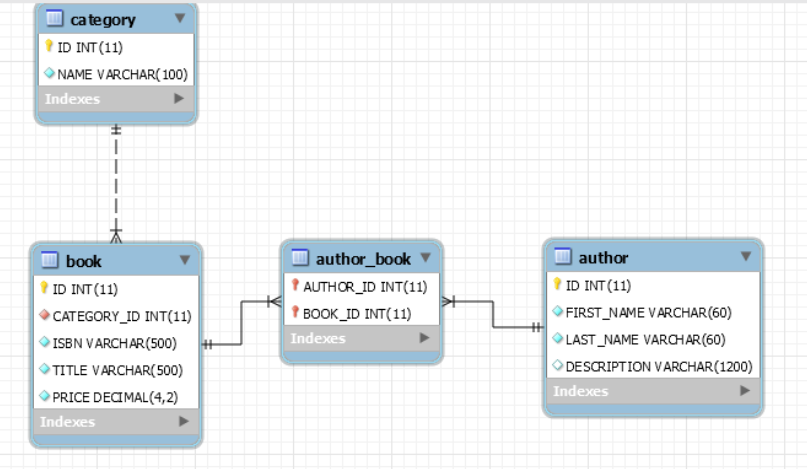
Part1-Assignment-2: Spring Hibernate

1. The program uses a databases named **hibernatedb** I created with MySql server(**User : root and password:** ), similar to the one on the model you provided in the assignment . we have four tables **book, author, author\_book** and **category** . the schema is the same on the assignment . I copied the one on the document

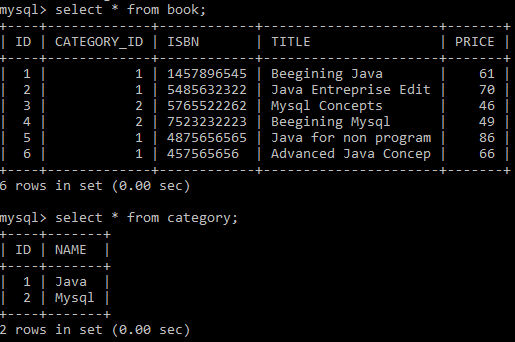


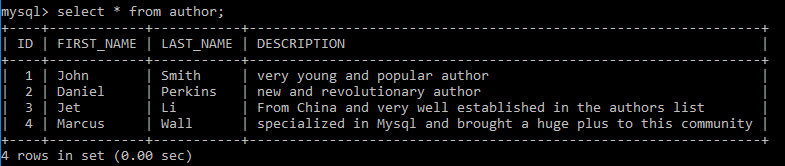
1. I used the Maven command line to create my project:

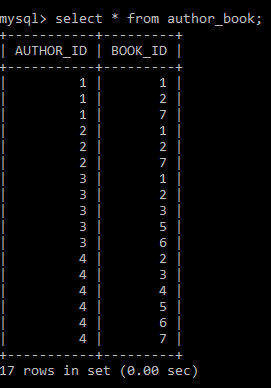
**mvn archetype:generate -DgroupId=edu.herve.spring.hibernate -SpringHibernate**

**-DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false**

1. I edited the **pom.xml** and I added the necessaries dependencies.
2. I created the folder **main\resources** manually .
3. I created the folder **main\resources\META-INF\sql** where I put my **shema.sql** corresponding to my database shema and **test-data.sql** to insert some data; the data are



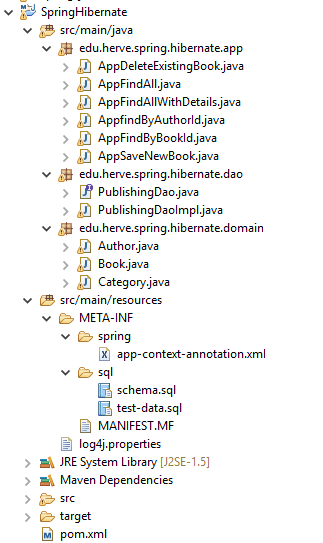




1. I created the folder **main\resources\META-INF\spring** where I put my **app-context-xml.xml**
2. I imported my project into Eclipse IDE and I created the packages:

