

## Gestion d'une base de données

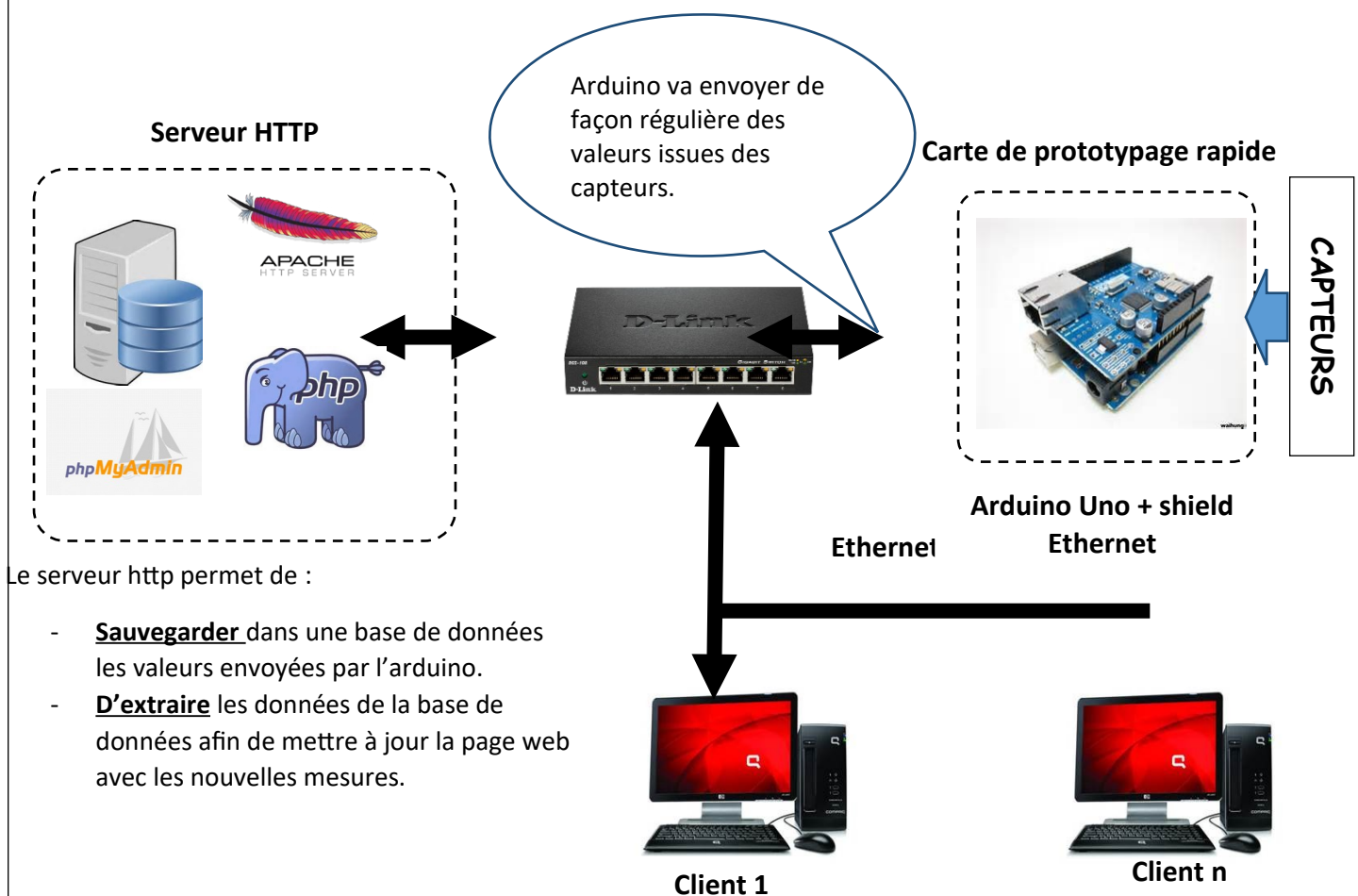
Dans ce TP, nous allons aborder la notion de base de données et de requêtes SQL.

### 1. Introduction :

En informatique, un **système de gestion de base de données** (abr. SGBD) est un logiciel système destiné à stocker et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.

