Step 17 – JDBC – BookRegister

Remember The BookRegister in Session 4? In that task, we can register a book, delete a book, and get books by author name, etc. We have used ArrayList<> for local storage. Time for trying something new! We will accomplish a new task this time – interacting with databases from a Java program by using JDBC.

The goal for this step:

- Learn to connect to a database
- Learn to insert objects into tables programmatically
- Learn to delete objects from table programmatically
- Learn to retrieve information from a database and convert it into usable Java objects
- Learn to inner join tables programmatically
- Recap Optional, Equals, Lambda, etc…

Task 0 – Code already provided

Note! You already have Author, Book, and Genre java classes available from Solutions.step17codebasis.src. We also have a class called AuthorRecord, which represents a author record retrieved from **authors** table.

What is the difference between Author and AuthorRecord? AuthorRecord has a tableId (id in authors table) and Author attributes.

Book has title, pages, genre, and author attributes, where author is Author (aggregation).

Author has name and nationality attributes.

Task 1 – Database support

Create a database called **booksDb** either from mysql command window, or workbench, or programmatically.

```
CREATE DATABASE booksDb;
CREATE USER 'user1' IDENTIFIED BY 'pass';
GRANT ALL on booksDb.* TO 'user1';
```

Create JDBCOps class.

1. In the constructor, you can register JDBC driver.
2. Create a method called createTable(), where you will create two tables: **books** and **authors**. **books** and **authors** are connected by **authorId**. **books** has **id** as primary key, and **authorId** as foreign key.

```
String createTable = "CREATE TABLE IF NOT EXISTS books( "
        + "id INT NOT NULL AUTO_INCREMENT, "
        + "title VARCHAR(45), "
        + "authorId INT, "
        + "pages  INT(11), "
        + "genre  VARCHAR(45), "
```

```
        + "primary key (id), "
        + "foreign key (authorId) references authors(id)) ";
```

```
createTable = "CREATE TABLE IF NOT EXISTS authors( "
        + "id INT NOT NULL AUTO_INCREMENT, "
        + "name VARCHAR(45), "
        + "nationality VARCHAR(45), "
        + "primary key (id)) ";
```

Task 2 - Query in JDBCOps

Let's do some queries. You will be able to get all books, get books by author, add a book, and delete a book.

1. Create a method called ArrayList<Book> getBooks(). You will get all registered books from **booksDb.books** and populate the retrieved information to usable java objects. Note that in **booksDb.books** table you have only authorId, how can we get author information from **auhors** table??
   HINT: You can use inner join to join **books** and **authors** tables!

2. Create a method called ArrayList<Book> getBooksbyAuthor().
   HINT: you can use inner join and having!

3. Create a method called ArrayList<AuthorRecord> getAuthors(), which returns all the records from **authors** table and populate the retrieved information into ArrayList<AuthorRecord> java object!

4. Now we want to add a book. Create a method called addBook(Book book). It will insert a new book object into **books** table.
   However, if the author of the book is not registered, we also want to update **authors** table!
   HINT: you can call getBooksbyAuthor() method first and check if the author of the new added book exists in **authors** table. If not, insert a new record into **authors** table. If the author exists, you only need to add **books** table.
   HARD question: how do we find out authorId in books table for the new added book?

   (THIS IS THE HARDEST QUESTION SO FAR)

5. We will want to be able to delete a book too. Create a deleteBook(Book book) method which allows you to delete a book record from books table by using book title.

Task 3 – BookRegister class

The BookRegister class will have following constructor:

```
private JDBCOps jdbcOps;

public BookRegister(JDBCOps jdbcOps) {
    this.jdbcOps = jdbcOps;
}
```

And it will support following methods:

public void addBook(Book book) – you add a book to database through JDBCOps methods

public void deleteBook(Book book) – you delete a book from database through JDBCOps methods

public void GetRegisteredBooksByAuthor(String author) – you get filtered books from database

public void GetRegisteredBooksByAuthor2(String author) - You get all books from database, and do filtering by author name?

Task 4 – Main

So you can manipulate your booksDb! Feel free to play with the functions you just created!