L'objectif de ces exercices est de créer une interface de consultation et de mise à jour (saisie / modification / suppression) des données de la table MATIERE créée lors des exercices pratiques (4).

Page web modèle : http://eric.univ-lyon2.fr/~jdarmont/docs/web/td7.php

1. Recopier le script `ex4req.php` sous le nom `ex5.php`. N'y conserver que l'affichage de la table MATIERE.
2. Placer le code de l'affichage de la table dans une fonction nommée `liste()` prenant en paramètre l'identifiant de connexion. Ajouter un lien « Nouvelle matière » sous le tableau d'affichage. L'URI cible est la page courante (variable d'environnement PHP_SELF) assortie du paramètre « action=form_ajout ». Vérifier le bon fonctionnement de la fonction `liste()` en l'appelant depuis le programme principal.
3. Écrire une fonction nommée `formulaire()` prenant en paramètre une action cible. Cette fonction doit afficher le formulaire dont la description suit.
    a. URI cible (attribut action) : la page courante, méthode : get
    b. Champ caché de nom « action » et de valeur le paramètre « action cible » de la fonction
    c. Champ texte de nom « codemat » de taille 3 et de taille maximum 3
    d. Champ texte de nom « libelle » de taille 20 et de taille maximum 20
    e. Champ texte de nom « coef » de taille 3 et de taille maximum 3
    f. Boutons « reset » et « submit »
4. Écrire une fonction nommée `maj()` prenant en paramètres l'identifiant de connexion et une requête de mise à jour. Cette fonction doit afficher la requête, l'exécuter puis appeler la fonction `liste()` en lui transmettant en paramètre l'identifiant de connexion.
5. Dans le programme principal, définir un `switch` sur l'action demandée par l'utilisateur (`$_GET["action"]`).
    a. Si l'action est « form_ajout », appeler la fonction `formulaire()` avec le paramètre « ajout » (qui représente l'action suivante).
    b. Si l'action est « ajout », placer dans une variable la requête d'insertion dans la table MATIERE à partir des données transmises depuis le formulaire à l'aide de la méthode « get ». Appeler la fonction `maj()` en lui passant en paramètres l'identifiant de connexion et la requête que vous venez de construire dynamiquement.
    c. L'action par défaut est l'affichage de la liste des matières, effectué en appelant la fonction `liste()` en lui passant en paramètre l'identifiant de connexion.
6. Tester l'ajout de quelques matières.
7. Dans la fonction `liste()`, ajouter à chaque ligne du tableau une cellule contenant un lien « Modification ». L'URI cible est la page courante assortie des paramètres « action=form_modif », « codemat=*code de la matière courante* », « libelle=*libellé de la matière courante* » et « coef=*coefficient de la matière courante* ».

8. Modifier la définition de la fonction `formulaire()` en lui ajoutant trois paramètres (`$codemat`, `$libelle` et `$coef`), tous avec une valeur par défaut vide. Dans le corps de la fonction, ajouter un champ caché de nom « codemat_old » et de valeur `$codemat`. Dans la définition de chacun des champs « codemat », « libelle » et « coef », ajouter une valeur par défaut égale aux paramètres `$codemat`, `$libelle` et `$coef` , respectivement.
9. Dans le programme principal, ajouter les deux options suivantes au `switch`.
    a. Si l'action est « form_modif », appeler la fonction `formulaire()` avec les paramètres « modif » plus les données transmises par URI (code de matière, libellé et coefficient, dans cet ordre).
    b. Si l'action est « modif », placer dans une variable la requête de modification de la table MATIERE à partir des données transmises depuis le formulaire à l'aide de la méthode « get ». Le code de matière pouvant être modifié, exploiter le champ « codemat_old ». Appeler la fonction `maj()` en lui passant en paramètres l'identifiant de connexion et la requête que vous venez de construire dynamiquement.
10. Tester la modification de quelques matières.
11. Dans la fonction `liste()`, ajouter à chaque ligne du tableau une cellule contenant un lien « Suppression ». L'URI cible est la page courante assortie des paramètres « action=suppr » et « codemat=*code de la matière courante* ».
12. Dans le programme principal, ajouter l'option suivante au `switch`. Si l'action est « suppr », placer dans une variable la requête de suppression dans la table MATIERE du code de matière transmis par URI. Appeler la fonction `maj()` en lui passant en paramètres l'identifiant de connexion et la requête que vous venez de construire dynamiquement.
13. Tester la suppression de quelques matières.
14. Vérifier la validité du code HTML et CSS de tous les éléments de la page `ex5.php` (affichage de la table MATIERE et formulaire de saisie/modification, notamment).
15. L'utilisation de la méthode « get » est-elle la plus sûre ? Comment remédier à ses inconvénients ?
16. Question subsidiaire : ajouter un formulaire de confirmation de la suppression d'une matière.

