UVS
UNIVERSITE VIRTUELLE DU SENEGAL

La gestion des base de données

Langage PHP avancé

El hadji Mamadou NGUER Enseignant chercheur en Informatique à l'UVS

Séquence 5 : La gestion des bases de données

- **Objectifs spécifiques :** A la suite de ce chapitre, l'étudiant doit être capable de :

1. décrire le SGBD MySQL
2. décrire la gestion d'une base de données MySQL depuis PHP
3. développer un site web dynamique en utilisant PHP et MySQL.
4. mettre en ligne un site web dynamique PHP/MySQL

Séquence 5 : La gestion des bases de données

Plan de la séquence :

Introduction

1. Le SGBD MySQL
2. Le serveur de bases de données
3. La création d'une base de données
4. Extension et objet de traitement de BD
5. La connexion au serveur
6. La déconnexion du serveur
7. L'affichage des données
8. La mise à jour des données
9. La mise en ligne d'un site web dynamique

Conclusion

Introduction

- On a toujours besoin dans une application web dynamique de stocker quelque part des **données** comme les profils utilisateurs, les messages de forums ou les options de navigation des membres, ... en utilisant :

  - les **variables**, mais celles-ci restent en mémoire seulement le temps de la génération de la page.

  - Et les **fichiers**, mais cela devient vite très compliqué dès qu'il y'a beaucoup de données à enregistrer.

- Mais les **bases de données (bd)** constituent le meilleur moyen de faire cela de façon simple et propre.

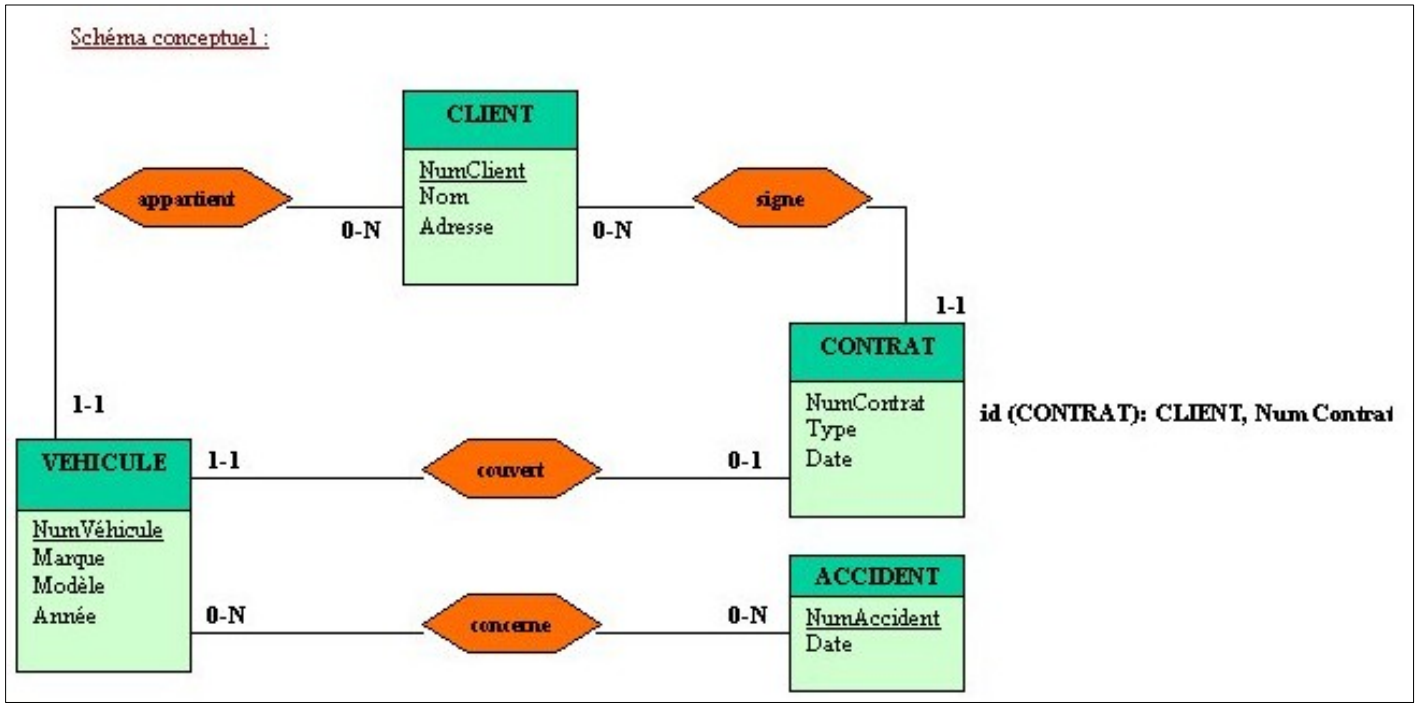
- Une **base de données** est une entité dans laquelle il **est** possible de stocker des **données** de façon structurée et avec le moins de redondance possible. Ces **données** doivent pouvoir être utilisées par des programmes, par des utilisateurs différents.

