PHP
&
Bases de données

PHP & BD

- Les 5 étapes pour l'écriture d'un script
  1. Se connecter au SGBD
  2. Sélectionner la BD
  3. Envoie d'une requête
  4. Récupération du résultat
  5. Se déconnecter
- Les exemples présentés se feront en utilisant la bibliothèque PDO (PHP Data Objects) qui permet de rendre l'application indépendante du SGBD.
- PDO est une bibliothèque orientée objet et utilise le mécanisme des exceptions.

PDO (PHP Data Objects)

- Création d'un objet instance de la classe qui tente de réaliser les **étapes 1** (connexion) et **2** (choix de la BD)

$bdd = new PDO(DSN, USER, PASSWD);

Le DSN (Data Source Name) est la chaîne de connexion spécifique au SGBD

```
$bdd = new PDO("mysql:host=sql.free.fr; dbname=test", "dupont", "truc" );
```

- **Etape 5** : se déconnecter

pas de fonction PDO

déconnexion automatique en fin de script

PHP & BD

- Etape 1 & étape 2 : 4 paramètres nécessaires dans toutes les pages et variables suivant les hébergeurs :

→Création d'un fichier de configuration

config.inc.php

```
$server = " sql.free.fr" ;
$user = "dupont"  ;
$passwd = "truc" ;
$database = "test" ;
```

→ inclure le fichier de config au début de chaque script qui doit utiliser la BD include "config.inc.php"

- Etape 1 & étape 2 regroupés dans le fichier connect.php

```
include "config.inc.php"
// tentative de connexion avec gestion des exceptions
try {
    $bdd = new PDO("mysql:host=$server;dbname=$database",
                                $user, $passwd);
    $bdd->setAttribute(PDO::ATTR_ERRMODE,
                PDO::ERRMODE_EXCEPTION); // pour debug
}
catch (Exception $e) {
        die('Erreur : ' . $e->getMessage()); // die arrête le script
}
```


- **Etape 3 & 4** : Exécution d'une commande SQL

Il y a plusieurs étapes :

- écrire la requête
- préparer la requête
- exécuter la requête
- traiter le résultat

2 cas à distinguer :

→ requête SELECT

→ DELETE, UPDATE, INSERT

- **Exemple :** suppression de l'étudiant nommé Dupont

- écrire la requête

```
$req = "delete from etudiant where nom =  'Dupont' ";
```

- préparer la requête

```
$stmt = $bdd->prepare($req);
```

- exécuter la requête

```
$ok = $stmt->execute();
```

- traiter le résultat

```
if ($ok) {  echo "Dupont supprimé"; }
else { echo "Problème lors de la suppression"; }
```

→ $ok est un booléen qui permet se savoir si la mise à jour s'est bien passée

- **Exemple :** liste des étudiants dans l'ordre alphabétique

- écrire la requête

```
$req = "SELECT nom, prenom FROM etudiant ORDER BY nom,
   prenom";
```

- préparer la requête

```
$stmt = $bdd->prepare($req);
```

- exécuter la requête

```
$ok = $stmt->execute();
```

- traiter le résultat

→ **traitement plus complexe pour un Select**

PHP & BD (SELECT)

Etape 3 & 4: *exécution d'une commande Select*

$stmt contient le résultat

- Ce résultat contient un ensemble d'enregistrements
- Il n'est possible d'accéder qu'à un seul enregistrement à la fois
- Une fonction **fetch_...** permette de récupérer l'enregistrement courant (au début le premier) et de passer au suivant
- Pour parcourir tous les enregistrements, il faut utiliser une boucle.
- La fonction **fetch_...** renvoie faux si il n'y a plus d'enregistrements

PHP & BD

- **Exemple :** liste des étudiants dans l'ordre alphabétique

- traiter le résultat

```
// boucle pour afficher tous les étudiants
while ( $etud = $stmt->fetch() ) {
  // $etud est un tableau associatif qui contient autant de cases que le select d'attributs ; ici les 2 cases nom et prenom
   echo $etud["nom"] . " " .$etud["prenom"];
 }
```


Gestion des erreurs

- La méthode errorInfo() permet d'obtenir le message d'erreur de la dernière commande SQL exécutée sur le stmt :

```
$req = "SELECT nom, prenom FROM etudiant ORDER BY nom, prenom";
$stmt = $bdd->prepare($req);
$ok = $stmt->execute();
// affichage du message si erreur il y a
$errorInfo = $stmt->errorInfo();
if ($errorInfo[0] != 0) {
      print_r($errorInfo);
}
```


PHP & BD : paramètres

Commande avec paramètres

- Exemple : suppression d'un étudiant à partir de son code *(mais la requête doit fonctionner pour n'importe quel code)*

