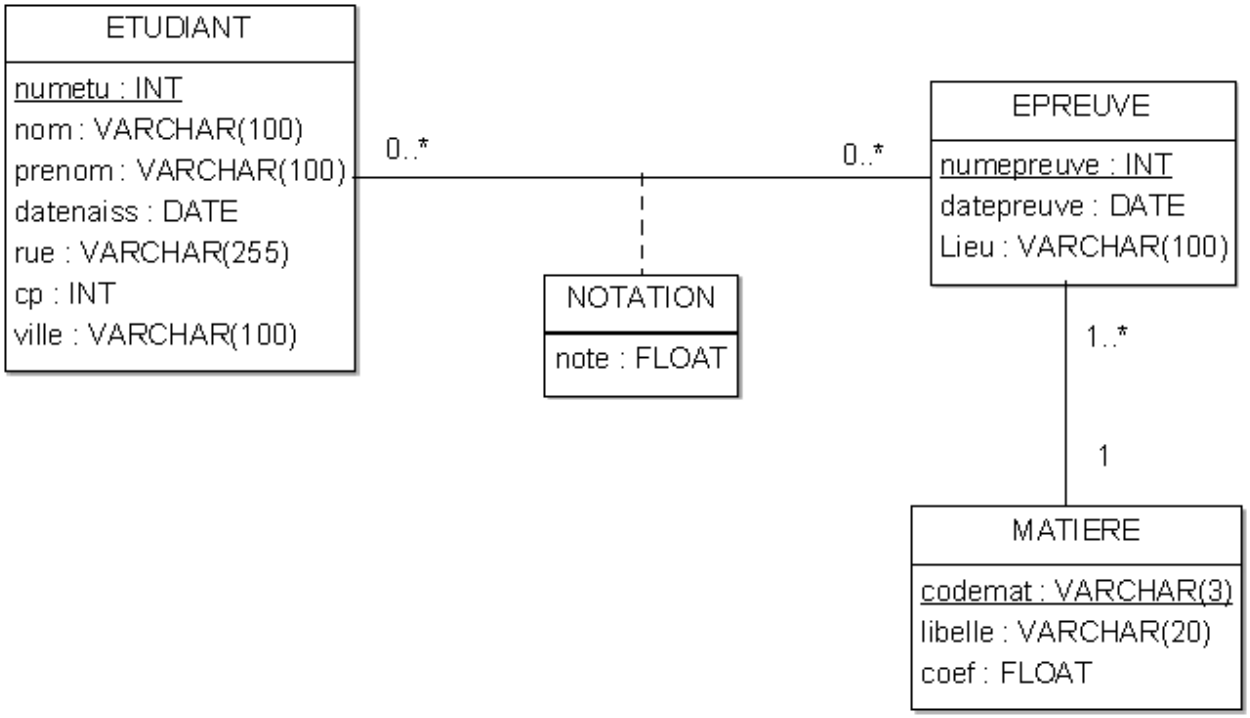
L3 MIASHS – Année 2018-2019
Programmation web
TD n° 4 : PHP objet, base de données et gabarits
J. Darmont (http://eric.univ-lyon2.fr/~jdarmont/), 04/03/19

Base de données

Soit la base de données « notation des étudiants » dont le **modèle conceptuel** est donné ci-dessous sous la forme d'un diagramme de classes UML.



La traduction du modèle conceptuel UML en **modèle logique** relationnel est donnée ci-dessous. Les attributs soulignés sont des clés primaires, ceux suivis d'un # des clés étrangères.

ETUDIANT (numetu, nom, prenom, datenaiss, rue, cp, ville)
MATIERE (codemat, libelle, coef)
EPREUVE (numepreuve, datepreuve, lieu, codemat#)
NOTATION (numetu#, numepreuve#, note)

Création de la base de données

1. En plus de lancer le serveur Apache depuis le panneau de contrôle de XAMPP, lancer également le serveur MariaDB si nécessaire.

2. Se connecter à phpMyAdmin (interface web écrite en PHP qui permet de gérer des bases de données MariaDB) à l'URL http://localhost/phpmyadmin/.

3. Depuis l'onglet « Bases de données » créer une nouvelle base de données nommée à votre convenance, puis la sélectionner.

4. Télécharger le fichier http://eric.univ-lyon2.fr/~jdarmont/docs/web/td4.sql, puis utiliser l'onglet « Import » pour insérer les tables et les données qu'il contient dans votre base.

5. Dans la partie gauche de l'écran de phpMyAdmin, vous devez maintenant pouvoir consulter la structure (**modèle physique**) et le contenu des tables que vous venez d'importer.

