

Tutoriel “Comment créer une base de donnée MySQL”, par Zaidou Tafridja

1. Les théories de bases de données :

Une base de données est un entité permettant de stocker des données de façon structurée et avec le moins de redondance possible.

Elle contient des renseignements tels que des chiffres, des dates, des mots etc. qui sont stockés dans des tables. Chaque ligne d'une table est un enregistrement.

Le MCD (Modèle Conceptuel des données):

Le modèle conceptuel des données est l'organisation des données. Il structure et renseigne toutes les caractéristiques de chaque type de données qui se trouvent dans la base de données. Chaque entité est lié à une autre, cette association est le lien qui existe entre ses 2 entités, par exemple entre un étudiant et une classe.

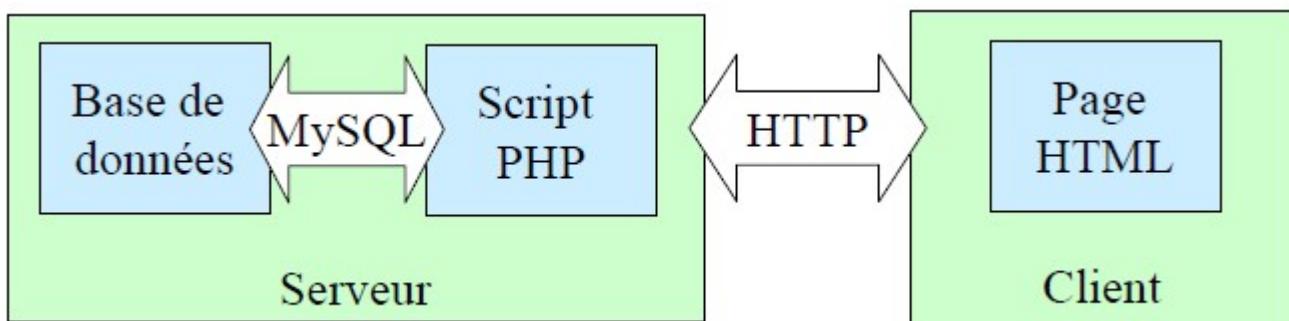
2. Création de la base de données MySQL:

Présentation :

MySQL est un système de gestion de base de données (SGBD). Il dérive de SQL (Structured Query Language) qui est un langage de requête vers les bases de données exploitant le modèle relationnel.

Le serveur de base de données MySQL est très souvent utilisé avec des langages de création sites web dynamiques (exemple: PHP).

