TP : Notation de films

On souhaite réaliser une interface web de notation de films par des spectateurs qui vote en ligne, en étant identifiés par un login + mdp.
Les résultats sont affichés et mis à jours en fonction des votes.

1. Mise en place de la base « notation »

- Créer son répertoire de travail

```
$ sudo mkdir -p /var/www/html/www.TP_films
$ sudo chmod -R 777 /var/www/html/www.TP_films
```

- La page d'accueil nommée **index.html** est fournie avec le TP. L'ouvrir et l'éditer pour repérer les éléments.

Le formulaire renvoie vers la page **traitementConnexion.php**
Cette page spécifique s'assure que les données saisies sont correctes (login et mot de passe), ainsi l'utilisateur est connecté.

- Se connecter sur **http://127.0.0.1/phpmyadmin/** et créer la base notations.
- Dans la base notations, créer la table **utilisateurs** : `Utilisateurs(id_utilisateur, login, mdp)`
- Ajouter les valeurs utilisateurs avec login et mdp :

1	bob	123
2	patrick	123
3	sandy	kungfu
4	krabs	dollar

On vérifie que tout est correct, avec les onglets « Structure » et « Parcourir » pour la table notation.

2. Traitement de la connexion

Les données saisies dans le formulaire d'accueil sont envoyées en *POST* à une page PHP **traitementConnexion.php**

- On va tout d'abord vérifier que les données sont bien reçues.
  Créer un fichier **traitementConnexion.php** et écrire les lignes :

```php
<?php
// récupère le login et mdp du formulaire en post et les stocke dans des variables php
$login = $_POST['login'];
$mdp = $_POST['mdp'];
echo "Bonjour ".$login."<br>";
echo "ton mot de passe est ".$mdp."<br>";
?>
```

- Ensuite on va essayer de se connecter à la base de données notations.
  Ajouter les lignes suivantes dans les balises PHP :

```php
$bdd = new PDO('mysql:host=localhost;port=3306;dbname=notations;charset=utf8', 'root', '*');
var_dump($bdd);    // affiche la valeur de $bdd
die();    // "tue" (arrête) le programme ici
```

(ATTENTION ! : le **'*'** doit être remplacé par le mot de passe **'root'** de phpMyAdmin)

Normalement, si la connexion s'est effectuée correctement, la page affiche : `object(PDO)#1 (0) { }`
C'est à dire que l'on a bien un objet de la classe ***PDO.***
(https://www.php.net/manual/fr/book.pdo.php)

> ***PHP Data Objects (PDO)*** *est une extension définissant l'interface pour accéder à une base de données avec PHP.*
> *Elle est orientée objet, la classe s'appelant PDO. PDO constitue une couche d'abstraction qui intervient entre l'application PHP et un système de gestion de base de données (SGDB) tel que MySQL, PostgreSQL ou MariaDB par exemple. La couche d'abstraction permet de séparer le traitement de la base de données proprement dite. PDO facilite donc la migration vers un autre SGBD puisqu'il n'est plus nécessaire de changer le code déjà développé. Il faut seulement changer les arguments de la méthode envoyés au constructeur.*
> *Pour récupérer les enregistrements d'une table de la base de données, la procédure classique en PHP consiste à parcourir cette table ligne par ligne en procédant à des aller-retour entre l'application PHP et le SGBD. Ceci risque d'alourdir le traitement surtout si les deux serveurs sont installés chacun sur une machine différente. PDO corrige ce problème en permettant de récupérer en une seule reprise tous les enregistrements de la table sous forme d'une variable PHP de type tableau à deux dimensions ce qui réduit le temps de traitement.*

Source : wikipédia https://fr.wikipedia.org/wiki/PHP_Data_Objects

- Ajoutons maintenant une requête SQL à l'aide des méthodes ***prepare()*** et ***execute()*** :

```php
$requete = $bdd->prepare("SELECT * FROM utilisateurs"); // effectue une requête dans la base
$requete->execute();

$reponse = $requete->fetchAll(PDO::FETCH_ASSOC);    // affiche le résultat
var_dump($reponse);
```