Connexion à la base de données

1. Créer un script PHP nommé connect.inc.php contenant uniquement l'affectation des valeurs idoines aux variables listées ci-dessous.
   ```
   $host = "localhost";
   $login = "root";
   $password = "";
   $dbname = "votre_base_de_donnees";
   ```
2. Créer le squelette d'une page web classique (en-tête, corps, etc.). La sauvegarder sous le nom td4.php.
3. Inclure le fichier connect.inc.php dans td4.php, puis créer un nouvel objet PDO (*PHP Data Object*) permettant de vous connecter à MariaDB en utilisant les informations saisies dans connect.inc.php. Tester. Remarque ?
4. Ajouter au script td4.php des instructions de gestion d'erreur (try/catch). En cas de connexion réussie, afficher un message du type « Connexion réussie. » (étonnant, non ?). En cas d'erreur, afficher l'erreur standard renvoyée par la méthode getMessage(). Tester de nouveau en introduisant des erreurs dans le nom du serveur, de la base de données et dans le couple (login, mot de passe) dans le fichier connect.inc.php, puis rétablissez la connexion correctement.

Interrogation de la base de données

Résultats attendus : http://eric.univ-lyon2.fr/~jdarmont/docs/web/td4.php?num=110

1. Dans td4.php, juste après la création de l'objet PDO (connexion à la base), ajouter à ce dernier l'attribut permettant de traiter erreurs de requête SQL comme des exceptions.
2. Créer un nouveau fichier nommé td4.class.php. Y créer une classe Requete contenant les attributs prives $pdo (un identifiant PDO), $nom et $req (requête SQL à exécuter). Créer un constructeur permettant d'initialiser ces attributs.
3. Inclure le fichier td4.class.php dans le script td4.php, juste après connect.inc.php. Tester. Si le message « Connexion réussie » ne s'affiche pas, c'est qu'il y a une erreur de syntaxe dans td4.class.php.
4. Dans td4.class.php, ajouter à la classe Requete un attribut prive $res et une méthode publique executer() qui, à l'aide de la connexion $pdo, prépare la requête $req et stocke le résultat dans $res. Exécuter ensuite $res.
5. Dans td4.php, créer un objet $qmat de classe Requete avec comme paramètres l'identifiant de connexion PDO créé lors de l'exercice précédent, le nom de requête « Liste des matières » et la requête SQL qui permet d'obtenir la liste de toutes les matières, avec tous leurs attributs descriptifs.
6. Appeler la méthode executer() pour $qmat. Tester. Si rien ne se passe, la requête s'est exécutée correctement. Introduire une erreur dans la requête et vérifier que le catch() la détecte bien. Si ce n'est pas le cas, déplacer la création de $qmat et l'appel à executer dans le bloc try ou revoir la question 1.
7. Dans td4.class.php, ajouter à la classe Requete un attribut privé $atts qui sera le tableau des noms des attributs à afficher dans le résultat de la requête. Ajouter l'initialisation de $atts au constructeur.
8. Créer un gabarit nommé td4.tpl.html qui définit un tableau HTML dont la légende est un champ {nom} et dont les données sont formées par deux blocs <tr> / <td> imbriqués, l'un pour les lignes de résultats d'une requête, l'autre pour les valeurs des attributs de chaque ligne, que l'on peut nommer {ligne.attribut.valeur}.
9. Télécharger le moteur de *templates* du phpBB Group à l'URL suivante : http://eric.univ-lyon2.fr/~jdarmont/docs/template.zip. Décompresser l'archive et copier le fichier template.class.php dans le répertoire courant de votre espace web. Dans td4.class.php, inclure template.class.php avant la définition de la classe Requete.
10. Ajouter à la classe Requete une méthode publique afficherTab() qui exploite le gabarit td4.tpl.html créé à la question 8. Cela nécessite notamment de parcourir les lignes de $res (foreach) et, pour chaque ligne, de parcourir les noms des attributs dans $atts (foreach). L'objectif est d'afficher *la valeur* de chaque attribut pour chaque ligne.
11. Dans td4.php, modifier la création de $qmat en ajoutant la liste des attributs à afficher en paramètre (« codemat », « libelle », « coef » – rappel : cette liste est un tableau de chaînes de caractères). Appeler la méthode afficherTab() pour $qmat après l'appel d'executer(). Tester.

12. Toujours dans td4.php, créer un objet $qnot de classe Requete avec comme paramètres l'identifiant de connexion PDO, le nom « Liste des notes », la requête SQL qui permet d'obtenir la liste des notes par étudiant.e et matière, ainsi que la liste d'attributs « nom », « prenom », « note » et « libelle ». Appeler les méthodes executer() et afficherTab() pour $qnot. Tester.
13. Créer un objet $qetu de classe Requete avec comme paramètres l'identifiant de connexion PDO, le nom « Moyenne par étudiant », la requête SQL qui permet d'obtenir la moyenne des notes (sans pondération) de chaque étudiant.e, ainsi que la liste d'attributs « numetu », « nom », « prenom », « moyenne ». Appeler les méthodes executer() et afficherTab() pour $qetu. Tester.
14. Affecter à une variable quelconque un numéro d'étudiant.e. Répéter la question précédente, mais uniquement pour cet étudiant.e.
15. Afin de rendre la dernière partie du script td4.php plus dynamique (question 13), permettre la saisie du numéro d'étudiant dont on veut la moyenne dans l'URL de la page.
16. Vérifier la validité du code HTML et CSS de votre page td4.php.

