Lab Exercises 9

Dr. Sarvar Abdullaev
s.abdullaev@inha.uz

December 17, 2020

You must attempt all exercises given in this lab. After completing it, you must push it to your corresponding GitHub Classroom repository. Note, you do not have to upload compiled Java bytecode or screenshot of your program's output.

1 Query Application for the `books` Database

Using the techniques shown in this lecture, define a complete query application for the `books` database. You can use Apache Derby's command-line IJ tool for accessing `books` database.

Below is the list of SQL queries you should try with IJ:

1. Select all authors from the Authors table.
2. Select a specific author and list all books for that author. Include each book's title, year and ISBN. Order the information alphabetically by the author's last name then by first name.
3. Select a specific title and list all authors for that title. Order the authors alphabetically by last name then by first name.
4. Provide any other queries you feel are appropriate.

After ensuring that your SQL queries return results, you should implement a simple Java console application which displays the results of above queries.

2 Address Book Modifications: Update and Delete

Reproduce an "Address Book" application as shown in your lecture example. You can find the source code here: `https://github.com/iuthub/java_2020_lecture11.git`.

You must add following modifications to the existing code:

1. Modify the program to provide a `JButton` that allows the user to call a method named `updatePerson` in `PersonQueries` class to update the current entry in the `AddressBook` database.
2. Modify the program to provide a `JButton` that allows the user to call a method named `deletePerson` in `PersonQueries` class to delete the current entry in the `AddressBook` database.

3 Address Book in JavaFX

You need to create a simple Address Book application as shown below using JavaFX.

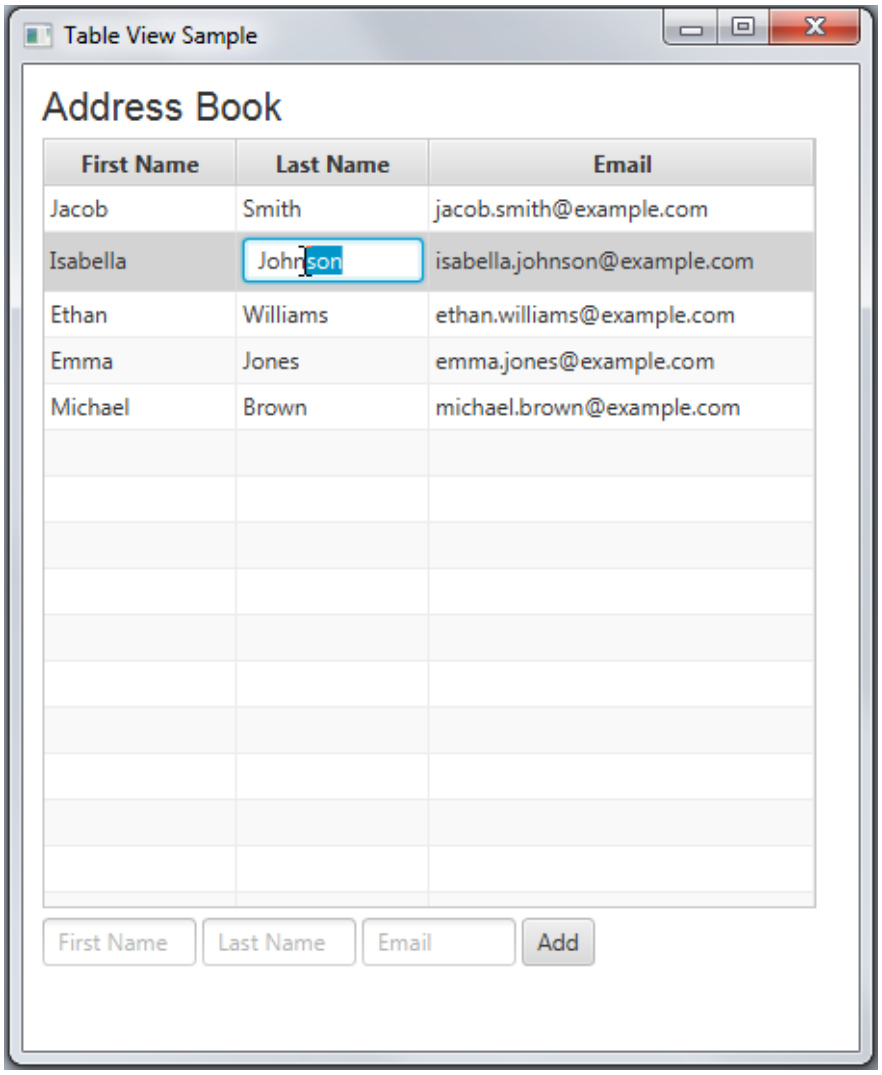

Figure 1: Address Book in JavaFX

You can split the project into 3 parts, as listed below.

3.1 Create database

Create 'addressbook2.sql' file where you should create `Addresses` table similar to the one used in task 2. Below is the code for that:

```
DROP TABLE Addresses;
CREATE TABLE Addresses
(
    AddressID INT NOT NULL GENERATED ALWAYS AS IDENTITY,
    FirstName VARCHAR (15) NOT NULL,
    LastName VARCHAR (30) NOT NULL,
    Email VARCHAR (30) NOT NULL,
    PhoneNumber VARCHAR (15) NOT NULL
);
```

Connect to embedded Apache Derby using IJ tool, create addressbook2 database and run above script there.

3.2 Design main window

Design the main window of your Address Book as shown in Figure 1. You must use JavaFX components such as `TableView`, `TableColumn`, and `TableCell`. You can populate a table by implementing the data model and by applying a cell factory. The table classes provide built-in capabilities to sort data in columns and to resize columns when necessary.

Create a simple form for adding new addressed below the `TableView`.

3.3 Implement business logic

In this section, you should implement all database related functionality including retrieving records from database, inserting and deleting records from it.

Below are the steps you should carry out:

1. Implement a plain class `Address` with all the necessary fields as in your database `Addresses` table, and corresponding getters and setters.
2. Create a separate `AddressRepository` class for retrieving and updating your *Address* objects reflected as rows in database. Use a concrete instance of `ObservableList<Address>` for storing the collection of `Address` object.
3. Populate `TableView` with retrieved data
4. Implement `Add` and `Delete` buttons. `Add` button should insert a new address into the database and display a new row in `TableView`. `Delete` button should remove the selected address both from `TableView` and database.

3.4 Ship your project

Create an artefact of your project using IntelliJ and copy your current working database into the folder with your artefact. Write a simple .bat file for executing your .jar artefact on a Windows machine.