Introduction

Conception de base de données

- La mise en place d'une base de données nécessite au préalable sa **modélisation au niveau conceptuel** (**MCD**) qui consiste à une représentation graphique et structurée des informations mémorisées.

- Cette étape de modélisation qui ne rentre pas dans le cadre de ce cours, est général faite dans un cours dédié aux Bases de données.



Introduction

Base de données relationnelles

- Les bases de données sont classées en plusieurs types : hiérarchiques, réseaux, relationnelles, objets.

- Aujourd'hui, les bases de données relationnelle (la 3ème évolution) est le type de base qui est le plus répandu.

- Une **base de données relationnelle** est une base de données où l'information est organisée dans des tableaux à deux dimensions appelés des relations ou tables, selon le modèle introduit par Edgar F. Codd en 1970.

- Selon ce modèle **relationnel**, une **base de données** consiste en une ou plusieurs relations (ou tables).

Personnes

nom	*prénom*	*adresse*	*téléphone*
Ndiaye	Ibra	2 Place Malick Sy	33 824 60 25
Dioum	Cheikh	32 allé Limamou Laye	33 997 17 40
Dioum	Dame	21 rue Kocc	33 961 15 20

Introduction

Les SGBD relationnelles

- En informatique, un **système de gestion de base de données** (**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.

- Il existe plusieurs SGBD relationnelles : MySQL, PostgreSQL, SQLite, Oracle, etc.

- Tous ces SGBD accèdent aux données en utilisant le langage **SQL** (Structured Query Language, en français langage de requête structurée) qui est un langage informatique normalisé servant à exploiter des bases de données relationnelles.

Introduction

Les objectifs de cette séquence

- On étudiera dans la suite comment créer une base de données et comment depuis un programme PHP se connecter et manipuler des bases de données de toute sorte.
- Cependant, MySQL qui est le SGBD le plus populaire utilisé avec PHP, sera mis en pratique.
- Il s'agira plus précisément de :
  - Installer un serveur de base de données MySQL
  - Créer une base de données MySQL en utilisant Phpmyadmin
  - Se connecter et se déconnecter à la base de données
  - Afficher des données de la base de données
  - Mettre à jour (ajouter, modifier, supprimer) des données de la bd
  - chercher des données de la bd de manière rapide et interactive.

Le SGBD MySQL

- MySQL est le SGBD le plus populaire utilisé avec PHP.
- Il est de type relationnel et fonctionne dans un serveur
- Sa première version date du 23 mai 1995 et sa dernière version (**5.7.21**) du 15 janvier 2018.
- Il est programmé en C et C++ et est très rapide, fiable et facile à utiliser
- Il utilise SQL (le langage de requête standard) et fonctionne sur plusieurs plates-formes et aussi sur le web.
- Il est développé et pris en charge par Oracle Corporation.
- Il est distribué sous une double licence (GPL et propriétaire) et téléchargeable depuis http://www.mysql.com
- Et son nom vient de la fille du co-fondateur Monty Widenius: My

Le serveur de bases de données

- Un **serveur de bases de données** est un système informatique qui fournit à d'autres ordinateurs des services liés à l'accès et à la récupération de données à partir d'une base de données.

