Java Académie #1

BOUHERREAU *Nacer*

MARIN *Olivier*

AHMED BACHA *Abdelkrim*

Janvier 2015

Projet java académie n°1

application batch

Descriptif de l'application

Le but de cette application est d'alimenter une base de données avec des informations sur des artistes, leurs albums ainsi que les chansons qu'ils ont joués.

Le programme (batch, java 8) lit dans un répertoire du projet des fichiers .music (valides) et les analyses afin de traités les données.

L'application permet d'ajouter de nouveaux artistes (avec leurs albums, chansons) mais aussi de faire une mise à jour des données existantes (exemple : ajout d'albums/chansons à un artiste, modification des noms des chansons ou des albums déjà existants).

Frameworks utilisés

Afin de simplifier le développement (ne pas réinventer la roue) et de se concentrer sur le coeur de l'application et la logique métier, nous avons utilisé quelques frameworks externes.

- Log4J 2.1 : pour la génération des logs de l'application et l'enregistrement dans un fichier (app.txt)

```
<dependency>
		<groupId>org.apache.logging.log4j</groupId>
		<artifactId>log4j-core</artifactId>
		<version>2.1</version>
</dependency>

<dependency>
		<groupId>org.apache.logging.log4j</groupId>
		<artifactId>log4j-api</artifactId>
		<version>2.1</version>
</dependency>
```

- Hibernate Core: comme ORM, pour la persistance des données.

```
<dependency>
		<groupId>org.hibernate</groupId>
		<artifactId>hibernate-annotations</artifactId>
		<version>3.5.6-Final</version>
</dependency>
```

- Hibernate Annotation: pour le mapping des objets métier (entités)

```
<dependency>
		<groupId>org.hibernate</groupId>
		<artifactId>hibernate-annotations</artifactId>
		<version>3.5.6-Final</version>
</dependency>
```

- MySQL Connector: pour la connexion à la base de données MySQL

```
<dependency>
		<groupId>mysql</groupId>
		<artifactId>mysql-connector-java</artifactId>
		<version>5.1.34</version>
</dependency>
```

- Apache commons-io et commons-csv: pour le traitement sur le dossier (parcours du dossier …) et le parse des fichiers (format CSV).

```
<dependency>
        <groupId>commons-io</groupId>
        <artifactId>commons-io</artifactId>
        <version>2.4</version>
</dependency>


<dependency>
        <groupId>org.apache.commons</groupId>
        <artifactId>commons-csv</artifactId>
        <version>1.0</version>
</dependency>
```

