



## Cours de MySQL

### 1) Introduction

De quoi va-t-on parler ?

Le principe de cette formation ça va être de vous apprendre les bases qu'il y a à connaître sur ce **Système de Gestion de Base de Données**. Base que vous pourrez ensuite utiliser pour créer votre propre site dynamique en Php ou tout autre langage de programmation.

**Prérequis ?**

Pour cette formation, il n'y a **aucun prérequis**, au contraire elle va vous permettre d'acquérir les bases qui vous permettront ensuite d'utiliser **MySQL**.

### 2) Définition

Qu'est-ce que **MySQL** ?

**MySQL** est ce qu'on appelle un **SGBDR** (ou **SGBD**) qui signifie **Système de Gestion de Base de Données Relationnelles**. Derrière ce nom un peu barbare se cache une définition plutôt simple.

MySQL est avant tout un SGBD, c'est un logiciel qui va vous permettre de gérer des bases de données. Dans ces bases de données vous allez pouvoir y stocker des informations, et ensuite pouvoir les modifier ou les récupérer de différentes manières.

En plus on dit que **MySQL** est **relationnel**, tout simplement parce que les données dans MySQL sont stockées sous formes de tableau, aussi appelé table qui pourront être relié. Ces tables vous les connaissez sûrement déjà si vous avez travaillé avec Excel ou n'importe quel tableur.

Globalement une table se définit de deux manières.

Vous avez d'abord les **colonnes** qui sont appelée ici des **champs**, ces colonnes permettent d'identifier les différentes informations que l'on va stocker.

Par exemple si on crée une **table utilisateur**, on aura une colonne **nom**, une colonne **prénom**, une colonne **mail**, une colonne **âge**, une colonne **téléphone**, une colonne **adresse** etc.

id	nom	prenom	mail
1	Dunia	Julien	juliendunia@internet.be
2	Bond	James	jamesbond@cinema.be

Ensuite on va stocker les informations en insérant a chaque fois une nouvelle ligne, dans le tableau. Je parle ici volontairement en tableau et en colonne, mais vous verrez que par la suite on parlera plutôt en **champ** et en **enregistrement**.

Pourquoi MySQL ?

Il y a plusieurs autres SGBD (Oracle, Access, PostgreSQL et plein d'autres). On a choisi **MySQL**, tout simplement parce l'on va choisir de travailler en PHP. Lorsqu'on aura commandé un serveur en générale, il y a de forte chance que MySQL soit déjà installé. C'est un système qui est actuellement très rependu, c'est à dire que la plupart des serveurs en sont équipés. Du coup en apprenant MySQL c'est l'assurance qu'on va pouvoir utiliser nos connaissances lorsque l'on va créer notre site dynamique. **C'est un système très utiliser du coup il y a beaucoup plus de documentations et informations sur internet.** Pareil pour tous les problèmes, vous pourrez trouver facilement une solution, vu que quelqu'un aura de grande chance d'avoir déjà eu ce problème avant vous.

### 3) Installation

On va installer un logiciel qui va permettre d'avoir un server web préconfigurer avec MySQL qui sera directement relié à PHP. Il existe plusieurs logiciels pour **Windows (wamp, xamp, easyphp** etc.), pour **Linux (lamp)** et pour **mac (mamp** version gratuite). Vous pouvez utiliser celui que vous voulez mais nous ici on va **installer Wamp**. Pour ce faire aller dans votre navigateur et taper **WampServer**. Quand vous êtes sur le site, aller dans téléchargement et choisissez votre version en fonction de votre configuration.

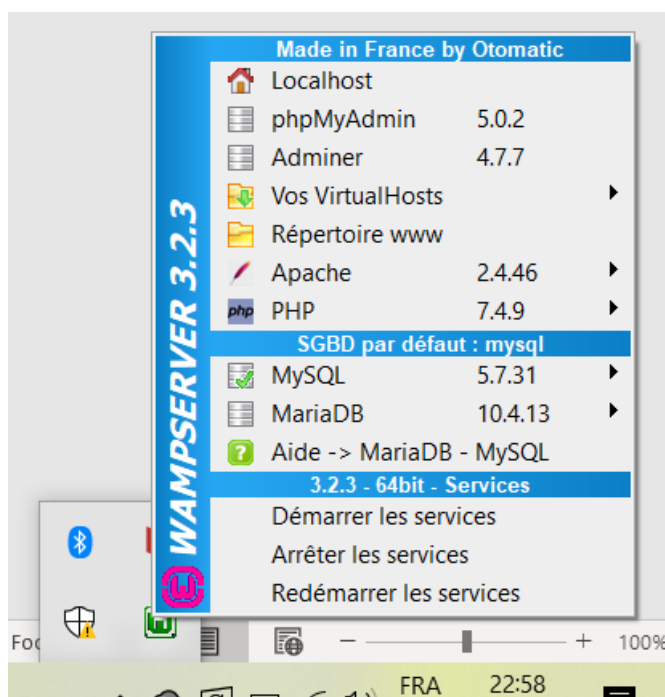


Une fois télécharger, avant l'installation il va vous falloir télécharger toutes les versions **Microsoft Visual C++ en x64 et x86** disponible ici (**2010 à maintenant**) :