- L'accès au serveur de bases de données peut se faire via un "front end" s'exécutant localement dans la machine d'un utilisateur (par exemple, phpMyAdmin), ou "back end" s'exécutant sur le serveur de base de données lui-même, accessible par shell distant.

- Il faut noter qu'un serveur de bases de données peut contenir plusieurs bases de données.

- Il existe plusieurs outils permettant d'avoir un serveur de bases de données. On peut citer **XAMPP** et **WAMP** qui sont parmi les plus connus.

- L'installation et la configuration du serveur de bases de données seront illustrées par vidéo dans la suite.

Création d'une base de données

- La **création** d'une base de données est faite en **exécutant** dans le serveur **une requête** de création de base de données soit :
  - En ligne de commande
  - Via un interface graphique comme Phpmyadmin
  - Dans un programme d'un langage
- **Exemple** de requête de création de base de données :

```
CREATE TABLE mabase.etudiants (
    Id INT(10) NOT NULL AUTO_INCREMENT ,
    Prenoms VARCHAR(100) NOT NULL ,
    Nom VARCHAR(100) NOT NULL ,
    Login VARCHAR(150) NOT NULL ,
    Mdp VARCHAR(150) NOT NULL ,
    Email VARCHAR(100) NOT NULL,
    Adresse VARCHAR(100) NOT NULL,
    PRIMARY KEY (Id(10))
) ENGINE =InnoDB;
```

Création d'une base de données

- La **création** d'une base de données est faite en **exécutant** dans le serveur **une requête** de création de base de données soit :
  - En ligne de commande
  - Via un interface graphique comme Phpmyadmin
  - Dans un programme d'un langage donnée comme PHP
- Nous allons dans ce cours :
  - créer une base de données manuellement via l'interface graphique offert par Phpmyadmin installé avec Xampp
  - Ensuite accéder et traiter la base de données depuis un programme PHP.

Extension et objet de traitement de BD

L'extension MySQLi et l'objet PDO

- A partir de PHP 5, on traite une base de données MySQL en utilisant:
  - L'extension MySQLi (le "i" signifie improved (amélioration))
  - ou PDO (Objets de données PHP)
- auparavant l'extension MySQL, obsolète depuis 2012, était utilisée.

Installation de MySQLi et de PDO

- Sous Linux et Windows: les extensions MySQLi et PDO sont automatiquement installées dans la plupart des cas, lorsque le paquet mysql de php5 est installé. Pour de plus amples détails, voir :
  - http://php.net/manual/en/mysqli.installation.php
  - http://php.net/manual/en/pdo.installation.php

Extension et objet de traitement de BD

Que choisir entre MySQLi ou PDO?

- Cela dépend de ce que vous voulez car chacun à ses avantages
- PDO fonctionnera sur 12 systèmes de base de données différents, alors que MySQLi fonctionnera uniquement avec les bases de données MySQL.
- Ainsi, si on doit changer la base de données d'un projet pour utiliser une autre base de données, PDO facilite le processus. Il suffit de changer la chaîne de connexion et quelques requêtes. Avec MySQLi, vous devrez réécrire le code entier - requêtes incluses.
- Ils sont tous orientés objet, mais MySQLi offre aussi une API procédurale.
- Les deux prennent en charge les déclarations paramétrées qui protègent contre l'injection de code SQL et sont très importantes pour la sécurité des applications Web.
- On verra dans la suite les trois façons de travailler avec PHP et MySQL: MySQLi (orienté objet), MySQLi (procédural) et PDO

Connexion au serveur

Ouvrir une connexion au serveur MySQL

- Avant d'accéder aux données dans la base de données MySQL, nous devons pouvoir nous connecter au serveur:

Exemple (MySQLi orienté objet)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$bdname = "bdname";

// Créer une connexion
$conn = new mysqli($servername, $username, $password, $bdname);

// Vérifier la connexion
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

Connexion au serveur

Ouvrir une connexion au serveur MySQL

Exemple (MySQLi procédural)

```php
<?php
  $servername = "localhost";
  $username = "username";
  $password = "password";
  $bdname = "bdname";

  // Créer une connexion
  $conn = mysqli_connect($servername, $username, $password,$bdname);

  // Vérifier la connexion
  if (!$conn) {
      die("Connection failed: " . mysqli_connect_error());
  }
  echo "Connected successfully";
  ?>
```

Connexion au serveur

Ouvrir une connexion au serveur MySQL

Exemple (PDO)

```php
<?php
$server = "localhost";
$user = "username";
$mdp = "password";

try {
    $conn = new PDO("mysql:host=$server;dbname=myDB", $user, $mdp);
    // Définir le mode d'erreur PDO comme l'exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Connected successfully";
 }catch(PDOException $e){
    echo "Connection failed: " . $e->getMessage();
 }
?>
```

Remarque: Un grand avantage de PDO est qu'il possède une classe d'exception pour gérer tout problème pouvant survenir dans une requête de base de données.

La déconnexion du serveur

Fermer la connexion

Fermer la connexion

- La connexion sera automatiquement fermée lorsque l'exécution du script termine. Pour fermer la connexion avant, on procède comme suit:

```
$conn->close();          (MySQLi orienté objet)

mysqli_close($conn);     (MySQLi Procédural)

$conn = null;            (PDO)
```

L’affichage des données

- L’affichage des données d’une base de données se fait en suivant les étapes suivants :
  - Formuler une requête de sélection des données,
  - Exécuter la requête pour obtenir les données,
  - Parcourir les données obtenues pour les extraire et les afficher.
- **Exemple (MySQLi Orienté Objet)**

```
$sql = "SELECT id, prenoms, nom FROM Persons";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // Afficher les données de chaque ligne
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Nom: " . $row["prenoms"]. " " . $row["nom"]. "<br>";
    }
} else {
    echo "0 results";
}
```

L’affichage des données

- **Exemple (MySQLi Procédural)**

```php
$sql = "SELECT id, prenoms, nom FROM Persons";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {

    //  Afficher les données de chaque ligne
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Nom: " . $row["prenoms"]. " " . $row["nom"]. "<br>";
    }

} else {
    echo "0 results";
}
```

L’affichage des données

- **Exemple (MySQLi PDO)**

```
$sql = "SELECT id, prenoms, nom FROM Persons";
$result = $conn->query($sql);

if ($result->rowCount > 0) {
    echo "<table><tr><th>ID</th><th>Name</th></tr>";
    // Afficher les données de chaque ligne
    while($row = $result->fetch()) {
        echo "<tr><td>".$row["id"]."</td><td>".$row["prenoms"]."
".$row["nom"]."</td></tr>";
    }
    echo "</table>";

} else {
    echo "0 results";
}
```

Mise à jour des données

Obtenir l'ID du dernier enregistrement inséré

- Si nous effectuons une insertion ou une mise à jour sur une table avec un champ auto-incrémenté, on peut obtenir immédiatement l'identifiant du dernier enregistrement ajouté ou mis à jour.