Un SGBD (en anglais DBMS pour database management system) permet d'inscrire, de retrouver, de modifier, de trier, de transformer ou d'imprimer les informations de la base de données. Il permet d'effectuer des comptes rendus des informations enregistrées et comporte des mécanismes pour assurer la cohérence des informations, éviter des pertes d'informations dues à des pannes, assurer la confidentialité et permettre son utilisation par d'autres logiciels<sup>[1]</sup>. Selon le modèle, le SGBD peut comporter une simple interface graphique jusqu'à des langages de programmation sophistiqués<sup>[2]</sup>. [source Wikipédia]

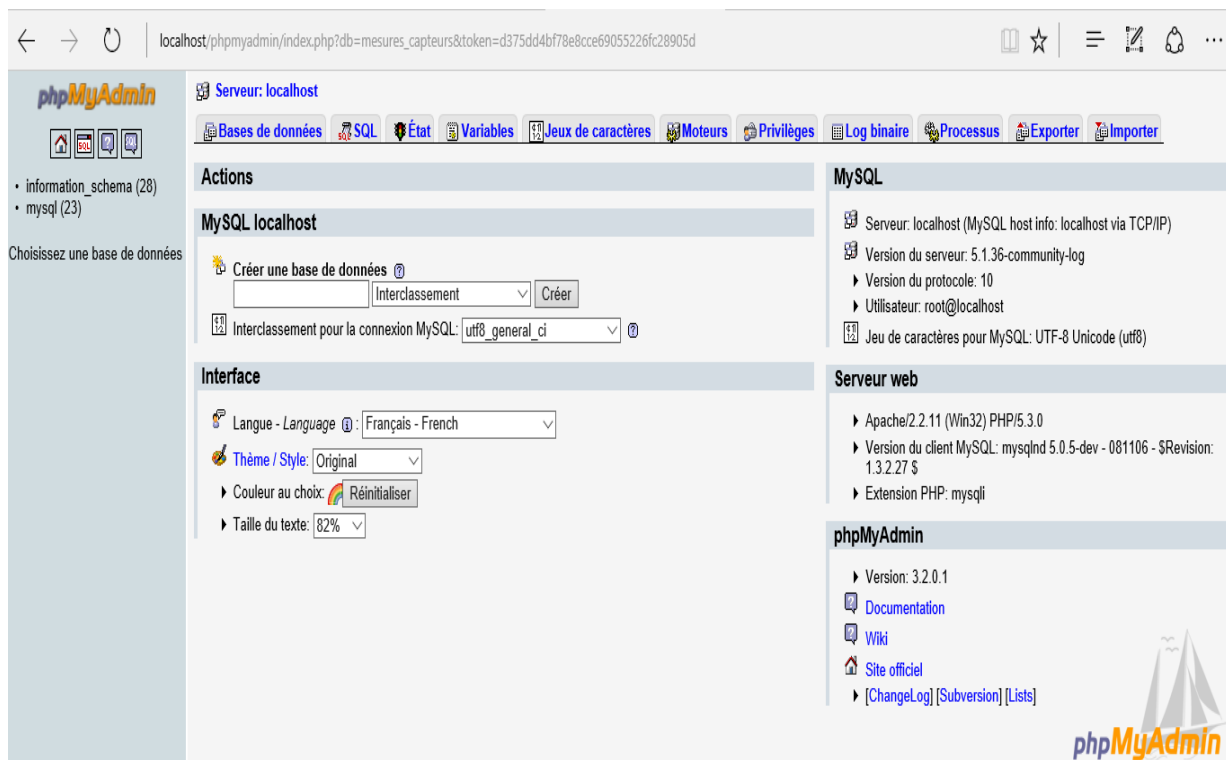
### 2. Structure physique du serveur



### 3. phpMyAdmin sous Lampserver

Afin de pouvoir mémoriser des informations, nous allons créer une base de données à partir du serveur Lamp. Lancer phpMyAdmin sur votre Raspberry. Vous devez obtenir la fenêtre ci-dessous.

**LAMP:**



Cette interface va nous permettre de créer une base de données et de la gérer.

