

# PHP - Bases de données et PHPMyAdmin

Date de dernière mise à jour : 27/06/2007 à 19:36

Source : <http://www.vulgarisation-informatique.com/php-bdd.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)

## Introduction

Les bases de données sont des éléments très importants en PHP. On les utilise pour stocker des données. Leur gros avantage est que via un langage dédié, on va pouvoir récupérer les données stockées très rapidement, faire de nombreuses opérations dessus, tout ça en un temps record !

Par exemple, le [forum informatique](#) de ce site repose entièrement sur l'utilisation d'une base de données qui contient les messages, les sujets, etc ...

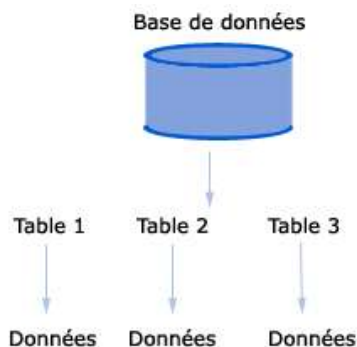
On abrège souvent "base de données" par BDD, donc si vous me voyez écrire cela plus loin, ne vous posez pas de questions ;) On appelle SGBD un système de gestion de base de données. L'avantage du SGBD est que pour aller chercher vos données, vous ne vous préoccupez pas de savoir où sont stockées les données ni comment, vous dites "Je veux telle donnée" et le SGBD va la chercher pour vous. Vous pourrez également récupérer des données d'une manière très compliquée qui dépendent d'énormément de facteurs, chose quasi impossible en stockant les données dans un fichier sans de nombreux calculs très lourds. Le langage que l'on utilise pour interroger et effectuer des opérations sur une ou plusieurs bases de données/tables s'appelle le SQL.

Pour tout ce qui suit, nous allons utiliser un SGBD appelé MySQL. Il s'agit du SGBD le plus répandu, vous ne devriez donc pas avoir de problème à faire fonctionner votre site ultérieurement. Cependant, PHP supporte d'autres SGBD dont voici une liste non exhaustive :

- mSQL
- Sybase
- MySQL
- Oracle
- Unix dbm
- Informix
- SQL Server
- PostgreSQL.

## Les tables

Une base de données contient des tables. Faisons une analogie : Vous avez un bureau qui représente la base de données. Sur ce bureau vous avez des dossiers, ce sont les tables. Ces dossiers contiennent des documents, pour nous les données.



Une table contient des champs. Chaque champ va pouvoir recevoir un type de données bien précis. On ne stocke pas un numéro de

sécurité sociale avec le même type que le nom et le prénom. On peut représenter une table sous la forme d'un tableau :

Champs	id_cat	nom	description
Données	1	Informatique	Une description ici ;)
	2	Catégorie de test	Une catégorie de test à voir

Ici, nous avons trois champs :

**id\_cat** - : c'est un numéro destiné à distinguer toutes les catégories. Plus tard, on verra qu'il s'agit aussi d'une question d'optimisation. On n'utilise pas le champ "nom" pour distinguer les catégories, mais un champ numérique, qui sera bien plus rapide à traiter pour MySQL.

**nom** - : Il s'agit d'un nom de catégorie

**description** - : il s'agit d'une description de notre catégorie

Chacune des données que vous allez stocker dans la table va remplir les champs, et une donnée correspond ici à une ligne du tableau. Par exemple pour stocker des articles on aurait pu prendre une liste de champs comme "numero, titre, texte" le tout formant la table "articles". Il est bien important de savoir qu'une table peut comporter un nombre infini de données (dans la limite de vos ressources matérielles bien entendu). Une base de données peut elle aussi comporter une infinité de tables.

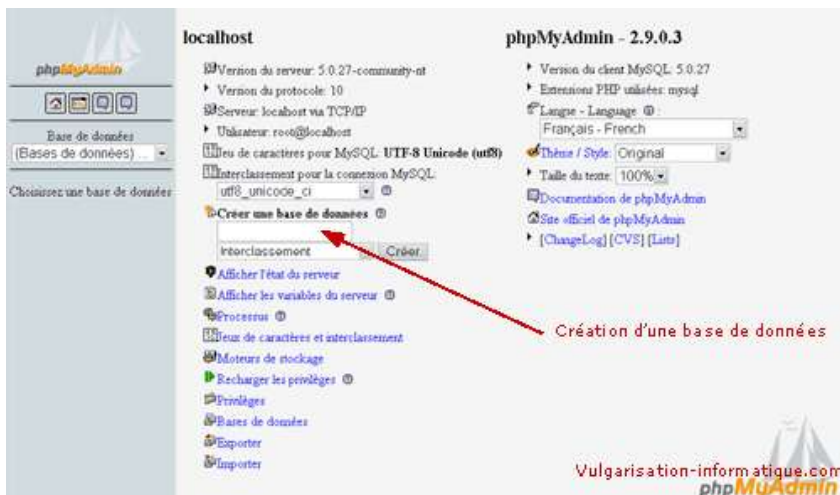
### PHPMyAdmin

PHPMyAdmin est une application PHP qui va vous permettre de gérer vos bases de données. Vous allez pouvoir créer des bases, des tables, etc... sans vous préoccuper du langage SQL qu'il faut écrire. Pour accéder à PHPMyAdmin, tapez dans votre navigateur l'adresse suivante :

