# Interagir avec une Base de Données en PHP

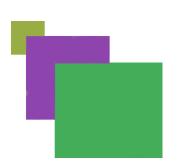
Université Virtuelle de Côte d'Ivoire



# Table des matières

Objectifs	3
Introduction	4
I - Création d'une base de données sous MySQL	5
1. Présentation des SGBD	5
2. Création d'une base de donnée MySQL	6
II - Exercice : Activité d'auto-évaluation No 1	8
III - Se connecter à une base de données MySQL	10
1. Connexion à une base de données avec mysqli	10
2. Connexion à une base de données mysql avec PDO	
IV - Exercice : Activité d'auto-évaluation No 2	16
V - Exécuter des requêtes SQL en PHP	18
1. Insertion de données	19
2. Sélection de données	21
3. Modification et suppression de données	21
4. Travaux dirigés	22
VI - Exercice : Activité d'auto-évaluation No 3	32
VII - Alternative entre base de données et les fichiers	33
1. Choix entre base de données et fichiers	
2. Utilisation de la base de données SQLite	33
VIII - Exercice : Activité d'auto-évaluation No 4	36

# **Objectifs**



À la fin de cette leçon, vous serez capable :

- De créer une base de données sous MySQL;
- De créer une connexion à une base de données MySQL;
- D'exécuter des requêtes SQL en PHP
- De stocker des informations dans une base de donnée SQLite.

## Introduction



- Aujourd'hui, la disponibilité de systèmes de gestion de base de données fiables permet aux organisations de toute taille de gérer des données efficacement, de déployer des applications utilisant ces données et de les stocker. Les bases de données sont actuellement au cœur du système d'information des entreprises.
- Une base de données est un ensemble de données bien structurées, apparentées, enregistrées sur un support de stockage représentant un univers du monde réel et accessibles via un système de gestion de base de données.
- L'avantage d'utiliser les bases de données pour stocker des informations par rapports aux fichiers que nous avons vu dans la leçon précédente, est que les bases de données offrent une meilleure organisation et structuration des données tout en garantissant leur sécurité et en facilitant leur accès et leur modification
- L'intérêt majeur de PHP est son interfaçage avec un grand nombre de bases de données d'une manière relativement simple et efficace. PHP s'utilise bien souvent avec MySQL, un SGBD rapide (à moyenne charge) et qui satisfait à la plupart des sites Internet. Dans le cadre de ce cours, MySQL fera l'objet principale de notre étude.

# Création d'une base de données sous MySQL



## 1. Présentation des SGBD

- Un SGBD est un logiciel ou application permettant d'interagir avec une base de données. Elle permet la création de la base de données, l'enregistrement, la modification, la récupération et la suppression des informations au sein de celle-ci. Pour interagir avec un SGBD, on utilise un langage appelé le SQL (Structured Query Language).
- Actuellement, la plupart des SGBD fonctionnent selon un mode client/serveur. Le serveur (sousentendu la machine qui stocke les données) reçoit des requêtes de plusieurs clients et ceci de manière concurrente. Le serveur analyse la requête, la traite et retourne le résultat au client. Le modèle client/serveur est assez souvent implémenté au moyen de l'interface des sockets.

IL existe plusieurs types de base de données ou de SGBD avec lesquels PHP peut interagir. On peut citer entre autres :

- Les SGBD payants :
  - Oracle Database
  - Microsoft SQL Server
  - DB2
  - Microsoft Access
  - Sybase
- Les SGBD gratuits :
  - MySQL
  - PostgreSQL
  - MariaDB
  - SQLite
  - MongoDB

Dans le cas de ce cours nous étudierons principalement le cas de MYSQL qui est une base de données directement intégrée dans wamp ou xampp, et qui est automatiquement prise par défaut par php.

MySQL et MariaBD sont deux bases de données ayant beaucoup de similitudes et créées par le même auteur.

## 2. Création d'une base de donnée MySQL

Pour créer et administrer une base de données MySQL, nous avons plusieurs possibilités. On peut utiliser :

- PhpMyAdmin qui est intégré dans wamp ou dans xampp
- MySQL Community Server, https://dev.mysql.com/downloads/mysql/

Il est possible d'administrer de façon très agréable une base de données MySQL par le biais de MySQL Workbench téléchargeable à l'adresse <a href="https://dev.mysql.com/downloads/workbench/">https://dev.mysql.com/downloads/workbench/</a> ou <a href="https://www.mysql.com/fr/products/workbench/">https://www.mysql.com/fr/products/workbench/</a>. MySQL Workbench n'intègre pas de serveur mysql ou moteur de base de données MySQl, mais offre une interface conviviale pour interagir avec ce dernier et administrer les bases de données qui s'y trouvent déjà.

Dans le cadre de ce cours nous allons créer une base de données MySQL en passant par PhpMyAdmin. Pour ce fait :

- Après avoir démarré wamp ou les services apache et mysql de xampp, il faut saisir dans la barre d'adresse d'un navigateur (de préférence chrome ou firefox) l'adresse http://localhost/phpmyadmin/
- Une fois sur phpmyadmin vous cliquez sur *bases de données* au niveau de la barre en haut de la page



Sur la page qui s'affiche se trouve une zone créer une base de données, vous y inscrivez le nom de votre base de données et vous choisissez comme interclassement utf-8\_general\_ci. L' interclassement est facultatif, dans le cadre de ce cours, nous choisissons comme nom de base de données ecole. Après avoir renseigné le nom de la base de données vous cliquez sur le bouton créer à côté.



- Une fois vous cliquez sur le bouton créer, vous verrez le nom de la base de données apparaître en base et à gauche dans la liste des base de données qui s'affichent.



- Cliquez sur le nom de la base de données qui s'affiche et une page va s'ouvrir avec une zone nouvelle table. mettez le nom de la table que vous voulez créer dans la case en bas et mettez dans la case d'à coté le nombre de colonnes de la table. Dans notre cas, nous choisissons comme nom de la table etudiants et comme nombre de colonnes 10. Après vous cliquez sur le bouton exécuter en bas à droite.



- Une page s'ouvre, où on vous demande de renseigner les champs de la table avec les propriétés. Vous renseignez les champs selon le formulaire d'enregistrement des étudiants dans la leçon précédente, conformément à l'image ci-dessous. Sachez que le champ *id* est la clé primaire de notre table et il à la valeur auto-incrémente (AI). A la fin, vous cliquez sur le bouton *sauvegarder* en bas à droite.



Félicitation, vous venez de créer une base de données *ecole* avec une table *étudiants*, nous verrons dans la suite du cours comment enregistrer automatiquement des informations dans cette table avec PHP

- Si vous voulez insérer des informations dans la table etudiants depuis phpmyadmin, vous partez dans l'onglet inserer en haut
- Si vous voulez voir les informations de la table etudiants, vous partez dans l'onglet afficher en haut

# Exercice: Activité d'autoévaluation No 1



#### Exercice

Une base d données est :

#### 0

un ensemble de données enregistrées sur un support de stockage représentant un univers du monde réel et accessibles via un système de recherche

### 0

une base d'informations enregistrées sur un disque dur représentant un univers du monde rée let accessible via un moteur de recherche

## 0

un ensemble de données bien structurées, apparentées, enregistrées sur un support de stockage représentant un univers du monde réel et accessibles via un système de gestion de base de données.

