



ISN – Informatique et Sciences du Numérique

BASE DE DONNEES – MySQL





- 1 – BASES DE DONNEES : INTRODUCTION 3**
 - 1.1 – BASES DE DONNEES 3
 - 1.2 – SYSTEME DE GESTION DE BASES DE DONNEES SDBG 3
 - 1.3 – LANGAGE SQL 4
 - 1.4 – LIAISON PHP ET MYSQL4 4

- 2 – MYSQL 5**
 - 2.1 – CREATION D’UNE BASE DE DONNEES 5
 - 2.1.1 – phpMyAdmin 5
 - 2.1.2 – Créer une table 5
 - 2.1.3 – Insérer une entrée dans la table 7
 - 2.2 – LANGAGE SQL 8
 - 2.2.1 – Requête dans phpMyAdmin 8
 - 2.2.2 – Sélectionner des données dans une base 9
 - 2.2.3 – Insérer des données dans la base 11
 - 2.2.4 – Modifier des données dans la base 11
 - 2.2.5 – Supprimer des données dans la base 12
 - 2.2.6 – Quelques fonctions 12

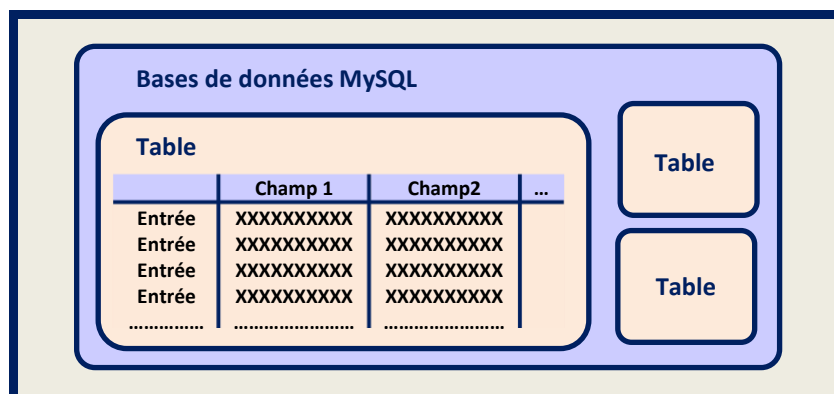
- 3 – MYSQL ET PHP 14**
 - 3.1 – CONNEXION A LA BASE DE DONNEES 14
 - 3.1.1 – Extension PDO 14
 - 3.1.2 – Connexion à MySQL avec PDO 15
 - 3.2 – REQUETE DE LECTURE D’UNE BASE DE DONNEES 15
 - 3.3 – REQUETE DE MODIFICATION D’UNE BASE DE DONNEES 17

1 – BASES DE DONNEES : INTRODUCTION

1.1 – BASES DE DONNEES

Une **base de données (BDD)** est un **ensemble de données stockées** de façon **organisée et hiérarchisée** dans un dispositif informatique. Cette organisation et hiérarchisation des données permet **une manipulation et un stockage efficace** de très grandes quantités d'informations.

Une base de données contient plusieurs **tables** constituées de plusieurs lignes appelées « **Entrées** » et plusieurs colonnes appelées « **champs** ».



Exemple : Table visiteurs

Cette table visiteur est constituée de 4 champs qui sont l'identifiant ID de l'entrée, le « Pseudo », l'adresse mail et de l'âge de chaque visiteur.

ID	Pseudo	E-mail	Age
1	jpaul	jp.dupont@gmail.com	42
2	lolo	laurent_martin@free.fr	36
3	bibou	bibou75@laposte.net	24
4	geronimo	Jerome228@hotmail.com	28
5

1.2 – SYSTEME DE GESTION DE BASES DE DONNEESDBG

Un **Système de Gestion de Bases de Données (SGBD)** est un programme qui permet **de créer et gérer** une base de données. Les SGBD les plus connus sont :

- **MySQL** : SGBD libre et gratuit qui fait partie des logiciels de gestion les plus utilisés au monde autant par le grand public (applications web principalement) que par des professionnels.
- **PostgreSQL** : SGBD libre et gratuit comme MySQL présentant beaucoup plus de fonctionnalités que MySQL mais moins utilisé.
- **SQLite** : SGBD libre et gratuit, très léger mais très limité en fonctionnalités.
- **Oracle** : SGBD très complet et utilisé par les très grandes entreprises mais il s'agit un logiciel payant.
- **Microsoft SQL Server** : le SGBD de Microsoft.