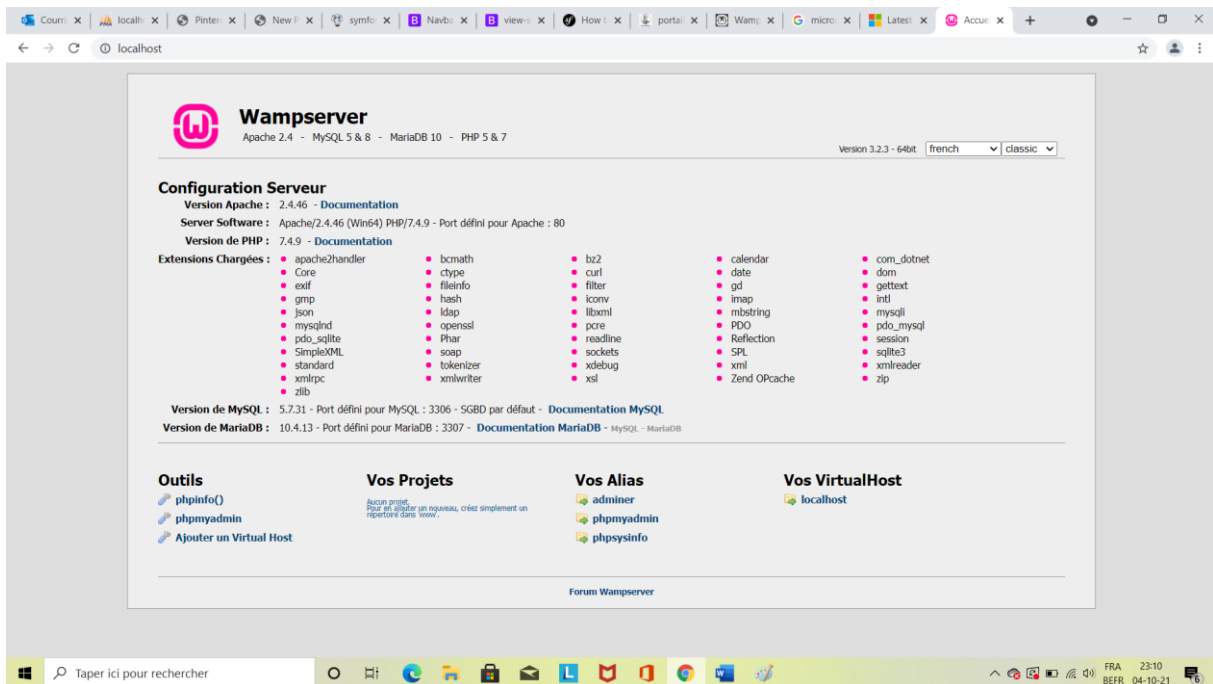
<https://docs.microsoft.com/fr-FR/cpp/windows/latest-supported-vc-redist?view=msvc-160>

Installer tous les fichiers, ensuite redémarrer votre ordinateur. Puis installer Wamp.

Une fois installer lancer **wampserver64**, et vérifier qu'il est bien lancé en bas à droite, dans votre barre de tâche.

