|  |  |
| --- | --- |
| C:\Users\FAHMANI\Desktop\logo-ofppt-1.jpg | *CF TEMARA / ISTA TEMARA* |
|  | DEVOIR  Année de Formation 2022/2023 |

Filière / groupe : DEVOWFS202

## **Exercice 1 : Création de la base de données**

Connectez-vous sur mySQL et exécutez le  [script SQL](https://lipn.univ-paris13.fr/~lacroix/teaching/R301/TP/docs/TP2/database_nobel.sql). Ce script, une fois exécuté, crée les tables nobels et categories. La table categories contient les différentes catégories de prix nobels (chimie, paix, etc) et la table nobels contient les informations sur les différents prix nobels reçus entre 1901 et 2010.

1. Exécuter la requête SQL suivante et vérifier qu’il y a 839 prix nobels dans la base de données.

**SELECT** **count**(**\***) **FROM** nobels;

## **Exercice 2: Affichage des 25 derniers prix nobels**

1. Créer un script *last.php* qui affiche les nom et prénom, catégorie et année des 25 derniers prix nobels sous forme d’une table HTML (un prix nobel par ligne).

Requête SQL :

**SELECT** **\*** **FROM** nobels **ORDER** **BY** year **DESC** **LIMIT** 25

Le script doit utiliser la méthode fetch (dans une boucle while) pour récupérer les résultats.

1. Créer une deuxième version du script qui utilise la méthode fetchAll pour récupérer les résultats.

## **Exercice 3 : Affichage des informations d’un prix nobel**

Créer un script *infos.php* qui affiche les informations d’un prix nobel dont l’identifiant est passé en paramètre dans l’url. Les informations seront affichées sous la forme d’une liste non ordonnée. S’il n’y a aucun identifiant en paramètre, le script doit afficher le message Aucun identifiant. Si l’identifiant ne correspond à aucun prix nobel, le script doit afficher Identifiant invalide.

Requête SQL pour récupérer les informations du prix nobel d’identifiant 1 :

**Select** **\*** **from** nobels **WHERE** id **=** 1

## **Exercice 4 : Création de la classe Model**

1. Créer le fichier *Model.php*et définir la classe Model. Définir le constructeur permettant la connexion à la base de données. Il faudra indiquer que les données envoyées sont en utf-8 et que le mode d’erreur est la levée d’exceptions. La classe devra implémenter le design pattern singleton.
2. Définir la méthode *get\_last()*retournant les 25 derniers prix nobels décernés dans un tableau PHP.

Requête SQL :

**SELECT** **\*** **FROM** nobels **ORDER** **BY** year **DESC** **LIMIT** 25

1. Définir la méthode get\_nb\_nobel\_prizes() retournant le nombre total de prix nobels dans la base de données.

Requête SQL :

**SELECT** **count**(**\***) **FROM** nobels

## **Exercice 5 : Page d’accueil du site**

Décompresser l’archive ***PrixNobel.zip*** et ajouter le fichier *Model.php*dans le répertoire PrixNobel. En utilisant le modèle, modifier la page *home.php* pour afficher le nombre de prix nobels dans la base de données à la place de **TO FILL**.

## **Exercice 6 : Affichage des 25 derniers prix nobels**

Définir le script ***last25.php*** qui affiche sous forme d’une table HTML les nom et prénom, la catégorie et l’année des 25 derniers prix nobels décernés. La page doit ressembler à ça :