1.3 – LANGAGE SQL

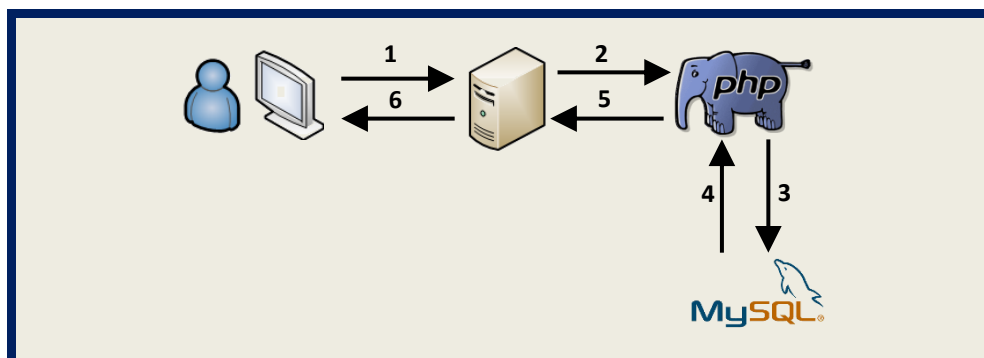
Le langage **SQL** (Structured Query Language) est un langage informatique normalisé servant à effectuer des **opérations sur des bases de données**. Il permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données.

La syntaxe des instructions SQL est assez simple. Les instructions s'écrivent d'une manière qui **ressemble à celle de phrases ordinaires en anglais**.

Exemple : sélection dans une table de la base de données

```
SELECT name, service
FROM employes
WHERE statut = 'stagiaire'
ORDER BY name;
```

1.4 – LIAISON PHP ET MYSQL4



- 1 : Le navigateur Web d'un utilisateur envoie une requête http.
- 2 : Le serveur reçoit la requête, récupère le fichier « php » et le passe au moteur PHP afin qu'il soit traité.
- 3 : Le moteur PHP commence à analyser le script. A l'intérieur de ce script se trouve une commande permettant de se connecter à la base de données et d'exécuter une requête. PHP ouvre une connexion vers le serveur de la base de données et transmet la requête appropriée.
- 4 : Le serveur de la base de données reçoit la requête de base de données et la traite, puis renvoie les résultats au moteur PHP.
- 5 : Le moteur PHP termine l'exécution du script et envoie le fichier HTML obtenu au serveur Web.
- 6 : Le serveur Web transmet la page HTML au navigateur, pour que l'utilisateur puisse voir le résultat.