<http://localhost/phpmyadmin>

Si vous avez installé EasyPHP, tapez ceci : <http://localhost/mysql/> (le / de fin est important)

Si tout s'est bien passé, vous allez arriver devant un écran de ce style :



Vous allez maintenant pouvoir créer votre base de données. Pour l'exemple, nous allons créer une base de données intitulée **actualites**. Mettez donc **actualites** dans la case **Créer une base de données** et cliquez sur **Créer**. Ne vous préoccupez pas du champ **interclassement**.



La base de données a été créée. Comme vous pouvez le constater, le langage SQL que PHPMyAdmin a utilisé pour créer cette base est **CREATE DATABASE actualites**

Il s'agit de ce qu'on appelle une requête (My)SQL. Nous allons maintenant créer une table **news** qui va contenir des actualités pour notre site Web. Réfléchissez maintenant au nombre de champs que nous pourrions mettre. Tout d'abord, il faut pouvoir identifier chacune des actualités postées. Je vous ai dit plus haut qu'il fallait utiliser un champ de type numérique. On va le nommer **id\_news**. Chaque news aura donc un id différent. Ensuite, il faut que l'on connaisse le titre de l'actualité, sa date de parution, l'auteur de la news et son texte. Cela nous fait donc 5 champs au total. Dans la case **nom**, indiquez **news**, dans la case **nombre de champs**, mettez **5**, puis cliquez sur **Exécuter**.

Champ	Type	Taille/ Valeurs	Entrelassement	Attributs	Null	Defaut	Extra
	VARCHAR				not null		
	VARCHAR				not null		
	VARCHAR				not null		
	VARCHAR				not null		
	VARCHAR				not null		

Vous voici maintenant devant une liste de champs qu'il va falloir déterminer. Le premier champ sera notre champ **id\_news**. Indiquez donc dans la première case **id\_news**. Pour le type de champ, je vous laisse regarder cette page : types de champs MySQL. Après réflexion, vous pourrez avoir une quantité de news assez importante, on va donc utiliser un type **SMALLINT** avec comme attribut **UNSIGNED**. Vous pourrez donc stocker 65536 news. Je pense que c'est correct non ?

Dans le champ **extra** indiquez **autoincrement**. Pourquoi ? car en fait, l'avantage de ce type de champ est que vous allez pouvoir ne pas vous préoccuper du numéro à ajouter, MySQL le fera pour vous. Quand vous ajouterez une news, vous n'aurez donc pas à aller récupérer le numéro maximum existant pour lui ajouter un et l'insérer dans la base. Sélectionnez ensuite la troisième option (icône avec un U), qui indique que ce champ sera un champ ayant une valeur unique (il s'agit également d'un **INDEX** qui va vous permettre d'accélérer les recherches sur la base, mais nous y viendrons plus tard).

Pour le second champ, il s'agit du titre de l'actualité. On peut donc nommer ce champ **titre**, et lui mettre un type **VARCHAR** (mettez **255** dans l'option "taille/valeurs").

Pour le troisième champ, il s'agit d'une date de parution. Vous connaissez ce qu'est un timestamp, nous n'allons pas utiliser ce type de champ car le timestamp de PHP et celui de MySQL sont différents. En revanche, nous allons utiliser un type **INT** avec l'attribut **UNSIGNED**, qui permettra de stocker des dates provenant de la fonction `time()` de PHP.

Pour le quatrième champ, nous allons utiliser le nom **auteur**, qui contiendra le pseudo du posteur de la news, avec le type **VARCHAR** et **20** caractères dans la zone "taille/valeurs". Il faut savoir qu'il serait plus efficace de stocker un numéro correspondant à un numéro de membre, seulement nous n'avons pas de table **membres** et je ne vais donc pas vous compliquer la tâche pour ce premier exercice.

Le dernier champ sera de type **TEXT** et portera le nom de **texte**. Il servira à contenir le texte de la news.

Choisissez dans la zone "moteur de stockage" l'option **MyISAM**. Je ne vous dis pas pourquoi, mais sachez que certains types de tables ne sont pas supportés chez certains hébergeurs, ce qui n'est pas le cas du type **MyISAM**. Ce n'est pas dramatique en soi car on peut changer de type de table via les options de PHPMyAdmin, mais les autres types offrent certaines spécificités que nous ne détaillerons pas ici et qui peuvent ne pas être adaptées. Une fois que tout est rempli, cliquez sur le bouton **Sauvegarder**. Si tout va bien, PHPMyAdmin exécutera la requête suivante :

```
CREATE TABLE `news` (
  `id_news` MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `titre` VARCHAR( 255 ) NOT NULL ,
  `date` INT UNSIGNED NOT NULL ,
  `auteur` VARCHAR( 20 ) NOT NULL ,
  `texte` TEXT NOT NULL ,
  UNIQUE ( `id_news` ) )ENGINE = MYISAM ;
```