Correction

```
<!DOCTYPE html>

<html>

    <head>
      <meta charset="utf-8" />
      <meta name="Author" content="Jérôme Darmont" />
      <meta name="Keywords" content="Programmation,Web,PHP,MySQL" />
      <meta name="Description" content="PHP &amp; MySQL" />
      <title>PHP &amp; MySQL : Mise à jour de base de données</title>
      <style type="text/css">
            table, th, td {
                  border: 1px solid black;
            }
            table {
                  border-collapse: collapse;
            }
            caption {
                  font-style: italic;
            }
            hr, fieldset {
                  width: 250px;
            }
            hr {
                  margin-left: 0;
            }
      </style>
    </head>

    <body>
      <?php
            include("connect.inc.php");

            function liste($idc) { // Liste des matières
                  $q = "SELECT * FROM MATIERE";
                  $res = $idc->query($q);
                  echo "<table>\n <caption>Liste des matières</caption>\n";
                  echo "<tr> <th>codemat</th> <th>libelle</th> <th>coef</th>
                   </tr>\n";
                  foreach ($res as $l)
                        echo "<tr> <td>" . $l["codemat"] . "</td> <td>" .
                              $l["libelle"] . "</td> <td>" . $l["coef"] .
                              "</td> <td><a href=\"" . $_SERVER["PHP_SELF"] .
                              "?action=form_modif&amp;codemat=" .
                              $l["codemat"] . "&amp;libelle=" . $l["libelle"] .
                              "&amp;coef=" . $l["coef"] .
                              "\">Modification</a></td> <td><a href=\"" .
                              $_SERVER["PHP_SELF"] .
                              "?action=suppr&amp;codemat=" . $l["codemat"] .
                              "\">Suppression</a></tr>\n";
                  echo "</table>\n";
                  echo "<p><a href=\"" . $_SERVER["PHP_SELF"] .
                        "?action=form_ajout\">Nouvelle matière</a></p>\n
                        <hr />\n";
            }

            function formulaire($action, $codemat = "", $libelle = "",
            $coef = "") { // Formulaire d'ajout/modification
                  echo "<form action=\"" . $_SERVER["PHP_SELF"] . "\"
                        method=\"get\">\n<fieldset>\n";
```

```
            echo "<input type=\"hidden\" name=\"action\"
                  value=\"$action\" />\n";
            echo "<input type=\"hidden\" name=\"codemat_old\"
                  value=\"$codemat\" />\n";
            echo "<p>Code matière : <input type=\"text\" name=\"codemat\"
                  size=\"3\" maxlength=\"3\" value=\"$codemat\" /></p>\n";
            echo "<p>Libellé : <input type=\"text\" name=\"libelle\"
                  size=\"20\" maxlength=\"20\" value=\"$libelle\" />
                  </p>\n";
            echo "<p>Coefficient : <input type=\"text\" name=\"coef\"
                  size=\"3\" maxlength=\"3\" value=\"$coef\" /></p>\n";
            echo "<p><input type=\"reset\" value=\"Annuler\" />
                  <input type=\"submit\" value=\"Valider\" /></p>\n";
            echo "</fieldset>\n</form>\n";
      }

      function maj($idc, $q) { // Mises à jour
            echo "<p>$q</p>\n";
            $idc->exec($q);
            liste($idc);
      }

      try {
            // Connexion
            $c = new PDO("mysql:host=$host;dbname=$dbname", $login,
                  $password);
            switch ($_GET["action"]) {
                  case "form_ajout": // Formulaire d'ajout
                        formulaire("ajout");
                  break;
                  case "ajout"; // Ajout
                        $q = "INSERT INTO MATIERE VALUES('" .
                              $_GET["codemat"] ."', '" .
                              $_GET["libelle"] . "', " .
                              $_GET["coef"] . ")";
                        maj($c, $q);
                  break;
                  case "form_modif": // Formulaire de modification
                        formulaire("modif", $_GET["codemat"],
                              $_GET["libelle"], $_GET["coef"]);
                  break;
                  case "modif": // Modification
                        $q = "UPDATE MATIERE SET codemat='" .
                              $_GET["codemat"] ."', libelle='" .
                              $_GET["libelle"] . "', coef=" .
                              $_GET["coef"] . " WHERE codemat='" .
                              $_GET["codemat_old"] ."'";
                        maj($c, $q);
                  break;
                  case "suppr": // Suppression
                        $q = "DELETE FROM MATIERE WHERE codemat='" .
                              $_GET["codemat"] ."'";
                        maj($c, $q);
                  break;
                  default: // Liste des matières
                        liste($c);
                  break;
            }
      } catch(PDOException $erreur) {
            echo "<p>Erreur : " . $erreur->getMessage() . "</p>\n";
      }
?>
```

```
      <p>Validation <a href="http://validator.w3.org/check/referer">HTML</a> |
<a href="http://jigsaw.w3.org/css-validator/check/referer">CSS</a></p>
    </body>

      <!-- Question 15 : L'utilisation de la méthode "get" fait apparaître
toutes les données en URI, ce qui la surcharge et peut être gênant lorsqu'on
transfère des données délicates comme des mots de passe. De plus, un utilisateur
mal intentionné peut s'en servir pour insérer des valeurs à sa convenance, voire
tenter de pirater le système par injection de code SQL malicieux. Il est donc
préférable d'utiliser la méthode "post", et donc de remplacer les liens
"Nouvelle matière", "Modification" et "Suppression" par des formulaires ne
contenant qu'un bouton. -->

</html>
```