#### 0

un logiciel ou application permettant d'interagir avec l'utilisateur. Elle permet le stockage d'information textuelle et binaire représentant un univers du monde réel

#### Exercice

#### Un SGBD est

#### 0

un ensemble de données bien structurées, apparentées, enregistrées sur un support de stockage représentant un univers du monde réel et accessibles via un système de gestion de base de données.

## 0

un logiciel ou application permettant d'interagir avec l'utilisateur. Elle permet le stockage d'information textuelle et binaire représentant un univers du monde réel

#### 0

logiciel permettant d'interagir avec une application utilisateur. Elle permet l'enregistrement et la récupération des informations sur un support de stockage par le biais d'une requête

## 0

une application permettant d'interagir avec une base de données. Elle permet la création de la base de données, l'enregistrement, la modification, la récupération et la suppression des informations au sein de celle-ci par le biais d'un langage appelé le SQL

_		
Exe	rc1	CA
LAU.	$\mathbf{L} \cup \mathbf{I}$	.uu

	vantage d'utiliser les bases de données pour stocker des informations par rapports aux fichiers est les bases de données : (3 réponses)
	sont plus volumineuses que les fichiers
	offrent une meilleure organisation et structuration des données
	sont plus persistantes que les fichiers
	offrent une meilleure sécurité aux données
	offrent un meilleur accroissement horizontal et vertical aux informations
	offrent un meilleur accès aux données
Exe	ercice
Parı	mi ces logiciels, lesquels sont des bases de données : (5 réponses)
	Microsoft Excel
	MySQL
	Oracle
	MyDB
	PogreSQL
	SQLite
	PostgreSQL
	DBMaria
	ElasticDB
	MariaDB

# Se connecter à une base de données MySQL



De façon générale, pour se connecter à une base de données ou à un SGBD par le biais d'un langage de programmation :

- Il faut le nom ou l'adresse IP de la machine serveur sur laquelle se trouve la base de données
- Il faut le nom du serveur de base de donnée (sgbd) auquel on aimerait accéder avec éventuellement le numéro de port sur lequel le serveur écoute
- Il faut le login et le mot de passe du compte par lequel passer pour accéder au SGBD
- Avec les informations ci-dessus nous pouvons nous connecter à un SGBD, toute fois, il faut aussi préciser le nom de la base de données sur laquelle effectuer les opérations. En effet, un serveur de base de données peut comprendre plusieurs bases de données en son sein.

Une fois connecter à une base de données, nous pouvons effectuer toute sorte de requêtes que nous désirons.

Il existe trois API principales pour se connecter à MySQL:

- L'extension mysql (pour les versions PHP < 4.1.3), elle est obsolète
- L'extension mysqli (mysql improved pour les versions PHP > 5)
- PHP Data Objects (PDO) qui permet de travailler avec différentes bases de données en conservant le même code.

Nous allons présenter comment se connecter à une base de données MySQL avec l'extension mysqli et PDO

## 1. Connexion à une base de données avec mysqli

Processus d'interaction avec une base de données par mysgli

L'exploitation de MySQL avec PHP s'effectue en plusieurs étapes:

- ouverture d'une connexion à MySQL
- sélection de la base de données

- requête sur la base de données
- exploitation des résultats de la requête
- fermeture de la connexion à MySQL

Les Fonctions utilisées sont:

- mysql\_connect
- mysql\_select\_db
- mysql\_query
- mysql\_close

## Ouvrir une connexion la base de données MySQL

Pour ouvrir une connexion, il faut définir quatre informations que sont :

- *La machine serveur*: dans notre cas, le nom du serveur est *localhost* (pour faire allusion à l'ordinateur sur lequel mysql se trouve actuellement), en effet, vous travaillez actuellement de façon locale sur votre ordinateur. Vous pouvez aussi utiliser l'adresse IP *127.0.0.1* qui est l'adresse IP locale.
- La base de données : nous allons prendre comme nom de base de données, la base de données que nous avons créée précédemment qui est ecole
- *Le nom de l'utilisateur* par lequel passer pour accéder à la base de données. Le nom de l'utilisateur par défaut sous MYSQL est *root*
- le mot de passe de l'utilisateur : par défaut, root n'a pas de mot de passe.

Ici, nous ne précisons pas le nom du SGBD, car il s'agit bel et bien de mysql.

Pour ouvrir une connexion on utilise la fonction *mysql\_connect* 

Ci-dessous, le code de connexion :

```
1 <?php
2 $serveur="localhost";
3 $user="root";
4 $password="";
5 $connexion = mysql_connect($serveur, $user, $password);
6 if($connexion){ // on teste si la connexion a réussie
7     echo "Connexion réussie";
8 }else{
9     echo "Connexion echouée";
10 }
11 ?>
```

## Sélectionner une base de données

- Une fois connecté à une base de données, il faut sélectionner la base de données sur laquelle effectuer les opérations.

- Pour sélectionner la base de données, on utilise la fonction *mysql\_select\_db*. La *syntaxe* est mysql\_select\_db(nom\_de\_la\_base\_de\_donnees)

## Opérations sur la base de données

- Une fois, une base de données sélectionnées, il faut effectuer des opérations sur la base de données.
- On peut avoir des opérations de sélection, d'ajout de données, de suppression ou de modification.
- Pour effectuer une opération, on utilise la fonction *mysql\_query*. La syntaxe est mysql\_query (requete, connexion). Le deuxième argument est facultatif
- Une fois la requête effectuée, si c'est une requête de sélection, il faut afficher les valeurs sélectionnées.
- Une fois les opérations finies, il faut fermer la connexion avec la fonction mysql\_close. La syntaxte est mysql\_close(\$connexion);



## Exemple : Exemple de connexion et d'interaction avec une base de données mysql par le biais de mysqli

```
1 <?php
2 $serveur="localhost";
3 $bd="ecole";
4 $user="root";
 5 $password="";
6 $connexion = mysql_connect($serveur, $user, $password);
7 if ($connexion) { // on teste si la connexion a réussie
    if (mysql_select_db($bd)) {
0
         $requete= "select * from etudiants";
10
         $resultat= mysql_query($requete, $connexion);
11
         //mysql_fetch_row retourne le nombre de ligne retournés par la
  réquête
12
          if (!mysql_fetch_row($resultat)) {
13
              echo "Aucun enregitrement d'étudiants effectué";
14
         }else{
15
              while($row = mysql_fetch_row($resultat)){
16
                  print_r($row);
17
                  echo "<br/>";
18
              }
19
          }
20
     }
21 mysql_close($connexion);
22 }else {
     echo "Connexion échouée";
24 }
25 ?>
```

