annee <br />";
    }
}
else{
    echo "Erreur dans l'exécution de la requête.<br />";
    echo "Message de MySQL : ".mysql_error($connexion);
}
mysql_close($connexion);?>
```

# Plan du cours

- 1 Introduction aux bases de données
- 2 PHPMyAdmin
- 3 Requêtes SQL
- 4 Accéder à MySQL depuis PHP
- 5 Compléments utiles de PHP**
- 6 Références
- 7 Application

`htmlEntities` permet de remplacer tous les caractères réservés de XHTML par l'entité correspondante. Par exemple, le caractère `&` sera remplacé par `&amp` ;

`mysql_escape_string` permet d'ajouter les caractères d'échappement dans une chaîne de caractères (utile pour insérer une valeur dans une requête MySQL).

`stripSlashes` permet de supprimer les caractères d'échappement dans une chaîne de caractères (utile pour récupérer une valeur d'un résultat de requête MySQL).

`header("Location : nouvelle_page.php")` redirige vers la page `nouvelle_page.php`

# Plan du cours

- 1 Introduction aux bases de données
- 2 PHPMyAdmin
- 3 Requêtes SQL
- 4 Accéder à MySQL depuis PHP
- 5 Compléments utiles de PHP
- 6 Références**
- 7 Application

- <http://dev.mysql.com/doc/>
- <http://www.php.net/>
- *Pratique de MySQL et PHP*, Philippe Rigaux, O'Reilly

# Plan du cours

- 1 Introduction aux bases de données
- 2 PHPMyAdmin
- 3 Requêtes SQL
- 4 Accéder à MySQL depuis PHP
- 5 Compléments utiles de PHP
- 6 Références
- 7 Application**

Reproduire le modèle de scripts permettant l'insertion, l'affichage et la suppression de films dans une base de données MySQL, de la manière suivante :

- 1 Se connecter sur PHPMyAdmin et créer une table Films permettant de stocker un titre, une année (éventuellement NULL) et un pays (écrit comme un code à deux caractères).
- 2 Utiliser PHPMyAdmin pour ajouter à la main quelques entrées dans la table.
- 3 Créer un script `affichage.php` permettant d'afficher sous la forme d'un tableau XHTML le contenu de la table Films, trié par titre.
- 4 Récupérer le formulaire `ajout_nouveau_film.html` et créer un script `insert.php` permettant d'insérer un nouvel élément dans la table de films à partir des paramètres passés par le formulaire.
- 5 Ajouter à la fin de `insert.php` une redirection vers `affichage.php`
- 6 Ajouter au tableau de `affichage.php` une colonne contenant des formulaires contenant des boutons *Supprimer*.
- 7 Écrire le script `supprimer.php` correspondant.