BASES DE DONNEES MySQL

Savoirs	Champs ciblé
S 4.7 Langages de programmation	Web dynamique : PHP, Javascript (2) SQL

1. INTRODUCTION

1.1. Qu'est-ce qu'une base de données

Définition : Un ensemble d'informations logiquement reliées entre elles. Plus précisément, nous appellerons base de données un ensemble structuré et organisé permettant le stockage de grandes quantités d'informations afin d'en faciliter l'exploitation (ajout, mise-à-jour, recherche de données).

Le stockage de ces informations en base de données permet l'évolution indépendante des programmes et des données.

La mise en œuvre de ce stockage se fait au moyen d'un outil logiciel spécialisé : Le Système de Gestion des Bases de Données (ou SGBD1[1])

1.2 Pourquoi PHP et le SGBD MySQL ?

Lorsqu'un script PHP produit des informations, il faut les stocker quelque part sinon, elles seront irrémédiablement perdues.

Pour stocker des données, il existe deux solutions:
⇒ Les enregistrer dans un fichier texte sur le serveur (quelque part dans l'arborescence de votre hébergement),
⇒ Les enregistrer dans une base de données relationnelle.

La sauvegarde dans un fichier présente l'avantage de pouvoir accéder rapidement aux données qu'il contient sans avoir à utiliser de logiciels tiers.
Cependant, les recherches, modifications ou suppressions de données sur une partie de l'information stockée n'est pas facile. Il est encore plus difficile de croiser des informations provenant de plusieurs fichiers.

Les bases de données ont été conçues pour ça.

Les informations y sont stockées sous forme de tables contenant des enregistrements.

[1] Le SGBD est un ensemble de programmes qui assure la gestion et l'accès à une base de données. Un SGBD est multiutilisateurs, c'est à dire que plusieurs utilisateurs peuvent accéder au SGBR et ainsi manipuler les données de la base en même temps.

Les opérations sur les tables et les enregistrements sont réalisées à l'aide de **requêtes** comme **SQL**, **S**tructured **Q**uery **L**anguage, interprété par un logiciel spécialisé **SGBD.**
Le **SGDB** constitue le serveur de base de données.

Plusieurs **SGBD** sont disponibles sur le marché : Oracle, Microsoft SQL Server, MySQL, SQLite…

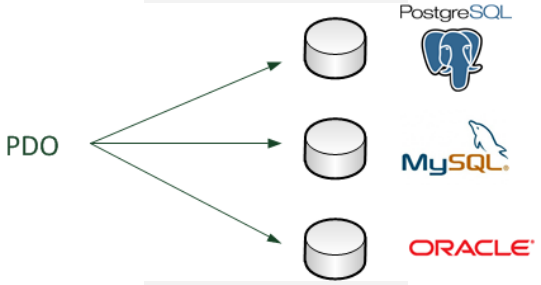
Nous traiterons dans la suite du serveur de base de données **MySQL**, qui est le plus souvent associé à **PHP**.

2. COMMENT SE CONNECTE-T-ON A LA BASE DE DONNEES EN PHP ?

PHP propose **plusieurs moyens** de se connecter à une base de données MySQL.

- L'extension ***mysql_*** : ce sont des fonctions qui permettent d'accéder à une base de données MySQL et donc de communiquer avec MySQL. Leur nom commence toujours par ***mysql_.*** Toutefois, ces fonctions sont vieilles et on recommande de ne plus les utiliser aujourd'hui.
- L'extension ***mysqli_*** : ce sont des fonctions améliorées d'accès à MySQL. Elles proposent plus de fonctionnalités et sont plus à jour.
- L'extension ***PDO*** : c'est un outil complet qui permet d'accéder à n'importe quel type de base de données. On peut donc l'utiliser pour se connecter aussi bien à MySQL que PostgreSQL ou Oracle.

Nous allons ici utiliser ***PDO*** car c'est cette méthode d'accès aux bases de données qui va devenir la plus utilisée dans les prochaines versions de PHP.