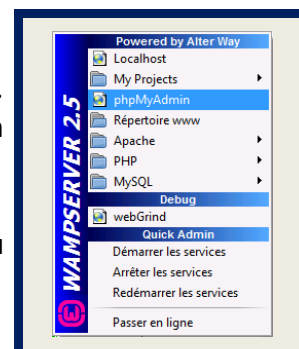
2 – MYSQL

2.1 – CREATION D'UNE BASE DE DONNEES

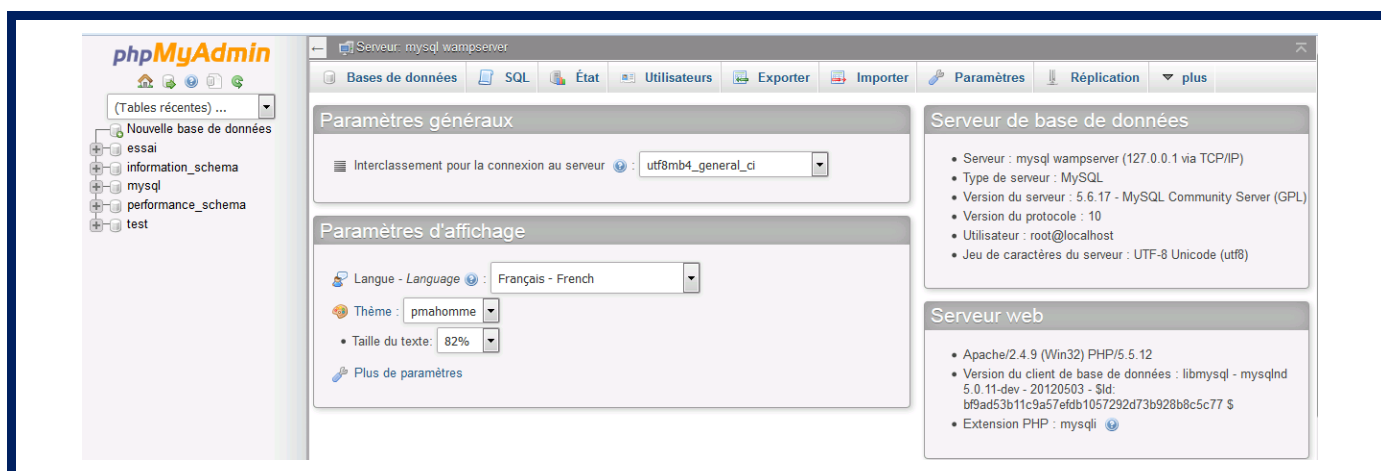
2.1.1 – phpMyAdmin

L'interface WEB **phpMyAdmin** permet de gérer les bases de données MySQL. Ecrit en PHP, il s'appuie sur le serveur Apache. Elle donc intégrée à la distribution Wamp.

Lorsque le serveur Wamp est lancé, phpMyAdmin est accessible à partir du menu de Wampserver.



Lorsque l'on clique sur phpMyAdmin, la page PHP suivante s'ouvre :



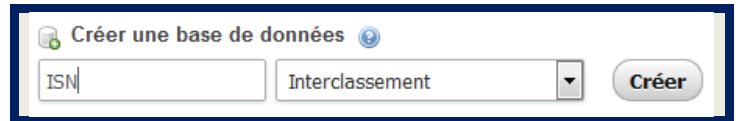
2.1.2 – Créer une table

Le but de ce paragraphe est de créer une base de données pour le site WEB « ISN au Lycée Gustave Eiffel ». Elle sera constituée pour l'instant d'une table « visiteurs » présentée ci-dessous.

ID	Pseudo	E-mail	Age
1	jpaul	jp.dupont@gmail.com	42
2	lolo	laurent_martin@free.fr	36
3	bibou	bibou75@laposte.net	24
4	geronimo	jerome228@hotmail.com	28

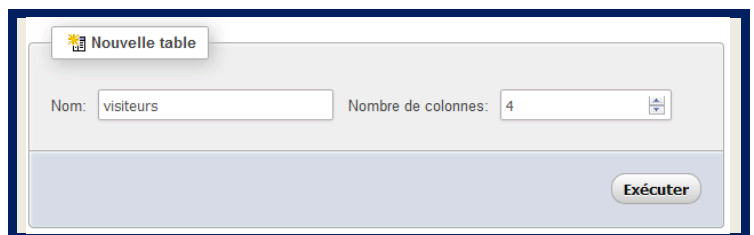
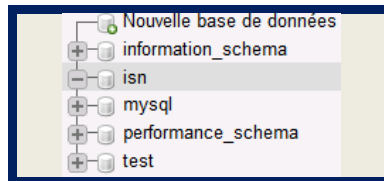
Créer une base de données :

- Cliquer sur « Nouvelle base de données ».
- Entrer le nom de la base de données puis cliquer sur « Créer ».

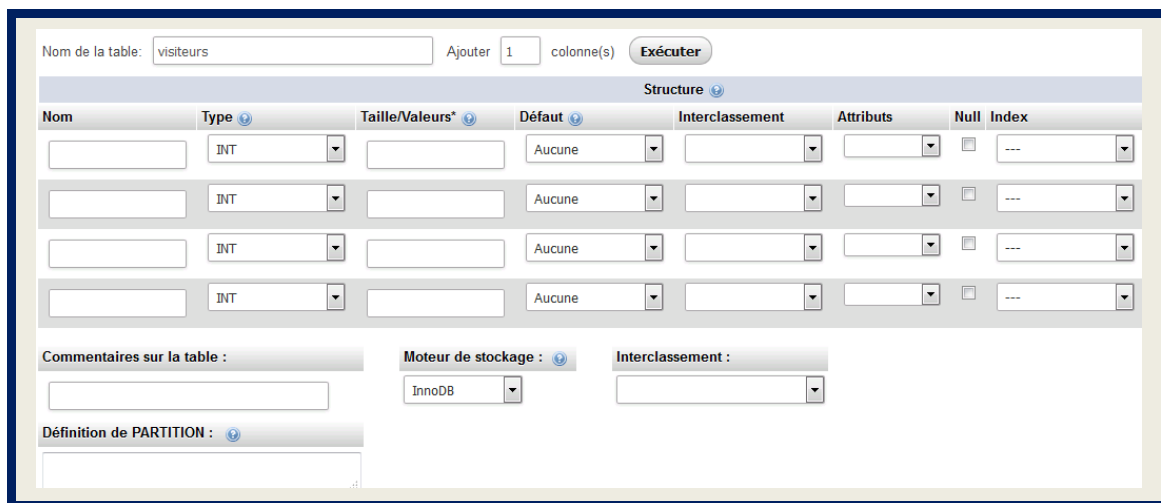


Créer une table :

