

BASE DE DONNÉES MYSQL ET PHP AVEC UN EXEMPLE

- 16 Aout 2017
- M. ESSADDOUKI
- Langage PHP
- 14494 Visites

I. Introduction

Création d'une base de données avec PHPMyAdmin.

On prendre un exemple de base de données :

Etudiant

Matricule (varchar 12)

Nom_etud (varvhar 30)

Prenom_etud (varchar 30)

Date_nais (date)

Sexe (varchar 1)

Adresse (varchar 50)

En utilisant **phpMyAdmin**, créer une base de données « **tp_php** », puis créer dans cette base de données, une table « Etudiants » comme le schéma suivant :

```
CREATE TABLE `etudiants` (  
  
    `matricule` varchar(12) NOT NULL,  
  
    `nom_etud` varchar(30) NOT NULL,  
  
    `prenom_etud` varchar(30) NOT NULL,  
  
    `date_nais` date NOT NULL,  
  
    `sexe` char(1) NOT NULL,
```

```
`adresse` varchar(50) NOT NULL,  
  
PRIMARY KEY(`matricule`)  
  
)
```

On peut créer la table avec le script DDL (Data Definition Language) suivant :

Pour exécuter ce script, sélectionner votre base « tp_php » dans phpMyAdmin, puis cliquer sur l'onglet « SQL », puis coller ce script et cliquer sur « Exécuter ».

Essayer maintenant les fonctionnalités suivantes :

- Insertion des données ;
- Modifier la structure d'un champ (ajouter, supprimer ou modifier le type d'un champ...) ;
- Vider ou supprimer la table ;
- Exporter /Importer la base de données.

Avant de continuer, je vous rappelles de types de requettes dans une base de données :

- **DDL (Data Definition Language)** : CREATE, ALTER et DROP.
- **DML (Data Manipulation Language)** : INSERT, UPDATE et DELETE.
- **DQL (Data Query Language)** : SELECT, SHOW et HELP
- **DCL (Data Control Language)** : GRANT et REVOKE.
- **DTL (Data Transaction Language)** : START TRANSACTION, SAVEPOINT, COMMIT et ROLLBACK.

Connexion à une base de données MySQL et affichage des requêtes.

En effet, PHP propose plusieurs moyens de se connecter à une base de données MySQL.

L'extension mysql : ce sont des fonctions qui permettent d'accéder à une base de données MySQL et donc de communiquer avec MySQL. Leur nom commence toujours par mysql_. Toutefois, ces fonctions sont vieilles et on recommande de ne plus les utiliser aujourd'hui.

L'extension mysqli : ce sont des fonctions améliorées d'accès à MySQL. Elles proposent plus de fonctionnalités et sont plus à jour.

L'extension PDO : c'est un outil complet qui permet d'accéder à n'importe quel type de base de données. On peut donc l'utiliser pour se connecter aussi bien à MySQL que PostgreSQL ou Oracle.

NB : Dans notre TP ; nous choisissons l'extension `mysqli` , car elle est plus utilisée à nos jours.

Connexion d'une base de données avec `mysqli` :

Pour se connecter à MySQL, on va utiliser cette fonction :

```
<?php

$bdd=mysqli_connect(serveur,utilisateur,mot_de_passe,base);

?>
```

- **\$bdd** : correspond à une variable où seront stockées les informations de la base de données. On s'en servira souvent.
- **serveur** : correspond au serveur MySQL (souvent **localhost** ou **127.0.0.1**).
- **utilisateur** : correspond au nom d'utilisateur pour se connecter au serveur SQL (par défaut **root**).
- **mot_de_passe** : correspond au mot de passe pour le serveur MySQL !
- **base** : correspond à votre base de données du serveur MySQL.

On peut tester si la connexion a réussi ou non :

```
<?php

if($bdd=mysqli_connect(serveur,utilisateur,mot_de_passe,base)){

    echo 'connexion réussite !!';

}

else{

    die(mysqli_connect_error());

}
```

?>

Dans cet exemple, MySQLi se connecte à la base 'tp_php' dans notre serveur local.

On utilise souvent la méthode fréquente suivante :

<?php

```
$bdd=mysqli_connect(serveur,utilisateur,mot_de_passe,base) or die(mysqli_connect_error());
```

?>

Je vous conseille de mettre ce code dans un fichier **connexion.php**, que vous insérerez ensuite dans chaque fichier qui travaille avec MySQL.