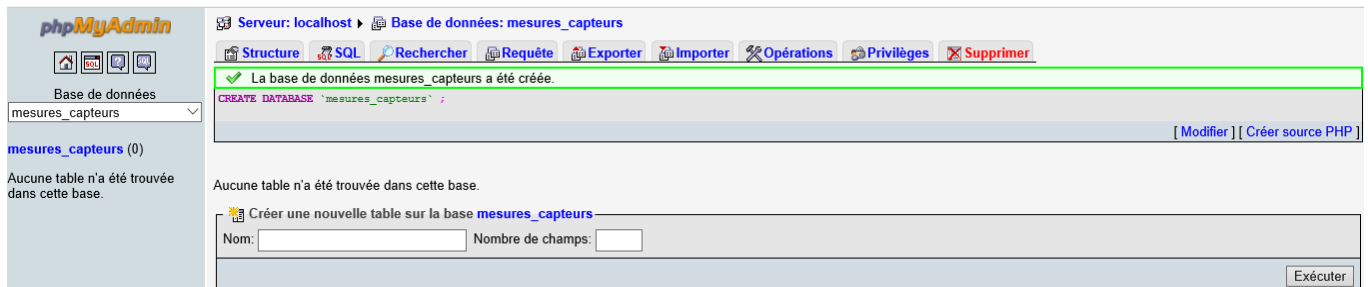
### 4. Création d'une base de données

La création de la base de données va se faire par l'intermédiaire de l'interface phpMyAdmin.



## Gestion d'une base de données (SGBDD)

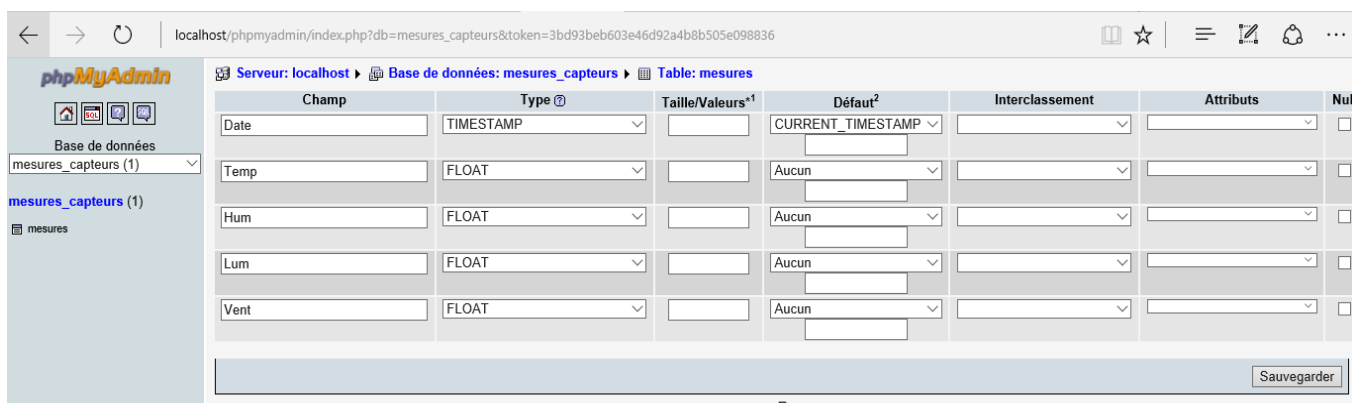
- Créer une base de données qui doit s'appeler mesures\_capteurs et cliquer sur créer. (utiliser « mesures\_prenom » à la place de « mesures\_capteurs » pour pouvoir distinguer les différents dispositifs si besoin)



Afin de pouvoir stocker des données, il faut créer une ou plusieurs tables en fonction de l'application désirée. Dans notre cas, on va utiliser seulement une table.

- Créer une table qui doit s'appeler mesures. Cette table va être utilisée pour stocker les valeurs envoyées par l'Arduino. On travaille toujours sur la serre et l'Arduino permet de recueillir des données sur la température, l'humidité, la luminosité, et la vitesse du vent. On a donc une table constituée de 4 champs. Ne pas oublier de cliquer sur exécuter.

Il faut compléter les champs de cette table.



Et cliquer sur sauvegarder.

### 5. Création d'un script php pour écrire dans une base de données

Afin que l'Arduino puisse écrire de façon cyclique dans la base de données, il faut créer un script php au niveau du serveur. Il faudra le créer dans le répertoire www de notre serveur (/var/www).

