JPA – Évaluation

L'objectif du projet est de réaliser la partie « Data Access Object » dans un projet de gestion de bibliothèque en ligne de commande.

L'ensemble des classes et interface est déjà codé, mais la configuration et les annotations JPA ne sont pas présente. La base de données est déjà créée.

Il faudra donc :

- Ajouter la configuration JPA
- Mettre en places le mapping entre les classes existantes et la base de données.
- Implémenter la classe `DaoManagerImpl`, en respectant l'interface `DaoManager`.

Schémas et explications

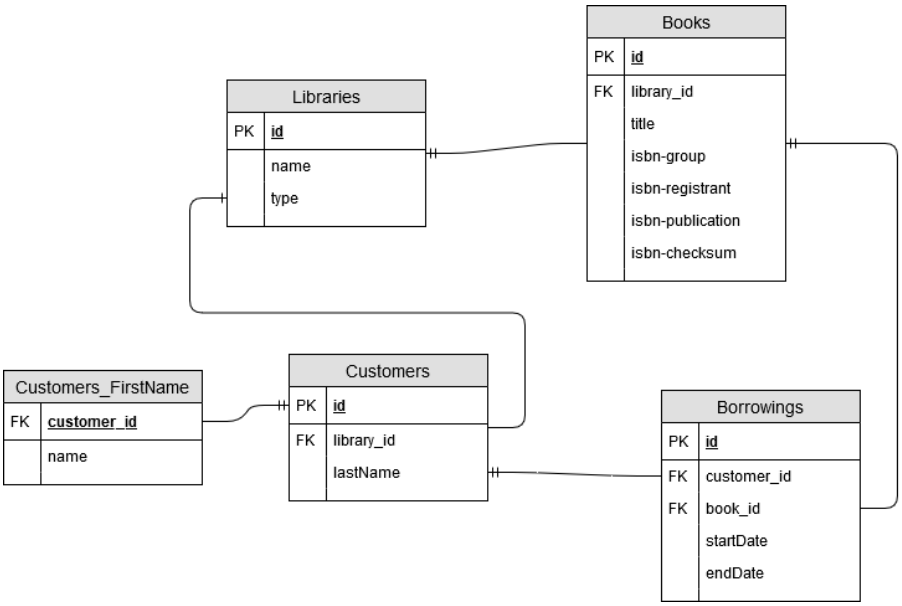

Figure 1 Schéma de la base de données relationnelle

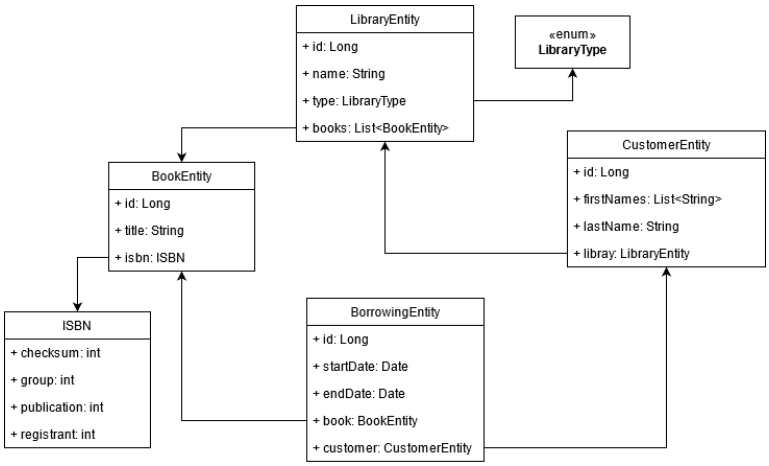

Figure 2 Schéma des classes Java du modèle de données

- Library : bibliothèque
    - Les bibliothèques peuvent soit être de type CENTRAL soit de type SECONDARY
- Book : Livre
    - Correspond à un livre physique présent dans une bibliothèque. Il peut être en plusieurs exemplaires
    - ISBN : identifiant du livre, par exemple : 2-7654-1005-4
- Customer : Client
    - Un client est lié à une unique bibliothèque
    - Un client à un nom de famille (lastName) et une liste de prénom (firstName)
- Borrowing : Emprunt
    - Représente l'emprunt d'un livre par un client. A un moment donné, un libre ne peut être emprunter que par une seule personne à la fois
    - startDate : correspond au début de l'emprunt
    - endDate : correspond à la date de retour
    - Le champs startDate est initialisé à la création de l'emprunt. Le champs endDate est initialement vide. Tant qu'il est vide, on considère que l'emprunt est en cours (active) et le livre ne peut pas être emprunter par un autre client. Cependant le client peut emprunter d'autres livres.

Initialisation

- Exécutez le script qui va créer la base de données avec les 5 tables.
- Importez le projet dans votre IDE
- Modifier dans le fichier `pom.xml`, la valeur de « artifactId » avec votre nom suivi de votre prénom, exemple : `<artifactId>thevenoux_remi</artifactId>`
- Ajoutez le fichier persistence.xml pour se connecter à la base
    - Pensez à vérifier l'url, le mot de passe et le « user »

Afin de ne pas modifier la structure de la base, assurez-vous que la valeur de l'attribut `'hibernate.hbm2ddl.auto'` soit à `'validate'` ou `'none' :`

```
<property name="hibernate.hbm2ddl.auto" value="validate" />
```

Ajout des mappings JPA

- Ajoutez les annotations nécessaires sur les classes suivantes :
    - LibraryEntity
    - BookEntity
    - ISBN
    - CustomerEntity
    - BorrowingEntity

Pensez à mettre à jour le fichier persistence.xml pour y déclarer les entités.

EntityManagerFactory

Dans la classe DaoManager, modifier les méthodes « start » et « stop » pour démarrer et arrêter l'EntityManagerFactory.

Implémentation des services

Implémenté les différentes méthodes pour ajouter les différents services de l'application. Il est préférable de commencer par les services liés à l'entité Library puis Books, puis Customer et enfin Borrowing, ce qui correspond normalement à l'ordre des choix possibles dans l'interface en ligne de commande.

Lazy Loading

Attention, vous serez amené à devoir gérer des problèmes liés au lazy loading sur les collections. Deux solutions sont possibles :

- Option 1 : Modifier le FetchType
- Option 2 : Créer une nouvelle classe, par exemple CustomerView et faire modifier l'interface DaoManager pour que celle-ci retourne des objets CustomerView plutôt que CustomerEntity. Il sera alors nécessaire de modifier les classes Controller et DisplayTool en conséquence.

J'aimerais que vous commentiez dans un fichier joint au projet (par exemple un fichier README à la racine) les choix que vous avez faits pour le LazyLoading ainsi que pour les autres mapping non triviaux.