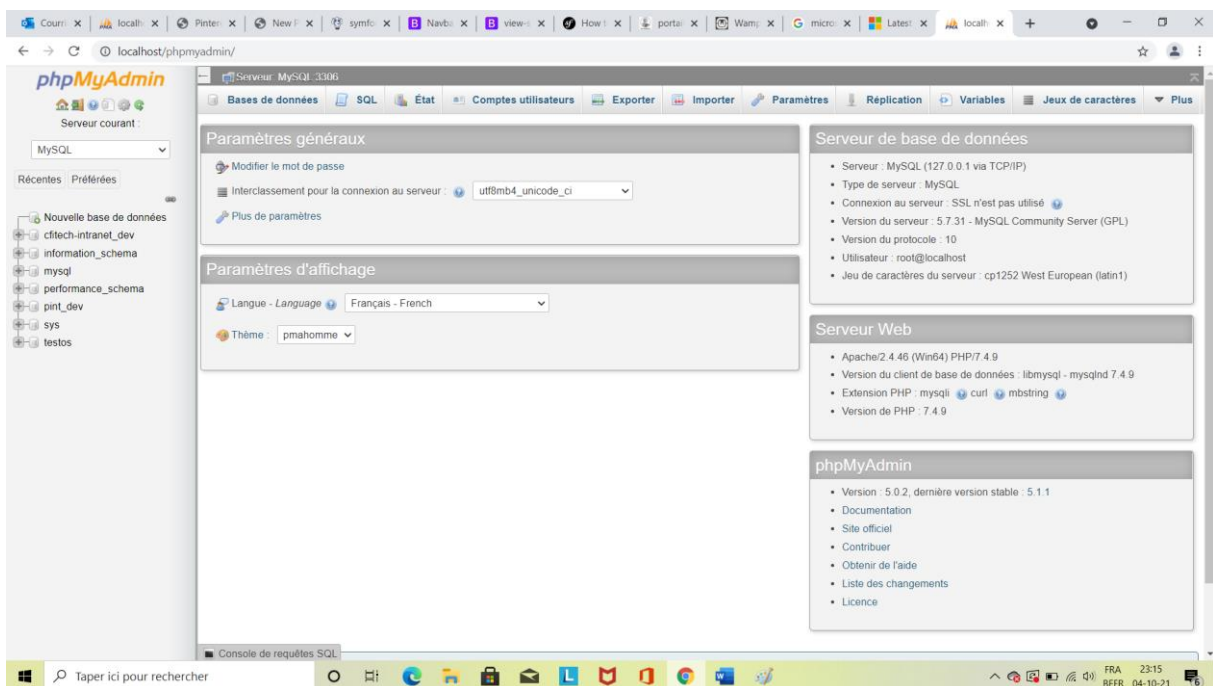


Une fois que vous avez ce résultat ci-dessus. Ouvrez votre navigateur et taper **localhost/**. Vous devriez tomber sur une page d'accueil qui correspond à la version et au logiciel que vous avez choisi.



Vous aurez accès à **phpMyAdmin** via cette page (il est dans outils en bas à gauche sur la page). Ou sinon vous pouvez taper directement dans le navigateur **localhost/phpmyadmin**.

Vous devriez tomber sur une interface qui ressemble à cela :



## 4) PhpMyAdmin

Qu'est que **PhpMyAdmin** ?

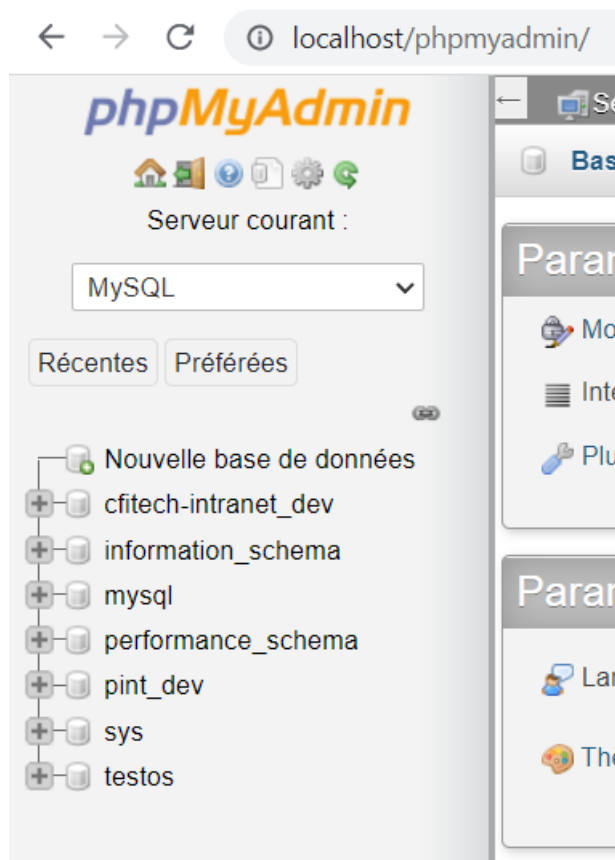
C'est un **outil qui a été développé en PHP** et qui va vous permettre de **gérer votre base de données**. On peut aussi gérer la base de données sans passer par phpMyAdmin, en utilisant les lignes de commandes mais pour débiter il est plus agréable et pratique d'avoir une interface visuelle pour gérer la base de données. Je vous montrerai aussi la manière avec ligne de commande.

Pourquoi PhpMyAdmin ?

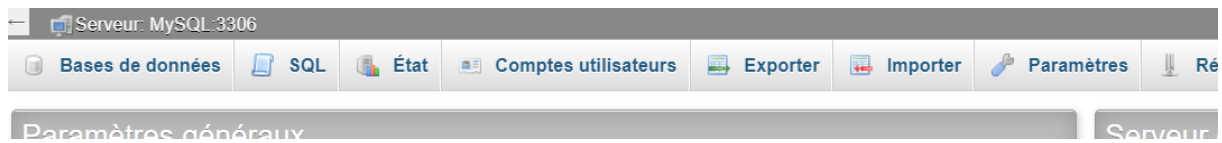
Tout simplement par ce qu'il est extrêmement répandu aussi. Quelque soit le serveur que vous allez commander, le serveur aura de forte chance d'être équipé de phpMyAdmin.