Recopier le script ci-dessous et le sauvegarder sous le nom `ecriture_bdd.php`.

```
1 <?php
2 //Déclaration des variables
3 $Temp = $_GET["Temp"];
4 $Hum = $_GET["Hum"];
5 $Lum = $_GET["Lum"];
6 $Vent = $_GET["Vent"];
7
8 try
9 {
10     $bdd = new PDO('mysql:host=localhost;dbname=mesures_capteurs;charset=utf8', 'root', '');
11 }
12 catch(Exception $e)
13 {
14     die('Erreur : '.$e->getMessage());
15 }
16 // On ajoute une entrée dans la table mesures
17 $bdd->exec('INSERT INTO mesures(Temp,Hum,Lum,Vent) VALUES('.$Temp.','.$Hum.','.$Lum.','.$Vent.')');
18 echo 'Donnee ecrite dans la bdd';
19 ?>
```

- Les lignes 3 à 6 nous permettent de récupérer les données qui ont été passées par l'URL.
- Les lignes 8 à 15 nous permettent d'ouvrir la base de données et d'envoyer un message d'erreur si celle-ci n'existe pas.
- La ligne 17 nous permet d'écrire dans la base de données les valeurs.
- La ligne 18 affiche un message pour nous indiquer que l'écriture s'est bien déroulée.

Pour tester notre script d'écriture dans une base de données, il suffit d'écrire dans l'URL :

(Si besoin pour trouver l'adresse IP du Raspberry faites un ifconfig.)

Afin de vérifier si les données ont bien été écrites dans la base de données, il faut retourner sur l'interface de phpMyAdmin et cliquer sur la table mesures. On doit voir apparaître les valeurs que nous avons envoyées par l'URL.

Adresse du serveur  
avec le localhost ou  
bien l'adresse IP du  
Raspberry qui  
héberge le serveur

Commande d'écriture dans la  
base de données envoyée par  
l'URL.

## Gestion d'une base de données (SGBDD)

localhost/phpmyadmin/index.php?db=mesures\_capteurs&token=3bd93beh603e46d02a4b8b505e098836

phpMyAdmin

Base de données  
mesures\_capteurs (1)

mesures\_capteurs (1)

mesures

Serveur: localhost ▶ Base de données: mesures\_capteurs ▶ Table: mesures

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations Vider Supprimer

✓ Affichage des enregistrements 0 - 0 (1 total, Traitement en 0.0004 sec.)

```
SELECT *  
FROM `mesures`  
LIMIT 0, 30
```

☐ Profilage [ Modifier ] [ Expli

Afficher : 30 enregistrement(s) à partir de l'enregistrement n° 0  
en mode horizontal et répéter les en-têtes à chaque groupe de 100

+ Options

	Date	Temp	Hum	Lum	Vent
<input type="checkbox"/>	2016-09-23 14:04:27	25	50	250	5.8589

Tout cocher / Tout décocher Pour la sélection :

Afficher : 30 enregistrement(s) à partir de l'enregistrement n° 0  
en mode horizontal et répéter les en-têtes à chaque groupe de 100

Opérations sur les résultats de la requête

Version imprimable Version imprimable (avec textes complets) Exporter CREATE VIEW

Faites plusieurs essais afin de vérifier l'écriture dans la base de données. Que constatez-vous au niveau de la table ?

### 6. Création d'un script php pour lire dans une base de données

Il est important de savoir lire les données de la base de données afin de pouvoir les utiliser dans une page HTML.

Recopier le script ci-dessous et le sauvegarder sous le nom `lecture_bdd.php`.

```
1 <?php
2 try
3 {
4     $bdd = new PDO('mysql:host=localhost;dbname=mesures_capteurs;charset=utf8', 'root', '');
5 }
6 catch(Exception $e)
7 {
8     die('Erreur : '.$e->getMessage());
9 }
10
11 // Si tout va bien, on peut continuer
12
13 // On récupère tout le contenu de la table
14 $reponse = $bdd->query('SELECT * FROM mesures');
15
16 // On affiche chaque entrée une à une
17 while($donnees=$reponse->fetch())
18 {
19     echo nl2br("affichage reponse."."\n");
20     echo "Date: ", $donnees['Date'], nl2br("\n");
21     echo "Temp=", $donnees['Temp'], nl2br("\n");
22     echo "Hum=", $donnees['Hum'], nl2br("\n");
23     echo "Lum=", $donnees['Lum'], nl2br("\n");
24     echo "Vent=", $donnees['Vent'], nl2br("\n");
25     echo nl2br("-----", "\n");
26 }
27 $reponse->closeCursor();
28 ?>
29
```

- Les lignes 2 à 9 nous permettent d'ouvrir la base de données.
- La ligne 14 récupère les données de la table mesures.
- Les ligne 17 à 24 affichent le contenu de la table.
- La ligne 27 referme la base de données.

