Annexe 1 : Langage PHP et Bases de Données

1 Principes de MySQL

- Accéder à une base **MySQL** depuis **PHP** à l'aide de 2 étapes :
  1) **Accéder à la base** qui est sur un serveur
     a) Se **connecter au serveur**
        donner `nom du serveur` + `login/mdp` utilisateur
     b) **Ouvrir la base** de données
        donner le `nom de la base`
  2) **Lancer la requête SQL** puis **traiter** le résultat
     a) écrire la **requête dans une chaîne** puis **l'exécuter**
     b) **récupérer le résultat** et le **gérer**

2 Accès à la base

2.1 Se connecter à la base

- Utilisation de **PDO** : un outil qui permet de se connecter à n'importe quel type de BdD (MySQL, Oracle, PostgreSQL, ...)
  $\Longrightarrow$vérifier que **PDO** est activé
- En PHP : **Construction d'un objet PDO** en spécifiant 3 paramètres (3 chaînes de caractères) :
  - Le **type de BdD**, le **nom du serveur** et le **nom de la BdD**
  - Le **login**
  - Le **mot de passe**

```
$bd=new PDO("mysql:host=<nomServ>;dbname=<nomBdD>","<login>","<mdp>");
```

- Indiquer que les caractères sont codés en **utf8**

```
$bd->exec('SET NAMES utf8');
```

- **Se connecter une seule fois** au début de la page (idée : écrire un fichier `base.php` qui sera inclus)

2.2 Gérer les éventuelles erreurs

- En cas d'erreur du **new** : retour d'une exception (Java)
- Capturer une exception :

```
try{
   $bd=new PDO("...","...","...");
   $bd->exec('SET NAMES utf8');
}
catch (Exception $e){
   die("Erreur : Connexion à la base impossible");
}
```

On essaie d'exécuter le `new PDO` (`"try { ... }"`) :

- Si KO : exécution du `"catch { ... }"` :
  Instruction **die** : affichage du message passé en paramètre puis arrêt du script PHP
- Si OK : exécution du code situé après la partie `"catch { ... }"`

3 Requêtes

• Exécution d'une requête : 4 étapes
1- **Préparer** la requête : Instruction `prepare`
2- **Exécuter** la requête : Instruction `execute`
3- **Récupérer** les résultats : Instruction `fetchall`
4- **Détruire** la requête : Instruction `closeCursor`

• Étape 1 : **Préparation** de la requête
1- Stocker le code de la requête dans une variable
2- Appliquer la **méthode** `prepare` à l'objet PDO avec la requête SQL passée en paramètre
$\Longrightarrow$ Ne pas oublier de récupérer le résultat retourné

```
$sql = ...; // Stocke le code SQL de la requête
$req = $bd->prepare ($sql); // Requête préparée
```

• Étape 2 : **Exécution** de la requête
Appliquer la méthode `execute` à la requête préparée

```
$req -> execute (); // Requête exécutée
```

• Étape 3 : **Récupération** du résultat de l'exécution
- Appliquer la méthode `fetchall` au résultat obtenu suite à l'exécution de la requête
- Résultat du `fetchall` :
Un tableau contenant tous les enregistrements résultant de l'exécution de la requête :
  • 1 **case** contient 1 **enregistrement**
  • 1 **enregistrement** contient 1 **tableau associatif** dont
    - La **clé** = **nom du champ** dans la base
    - La **valeur** = **valeur du champ** dans la base

```
$lesEnreg = $req->fetchall (); // Requête exécutée
```

• Étape 4 : **Destruction** de la requête

```
$req -> closeCursor (); // Requête détruite
```

Exemple : (`mySQL_ex1.php`)

```
<!DOCTYPE html>

<html lang='fr'>
   <head>
      <meta charset='utf-8' />
      <title>MySQL - Exemple 1</title>
   </head>
   <body>
      <h1>MySQL - Exemple 1</h1>
      Cette page est le résultat du code ci-après : <br />
      <img src="Images/mySQL_ex1.png" alt="Code de mySQL_ex1.php"> : <br />

<?php
      try{
         $bd=new PDO("mysql:host=localhost;dbname=maBdD","login","mdp");
         $bd->exec('SET NAMES utf8');
      }
      catch (Exception $e){
         die("Erreur : Connexion impossible");
      }
```

```
        $sql="SELECT * FROM AVIONS WHERE Loc='Paris'";
        $req=$bd->prepare($sql); // étape 1 : Préparation
        $req->execute(); // étape 2 : Exécution
        $lesEnreg=$req->fetchall(); // étape 3 :Récupération
        $req->closeCursor(); // étape 4 : Destruction

        echo "<pres>";
        print_r ($lesEnreg);
        echo "</pres>";
?>

    </body>
</html>
```

4 Injections SQL

- Possibilité de construire la requête SQL :
  La requête étant une chaîne de caractères on peut la paramétrer en y insérant des variables (`$_POST`, ...)
  !!!! Mais attention aux INJECTIONS SQL !!!!!
- Injection SQL :
  - Un **formulaire** permet de saisir des valeurs qui seront ensuite utilisées dans une requête
  - L'**utilisateur**, au lieu de donner la valeur attendue, **tape une requête SQL**
  - La **requête de l'utilisateur** sera alors intégrée (injectée) dans votre requête ...
    Le code SQL qui sera exécuté contiendra la requête saisie par l'utilisateur !