Marie-andrée healey-côté

Programmation Web avancée

582-31B-MA, gr.24610

TP 1

**Système web PHP orienté objet avec une base de données MySQL**

Travail présenté à

M. Marcos Sanches

Département de Formation continue – Programme NWE.0F

Collège de Maisonneuve

Le 14 octobre 2025

# Présentation du projet

## Le but de ce projet est de concevoir une **application web complète de gestion d'une librairie** (livres, auteurs, éditeurs, catégories), en appliquant les principes de la **programmation orientée objet (POO)** en PHP, avec une base de données **MySQL** et une interface en **HTML/CSS**.

Cette application permet de :

* Gérer une base de données de livres
* Organiser les livres par catégories, auteurs et éditeurs
* Effectuer toutes les opérations CRUD (Créer, Lire, Modifier, Supprimer)

## Fonctionnalités principales

**Gestion des livres**

* Ajout d’un nouveau livre avec :
  + Titre
  + Année de publication
  + Genre
  + Catégorie (liée)
  + Auteur (lié)
  + Éditeur (lié)
* Modification ou suppression d’un livre
* Affichage dynamique de tous les livres en base de données

**Gestion des catégories**

* Sélectionner une catégorie dans liste existante

**Gestion des auteurs**

* Sélectionner un auteur dans une liste existante
* Associer un auteur à un ou plusieurs livres

**Gestion des éditeurs**

* Sélectionner un éditeur dans une liste existante
* Associer un éditeur à un ou plusieurs livres

## Architecture du projet

Site-librairie/

Classe/ ← Dossier contenant les classes PHP

CRUD.php ← Classe de connexion PDO centralisée avec méthodes CRUD

Livre.php ← Classe Livre avec méthodes CRUD

Categorie.php ← Classe Catégorie avec méthodes CRUD

Auteur.php ← Classe Catégorie avec méthodes CRUD

Editeur.php ← Classe Catégorie avec méthodes CRUD

css/

style.css ← Feuille de style du site

sql/

librairie.sql ← Script SQL de création de la base de données

livre-index.php ← Page principale (liste des livres)

livre-create.php ← Formulaire d'ajout de livres

livre-edit.php ← Formulaire de modification/suppression de livres

livre-update.php ← Formulaire qui « MAJ » les informations d’un livre dans la bd en fonction

des données du formulaire fournies et redirige l'utilisateur vers livre-index.php

livre-show.php ← Script qui récupère les détails du livre de la bd et affiche les informations.

livre-store.php ← Script qui traite les données de formulaire soumises via POST pour créer

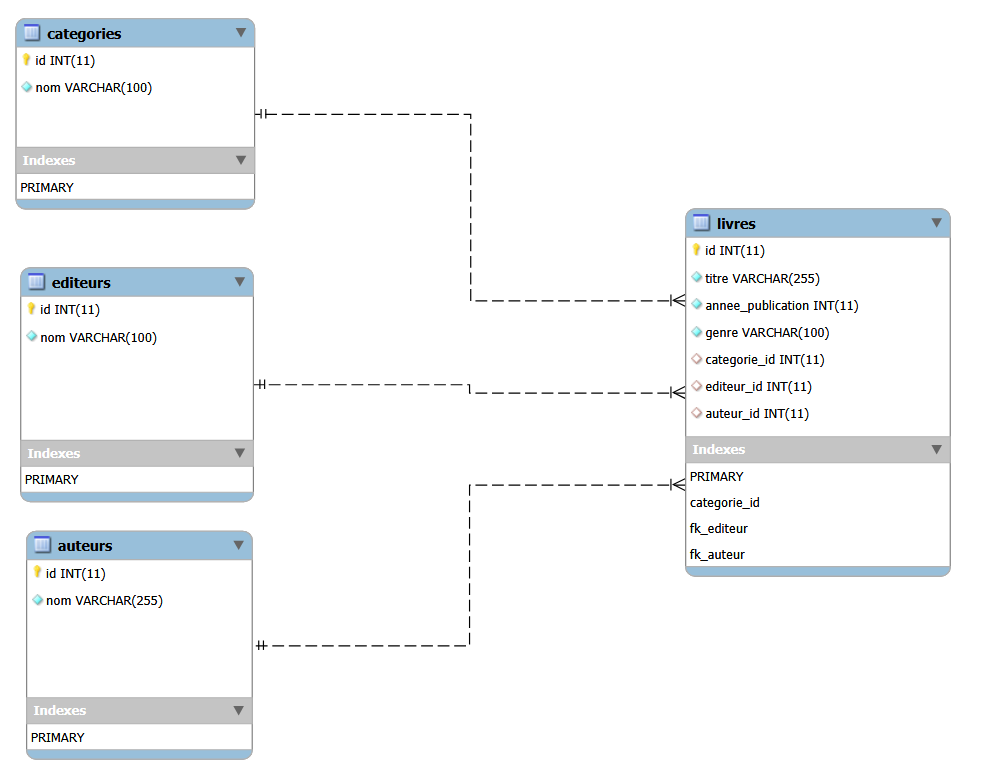
un nouvel enregistrement d’un livre

## Utilisation de la POO

* Chaque entité est représentée par une classe dédiée.
* Connexion PDO via une classe CRUD.
* Utilisation de méthodes CRUD (Create, Read, Update, Delete) dans chaque classe.
* Inclusion d’un destructeur \_\_destruct() pour libérer proprement les connexions à la base de données.

## Interface utilisateur

* Interface HTML simple, claire et responsive
* Navigation entre livres, auteurs, éditeurs et catégories
* Formulaires d’ajout et de modification bien intégrés
* CSS personnalisée dans style.css (tableaux, boutons, etc.)

Structure de la base de données (ERD)

## Sécurité et bonnes pratiques

* Utilisation de **requêtes préparées** (PDO prepare/execute) pour éviter les injections SQL
* Nettoyage de la connexion à la base avec \_\_destruct()
* Séparation logique du code (MVC simplifié)

## Bonnes pratiques POO appliquées

* Encapsulation : les propriétés sont privées
* Constructeur et destructeur (\_\_construct / \_\_destruct)
* Séparation des responsabilités : chaque classe gère un seul type d'entité
* Réutilisabilité : chaque classe peut être réutilisée dans d'autres projets

## Conclusion

Ce projet est un excellent exemple de :

* Structuration d’un projet web en PHP avec POO
* Intégration de base de données relationnelle avec MySQL
* Développement d’une interface utilisateur fonctionnelle
* Mise en œuvre de bonnes pratiques techniques

Il peut servir de **base solide** pour un projet plus avancé (bibliothèque municipale, médiathèque, gestion de stock, etc.).