Le résultat est écrit *en ligne,* mais il est de la forme :

```
array(4) {
  [0]=> array(3) { ["id_utilisateur"]=> string(1) "1" ["login"]=> string(3) "bob" ["mdp"]=> string(3) "123" }
  [1]=> array(3) { ["id_utilisateur"]=> string(1) "2" ["login"]=> string(7) "patrick" ["mdp"]=> string(3) "123" }
  [2]=> array(3) { ["id_utilisateur"]=> string(1) "3" ["login"]=> string(5) "sandy" ["mdp"]=> string(6) "kungfu" }
  [3]=> array(3) { ["id_utilisateur"]=> string(1) "4" ["login"]=> string(5) "krabs" ["mdp"]=> string(6) "dollar" }  }
```

On obtient à chaque fois un tableau de clés/valeurs représenté par les *array*.
array() est un objet de type dictionnaire qui permet de voir apparaître la structure de la base telle que nous l'avons définie.
Ce tableau contient 4 lignes qui contiennent chacune les valeurs de nos 3 attributs.

Note :
La méthode ***query()*** peut être également utiliser pour effectuer une requête, mais les requêtes effectuées par ***prepare()*** et ***execute()*** sont plus sûres, notamment dans les cas comme le notre où l'on passe des paramètres en *POST* (risque avec l'injection de code malveillant). https://www.php.net/manual/fr/pdo.prepare.php

- Tester des requêtes diverses sur cette base.

3. Affichage du contenu de la base

Maintenant que la base est accessible et que l'on sait comment y récupérer des valeurs, nous allons les afficher plus « proprement » et nous en servir.

L'objectif est d'afficher sur la page **traitemenConnexion.php** la liste des films qui sont à noter.

- Créer une seconde table nommées **films** : `Films(id_films, nom, realisateur)`
  Les valeurs contenues sont :

1	Dune	Denis Villeneuve
2	Le sommet des dieux	Patrick Imbert
3	Bac nord	Cédric Gimenez
4	Le dernier duel	Ridley Scott

- Avec la méthode précédente, effectuer une requête pour afficher les noms des films.
  La méthode ***var_dump()*** permet d'obtenir la structure suivante :

```
array(4) {
  [0]=> array(1) { ["nom"]=> string(4) "Dune" }
  [1]=> array(1) { ["nom"]=> string(19) "Le sommet des dieux" }
  [2]=> array(1) { ["nom"]=> string(8) "Bac nord" }
  [3]=> array(1) { ["nom"]=> string(15) "Le dernier duel" } }
```

On voit que la clé « nom » correspond dans chaque cas à une valeur différente.

- Supprimer l'affichage les valeurs en effaçant les lignes :

```
var_dump($reponse);
die();
```

Nous allons parcourir l'objet ***$reponse*** à l'aide d'une boucle pour afficher chacune des valeurs. Pour cela, on utilise l'instruction **foreach( ) { }**.

```
foreach($reponse as $value){
   echo $value;
}
```

- Modifier le code pour obtenir l'affichage ci-contre :

Bonjour bob

Les films à noter sont :

Dune

Le sommet des dieux

Bac nord

Le dernier duel

4. Authentification de la connexion

Maintenant, nous allons nous assurer que l'utilisateur existe et que son mot de passe est correct.

- Créer la requête spécifique de connexion suivante :

```
// on prépare une requête de connexion : test de la connexion 0 si erreur, 1 si utilisateur authentifié
// avec le couple de variables php login & mdp
$requeteCnx = $bdd->prepare("SELECT COUNT(*) AS cnx FROM utilisateurs WHERE login=? AND mdp=?");
$requeteCnx->execute([$login, $mdp]);
$connexion = $requeteCnx->fetch(PDO::FETCH_ASSOC);
```

Avec la méthode ***var_dump()***, on peut afficher ***$connexion*** et cela donne dans notre cas :

C'est bien une réussite de connexion !

```
array(1) {
  [0]=> array(1) { ["cnx"]=> string(1) "1" }
}
```

Essayer avec un autre mot de passe…

- Nous allons maintenant créer un message différent en cas de mauvaise connexion en utilisant une structure conditionnelle **if ( ) { } else { }**

```php
if ($connexion['cnx'] == "1"){
   echo "Connexion Ok !";
}else{
   echo "<h1>Hummm... Erreur de connexion !</h1>";
   echo "<a href='index.html'>retour page d'accueil...</a>";
}
```

Soit la connexion est correcte (le couple login + mdp existe), soit la connexion est incorrecte et la page affiche un lien pour retourner à l'accueil, la page **index.html**.