→ remplacer la valeur du code par un ?

```
$req = "delete from etudiant where codetu = ? ";
$stmt = $bdd->prepare($req);
```

- exécuter la requête en donnant une valeur pour le code

```
$ok = $stmt->execute(array(24));
```

→ la fonction execute accepte en paramètre un tableau avec autant de valeurs que de ? dans la requête SQL

→ la requête exécutée est `delete from etudiant where codetu=24`

PHP & BD : paramètres

- Il est alors possible d'exécuter la requête avec une autre valeur (sans refaire le prépare)

```
$ok = $stmt->execute(array(123));
```

→ la requête exécutée est `delete from etudiant where codetu=123`

PHP & BD : paramètres

Exemple avec plusieurs paramètres

- Ecrire une fonction permettant de modifier les nom et prenom d'un étudiant à partir de son code.

```
function modif_etud($p_nom,$p_prenom,$p_code) {
  $req = "update etudiant set nom=?, prenom=? where codetu = ? ";
  $stmt = $bdd->prepare($req);
  $ok = $stmt->execute(array($p_nom,$p_prenom,$p_code));
}
```

→ il doit y avoir autant de valeurs dans le tableau que de paramètres dans la requête (autant que de ?)

→ les ? sont remplacés par les valeurs dans l'ordre d'apparition

PHP & BD : paramètres

Utilisation de paramètres nommés

à la place des ?, on peut utiliser des paramètres nommés *(représentés par le nom du paramètre précédé de :)*

Exemple précédent :

```
$req = "update etudiant set nom=:nom, prenom=:prenom
where codetu = :code ";
```

avec bindParam ➔ attention fait le lien uniquement avec une variable (passage par référence)

PHP & BD : paramètres

Association de paramètres nommés avec des variables

Il est possible de faire le lien entre un paramètre d'une requête SQL et une variable PHP.

Cela se fait avant l'étape d'exécution à l'aide bindParam

Exemple précédent :

```
$req = "update etudiant set nom=:nom, prenom=:prenom
                where codetu = :code ";
$stmt = $bdd->prepare($req);
$stmt->bindParam(':nom', $p_nom);
$stmt->bindParam(':prenom', $p_prenom);
$stmt->bindParam(':code', $p_code);
$stmt->execute()
```

Quelques exemples

PHP & BD : tableau

- **Exemple1** : Tableau HTML contenant la liste des étudiants

```
<?php
include "config.inc.php"
include "connect.php"
// la requête
$req = "SELECT nom, prenom
        FROM Etudiant ORDER BY nom, prenom";
// préparation
$stmt = $bdd->prepare($req);
// exécution
$stmt->execute();
```


PHP & BD : tableau

```
// récupération des étudiants 1 par 1 et affichage dans
   une ligne d'un tableau HTML
echo '<table>';
while ($etud = $stmt->fetch() )
{
   echo '<tr><td>' ;
   echo $etud["nom"];
   echo '</td><td>';
   echo $etud["prenom"];
   echo '</td></tr>';
 }
echo '</table>';
?>
```


PHP & BD : formulaires

Exemple 2 : zone de liste d'un formulaire contenant la liste des étudiants pour en choisir un

La zone de liste affiche les nom et prénom des étudiants mais la valeur renvoyé est le code étudiant

```
<select name="etudiant">
 <option value="13"> jean dupont </option>
 <option value="21"> pierre durand </option>
 <option value="7"> leo leroy </option>
  ...
</select>
```

Les valeurs (code, nom et prenom de l'étudiant) doivent être récupérées dans la BD

PHP & BD : formulaires

Accès à la base pour construire le formulaire

```
<?php
…
$req = "SELECT code, nom, prenom
          FROM Etudiant ORDER BY nom, prenom";
$stmt  = $bdd->prepare($req);
$stmt->execute();
// zone de liste etudiant
echo '<select name="etudiant">';
while ($etud = $stmt->fetch())
{
   echo '<option value="'. $etud["code"].'">';
   echo $etud["prenom"].' '. $etud["nom"];
   echo '</option>';
 }
echo '</select>';
?>
```


PHP & BD : gestion de données

Formulaire de gestion de données

- Ex : suppression d'un étudiant


```
<form action="delete.php" method="post">
```

delete.php

```
// récupère le code de l'étudiant à supprimer
$code_etud=$_POST["etudiant"];
// exécution de la commande DELETE parametrée par le
   code etudiant
$stmt = $bdd->prepare("delete from etudiant
                                    where code= ?);
$ok = $bdd->execute( array($code_etud) );
```