L'interface fonctionne de la manière suivante.

A gauche vous avez un petit panneau de navigation, où il y a les bases de données sur votre serveur. Vous avez des bases de données par défaut qui sont la (performance\_schema, sys, mysql etc). Ce sont des variables de configuration, vous n'aurez jamais à y toucher.



Ensuite vous avez les différents onglets d'action, on va en reparler au fur et à mesure.



Et pour finir on a tout le contenu qui changera suivant la page sur laquelle on est.



On peut voir ici que dans paramètres d'affichage qu'on peut changer de thème, à vous de choisir lequel vous plait bien.

## 5) Création de notre base de données

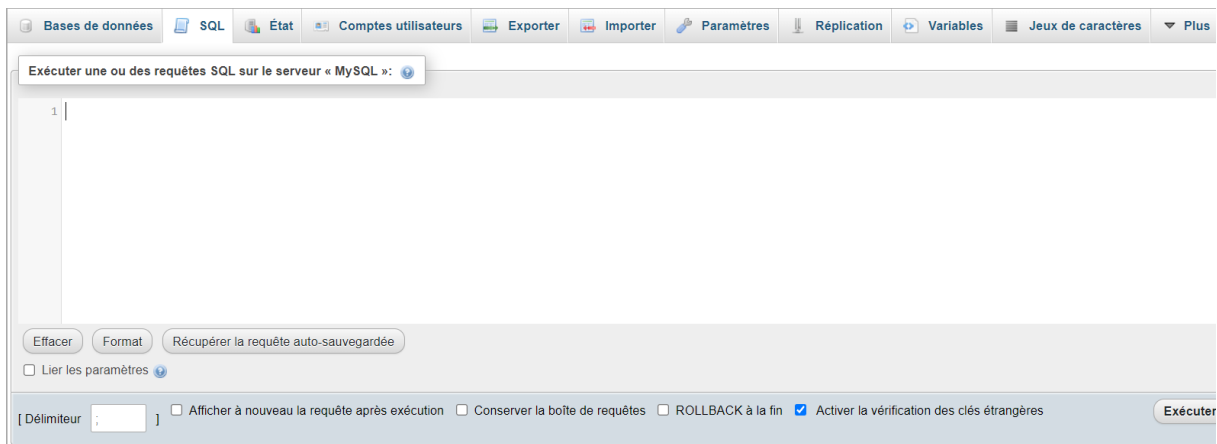
Maintenant que nous avons tout bien installer, on va **créer** notre première **base de données**. C'est cette base de données qui va nous permettre par la suite de stocker les tables qui vont permettre d'avoir les différentes informations de notre site web.

Pour créer une base de données, c'est très simple. Ici on va utiliser l'interface phpMyAdmin, mais il faut savoir que toutes les actions qu'on va faire sous cette interface vont être traduit dans le **langage SQL**, qui permet de communiquer avec le serveur. Au fur et à mesure on verra les équivalents et comment on fait les opérations directement en utilisant les **commandes SQL**.

Qu'est-ce que SQL ?