- Cliquer sur « Nouvelle base de données ».
- Entrer le nom de la table en indiquant le nombre de champs. Puis cliquer sur « Exécuter ».



La page suivante permet de paramétrer les différents champs :



Il n'est pas nécessaire de renseigner tous les items. Il faut cependant compléter les sections suivantes :

Section	Description
Nom	Permet de définir le nom du champ
Type	Permet de définir le type de données que va stocker le champ : entiers, texte, VARCHAR (chaîne de caractère de taille variable) date...
Taille/Valeurs	Permet d'indiquer la taille maximale du champ. Cette section est utile pour le type VARCHAR notamment, afin de limiter le nombre de caractères autorisés
Index	Active l'indexation du champ afin qu'il soit adapté aux requêtes de recherche. Le plus souvent, on utilise l'index PRIMARY sur les champs de type « id ».
AI	Auto-Increment : permet au champ de s'incrémenter tout seul à chaque nouvelle entrée. On l'utilise fréquemment sur les champs de type « id ».

MySQL propose une quantité très importante de types de données :

Type	Description
NUMERIQUE	Nombres : entiers (INT), petits nombres entiers (TINYINT), grands nombres entiers (BIGINT), booléens (BOOLEAN), nombre binaire (BIT), nombres décimaux (DECIMAL, FLOAT,...), etc.
DATE et TEMPS	De nombreux types différents permettent de stocker une date (DATE, YEAR), une heure (TIME, TIMESTAMP), ou les deux à la fois (DATETIME).
CHAINE DE CARACTERES	De nombreux types différents permettent de stocker une chaînes de caractères de longueur fixe (CHAR) ou variable (VARCHAR) ou un texte de différentes longueurs (TINYTEXT, TEXT, ..., LONGTEXT).
SPATIAL	Données utiles pour la cartographie.

- Compléter les différentes sections de la table « visiteurs » puis cliquer sur « Sauvegarder ».

- Cliquer sur « Structure » pour visualiser la structure de la table.

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	<u>id</u>	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
2	pseudo	varchar(8)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
3	e_mail	varchar(256)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes
4	age	int(11)			Non	Aucune		Modifier Supprimer Primaire Unique Index Spatial Texte entier Valeurs distinctes

Le champ « id » est souligné car il s’agit de la clé primaire.

2.1.3 – Insérer une entrée dans la table

- Cliquer sur « Insérer » puis compléter les champs correspondant au premier visiteur (id = 1). Cliquez ensuite sur « Exécuter ».

Il n’est pas nécessaire de remplir le champ « id » car la valeur est automatiquement calculée et incrémentée grâce à l’option « Auto_Increment ».

- Réaliser le même travail pour les trois autres visiteurs. Puis cliquer sur « Afficher » pour voir la table.

	id	pseudo	e_mail	age
<input type="checkbox"/>	1	jpaul	jp.dupont@gmail.com	42
<input type="checkbox"/>	2	lolo	laurent_martin@free.fr	36
<input type="checkbox"/>	3	bibou	bibou75@laposte.net	24
<input type="checkbox"/>	4	geronimo	jerome228@hotmail.com	28

2.2 – LANGAGE SQL

2.2.1 – Requête dans phpMyAdmin

Pour effectuer une requête vers la base de données à l'aide de phpMyAdmin, il faut cliquer sur l'onglet « SQL ». La fenêtre suivante s'ouvre :

Exécuter une ou des requêtes SQL sur la base isn:

```
1 SELECT * FROM `visiteurs` WHERE 1
```

Colonnes
id
pseudo
e_mail
age

SELECT * SELECT INSERT UPDATE DELETE Vider

[Délimiteur :] Afficher à nouveau la requête après exécution Conserver la boîte de requêtes Exécuter

Exemple : Zone de texte – Formulaire.php

```
SELECT * FROM `visiteurs` WHERE 1
```

