PROGRAMMATION PHP-8 – MYSQL-8
MYSQL – PDO – INSERT – UPDATE - DELETE
L2S – 7-9 AVRIL 2025
3 JOURNEES

http://php.net/manual/fr/langref.php
http://www.w3schools.com/php/
https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql

SOMMAIRE

Edition : avril 2025

PHP – MYSQL – PDO - SUITE

00. Présentation du document

Exemples et exercices

- Les exemples sont présentés dans un chapitre en vert avec le mot clé : exemple-
- Les dossiers d'exemples sont fournis dans l'article qui contient ce fichier de cours.
  - ➔ http://bliaudet.free.fr/IMG/zip/PHP-03-PHP-MySQL.zip
  - ➔ Chargez ce fichier et mettez-le dans le dossier « php » du répertoire web « www » du serveur WEB. Vous pouvez aussi structurer les choses avec des dossiers J1, J2, etc. correspondant aux journées de travail.

- Les exercices à faire sont présentés dans un chapitre en jaune avec les mots-clés : TP- et exercice-
  - ➔ Les sources pour les exercices, quand il y en a, sont fournis dans le dossier des exemples.

C'est la deuxième partie

8. Select variable et faille XSS : utilisation d'un formulaire, GET ou POST

Le problème : avoir une requêtes variables : where realisateur = ? (exemples-5)

Objectif

- Mettre une variable dans une requête :

```
$reqSQL = 'SELECT *
        FROM films
        WHERE realisateur like ?
        order by annee
';
```

  ➔ Le ? c'est une variable.

- D'où vient la variable ?

  ➔ D'un formulaire avec un $_GET ou un $_POST

Technique 1 : test en dur - exemple 5-index-1

On ne fait plus :

```
$reqPHP = $bdd->query($reqSQL);
```

On va faire :

```
$reqPHP = $bdd->prepare($reqSQL);
$reqPHP->execute(array('%manki%')); //'manki%' prend la place du ? dans $reqSQL
```

Technique 2 : faille XSS - exemple 5-index-2

On met le un $_GET dans la $reqSQL :

```
$reqSQL = 'SELECT *
    FROM films
    WHERE realisateur = \'' . $_GET['realisateur'] . '\'
    order by realisateur, annee
';


$reqPHP = $bdd->query($reqSQL);
$films = $reqPHP->fetchAll();
```

Il y a un risque de faille XSS : si on saisit : toto \' or \'a\'=\'a, le select renverra toute la table !
Il faut donc éviter ça !

Technique 3 : sans faille XSS - exemple 5-index-3

Pour éviter la faille XSS, on écrit la requête avec un ou des « ? ».
Dans le execute() on passe en paramètres la ou les valeurs pour prendre la place du ou des « ? »

```
$reqSQL='SELECT *
    FROM films
    WHERE realisateur = ?
    order by realisateur, annee
';


// prepare et execute : version avec ? dans le reqSQL
$reqPHP=$bdd->prepare($reqSQL);
$reqPHP->execute(array(
    $_GET['realisateur']
));
```

Technique 4 : sans faille XSS, avec alias - exemple 5-index-3bis

- Une autre écriture, plus lisible, qui évite les « ? », est possible :
- On remplace les « ? » par des alias « :nom »
- Dans le execute() on passe des couples clé-valeur : la clé correspond au nom de l'alias.
- **C'est la solution qu'on va privilégier.**

```
$reqSQL='SELECT *
    FROM films
    WHERE realisateur = :realisateur
    order by realisateur, annee
';
echo $reqSQL.'<br/>';

// prepare et execute : version avec ? dans le reqSQL
$reqPHP=$bdd->prepare($reqSQL);
$reqPHP->execute(array(
    'realisateur'=> $_GET['realisateur']
));
```

Exercice : (6) tester les exemples

Exercice : (7) dans l'exercice MVC de SELECT précédent :

Ajoutez un select variable avec un formulaire (sur users)

9. DML

Ajouter, modifier, supprimer des données dans une table (exemple-6-insert-update-delete)

Select via phpMyAdmin – Afficher

- On peut utiliser l'interface graphique.
- On peut aussi entrer une commande SQL.