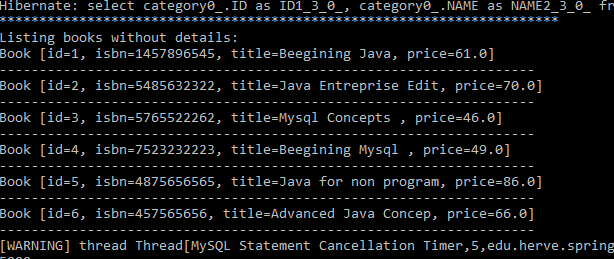
* **edu.herve.spring.jdbc.domain** containing the classes **Book.java, Author.java** and **Category.java** corresponding to the tree tables book, author and category on the database.
* **edu.herve.spring.jdbc.dao** containing the interface **PublishingDao.java where i created the methods corresponding to the functionalities of my program.**  The class **PublishingDao DaoImpl.java** implementing the interface **PublishingDao Dao.java**.
* **edu.herve.spring.jdbc.app** Containing the classes executing each of the functionalities listed in the assignment.
* **AppFindAll.java :** find all books without authors and categories
* **AppFindAllWithDetails.java:** find all books with authors and categories.
* **AppFindByBookId.java:** find a book with authors and category by book's ID
* **AppfindByAuthorId.java:** Find all books for one author id who has more than one book in the database.
* **AppSaveNewBook.java:** add and save in your database a new book.
* **AppDeleteExistingBook.java:** delete the saved book from the database.

This is the image of the structure of my project in Eclipse

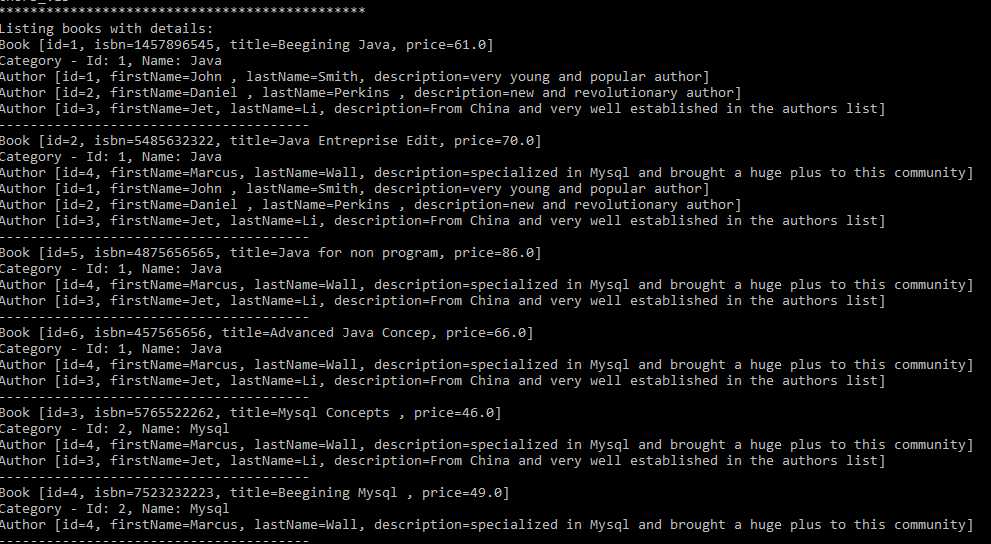
****

1. the building and the execution of my application has the following results.

