

# PHP /MYSQL

# Bases de Données et SGBD

- ❑ La **base de données** (BDD) est un système qui enregistre des informations.

Ce qui est très important ici, c'est que ces informations sont toujours **classées**.

- ❑ Les **SGBD (Système de Gestion des Bases de Données)** sont les programmes qui se chargent du stockage des données. Les plus connus sont, pour rappel :
  - **MySQL** : libre et gratuit, c'est probablement le SGBD le plus connu. Nous l'utiliserons dans cette partie ;
  - **PostgreSQL** : libre et gratuit comme MySQL, avec plus de fonctionnalités mais un peu moins connu ;
  - **SQLite**: libre et gratuit, très léger mais très limité en fonctionnalités ;
  - **Oracle** : utilisé par les très grosses entreprises ; sans aucun doute un des SGBD les plus complets, mais il n'est pas libre et on le paie le plus souvent très cher ;

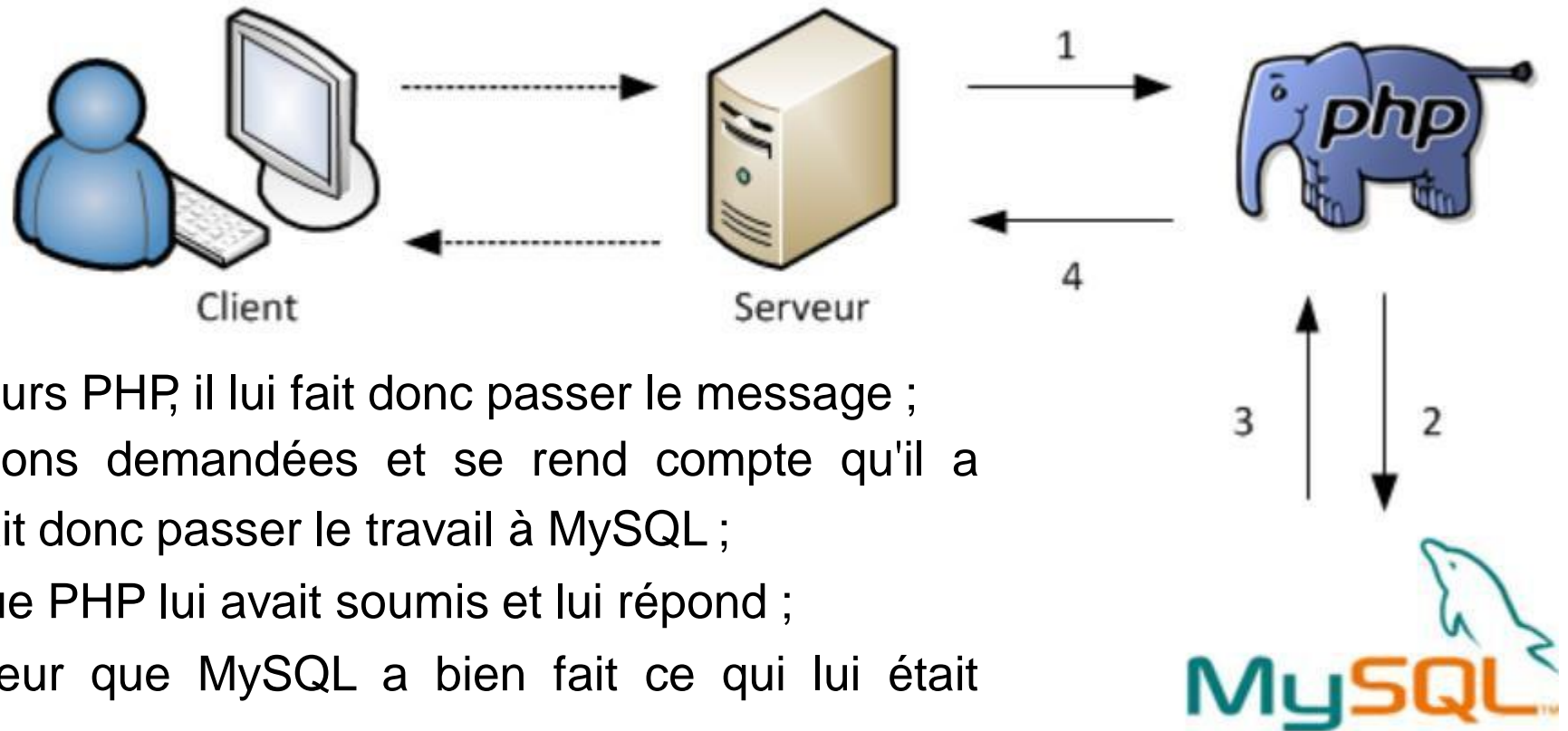