### Pour plus d'information sur l'utilisation de mysqli, voici quelques liens :

- https://fr.wikiversity.org/wiki/PHP/Base\_de\_donn%C3%A9es
- https://www.youtube.com/watch?v=FCWwi7M13jY
- https://www.commentcamarche.com/contents/784-php-bases-de-donnees

- http://php.net/manual/fr/function.mysql-connect.php
- https://eusebius.developpez.com/phpmysql-base/
- http://icps.u-strasbg.fr/~genaud/courses/webtech/php/ch06.html
- https://www.valhalla.fr/2005/09/25/construire-une-base-de-donnees-simple-avec-php-et-mysql/
- https://mysql.developpez.com/cours/
- https://www.w3schools.com/php/php\_mysql\_connect.asp
- https://www.youtube.com/watch?v=cy1tMLREX88

## 2. Connexion à une base de données mysql avec PDO

### Activer PDO

PDO est un outil complet qui permet d'accéder à n'importe quel type de base de données de la même manière. On peut donc l'utiliser pour se connecter aussi bien à MySQL que PostgreSQL ou Oracle.

Normalement, PDO est activé par défaut. Pour le vérifier,

- faites un clic gauche sur l'icône de WAMP dans la barre des tâches,
- puis allez dans le menu PHP / Extensions PHPet vérifiez que php\_pdo\_mysql est bien coché.

## Se connecter à MySQL avec PDO

Nous allons avoir besoin de quatre renseignements :

- le nom de l'hôte : c'est l'adresse de l'ordinateur où MySQL est installé (comme une adresse IP). Le plus souvent, MySQL est installé sur le même ordinateur que PHP : dans ce cas, mettez la valeur localhost (cela signifie « sur le même ordinateur »). Néanmoins, si votre serveur est sur internet ou dans un réseau, il faut utiliser son nom ou son adresse IP dans le réseau.
- *la base* : c'est le nom de la base de données à laquelle vous voulez vous connecter. Dans notre cas, la base s'appelle *ecole* que nous avons créée précédemment;
- le login : le compte par lequel nous accédons à la base de données., ici c'est root;
- *le mot de passe* : le mot de passe associé au compte choisi, notre compte root choisi est sans mot de passe.

A partir des quatre informations ci-dessus, il faut définir une *url* ou *chaîne de connexion* qui est de la forme "mysql:host=localhost;dbname=ecole" ou "mysql:host=localhost;dbname=ecole;charset=utf8".

Pour se connecter à la base de données, on utilise la syntaxe : new PDO(chaine\_de\_connexion, login, password);



## Exemple: Exemple de connexion à une base de données en utilisant PDO

```
1 <?php
2 $serveur="localhost";
3 $bd="ecole";
4 $user="root";</pre>
```

```
5 $password="";
6 $url = 'mysql:host='.$serveur.';dbname='.$bd.';charset=utf8';
7 $connexion = new PDO($url,$user,$password);
8
9 if($connexion){ // on teste si la connexion a réussie
10 echo "Connexion réussie";
11 }else {
12 echo "Connexion échouée";
13 }
14 ?>
```

## Tester la présence d'erreurs

Si les information informations (nom de l'hôte, de la base, le login et le mot de passe) sont mal renseignées, des erreurs peuvent subvenir. Les accès aux bases de données sont très délicats et peuvent très souvent générer des erreurs, pour ce fait, il faut mettre les instructions dans une structure *try catch* afin de gérer les erreurs et les exceptions

## Exécuter une requête

- Pour exécuter une requête, on utilise la syntaxe :\presultat= \presultat= \p
- *\$resulat* contient des tableaux associatifs représentant les tuples (lignes) renvoyés par la base de données.
- Pour parcourir \$resultat et récupérer les tuples un à un, on utilise l'instruction \$ligne = \$resultat->fetch();

## 7

## Exemple : Exemple de connexion et de récupération de données par PDO

```
1 <?php
2 $serveur="localhost";
3 $bd="ecole";
4 $user="root";
5 $password="";
 6 $url = 'mysql:host='.$serveur.';dbname='.$bd.';charset=utf8';
7try {
   $connexion = new PDO($url,$user,$password);
    if($connexion){
10
         $reponse = $connexion->query('SELECT * FROM etudiants');
11
         //On parcour le tableau pour récuperer les tuples et afficher les
  valeurs
12
         while ($donnee = $reponse->fetch())
13
          {
14
          echo $donnee['nom'] . '<br />';
15
16
         $reponse->closeCursor();//on ferme la connexion
      }else {
18
         echo "Connexion échouée";
19
20 }catch(Exception $e){
     // En cas d'erreur, on affiche un message et on arrête tout
      die('Erreur : '.$e->getMessage());
23 }
```

24 25 ?>

## Dans la suite de ce cours, nous travaillerons avec PDO

## Quelques ressources:

- https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/914292 lire-des-donnees
- https://www.numelion.com/se-connecter-a-une-base-de-donnees-en-php.html

# **Exercice : Activité d'autoévaluation No 2**



Exe	ercice
Po	ur se connecter à MySQL, on peut utiliser : (3 réponses)
	L'extension PhpMyDB
	PHP Data Objects (PDO)
	L'extension PhpMySQL
	L'extension mysqli
	L'extension MysqlPhp
	L'extension mysql
Exe	ercice
L'a	vantage de PDO par rapport aux autre méthode de connexion à une base de données est que PDO
0	permet un accès plus robuste et plus sécurisé aux base de données
0	permet d'interagir avec plusieurs base de données tout en conservant le même code
0	offre un accès simple, facile, et agréable aux bases de données
0	est plus rapide que les autres méthodes
Exe	ercice
Poi	ur se connecter à une base de données MySQL, il faut les quatre informations suivantes :
	un nom d'hébergeur web
	un nom de serveur
	un code de sécurité généré par MySQL
	un nom de base de données

Exercice: Activité d'auto-évaluation No 2

☐ un code d'accès PHP			
un nom de table de stoc	kage de données		
☐ un login			
un mot de passe			
Exercice			
La syntaxe PDO pour se connecter à une base de donnés est :			
connexion =	(chaine_de_connexion, login, password);		

## Exécuter des requêtes SQL en PHP



Pour interagir avec une base de données, il faut avoir une connaissance de base en SQL. Pour savoir comment écrire des requêtes SQL sous MySQL, vous pouvez consulter les supports suivants :

- https://fr.wikibooks.org/wiki/Les\_bases\_de\_donn%C3%A9es/Les\_requ%C3%AAtes\_en\_SQL
- https://openclassrooms.com/fr/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql
- https://sqlpro.developpez.com/cours/sqlaz/sousrequetes/
- https://sql.sh/

L'objectif de cette partie n'est pas de vous apprendre à écrire des requêtes SQL mais de vous apprendre à exécuter des requêtes sql et de les traiter en PHP.

