

Initiation SQL

Kézako Michou ???

La SQL (Structured Query Language) est un langage de programmation servant à exploiter des bases de données (BDD).

Et puis ?

Comme pour Wordpress, nous devons utiliser un serveur pour pouvoir abriter notre BDD 😊.
Je vous laisse deviner ? 🤔 ... WampServer

WampServer : Où j'aurais pu dire XAMPP ou MAMP (pour MAC), est un environnement web intégrant plusieurs serveurs (Apache, MySQL & MariaDB) et un interpréteur de script PHP fonctionnant localement sur la machine.

Lien : <http://wampserver.aviatechno.net/?lang=fr>

Pour son installation, référez vous au cours WordPress.



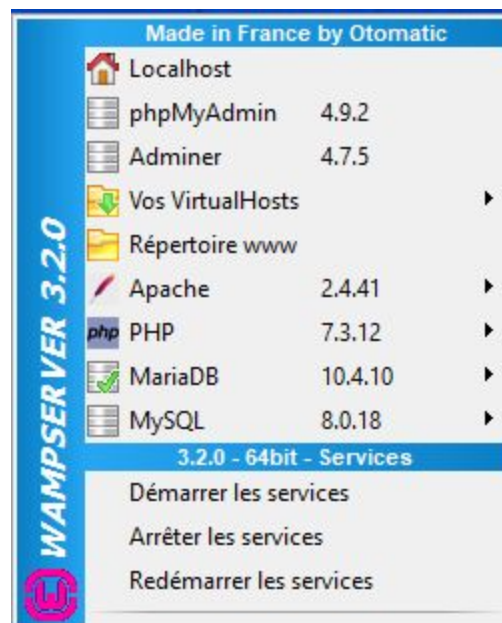
Une fois installé, Wamp se glisse dans la barre des tâches / notifications en bas à droite et doit être vert.

Via clique gauche, vous retrouvez ces informations,

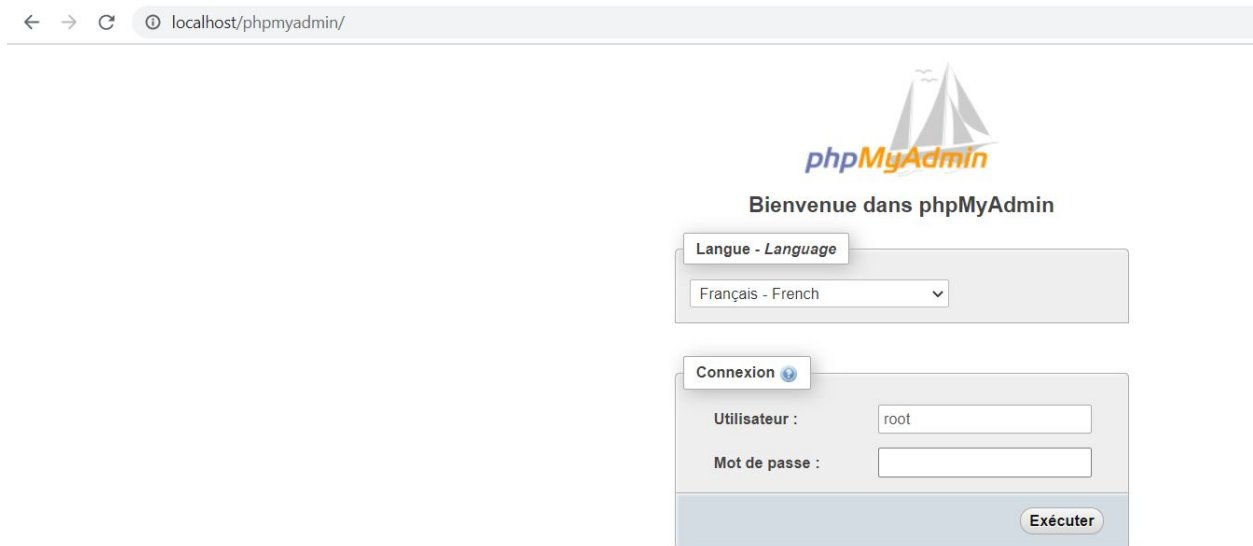
phpMyAdmin : Est l'accès aux bases de données de vos sites web. Par défaut, l'identifiant est **root** est il n'y a pas de mot de passe.

