

Renaud DAVID

Sébastien SEILLER

# BDovore : Rapport Technique

---

Encadrants : Fabrice BOUQUET, Mathias COQBLIN  
Master Informatique 1<sup>ère</sup> année  
Université de Franche-Comté

2010 – 2011

# Sommaire

1. Base de données .....	3
2. Modifications sur les Classes.....	5
3. Synchronisation & Web-service .....	6

## 1. Base de données

La figure 1 en page 4 représente le modèle de la base de données les tables les plus importantes seront décrites ci-dessous.

**Album\_Auteur** : Cette table représente le lien entre un album et un auteur et porte le rôle de l'auteur. Ce rôle est codé sur un entier :

- 0 pour le rôle de dessinateur.
- 1 pour le rôle de coloriste.
- 2 pour le rôle de scénariste.

**Edition\_Bibliotheque** : Cette table permet de gérer l'aspect multiutilisateur, chaque utilisateur ayant sa propre bibliothèque. C'est également grâce à cette table que sont gérées les modifications sur les ouvrages possédés par l'utilisateur ceci se faisant grâce à la table Transaction.

**Transaction** : Cette table permet d'enregistrer les modifications qu'un utilisateur apporte aux ouvrages. Elle permettra lors d'une synchronisation de détecter les modifications concurrentes entre les données locales et distantes.

On se rapportera à un seul enregistrement dans la table par édition :

- Ajout + Suppression → Supprimer l'enregistrement de la table.
- Modification + Suppression → Suppression.
- Modification + Modification → Modification.
- Suppression + Ajout → Modification.
- Tous les autres cas sont impossibles.