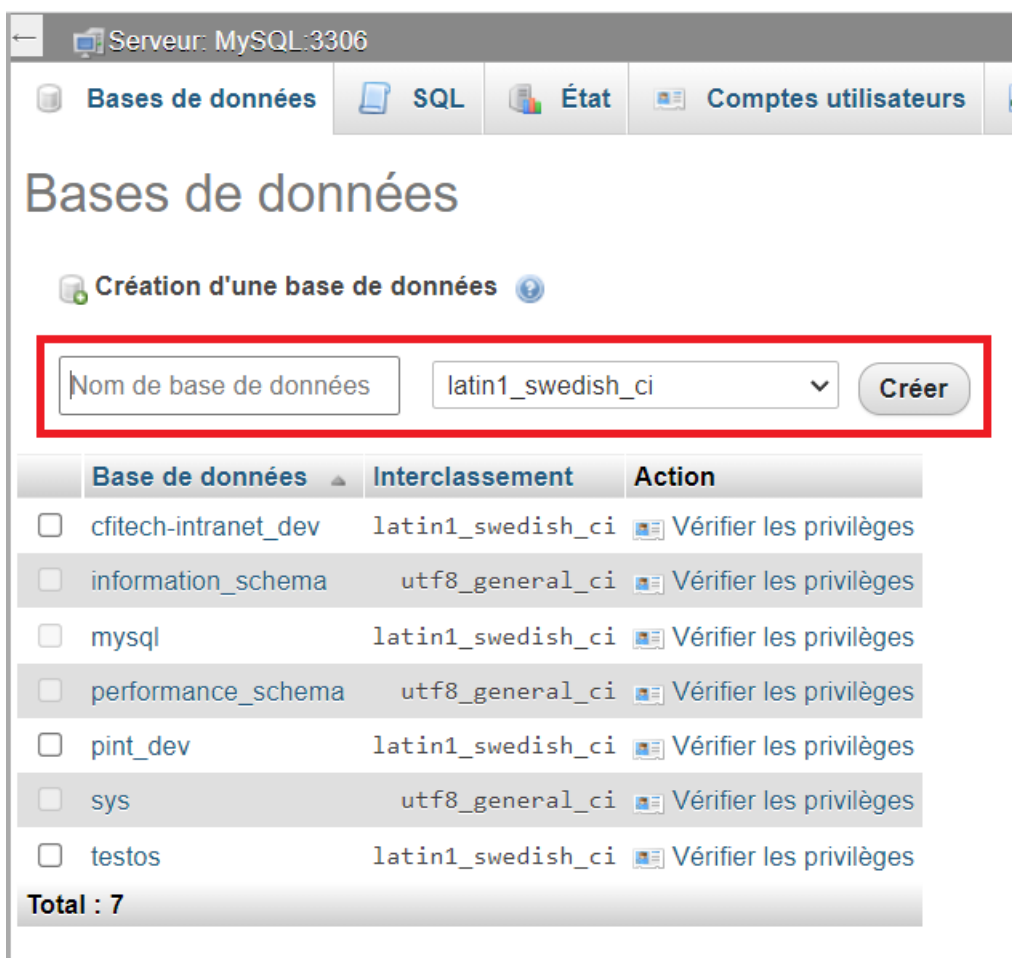
Le langage **SQL (Structured Query Language)** est un langage informatique utilisé pour exploiter des bases de données. Il permet de façon générale la définition, la manipulation et le contrôle de sécurité de données. ... Il est bien supporté par la très grande majorité des systèmes de gestion de base de données (SGBD).

Pour ouvrir les commandes SQL il suffit de **cliquer** sur **l'onglet SQL**. C'est là qu'on écrira les différentes commandes SQL (**requêtes SQL**).



The screenshot shows the 'SQL' tab selected in the top navigation bar. Below the navigation bar, there is a text area for entering SQL queries. At the bottom, there are buttons for 'Effacer', 'Format', and 'Récupérer la requête auto-sauvegardée'. There is also a checkbox for 'Lier les paramètres' and a row of checkboxes for execution options: 'Afficher à nouveau la requête après exécution', 'Conserver la boîte de requêtes', 'ROLLBACK à la fin', and 'Activer la vérification des clés étrangères' (which is checked). An 'Exécuter' button is on the far right.

Pour créer la base de données allez dans la **page d'accueil** de **phpMyAdmin**, vous cliquez sur **l'onglet Base de Données**.

