Plan

Interfaçage avec une base de données

Introduction

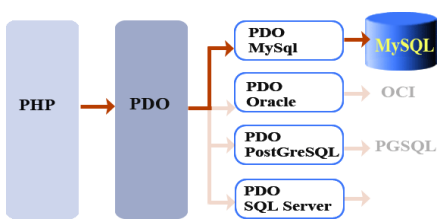
- PHP propose plusieurs moyens de se connecter à une base de données MySQL:
  - L'extension mysql_ : ce sont des fonctions qui permettent de communiquer avec une base de données MySQL.
    - Ces fonctions sont vieilles et on recommande de ne plus les utiliser aujourd'hui.
  - L'extension mysqli_ : ce sont des fonctions améliorées d'accès à MySQL. Elles proposent plus de fonctionnalités et sont plus à jour.
  - L'extension PDO : une interface pour accéder à n'importe quel type de base de données depuis PHP.

Interfaçage avec une base de données

PHP Data Objects (PDO)

Pourquoi PDO ?

PDO permet l'accès à n'importe quel SGBD



Interfaçage avec une base de données

Connexion avec PDO

- Il faut spécifier les paramètres suivants:
  - le nom d'hôte (localhost) ;
  - la base de données (test) ;
  - le login (root) ;
  - le mot de passe.

Data Source Name. Change en fonction du type de SGBD auquel on se connecte.

```
$bdd = new PDO('mysql:dbname=isima;host=localhost;port=3307;charset=utf8', 'root', '');
```

Adresse du serveur

Login et mdp

Interfaçage avec une base de données

Connexion avec PDO

```
try
{
$bdd = new PDO('mysql:dbname=isima;host=localhost;port=3307;charset=utf8', 'root', '');
}
catch (Exception $e)
{
    die('Erreur : ' . $e->getMessage());
}
```


Interfaçage avec une base de données

Requêtes avec PDO

- Deux méthodes:
  - exec($request) : pour modifier la base de données
  - query($request) : pour extraire des données de la base

```
$login="Elkosantini";
$res = $bdd->exec('UPDATE etudiants SET login="'.$login.'" WHERE CIN="01234566"');
```

Lecture cryptée du mot de passe

```
$pwd="test";
$res = $bdd->exec('UPDATE etudiants SET mdp="'.md5($pwd).'" WHERE CIN="01234566"');
echo "nombre de lignes modifiées = $res";
```

CIN	Nom	login	mdp
01234566	Sabeur	Elkosantini	098f6bcd4621d373cade4e832627b4f6

Interfaçage avec une base de données

Requêtes SELECT avec PDO

- La méthode query($request) renvoie un objet de type *PDOStatement*.
- Les méthodes les plus importantes sont:
  - ✓ fetch: Récupère la ligne suivante d'un jeu de résultats PDO
  - ✓ FetchAll: Retourne un tableau contenant toutes les lignes du jeu d'enregistrements
  - ✓ rowCount : Retourne le nombre de lignes affectées par le dernier appel à la fonction execute() ou query()
  - ✓ nextRowset: Avance à la prochaine ligne de résultats d'un gestionnaire de lignes de résultats multiples

Interfaçage avec une base de données

Requêtes SELECT avec PDO

```
$reponse = $bdd->query('SELECT * FROM etudiants');
while ($donnees = $reponse->fetch())
{ ?>
Le nom de l'étudiant est <?php echo $donnees['Nom']; } ?> <BR>
```

```
$reponse = $bdd->query('SELECT * FROM etudiants');
while ($donnees = $reponse->fetch())
{
echo "Le nom de l'étudiant est:".$donnees['Nom']."<BR>";
}
```

Requêtes SELECT avec PDO

- Utilisation de fetchAll: permet de mettre toutes les lignes dans un tableau

```php
$reponse = $bdd->query('SELECT * FROM etudiants');
$tableau = $reponse->fetchAll();
foreach ($tableau as $row) {
    echo $row['Nom']."<br />\n";
}
```


Injection SQL

- Injection SQL est une méthode d'attaque qui consiste à modifier une requête SQL en injectant des morceaux de code non filtrés, généralement par le biais d'un formulaire

```php
$login="' or '1'='1";
$mdp="' or '1'='1";
```

```php
$sql="SELECT * FROM etudiants where login='".$login."' AND mdp ='".$mdp."'";
$reponse = $bdd->query($sql);
if ($reponse->rowCount()>0){
    $donnees= $reponse->fetch();
    echo "Bienvenue ".$donnees['Nom']."<br />";
}
```

SELECT * FROM etudiants where login='' or '1'='1' AND mdp ='' or '1'='1'

Injection SQL

- Injection SQL est une méthode d'attaque qui consiste à modifier une requête SQL en injectant des morceaux de code non filtrés, généralement par le biais d'un formulaire

```php
$login="' or '1'='1' ;//";
$mdp="' or '1'='1";
$sql="SELECT * FROM etudiants where login='".$login."'
        AND mdp ='".md5($mdp)."'";
```

```
SELECT * FROM etudiants where login='' or '1'='1' ;
                    //' AND mdp ='59725b2f19656a33b3eed406531fb474'
```


Requêtes SELECT avec PDO

- Utilisation des requêtes préparées

```php
$reponse = $bdd->prepare('SELECT * FROM etudiants where CIN=? AND login =?');
$CIN='01234566';
$login="Elkosantini";
$reponse->execute([$CIN, $login]);

$tableau = $reponse->fetchAll();
foreach ($tableau as $row) {
    echo $row['Nom']."<br />\n";
}
```

Requêtes SELECT avec PDO

- Fin de traitement des résultats

```php
$reponse->closeCursor();
```

Eviter d'avoir des problèmes à la requête suivante