Pour tester notre script d'écriture dans une base de données, il suffit d'écrire dans l'URL :

← → ↻ | 127.0.0.1/serveur/lecture\_bdd.php

affichage reponse  
Date: 2016-09-23 14:04:27  
Temp=25  
Hum=50  
Lum=250  
Vent=5.8589  
-----

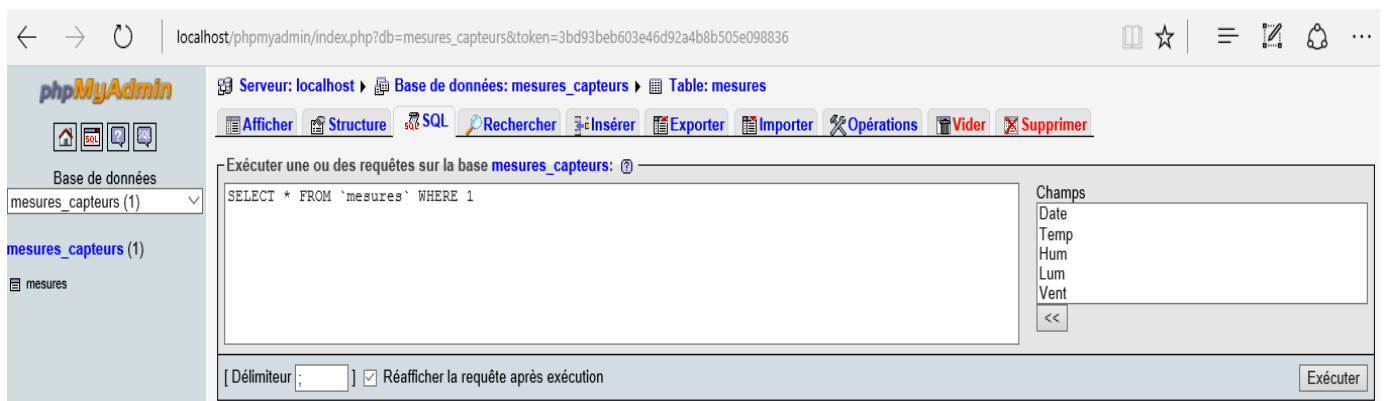
## Gestion d'une base de données (SGBDD)

Vérifier que les données affichées correspondent aux données que vous avez écrites dans labdd. Que constatez-vous ?

### 7. Requête SQL

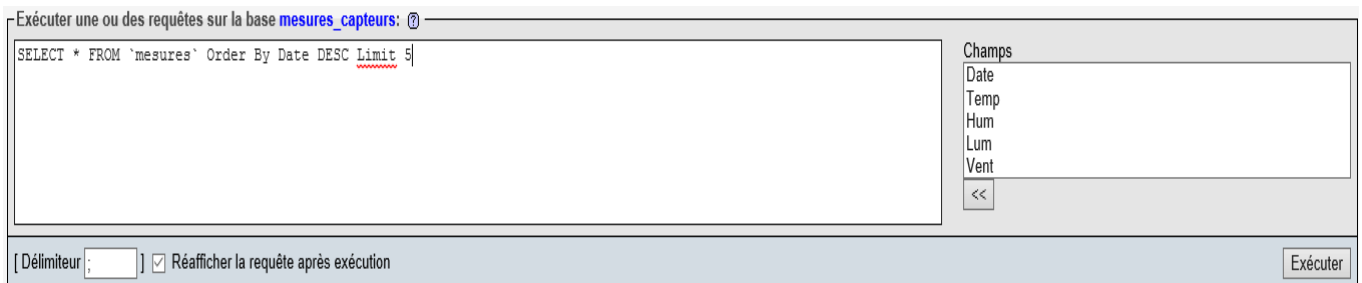
On souhaite afficher les dernières mesures qui ont été mises en mémoire dans la base de données. Pour cela, auparavant, veuillez enregistrer des données dans la base de données (au moins 10 enregistrements).