- **Exemple (MySQLi Orienté Objet)**

```
$sql = "INSERT INTO MyGuests (prenoms, nom, email)
VALUES ('Saliou', 'Deme', 'saliou@exemple.com')";

if ($conn->query($sql) === TRUE) {
    $last_id = $conn->insert_id;
    echo "Enregistrement créé avec succès. La dernière ID insérée est: " . $last_id;
} else {
    echo "Erreur: " . $sql . "<br>" . $conn->error;
}
```

Mise à jour des données

Obtenir l'ID du dernier enregistrement inséré

- **Exemple (MySQLi Procédural)**

```
if (mysqli_query($conn, $sql)) {
    $last_id = mysqli_insert_id($conn);
    echo "Enregistrement créé avec succès. La dernière ID insérée est: " . $last_id;
} else {
    echo "Erreur: " . $sql . "<br>" . mysqli_error($conn);
}
```

- **Exemple (MySQLi PDO)**

```
$sql = "INSERT INTO MyGuests (prenoms, nom, email)
VALUES ('Saliou', 'Deme', 'saliou@exemple.com')";
    // On utilise exec () car aucun résultat n'est retourné
    $conn->exec($sql);
    $last_id = $conn->lastInsertId();
    echo " Enregistrement créé avec succès. La dernière ID insérée est : " . $last_id;
```

Mise à jour des données

Insertion multiple

- On utilise la fonction **mysqli_multi_query ()** pour des insertions multiples.
- **Exemple (MySQLi Orienté Objet)**

```
$sql = "INSERT INTO MyGuests (prenoms, nom, email)
VALUES ('Saliou', 'Deme', 'saliou@exemple.com')";
$sql .= "INSERT INTO MyGuests (prenoms, nom, email)
VALUES ('Cheikh', 'Sarr', 'cheikh@exemple.com')";

if ($conn->multi_query($sql) === TRUE) {
    echo "Enregistrement créé avec succès";
} else {
    echo "Erreur: " . $sql . "<br>" . $conn->error;
}
```

- **Exemple (MySQLi Procédural)**

```
if (mysqli_multi_query($conn, $sql)) {
    echo "Enregistrement créé avec succès";
} else {
    echo "Erreur: " . $sql . "<br>" . mysqli_error($conn);
}
```

Mise à jour des données

Insertion multiple

- **Exemple (MySQLi PDO)**

```
try {
    //…
    $conn->beginTransaction(); // Commencer la transaction
    // Requêtes SQL
    $conn->exec("INSERT INTO MyGuests (prenoms, nom, email)
    VALUES ('Saliou', 'Deme', 'saliou@exemple.com')");
    $conn->exec("INSERT INTO MyGuests (prenoms, nom, email)
    VALUES ('Cheikh', 'Sarr', 'cheikh@exemple.com')");
    //Lancer la transaction
    $conn->commit();
    echo "Enregistrements créés avec succès";
    }catch(PDOException $e){
    // Retourner la transaction si echec
    $conn->rollback();
    echo "Erreur: " . $e->getMessage();
    }
```

Mise à jour des données

Les requêtes paramétrables

- Une requête paramétrable est une fonction permettant d'exécuter les mêmes requêtes SQL à plusieurs reprises avec une efficacité élevée.
- Elles sont très utiles contre les injections SQL.
- Et elles fonctionnent comme suit:

1. **La préparation:** un modèle de requête SQL est créé et envoyé à la base de données. Certaines valeurs appelées paramètres (étiquetés "?") ne sont pas spécifiées. **Exemple:** INSERT INTO Personne VALUES (?,?,?)

2. **Traitement:** la base de données analyse, compile et effectue l'optimisation des requêtes sur le modèle de requête SQL et stocke le résultat sans l'exécuter

3. **L'exécution:** plus tard, l'application affecte des valeurs aux paramètres et la base de données exécute la requête. L'application peut exécuter la déclaration autant de fois qu'on le souhaite avec différentes valeurs

Mise à jour des données

Les requêtes paramétrables

- Par rapport à l'exécution directe des requêtes SQL, les requêtes paramétrables ont trois avantages principaux:

  - Elles réduisent le temps d'analyse car la préparation de la requête n'est effectuée qu'une seule fois (même si la requête est exécutée plusieurs fois)

  - Les paramètres liés réduisent la bande passante au serveur car il vous faut envoyer uniquement les paramètres à chaque fois, et non la requête entière

  - Elles sont très utiles contre les injections SQL, car les valeurs des paramètres, qui sont transmises ultérieurement à l'aide d'un protocole différent, n’auraient pas besoin d’être échappées. Si le modèle de déclaration d'origine n'est pas dérivé de l'entrée externe, l'injection SQL ne peut pas se produire.