DML via phpMyAdmin – SQL

- On peut entrer les commandes SQL : INSERT, UPDATE et DELETE.
- Le système propose un pré-remplissage des commandes.

DML en PHP : INSERT

```
$reqSQL='
    INSERT INTO films
    (titre, realisateur, annee) VALUES
    (:titre, :realisateur, :annee)
';
```

```
//exemple('Dead Man','Jim JarmushKing Vidor','1995');

$requete=$bdd->prepare($reqSQL)

$resultat=$req-> execute(array(
    'titre'=>$_GET['titre'],
    'realisateur'=>$_GET['realisateur'],
    'annee'=>$_GET['annee']
)); // or die(print-r($bdd->errorInfo())) ;

/* le or die est inutile avec la connexion en ERRMODE */
/* $resultat pour traiter les erreurs proprement, sans ERRMODE */
```

DML en PHP : DELETE

```
$reqSQL='
    DELETE FROM films
    WHERE titre = :titre AND realisateur = :realisateur'
;

$requete=$bdd->prepare($reqSQL);
$resultat=$requete->execute(array(
    'titre'=>$_GET['titre'],
    'realisateur'=>$_GET['realisateur']
));

/* pour tester le résultat : 0 si pas de DELETE */
if($requete->rowCount() ){ // rowCount compte le nombre de delete
    echo '<br/>DELETE effectué ' .$requete->rowCount(). ' fois';
}
else {
    echo '<br/> Le DELETE a échoué';
}
```

- rowCount permet de savoir combien de delete on été effectués.
- http://www.astuces-webmaster.ch/page/mysql-pdo

ATTENTION au DELETE !!

- Attention au delete : quand une donnée est supprimée, on ne peut pas la récupérer si on est en mode validation (autocommit) ce qui est le plus fréquent !

- Il faut donc faire des vérifications, par exemple :

```
if (
   !isset($_GET['realisateur']) or
   !isset($_GET['titre']) or
   $_GET['realisateur']=='' or
   $_GET['titre']==''
){
   echo '<br/> Vous n\'avez pas saisi tous les paramètres';
}
```

DML en PHP : UPDATE

```
$reqSQL='
          UPDATE films
          SET duree=:duree
          WHERE titre = :titre AND realisateur = :realisateur
';
```

ATTENTION à l'UPDATE!!

- Attention à l'UPDATE : quand une donnée est modifiée, on ne peut pas la récupérer si on est en mode validation (autocommit) ce qui est le plus fréquent !

Bons usages

- A la place de :

```
'titre'=>$_GET['titre']
```

- on aura

```
'titre'=>$titre
```

➔ Les variables $titre, $realisateur, etc. seront récupérées via un $_POST ou un $_GET.

TP

Tester les exemples du cours

Dans le dossier de code fourni avec le cours, tester les exemples cours.
Regardez le code.
Cherchez à comprendre l'organisation.

Tester BD_WEB_PHP_MVC

Dans le dossier de code fourni avec le cours, tester le code du dossier BD_WEB_PHP_MVC.
C'est un code MVC avec une entrée unique.
Il permet de faire un SELECT, un INSERT et un DELETE.
Testez l'application.
Cherchez à comprendre l'organisation du code.
.

Projet Posts

Dans le dossier J2, il y a un corrigé du précédent TP pour le projet POST.
Il y a 3 versions :

- V1 : le HTML a été transformé en PHP. On include les header, nav commun. On gère la BD. On une page qui affiche les posts, une page qui affiche les users (avec des SELECT dans la BD).
- V2 : on a un premier niveau de structuration MVC avec les dossiers vues et modeles et 3 contrôleurs à la racine.
- V3 : on finalise : on ajoute un dossier ctrls pour les contrôleurs : il faut bien gérer tous les liens. Tout part du dossier ctrls. On crée un index qui appelle le contrôleur d'accueil. On rajoute un menu pour les comments.

Testez ces versions.
Cherchez à comprendre l'organisation du code.

On va ajouter un insert et un delete pour les users en suivant le modèle de BD_WEB_PHP_MVC.
Vous testerez en ajoutant un user puis en le supprimant.
Vous testerez le delete avec les autres user.