Cette requête signifie « **Afficher tout le contenu de la table 'visiteurs'** ». L'exécution de cette requête donne le résultat suivant :

Nombre de lignes : 25

Trier sur l'index: Aucune

+ Options

	id	pseudo	e_mail	age
<input type="checkbox"/>	1	jpaul	jp.dupont@gmail.com	42
<input type="checkbox"/>	2	lolo	laurent_martin@free.fr	36
<input type="checkbox"/>	3	bibou	bibou75@laposte.net	24
<input type="checkbox"/>	4	geronimo	jerome228@hotmail.com	28

Tout cocher Pour la sélection :

Nombre de lignes : 25

2.2.2 – Sélectionner des données dans une base

Pour sélectionner des données dans une base il faut utiliser la requête « SELECT ». Cette requête comporte trois clauses permettant de rechercher et d'ordonner les données.

WHERE :

La clause « **WHERE** » ajoute une condition à la recherche. Le mot clé « **LIKE** » permet d'ajuster la recherche autour d'une chaîne de caractères.

Exemples : SELECT

```
SELECT * FROM visiteurs WHERE pseudo = "lolo"
```

Permet d'afficher le visiteur dont le pseudo est « lolo » :

	id	pseudo	e_mail	age
<input type="checkbox"/>	2	lolo	laurent_martin@free.fr	36

```
SELECT * FROM visiteurs WHERE age < 30
```

Permet d'afficher la liste des visiteurs dont l'âge est inférieur à 30 :

	id	pseudo	e_mail	age
<input type="checkbox"/>	3	bibou	bibou75@laposte.net	24
<input type="checkbox"/>	4	geronimo	jerome228@hotmail.com	28

```
SELECT * FROM visiteurs WHERE pseudo LIKE "b%"
```

Permet d'afficher la liste des visiteurs dont le pseudo commence par le caractère « b ».

	id	pseudo	e_mail	age
<input type="checkbox"/>	3	bibou	bibou75@laposte.net	24

```
SELECT * FROM visiteurs WHERE pseudo LIKE "_e%"
```

Permet d'afficher la liste des visiteurs dont le pseudo a comme second caractère le caractère « e ».

	id	pseudo	e_mail	age
<input type="checkbox"/>	4	geronimo	jerome228@hotmail.com	28

```
SELECT * FROM visiteurs WHERE e_mail LIKE "%gmail%"
```

Permet d'afficher la liste des visiteurs dont l'adresse mel contient la chaîne de caractères « gmail ».

	id	pseudo	e_mail	age
<input type="checkbox"/>	1	jpaul	jp.dupont@gmail.com	42

**ORDER BY :**

La clause « **ORDER BY** » spécifie dans quel ordre sera effectué le tri du résultat. Les mots clés « **ASC** » et « **DESC** » précisent l'ordre croissant et décroissant. Par défaut le tri aura lieu dans l'ordre croissant.

Exemples : ORDER BY

```
SELECT * FROM visiteurs ORDER BY pseudo ASC
```

Permet d'afficher la liste des visiteurs ordonnée par liste alphabet croissante des pseudos :

	id	pseudo	e_mail	age
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	3	bibou	bibou75@laposte.net	24
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	4	geronimo	jerome228@hotmail.com	28
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	1	jpaul	jp.dupont@gmail.com	42
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	2	lolo	laurent_martin@free.fr	36

```
SELECT * FROM visiteurs WHERE age < 30 ORDER BY age DESC
```

Permet d'afficher la liste, ordonnée par âge décroissant, des visiteurs dont l'âge est inférieur à 30 :

	id	pseudo	e_mail	age
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	4	geronimo	jerome228@hotmail.com	28
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	3	bibou	bibou75@laposte.net	24

LIMIT :

La clause « **LIMIT** » est utilisée pour spécifier le nombre maximum de résultats que l'on souhaite obtenir. Si un seul chiffre est indiqué, il s'agit du nombre maximal de résultats. Si deux chiffres sont indiqués séparés par une virgule, le premier chiffre constitue l'offset, c'est-à-dire à partir de laquelle entrée sera comptabilisé le nombre maximal de résultats.

Cette clause doit être utilisée avec la clause « **ORDER BY** ».