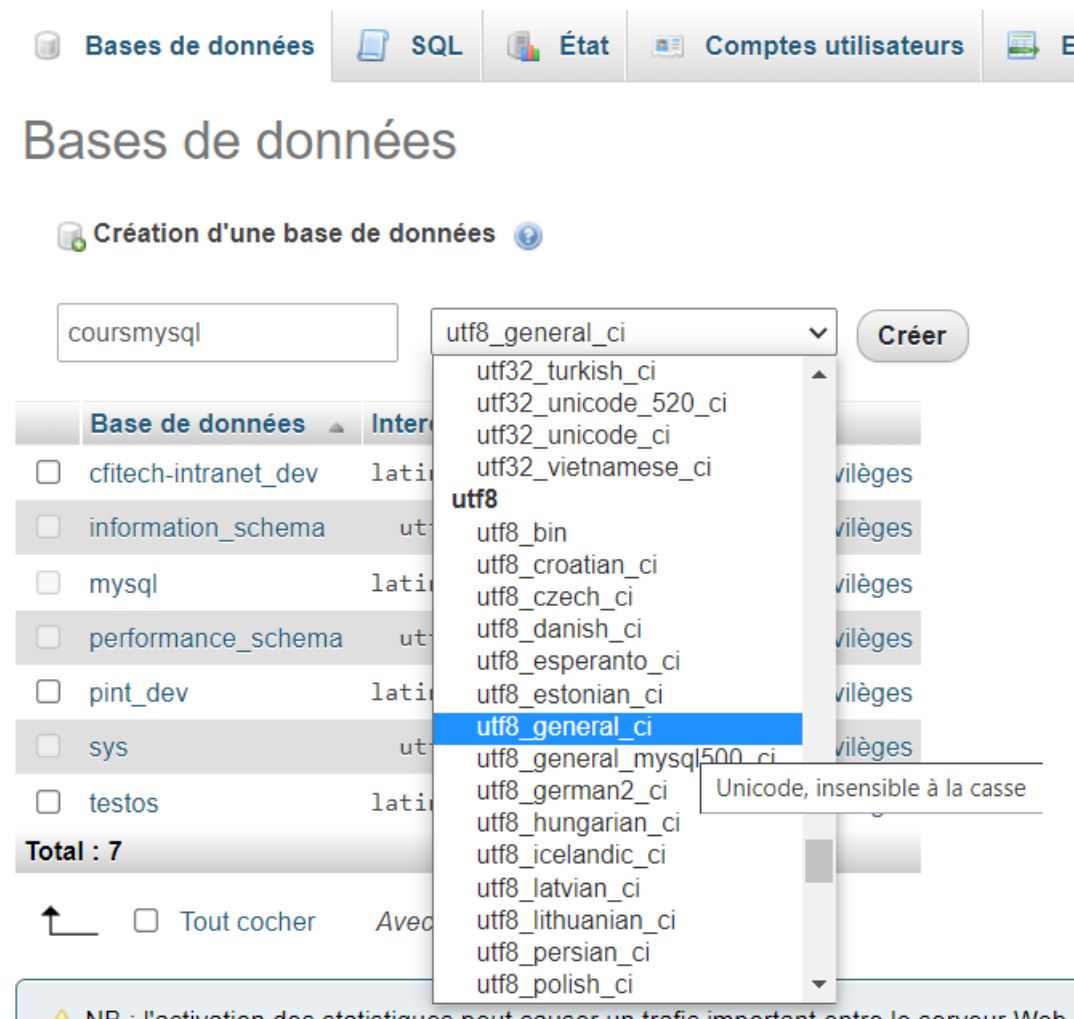


The screenshot shows the 'Bases de données' (Databases) page. At the top, there is a navigation bar with 'Bases de données' selected. Below it, the title 'Bases de données' is displayed. Underneath, there is a link for 'Création d'une base de données'. A red rectangle highlights the form for creating a new database, which includes a text input for 'Nom de base de données', a dropdown menu for 'latin1\_swedish\_ci', and a 'Créer' button. Below this, there is a table listing existing databases.

Base de données	Interclassement	Action
<input type="checkbox"/> cfitech-intranet_dev	latin1_swedish_ci	<a href="#">Vérifier les privilèges</a>
<input type="checkbox"/> information_schema	utf8_general_ci	<a href="#">Vérifier les privilèges</a>
<input type="checkbox"/> mysql	latin1_swedish_ci	<a href="#">Vérifier les privilèges</a>
<input type="checkbox"/> performance_schema	utf8_general_ci	<a href="#">Vérifier les privilèges</a>
<input type="checkbox"/> pint_dev	latin1_swedish_ci	<a href="#">Vérifier les privilèges</a>
<input type="checkbox"/> sys	utf8_general_ci	<a href="#">Vérifier les privilèges</a>
<input type="checkbox"/> testos	latin1_swedish_ci	<a href="#">Vérifier les privilèges</a>
Total : 7		

Là vous lui donnez un nom, de préférence un nom qui correspond à votre site web. Je vais l'appeler pour cette formation : **coursmysql**. Et avant de créer on doit choisir la table de caractère qui sera utilisé par votre base de données.

Le principal problème c'est que dans le monde il existe tout un tas de langage et chaque langage n'utilise pas forcément la même table de caractères. Par exemple aux Etats-Unis en anglais il n'a pas d'accent comme dans le langage français. Donc pour régler ce problème on choisit l'**UTF8 Générale** qui est la table la plus large avec le plus de caractères.



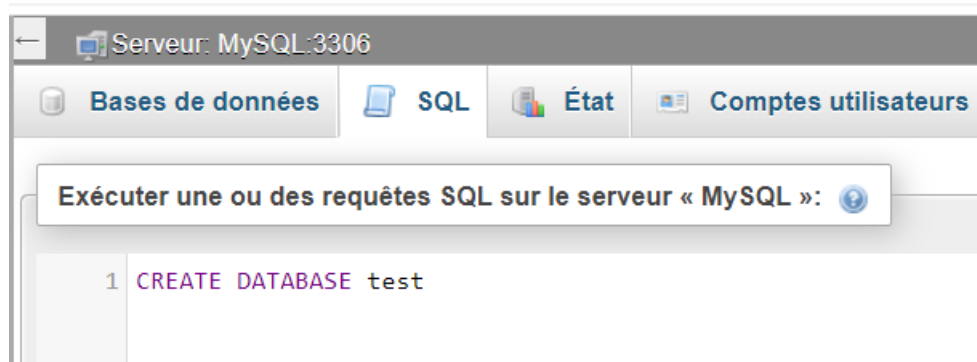
Le « **ci** » indique juste **Case Insensitive** qui signifie qu'au niveau de la table, une majuscule ou une minuscule aura la même valeur.