Il existe deux manières d'exécuter des requêtes PDO, nous avons :

- Les requêtes simples
  - Pour exécuter une simple requête de sélection, on utilise la fonction méthode ou fonction *query()*
  - Pour exécuter une simple requête d'insertion, de modification ou de suppression, on utilise la fonction méthode *exec()*
  - query() et exec() prennent en paramètre la requête SQL à exécuter
- Les requêtes préparées : Elles se font de deux manières : Les requêtes préparées avec des marqueurs ? et Les requêtes préparées nominatifs
  - Pour préparer une requête, on utilise la méthode *prepare()*, elle prend en paramètre la requête SQL à exécuter.
  - Pour exécuter une requête préparée, on utilise la méthode *execute()*, elle prend en paramètre un tableau contenant les valeurs à injecter dans la requête

*NB* : Les requêtes préparées présentent un intérêt lorsque nous devons injecter des valeurs ou variables dans la requêtes, sinon vaut mieux utiliser des requêtes simples.

## 1. Insertion de données

## Comment écrire une requête SQL d'insertion

- Syntaxe générale d'une requête SQL d'insertion :

INSERT INTO table (nom\_colonne\_1, nom\_colonne\_2, ...) VALUES ('valeur 1', 'valeur 2', ...) ; Si nous ne voulons pas énumérer les colonnes de la table, nous pouvons l'écrire de cette manière :

*INSERT INTO* table *VALUES* ('valeur 1', 'valeur 2', ...); : Toutefois, il faut énumerer toutes les valeurs dans l'ordre des champs de la table

#### Exemple 1:

INSERT INTO client (prenom, nom, ville, age) VALUES ('Rébecca', 'Armand', 'Saint-Didierdes-Bois', 24);

ou bien: INSERT INTO client VALUES ('Rébecca', 'Armand', 'Saint-Didier-des-Bois', 24);

Exemple 2 : insertion de plusieurs valeurs

INSERT INTO client (prenom, nom, ville, age) VALUES

('Rébecca', 'Armand', 'Saint-Didier-des-Bois', 24),

('Aimée', 'Hebert', 'Marigny-le-Châtel', 36),

('Marielle', 'Ribeiro', 'Maillères', 27);

Syntaxe particulière d'une requête SQL d'insertion :

INSERT INTO table SET nom\_colonne1='valeur1', nom\_colonne2 = 'valeur2',...

Exemple:

INSERT INTO client SET prenom='Rébecca', nom='Armand', ville= 'Saint-Didier-des-Bois', age =24;

*NB* : Il n'est pas obligé d'écrire les mot clés SQL (INSERT INTO, VALUES, SET, ...) en majuscule, on pourrait bien les écrire en minuscule.

Considérons une table client avec champs id, nom, tel, email. Voici ci-dessous quelques requêtes d'insertion en PHP; les variables \$url, \$user et \$password sont les variable définies précédemment

## Exemple : Exemple de requête simple d'insertion

```
$connexion = new PDO($url,$user,$password);
$connexion->exec("INSERT INTO client VALUES
(1,'Akissi','75845236','akissi@yahoo.fr')");
```

Si on considère que le id est auto-incrémente, il n'est pas obligé de préciser la valeur du id, on pourrait fait comme suit :

```
$connexion->exec("INSERT INTO client VALUES('',
'Akissi','75845236','akissi@yahoo.fr')") ;// on définit une chaîne vide pour id
ou
```

```
$connexion->exec("INSERT INTO client(nom,tel,email) VALUES
('Akissi','75845236','akissi@yahoo.fr')")
```

*NB*: Lorsque vous faites des insertions, des suppressions, ou des modifications dans la base de données depuis PHP, pour voir les modifications dans phpmyadmin, il faut cliquer sur l'onglet afficher en haut pour que la page se rafraîchisse.

## 7

## Exemple: Exemple de requête simple d'insertion avec injection de variables

```
1 $connexion = new PDO($url,$user,$password);
2 $nom= "Yao Bertin";
3 $tel="47856321";
4 $email="yao@gmail.com";
5 $connexion->exec("INSERT INTO client VALUES('','$nom','$tel','$email')");
6 echo "Insertion effectuée avec succès";
7
```

Il faut noter qu'injecter des variables provenant d'un formulaire dans une requête SQL tel que nous venons de le faire présente des failles de sécurité, pour mieux faire les choses, il faut utiliser des requêtes préparées

## Exemple : Exemple de requête préparée avec marqueur ?

## Exemple : Exemple de requête préparée nominatif

Ici, les paramètres de la requête ont des noms précédés de deux points (:). La méthode exécute() prend en paramètre un tableau associatif dont les clés sont les paramètres de la requête et les valeurs sont les valeurs à attribuer aux paramètres.

```
1 $connexion = new PDO($url,$user,$password);
2 $nom= "Zamble Bi Jean";
3 $tel="077542136";
4 $email="jean@gmail.com";
5 $requete = $connexion->prepare("INSERT INTO client(nom,tel,email) VALUES(: nom,:tel,:email)");
6 $requete->execute(array("nom"=>$nom,"tel"=>$tel,"email"=>$email));
7 echo "Insertion effectuée avec succès";
```

Dans la suite de ce cours, nous ne travaillerons qu'avec les requêtes préparées nominatifs

## 2. Sélection de données

## Comment écrire une requête de sélection

Une requête de sélection est de la forme : SELECT nom\_du\_champ FROM nom\_du\_tableau ou bien SELECT nom\_du\_champ FROM nom\_du\_tableau WHERE ...



## Exemple : Exemple de requête de sélection

```
1 $connexion = new PDO($url,$user,$password);
2 $requete = $connexion->prepare("SELECT * FROM client");
3 $requete->execute();
4 echo "Les clients sont : <br/>
5 while($data= $requete->fetch()){
6    echo $data["nom"]."<br/>";
7 }
```

## Exemple : Exemple de requête de sélection avec clausse WHERE

```
1 $email="jean@gmail.com";
2 $connexion = new PDO($url,$user,$password);
3 $requete = $connexion->prepare("SELECT nom FROM client WHERE email=:email");
4 $requete->execute(array("email"=>$email));
5 if($data= $requete->fetch()){
6    echo "Le nom du client ayant l'email $email est : <br/>;
7    echo $data["nom"];
8 }
```

## 3. Modification et suppression de données

Comment écrire une requête de modification (mise à jour ou update)?