Les requêtes simples :

Nous allons apprendre à gérer les requêtes simples.

Le code de la requête est le suivant :

<?php

```
$resultat=mysqli_query($bdd,'requete');
```

?>

\$resultat est la variable où seront stockées les données.

\$bdd est la variable retournée par la fonction de connexion à la base de données, voir plus haut.

'requete' est une requête SQL.

Exemple:

```
<?php
```

```
    $resultat=mysqli_query($bdd,'SELECT * FROM etudiants');
```

```
?>
```

Voilà, **requete** était une simple requête SQL !

Mais maintenant, comment afficher ces résultats ?

Afficher le résultat d'une requête simple : Pour afficher le résultat de ce type de requêtes, il suffit de fetcher dans une boucle while.

```
<?php
```

```
    while($donnee=mysqli_fetch_assoc($resultat)){
```

```
        echo $donnee['matricule']." <br />";
```

```
        echo $donnee['nom_etud']." <br />";
```

```
        echo $donnee['prenom_etud']." <br />";
```

```
        echo $donnee['date_nais']." <br />";
```

```
        if ($donnee['matricule']=='M') echo 'Masculin <br />';
```

```
        else echo 'Féminin <br />';
```

```
        echo $donnee['matricule']." <br />";
```

```
    }
```

```
?>
```

Que se passe-t-il ? En réalité, c'est relativement simple : `$donnees` est transformé en un tableau associatif avec les noms des colonnes demandées dans la requête. Cette transformation est réalisée grâce à la fonction `mysqli_fetch_assoc($resultat)`, `$resultat` étant le résultat brut de la requête. Cette fonction sert aussi et surtout à passer au résultat suivant, d'où l'utilisation de la boucle `while`.

A la fin, il ne faut pas oublier de fermer le curseur avec la fonction suivante :

```
<?php

    mysqli_free_result($resultat);

?>
```

Fermer le curseur permet de libérer les données récupérées lors d'une requête SQL.

Résumé : (contenue du fichier « **affichage.php** »)

```
<?php

include_once('connexion.php');

$resultat=mysqli_query($bdd,'SELECT * FROM etudiants');

while($donnee=mysqli_fetch_assoc($resultat)){

    echo $donnee['matricule']." <br />";

    echo $donnee['nom_etud']." <br />";

    echo $donnee['prenom_etud']." <br />";

    echo $donnee['date_nais']." <br />";
```

```
if ($donnee['matricule']=='M') echo 'Masculin <br />';
```

```
else echo 'Féminin <br />';
```

```
echo $donnee['matricule']." <br />";
```

```
}
```

```
mysqli_free_result($resultat);
```

```
?>
```

NB : Pour déconnecter la base de données, utiliser la fonction :

```
<?php
```

```
mysqli_close($bdd);
```

```
?>
```

Le nombre de lignes :

Il est parfois pratique, quand nous avons un jeu de résultats de savoir combien de lignes il possède, sans pour autant le parcourir. Pour ce faire, il suffit d'utiliser la fonction **mysqli_num_rows** :

```
<?php
```

```
echo 'le nombre des étudiants est :'.mysqli_num_rows($resultat);
```

```
?>
```

Le dernier id ajouté :

C'est une autre petite fonction bien utile. Vous faites un insert dans une table avec un **auto-increment** et vous voulez récupérer le dernier id ajouté sans pour autant passer par un "select max (id) from table1".

```
<?php
```

```
    $last_id=mysqli_insert_id($bdd);
```

```
?>
```

Affichage des résultats dans une table (HTML) :

Passons directement vers notre exemple :

```
<table>
```

```
    <thead>
```

```
        <tr>
```

```
            <th>Matricule</th>
```

```
            <th>Nom</th>
```

```
            <th>Prénom</th>
```

```
            <th>Date de naissance</th>
```

```
            <th>Sexe</th>
```

```
            <th>Adresse</th>
```

```
        </tr>
```

```
    </thead>
```

```
</table>
```

```
<?php
```

```
    include_once('connexion.php');
```



```

$resultat=mysqli_query($bdd,'SELECT * FROM etudiants');

while($donnee=mysqli_fetch_assoc($resultat)){

    echo '<tr>';

        echo "<td>".$donnee['matricule']." </td>";

        echo "<td>".$donnee['nom_etud']." </td>";

        echo "<td>".$donnee['prenom_etud']." </td>";

        echo "<td>".$donnee['date_nais']." </td>";


        if ($donnee['matricule']=='M') echo '<td>Masculin </td>';

        else echo '<td>Féminin </td>>';


        echo "<td>".$donnee['adresse']." </td>";

        echo '</tr>';

    }

    mysqli_free_result($resultat);

?>