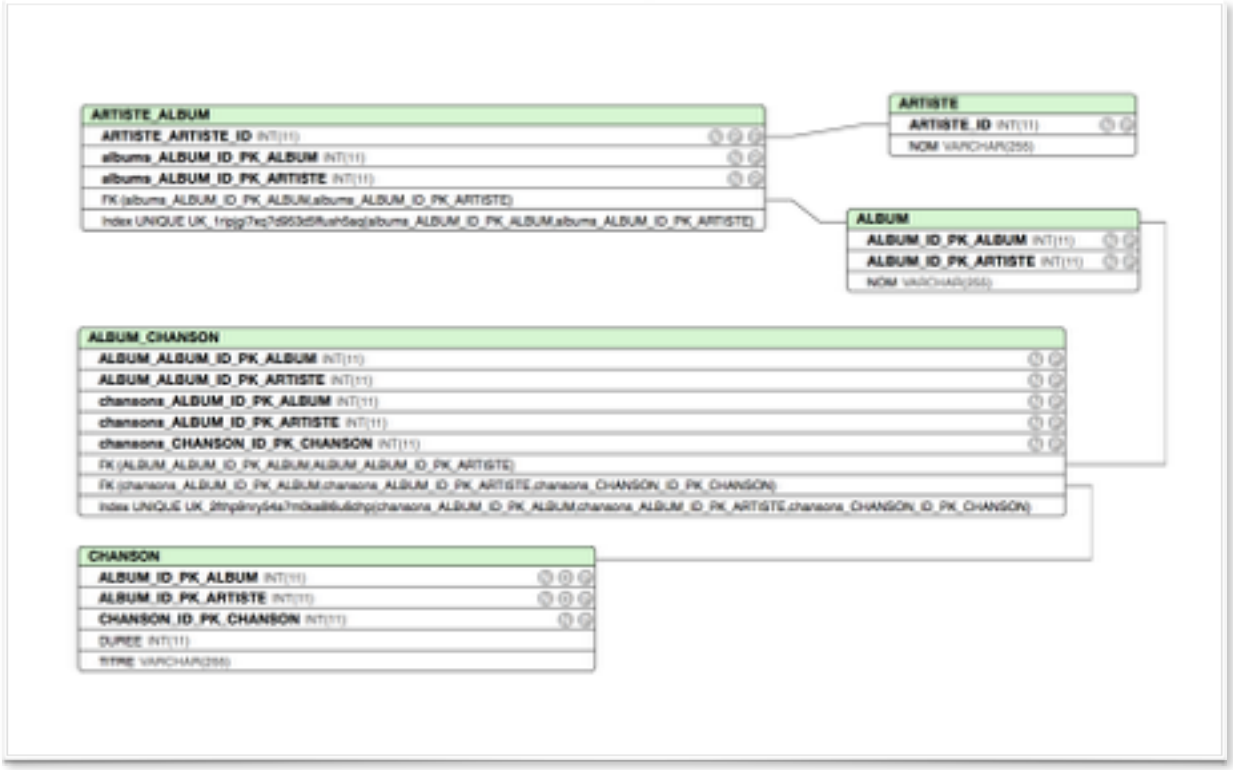
Modèle de données

Pour réaliser cette application, nous avons choisi de représenter chaque objet du domaine (entités) par une table en base de données dans un premier temps, chose évidente et qui simplifie la persistance. Comme base de données, nous avons utilisé « MySQL » et connecteur java/Hibernate adéquat.

On relie les tables entre elles (avec des compositions) en ajoutant une liste d'albums dans la *classe* ***Artiste*** et une liste de chanson dans la *classe* ***Album*** (on parle de classe car on mappe les objets java).

Pour garantir **l'unicité des données par artiste** (code des albums, numéros des chansons), on ajoute des clés étrangères (FOREIGN KEYS) vers les tables d'association (ARTISTE_ALBUM, ALBUM_CHANSON), et des clés primaires composées *(composite-id)* sur les tables *ALBUM* et *CHANSON*.

modèle physique de données



Architecture de l'application et descriptif des packages

Arborescence du projet sous Eclipse

L'application est réalisée en couches (domain, dao, contrôleur) afin de limiter les dépendances des éléments, les doublons de codes et de structurer le code par logique.

Le cycle de vie du projet est géré avec Maven, donc tous les fichiers sources de l'application sont stockés sous ***« src/main/java »*** puis dans des packages différents selon la logique et l'utilité.

Liste des packages et leurs descriptifs :

- **org.jacademie.projet1** : contient le fichier de lancement de l'application.
- **org.jacademie.projet1.constants** : contient une classe qui gère les constantes globales de l'application.
- **org.jacademie.projet1.controller** : contient la logique (traitement, enregistrement) de l'application.
- **org.jacademie.projet1.dao** : contient les classes qui interagissent avec Hibernate pour récupérer, mettre à jour ou supprimer un objet en base de données.
- **org.jacademie.projet1.domain** : contient les classes qui définissent les objets métier (entités) de l'application et leurs clés composées (AlbumId, ChansonId).
- **org.jacademie.projet1.utils** : contient les classes qui fournissent des fonctions utiles à l'application (HibernateUtils : pour la gestion des connexions, sessions avec hibernate, FileUtils : pour le traitement des fichiers et des répertoires scanners).

Sous ***« src/main/resources »***, on retrouve le dossier qui sera scanner (music) ainsi que le répertoire vers lequel les fichiers traités seront déplacés (processed). On trouve aussi deux fichiers importants, ***« log4j2.xml »*** qui définit la configuration du framework de log et ***« hibernate.cfg.xml »*** qui contient les relations de mapping des classes, les informations relatives à la connexion à la base de données et quelques options.

Remarque : il n'y a pas de dossier ***« mapping »*** car l'ensemble du mapping de l'application a été fait avec les annotations.

Configuration avant le lancement

Avant de lancer l'application, quelques étapes sont nécessaires pour finaliser la configuration.

- étape 01 : installer un serveur MySQL si vous ne disposez pas d'un sur votre machine.
- étape 02 : créer une base de données pour l'application (dans phpmyadmin ou sur MySQL)

```
CREATE DATABASE musique DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;
```

Un fichier ***« script_bdd.txt »*** contenant la requête SQL se trouve dans le projet (au même niveau que le pom.xml, dans le dossier projet1).

- étape 03 : dans le répertoire ***« src/main/resources »***, il faut modifier les informations de connexion à la base de données MySQL (utilisateur, mot de passe, url de connexion avec port de MySQL).

```
<property name="hibernate.connection.url">jdbc:mysql://localhost:8889/musique</property>
<property name="hibernate.connection.username">root</property>
<property name="hibernate.connection.password">root</property>
```

Pour Mac, il n'y rien a changé à ce niveau.

- étape 04: après avoir ouvert le projet sur Eclipse, faire un « Maven - Update Project » pour télécharger les dépendances (frameworks) en local. Exécuter ensuite le fichier « Main.java » se trouvant dans le package **org.jacademie.projet1**.

Note : Les fichiers traités par l'application se trouvent sous ***« src/main/resources/music ».***