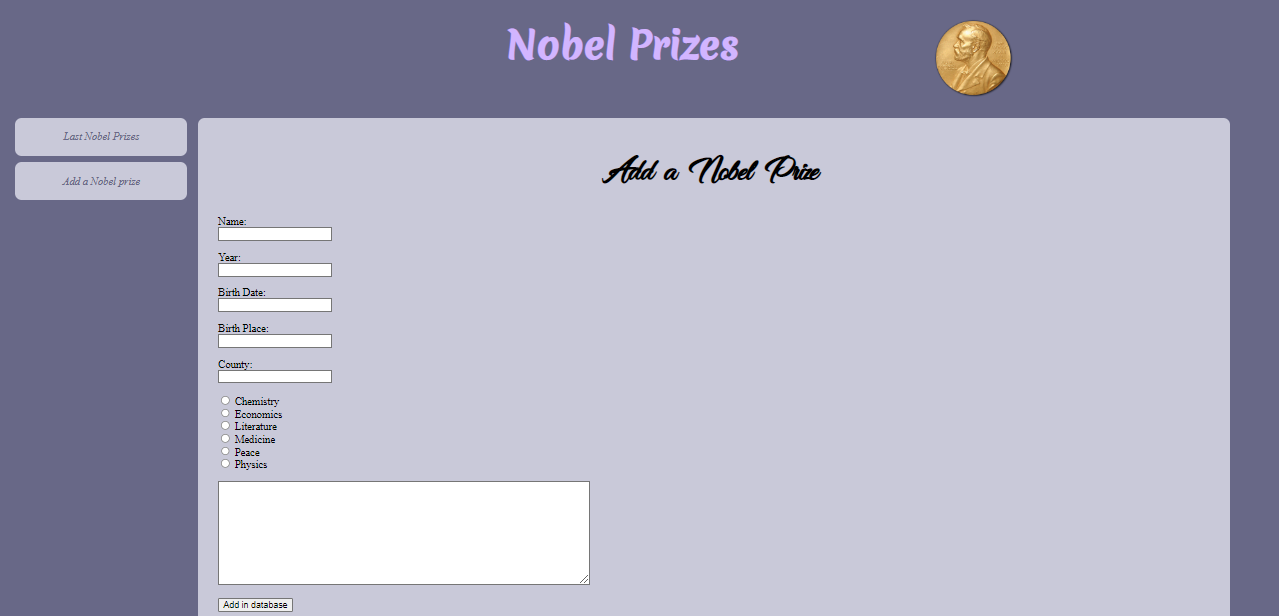
Le script utilisera la classe Model pour récupérer les informations sur les prix nobels.

**Remarque :** Ce script doit donc afficher la même table HTML que le script *last.php* mais en utilisant la classe Model (et le code CSS).

## **Exercice 7 : Informations concernant un prix nobel**

1. Définir la méthode *get\_nobel\_prize\_informations()*prenant en paramètre un entier et retournant un tableau contenant toutes les informations du prix nobel dont l’identifiant est égal à cet entier ou false s’il n’existe pas.
2. Définir le script *informations.php*. Ce dernier teste s’il existe dans l’url un paramètre id. Si ce dernier existe et correspond à un nombre entier strictement positif, il affiche toutes les informations du prix nobels correspondant s’il existe ou un message indiquant qu’il n’existe aucun prix nobel ayant cet identifiant dans la base de données.
3. Modifier le script *last25.php* pour que les noms et prénoms de chaque prix nobel soit un lien hypertexte sur le fichier *informations.php* contenant les informations de ce prix nobel.

## **Exercice 8 : Formulaire d’ajout d’un prix nobel**

1. Définir la méthode *get\_categories()* retournant un tableau de chaînes de caractères correspondant aux différentes catégories de prix nobels.
2. Dans le fichier *form\_add.php,* définir un formulaire pour ajouter un prix nobel dans la base de données. Ce formulaire doit ressembler à ce formulaire :  La soumission du formulaire appellera le script *add.php*. Les boutons radio doivent être générés en fonction des catégories présentes dans la base de données.

## **Exercice 9 : Ajout d’un prix nobel**

1. Définir la méthode add\_nobel\_prize() dans la classe Model. Cette méthode prend en paramètre un tableau $infos contenant les clés year, category, name, birthdate, birthplace, county et motivation ; elle ajoute dans la base de données un prix nobel avec les informations contenues dans $infos.

Requête SQL permettant d’ajouter le prix nobel de la paix 2019 dans la base de données.

**INSERT** **INTO** nobels (year, category, name, birthdate, birthplace, county, motivation) **VALUES** (2019, "peace", "Abiy Ahmed", 1976, "Agarao", "Ethiopia", "Abiy was awarded the 2019 Nobel Peace Prize for his work in ending the 20-year post-war territorial stalemate between Ethiopia and Eritrea.")

1. Définir la fonction check\_data(). Cette fonction vérifie si :
   * $\_POST["name"] est défini et n’est pas une chaîne vide ou constituée uniquement d’espaces,
   * $\_POST["category"] est défini et n’est pas une chaîne vide ou constituée uniquement d’espaces,
   * $\_POST["year"] est défini et correspond à un nombre entier strictement positif.

Si l’une de ces conditions n’est pas vérifiée, la fonction retourne false. Autrement, la fonction retourne le tableau $infos créé à partir des informations contenues dans $\_POST. Si certaines valeurs sont manquantes ou incorrectes (chaîne vide ou constituée uniquement d’espaces, valeur non entière pour la date, etc), elle met la valeur null.

1. Définir le script add.php qui vérifie si le formulaire a été soumis. Si c’est le cas, il essaie d’ajouter un prix nobel dans la base de données. Dans tous les cas, il affiche un message pour indiquer si un prix nobel a été ajouté, s’il y a eu une erreur ou si aucun formulaire n’a été soumis.