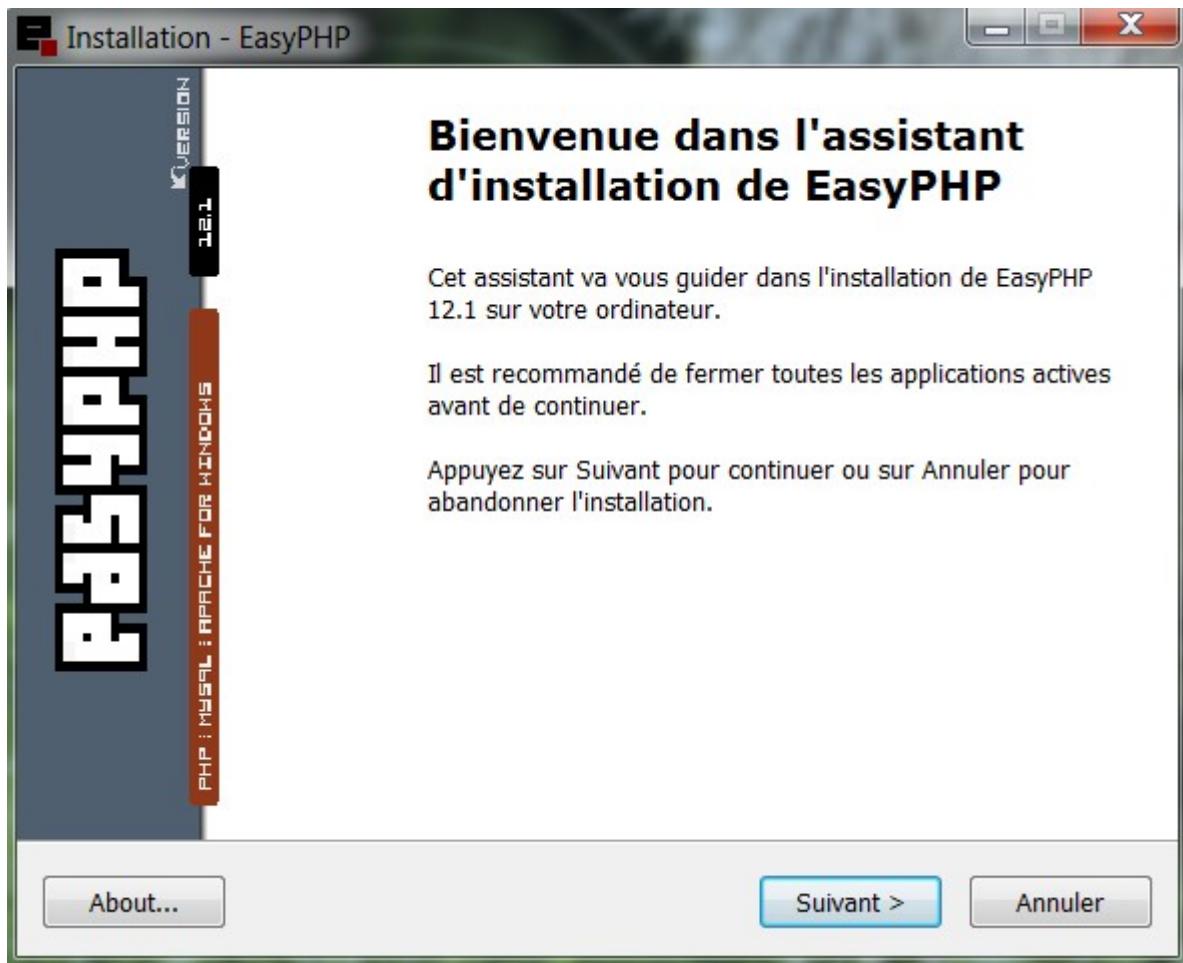
Exemple:



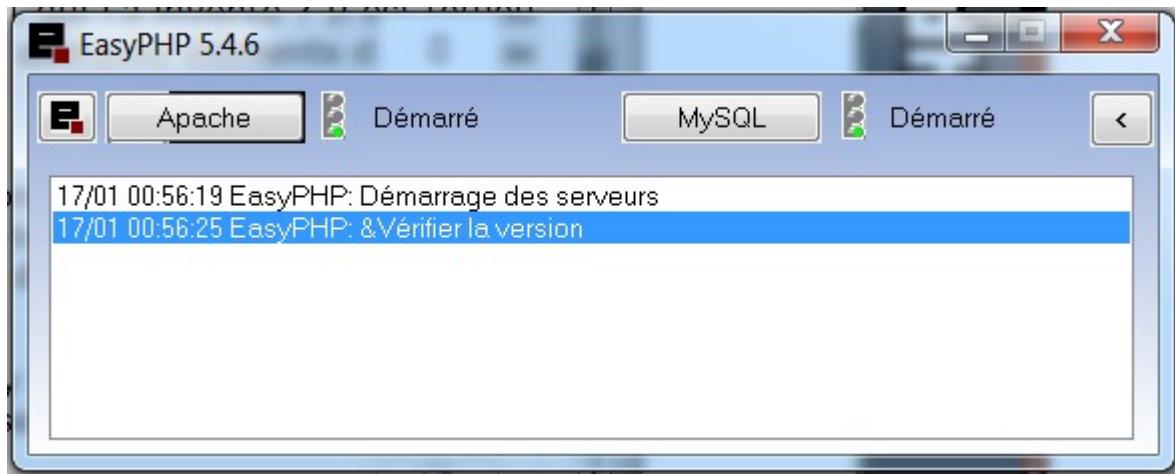
Nous allons pouvoir créer notre première base de données sur un système d'exploitation Windows 7. Pour cela nous avons besoin de mettre en place un serveur de base de données MySQL.

Nous allons installer donc EasyPHP, c'est un environnement de travail comprenant le serveur web Apache, le système de base de données MySQL et le support du langage PHP.