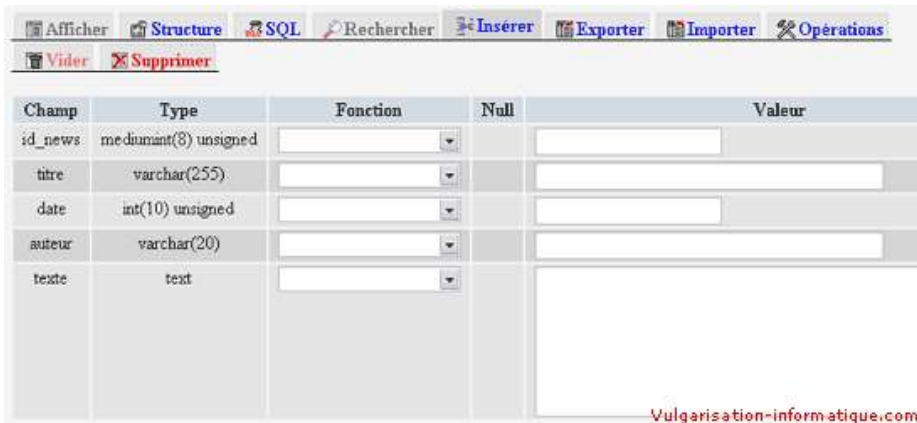
Vous vous retrouvez devant cet écran :

Champ	Type	Entrelassement	Attributs	Null	Defaut	Extra	Action
id_news	MEDIUMINT(8)		UNSIGNED	Non		auto-increment	
titre	VARCHAR(255)			Non			
date	INT(10)			Non			
auteur	VARCHAR(20)			Non			
texte	TEXT			Non			

Maintenant, nous allons insérer des données dans cette table **news**. Pour ce faire, remarquez les onglets présents en haut du cadre de droite :



Cliquez sur l'onglet **Insérer**. Vous obtenez à l'écran une représentation des différents champs avec des zones de texte pour vous permettre de les remplir :



Il va maintenant falloir remplir les champs. Voici ce que vous pouvez mettre :

Tout d'abord dans le champ **id\_news**, ne mettez rien, en effet on a dit que c'était un champ qui se remplissait automatiquement. Pour le titre, à vous de trouver un titre accrocheur :p. En ce qui concerne la date, faites dans un fichier PHP ceci :

```
<?php
echo
time
();
?>
```


Notez la valeur obtenue et renseignez-la dans le champ **date**. Pour le champ **auteur**, indiquez votre pseudo. Mettez ensuite un texte dans la zone de texte correspondant au champ **texte**, ce sera le texte de la news. Cliquez ensuite sur **Exécuter**.



Voilà ce qu'a pu donner la requête :

```
INSERT INTO `news` ( `id_news` , `titre` , `date` , `auteur` , `texte` ) VALUES ( NULL , 'Bienvenue sur mon site !', '1166970760', 'Anthony', 'Ceci est ma première actualité !');
```

Si vous souhaitez maintenant consulter les données présentes dans la table, vous pouvez cliquer sur l'onglet **Afficher**. Voici ce que vous obtenez (j'ai inséré une autre news entre temps pour vous montrer que l'on peut en mettre le nombre qu'on veut) :



The screenshot shows the MySQL database interface. At the top, there is a menu bar with options: Afficher, Structure, SQL, Rechercher, Insérer, Exporter, Importer, and Operations. Below the menu, there are buttons for 'Vider' and 'Supprimer'. The main area displays the SQL query: `SELECT * FROM 'news' LIMIT 0, 30`. Below the query, there are buttons for 'Modifier', 'Expliquer SQL', 'Créer source PHP', and 'Actualiser'. The results section shows two records in a table with columns: id\_news, titre, date, auteur, and texte. The first record has id\_news 1, titre 'Bienvenue sur mon site !', date 1166970760, auteur Anthony, and texte 'Ceci est ma première actualité !'. The second record has id\_news 2, titre 'Autre actualité', date 1166970920, auteur Anthony, and texte 'Ceci est une deuxième actualité !'. Below the table, there are controls for 'Afficher' (set to 30), 'en mode horizontal', and 'et répéter les en-têtes à chaque groupe de 100'. The 'Trier sur l'index' is set to 'aucune' and there is an 'Exécuter' button. At the bottom, there is a footer with the text 'Vulgarisation-informatique.com'.

J'espère que ce chapitre sur PHPMyAdmin et une introduction à MySQL vous ont permis de comprendre les concepts fondamentaux de ce qu'est une base de données, car pour le prochain chapitre, nous allons maintenant parler de requêtes, le tout mélangé avec du code PHP :p

Chapitre suivant



Les bases de données MySQL

Introduction

Connexion au serveur MySQL

Sélection de la base de données

Déconnexion du serveur MySQL

Lire des données dans une table

Source : <http://www.vulgarisation-informatique.com/php-bdd.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)