Exemples : LIMIT

```
SELECT * FROM visiteurs ORDER BY id ASC LIMIT 2
```

Permet d'afficher une liste 2 visiteurs ordonnés par « id » croissant :

	id	pseudo	e_mail	age
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	1	jpaul	jp.dupont@gmail.com	42
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	2	lolo	laurent_martin@free.fr	36

```
SELECT * FROM visiteurs ORDER BY id ASC LIMIT 2,2
```

Permet d'afficher une liste 2 visiteurs à partir de la seconde entrée (non incluse) :

	id	pseudo	e_mail	age
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	3	bibou	bibou75@laposte.net	24
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	4	geronimo	jerome228@hotmail.com	28

2.2.3 – Insérer des données dans la base

La requête « **INSERT** » permet d'ajouter des lignes de données (enregistrements) dans une base de données.

Exemples : INSERT

```
INSERT INTO visiteurs VALUES ('', "j2m0711", "jm_montaud@gmail.com", 18)
```

Cette méthode permet insérer rapidement une ligne en donnant une valeur à tous les champs dans l'ordre :

	id	pseudo	e_mail	age
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	1	jpaul	jp.dupont@gmail.com	42
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	2	lolo	laurent_martin@free.fr	36
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	3	bibou	bibou75@laposte.net	24
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	4	geronimo	jerome228@hotmail.com	28
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	5	j2m0711	jm_montaud@gmail.com	18

```
INSERT INTO visiteurs (age,e_mail,pseudo,id) VALUES (20, "marc_dubois@free.fr" , "md92", '')
```

Cette méthode permet insérer une ligne sans donner obligatoirement une valeur à tous les champs et sans respecter obligatoirement l'ordre des champs :

	id	pseudo	e_mail	age
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	1	jpaul	jp.dupont@gmail.com	42
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	2	lolo	laurent_martin@free.fr	36
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	3	bibou	bibou75@laposte.net	24
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	4	geronimo	jerome228@hotmail.com	28
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	5	j2m0711	jm_montaud@gmail.com	18
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	6	md92	marc_dubois@free.fr	20

2.2.4 – Modifier des données dans la base

La requête « **UPDATE** » permet de modifier un ou plusieurs champs dans un enregistrement.

Exemple : UPDATE

```
UPDATE visiteurs SET e_mail="m_dubois2@gmail.com" WHERE pseudo="md92"
```

Modifie l'adresse mail du visiteur dont le pseudo est « md92 ».

<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	6	md92	m_dubois2@gmail.com	20
--	---	------	---------------------	----

2.2.5 – Supprimer des données dans la base

La requête « **DELETE** » permet de supprimer d'une table un ou plusieurs enregistrements correspondant à une condition particulière. Cette requête est à manipuler avec grande précaution.

Exemple : DELETE

```
DELETE FROM visiteurs WHERE pseudo="bibou"
```

Supprime le visiteur dont le pseudo est « bibou ».

	id	pseudo	e_mail	age
<input type="checkbox"/>	1	jpaul	jp.dupont@gmail.com	42
<input type="checkbox"/>	2	lolo	laurent_martin@free.fr	36
<input type="checkbox"/>	4	geronimo	jerome228@hotmail.com	28
<input type="checkbox"/>	5	j2m0711	jm_montaud@gmail.com	18
<input type="checkbox"/>	6	md92	m_dubois2@gmail.com	20

2.2.6 – Quelques fonctions

UPPER :

La fonction « **UPPER()** » permet de transformer tous les caractères minuscules d'une chaîne de caractère en majuscules.

Exemple : UPPER

```
SELECT UPPER(pseudo) FROM visiteurs
```

UPPER(pseudo)
JPAUL
LOLO
GERONIMO
J2M0711
MD92

LENGTH :

La fonction « **LENGTH()** » permet de calculer la longueur d'une chaîne de caractères.

Exemple : LENGTH

```
SELECT LENGTH(pseudo) AS lg_nom FROM visiteurs
```

Cette requête affecte à la variable « lg_nom » le nombre de caractères de chacun des pseudos.

lg_nom
5
4
8
7
4

**AVG :**

La fonction d'agrégation « **AVG()** » permet de calculer une valeur moyenne sur un ensemble d'enregistrement de type numérique et non nul.

Exemple : AVG

```
SELECT AVG(age) AS age_moyen FROM visiteurs
```

Cette requête affecte à la variable « age_moyen » la moyenne de l'âge des visiteurs.

age_moyen
28.8000

MIN et MAX :

Les fonctions d'agrégations « **MIN()** » et « **MAX()** » permettent de retourner la plus petite valeur et la plus grande d'une colonne sélectionnée. Cette fonction s'applique aussi bien à des données numériques qu'à des données alphanumériques.