Essayer avec des mauvais mots de passe pour vérifier…

5. Une fois connecté, on va noter !

L'utilisateur étant connecté à la base, il faut le diriger vers une nouvelle page pour afficher la liste des films à noter… Sous forme de formulaire bien sûr !

- Toujours dans la page **traitementConnexion.php**, en cas de connexion réussie, il faut renvoyer l'utilisateur vers la nouvelle page **note.php**.

```php
if ($connexion['cnx'] == "1"){
  // récupération de l'id_utilisateur de la personne connecté
   $requete = $bdd->prepare("SELECT id_utilisateur FROM utilisateurs WHERE login=? AND mdp=?");
   $requete->execute([$login, $mdp]);
   $utilisateur_connecte = $requete->fetch(PDO::FETCH_ASSOC);
   $id = $utilisateur_connecte['id_utilisateur'];
   header('Location: note.php?id='.$id);    // envoi vers la page note.php
}else{
   echo "<h1>Hummm... Erreur de connexion !</h1>";
   echo "<a href='index.html'>retour page d'accueil...</a>";
}
```

On récupère au passage la valeur de l'identifiant de l'utilisateur que l'on transmet en *GET* à la page destination par la méthode ***header()***.

- Création d'une nouvelle page **note.php** :
  On reprend la structure et la mise en page de la page d'accueil. La structure est de base est donnée dans le fichier **structure_page_note.php** où le code comprend les deux parties distinctes, PHP pour le traitement et HTML pour l'affichage.

- Pour afficher les noms des films proposés à la notation, il faut d'abord se connecter à la base : dans **note.php**, ajouter dans la partie PHP la requête nécessaire pour obtenir l'ensemble des noms de films dans une variable PHP ***$films***.
  (Utiliser le ***fetchAll()***)

- Le formulaire étant assez « répétitif », nous allons utiliser une boucle **foreach( ) { }** comme vu précédemment pour afficher les propositions :

```
<form id="formNote" method="post" action="resultat.php">
<div class="formulaire">
<h1> Notez votre film !</h1>
<?php
  foreach($films as $film){
  echo '<div> <h2>'.$film['nom'].'</h2>
  <div>
  <label class="label"> 1 </label>
  <input type="radio" name="note" value="1">
  <input type="radio" name="note" value="2">
  <input type="radio" name="note" value="3">
  <input type="radio" name="note" value="4">
  <input type="radio" name="note" value="5">
  <label class="label"> 5 </label>
  </div>
  </div>';
  } ?>
<div class="bouton">
  <input type="submit" value="Envoyer" />
</div>
</div>
</form>
```

Vérifier que tout s'affiche correctement en affichant sur le navigateur, puis en utilisant l'inspecteur (touche F12).

- Dans ce formulaire, on envoie à la page **resultat.php**, mais chacune des notes fournies a le même nom !!

Il faut donc modifier la requête pour récupérer aussi l'identifiant du film ***id_film*** et l'insérer à la fin de la chaîne de caractère du nom de variable du formulaire :

```
$note_film = "note".$film['id_film']; // le nom du champ prend en compte l'id du film
echo '<div> <h2>'.$film['nom'].'</h2>
<div>
<label class="label"> 1 </label>
<input type="radio" name="'.$note_film.'" value="1">
```

On passe ici par une variable ***$note_film*** intermédiaire.

- Créer un fichier très simple **resultat.php** pour vérifier l'affichage des variables note1, note2, etc.