Une requête de mise à jour s'écrit de la forme :

```
UPDATE table

SET colonne_1 = 'valeur 1', colonne_2 = 'valeur 2', colonne_3 =
'valeur 3'

WHERE condition
```

## Exemple : Exemple de mise à jour

```
1 $email="jean@gmail.com";
2 $nom="Mariam Koffi";
3 $connexion = new PDO($url,$user,$password);
4 $requete = $connexion->prepare("UPDATE client SET nom=:nom WHERE email=: email");
5 $requete->execute(array("nom"=>$nom,"email"=>$email));
6 if($requete->rowCount()>0){
```

```
7 echo "Mise à jour effectuée avec succès";
8 }else echo "Mise à jour échouée";
9
```

## Comment écrire une requête de Suppression ?

Une requête de mise à jour s'écrit de la forme :

```
DELETE FROM `table` WHERE condition
```



## Exemple : Exemple de requête de suppression

```
1 $connexion = new PDO($url,$user,$password);
2 $requete = $connexion->prepare("DELETE FROM client WHERE id=:id");
3 $requete->execute(array("id"=>2));
4 if($requete->rowCount()>0){
5    echo "Suppression effectuée avec succès";
6 }else echo "Suppression échouée";
7
```

## 4. Travaux dirigés

## Libellé

Dans la leçon précédente, nous avons créé un formulaire d'enregistrement des étudiants et nous avons enregistrer les informations dans un fichier. Nous allons reprendre le même travail, mais cette fois ci en enregistrant les informations dans une base de données. Les informations seront enregistrées dans la table etudiants de la base de données ecole que nous avons créée. Les mots de passe sont cryptés avec la fonction PASSWORD() de MySQL.

#### Correction

- On donne comme id au formulaire myform
- Les paramètres de connexion seront dans un fichier config.php que nous allons inclure dans les pages de requête
- Nous allons créer une fonction enregFichier() dans le fichier fonctions.php chargée d'enregistrer les images
- Pour la suppression d'un étudiant, on va créer une page suppression.php
- Nous allons écrire une fonction supEtudiant() chargée de supprimer un étudiant lorsqu'on clique sur supprimer. La fonction prend comme argument l'id de l'étudiant et le bouton sur lequel on clique.
- Nous utiliserons ici ajax pour transmettre les informations au serveur. Pour ceux qui ne connaissent pas ajax, c'est une technologie JavaScript permettant de charger une page web tout en restant sur la même page où l'on se trouve, on parle de chargement asynchrone. Voici quelques liens pour vous :

https://openclassrooms.com/fr/courses/1631636-simplifiez-vos-developpements-javascript-avec-jquery/1636305-premiers-pas-avec-ajax

http://creersonsiteweb.net/page-jquery-javascript-ajax https://sutterlity.gitbooks.io/apprendre-jquery/content/la\_methode\_ajax.html https://ajax.developpez.com/cours/

## Contenu de la page config.php

```
1 <?php
2 $serveur="localhost";
3 $bd="ecole";
4 $user="root";
5 $password="";
6 $url = 'mysql:host='.$serveur.';dbname='.$bd;
7 ?>
```

#### Contenu de la page index.php

```
1 <!DOCTYPE HTML>
   2 < html >
   3 <head lang="fr">
   4 <meta http-equiv="content-type" content="text/html" />
   5 <meta charset="utf-8"/>
       <meta name="viewport" content="width=device-width, initial-scale=1,</pre>
    shrink-to-fit=no"/>
       <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet" />
   0
      <script src="js/jquery-3.3.1.min.js" type="text/javascript"></script>
  11
       <script src="js/popper.min.js" type="text/javascript"></script>
  12.
       <script src="bootstrap/js/bootstrap.min.js" type="text/javascript">
       <script src="js/validator.min.js"></script>
  14
        <style type="text/css">
  15
          form {
  16
               width: 800px;
  17
               max-width: 100%;
  18
               background-color: rgb(245,245,245);
  19
               border-radius: 7px 7px 0px 0px;
              padding: 10px;
  20
  21
               text-align: left;
  22
               margin: 25px;
  23
               box-shadow: 0px 2px 8px rgba(32,32,32,0.5);
  24
            .display-none {
  26
                display: none;
  2.7
            }
  28
        </style>
       <script type="text/javascript">
  30
          function choisirPhoto(x){
  31
                $(x).parents('.image-viewer').children('input[type="file"]').
    click();
  32
  33
            function show_apercu(x){
               var files = jQuery(x)[0].files;
                  if (files.length > 0) {
                    var file=files[0];
  36
  37
                   if(file.size<=2*1024*1024){
  38
                     var file=files[0];
                        $(x).prevAll('.file-name').html(file.name+" "+parseInt
(file.size/1024) + ' Ko');
```

```
40
                       $(x).prevAll('.img_apercu').attr('src', window.URL.
  createObjectURL(file));
41
                       $(x).prevAll('.img_apercu').show();
42
                       $(x).parent('.image-viewer').children('a').show();
43
                 }else{
44
                  alert ("Image trop lourde, le poids doit être inférieur à
  2M");
45
                 }
46
               }
47
          }
48
49
          function removeFile(x){
50
              $(x).nextAll('input').val("");
51
              $(x).prevAll('.file-name').html("");
52
              $(x).prevAll('.img_apercu').hide();
              $(x).parent('.image-viewer').children('a').hide();
53
54
          }
      </script>
55
56 </head>
57 <body>
      <a href="etudiants.php" class="btn btn-primary btn-sm" style="margin:</pre>
  25px 50px;">Voir les étudiants</a>
      <h2 style="text-decoration: underline;"><center>Formulaire
  d'inscription</center></h2>
60
      <center>
61
        <form id="myform" data-toggle="validator" role="form" method="post"</pre>
  62
63
              <label class="col-sm-4 col-form-label">Nom* : </label>
64
              <div class="col-sm-8">
65
                  <input type="text" class="form-control" name="nom" pattern=</pre>
  "^[A-Z][A-Za-z' -]+" required />
66
                  <div class="help-block with-errors">Au moins 2 Caractères
  et commencé par une majuscule</div>
67
               </div>
68
           </div>
69
           <div class="form-group row">
70
              <label class="col-sm-4 col-form-label">Tel* : </label>
71
              <div class="col-sm-8">
                  <input type="text" pattern="^[0-9]{8}$" name="tel" class=</pre>
  "form-control" data-error="Le Numéro doit être 8 chiffres" required />
73
                  <div class="help-block with-errors"></div>
74
              </div>
75
           </div>
           <div class="form-group row">
76
77
              <label class="col-sm-4 col-form-label">Email : </label>
78
              <div class="col-sm-8">
79
                  <div class="input-group">
80
                       <input type="email" class="form-control" name="email"</pre>
  data-error="Le format d'email n'est pas correct" required />
                       <div class="input-group-prepend">
81
82
                        <span class="input-group-text">@</span>
83
                       </div>
84
                   </div>
85
                  <div class="help-block with-errors"></div>
86
              </div>
87
           </div>
88
           <div class="form-group row">
89
              <label class="col-sm-4 col-form-label">Date de Naissance : 
  label>
              <div class="col-sm-8">
```

```
91
                    <input type="date" class="form-control" name=</pre>
   "date_naissance" />
92
               </div>
93
            </div>
            <div class="form-group row">
95
               <label class="col-sm-4 col-form-label">Matricule* : </label>
96
               <div class="col-sm-8">
97
                    <input type="text" name="matricule" placeholder="Ex:</pre>
   UV4578AX" pattern="^UV[0-9]{4}[A-Z]{2}" required class="form-control" />
08
                    <div class="help-block with-errors">
90
                        Le matricule doit commencer par UV suivi de 4
   chiffres, puis se termine par deux lettres Majuscule
101
               </div>
            </div>
103
            <div class="form-group row">
               <label class="col-sm-4 col-form-label">Ville</label>
105
               <div class="col-sm-8">
106
                    <select class="form-control" name="ville" >
                        <option></option>
108
                        <option>Abidjan</option>
109
                        <option>Abengourou</option>
110
                        <option>Bouaké
111
                        <option>Daloa
112
                        <option>Gagnoa</option>
113
                        <option>Korogho</option>
114
                        <option>Man</option>
115
                        <option>Yamoussoukro</option>
116
                    </select>
117
               </div>
118
             </div>
             <div class="form-group row">
119
120
               <label class="col-sm-4 col-form-label">Sexe : </label>
121
               <div class="col-sm-8">
122
                    <div class="custom-control custom-radio custom-control-</pre>
   inline">
123
                      <input type="radio" id="sexe1" name="sexe" value="M"</pre>
   class="custom-control-input"/>
                      <label class="custom-control-label" for="sexe1">Masculin
   </label>
125
                    </div>
126
                    <div class="custom-control custom-radio custom-control-</pre>
   inline">
                      <input type="radio" id="sexe2" name="sexe" value="F"</pre>
   class="custom-control-input"/>
                      <label class="custom-control-label" for="sexe2">Féminin/
   label>
129
                    </div>
130
               </div>
131
             </di>
132
             <div class="form-group row">
133
               <label class="col-sm-4 col-form-label">Mot de passe: </label>
134
               <div class="col-sm-8">
135
                   <input type="password" id="inputPassword" data-minlength=</pre>
   "8" class="form-control" />
136
                    <div class="help-block with-errors">Minimum 8 caractères/
   div>
137
               </div>
138
             </div>
139
             <div class="form-group row">
```

```
<label class="col-sm-4 col-form-label">Confirmez le mot de
   passe: </label>
141
                <div class="col-sm-8">
142
                    <input type="password" class="form-control" name=</pre>
   "motdepasse" data-minlength="8" id="inputPasswordConfirm" data-match=
   "#inputPassword" data-match-error="Les deux mots de passe ne correspondent
   pas" />
                    <div class="help-block with-errors">Minimum 8 caractères/
   div>
144
               </div>
145
             </div>
146
147
              <div class="form-group row" >
148
                    <label class="col-sm-4">Photo:</label>
149
                    <div class="col-sm-8 image-viewer">
150
                  <span class="btn btn-primary btn-sm" onclick="choisirPhoto</pre>
   (this); ">Choisir une image</span>
                  <img class="display-none img_apercu" style="width: 150px;</pre>
   height: auto;" />
152
                 <span class="file-name"></span>
153
                  <a id="file" class="btn btn-danger display-none btn-remove"</pre>
   onclick="removeFile(this);">X</a>
                      <input type="file" class="form-control" style="display:</pre>
   none; " name ="photo" id="photo" onchange="show_apercu(this);"
   "image/*"/>
155
                    </div>
156
             </div>
157
             <div id="result" style="padding: 5px;text-align: center;"></div>
158
             <hr />
             <center><button type="submit" class="btn btn-danger btn-sm">
   Valider</button></center>
160
         </form>
161
       </center>
162
               <script type="text/javascript">
163
     $(function(){
164
        $('#myform').submit(function(e) {
165
166
         e.preventDefault();
167
         var \$form = \$(this);
168
         var formdata = (window.FormData) ? new FormData($form[0]) : null;
         var donnees = (formdata !== null) ? formdata : $form.serialize();
170
         $.ajax({
171
               url: $form.attr("action"),
172
               type: 'post',
173
               contentType: false, // obligatoire pour de l'upload
174
               processData: false, // obligatoire pour de l'upload
175
               data: donnees,
176
               success: function (data) {
177
                    var rep= JSON.parse(data);
178
                    if(rep.ok){
179
                        document.getElementById('myform').reset();
     réinitialisation du formulaire
180
                        $('#myform').find('.img_apercu').hide();
181
                        $('#myform').find('.file-name').hide();
182
                        $('#myform').find('.btn-remove').hide();
183
                        $("#result").html(rep.msg);
184
                        $("#result").show(500).delay(5000).hide(500);
185
                    }else{
186
                        $("#result").html(rep.msg);
187
                        $("#result").show(500).delay(7000).hide(500);
188
                    }
189
```

```
190 error:function(){alert("echec");}

191 });

192 });

193 });

194 </script>

195 </body>

196 </html>
```

