

## Tutoriel "Comment créer une base de donnée MySQL" par Zaidou Tafriidja

### 1. Les théories de bases de données :Souligné

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 de données conceptuels):

Le modèle de données conceptuels 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.

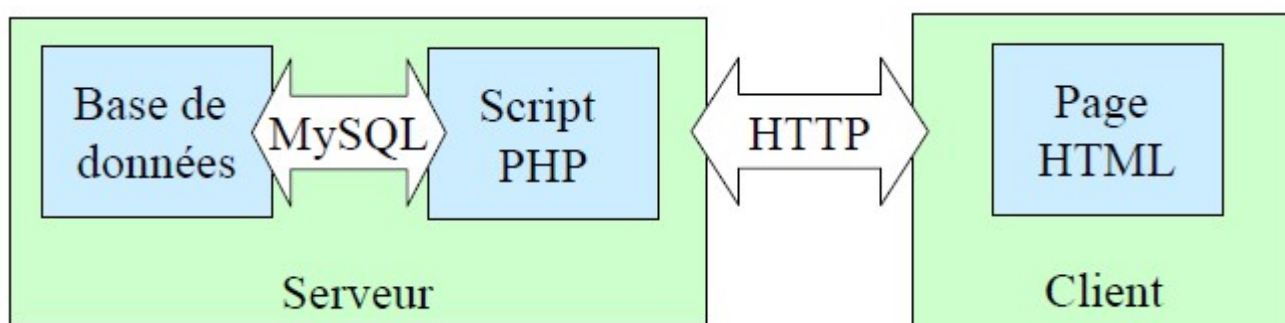
### 2. Création de la base de données MySQL:Souligné

#### 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.

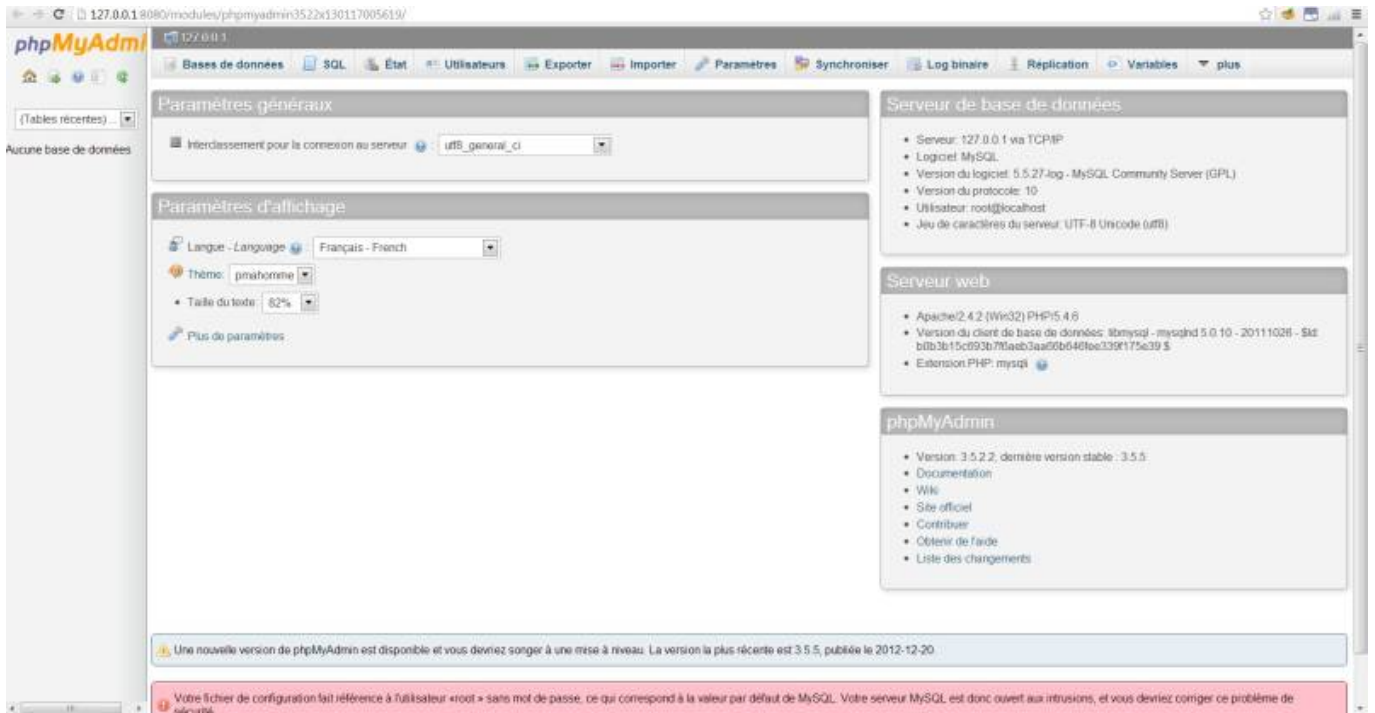


Une fois l'installation terminée, on exécute le raccourci « easy php.exe » et on se retrouve sur cette fenêtre.