Pour accéder à PhpMyAdmin, il existe deux solutions :

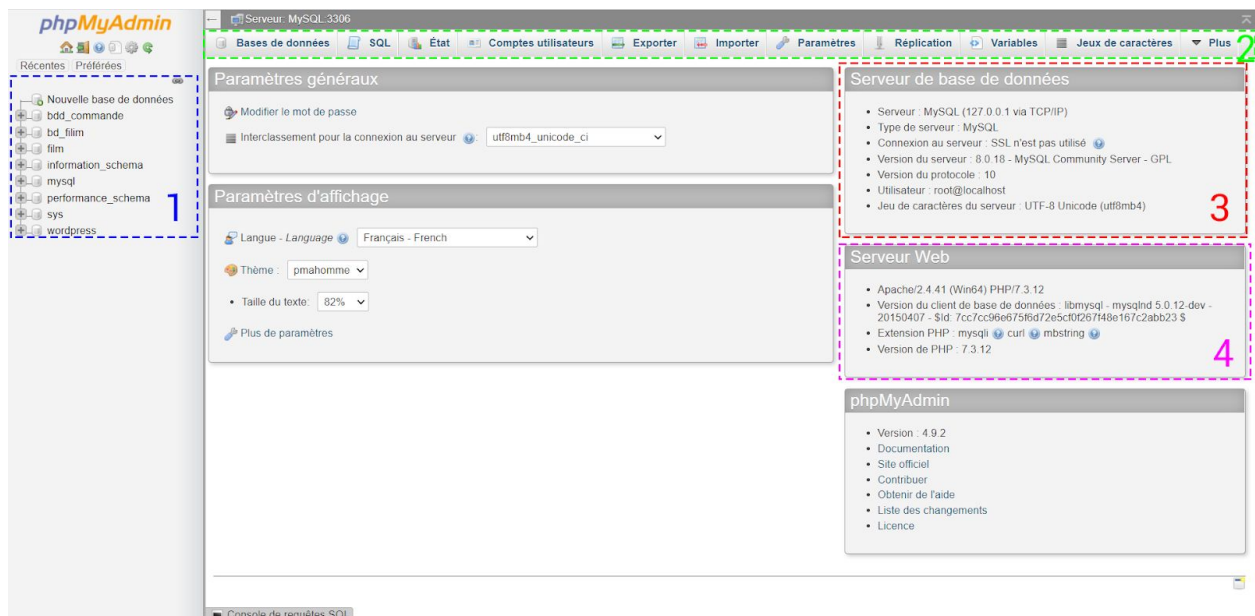
PS : Lors de son utilisation sur un environnement professionnel, ne pas laisser les accès par défaut.



1. Vous cliquez tout simplement sur phpMyAdmin (Rien de compliqué ...), cela ouvrira phpMyAdmin dans le navigateur que vous aurez configuré par défaut lors de l'installation de Wamp.
2. Vous ouvrez votre navigateur préféré (Et j'ose espérer que ce n'est pas InternetExplorer) et vous entrez cette adresse : <http://localhost/phpmyadmin/>



L'interface



1. La liste de vos bases de données
2. Les paramètres de phpMyAdmin
3. Les informations du serveur de BDD (Bien vérifié que vous utilisez MySQL et non MariaDB)
4. Les informations du serveur web

Créer une BDD

Pour créer une base de donnée, c'est comme le Port-Salut tout est marqué dessus.

- Cliquez sur "Nouvelle Base de données"
- Entrez son nom
- Sélectionner l'encodage de caractères (je vous recommande **utf8_general_ci**)
- Cliquez sur Créer

Bases de données

Création d'une base de données :

maBaseDeDonnee utf8_general_ci **Créer**

Base de données	Interclassement	Action
<input type="checkbox"/> bdd_commande	utf8_general_ci	Vérifier les privilèges
<input type="checkbox"/> bd_filim	utf8_general_ci	Vérifier les privilèges
<input type="checkbox"/> film	utf8mb4_0900_ai_ci	Vérifier les privilèges
<input type="checkbox"/> information_schema	utf8_general_ci	Vérifier les privilèges
<input type="checkbox"/> mysql	utf8mb4_0900_ai_ci	Vérifier les privilèges
<input type="checkbox"/> performance_schema	utf8mb4_0900_ai_ci	Vérifier les privilèges
<input type="checkbox"/> sys	utf8mb4_0900_ai_ci	Vérifier les privilèges
<input type="checkbox"/> wordpress	utf8_general_ci	Vérifier les privilèges
Total: 8	utf8mb4_0900_ai_ci	

↑ ☐ Tout cocher Avec la sélection : Supprimer

Félicitations, vous venez de créer votre première base de donnée 🤖 (like a boss). Mais vous allez me dire qu'elle est bien vide. Effectivement, pour l'instant nous n'avons rien à y mettre, mais cela ne va pas durer !