### Le contenu de la page inscription.php

```
1 <?php
 2 require('config.php');
 3 require("fonctions.php");
 5 $repertoire="users/";
6 if (!is_dir($repertoire)) {
     mkdir("users");
8 }
9 //Enregistrement de l'image
10 $photo=enregFichier("photo", $repertoire);
12 //Enregistrement des informations dans la base de données
13 if (isset ($_POST) && !empty ($_POST)) {
14
    try {
15
          $connexion = new PDO($url,$user,$password);
16
          if($connexion){
17
               $requete = $connexion->prepare("INSERT INTO etudiants
18
                                                SET nom=:nom,
19
                                                     matricule=:matricule,
20
                                                     tel=:tel,
21
                                                     email=:email,
                                                     ville=:ville,
23
                                                     sexe=:sexe,
24
                                                     date_naissance=:
  date_naissance,
                                                    motdepasse =PASSWORD(:
  motdepasse),
                                                    photo=:photo
27
                                                ");
2.8
               $requete->execute(array("nom"=>$_POST["nom"],
29
                                        "matricule"=>$_POST["matricule"],
30
                                        "tel"=>$_POST["tel"],
31
                                        "email"=>$_POST["email"],
                                        "ville"=>$_POST["ville"],
32.
33
                                        "sexe"=>$_POST["sexe"],
34
                                        "date_naissance"=>$_POST[
  "date_naissance"],
                                        "motdepasse"=>$_POST["motdepasse"],
36
                                        "photo"=>$photo,
37
38
                                  );
39
               if($requete->rowCount()>0){
                   echo json_encode(array('ok'=> true,'msg'=> "Inscription
  effectuée avec succès"));
              }else {
42
                   echo json_encode(array('ok'=> false,'msg'=> "Inscription
  échouée"));
44
               $requete->closeCursor();//on ferme la connexion
```

```
46
47
          }else {
48
              echo json_encode(array('ok'=> false,'msg'=> "Connexion échouée"
49
          }
50
      }catch(Exception $e){
51
          // En cas d'erreur, on affiche un message et on arrête tout
52
          die('Erreur : '.$e->getMessage());
53
      }
54 }
55 ?>
```

