

1. Introduction

Le but du TP est de réaliser les bases d'un site marchand de vente de livres. On utilisera pour cela la base de données livres (Fig. 2). La première étape (bien modeste) va consister à créer un compteur du nombre de visites et à créer des requêtes SQL permettant de consulter la BDD de livres.

2. Création du compteur (fichier)

On va implémenter le compteur à l'aide d'un fichier texte qui contient le nombre de visites. À chaque chargement de la page, un code PHP charge ce fichier, incrémente le compteur et sauve le résultat dans ce même fichier. Créer une page PHP qui implémente ce mécanisme et affiche le nombre total de visites de la page.

3. Utilisation des compteurs (cookie)

Le problème de ce type de méthode est qu'une personne rechargeant la même page 10 fois va incrémenter artificiellement le compteur. Pour remédier à cela, implémenter un mécanisme qui positionne un cookie d'une durée de vie d'une heure chez l'utilisateur. Le compteur ne doit être incrémenté que si le cookie n'est pas positionné (auquel cas on le positionne). Implémenter ce mécanisme sous forme d'une fonction qui incrémente (éventuellement) la valeur du compteur dans le fichier et renvoie celui-ci.

4. Consultation du catalogue

1. Créer une page d'accueil (Fig. 1) qui comporte une bannière supérieure, un menu à gauche et une partie principale à droite. Pour cela, utiliser `<header>`, `<section>`, `<nav>`, `<div>`, ... ainsi qu'une feuille de styles. Détails :
 - (a) **La bannière supérieure** : codée par `<header>` contenant 3 sections (`<section>`) :
La 1ère section contient le nombre de visites, la section du milieu contient le titre du site et la section de droite contient 2 lignes pour *Bienvenue Nom Prénom* et *Quitter* (on affichera uniquement ces textes pour l'instant). Donner un attribut *id* pour chaque section.
 - (b) **Le menu** : codé par `<nav>` contenant une liste de balises de type texte (`<input>`) :
Chaque input contiendra un attribut événement *onkeyup* permettant d'exécuter une fonction JavaScript (AJAX) lorsqu'une touche du clavier est relâchée. La fonction recherchera des informations dans la BDD livres selon la saisie. Les informations s'afficheront dans la partie principale à droite. On affichera uniquement du texte pour l'instant.

3 visiteurs	Vente de livres	Bienvenue Prénom Nom Quitter
Recherche : <ul style="list-style-type: none"> • Par Auteur : <input style="width: 100px;" type="text" value="ab"/> • Par Titre : <input style="width: 100px;" type="text"/> 	Bienvenue sur le site de la bibliothèque virtuelle	
Auteurs : <ol style="list-style-type: none"> 1. Cabanel 2. Chabrol 3. Haber 4. Stableford 		

Fig. 1 – Un exemple de présentation du site en cours de réalisation

- (c) **La partie principale** : codée par `<section>` composée d'un paragraphe `<p>` pour « *Bienvenu sur le site de la bibliothèque virtuelle.* » et de 2 divisions (`<div>`) afin de pouvoir afficher simultanément 2 listes de données, résultats des recherches faites par l'internaute. Donner un attribut `id` pour chaque division.
2. Implémenter en SQL les recherches suivantes et les tester sous la BDD *livres* (postgresql)
 - a) recherche d'auteurs connaissant une partie du nom,
 - b) recherche d'ouvrages connaissant une partie du titre,
 - c) recherche d'ouvrages connaissant le code d'un auteur,
 - d) recherche d'exemplaires connaissant le code d'un ouvrage (Nom de l'éditeur et Prix).
 3. Une fois familiarisé avec la BDD *livres*, intégrer la 1ère requête SQL dans un programme *requete.php* à tester indépendamment du site web. Utiliser [PDO](#) (PHP Data Objects) pour accéder à la BDD depuis PHP, exécuter la requête (*query()*, *fetch()*, ...) et afficher le résultat sous forme d'un tableau HTML. Un fois familiarisé avec *PDO*, passer au TP2.