# Communication avec SGBD

## ❑ Les ordres au SGBD sont donnés en langage SQL :

- Vous allez devoir communiquer avec le SGBD pour lui donner l'ordre de récupérer ou d'enregistrer des données. Pour lui « parler », on utilise le langage SQL.
- La bonne nouvelle, c'est que le langage SQL est un standard, c'est-à-dire que quel que soit le SGBD que vous utilisez, vous vous servirez du langage SQL.
- La mauvaise, c'est qu'il y a en fait quelques petites variantes d'un SGBD à l'autre, mais cela concerne généralement les commandes les plus avancées.

# PHP et MySQL

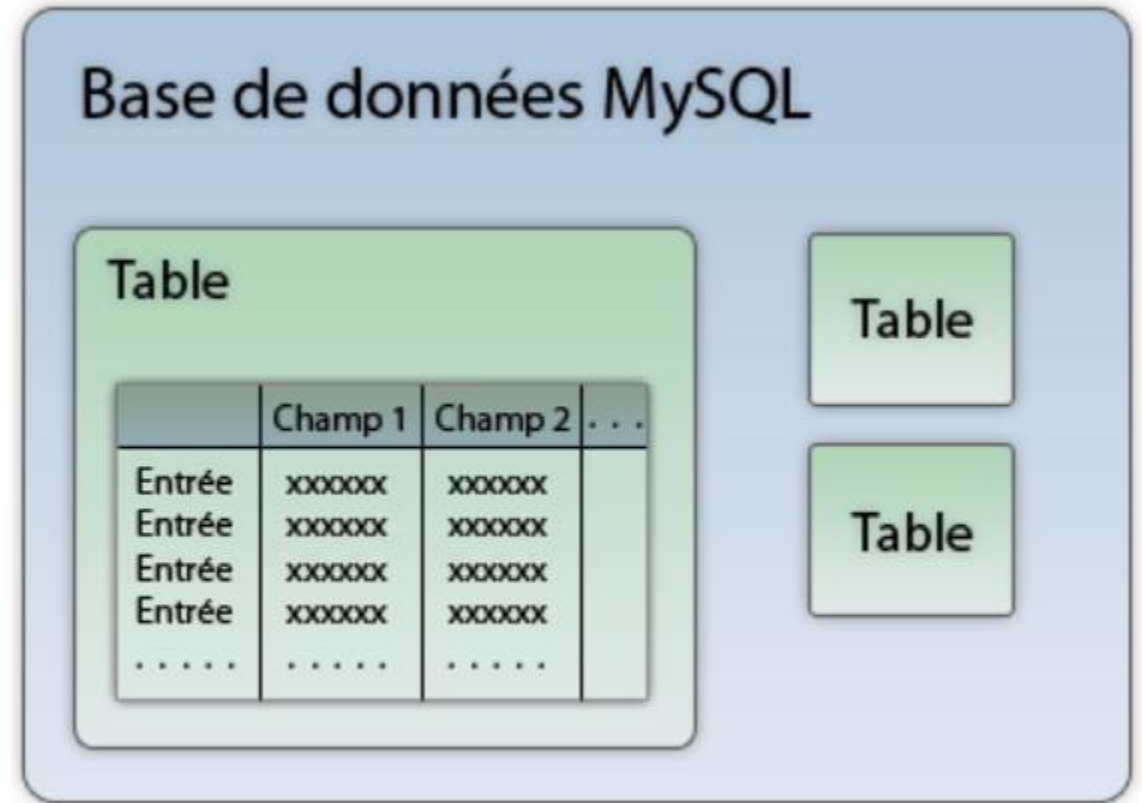
❑ Le PHP est l'intermédiaire entre vous et mysql



1. Le serveur utilise toujours PHP, il lui fait donc passer le message ;
2. PHP effectue les actions demandées et se rend compte qu'il a besoin de MySQL. Il fait donc passer le travail à MySQL ;
3. MySQL fait le travail que PHP lui avait soumis et lui répond ;
4. PHP renvoie au serveur que MySQL a bien fait ce qui lui était demandé.