#### Le contenu de la page etudiants.php

```
1 <!DOCTYPE HTML>
2 <html>
3 <head lang="fr">
4 <meta http-equiv="content-type" content="text/html" />
   <meta charset="utf-8"/>
      <meta name="viewport" content="width=device-width, initial-scale=1,</pre>
6
  shrink-to-fit=no"/>
      <script src="js/jquery-3.3.1.min.js" type="text/javascript"></script>
      <script type="text/javascript">
8
9
          function supEtudiant(ob,id){
10
               if(confirm("Confirmez vous la suppresion ?")){
11
                         $.ajax({
12
                           url: "suppression.php",
13
                           type: 'post',
14
                           data: "id="+id,
15
                           success: function (data) {
16
                                var rep= JSON.parse(data);
17
                                if(rep.ok){
18
                                    alert ("Suppresion effectuée avec succès");
19
                                    $(ob).parents('tr').remove();
2.0
                                }else{
2.1
                                   alert ("Suppression echouée");
22
23
                           },
24
                           error:function() {alert("echec");}
25
                       });
26
               }
27
          }
28
29
      </script>
      <style type="text/css">
30
31
          table {
32
              border-collapse; collapse;
33
          }
34
          tr:nth-child(odd){
35
              background-color: rgb(250,250,250);
36
          }
37
          td,th {
38
              font-size: 1.2em;
39
              padding: 5px 15px;
40
              border: 1px solid rgba(32,32,32,0.5);
41
          }
42
          h2 {
43
              text-align: center;
```

```
44
             margin: 25px 0px;
45
             color: rgb(0,0,150);
46
             background-color: aqua;
47
          .photo {
49
             height: 100px;
50
             width: auto;
51
52
          .btsup {
53
             padding: 2px 5px;
54
             font-size: 1.2em;
55
            cursor: pointer;
56
             color: white;
57
             text-decoration: none;
58
             background-color: red;
59
             border-radius: 5px;
60
         }
61
          .btsup:hover {
             box-shadow: 0px 2px 6px rgba(255,0,0,0.5);
63
             border-radius: 10px;
64
         }
65
      </style>
66 </head>
67 <body>
68
69 <?php
70 require('config.php');
71 try {
72
      $connexion = new PDO($url,$user,$password);
      if($connexion){
         $requete = $connexion->prepare("SELECT * FROM etudiants");
75
         $requete->execute();
76
          if ($requete->rowCount()>0) {
77
             echo "<center> <h2> Les étudiants sont : </h2>";
78
             echo "";
             echo "
79
80
                     nom
81
                    Tel
82.
                    Email
83
                    Matricule
84
                    Ville
85
                    Sexe
86
                    Date de Naissance 
87
                     Photo 
                     Supp 
                  ";
89
             while($data=$requete->fetch()){
90
91
                echo "";
                 echo "".$data["nom"]."";
92
93
                 echo "".$data["tel"]."";
                 echo "".$data["email"]."";
94
95
                 echo "".$data["matricule"]."";
96
                 echo "".$data["ville"]."";
97
                 echo "".$data["sexe"]."";
98
                 echo "".$data["date_naissance"]."";
99
                 echo "";
100
                 if(!empty($data["photo"])) {
```

```
echo "<img class='photo' src='users/".$data["photo"].
    "' />";
 102
                    }else echo "<img class='photo' src='users/user.png' />";
 103
                    echo "";
                    echo " <a class='btsup' onclick='supEtudiant(this,".</pre>
    $data['id'].");'>X</a> ";
                    echo "";
 105
 106
                }
 107
            }else {
 108
                echo "<h2> <center>Aucun étudiant inscrit </center></h2>";
 109
            }
 110
 111
            $requete->closeCursor();//on ferme la connexion
 112
 113
      }else {
 114
           echo "Connexion échouée";
 115
       }
 116 }catch (Exception $e) {
       // En cas d'erreur, on affiche un message et on arrête tout
 118
       die('Erreur : '.$e->getMessage());
 119 }
 120 ?>
 121 </body>
122 </html>
```

#### Le contenu de la page suppression.php

```
1 <?php
2 require('config.php');
3 if(isset($_POST) && !empty($_POST)){
    trv{
          $connexion = new PDO($url,$user,$password);
5
6
         if($connexion){
7
             $requete = $connexion->prepare("DELETE FROM etudiants WHERE
  id=:id");
             $requete->execute(array("id"=>$_POST["id"]));
9
             if($requete->rowCount()>0){
10
                 echo json_encode(array('ok'=> true,'msg'=> "Suppression
  effectuée avec succès!"));
11
             }else{
12
                  echo json_encode(array('ok'=> false,'msg'=> "Suppression
  échouée"));
13
14
             $requete->closeCursor();
15
         }else {
16
              echo json_encode(array('ok'=> false,'msg'=> "Connexion échouée"
          }
18
     }catch(Exception $e){
         echo json_encode(array('ok'=> false,'msg'=> $e->getMessage()));
20
21 }
22 ?>
```

#### Contenu du fichier fonctions.php

```
1 <?php
2
3 function getFileUniqueName($nomfichier,$repertoire) {
4  while(file_exists($repertoire.$nomfichier)){</pre>
```

```
$tab= getFileExt($nomfichier);
          $suffixe= rand(1,10000000);
          $nomfichier=$tab[0].$suffixe.".".$tab[1];
 8
9
     return $nomfichier;
10 }
11 function getFileExt($nomfichier) {
    $extensionPresumee = explode('.', $nomfichier);
13
      $extension= strtolower($extensionPresumee[count($extensionPresumee)-1
      $nom_sans_extention=$extensionPresumee[0];
      for ($i=1;$i<count($extensionPresumee)-1;$i++){</pre>
          $nom_sans_extention = $nom_sans_extention .".". $extensionPresumee[
16
17
      return array($nom_sans_extention,$extension);
19 }
20
21 function enregFichier($nom, $repertoire){
    if(isset($_FILES[$nom])&& !empty($_FILES[$nom]['name'])){
         $nomfichier= utf8_encode($_FILES[$nom]['name']);
          $tmp=$_FILES[$nom]['tmp_name'];
24
25
          if (is_uploaded_file($tmp))
           {
27
              if(file_exists($repertoire)){
28
                  if(file_exists($repertoire.$nomfichier)){
29
                      $nomfichier =getFileUniqueName($nomfichier,$repertoire
  );
31
                  if (move_uploaded_file($tmp,$repertoire.$nomfichier)){
32
                      return $nomfichier;
33
34
              }
35
           }
37
      return "";
38 }
39 ?>
```

Remarque: Nous remarquons que le plugin bootstrap validator gère mal les requêtes ajax

# **Exercice : Activité d'autoévaluation No 3**



Exercice
Pour exécuter des requêtes simples, on utilise la méthode () pour les sélections de données et la méthode () pour les insertion , mise à jour et suppression
Exercice
La syntaxe d'une requête SQL d'insertion est : table (nom_colonne_1, nom_colonne_2,) ('valeur 1', 'valeur 2',) ;  Exercice
L'avantage des requêtes préparées est :
<ul> <li>Qu'elles sont plus rapides</li> <li>Qu'elles offrent une meilleure sécurité contre injections SQL</li> </ul>
O qu'elles sont plus facile à écrire
O qu'elles sont plus persistante  Exercice
La syntaxe d'une requête SQL de sélection est : nom_du_champ nom_du_tableau  Exercice
La syntaxe d'une requête SQL de mise à jour est :
<pre>table colonne_1 = 'valeur 1', colonne_2 = 'valeur 2',</pre>
colonne_3 = 'valeur 3'
condition
Exercice
La syntaxe d'une requête SQL de suppression est : 'table' condition