```

II. Mise à jour d'une table de base de données.

Continuons avec notre table « Etudiants » :

Etudiants(matricule, nom_etud, prenom_etud, date_nais, sexe, adresse)

Cette opération s'appelle « **CRUD** », c'est-à-dire **C**reate ; **R**ead ; **U**ppdate et **D**eleter d'un enregistrement d'une table.

Création d'un nouveau Enregistrement :

Ajouter la ligne suivante à la fin du fichier **index.php** qui contient le tableau réalisé dans le tp précédent:

```
<a href="ajouter.php">Ajouter un Etudiant </a>
```

Puis créer le fichier **ajouter.php**, qui contient le formulaire de saisie des données de l'étudiant :

```
<h1>Nouveau Etudiant : </h1>
```

```
<form action="ajouter.php" method="POST">
```

```
<label>Matricule : </label>
```

```
<input type="text" name="matricule" size="12" /><br />
```

```
<label>Nom : </label>
```

```
<input type="text" name="nom_etud" size="30" /><br />
```

```
<label>Prénom : </label>
```

```
<input type="text" name="prenom_etud" size="30" /><br />
```

```
<label>Date de naissance : </label>
```

```
<input type="date" name="date_nais" /><br />
```

```
<label>Sexe :</label>
```

```
<select name="sexe">
```

```
<option value="M">Masculin</option>
```

```
<option value="F">Féminin</option>
```

```
</select>
```

```
<label>Adresse : </label>
```

```
<input type="text" name="adresse" size="30" /><br />
```

```
<input type="submit" value="Enregistrer" />
```

```
<input type="reset" value="Effacer " /><br /><br />
```

```
<a href="index.php">Retour a la page d'accueil</a>
```

```
</form>
```

Puis nous ajoutons le script d'ajout d'un étudiant dans la base de données juste après le formulaire précédent :

```
?php
```

```
require_once 'connexion.php';
```

```

if($_POST){

    extract($_POST);

    $sql="INSERT INTO etudiants
VALUES('$matricule','$nom_etud','$prenom_etud','$date_nais','$sexe','$adresse')";

    $resultat=mysqli_query($bdd,$sql);

    if($resultat){

        echo "Etudiant enregistré avec succès.";

    }

    else{

        echo "Erreur d'enregistrement d'un étudiant";

    }

    mysqli_close($bdd);

}

?>

```

Remarque :

Extract(\$array) : C'est une fonction qui permet de créer des variables à partir des éléments du tableau \$array, dont les noms des variables sont les indices du tableau et ces valeurs sont les valeurs du tableau associées.

Dans notre exemple, extract(\$_POST) est équivalent à :

```

<?php

$matricule=$_POST['matricule'];

```

```
$nom_etud=$_POST['nom_etud'];
```

```
$prenom_etud=$_POST['prenom_etud'];
```

```
$date_nais=$_POST['date_nais'];
```

```
$sexe=$_POST['sexe'];
```

```
$adresse=$_POST['adresse'];
```

```
?>
```

Modification d'un Enregistrement :

Dans le fichier **index.php** ajouter la colonne suivante :

partie (header) :

```
<th>Actions</th>
```

partie (body) avant la cloture du tableau avec </tr> :

```
<?php
```

```
$matricule=$donnee['matricule'];
```

```
echo"<a href='modifier.php?matricule=$matricule' >Modifier</a>";
```

```
?>
```

Ce dernier code va générer un lien comme :

http://localhost/tp_php/modifier.php?matricule=123

(Supposant que "123" est le matricule de l'étudiant qui nous voulons modifier ces informations).

Puis créer le fichier **modifier.php** :

Les variables du formulaire seront reçues à l'aide d'un Array **_GET**.

