

# Chapitre 7 :

## PHP & MySQL : Partie 1

### 1. Gestion des données avec MySQL

#### a. La requête de création de table

On peut créer la table manuellement via l'interface phpMyadmin, mais il y a aussi un autre moyen pour créer une table. **Nous pouvons nous même dire directement au système de créer la table à l'aide d'une requête SQL.**

```
CREATE TABLE `ecole`.`eleve` (  
  `numero` INT( 5 ) NOT NULL ,  
  `nom` VARCHAR( 30 ) NOT NULL ,  
  `prenom` VARCHAR( 30 ) NOT NULL ,  
  `datenaissance` DATE NOT NULL ,  
  `adresse` TINYINT NOT NULL ,  
  PRIMARY KEY ( `numero` )  
)
```

Dans cet exemple ecole est la base de données et eleve est le nom de la table. **Le mot clé à retenir pour la création de table est CREATE TABLE.** Il ne faut pas oublier de séparer la définition de chaque champ par une virgule.

#### b. Insertion de données dans un table : INSERT INTO

En spécifiant le nom de la table, la commande INSERT INTO permet d'insérer des données sous forme d'enregistrements dans une table. Les valeurs à insérer sont listées dans la clause values et séparées par des virgules.

```
INSERT INTO `eleve` (  
  `numero` ,  
  `nom` ,  
  `prenom` ,  
  `adresse`  
)  
VALUES (  
  '1', 'Cassidy', 'Hope', 'Appartement 4 21000 Rue de la charité Pays'  
)
```

#### c. Sélection de données : SELECT

On peut avoir la liste de tous les élèves de la table élève en utilisant le mot clé SELECT, suivie de la clause FROM, puis du nom de la table.

La requête associée est donc :

```
SELECT * FROM `eleve` ;
```

**L'utilisation du signe \* veut dire que l'on veut avoir tout les enregistrements.** Dans le cas où on veut seulement avoir les enregistrements répondant à certains critères, on emploie le mot clé WHERE.

On spécifie une certaine condition dans la clause where :

```
SELECT * FROM `eleve` WHERE `numero` =1.
```

Cette requête demande à la base de données de sélectionner l'élève numéro 1, et de nous retourner le résultat.

**Il peut y avoir plusieurs conditions :**

```
SELECT *  
FROM `eleve`  
WHERE `nom` LIKE 'Cassidy'  
AND `prenom` LIKE 'Hope'
```

Traduite en langage humain, cela signifie qu'il faut sélectionner et afficher tout les élèves dont le nom est "Cassidy" et dont le prénom est "Hope"

#### **d. Modification de données : UPDATE**

Le mot clé utilisé pour la modification des données est UPDATE. **En l'associant à la clause WHERE, il est possible de rechercher des données ayant certains critères et les modifier.**

Pour modifier le prénom de l'élève numéro 1 par exemple on utilisera la syntaxe suivante:

```
UPDATE `eleve` SET `prenom` = 'Vanessa' WHERE `eleve`.`numero` =1 ;
```

Il est aussi possible de modifier plusieurs champs d'un seul coup avec cette syntaxe :

```
UPDATE `eleve` SET `nom` = 'Paradis', `prenom` = 'Vanessa' WHERE `eleve`.`numero` =1 ;
```

Ici, on modifiera le nom et le prénom à la fois. Il ne faut pas oublier que les champs sont toujours séparés par des virgules, contrairement aux conditions, qui sont séparées par des opérateurs booléens (AND, OR, ...)

#### **e. Suppression de données : DELETE**

On peut procéder de la même façon pour supprimer un (ou plusieurs) enregistrement(s) sauf que cette fois ci on utilisera les mots clé DELETE FROM.

```
DELETE FROM `eleve` WHERE `numero` =1 ;
```

La requête signifie qu'il faut supprimer de la table eleve l'élève dont 'numéro' = '1'.

## **2. Interaction PHP et MySQL**

### **a. Insertion des données saisies**

Dans un premier temps, nous allons suivre pas à pas les étapes de la récupération des données d'un formulaire pour l'introduire ensuite dans la base.

Dans un second temps, nous allons faire l'inverse. Nous sélectionnerons des données dans la base que l'on va ensuite récupérer à l'aide de PHP et ensuite on l'affichera dans un tableau html.

Pour y voir plus clair, nous allons séparer dans un premier temps le formulaire HTML du traitementPHP. Afin d'éviter de trainer sur des sujets qui dépassent le cadre de ce chapitre, on ne va pas faire une explication détaillé sur les formulaires.

Puisque cette notion a déjà été abordée dans un des chapitres précédents, on se contentera d'utiliser un formulaire tout fait. Il est conseillé à ceux qui en ont besoin de se rendre dans les **tutoriels XHTML** pour en savoir plus sur les **formulaires**.

#### **Code XHTML du formulaire dans formulaire.html**

```
<form method="post" action="traitement.php">

  <legend>Renseignement sur un élève</legend>

  <label for="nom">Nom: </label>
  <input type="text" name="nom" id="nom" />

  <label for="prenom">Prénom: </label>
  <input type="text" name="prenom" id="prenom" />

  <label for="adresse">Adresse: </label>
  <textarea name="adresse"> </textarea>

  <input type="submit" value="envoyer" />
</form>
```

Voici donc un formulaire de saisie qui nous permettra de saisir des élèves.

### **b. Traitement des données saisies**

Une fois que l'on aura saisi les données, l'étape suivante consistera à les récupérer et à les insérer dans la base de données.