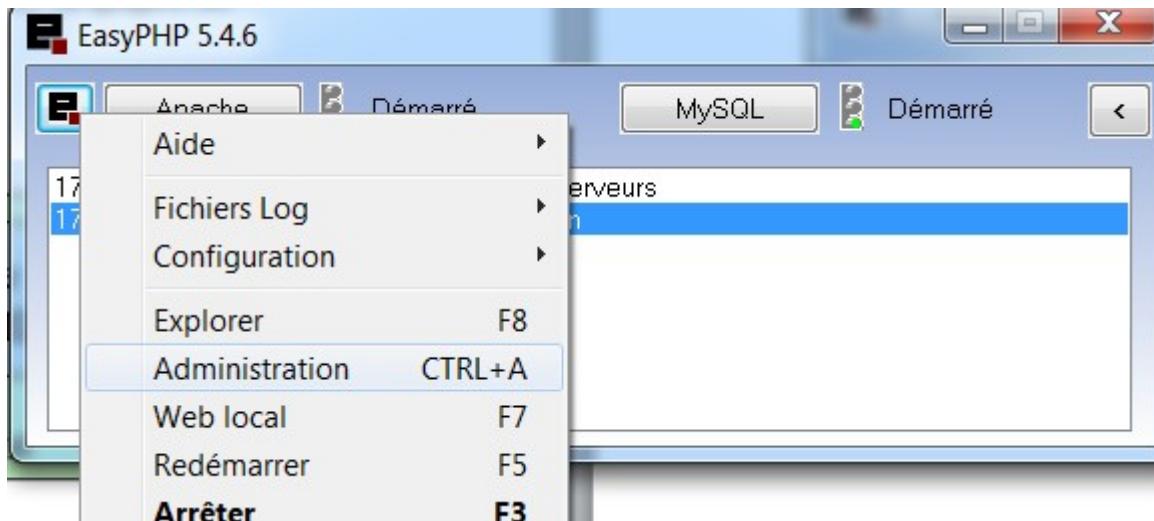
L'avantage de ce logiciel, c'est qu'il fournit phpMyAdmin, une interface permettant de manipuler très facilement les enregistrements de la base de données MySQL.



Une fois l'installation terminé, on exécute le raccourci « easy php.exe » et on se retrouve sur cette fenêtre.



Les 2 serveurs (apache et mysql) sont lancés, cliquez sur « Administration »



Cliquez ensuite sur « ouvrir » l'environnement « phpMyAdmin »

The screenshot shows the phpMyAdmin dashboard. At the top, it displays system information: APACHE 2.4.2, MYSQL 5.5.27, and PHP 5.4.6. Below this are three main sections: 'FICHIERS LOCAUX' (Local Files), 'MODULES' (Modules), and 'FICHIERS PORTABLES' (Portable Files). The 'TESTEUR DE CODE' (Code Tester) section is expanded, showing a code editor with an example PHP script and a 'Interpréter le code' (Run code) button. At the bottom, there are social sharing links and a donation link.

Nous arrivons sur la page d'accueil de « phpMyAdmin »

The screenshot shows the phpMyAdmin configuration page. In the top left, it says "Aucune base de données". The "Paramètres généraux" section has "Interclassement pour la connexion au serveur" set to "utf8_general_ci". The "Paramètres d'affichage" section shows "Langue - Language" as "Français - French" and "Thème" as "pmishome". The "Serveur de base de données" panel lists MySQL server details: "Serveur: 127.0.0.1 via TCP/IP", "Logiciel: MySQL", "Version du logiciel: 5.5.27-log - MySQL Community Server (GPL)", "Version du protocole: 10", "Utilisateur: root@localhost", and "Jeu de caractères du serveur: UTF-8 Unicode (utf8)". The "Serveur web" panel shows "Apache/2.4.2 (Win32) PHP/5.4.8" and "Version du client de base de données: libmysql - mysqlnd 5.0.10 - 20111028 - Sid b1b3e15c893e7fdaeb3aa06b646f0e239f175e39". The "phpMyAdmin" panel lists links: "Version 3.5.2.2, dernière version stable : 3.5.5", "Documentation", "Wiki", "Site officiel", "Contribuer", "Obtenir de l'aide", and "Liste des changements". A message at the bottom left says "Une nouvelle version de phpMyAdmin est disponible et vous devriez songer à une mise à niveau. La version la plus récente est 3.5.5, publiée le 2012-12-20". A warning message at the bottom right says "Votre fichier de configuration fait référence à l'utilisateur «root» sans mot de passe, ce qui correspond à la valeur par défaut de MySQL. Votre serveur MySQL est donc ouvert aux intrusions, et vous devriez corriger ce problème de sécurité!".