# Structure d'une BDD

- ❑ Une base de données est un ensemble de tables liées.
- ❑ Chaque table contient un ensemble de champs.
- ❑ Au lieu maintenant de stocker nos enregistrements dans des fichiers nous allons directement les stockées dans les tables qui leurs correspondent.



Organisation d'une base de données MySQL

# phpMyAdmin

The screenshot displays the phpMyAdmin interface in a web browser. The browser's address bar shows `http://localhost/phpmyadmin/index.php`. The interface is for a MySQL server (3306). On the left, a sidebar lists database structures: `Nouvelle base de données`, `base_php`, `information_schema`, `ma`, `mabase`, `mysql`, `performance_schema`, and `sys`. The main content area is divided into several panels:

- Paramètres généraux**: Includes a link to `Modifier le mot de passe` and a dropdown for `Interclassement pour la connexion au serveur` set to `utf8mb4_unicode_ci`.
- Paramètres d'affichage**: Includes a dropdown for `Langue - Language` set to `Français - French`, a dropdown for `Thème` set to `pmahomme`, and a `Taille du texte` dropdown set to `82%`. A link for `Plus de paramètres` is also present.
- Serveur de base de données**: Lists server details:
  - Serveur : MySQL (127.0.0.1 via TCP/IP)
  - Type de serveur : MySQL
  - Connexion au serveur : SSL n'est pas utilisé
  - Version du serveur : 5.7.24 - MySQL Community Server (GPL)
  - Version du protocole : 10
  - Utilisateur : root@localhost
  - Jeu de caractères du serveur
- Serveur Web**: Lists web server details:
  - Apache/2.4.37 (Win64) PHP/7.2.14
  - Version du client de base de données : 5.0.12-dev - 20150407 - 38fea24f2847fa7519001
  - Extension PHP : mysqli
  - Version de PHP : 7.2.14
- phpMyAdmin**: Lists version and links:
  - Version : 4.8.4
  - Documentation
  - Site officiel
  - Contribuer

A **WAMP SERVER 3.1.7** status overlay is visible on the right side of the interface. It includes the text `Made in France by Otomatic` and a table of installed services:

WAMP SERVER 3.1.7	
Localhost	
phpMyAdmin	4.8.4
Adminer	4.7.0
Yos VirtualHosts	
Répertoire www	
Apache	2.4.37
PHP	7.2.14
MySQL	5.7.24
MarjaDB	10.3.12
3.1.7 - 64bit - Services	
Démarrer les services	
Arrêter les services	
Redémarrer les services	

# phpMyAdmin

## 1- Création d'une base de données

**Création d'une base de données :**

Nom de base de donnée:  Créer

Base de données	Interclassement	Action
<input type="checkbox"/> base_php	utf8_general_ci	Vérifier les privilèges
<input type="checkbox"/> information_schema	utf8_general_ci	Vérifier les privilèges
<input type="checkbox"/> mabase	utf8mb4_bin	Vérifier les privilèges
<input type="checkbox"/> ma_base	utf8_bin	Vérifier les privilèges
<input type="checkbox"/> ma_base_php	utf8_bin	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"/> sys	utf8_general_ci	Vérifier les privilèges
<b>Total: 8</b>	<b>latin1_swedish_ci</b>	



# phpMyAdmin

## 2- Création des tables

Serveur courant : MySQL

Récentes Préférences

Nouvelle base de données

base\_php

Nouvelle table

client

compte

information\_schema

ma

mabase

mysql

performance\_schema

sys

Filtres

Contenant le mot :

Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> client	★ Parcourir Structure Rechercher Insérer Vider Supprimer	2	MyISAM	utf8_general_ci	2,1 kio	-
<input type="checkbox"/> compte	★ Parcourir Structure Rechercher Insérer Vider Supprimer	2	MyISAM	utf8_general_ci	3,1 kio	-
2 tables	Somme	4	MyISAM	utf8_general_ci	5,1 kio	0 o

↑ ☐ Tout cocher Avec la sélection :

Imprimer Dictionnaire de données

Nouvelle table

Nom : Nombre de colonnes : 4

Exécuter



# phpMyAdmin

## 2- Création des tables

Nom de table:  x Ajouter  colonne(s) Exécuter

Structure ?							
Nom	Type ?	Taille/Valeurs* ?	Valeur par défaut ?	Interclassement	Attributs	Null	Index
<input type="text"/>	INT <input type="button" value="v"/>	<input type="text"/>	Aucun(e) <input type="button" value="v"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	---
<input type="text"/>	INT <input type="button" value="v"/>	<input type="text"/>	Aucun(e) <input type="button" value="v"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	---
<input type="text"/>	INT <input type="button" value="v"/>	<input type="text"/>	Aucun(e) <input type="button" value="v"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	---
<input type="text"/>	INT <input type="button" value="v"/>	<input type="text"/>	Aucun(e) <input type="button" value="v"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	---

Commentaires de table :

Interclassement :

Moteur de stockage : ?

Définition de PARTITION : ?

Partitionner par :  ( Expression ou liste de co )

Partitions :

Aperçu SQL Enregistrer

# phpMyAdmin

## 3- Insertion/modification/suppression d'enregistrements

The screenshot shows the phpMyAdmin interface. On the left is a sidebar with a tree view of databases and tables. The main area displays the 'client' table in the 'base\_php' database. A green message bar at the top of the main area indicates that 2 lines are displayed. Below this, a SQL query editor shows the query: `SELECT * FROM `client``. To the right of the query editor is a checkbox for 'Profilage'. Below the query editor, there are controls for 'Tout afficher', 'Nombre de lignes' (set to 25), and 'Filtrer les lignes'. Below these controls is a '+ Options' section with a table view. The table has columns: nom, prenom, adresse, and CIN. The first row of data is: Client1, Client1, Adresse, EE123456. At the bottom of the table view, there are checkboxes for 'Tout cocher' and 'Avec la sélection' followed by 'Éditer', 'Copier', and 'Supprimer' buttons.

phpMyAdmin

Serveur courant : MySQL

Récentes Préférées

Nouvelle base de données  
base\_php  
Nouvelle table  
client  
compte  
information\_schema  
ma  
mabase  
mysql  
performance\_schema

Serveur: MySQL:3306 » Base de données: base\_php » Table: client

Parcourir Structure SQL Rechercher Insérer Exporter

✓ Affichage des lignes 0 - 1 (total de 2, traitement en 0,0000 seconde(s).)

`SELECT * FROM `client``

☐ Profilage [Éditer]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes : Chercher dans c

+ Options

	nom	prenom	adresse	CIN
<input type="checkbox"/> Éditer Copier Supprimer	Client1	Client1	Adresse	EE123456

↑ ☐ Tout cocher Avec la sélection : Éditer Copier Supprimer

# phpMyAdmin

## 4- Ordre SQL

← Serveur: MySQL:3306 » Base de données: base\_php » Table: client

Parcourir Structure SQL Rechercher Insérer Exporter Importer Privilèges Opérations Plus

Exécuter une ou des requêtes SQL sur la table « base\_php.client »:

```
1 DELETE FROM `client` WHERE 0
```

Colonnes

nom  
prenom  
adresse  
CIN

<<

SELECT \* SELECT INSERT UPDATE DELETE Effacer Format

Récupérer la requête auto-sauvegardée

☐ Lier les paramètres

[ Délimiteur ; ] ☒ Afficher à nouveau la requête après exécution ☐ Conserver la boîte de requêtes ☐ ROLLBACK à la fin

☒ Activer la vérification des clés étrangères

Simuler la requête

Exécuter

# Etablir la connexion entre PHP et la BDD

## ❑ L'extension PDO :

- C'est un outil complet qui permet d'accéder à n'importe quel type de base de données.
- Le gros avantage de PDO est que vous pouvez l'utiliser de la même manière pour vous connecter à MySQL ainsi qu'à n'importe quel autre type de base de données (PostgreSQL, Oracle...).