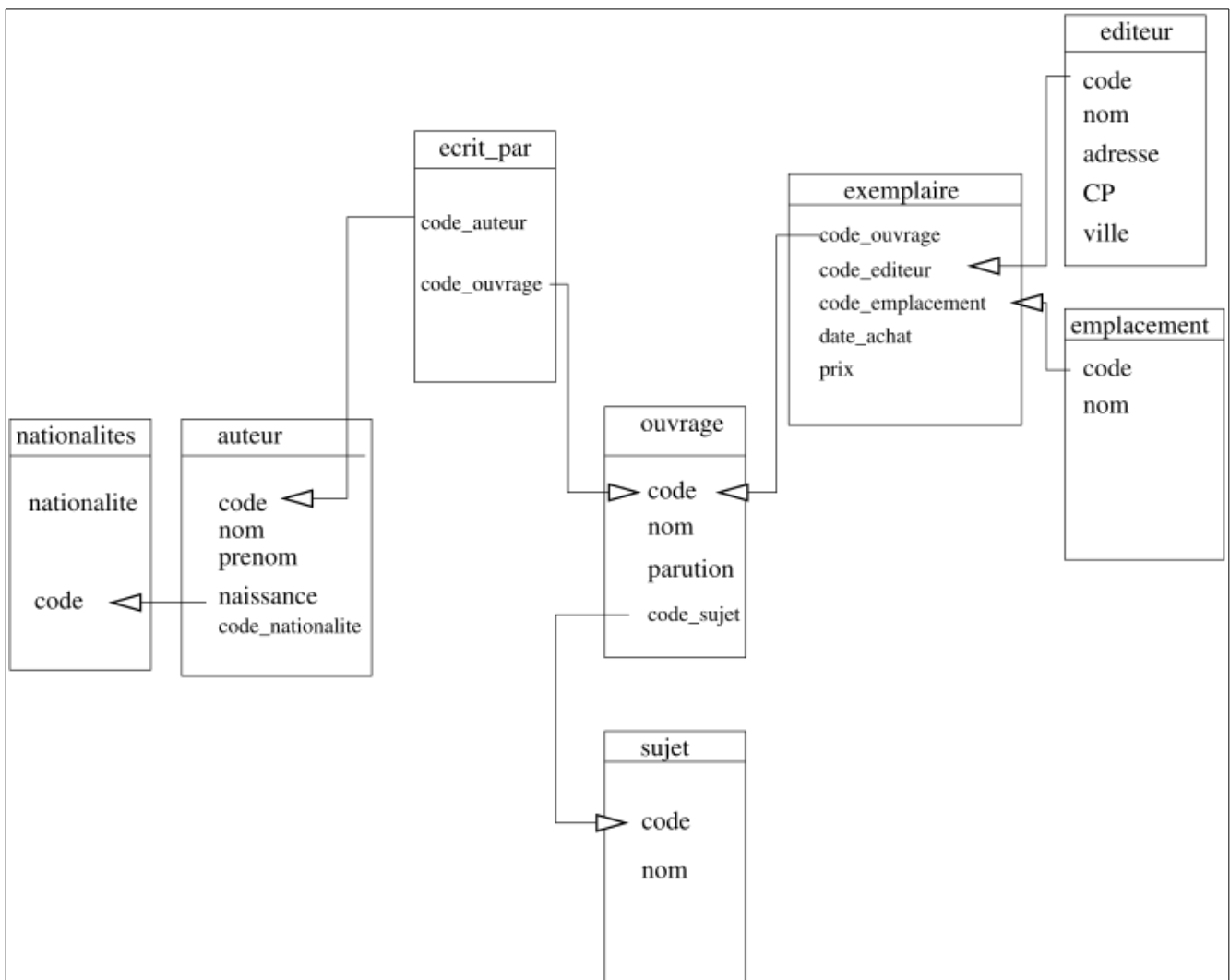


Fig. 2 – Le MCD de la base de données livres. Les clés étrangères et primaires sont représentées par les flèches qui vont de la clé étrangère à la clé primaire.

*NB : Attention, dans la table *exemplaire*, le champs *code* (le code de l'exemplaire qui est la clef primaire) n'apparaît pas mais vous devez considérer qu'il existe.*