- **Connexion à la base**

Avant tout, il est quand même important de savoir qu'à chaque fois que l'on veut effectuer un traitement mysql en PHP, il nous faut ouvrir et fermer une connexion à la base. Que se soit une insertion de données dans la base ou une sélection, toute communication avec la base nécessite la connexion à la base de donnée tout au début de la manipulation.

Une fois que l'on en aura fini avec la base, il faudra libérer et fermer cette connexion.

**Les codes de connexion et de déconnexion risquent d'être utilisé partout où il y a une communication avec la base.** C'est pourquoi on a intérêt à les mettre sous forme de fonction, et à faire en sorte que les fonctions soient accessibles partout. Et voilà le bon retour des termes de fonctions et inclusion de fichier dont on ne peut jamais se passer.

De nombreuses fonctions de base sont utilisées pour la connexion, en voici quelques une qui sont bonnes à savoir :

- Connexion au serveur: `mysql_connect()`
- Sélection de la base de données : `mysql_select_db()`
- Fermeture de la base : `mysql_close()`
- Code dans `fonctionBD.php`

```
<?php
//ouverture de la connexion
function connect_bd()
{
    $nomserveur='localhost'; //nom du seueur
    $nombd='ecole'; //nom de la base de données
    $login='root'; //login de l'utilisateur
    $pass=""; // mot de pass
    $bd=mysql_connect($nomserveur, $login, $pass)or die("Connexion échouée");
    mysql_select_db($nombd,$bd)or die("La base ne peut pas être sélectionnée");
    return $bd;
}

//fermeture de la connexion
function deconnect_bd($bd)
{
    mysql_close($bd);
    $db=0;
}
?>
```

Ces deux fonctions `connect_bd` et `deconnect_bd` servent respectivement à l'ouverture et à la fermeture de la connexion. Le fichier sera alors inclus à chaque fois que l'on aura besoin d'une communication avec la base.

- **Traitement de données**

Passons maintenant au vif du sujet. On a maintenant les données, il ne nous reste plus qu'à les récupérer, communiquer avec la base et les insérer dans la base. Tout cela va se dérouler dans un fichier `traitement.php`.

```
<?php
//inclure les fonctions pour la base
include("fonctionBD.php");

//récupération des données et les formater un peu
```

```

$nom = addslashes($_POST["nom"]);
$prenom = addslashes($_POST["prenom"]);
$adresse = addslashes($_POST["adresse"]);

//Insertion de données
$bd=connect_bd();
if($bd<>0)
{

    $requete="insert into eleve VALUES ('', '$nom','$prenom','$adresse')";
    $resultat=@mysql_query($requete,$bd);
    if($resultat<>FALSE)
    {
        echo 'Enregistrement réussi, merci. Les valeurs insérées sont :<br />';
        echo $nom.'<br />';
        echo $prenom.'<br />';
        echo $adresse.'<br />';

    }
    else
    {
        echo "Echec de l'enregistrement, désolé";
    }
    deconnect_bd($bd);
}
?>

```

Après l'inclusion des fichiers, on a récupéré les données à l'aide du tableau \$\_POST qui n'est plus un inconnu pour nous.

Il est vrai qu'il y a encore un certain nombre de choses que l'on a zappé, comme les vérifications des champs saisi par exemple. Entre temps, le javascript peut s'occuper de cela. **Néanmoins, on a essayé d'être un petit peu prudent et on a utilisé la fonction addslashes.** Cette fonction rajoutera des antislashes dans une chaîne là où il y a des caractères à échapper. Si jamais l'utilisateur nous a par exemple rentré une adresse telle que " avenue de l'indépendance ", la fonction nous rajoutera automatiquement un antislash avant le guillemet afin d'éviter toute erreur.

Ceci n'est qu'un exemple, mais de bon nombre de formatages peuvent et doivent être effectués avant que les données ne soient prêtes à être insérées dans la base : **Laisser un internaute remplir des champs que l'on insère dans la base de données présente un risque potentiel, tant dans l'intégrité des données que du point de vue de la sécurité.**

Ensuite vient le tour de l'insertion de la base. C'est la fonction mysql\_query qui s'occupe de l'envoi de la requête d'insertion au serveur.

Remarque : Pour mieux gérer la clé primaire, on a mis le champ numéro en auto incrément. Pour ce faire, passer dans la structure du champ le paramètre 'extra( en auto incrément. On n'aura plus besoin alors de saisir le numéro, il s'incrémentera automatiquement à chaque insertion d'un enregistrement.

### c. Sélection et Affichage des données

Maintenant, faisons l'inverse. Après avoir inséré bon nombre de données dans la base, essayons de les sélectionner et de les afficher.

### Code dans selection.php

```
<?php
//select
$bd=connect_bd();
if($bd<>0)
{
    $requete="select * from eleve";
    $resultat=@mysql_query($requete,$bd);
    if($resultat<>FALSE)
    {
        echo "<table border='1'>";
        //entête du tableau
        echo "<tr>";
        echo "<th>Numero</th>";
        echo "<th>Nom</th>";
        echo "<th>Prénom</th>";
        echo "<th>adresse</th>";
        echo "</tr>";
        //fin de l'entête

        while ($row=mysql_fetch_array($resultat))
        {
            echo "<tr>";
            echo "<td>".$row['numero']. "</td>";
            echo "<td>".$row['nom']. "</td>";
            echo "<td>".$row['prenom']. "</td>";
            echo "<td>".$row['adresse']. "</td>";
            echo "</tr>";
        }
        echo "</table>";
    }
    else
    {
        echo "Echec de l'enregistrement, désolé";
    }
    deconnect_bd($bd);
}
?>
```

La fonction `mysql_fetch_array()` nous retourne une ligne de résultats sous la forme d'un tableau, dont les clés sont les noms des champs sélectionnés (numero, nom,...)