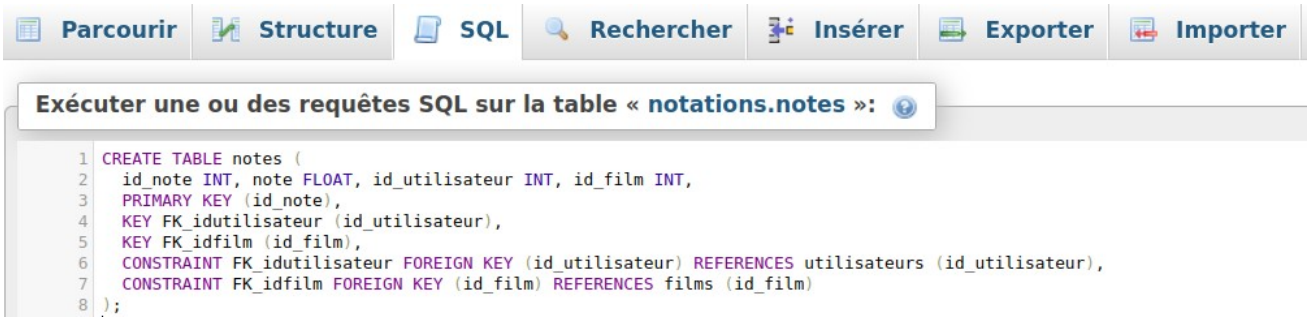
6. Saisir les notes dans la base

Pour saisir les notes dans la base, il faut d'abord créer une table notes de structure :

Notes(id_note, note, #id_utilisateur, #id_film)

Cette table prend en compte 2 clé étrangères : **id_utilisateur** et **id_film**.

- Pour créer cette table, dans phpMyAdmin, on se place dans la base notations et on effectue la requête SQL suivante :



On définit ainsi les 2 *FOREIGN KEY* et leurs contraintes respectives, du fait de leur appartenance à d'autres tables.

Le résultat dans l'onglet « structure » donne :

	#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut
☐	1	**id_note**	int			Non	*Aucun(e)*
☐	2	**note**	float			Oui	*NULL*
☐	3	**id_utilisateur**	int			Oui	*NULL*
☐	4	**id_film**	int			Oui	*NULL*

Ensuite, il va falloir récupérer dans la même page PHP les **id_utilisateur**, **id_film** et leur note associée pour pouvoir créer une ligne de valeurs dans la table **notes**.

- Dans la page **resultat.php**, afficher les valeurs de :
  - **id_utilisateur**, passé en *GET* depuis la page de connexion
  - **id_film** noté
  - **note** attribuée au film

Le rendu sera celui ci-dessous

l'identifiant utilisateur est : 1
Le film id N°1 est noté 1
Le film id N°2 est noté 3
Le film id N°3 est noté 3
Le film id N°4 est noté 4

Dans **note.php**, on remplacera l'attribut action du formulaire par :

```
action=<?php echo "resultat.php?id=".$id; ?>
```

De cette façon, l'**id_utilisateur** transitera jusqu'à la page **resultat.php**

Ensuite, il faut se connecter à la base et récupérer les **id_film** avec une boucle **foreach( )**

```
$requete_films = $bdd->prepare("SELECT id_film FROM films");
$requete_films->execute();
$films = $requete_films->fetchAll(PDO::FETCH_ASSOC);
foreach($films as $film){
   ... }
```

Tous ces résultats vont maintenant être écrits dans la table **notes.**

- Dans un premier temps, on récupère le maximum des id des notes de façon à ajouter à la table **notes** une nouvelle entrée. Il faut formuler une requête de type :

```
SELECT MAX(id_note) AS maxi FROM notes
```

Pour récupérer la valeur, il faudra utiliser la méthode ***fetch( )*** et non ***fetchAll( )***, car il n'y en a qu'une.

- Ensuite, dans le cas où une note a été saisie, on met à jour la table **notes** en insérant un nouveau tuple. Vérifier que la base se met bien à jour.

- Il reste à créer une page de résultat **affichage.php** qui affiche les notes globales des films.

On pourra utiliser la méthode header( ) vue plus haut qui renvoi automatiquement à une autre page, cette fois sans faire passer de paramètre :

```
header('Location: affichage.php);    // envoi vers la page affichage.php
```