# Etablir la connexion entre PHP et la BDD

```
try{
$bdd= new PDO('mysql:host=localhost; dbname=base_php; charset=utf8','root','');
}
catch(Exception $e)
{
    die('erreur: '.$e->getMessage());
}
```

La ligne de code crée un objet \$bdd. C'est un objet qui représente la connexion à la base de données. On crée la connexion en indiquant dans l'ordre les paramètres suivants:

- le nom d'hôte (localhost) ;
- la base de données (base\_php) ;
- le login (root) ;
- le mot de passe (sous WAMP il n'y a pas de mot de passe, j'ai donc mis une chaîne vide, sous MAMP le mot de passe est root).

# Récupérer des données

```
$query= $bdd->query("Select CIN from client;");  
while($data=$query->fetch())  
{  
    echo $data['CIN'];  
}  
  
$query->closeCursor();
```

- La méthode **query** est appelé via l'objet **\$bdd** récupéré comme résultat de la connexion via PDO.
- Cette méthode permet d'interroger votre base de données en utilisant la requête SQL spécifiée.
- La méthode **fetch** permet de récupérer le résultat lu ligne par ligne.
- **\$data** est un tableau associatif qui contient tous les enregistrements obtenus via la requête SQL lancée.
- Ces cases disposent du même nom spécifié dans votre table.
- Ici **\$data** est un tableau à une colonne nommée CIN.

# Récupérer des données

```
$q1=$bdd->query("select nom, prenom from client, compte where client.CIN= compte.CIN and compte.montant<0;");

while($d1=$q1->fetch())
{
    echo "
        <table bgcolor='#FFF0F5' border=0 >
        <tr >
        <td width='170px'>Nom: ".$d1['nom']."</td>
        <td width='153px'>Prenom: ".$d1['prenom']."</td>
        </tr> </table>";
}

$q1->closeCursor();
```

- Le code suivant permet de sélectionner le **nom**, le **prénom** des **clients** dont le **montant** dans leurs **comptes** bancaires **est inférieur à 0**. (des débiteurs).
- Les résultats obtenus sont affichés sous format d'un tableau html.
- **\$d1** contient deux colonnes **nom** et **prénom** et l'ensemble des lignes résultats de la requête.
- Chaque ligne de **\$d1** sera affichée au niveau d'un **tr** et deux **td** du tableau html.



# Ecrire des données en utilisant des marqueurs

```
//  
$query=$bdd->prepare("insert into client(nom,prenom,adresse,CIN) values(?,?,?,?);");  
$query->execute(array($nom,$prenom,$adresse,$CIN));  
$query->closeCursor();
```

- Les marqueurs nous permettent de signaler à la requête à exécuter qu'ils seront remplacés par la valeur des variables passées au niveau de la méthode ***execute***.
- Ce bout de code permet d'insérer un nouveau client dont les informations sont stockées dans les variables php ***\$nom***, ***\$prenom***, ***\$adresse*** et ***\$CIN***.

Ces données peuvent être récupérées via ***\$\_POST*** au niveau d'un formulaire.

# Modifier des données en utilisant des marqueurs

```
$query= $bdd->prepare("Update compte set montant = ? where numCompte=?;");  
$query->execute(array($m,$numC));  
$query->closeCursor();
```

- Ce bout de code permet de modifier le **montant** du compte dont le **numéro** est stocké dans **\$numC**.
- La nouvelle valeur du **montant** est stockée dans **\$m**.

# Supprimer des données en utilisant des marqueurs

```
...  
$q1=$bdd->prepare("delete from compte where CIN=?;");  
$q1->execute(array($CIN));  
$q1->closeCursor();
```

- Ce bout de code permet de supprimer le **compte** du client dont le **CIN** est **\$CIN**.