## **Exercice 10 : Suppression d’un prix nobel**

1. Ajouter la méthode remove\_nobel\_prize() dans la classe Model. Cette méthode prend en paramètre un entier correspondant à un identifiant de prix nobel et supprime le prix nobel associé dans la base de données.

Requête SQL permettant de supprimer le prix nobel d’identifiant 1 :

**DELETE** **FROM** nobels **WHERE** id **=** 1

1. Définir le script *remove.php*. Ce script détermine s’il existe un paramètre id dans l’url correspondant à un entier. Dans ce cas, il détermine si ce paramètre est l’identifiant d’un prix nobel. Il supprime le prix nobel de la base le cas échéant.

Dans tous les cas, le script affiche l’un des trois messages suivants :

* + There is no id in the url.
  + There is no nobel prize with such id.
  + The nobel prize has been removed.

1. Modifier le script *last25.php* de manière à afficher à côté de chaque prix nobel l’image [](https://lipn.univ-paris13.fr/~lacroix/teaching/R301/TP/docs/TP2/remove-icon.png). Pour cela, ajouter une colonne dans la table HTML. Les cases de cette colonne devront appartenir à la classe sansBordure. La balise img devra appartenir à la classe icone.

Faire en sorte que lors du clic sur l’une des images, le prix nobel associé soit supprimé par appel du script *remove.php*.

## **Exercice 11 : Modification d’un prix nobel**

1. Ajouter la méthode updateNobelPrize dans la classe Model. Cette méthode prend en paramètre un tableau contenant les clés 'id', 'year', 'category', 'name', 'birthdate', 'birthplace', 'county' et 'motivation'. La méthode doit mettre à jour toutes les informations du prix nobel d’identifiant id avec les autres valeurs du tableau.

Requête SQL modifiant le prix nobel d’identifiant 1 pour qu’il corresponde au prix nobel de la paix 2019.

**UPDATE** nobels **SET** year **=** 2019, category **=** "peace", name **=** "Abiy Ahmed", birthdate **=** 1976, birthplace **=** "Agarao", county **=** "Ethiopia", motivation **=** "Abiy was awarded the 2019 Nobel Peace Prize for his work in ending the 20-year post-war territorial stalemate between Ethiopia and Eritrea." **WHERE** id **=** 1

1. Définir le script *form\_update.php*. Ce script détermine s’il existe un paramètre dans l’url de nom id et s’il correspond à l’identifiant d’un prix nobel dans la base de données. Si c’est le cas, le script affiche le formulaire de modification des informations du prix nobel. Les informations du prix nobel devront être affichées comme valeur par défaut des différents champs de saisie. Dans le cas contraire, le script affichera un message d’erreur.

**Attention :** l’identifiant sera transmis comme un paramètre caché.

**Remarque :** La méthode sera la méthode post. Le script appelé lors de la soumission sera update.php.

1. Définir le script *update.php* qui modifie les informations du prix nobel. Les modifications ne seront faites que si
   * $\_POST["Id"] est défini et correspond à un nombre entier strictement positif,
   * $\_POST["Name"] est défini et n’est pas une chaîne vide ou constituée uniquement d’espaces,
   * $\_POST["Category"] est défini et n’est pas une chaîne vide ou constituée uniquement d’espaces,
   * $\_POST["Year"] est défini et correspond à un nombre entier strictement positif.

Si certaines valeurs ne sont pas précisées, la valeur par défaut est null. Dans tous les cas, le script affiche un message pour indiquer si la base de données a été mise à jour.

1. Modifier le script last25.php afin que la table HTML contienne une colonne de plus. Les cases de cette colonne devront appartenir à la classe sansBordure. Chaque case contiendra l’image [](https://lipn.univ-paris13.fr/~lacroix/teaching/R301/TP/docs/TP2/edit-icon.png) correspondant à un lien hypertexte sur le script *form\_update.php*. La balise img devra appartenir à la classe icone.

## **Exercice 12 : Pagination de l’affichage de tous les prix nobels**

Le but de cet exercice est d’afficher tous les prix nobels de la base de données. Vu le nombre, cet affichage sera paginé, avec 25 prix nobels affichés sur chaque page (L’affichage des prix nobels sera similaire à l’affichage des résultats de différents moteurs de recherche).

Cet affichage se fera via le script list\_all.php. Ce script affichera une page de prix nobels. Le numéro de la page devra être donné dans l’url par le paramètre page. Si la base de données contient 80 prix nobels, les urls affichant les résultats devront être :

* http://localhost/~num\_etudiant/list\_all.php?page=1
* http://localhost/~num\_etudiant/list\_all.php?page=2
* http://localhost/~num\_etudiant/list\_all.php?page=3
* http://localhost/~num\_etudiant/list\_all.php?page=4

1. Définir la méthode get\_nobel\_prizes\_with\_limit($offset, $limit) retournant un tableau contenant les $limit prix nobels de la base à partir du numéro $offset.

