The purpose of this assignment is to learn interacting with databases using JDBC.

Exercise 1 – Books

1. Download H2 database **version 1.4.200**

- Go to https://www.h2database.com/html/download-archive.html and download the "All Platforms" zip for version **1.4.200** (direct link: https://h2database.com/h2-2019-10-14.zip).
- Unzip the archive and copy the jar file (e.g., *h2-1.4.200.jar*) located in the bin folder to a convenient folder.

2. Start the H2 database service and create the database from the H2 web console:

- Open a terminal, change to the directory containing the jar file and start the database service by running the following command (**DON'T CLOSE the terminal to keep the process running!**)

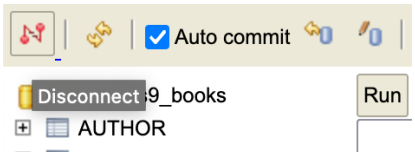
```
java -jar h2-1.4.200.jar
```

- After the browser opens showing the login page (by default it will be hosted on http://localhost:8082):
    - Enter the **JDBC URL** to create a "books" database on your local filesystem

      E.g. `jdbc:h2:/Users/avim/dbs/books` will create the "books" DB in the `/Users/avim/dbs/` folder.
    - Login to the web console with username `sa` (leave password field blank)
- Create the *Author* and *Book* tables by running the following SQL commands in the H2 web console:

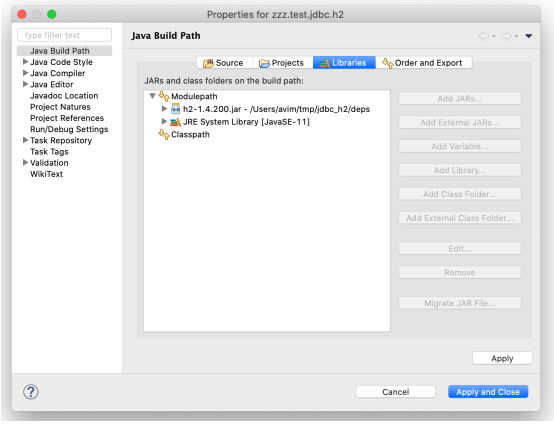
```
create table author (
  first_name varchar2(20) not null,
  last_name varchar2(20) not null,
  birth_year integer,
  primary key (first_name, last_name)
);

create table book (
  title varchar2(50),
  pub_year integer,
  publisher varchar2(20),
  author_first_name varchar2(20),
  author_last_name varchar2(20),
  primary key (title),
  foreign key(author_first_name, author_last_name) references author(first_name, last_name)
);
```

- **Disconnect / Logout** from the H2 web console by clicking the "disconnect" icon on the top left corner, but **keep the terminal / process open**!



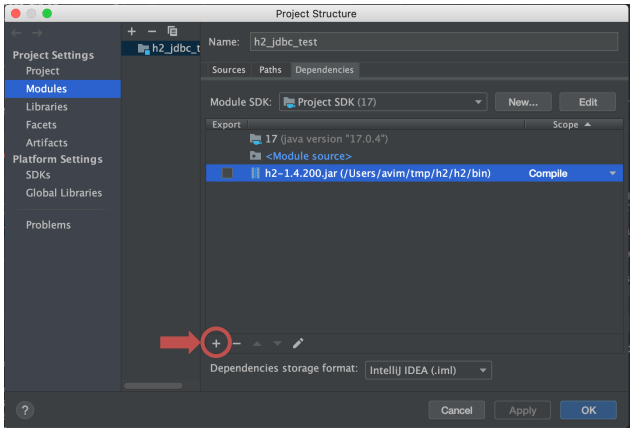
3. Setup project dependencies in IDE:

In your preferred IDE add the jar dependency to your project:



Eclipse

Project properties → Java Build Path → Libraries tab → Add external JARs



IntelliJ

Project settings → Modules → Dependencies tab → select the (+) icon and add the jar file

4. Programming part

Open the `es1/Books.java` file, which contains a skeleton program for this exercise with some initial empty methods that you are required to implement with the following logic:

- *insertAuthor* : adds a new author to the database.
- *insertBook* : adds a new book to the database. Notice: the author's firstName and lastName must match an existing author stored in the database, as they are foreign keys!
- *printAuthors* : queries the database for all authors and prints to console the author's details.
- *printBooks* : queries the database for all books and prints to console title, year and publisher.
- *printBooksFull* : queries the database for all books and prints to console title, year, publisher, author's first and last name (**HINT**: use inner join)
- *printTable* : queries the database for a given *table* parameter and prints to console each table's row data. Use the *ResultSetMetaData* class to gather the number of columns and to handle various datatypes each column has.

For all the beforementioned methods implement the following versions:

a) In a first version implement all mentioned methods using **statements**.
b) In a second version, copy all methods and reimplement them using **prepared statements** and the **try with resources**.
c) In a third and final version, introduce 2 additional classes (Author and Book) that map both tables. Implement 2 new *insertAuthor* and *insertBook* methods that receive Author / Book objects as parameters instead of Strings and Integers. Finally implement *getAllAuthors* and *getAllBooks* methods which return collections of Authors / Books objects.

Exercise 2 – Address book

Write a java program that implements a basic address book that stores its contents to a H2 database using JDBC. The address book should allow to store contacts with at least the following information:

- First name
- Last name
- Phone number

- Mobile phone number
- Email address

Add all the required methods to create, update, delete and find contacts. Test your implementation by adding some entries, modifying at least one of them and finally deleting one. The entries to be modified and deleted should be taken by using the find method.

IMPORTANT:

- add a comment in your code with the SQL command used to create your database!
- Make sure to avoid the following keywords for table names and field names: https://www.h2database.com/html/advanced.html#keywords

Exercise 3 – Player catalog (Optional)

Write a java program that implements a catalog to store players and some of their properties such as

- First name
- Last name
- Year of birth
- Current team name
- Current value

Use a H2 database with JDBC to store the data and implement the required methods to create, update, delete and find a player. Test your implementation by adding some players, modifying at least one of them (e.g. it's team or value) and finally deleting one.

IMPORTANT: add a comment in your code with the SQL command used to create your database!