Entrez dans le champ le nom de la base de données et cliquez sur créer, pour l'exemple nous allons créer une base de données qui contiendra les informations des étudiants de chaque classe.

The screenshot shows the top navigation bar of phpMyAdmin. The "Bases de données" tab is active and highlighted in blue. Other tabs include "SQL", "État", "Utilisateurs", "Exporter", and "Importer".

Bases de données

The screenshot shows the "Créer une base de données" (Create Database) page. It features a "Créer" button and a dropdown menu with options like "Gestion des etudiants" and "Interclassement". Below the menu, it says "Aucune base de données".

A présent cliquez sur la base de données que vous venez de créer.

The screenshot shows the database list page. It displays a single entry: "gestion des etudiants" which is marked as "Répliqué" (Replicated). There is a "Vérifier les privilèges" (Check privileges) link next to it. At the bottom, there is a "Total: 1" summary and a "Aller à la base de données" (Go to database) button. Below the list, there is a "Tout cocher / Tout décocher Pour la sélection : [checkbox] Supprimer" (Select all / Deselect all for selection: [checkbox] Delete) button.

Entrez dans le champ « Nom » le nom de la première table « étudiant » et entrez « 8 » dans le nombre de colonnes puis cliquez sur « exécuter ». Nous allons voir pourquoi.

Nouvelle table

Nom:	etudiant	Nombre de colonnes:	8
------	----------	---------------------	---

Une fois toutes les informations remplies, cliquez sur sauvegardé.

Nom de la table: etudiant Ajouter 1 colonne(s) Exécuter

Structure

Nom	Type	Taille/Valeurs*	Défaut	Interclassement	Attributs	Null	Index	A_I	Commentaires	Type MIME
id_etudiant	INT	5	Aucune					PRIMARY		
nom	VARCHAR	20	Aucune							
prenom	VARCHAR	20	Aucune							
date_de_naissance	DATE		Aucune							
adresse	VARCHAR	100	Aucune							
code_postal	VARCHAR	5	Aucune							
ville	VARCHAR	50	Aucune							
id_classe	INT	3	Aucune							

Commentaires sur la table: Moteur de stockage: Interclassement:

InnoDB

Définition de PARTITION: Sauvegarder

Dans la première colonne tout à gauche, nous allons rentrer les noms des champs de la table. Le premier champ est généralement l'ID, il permet d'obtenir avec certitude les données de l'enregistrement associé. Il garantit l'intégrité de ce dernier. Les autres champs seront utilisé pour stocker ce que nous avons besoin :

- Le nom de l'étudiant
- Son prénom
- Sa date de naissance
- Son adresse
- Son code postal
- Sa ville

« id_classe » est la clé étrangère de la table classe.

Une clé étrangère est un attribut qui point vers la clé primaire d'une autre table. L'objectif de cette clé étrangère est d'assurer l'intégrité référentielle des données.

Nous créons le champ maintenant et nous établirons par la suite la relation entre la table « etudiant » et la table « classe ». Dans la 2ème colonne se trouve les types de données possible pour chaque champ.

Les principaux types de données dans MySQL :

- Les types numériques
- Les dates et heures
- Les textes et chaînes de caractères

C'est ces 3 types que nous allons d'ailleurs utiliser.

La colonne « Null » permet au champ de contenir des valeurs nulles.

La colonne « index » possède plusieurs valeurs. Nous allons l'utiliser juste une fois et seulement pour l'id_etudiant, comme étant la clé primaire, « **Toute table doit avoir une clé primaire** ».

Nous allons répéter la même procédure pour créer la table « classe », elle est identifiée par un « id_classe » et un nom.

Nos 2 tables sont maintenant créées mais pas encore liées.