Remplir ma BDD

Pour la remplir, il existe comme partout dans le numérique, plusieurs méthodes :

- Via langage SQL directement sur phpMyAdmin
- Via la souris
- Via un fichier .sql

Le langage SQL (1^{ere} partie)

- Les Types

Comme tout langage, lorsque vous stockez une information en mémoire, il faut indiquer son type.

Type	Date par défaut ("zéro")
DATE	'0000-00-00'
DATETIME	'0000-00-00 00:00:00'
TIME	'00:00:00'
YEAR	0000
TIMESTAMP	0000000000000000

Type	Nombre d'octets	Minimum	Maximum
TINYINT	1	-128	127
SMALLINT	2	-32768	32767
MEDIUMINT	3	-8388608	8388607
INT	4	-2147483648	2147483647
BIGINT	8	-9223372036854775808	9223372036854775807

Texte	CHAR(5)	Mémoire requise	VARCHAR(5)	Mémoire requise
"	' '	5 octets	' '	1 octet
'tex'	'tex '	5 octets	'tex'	4 octets
'texte'	'texte'	5 octets	'texte'	6 octets
'texte trop long'	'texte'	5 octets	'texte'	6 octets

Type	Ecriture	Signification
DECIMAL	DECIMAL(5,3)	5 chiffres au total dont 3 derrière la virgule
NUMERIC	NUMERIC(5,3)	5 chiffres au total dont 3 derrière la virgule
FLOAT	FLOAT	N'accepte pas de paramètre comme DECIMAL
DOUBLE	DOUBLE	N'accepte pas de paramètre comme DECIMAL
REAL	REAL	N'accepte pas de paramètre comme DECIMAL

- Les Mots-Clés importants

UNSIGNED Indique le chiffre n'est pas négatif

AUTO_INCREMENT Indique qu'à chaque nouvel ajout, le chiffre augmente de 1 automatiquement

NOT NULL Indique que le champ ne doit pas être vide

PRIMARY KEY Indique que le champ est une clé primaire

FOREIGN KEY Indique que le champ est une clé étrangère

Les clés primaires et étrangères sont très importantes. Elle représente l'identifiant unique de la table, comme un client enregistré dans une société.

- **Gestion des tables**

Comment gérer les tables, c'est très simple :

CREATE TABLE

DROP TABLE

ALTER TABLE

INSERT INTO

Ces commandes, permettent respectivement de créer, supprimer, modifier une table et d'y insérer des données.

- **Gestion des données et les opérateurs**

SELECT

WHERE

Ces commandes permettent respectivement de sélectionner (Select) des données suivant une condition (Where).

Opérateur	Signification
=	égal
<	inférieur
<=	inférieur ou égal
>	supérieur
>=	supérieur ou égal
<> ou !=	différent
<=>	égal (valable pour <code>NULL</code> aussi)

- Les jointures

INNER JOIN

Les jointures permettent de lier plusieurs tables ensemble et de récupérer les données des tables sélectionnées.

Le langage l'UML

Prenons quelques secondes et imaginons que l'on veut créer une base de données regroupant les films, les salles et dates de projection, les acteurs ...

Il faut tout d'abord poser le cadre et se mettre sur une feuille pour pouvoir imaginer tout ça.

Pour cela, on utilise généralement de l'**UML (Unified Modeling Language)**. (Voici pour le réfractaire du Stylo un site pour faire de l'UML <https://app.diagrams.net>)