# Alternative entre base de données et les fichiers



## 1. Choix entre base de données et fichiers

Les bases de données à l'instar de MySQL offrent un meilleur stockage et sécurité des informations et un accès très convivial. Toute fois, elles peuvent s'avérer dans certaines situations être un peu contraignantes. En effet,

- Une base de données client-serveur à l'instar de MySQL nécessite un téléchargement (des Mo) e une installation d'abord avec souvent quelques configurations.
- Si nous devons envoyer notre projet sur un autre ordinateur, il faudrait que ce dernier soit dans le même réseau que la base de données (sinon on ne pourra plus accéder à la base de données) ou bien il faudrait à nouveau installer le SGBD et récréer la base de données sur l'ordinateur cible.

De telles manipulations s'avère fastidieuses pour certaines applications (embarquées) qui ont besoin d'avoir la base de données intégrée en leur sein et transportable partout où se trouve l'application. Pour remédier à ce problème, à défaut d'utiliser des fichiers pour stocker les données, nous pouvons utiliser une *base de données SOLite*.

Une base de données SQLite est une sorte de fichier avec pour extension .sqlite qui est directement intégré dans le projet et qui permet de stocker et manipuler des données en utilisant des requêtes SQL.

Dans la suite de ce cours, nous allons apprendre de façon très succincte comment utiliser une base de données SQLite dans un projet PHP

## 2. Utilisation de la base de données SQLite

## Activation de SQLite

Avant d'utiliser SQLite en PHP, il faut d'abord activer l'extension php\_pdo\_sqlite.dll et php\_sqlite3.dll

- à partir de l'icône de wamp qui apparaît dans la barre des tâche dans le menu php>php extentions, il faut cocher php\_sqlite3 et php\_pdo\_sqlite.
- ou bien dans le fichier php.ini dans le repertoire wamp>bin>php il faut décommenter (enlever les points virgules) les lignes extension=php\_pdo\_sqlite.dll

### extension=php\_sqlite3.dll

Après il faut redémarrer wamp pour que les modifications soient prises en compte.

La manière d'écrire et d'exécuter des requêtes SQL est identique à ce que nous avons vu précédemment.

#### Création d'une base de données

Soit une base de données shop qu'on aimerait créer, on peut procéder comme suit :

```
1 <?php
2 $dbname='shop';
3 if(!class_exists('SQLite3'))
4   die("SQLite 3 n'est pas suppporté.");
5
6 $base=new SQLite3($dbname, 0666);
7 ?>
```

#### Création d'une table

Soit la table clients de champs id, nom, ville, age

```
1 <?php
  2 $dbname='shop';
  3 $mytable ="clients";
  5 if (!class_exists('SQLite3'))
  6 die("SQLite 3 NOT supported.");
  8 $base=new SQLite3($dbname, 0666);
 10 $query = "CREATE TABLE $mytable(
 ID bigint (20) NOT NULL PRIMARY KEY,
 12
             nom text NOT NULL,
 13
             ville VARCHAR(30),
 14
             age int(3)
 15
             ) ";
 17 $results = $base->exec($query);
 18 ?>
19
```

On pouvait utiliser la syntaxe SQL CREATE TABLE IF NOT EXISTS \$mytable, cette syntaxe évite de récréer une table qui existe déjà

## Insertion d'une information dans la table

```
1 <?php
2 $dbname='shop';
3 $mytable ="clients";
4
5 if(!class_exists('SQLite3'))
6  die("SQLite 3 NOT supported.");
7</pre>
```

```
8 $base=new SQLite3($dbname, 0666);
9 $id=1;
10 $nom="Yao";
11 $ville="Abidjan";
12 $age=21;
13 $query = "INSERT INTO clients(id, nom, ville, age) VALUES ($id, '$nom', '$ville ',$age)";
14
15 $results = $base->exec($query);
16 ?>
17
```

## Lire une information

```
1 <?php
 2 $dbname='shop';
3 $mytable ="clients";
5 if(!class_exists('SQLite3'))
6 die("SQLite 3 NOT supported.");
8 $base=new SQLite3($dbname, 0666);
9 $query = "SELECT * FROM $mytable";
10 $results = $base->query($query);
11 $row = $results->fetchArray();
13 if (count ($row) > 0)
14 {
15 echo "Nom : ". $row['nom']." <br/>";
16 echo "Ville: " . $row['ville']."<br/>";
17 echo "Age :". $row['age'];
18 }
19 ?>
```

## Voici quelques liens pour vous perfectionner en SQLite

- http://www.finalclap.com/faq/170-php-sqlite-pdo
- https://www.scriptol.fr/sql/tutoriel-sqlite.php
- https://www.scriptol.fr/sql/tutoriel-sqlite-table.php
- https://lehollandaisvolant.net/tuto/sqlite/
- http://php.net/manual/fr/ref.sqlite.php
- http://php.net/manual/fr/book.sqlite.php

# **Exercice : Activité d'autoévaluation No 4**



Exercice			
Une base de données SQLite est :			
0			
une base de données clientt-serveur directement intégré dans le projet et qui permet de stocker et manipuler des données en utilisant des requêtes SQL.			
0			
une sorte de fichier avec pour extension . <i>sqlite</i> qui est directement intégré dans le projet et qui permet de stocker et manipuler des données en utilisant des requêtes SQL.			
O un fichier de stockage d'informations accessible par le biais d'un moteur elastic search			
O une base de données MySQL plus réduite			
Exercice			
Pour utiliser une base de données MySQL, il faut activer dans PHP les extensions : (2 réponses)			
<pre>php_object_pdo</pre>			
□ php_pdo_sqlite			
□ php_connect_sqlite			
□ php_sqlite ou php_sqlite3			
Exercice			
Veuillez compléter le code ci dessous pour pour créer une base de donnée SQLite			
php</td			
\$db='ecole';			
if(! ('SQLite3'))			
die("SQLite 3 NOT supported.");			
\$base= ( , 0666);			

?>

## Exercice

Veuillez compléter le c	ode ci-dessous pour	r insérer une information o	dans une table SQLite:	
php</td <td></td> <td></td> <td></td> <td></td>				
\$dbname='ecole';				
(!	('SQLite3'	))		
die("SQLite 3 NOT supported.");				
\$base=	(	, 0666);		
\$id=3;				
<pre>\$nom="Soro";</pre>				
<pre>\$ville="Daloa";</pre>				
\$age=30;				
\$query = "		users(id,nom,ville,age)	(	,
,	',	)";		
\$base->	( );			
?>				