Mise à jour des données

Les requêtes paramétrables

- **Exemple (MySQLi)**

```
// Préparation et liaison
$stmt = $conn->prepare("INSERT INTO Personnes (prenoms, email) VALUES (?, ?)");
$stmt->bind_param("sss", $prenoms, $email);

// Définir les paramètres et exécuter
$prenoms= "Jatu";
$email = "jatu@exemple.com";
$stmt->execute();

$prenoms= "Mareem";
$email = "mareem@exemple.com";
$stmt->execute();

$prenoms= "Njole";
$email = "njole@exemple.com";
$stmt->execute();
```

Mise à jour des données

Les requêtes paramétrables
Explications (MySQLi)

- "INSERT INTO Personnes (prenoms, email) VALUES (?, ?)"

Dans cette requête SQL, on insert un point d'interrogation (?) qui sera substituer en une valeur entière, chaîne, double ou blob.

- $stmt->bind_param("sss", $prenoms, $nom, $email);

Cette fonction lie les paramètres à la requête SQL et indique à la base de données quels sont les paramètres. L'argument "sss" répertorie les types de données que les paramètres sont. Le caractère s indique à mysql que le paramètre est une chaîne.

- L'argument peut être l'un des quatre types: i (integer), d (double), s (string) et b (BLOB).
- On doit avoir l'un d'eux pour chaque paramètre. Ainsi en indiquant à MySQL quel type de données attendre, on minimise le risque d'injections SQL.

Mise à jour des données

Les requêtes paramétrables

- **Exemple (PDO)**

```
// Préparez la requête sql et lier les paramètres
$stmt = $conn->prepare("INSERT INTO Personnes (prenoms, nom) VALUES (:prenoms, :nom)");
$stmt->bindParam(':prenoms', $prenoms);
$stmt->bindParam(':nom', $nom);

$prenoms= "Michel";
$nom= "Ndong";
$stmt->execute();      // Insérer une ligne

$prenoms= "Mary";
$nom= "Faye";
$stmt->execute();      // Insérer une autre ligne

$prenoms= "Juliette";
$nom= "Ndiaye";
$stmt->execute();      // Insérer une autre ligne
```

Mise en ligne d'un site web dynamique

Motivations :

- Durant ce cours nous avons appris à créer un site web dynamique en local
- Cependant un site web créé en local n'est pas accessible sur le web;
- Pour qu'il soit acessible sur le Web, il faudra :

  - trouver un hébergeur pour votre site web
  - disposer d'un nom de domaine
  - et procéder à la mise en ligne qui consiste à :
    - importer la base de données locale dans le serveur distant.
    - transférer le site web vers le serveur après avoir modifier les paramètres liés au serveur de base de données,

Mise en ligne d'un site web dynamique

Le nom de domaine

- Un nom de domaine est un « masque » sur une adresse IP. Son but est de retenir et communiquer facilement l'adresse d'un ensemble de serveurs (site web, courrier électronique, FTP).

Par exemple, **gouv.sn** est plus simple à mémoriser que 196.1.94.186.

Syntaxe

- Un nom de domaine est constitué de deux parties : ***nomdomaine.extension***
  - où ***nomdomaine*** est le nom du domaine proprement dit
  - ***extension*** est l'extension (aussi appelée « TLD », de l'anglais *top-level domain*). Il existe grosso modo une extension par pays.

  - En général, un site web voit son adresse précédée par www, comme par exemple www.gouv.sn. www ne fait pas parti du domaine, c'est juste un sous domaine.