Et voilà vous avez créé votre première base de données. Vous pourrez voir à gauche dans les bases de données qu'on y retrouvera cette fois ci « **coursmysql** ».

Là c'est phpMyAdmin qui a fait la magie pour nous et qui a donc créé la base de données. Comment faire si on n'a pas cette interface-là. On peut utiliser ce qu'on appelle les requêtes SQL et le langage SQL. Toutes les opérations que l'on va faire dans phpMyAdmin, on peut les écrire à travers une phrase qui s'appelle une **requête SQL**.



Ici on peut cliquer sur l'onglet **SQL**. Pour créer une base de données en SQL, il suffit de taper :



- **CREATE DATABASE test**

Ensuite **appuyer** sur **exécuter**, et vous verrez que cette **requête/commande** a donc créé la base de données « **test** ».

Ici vous remarquerez qu'on n'a pas précisé d'encodage pour cette base de données, on n'a pas précisé qu'on voulait **utf8\_generale\_ci**.

Pour y remédier on clique sur la base de données « **test** », une fois dedans on clique sur l'onglet **SQL**, cette fois ci il va lancer le SQL propre à cette base de données. Ensuite on va exécuter une requête :



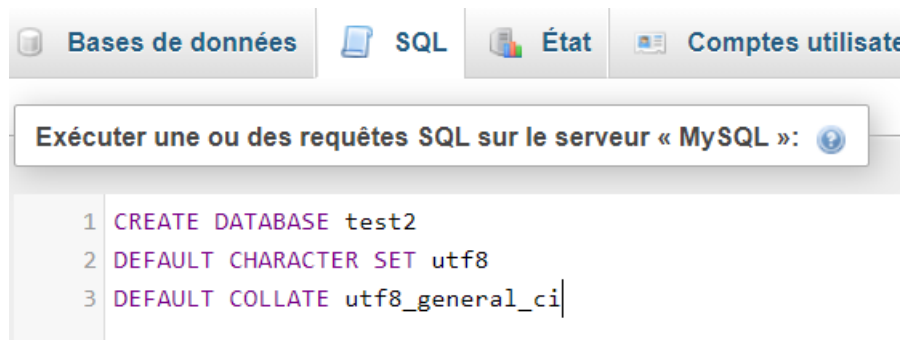
- **SHOW VARIABLES**

Elle va nous montrer toutes les variables qui correspondent à cette base de données.

On verra ici qu'il a choisi par défaut le langage latin1 :

character_set_client	utf8mb4
character_set_connection	utf8mb4
character_set_database	latin1
character_set_filesystem	binary
character_set_results	utf8mb4

Pour créer une base de données avec comme langage **utf8mb4\_general\_ci** en SQL il faut faire donc la commande :



- **create database test2**  
**DEFAULT CHARACTER SET utf8**  
**DEFAULT COLLATE utf8mb4\_general\_ci**

La maintenant quand on est dans **test2** et qu'on fait « **show variables** », on pourra voir que **character\_set\_database** est sur **utf8**.

A noter que quand on écrit en SQL on peut mettre en majuscule ou minuscule c'est pareil. Mais par convention on choisit d'écrire en majuscule pour tout simplement séparer les parties variables des parties qui sont fixes.

## 6) Suppression Base de Données

On a créé trop de base de données donc on va en supprimer certaines. Pour **supprimer** une base de données en SQL tout d'abord il suffit d'ouvrir SQL et de taper cette requête-là :



Pour le faire via l'interface **phpMyAdmin**, il suffit de cliquer sur l'**onglet base de données**, de **cocher** la base de données **qu'on veut supprimer**, ensuite **d'appuyer** sur **supprimer**, on aura un message de confirmation avec d'ailleurs la **requête SQL de suppression** comme vu ci-dessus :

The screenshot shows the phpMyAdmin interface with the 'Bases de données' tab selected. A table lists various databases, including 'test2', which is selected with a checkbox. A 'Confirmer' dialog box is open, asking for confirmation to delete the database 'test2' using the SQL command 'DROP DATABASE `test2` ;'. The 'Supprimer' button is also highlighted.