Puis on affiche les valeurs (les informations) de l'étudiant dans une autre formulaire, qui va envoyer les valeurs modifiées vers la même page en utilisant cette fois la méthode **POST**. Après la réception des valeurs (modifiées) envoyés par la méthode POST, on va les stocker dans la base de données.

```
<h1>Modification d'un Etudiant :</h1>
```

```
<?php
```

```
require_once 'connexion.php';
```

```
if(isset($_GET['matricule'])){\
```

```
    $matricule=$_GET['matricule'];
```

```
    $sql="SELECT * FROM etudiants WHERE matricule='$matricule'";
```

```
    $resultat=mysqli_query($bdd,$sql);
```

```
    if($resultat){
```

```
        $rows=mysqli_fetch_assoc($resultat);
```

```
        extract($rows);
```

```
        mysqli_free_result($resultat);
```

```
    }
```

```
    else{\
```

```
        echo "aucun etudiant avec le matroculé :".$matricule;
```

```
        return;
```

```
    }

    }else{

        echo 'veuillez choisir un etudiant !';

        return;

    }

?>
```

Le formulaire de modification :

```
<form action='modifier.php?matricule=<?php echo $matricule; ?>' method="POST">

<label>Matricule : </label>

<input type="text" name="matricule" value='<?php echo $matricule; ?>' size="12" /><br />

<label>Nom : </label>

<input type="text" name="nom_etud" value='<?php echo $nom_etud; ?>' size="30" /><br />

<label>Prénom : </label>

<input type="text" name="prenom_etud" value='<?php echo $prenom_etud; ?>' size="30" /><br />

<label>Date de naissance : </label>

<input type="date" name="date_nais" value='<?php echo $date_nais; ?>' /><br />
```

```
<label>Sexe :</label>
```

```
<select name="sexe">
```

```
<option value="M" <?php if($sexe=="M") echo 'selected';?> >Masculin</option>
```

```
<option value="F" <?php if($sexe=="F") echo 'selected';?> >Féminin</option>
```

```
</select>
```

```
<label>Adresse : </label>
```

```
<input type="text" name="adresse" value='<?php echo $adresse; ?>' size="30" /><br />
```

```
<input type="submit" value="Modifier" />
```

```
<input type="reset" value="Effacer " /><br /><br />
```

```
<a href="index.php">Retour a la page d'accueil</a>
```

```
</form>
```

Le script de modification:

```
<?php
```

```
    if($_POST){
```

```
        extract($_POST);
```

```
        $sql="UPDATE etudiants SET nom_etud='$nom_etud',
```

```
            prenom_etud='$prenom_etud',
```



```

        date_nais='$date_nais',

        sexe='$sexe',

        adresse='$adresse'

        WHERE matricule='$matricule'

    ";

    $resultat=mysqli_query($bdd,$sql);

    if($resultat) echo "Etudiant modifié avec succès";

    else echo "Erreur de modification d'un étudiant";

    mysqli_close($bdd);

}

?>

```

Remarque : Pour le bon fonctionnement de ce script, insérer la 3^{ème} partie (qui commence par if(\$_POST) ...) au début du fichier.

Suppression d'un Enregistrement :

Dans le fichier « index.php » ajouter la ligne suivante :

```

<?php

echo "<td>";

echo "<a href='modifier.php?matricule=$matricule'>Modifier</a>";

echo ' / ';

```

```
echo "<a href='supprimer.php?matricule=$matricule'>Supprimer</a>";
```

```
echo "</td>";
```

```
?>
```

Le script de suppression :

```
<?php
```

```
    require_once 'connexion.php';
```

```
    if(isset($_GET['matricule'])){
```

```
        $matricule=$_GET['matricule'];
```

```
        $sql="DELETE FROM etudiants WHERE matricule='$matricule'";
```

```
        $resultat=mysqli_query($bdd,$sql);
```

```
        if($resultat){
```

```
            header('Location:index.php?action=vrai');
```

```
        }
```

```
        else{
```

```
            header('Location:index.php?action=non');
```

```
        }
```

```
    }else{
```

```
        header('Location:index.php');
```

```
    }
```

```
?>
```

Et pour afficher le message « Etudiant supprimé avec succès » ; on ajoute dans la page « **index.php** » :

```
<?php

if(isset($_GET['action'])){

    if($_GET['action']=='vrai'){

        echo "<div>Etudiant supprimé avec succès </div>";

    }

    else if($_GET['action']=='non'){

        echo "<div>Erreur de suppression de l'étudiants </div>";

    }

}

?>
```

Comme ça on termine la mise à jour de notre table de base de données.