

PHP et les bases de données

C. BENSARI

- Un langage de programmation backend à la responsabilité de faire la communication avec les bases de données
- Chaque langage de programmation possède un API (**Application Programming Interface**) qui lui permet de faire des connexions à une base de données, idéalement, quelque soit le type du serveur SGBD
 - JAVA : JDBC, Hibernate (ORM), ..
 - PHP : mysqli (mysql), OCI (oracle), PDO, ..

- PHP propose deux styles de syntaxe avec lesquels on peut communiquer avec une base données.
 - Style procédural : utilisation de fonctions avec des appels en séquences d'instructions
 - Style orienté objet : utilisation des notions d'objet et de classes

- “**mysqli**” est le nouveau nom de l'interface php **mysql** (historiquement mysql)
- mysqli est une API spécifique qui transcrit dans PHP les commandes MySQL
- mysqli est l'une des nombreuses interfaces que PHP intègre pour accéder aux bases de données de divers constructeurs (Postgresql, Oracle, MongoDB, etc.)

MYSQLI (procédural ou impératif)

- Dans un style impératif, il faut suivre les étapes suivantes :
 - Récupération de la ressource (BDD) sur laquelle on doit se connecter
 - Connexion à la ressource
 - Exécution de la requête
 - Récupération des résultats (**Uniquement dans le cas du Select**)

```
$db = mysqli_init();  
mysqli_real_connect($db, 'localhost', 'root', '', 'testemployes');  
$rs = mysqli_query($db, 'SELECT * FROM employers');  
$data = mysqli_fetch_all($rs, MYSQLI_ASSOC);
```

Libération des resultset (procédural)

- Une fois une recherche effectuée et les résultats ne seront plus utilisés, il est préférable de libérer les resultset :

```
mysqli_free_result($rs);
```

- Ceci va permettre la libération de l'espace mémoire utilisé par la variable \$rs
- Ceci est nécessaire si les requêtes effectuées envoient un volume important de données
- Par défaut, php supprime automatiquement les results à la fin du script

Fermeture d'une session MYSQLI

- Php ferme automatiquement chaque session de connexion à une base données
- Il est préférable de le faire manuellement à la fin du script php :

```
mysqli_close($db);
```

MYSQLI (Orienté Objet)

- On rend disponible une ressource “base de données” en instanciant la classe mysqli :

```
$db = new mysqli('nom_machine', 'utilisateur', 'mot_de_passe', 'nom_base_de_donnees');
```

- Une requêtes SQL peut être exécutée avec la méthode query

```
$res = $db->query('INSERT INTO personne (age) VALUES (26)');
```

- Pour les requêtes de type **SELECT**, la méthode **query** envoie un tableau comprenant tous les résultats trouvés

- Pour libérer les resultset de la mémoire :

```
$rs->free();
```

- Pour fermer la session de connexion à la base de données :

```
$db->close();
```

Les requêtes préparées (prepared statements)

- Une requête MYSQLI peut être créée en plusieurs étapes :
 - Préparation de la requête;
 - Association des paramètres;
 - Exécution de la requête
- L'avantage d'utiliser les requêtes préparées est d'éviter les attaques de type « *sql injection* »
- SQL Injection se définit par le fait qu'un utilisateur malveillant puisse insérer ou modifier des requêtes SQL via l'interface web de l'application (formulaires, url, ..)

Les requêtes préparée -Exemple

- Exemple d'une requête préparée (procédural)

```
$db = mysqli_init();  
mysqli_real_connect($db, 'localhost', 'root', '', 'testemployes');  
$num = 1000;  
$requete = "SELECT * FROM employers WHERE noemp = ?";  
$stmt = mysqli_stmt_init($db);  
mysqli_stmt_prepare($stmt, $requete);  
mysqli_stmt_bind_param($stmt, 'i', $num);  
mysqli_stmt_execute($stmt);  
$rs = mysqli_stmt_get_result($db, $stmt);  
$data = mysqli_fetch_assoc($rs);  
var_dump($data);  
mysqli_free_result($rs);  
mysqli_close($db);
```

Les requêtes préparée -Exemple

- Exemple d'une requête préparée (Orienté objet) :

```
$mysqli = new mysqli('localhost', 'root', '', 'testemployees');  
$num = 1000;  
$stmt = $mysqli->prepare("SELECT * FROM employers WHERE noemp = ?");  
$stmt ->bind_param("i", $num);  
$stmt ->execute();  
$rs = $stmt->get_result();  
$data = $rs -> fetch_array(MYSQLI_ASSOC);  
var_dump($data);  
$rs->free();  
$mysqli->close();
```