Base de données	Interclassement	Action
<input type="checkbox"/> cfitech-intranet_dev	latin1_swedish_ci	Vérifier les privilèges
<input type="checkbox"/> coursmysql	utf8_general_ci	Vérifier les privilèges
<input type="checkbox"/> information_schema	utf8_general_ci	Vérifier les privilèges
<input type="checkbox"/> mysql	latin1_swedish_ci	Vérifier les privilèges
<input type="checkbox"/> performance_schema	utf8_general_ci	Vérifier les privilèges
<input type="checkbox"/> pint_dev	latin1_swedish_ci	Vérifier les privilèges
<input type="checkbox"/> sys	utf8_general_ci	Vérifier les privilèges
<input checked="" type="checkbox"/> test2	utf8_general_ci	Vérifier les privilèges
<input type="checkbox"/> testos	latin1_swedish_ci	Vérifier les privilèges

Total : 9

Avec la sélection : **Supprimer**

**Confirmer**  
Une base de données est sur le point d'être DÉTRUITE ! Faut-il vraiment exécuter « DROP DATABASE `test2` ; » ?  
**OK** **Annuler**

**Attention** c'est irréversible, une fois supprimer la base de données est perdu.

## 7) Création de Tables

Maintenant qu'on est en mesure de créer une base de données, nous sommes en mesure de créer nos premières tables.

Qu'est-ce qu'une table ?

Comme je vous l'ai dit plus haut, les tables vont nous permettre de stocker les différentes entités de notre application.

On va créer deux tables dans notre base de données « **coursmysql** » :

Une **table users** :

- **firstname VARCHAR 255**
- **lastname VARCHAR 255**
- **gender ENUM('M','F','X')**
- **date\_of\_birth DATE**

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 <b>firstname</b>	varchar(255)	utf8mb4_general_ci		Non	Aucun(e)			 Modifier  Supprimer  Plus
<input type="checkbox"/>	2 <b>lastname</b>	varchar(255)	utf8mb4_general_ci		Non	Aucun(e)			 Modifier  Supprimer  Plus
<input type="checkbox"/>	3 <b>gender</b>	enum('M', 'F', 'X')	utf8mb4_general_ci		Non	Aucun(e)			 Modifier  Supprimer  Plus
<input type="checkbox"/>	4 <b>date_of_birth</b>	date			Non	Aucun(e)			 Modifier  Supprimer  Plus

## Exos

### 1) Une table articles :

- **article\_name VARCHAR 255**
- **description TEXT**
- **creation\_date DATE**

On va commencer par créer la table **Users**, c'est ultra simple. Il suffit d'aller dans **phpMyAdmin** et de **cliquer** sur le nom de **votre base de données**. Ensuite de donner le nom de la table donc « **users** » ainsi que le nombre de colonnes, ici on aura **4 colonnes** car on a **4 champs**.



Après exécution, vous tomberez sur une page avec les paramètres à remplir pour chaque champ que l'on va créer.

On va tout d'abord mettre le nom du champ, puis le type. Vous verrez qu'il y a beaucoup de type sur MySQL qui permettent de stocker différentes informations.

Les **types Numérique** qui permettent de stocker des entiers, des décimaux, des booléens, ...

La principale différence entre les 5 types d'entier (**INT**), c'est juste une différence de taille. Un **TINYINT** sera stocké sur un seul octet, du coup le nombre de possibilité est limité. Il sera sur 2 puissances 8, soit 256 possibilités. Du coup les valeurs stockées avec ce format-là seront petite et prendrons moins de place. Elles iront de -128 à 127.

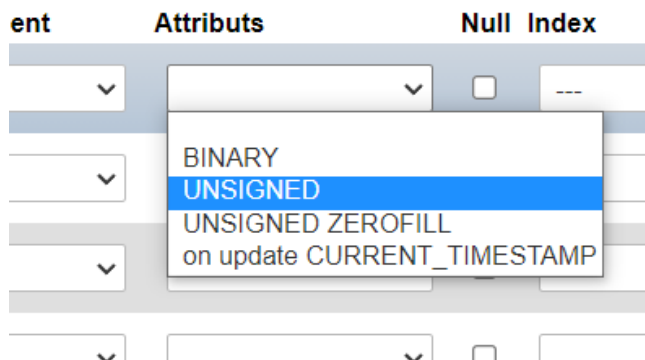
Le **SMALLINT** ce sera de - 32768 à 32767, c'est 2 puissances 16.

Le **MEDIUMINT** ce sera 2 puissances 32.

Le **INT** ce sera 2 puissances 64.

Le **BIGINT** ce sera 2 puissances 128.

Si on ne veut pas prendre en compte les valeurs négatives pour un champ par exemple un âge, il suffit d'aller au niveau d'attribut et sélectionner **UNSIGNED**.



En choisissant **UNSIGNED** ça veut dire que votre **valeur n'a pas de signe**. En gros par exemple pour un **TINYINT** qui allait de base de -128 à 127, vu qu'on enlève toutes les valeurs négatives, quand il sera **UNSIGNED** il ira de 0 à 255.

Ensuite on a **DECIMAL**, **FLOAT**, **DOUBLE** et **REAL** qui vous permettent de stocker des valeurs **décimales** (donc avec la virgule), et les différentes formes influent sur la taille et le nombre de décimales que vous allez pouvoir sauvegarder.

On a pour finir **BIT**, **BOOLEAN**, **SERIAL**.