Exemple : MIN et MAX

```
SELECT MIN(age) AS age_min FROM visiteurs
```

age_min
18

```
SELECT MAX(age) AS age_max FROM visiteurs
```

age_max
42

COUNT :

La fonction d'agrégation « **COUNT()** » permet de compter le nombre d'enregistrement dans une table.

Exemple : COUNT

```
SELECT COUNT(*) AS nb_visiteurs FROM visiteurs
```

nb_visiteurs
5

3 – MYSQL ET PHP

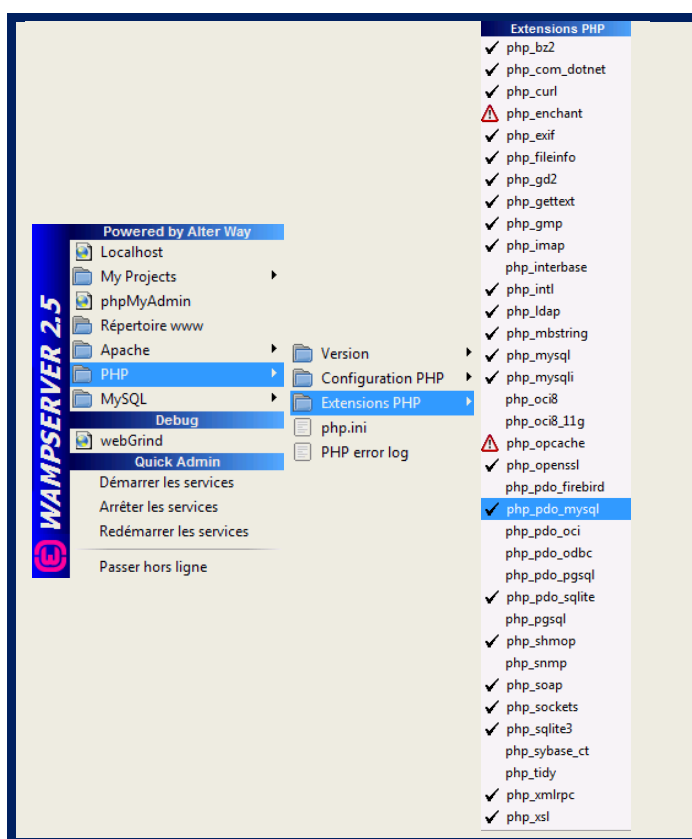
3.1 – CONNEXION A LA BASE DE DONNEES

3.1.1 – Extension PDO

Avant de pouvoir manipuler une base de données à partir de PHP, il est nécessaire d'établir une connexion.

PHP propose plusieurs moyens pour se connecter à une base de données MySQL. Nous choisirons l'**extension PDO** qui est un outil complet qui permet d'accéder à n'importe quel type de base de données.

Cette extension est normalement est activée 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 PHP** » et vérifiez que « **php_pdo_mysql** » est bien coché.





3.1.2 – Connexion à MySQL avec PDO

Pour établir la connexion 4 informations sont nécessaires.

Nom de l'hôte : Il s'agit de l'adresse de l'ordinateur (adresse IP) où MySQL est installé. Le plus souvent, MySQL est installé sur le même ordinateur que PHP : dans ce cas le nom de l'hôte est « **localhost** ».

Nom de la base : Il s'agit du nom de la base de données à laquelle on veut accéder.

Login : il permet associé au mot de passe l'identification. Ce login est donné par l'hébergeur du site WEB. En local le login est « **root** ».

Mot de passe : il permet associé au login l'identification. Ce mot de passe est donné par l'hébergeur du site WEB. En local il n'y a pas de mot de passe.

La syntaxe pour réaliser la connexion à une base de données via l'extension PDO est :

Syntaxe

```
<?php
    $bdd = new PDO('mysql:host=nom_hote;dbname=nom_base','login','mot_passe');
?>
```

Cette ligne de code permet de créer un **objet** « **\$bdd** » représentant la connexion à la base de données. La connexion est créée en indiquant dans l'ordre dans les paramètres :

- le nom d'hôte ;
- le nom de la base de données ;
- le login ;
- le mot de passe.