Description séquentielle entre PHP et MySQL lors d'un échange de données :

Etape 1 : Création d'une liaison avec le serveur

Etape 2 : Sélection d'une base

Etape 3 : Préparation de la requête

Etape 4 : Envoi de la requête à la base et réception d'un éventuel résultat

Etape 5 : Fermeture de la liaison

2.1 Création d'une liaison avec le serveur

```
$serveur = "localhost";
$login = "root";
$pass = "";
$connexion = new PDO("mysql:host=$serveur;dbname=BDD" , $login, $pass);
$connexion -> setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

2.2 MySQL est donc un langage typé, ci-dessous les principaux types :

Type de données MySQL	Définition	Syntaxe
VARCHAR	Chaîne de caractères de taille variable	VARCHAR(length)
TINYTEXT	Texte contenant au maximum 255 caractères	
TEXT	Texte contenant au maximum 65 535 caractères	
MEDIUMTEXT	Texte contenant au maximum 16 777 215 caractères	
LONGTEXT	Texte contenant au maximum 4 294 967 295 caractères	
TIME	Temps au format '00:00:00'	TIME
DATE	Date au format ISO 'AAAA-MM-JJ'	DATE
TINYINT	Entier compris entre –128 et 127	
SMALLINT	Entier compris entre –32 768 et 32 767	SMALLINT[(length)] [UNSIGNED]
INT	Entier compris entre – 2 147 483 648 et 2 147 483 647	INTEGER[(length)] [UNSIGNED]
BIGINT	Entier compris entre ≈ – 9 10^{18} et 9 10^{18}	BIGINT[(length)] [UNSIGNED]
FLOAT	Nombre à virgule	FLOAT[(length,decimals)] [UNSIGNED]

TEXT et BLOB
Les types TINYBLOB, BLOB, MEDIUMBLOB et LONGBLOB sont identique à TEXT.
Mais BLOB est sensible à la casse, TEXT est insensible à la casse.

2.4 Insertion de données, Modification et suppression de données

Insertion standard :

```
$insertion= « INSERT INTO nom_de_table (champ_1, champ_2,…)VALUES('valeurs1', 'valeurs2',…) » ;
$connexion->exec($insertion);
```

Si un champ est défini comme auto incrémenté, il suffit de lui donner la valeur NULL pour que MySQL se débrouille tout seul.

Insertion via une sous requête :

```
$insertion= « INSERT INTO nom_de_table (champ_1, ….)
SELECT …. » ;
$connexion->exec($insertion);
```

Insertion multiple en une passe :

```
$insertion= « INSERT INTO table (champ_1, champ_2) VALUES (‘val1’, ‘val2’),
VALUES (‘val21’, ‘val22’), VALUES (‘val31’, ‘val32’) » ;
$connexion->exec($insertion);
```

```
$modif= « UPDATE nom_de_table SET champ_1 = valeur1 WHERE champ_2 = valeur2 » ;
$connexion->exec($modif);
```

Supprimer des données (DELETE)

```
$suppr=  « DELETE FROM nom_de_table WHERE champ_1=valeur1”;
$connexion->exec($suppr);
```

2.5 Récupération de données, filtrage des données

Récupérer des données (SELECT)

Toutes les données de la table :

```
$requete= $connexion->prepare(« SELECT *FROM nom_de_table ») ;
$requete->execute();
```

Affichage sélectif d’un champ :

```
$requete = $connexion->prepare(« SELECT champ FROM nom_table »);
$requete->execute();
```

Trier avec « ORDER BY »

```
$requete= $connexion->prepare(« SELECT *FROM nom_de_table ORDER BY champ [DESC|ASC]”);
$requete->execute();
```

Retenir la valeur maximale ou minimale

```
$requete = $connexion->prepare("SELECT MIN(champ) FROM nom_table");
$requete = $connexion->prepare("SELECT MAX(champ) FROM nom_table");
$requete->execute();
```

Filtrer avec la clause WHERE

Syntaxe générale :
[SELECT | UPDATE | DELETE]
WHERE condition

Autre exemple ⇒ WHERE champs IN (‘valeur1’, ‘valeur2’, ‘valeur3’)
Autre exemple ⇒ WHERE champs NOT IN (‘valeur1’, ‘valeur2’, ‘valeur3’)
Autre exemple ⇒ WHERE champs BETWEEN ‘limite1’ AND ‘limite2’
Autre exemple ⇒ WHERE champs LIKE ‘expression’

3. FERMETURE DE LA LIAISON

```
$connexion = null;
```