Requête SQL pour afficher du 51ème au 75ème prix nobel (dans l’ordre décroissant des années d’attribution des prix nobels) :

**Select** **\*** **from** nobels **ORDER** **BY** year **DESC** **LIMIT** 25 **OFFSET** 50

**Remarque :** Les prix nobels seront ordonnés selon l’ordre décroissant des dates d’attribution des prix nobels. Ainsi, get\_nobel\_prizes\_with\_limit(0,25) retournera le même tableau que la méthode get\_last().

1. Créer le script list\_all.php. Celui-ci vérifie qu’il existe dans l’url un paramètre de nom page correspondant à un entier strictement positif (Si ce n’est pas le cas, il prend la valeur 1). Il affiche alors la page de prix nobels correspondante ou un message d’erreur si cette page n’existe pas.

**Remarque :** L’affichage des 25 prix nobels de la page doit être similaire à celui du script last25.php. Il est intéressant de créer un script list\_table.php qui, étant donné un tableau PHP de prix nobels, affiche ce tableau sous la forme d’une table HTML. Ce script est alors inclus dans les scripts last25.php et list\_all.php pour afficher les prix nobels.

1. Ajouter dans la vue les liens des différentes pages (comme pour le moteur de recherche google) sous la liste des 25 prix nobels affichés. Si la page affichée est la page de résultats numéro 5, alors la liste doit ressembler à ceci.

Liens

## **Exercice 13 : Recherche dans la base de données**

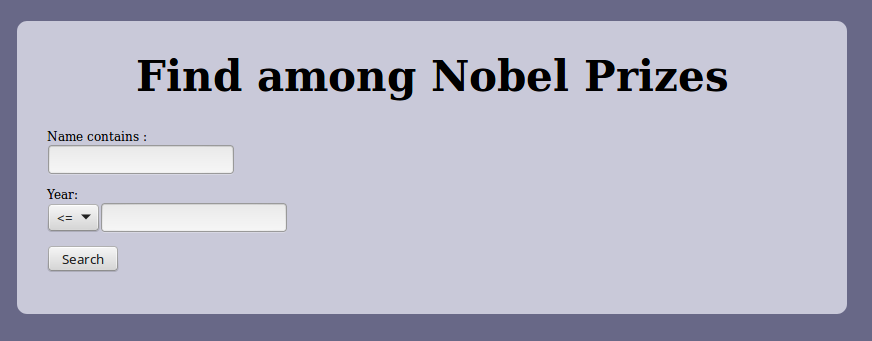
1. Ajouter dans la classe Model la méthode search\_nobel\_prizes($filters, $offest, $limit). Cette méthode permet de rechercher les prix nobels selon certains critères. Pour cela, la méthode prend en paramètre un tableau $filters.
   * Si le tableau $filters contient la clé Name, alors le nom du prix nobel doit contenir la chaîne $filters["Name"] comme sous-chaîne (condition LIKE en SQL).
   * Si le tableau $filters contient les clés Year et SignYear, et si $filters["SignYear"] est égal à <=, >=, ou ==, alors l’année d’attribution du prix nobel doit être inférieure ou égale, supérieure ou égale ou égale à l’année $filters["Year"].

La méthode prend également en paramètre deux entiers $offset et $limit et retourne une partie des résultats en fonction de $offset et $limit (pagination des résultats).

**Remarque :** La méthode search\_nobel\_prizes($filters, $offest, $limit) doit retourner les mêmes résultats que get\_nobel\_prizes\_with\_limit($offest, $limit) si le tableau $filters est vide.

1. Définir le script form\_search.php qui affiche un formulaire de recherche. Celui-ci doit contenir :
   * un champ de type texte pour rechercher selon le nom,
   * un champ de type select pour choisir parmi >=, <= ou ==,
   * un champ de type texte pour entrer un année. Ce champ et le précédent doivent permettre d’effectuer une recherche selon l’année d’attribution du prix nobel.

Le formulaire doit ressembler à :



1. Définir le script search.php. Celui-ci affiche les résultats de la recherche avec pagination. Pour cela, il détermine si le formulaire de recherche a été soumis. Si c’est le cas, il sauvegarde dans la variable $\_SESSION les paramètres de la recherche. Il détermine ensuite s’il existe dans l’url un paramètre de nom page et affiche la page de résultats correspondante. Dans le cas contraire, il affiche la première page de résultats de la recherche.
2. Modifier le code pour permettre de chercher également selon les catégories (0, 1 ou plusieurs catégories peuvent être utilisées dans la recherche).