Questions subsidiaires :

17. Modifier le gabarit td4.tpl.html et la méthode afficherTab() afin d'afficher le nom des attributs retournés par la requête en entête du tableau résultat.
18. Créer un nouveau gabarit et ajouter une méthode afficherListe() à la classe Requete pour afficher les résultats de requête dans une liste à puce (en séparant les valeurs des attributs par des virgules).

Correction

```
<?php // connect.inc.php
    $host = "localhost";
    $login = "darmont";
    $password = "XXX";
    $dbname = $login;
?>
```

```
<table> <!-- td4.tpl.html -->
    <caption>{nom}</caption>
    <tr>
        <!-- BEGIN tableau -->
        <th>{tableau.entete}</th>
        <!-- END tableau -->
    </tr>
    <!-- BEGIN ligne -->
    <tr>
        <!-- BEGIN attribut -->
        <td>{ligne.attribut.valeur}</td>
        <!-- END attribut -->
    </tr>
    <!-- END ligne -->
</table>
```

```
<?php // td4.class.php

    require("template.class.php");

    class Requete {
        private $pdo;   // Identifiant de connexion
        private $nom;   // Nom de la requête
        private $req;   // Requête à exécuter
        private $res;   // Résultat de la requête
        private $atts;  // Attributs à afficher

        function __construct($param_pdo, $param_nom, $param_req, $param_atts) {
            $this->pdo = $param_pdo;
            $this->nom = $param_nom;
            $this->req = $param_req;
            $this->atts = $param_atts;
        }

        public function executer() {
            $this->res = $this->pdo->prepare($this->req);
            $this->res->execute();
        }

        public function afficherTab() {
            // Définition du gabarit
            $gab = new Template("./");
            $gab->set_filenames(array("body" => "td4.tpl.html"));
            // Légende
            $gab->assign_vars(array("nom" => $this->nom));
            // Entête
            foreach($this->atts as $att)
                $gab->assign_block_vars("tableau", array("entete" => $att));
```

```
            // Données
            foreach ($this->res as $ligne) {
                $gab->assign_block_vars("ligne", array("rien" => ""));
                foreach($this->atts as $att)
                    $gab->assign_block_vars("ligne.attribut",
                                            array("valeur" => $ligne[$att]));
            }
            // Affichage du gabarit
            $gab->pparse("body");
        }
    }
?>
```

/* tableau.css */

```
body {
    font-family: arial;
}

table, th, td {
    border: 1px solid black;
}

table {
    border-collapse: collapse;
}

caption {
    font-style: italic;
}
```

```
<!DOCTYPE html> <!-- td4.php -->
<html lang="fr">

    <head>
        <meta charset="utf-8" />
        <meta name="Author" content="Jérôme Darmont" />
        <meta name="Keywords" content="Programmation,Web,PHP,MariaDB" />
        <meta name="Description" content="PHPobjet Base de données Templates" />
        <title>Interrogation de base de données</title>
        <link rel="stylesheet" type="text/css" href="tableau.css" />
    </head>

    <body>
        <?php
            require("connect.inc.php");
            require("td4.class.php");
            try {
                // Connexion
                $c = new PDO("mysql:host=$host;dbname=$dbname", $login,
                             $password);
                echo "<p>Connexion réussie.</p>\n";
                $c->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

                // Liste des matières
                $qmat = new Requete($c,
                                    "Liste des matières",
                                    "SELECT * FROM matiere",
                                    array("codemat", "libelle", "coef")
                );
                $qmat->executer();
                $qmat->afficherTab();
```

```
                // Liste des notes
                $qnot = new Requete($c,
                                    "Liste des notes",
                                    "SELECT nom, prenom, note, libelle
                                     FROM etudiant, notation, epreuve, matiere
                                     WHERE etudiant.numetu = notation.numetu
                                     AND notation.numepreuve=epreuve.numepreuve
                                     AND epreuve.codemat = matiere.codemat",
                                    array("nom", "prenom", "note", "libelle")
                );
                $qnot->executer();
                $qnot->afficherTab();

                // Moyenne par étudiant
                $qetu = new Requete($c,
                                    "Moyenne par étudiant",
                                    "SELECT e.numetu, nom, prenom,
                                            AVG(note) moyenne
                                     FROM etudiant e, notation n
                                     WHERE e.numetu = n.numetu
                                     GROUP BY nom, prenom",
                                    array("numetu", "nom", "prenom", "moyenne")
                );
                $qetu->executer();
                $qetu->afficherTab();

                // Moyenne d'un étudiant
                //$num = 110;
                $num = $_GET["num"];
                $q1etu = new Requete($c,
                                     "Moyenne de l'étudiant no $num",
                                     "SELECT e.numetu, nom, prenom,
                                             AVG(note) moyenne
                                      FROM etudiant e, notation n
                                      WHERE e.numetu = n.numetu
                                      AND e.numetu = $num
                                      GROUP BY nom, prenom",
                                     array("numetu", "nom", "prenom", "moyenne")
                );
                $q1etu->executer();
                $q1etu->afficherTab();
            } 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>

</html>
```