

TP PHP & Mysql Avec HTML

TOUS LES FICHIERS CREEES ICI SERONT SUR VOTRE ESPACE PERSONNEL SUR LE SITE WEB
Pensez à organiser votre espace : dossier CSS et dossier IMAGES.

Accès en PHP aux bases de données par les objets PDO

Objectif : On veut afficher des informations en provenance d'une base de données grâce à un script PHP

Pré-requis : Les TP Formulaire et sql1-TPSql avec PHP

Description de l'objectif :

Dans un TP précédent vous avez créé une page qui teste un nom d'utilisateur et un mot de passe issu d'une base de données.

La requête SQL ne renvoyait qu'une seule valeur et donc ne nécessitait pas de traitement dans une boucle.

Souvent, il faut afficher (sur une page HTML par exemple), les données issues d'une commande SELECT qui fournit un tableau de plusieurs lignes et plusieurs colonnes. C'est donc une variable complexe de type TABLEAU qu'il faut traiter dans une boucle qui lit chaque ligne et chaque colonne.

Dans ce TP, vous allez intégrer ces données dans une liste HTML (<SELECT>) et dans un tableau HTML (<TABLE>)

La base de données : Serveur **10.69.88.1**, base **adl**, nom d'utilisateur : **templabo**, Mot de passe : **templabo**

Travail 1 : Testez l'accès à la base de données et observez sa structure et son contenu.

Rappel : vous pouvez utiliser la commande « mysql.exe » en mode console Windows, ou un logiciel graphique comme « HeidiSQL ».

Testez les commandes SQL suivantes :

1) Liste des dates disponibles dans la table (uniquement le mois et l'année) :

```
SELECT DISTINCT (LEFT (DATE, 7)) AS 'DATE' FROM temperature;
```

date
2017-07
2017-09
2017-10
2017-11
2017-12
2018-01
2018-02

2) Liste des températures moyennes pour chaque jour d'un mois donné (Ex : 2018-02, Février 2018) :

```
SELECT LEFT (DATE, 10) AS 'DATE', AVG(t)/1000 AS 'TEMP MOY'  
FROM temperature  
WHERE LEFT (DATE, 7) = '2018-02' GROUP BY LEFT (DATE, 10)
```

Date	Temp Moy
2018-02-01	25.59735417
2018-02-02	25.35131250
2018-02-03	21.22375000
2018-02-04	19.66772917
2018-02-05	22.24594000
2018-02-06	23.70810417
2018-02-07	23.69639583
2018-02-08	23.81555100

Travail 2 :

Ouvrez le fichier **index.php** :

On y trouve une zone php contenant :

- La connexion au serveur SQL par les Php Data Object (PDO)
- L'exécution de la requête de la question 1 du Travail 1 (requête préparée)
- L'affichage du résultat « brut » avec la commande php : **print_r** pour comprendre la structure du tableau.

A FAIRE : Observez le tableau à 2 dimensions affiché par la commande **print_r**

Travail 3 :

Dans le formulaire HTML, on va compléter la balise <SELECT> en créant les <OPTION> avec du PHP (si besoin, revoyez l'utilisation des balises <select> <option> sur w3schools).



Voilà, nous y sommes !!!

On va encore mélanger du code HTML et PHP !

C'est inévitable MAIS on peut faire en sorte que ce soit « beau » :

Remarquez dans le fichier fourni : une balise < ?php ?> qui affiche une variable **\$lesOptions** dans le HTML. C'est une méthode propre pour séparer le gros du PHP et le HTML.

La variable sera créée dans la zone PHP. Elle est ensuite intégrée dans le HTML :

Insérez ce code PHP en fin de PHP, testez et observez le résultat reçu par le navigateur :

```
foreach ( $data as $ligne ) // pour chaque ligne du tableau
{
    $lesOptions .= "<option value='$ligne[DATE]'> $ligne[DATE] </option>";
}
```

Explications :

Dans cet exemple, à chaque passage dans la *foreach*, on traite 1 ligne avec 1 colonne. Avec l'option « PDO::**FETCH_ASSOC** », on accède à la donnée par le nom de la colonne : DATE. C'est un tableau associatif.

On « fabrique » des balises HTML <option> en y incorporant des informations PHP, et on stocke le tout dans la variable **\$lesOptions**.

Le signe « **.=** » correspond à une concaténation : on ajoute une nouvelle <option> à chaque passage dans la boucle « foreach ».

TESTEZ le résultat, vous devez obtenir ceci :

2017-07	▼
2017-07	
2017-09	
2017-10	
2017-11	
2017-12	
2018-01	
2018-02	
2018-03	

Ensuite, affichez le code source de la page HTML pour voir les balises <option> ajoutée par le PHP.

Travail 4 : On cherche à obtenir le résultat suivant : Quand on appuie sur Valider, le tableau avec la liste des températures moyenne du mois choisi s'affiche :

2018-03	▼	Valider
---------	---	---------

Date	Moyenne
2018-03-01	20.92556250
2018-03-02	20.43595833
2018-03-03	18.21200000
2018-03-04	16.96725000
2018-03-05	16.16245833
2018-03-06	15.65862500
2018-03-07	15.31358333
2018-03-08	15.21062500
2018-03-09	18.52062500
2018-03-10	17.41383333
2018-03-11	17.44377083
2018-03-12	20.60518750
2018-03-13	21.93727083
2018-03-14	22.02672727

Pour y parvenir, voici quelques rappels des TP précédents :

- a. Le formulaire n'a pas d'action donc le bouton OK recharge la page
- b. Comme la *method* est GET et l'attribut du <select> est *name='mois'*, la variable PHP `$_GET['mois']` contiendra la valeur choisie.
- c. Un tableau HTML se fait avec la balise <table> et les balises <tr> <th> et <td>
- d. Pour tester si on a choisi une date et appuyé sur le bouton valider, il faut tester l'existence de la variable PHP `$_GET['mois']`

```
Exemple :      if ( isset($_GET['mois']) )
                {
                    // Faire la requête de la question 2 du travail 1
                    // en y intégrant la date choisie : $_GET['choix']
                    // Intégrer le résultat de la requête à un
                    // tableau HTML
                }
```