Mise en ligne d’un site web dynamique

Le nom de domaine

Pour réserver un nom de domaine, on a deux solutions :

- Passer par un **registrar** spécialisé. C'est un organisme qui sert d'intermédiaire entre l'ICANN (l'organisation qui gère l'ensemble des noms de domaine au niveau international) et vous. Kheweul.com SA et Sonatel Multimedia sont de célèbres registrars sénégalais.
- Vous pouvez aussi commander le nom de domaine en même temps que l'hébergement (c'est ce qui est conseillé).
- Cependant on peut avoir un **nom de domaine gratuit** si on décide d’héberger gratuitement son site web.

Mise en ligne d'un site web dynamique

L'hébergement d'un site web

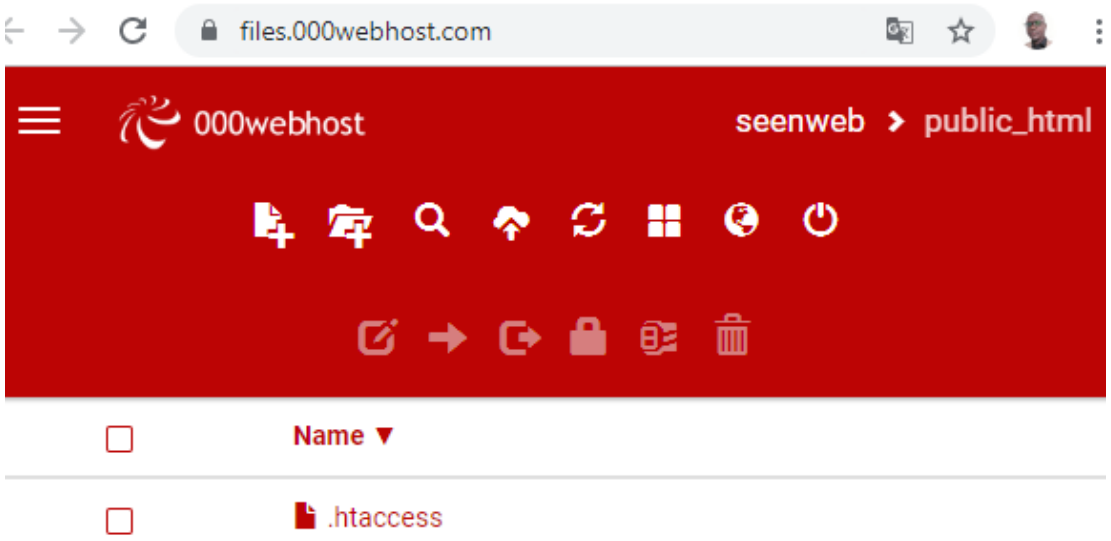
- Sur Internet, tous les sites web sont stockés dans des serveurs (figure ci-dessus).
- La société mettant à votre disposition un serveur web (ordinateur connecté en permanence à internet) est appelée hébergeur.
- On distingue deux principales catégories d'hébergeurs :
  - les hébergeurs gratuits (000webhost.com par exemple).
  - les hébergeurs professionnels (kheweul.org, arvixe.com par exemple).
- L'hébergeur vous fournit le **nom du serveur**, les **paramètres de connexion** nécessaire pour transférer votre site ainsi qu'un **cPanel** qui permet plusieurs paramétrages dont celui du serveur de base de données.
- Il peut aussi fournir un ensemble d'outils : un **gestionnaire de fichiers** en ligne qui permet d'uploader votre site dans le serveur, un **construction de site web**, etc.

Mise en ligne d'un site web dynamique

Le transfert d'un site web

- Le transfert d'un site web vers le serveur peut se faire à l'aide :
  - d'un **client FTP** comme Filezilla. Mais il faudra disposer des paramètres de connexion au serveur fournis par l'hébergeur.

- ou d'un **gestionnaire de fichiers en ligne**



Mise en ligne d'un site web dynamique

Panneau de configuration

- Plusieurs hébergeurs utilisent cPanel qui est un panneau de configuration basé sur Linux conçu pour les hébergeurs web.
- Il permet l'accès et le paramètrage de plusieurs outils dont le **serveur de base de données.**