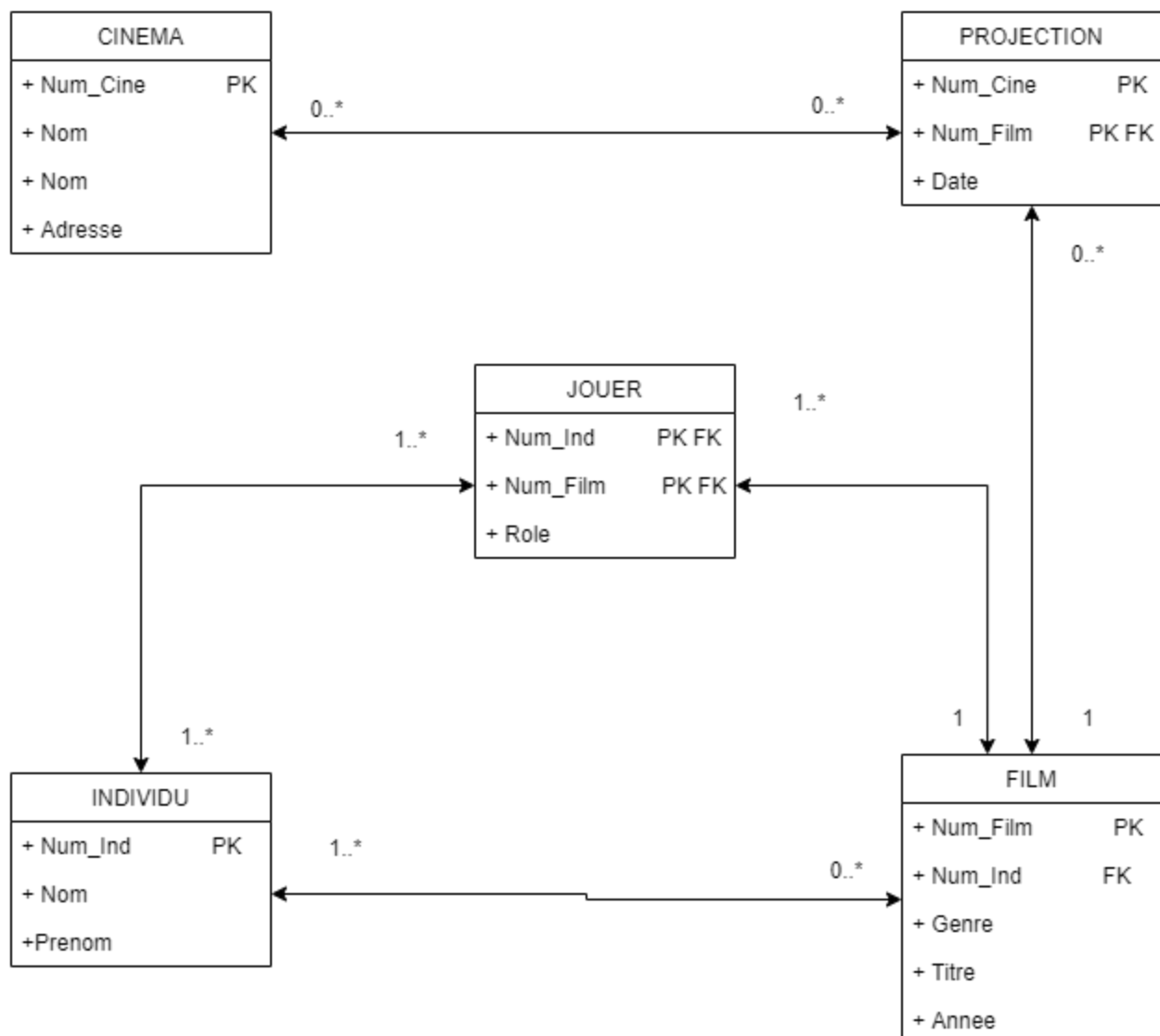


Table Name
+ field:
+ field:
+ field:

Représente le nom de la table et les champs

0

1

0..*

0..1

1..*

PK

FK

Représente la relation entre les tables (Exemple : 1 cinéma peut projeter 0 ou plusieurs projections (0..*))

Représente Primary ou Foreign Key

Le langage SQL (2^{ème} partie)

Maintenant que nous avons posé l'architecture de notre BDD, nous pouvons commencer à l'écrire en langage SQL.

Tout d'abord les créations des tables : (Ici exemple, voir fichier sql pour le reste)

```
-- =====*/
-- Table : cinema */
-- =====*/

DROP TABLE IF EXISTS cinema;
CREATE TABLE IF NOT EXISTS cinema (
num_cine      INT UNSIGNED NOT NULL AUTO_INCREMENT,
nom           VARCHAR(50)          null,
adresse       VARCHAR(75)          null,
constraint PK_CINEMA primary key (num_cine)
) ENGINE=InnoDB CHARACTER SET utf8;
```


Maintenant ajoutons y les données :

```
-- Base de donnees: 'film'
--
-- Contenu de la table 'cinema'
--

INSERT INTO cinema (num_cine, nom, adresse) VALUES(1, 'Le Renoir', '13100
Aix-en-Provence');
INSERT INTO cinema (num_cine, nom, adresse) VALUES(2, 'Le Fontenelle', '78160
Marly-le-Roi');
INSERT INTO cinema (num_cine, nom, adresse) VALUES(3, 'Gaumont Wilson', '31000
Toulouse');
INSERT INTO cinema (num_cine, nom, adresse) VALUES(4, 'Espace Cin', '93800
Epinay-sur-Seine');
--
```

Cherchons y des données :

```
SELECT *
FROM cinema
WHERE nom like '%reno%';
-- On sélectionne toutes les données des colonnes de la table cinema lorsque le
nom ressemble à "Reno"
```

Résultat de la requête :

num_cine	nom	adresse
1	Le Renoir	13100 Aix-en-Provence