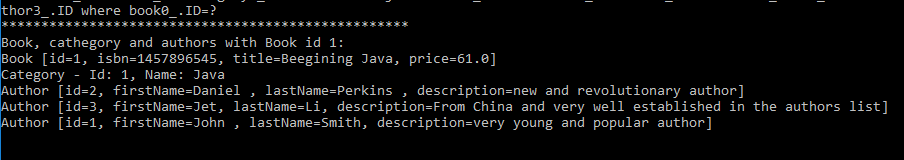
* for the methods **findAll()**



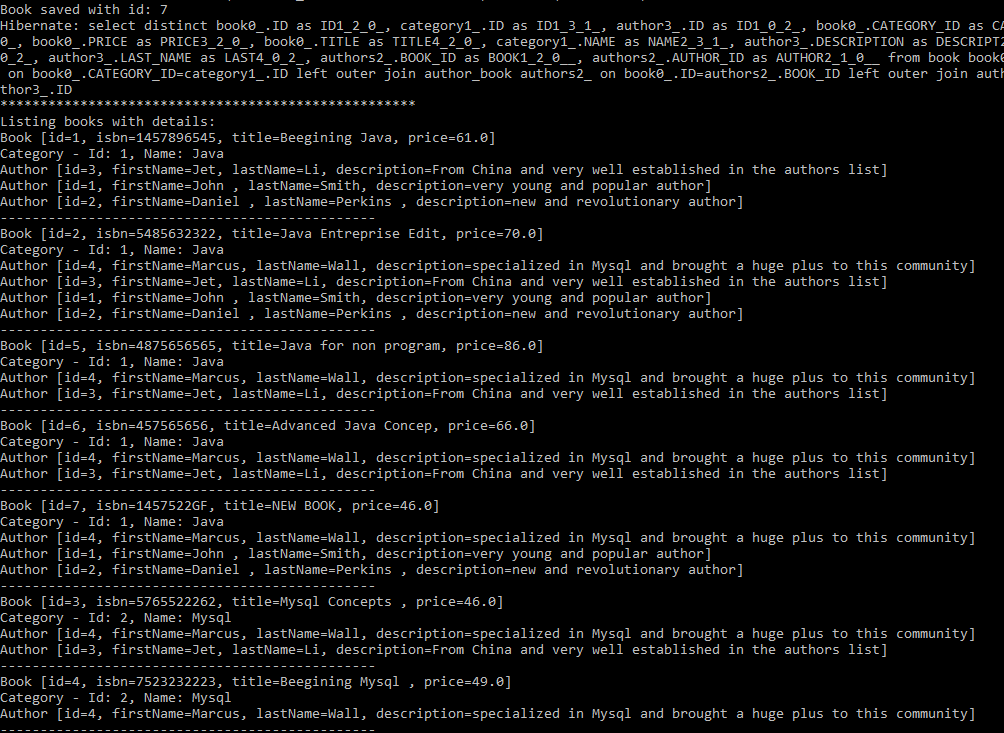
* for the method **findAllWithDetail()**

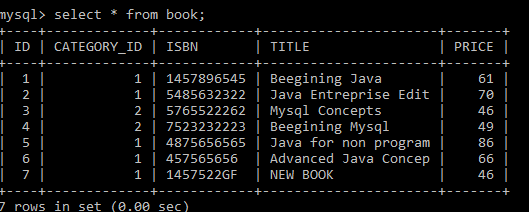


* For the method **findById(int id)**

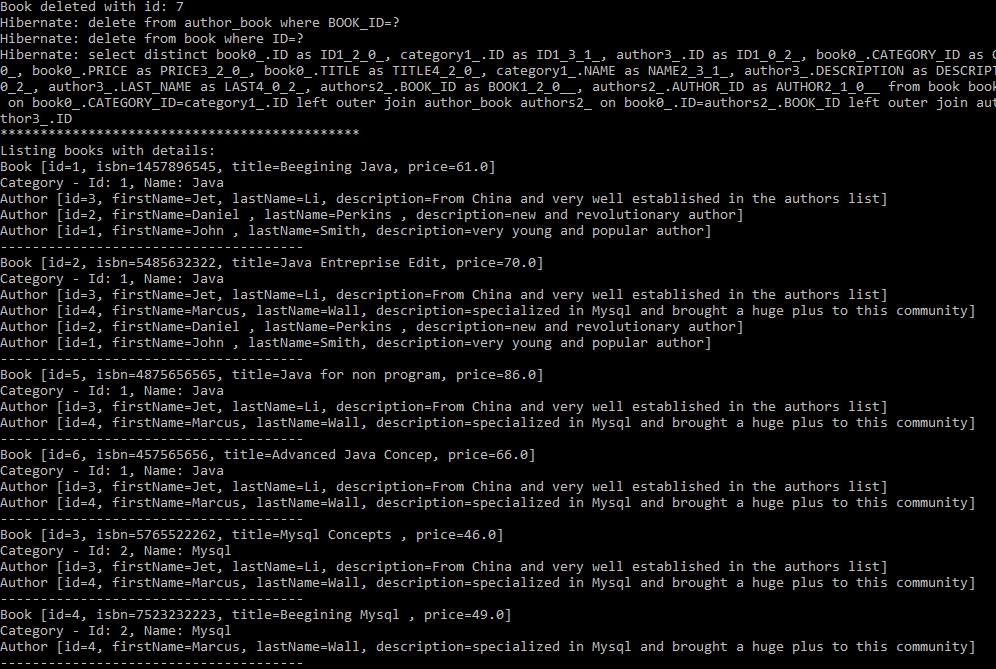


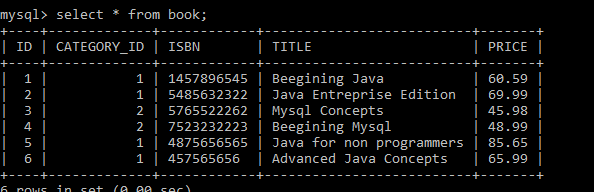
* For the method **save(Book newBook)**





* For the method **delete(Book newBook)**





* For the method **findByAuthorId(int id)**