- Cliquer sur l'onglet SQL.



The screenshot shows the phpMyAdmin interface. The browser address bar indicates the URL: localhost/phpmyadmin/index.php?db=mesures\_captteurs&token=3bd93beb603e46d92a4b8b505e098836. The interface shows the 'mesures\_captteurs' database selected, with the 'mesures' table chosen. The 'SQL' tab is active, displaying the query: `SELECT * FROM `mesures` WHERE 1`. The 'Champs' (Fields) list on the right includes Date, Temp, Hum, Lum, and Vent. The 'Exécuter' (Execute) button is visible at the bottom right.

- Taper la requête ci-dessous et cliquer sur « Exécuter » :
- Que constatez-vous ?



The screenshot shows the phpMyAdmin interface with the 'SQL' tab active. The query entered is: `SELECT * FROM `mesures` Order By Date DESC Limit 5`. The 'Champs' (Fields) list on the right remains the same. The 'Exécuter' (Execute) button is visible at the bottom right.

- Cliquer sur modifier et écrire une requête qui permet d'obtenir la dernière valeur enregistrée.

### Requête :

## Gestion d'une base de données (SGBDD)

- Ecrire un nouveau script php que l'on appellera `lecture_bdd_last.php`, qui permet d'afficher la dernière donnée enregistrée dans la table.
- Modifier le fichier `Affichage_Mesures.php` afin de faire apparaître la date d'enregistrement des données affichées. Il faut renommer ce fichier en **Affichage\_Mesures2.php**.

Vous devez obtenir la présentation ci-dessous. La date doit être écrite en rouge.

Capteurs de la serre			
Mesures des capteurs	Température	26	°C
	Humidité	56	%
	Luminosité	259	Lux
	Vitesse du vent	4.36	m/s
Date des mesures		2016-09-23 15:35:18	

- On souhaite rafraîchir la page html toutes les 10s. Rechercher la balise méta à ajouter et compléter le fichier HTML.
- En utilisant la bibliothèque RGraph, visualiser les dernières mesures de température sous forme de graphique.



## Gestion des utilisateurs

### 1. Présentation :

On souhaite maintenant que l'accès au site permettant d'observer les mesures au niveau de la serre ne soit accessible qu'aux personnes ayant fait une demande d'inscription auprès de l'administrateur.

La figure ci-dessous montre la page associée au fichier index.html.

## Serveur Web Embarqué

identifiant:

Mot de passe:

### S'enregistrer

Nom:

Prenom:

Date de naissance:

Mot de passe:

On considère que  
l'enregistrement se fait  
automatiquement.

### 2. Travail à faire

Modifier l'ensemble des fichiers pour que la demande d'enregistrement soit traitée par votre site.

Pour vous aider :

- Les enregistrements doivent se faire dans une base de données appelée autorisation et les données doivent être stockées dans une table nommée enregistrements.
- Pour enregistrer dans une bdd les informations concernant la personne, il faut dans un premier temps préparer la requête.

## Gestion d'une base de données (SGBDD)

```
$req = $bdd->prepare('INSERT INTO enregistrements(Nom,Prenom,Date_naissance,motpasse) VALUES(:Nom, :Prenom, :Date_naissance, :motpasse)');
$req->execute(array(
    'Nom' => $Nom,
    'Prenom' => $Prenom,
    'Date_naissance' => $Date_naissance,
    'motpasse' => $motpasse
));
```

- Pour lire dans une bdd les informations concernant la personne, il faut dans un premier temps préparer la requête.

```
$req = $bdd->prepare('SELECT Nom, motpasse FROM enregistrements WHERE Nom = :Nom AND motpasse = :motpasse');
$req->execute(array('Nom' => $_GET["login"], 'motpasse' => $_GET["motpasse"]));

// On affiche chaque entrée une à une
while($donnees=$req->fetch())
{
    echo nl2br("affichage reponse"."\\n");
    echo "Enregistrement: ", $donnees['Nom'], nl2br("\\n");
    echo "Enregistrement: ", $donnees['motpasse'], nl2br("\\n");
    echo nl2br("-----"."\\n");
    header('Location: Affichage_Mesures.php');
}
$req->closeCursor();
```

### 3. Pour aller plus loin

On souhaite visualiser la liste des personnes inscrites sur le site. Proposer une solution qui nous permet de la visualiser.