Exemple : connexion à la base de données « isn » en local

```
<?php
    $bdd = new PDO('mysql:host=localhost;dbname=isn','root','');
?>
```

3.2 – REQUETE DE LECTURE D'UNE BASE DE DONNEES

La syntaxe pour effectuer une requête de lecture en PHP est la suivante :

Syntaxe

```
$reponse = $bdd->query('Tapez votre requête SQL ici');
```

L'objet « **\$reponse** » contient maintenant la réponse de MySQL. Cependant « **\$reponse** » **n'est pas organisée et est inexploitable**.



Pour **récupérer une entrée** il faut exécuter la fonction « **fetch()** » sur la réponse de MySQL.

Syntaxe

```
$donnees = $reponse->fetch();
```

Le **tableau « \$donnees »** contient champ par champ les valeurs de la première entrée. Chaque fois la fonction « **fetch()** » est exécuté on passe à l'entrée suivante. Il faut donc faire une boucle pour parcourir les entrées une à une. La boucle est répétée autant de fois qu'il y a d'entrées dans votre table.

La fonction « **fetch()** » renvoie « **false** » lorsqu'elle est arrivée à la fin des données, c'est-à-dire que toutes les entrées ont été passées en revue.

Exemple : lecture de la table « visiteurs »

```
<?php
$bdd = new PDO('mysql:host=localhost;dbname=isn','root','');
$reponse = $bdd->query('SELECT * FROM visiteurs');
while($donnees = $reponse->fetch()) {
    ?>
    <p>
        <strong>ID</strong> : <?php echo $donnees['id']; ?><br/>
        <strong>Pseudo</strong> : <?php echo $donnees['pseudo'];?><br/>
        <strong>E_mail</strong> : <?php echo $donnees['e_mail'];?><br/>
        <strong>Age</strong> : <?php echo $donnees['age'];?><br/>
    </p>
    <?php
}
$reponse->closeCursor();
?>
```

```

ID : 1
Pseudo : jpaul
E_mail : jp.dupont@gmail.com
Age : 42

ID : 2
Pseudo : lolo
E_mail : laurent_martin@free.fr
Age : 36

ID : 4
Pseudo : geronimo
E_mail : jerome228@hotmail.com
Age : 28

ID : 5
Pseudo : j2m0711
E_mail : jm_montaud@gmail.com
Age : 18

ID : 6
Pseudo : md92
E_mail : m_dubois2@gmail.com
Age : 20
```




La fonction « **closeCursor()** » provoque la « fermeture du curseur d'analyse des résultats ». Cette fonction doit être appelée lorsqu'une requête a été traitée, afin tout problème à la requête suivante.

Exemple : lecture des pseudos de la table « visiteurs »

```
<?php
$dbdd = new PDO('mysql:host=localhost;dbname=isn','root','');
$reponse = $dbdd->query('SELECT pseudo FROM visiteurs');
while($donnees = $reponse->fetch()) {
    ?>
    <p>
    <strong>Pseudo</strong> : <?php echo $donnees['pseudo'];?><br/>
    </p>
    <?php
}
$reponse->closeCursor();
?>
```

Pseudo : jpaul
Pseudo : lolo
Pseudo : geronimo
Pseudo : j2m0711
Pseudo : md92

3.3 – REQUETE DE MODIFICATION D'UNE BASE DE DONNEES

La syntaxe pour effectuer une requête de modification de la base de données en PHP est la suivante :

Syntaxe

```
$dbdd->exec('Tapez votre requête SQL ici');
```

Exemple : Nouvelle entrée

```
<?php
$dbdd = new PDO('mysql:host=localhost;dbname=isn','root','');
$dbdd->exec('INSERT INTO visiteurs VALUES (\'\', "klat", "k_lattanzio@free.fr", 21)');
echo "Nouvelle entrée ajoutée ";
?>
```

Nouvelle entrée ajoutée

	id	pseudo	e_mail	age
<input type="checkbox"/>	1	jpaul	jp.dupont@gmail.com	42
<input type="checkbox"/>	2	lolo	laurent_martin@free.fr	36
<input type="checkbox"/>	4	geronimo	jerome228@hotmail.com	28
<input type="checkbox"/>	5	j2m0711	jm_montaud@gmail.com	18
<input type="checkbox"/>	6	md92	m_dubois2@gmail.com	20
<input type="checkbox"/>	7	klat	k_lattanzio@free.fr	21

Il s'agit de la même syntaxe pour les requêtes « INSERT », « UPDATE » et « DELETE ».