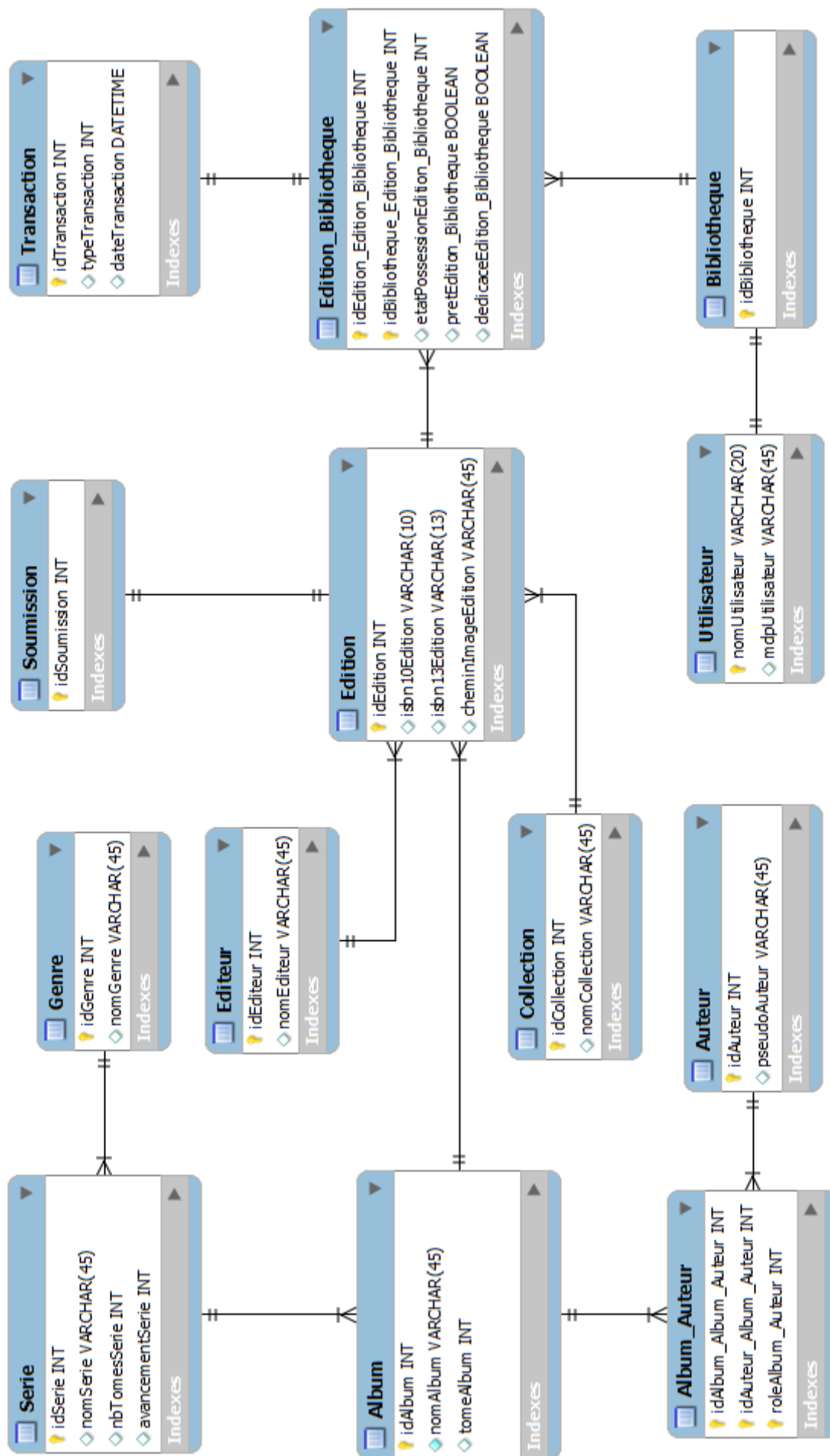


Figure 1 - MCD

## 2. Modifications sur les Classes

Afin de correspondre à la nouvelle base de données la classe `EditionBibliotheque` à été ajoutée, elle permet de gérer toutes les données dépendantes de l'utilisateur comme l'état de possession ou encore le fait que l'on ait prêté ou non un ouvrage.

L'état de possession est un entier pouvant prendre trois valeurs :

- 0 représente le fait que l'on ne possède pas l'ouvrage.
- 1 représente le fait que l'on possède l'ouvrage.
- 2 représente le fait que l'on veut acheter l'ouvrage.

Les données de mises à jour ont été transférées de la classe `Edition` à la classe `EditionBibliotheque`.

La plupart des classes accédant à la base de données à travers des requêtes SQL ont été modifiées les champs inutiles dans les classes existantes ont été supprimés.

Il reste à modifier les classes `Tables.java` et `StatsQuery.java`

La gestion des dessinateurs, coloristes et scénaristes se fait maintenant à travers la classe `Auteur` qui contient un champ rôle correspondant à celui de la base de données.

A savoir qu'il prend les valeurs :

- 0 pour le rôle de dessinateur.
- 1 pour le rôle de coloriste.
- 2 pour le rôle de scénariste.

### 3. Synchronisation & Web-service

Méthodes impliquant le web-service :

1° Récupérer une liste de détails d'un album

- paramètre : id album, liste des détails souhaités
- retour : détails de l'album

2° Récupération de la collection de l'utilisateur

- paramètre : date de dernière synchro
- retour : null si aucune modification serveur, liste de couples <id,etat>

3° Envoi des modifications de la collection de l'utilisateur

- paramètre : liste de couples <id,etat>
- retour : date serveur

4° Récupération de la date/heure serveur :

- paramètre : aucun
- retour : date du serveur (format unix ?)