Pour créer la relation entre ces 2 tables, cliquez sur la structure de la table « etudiant » puis « vue relationnelle »

The screenshot shows the MySQL Workbench interface with the 'etudiant' table selected. In the 'Structure' tab, the 'id_classe' column is highlighted. Below the table structure, there's a section for creating relationships:

- Tout cocher / Tout décocher Pour la sélection :** with checkboxes for 'Afficher', 'Modifier', 'Supprimer', 'Primaire', 'Unique', and 'Index'.
- Version imprimable**, **vue relationnelle**, **Suggérer des optimisations quant à la structure de la table**, and **Suivre la table**.
- Ajouter** button with a dropdown set to '1'.
- colonne(s)** radio buttons: 'En fin de table' (selected), 'En début de table', and 'Après'.
- id_etudiant** dropdown for the foreign key.
- Exécuter** button.

Enfin sélectionner l'id_classe de la table « classe » pour effectuer la relation et valider.

The screenshot shows the 'relation' tab for the 'etudiant' table. A dropdown menu is open under the 'id_classe' column, listing available foreign keys:

- Aucun i**
- gestion des etudiants.classe.id_classe** (selected)
- gestion des etudiants.etudiant.id_etudiant**

Nous avons terminé la création complète de notre base de données. La relation entre les 2 tables est maintenant effective. Nous allons pouvoir insérer des données. Nous commençons par insérer 2 classes par exemple.

Colonne	Type	Fonction	Null	Valeur
id_classe	int(11)	<input type="text"/>	<input type="button" value="▼"/>	<input type="text"/>
nom	varchar(20)	<input type="text"/>	<input type="button" value="▼"/>	BTS 1 SIO
Exécuter				
<input type="checkbox"/> Ignorer				

Colonne	Type	Fonction	Null	Valeur
id_classe	int(11)	<input type="text"/>	<input type="button" value="▼"/>	<input type="text"/>
nom	varchar(20)	<input type="text"/>	<input type="button" value="▼"/>	BTS 2 SIO
Exécuter				

Ensuite on insère des étudiants, en précisant pour chaque étudiant la classe dont il se trouve. (« id_classe » est la clé étrangère qui point vers la clé primaire de la table « classe », seules les valeurs existantes dans la table « classe » apparaissent ici dans la liste déroulante du champ « id_classe »)

Colonne	Type	Fonction	Null	Valeur		
id_etudiant	int(5)	<input type="text"/>	<input type="button" value="▼"/>	<input type="text"/>		
nom	varchar(20)	<input type="text"/>	<input type="button" value="▼"/>	Zaidou		
prenom	varchar(20)	<input type="text"/>	<input type="button" value="▼"/>	Tafridja		
date_de_naissance	date	<input type="text"/>	<input type="button" value="▼"/>	1992-07-30		
adresse	varchar(100)	<input type="text"/>	<input type="button" value="▼"/>	av de la valeuse		
code_postal	varchar(5)	<input type="text"/>	<input type="button" value="▼"/>	14200		
ville	varchar(50)	<input type="text"/>	<input type="button" value="▼"/>	Hérouville St Clair		
id_classe	int(3)	<input type="text"/>	<input type="button" value="▼"/>	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td></tr><tr><td>2</td></tr></table>	1	2
1						
2						
Exécuter						

Maintenant, grâce au langage SQL nous allons vérifier si notre relation est bien faite.

Exécuter une ou des requêtes SQL sur la base gestion des etudiants: [?](#)

```
1 SELECT *
2 FROM etudiant, classe
3 WHERE etudiant.id_classe = classe.id_classe
4 AND classe.nom = "BTS 1 SIO"
```

Résultat de la requête :

+ Options

id_etudiant	nom	prenom	date_de_naissance	adresse	code_postal	ville	id_classe	id_classe	nom
3	Zaidou	Tafnidja	1992-07-30	av de la valeuse	14200	Hérouville St Clair	1	1	BTS 1 SIO
4	J	Audrey	1992-02-05	monadresse	14000	Caen	1	1	BTS 1 SIO

Conclusion : La base de données est maintenant créée et fonctionnelle, on pourrait ainsi l'utiliser dans une page php.

From:

<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**



Permanent link:

<http://slamwiki2.kobject.net/si3/etudiants/td7/zaidou?rev=1358438646>

Last update: **2019/08/31 14:42**