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





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.



# Bases de données

**Créer une base de données**

Aucune base de données

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

| Base de données                                | Réplication maître  |
|--|---|
| <input type="checkbox"/> gestion des etudiants | <input checked="" type="checkbox"/> Répliqué <input type="button" value="Vérifier les privilèges"/> |
| <b>Total: 1</b>                                | <input type="button" value="Aller à la base de données"/>   |

Pour la sélection :

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:  Nombre de colonnes:

Une fois toutes les informations remplies, cliquez sur sauvegardé.

Nom de la table:  Ajouter  colonne(s)

| Structure         |         |                   |          |                 |           |                          |         |                                     |              |           |
|-------------------|---------|-------------------|----------|-----------------|-----------|--------------------------|---------|-------------------------------------|--------------|-----------|
| Nom               | Type    | Taille/Valeurs* 1 | Défaut 2 | Interclassement | Attributs | Null                     | Index   | A_I                                 | Commentaires | Type MIME |
| id_etudiant       | INT     | 5                 | Aucune   |                 |           | <input type="checkbox"/> | PRIMARY | <input checked="" type="checkbox"/> |              |           |
| nom               | VARCHAR | 20                | Aucune   |                 |           | <input type="checkbox"/> | ---     | <input type="checkbox"/>            |              |           |
| prenom            | VARCHAR | 20                | Aucune   |                 |           | <input type="checkbox"/> | ---     | <input type="checkbox"/>            |              |           |
| date_de_naissance | DATE    |                   | Aucune   |                 |           | <input type="checkbox"/> | ---     | <input type="checkbox"/>            |              |           |
| adresse           | VARCHAR | 100               | Aucune   |                 |           | <input type="checkbox"/> | ---     | <input type="checkbox"/>            |              |           |
| code_postal       | VARCHAR | 5                 | Aucune   |                 |           | <input type="checkbox"/> | ---     | <input type="checkbox"/>            |              |           |
| ville             | VARCHAR | 50                | Aucune   |                 |           | <input type="checkbox"/> | ---     | <input type="checkbox"/>            |              |           |
| id_classe         | INT     | 3                 | Aucune   |                 |           | <input type="checkbox"/> | ---     | <input type="checkbox"/>            |              |           |

Commentaires sur la table:

Moteur de stockage: InnoDB

Interclassement:

Définition de PARTITION:

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’etudiant - 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 utilisés. 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 créer mais par encore liées. Pour créer la relation entre ces 2 tables, cliquez sur la structure de la table « etudiant » puis « vue relationnelle »

|                          |             |             |                   |     |        |  |  |  |
|--------------------------|-------------|-------------|-------------------|-----|--------|--|--|--|
| <input type="checkbox"/> | 7 ville     | varchar(50) | latin1_swedish_ci | Non | Aucune |  |  |  |
| <input type="checkbox"/> | 8 id_classe | int(3)      |                   | Non | Aucune |  |  |  |

Tout cocher / Tout décocher Pour la sélection :

1 colonne(s)
  En fin de table
  En début de table
  Après

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

The screenshot shows a form with a dropdown menu for the 'id\_classe' field. The dropdown is open, showing two options: 'gestion des etudiants.classe.id\_classe' (highlighted) and '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 exemples.

The first screenshot shows a form with the following fields:

| Colonne   | Type        | Fonction             | Null | Valeur               |
|-----------|-------------|----------------------|------|----------------------|
| id_classe | int(11)     | <input type="text"/> |      | <input type="text"/> |
| nom       | varchar(20) | <input type="text"/> |      | BTS 1 SIO            |

Below the form is an **Exécuter** button and a checkbox labeled  Ignorer.

The second screenshot shows a similar form with the following fields:

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

Below the form is an **Exécuter** button and a checkbox labeled  Ignorer.

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 existants 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="text"/>           |
| nom               | varchar(20)  | <input type="text"/> |      | Zaidou                         |
| prenom            | varchar(20)  | <input type="text"/> |      | Tafridja                       |
| date_de_naissance | date         | <input type="text"/> |      | 1992-07-30                     |
| adresse           | varchar(100) | <input type="text"/> |      | av de la valeuse               |
| code_postal       | varchar(5)   | <input type="text"/> |      | 14200                          |
| ville             | varchar(50)  | <input type="text"/> |      | Hérouville St Clair            |
| id_classe         | int(3)       | <input type="text"/> |      | <input type="text" value="1"/> |

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 étudiants: ?

```

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 | Tafridja | 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 un site web php par exemple.

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

Permanent link: <http://slamwiki2.kobject.net/si3/etudiants/td7/zaidou?rev=1358436980>

Last update: 2019/08/31 14:43

