# Le contexte

Je veux me construire une application pour gérer ma bibliothèque (Ma femme et moi avons à peu près 3000 bouquins).

Dans un premier temps, ça va rester en local. Ensuite, j’aimerais mettre l’appli sur un serveur pour pouvoir consulter quand je suis dans une librairie par exemple (ça m’est déjà arrivé d’acheter 2 fois le même bouquin !)

# Le problème

Si je constitue la liste par un SELECT sur une table (au sens MySQL), j’obtiens bien la liste des livres.

Si je constitue la liste par un SELECT sur une vie (view au sans MySQL), ça ne plante pas, mais la liste est vide.

Je ne rentre plus dans la description de la structure de l’appli (on est d’accord, un tantinet complexe), il me semble que tu as bien capté le cheminement.

Mais mon but au départ était de comprendre l’architecture logicielle MVC (Modèle Vue Contrôleur). Après une première impression déroutante, je trouve que c’est plutôt bien structuré, et on peut réutiliser ça dans pas mal de situations (mais je suis peut-être un mec un peu bizarre, parce que ça ne me gêne pas quand c’est un peu compliqué !). C’est ce que je fais en réutilisant et adaptant le code construit dans le cours H2PROG.

# Situation 1 avec le SELECT sur la table livre

Le controller et les classes utilisées sont :

LivresController.controller.php

LivreManager.class.php

Livre.class.php

Là, je n’utilise pas le nom de l’auteur, mais juste auteur\_id qui se trouve dans la table livre. Le SELECT dans LivreManager.class.php est :

$req = $this->getBdd()->prepare("SELECT id,

domaine,

auteur\_id,

titre,

serie,

num\_serie,

langue,

editeur,

format\_livre,

an\_edition,

proprietaire,

rangement,

pret,

cover,

a\_lire,

est\_lu,

commentaire

FROM livre

WHERE domaine <> 'BD'

ORDER BY domaine,

serie,

num\_serie");

$req->execute();

Le fichier index.php est :

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Et le résultat est une liste remplie :

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

# Situation 2 avec le SELECT sur la vue v\_livre

La vue v\_livre est créée avec le script suivant :

CREATE

VIEW `v\_livre` AS

SELECT

`l`.`id` AS `id`,

`l`.`domaine` AS `domaine`,

`l`.`auteur\_id` AS `auteur\_id`,

`p`.`nom\_artiste` AS `auteur`,

`l`.`titre` AS `titre`,

`l`.`serie` AS `serie`,

`l`.`num\_serie` AS `num\_serie`,

`l`.`langue` AS `langue`,

`l`.`editeur` AS `editeur`,

`l`.`format\_livre` AS `format\_livre`,

`l`.`an\_edition` AS `an\_edition`,

`l`.`proprietaire` AS `proprietaire`,

`l`.`rangement` AS `rangement`,

`l`.`pret` AS `pret`,

`l`.`cover` AS `cover`,

`l`.`a\_lire` AS `a\_lire`,

`l`.`est\_lu` AS `est\_lu`,

`l`.`commentaire` AS `commentaire`

FROM

(`livre` `l`

LEFT JOIN `personne` `p` ON ((`l`.`auteur\_id` = `p`.`id`))) ;

Le controller et les classes utilisées sont :

ViewlivresController.controller.php

ViewlivreManager.class.php

Viewlivre.class.php

Là, je veux afficher le nom de l’auteur par l’intermédiaire d’un SELECT sur la vue v\_livre dans ViewlivreManager.class.php :

public function chargementViewlivres() {

$req = $this->getBdd()->prepare("SELECT id,

domaine,

auteur\_id,

auteur,

titre,

serie,

num\_serie,

langue,

editeur,

format\_livre,

an\_edition,

proprietaire,

rangement,

pret,

cover,

a\_lire,

est\_lu,

commentaire

FROM v\_livre

WHERE domaine <> 'BD'

ORDER BY domaine,

serie,

num\_serie");

$req->execute();

$mesViewlivres = $req->fetchAll(PDO::FETCH\_ASSOC);

$req->closeCursor();

Dans ce cas, le fichier index.php est :

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Le résultat est une liste VIDE :



# Prolongement

J’ai aussi essayé avec d’autres cas : j’ai aussi une liste de documents, là aussi avec des auteurs, et j’obtiens les mêmes résultats.

J’ai envoyé un mail à H2prog en fin de semaine dernière (en même temps que j’ai demandé de l’aide sur Discord), mais je n’ai pas de réponse. Et il n’y a malheureusement pas de système de questions/réponses dans H2prog.