5° Récupération de la liste de tous les nouveaux albums du serveur

- paramètre : date de dernière synchro
- retour : null si aucune modification serveur, liste des nouveaux id d'album + date serveur

6° Soumission :

- paramètre : toutes les données d'un album
- retour : id de soumission

7° Etat de validation :

- paramètre : id de soumission
- retour : <id,etat\_validation>

8° Mise à jour de la base des auteurs

- paramètre : dernier id
- retour : liste des scénaristes

9° Mise à jour de la base des séries

- paramètre : id
- retour : liste des séries

#### 10° Mise à jour de la base des genres

- paramètre : id
- retour : liste des genres

#### 11° Mise à jour de la base des éditeurs

- paramètre : id
- retour : liste des éditeurs

#### 12° Mise à jour de la base des collections

- paramètre : id
- retour : liste des collections

Couple <id,etat> avec etat :

- 0 : suppression
- 1 : ajout
- 2 : modification

Couple <id,etat\_validation> :

- 0 : en cours
- 1 : validé
- 2 : refusé doublon
- 3 : refusé

Déroulement d'une synchronisation :

- Etape 1 : Données indépendantes de l'utilisateur
  - Méthodes 5 + 8 à 12 (client → serveur)
- Etape 2 Données utilisateur :
  - Méthode 2 (client → serveur)
  - Gestion conflit collection utilisateur (intervention si nécessaire)
  - Méthode 3 (client → serveur)
- Etape 3 Gestion des soumissions :
  - Récupération des validations Méthode 7 (client → serveur)
    - Suppression de l'album temporaire
    - Rien à faire si état inchangé
    - Action utilisateur pour compléter
  - Soumission des nouveaux Méthode 6 (client → serveur)

Lors de modifications concurrentes entre les données locales et distantes, suivant le type d'enregistrement présent dans la table transaction on doit effectuer une action spécifique. Voici un tableau récapitulatif de tous les cas possibles (une croix indique que le cas est impossible) :

Base locale	Base distante	Table transaction	Action à effectuer
Présent	Présent	Rien	Données identiques
		Ajout	Demander version à conserver
		Modification	Demander version à conserver
		Suppression	X
	Absent	Rien	Supprimer local
		Ajout	Ajouter site
		Modification	Demander utilisateur
		Suppression	X
Absent	Présent	Rien	Ajouter local
		Ajout	X
		Modification	X
		Suppression	Supprimer site
	Absent	Rien	Rien à faire
		Ajout	X
		Modification	X
		Suppression	Rien à faire

Figure 2 - Cas d'utilisation de la table Transaction

Pour voir en détail le protocole de communication entre le logiciel et le web-service se référer au fichier WSDL fourni avec les sources du projet. Voici toutefois la description du format CSV utilisé lors des échanges :

caractère séparateur : ','

une entrée par ligne (si plusieurs entrées, par exemple plusieurs albums ou éditions à récupérer)

retour de la méthode getDetails :

nomAlbum, tomeAlbum, nomSerie, nomGenre, auteurs, dessinateurs, coloristes, isbn10Edition, isbn13Edition, cheminImageEdition, etatPossessionEdition\_Bibliotheque, pretEdition\_Bibliotheque, dedicaceEdition\_Bibliotheque

retour de la méthode getBibliotheque :

idEdition\_Edition\_Bibliotheque, etatPossessionEdition\_Bibliotheque, pretEdition\_Bibliotheque, dedicaceEdition\_Bibliotheque



paramètres de la méthode setBibliotheque :

idEdition\_Edition\_Bibliotheque, etatPossessionEdition\_Bibliotheque, pretEdition\_Bibliotheque, dedicaceEdition\_Bibliotheque

paramètre de la méthode getAlbum :

idAlbum

retour de la méthode getAlbum :

idAlbum, nomAlbum, tomeAlbum

paramètre de la méthode getAuteur:

idAuteur

retour de la méthode getAuteur:

idAuteur, pseudoAuteur

paramètre de la méthode getSerie :

idSerie

retour de la méthode getSerie:

idSerie, nomSerie, nbTomesSerie, avancementSerie

paramètre de la méthode getGenre :

idGenre

retour de la méthode getGenre :

idGenre, nomGenre

paramètre de la méthode getEditeur:

idEditeur

retour de la méthode getEditeur :

idEditeur, nomEditeur

paramètre de la méthode getCollection :

idCollection

retour de la méthode getAlbum :

idCollection, nomCollection