**Project**

Maven was used with Spring 3.1.2, and Java 17. Usually I would do testing with Spock/groovy, however I did not have a great time trying to get that to work with H2/JPA so went back to just using JUnit.

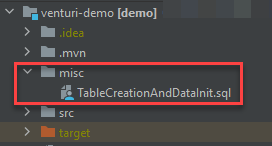
**Database**

This was my first time using H2 and in-memory DB with JPA so it took a while to get myself up and running. Most of my experience has been with DAOs and hand writing all my queries.

Probably could have had the availability in a separate table, but with there being only one copy of each book I decided to make it simpler for myself and just have available bool and borrower\_id be on the book table.

Comments are left throughout the code with what the SQL queries would have been if not using JPA.

You can find table creation queries and the Inserts I used for data initialisation within the project, under the misc folder in the project directory



**book**

* Book\_id Identity PK
* Title
* available
* borrower\_id
  + FK student student\_id

**book\_author**

* book\_id PK
  + FK book book\_id
* author\_id
  + FK author author\_id PK

**author**

* Author\_id identity PK
* name

**student**

* student\_id identity PK
* name

**ER Diagram**

Today was my first time screwing around with H2, so it took a bit longer than I expected. Was going to use the Intellij DB Diagram tool to create the ER Diagram, but the Intellij I have installed at home is Community Edition, which doesn’t have the DB tools apparently. So instead of a nice IntelliJ diagram, you get a handcrafted diagram from yours truly. Last time I had to make one of these was in Uni, so apologies if i got the ‘one to many’ back to front.

