# Spring Data Exam

# Library Management System

## Functionality Overview

The application should be able to easily **import** hard-formatted data and **support functionalities** for also **exporting** the imported data. The application is called – **Library Management System**.

Look at the pictures below to see what must happen:

* The home page before importing anything:
* The import JSON page before importing anything:
* Import the books first:
* Import the library members second:
* The import JSON page after importing both files:
* The import XML page before importing the given data:
* Import the borrowing records data:
* The import XML page after importing the data:
* The home page after the data is imported:
* Export the borrowing records before 2021-09-10:

## Model Definition

There are 3 main models that the **Library Management System database** application should contain in its functionality.

We have provided you with an Entity-Relationship (ER) diagram that represents the database schema. The ER diagram displays the tables and their relationships, along with the field types for each table.

Name the entities and their class members **exactly** in the **format stated** above.

All fields are **NOT NULL** unless explicitly stated to be nullable.

Ensure that the **Java** code **validates** data against the specified constraints before persisting it into the database. Handle any validation errors gracefully and provide meaningful feedback to the users.

Design them in the **most appropriate** way, considering the following **data constraints**:

### Book

* id – accepts **integer** values, a **primary identification field, an auto incremented field**.
* title – accepts **char sequence** (between **3** to **40** inclusive). The values are **unique in the database**.
* author - accepts **char sequence** (between **3** to **40** inclusive).
* **description** - a long and detailed description of the book with a character length value higher than or equal to **5**.
* **available** – accepts a true or false, representing the availability status of the book.
* **genre** – String enumeration, one of the following – **CLASSIC\_LITERATURE, SCIENCE\_FICTION, FANTASY**
* **rating** – accepts number values that are positive.

### Library Member

* id – accepts **integer** values, a **primary identification field, an auto incremented field**.
* first name - accepts **char** **sequence** (between **2** to **30** inclusive).
* **last name** -accepts **char** **sequence** (between **2** to **30** inclusive).
* address - accepts **char** **sequence** (between **2** to **40** inclusive). Can be nullable.
* **phone number** - accepts **char** **sequence** (between **2** to **20** inclusive). The values are **unique in the database**.

### Borrowing Record

* id - accepts **integer** values, a **primary identification field, an auto incremented field**.
* borrow date - a date in the "**yyyy-MM-dd**" format.
* return date - a date in the "**yyyy-MM-dd**" format.
* remarks - can be used to store any relevant information and might be helpful for tracking and managing the borrowing records. Accepts char sequence (between 3 to 100 inclusive). Can be nullable.
* Constraint: The borrowing\_records table has a relation with books table.
* Constraint: The borrowing\_records table has a relation with library\_members table.

### Relationships

Your partners gave you a little hint about the more complex relationships in the database so that you can implement it correctly.

One **Borrowing Record** may have only one **Book**, but one **Book** may be in many **Borrowing Records**.

One **Borrowing Record** may have only one **Library Member**, but one **Library Member** can be in many **Borrowing Records**.

## Data Import

Use the provided files to populate the database with data. Import all the information from those files into the database.

**You are not allowed to modify the provided files.**

**ANY INCORRECT** data should be **ignored** and a message:

"**Invalid {book / library member/ borrowing record} should be printed**."

**When the import is finished:**

"**Successfully imported {book / library member/ borrowing record} {author – title/ first name – last name/ title – borrow date}**"

**The Judge** will only accept **file paths** in a specific format. When dealing with file paths for files, please adhere to the following format: "src/main/resources/files/xml/format-example.xml"

### JSON Import

Your new colleagues have prepared some JSON data for you to import.

#### Books (books.json)

##### Constraint

* **If a book with the same title already exists in the DB return "Invalid book".**

Please be aware that due to variations in local settings on different computers, the representation of decimal numbers (Double) may differ. In some regions, the decimal separator is a comma (,), while in others, it is a dot (.).

Judge local settings represent all decimal numbers using a dot (.) as the decimal separator. *Locale.US*

#### Library Members (library-members.json)

##### Constraint

* **If a library member with the same phone number already exists in the DB return "Invalid library member".**

### XML Import

Your new colleagues have prepared some XML data for you to import.

#### Borrowing Records (borrowing-records.xml)

##### Constraint

* **If a book with the given title doesn't exist in the DB return "Invalid borrowing record".**
* **If a library member with the given id doesn't exist in the DB return "Invalid borrowing record".**

## Data Export

Get ready to export the data you have imported in the previous task. Here you will have some complex database querying. Export the data in the formats specified below.

#### Export the Borrowing records before 2021-09-10 from the Database

* Extract from the database the **book title, book author, date borrowed and the full name (first name and last name) of the library member**.
* **Filter only books that are SCIENCE\_FICTION and order them by the borrow date in descending order.**
* Return the information in this format:

**"Book title: {bookTitle}**

**"\*Book author: {bookAuthor}**

**"\*\*Date borrowed: {dateBorrowed}**

**"\*\*\*Borrowed by: {firstName} {lastName}**

**. . ."**