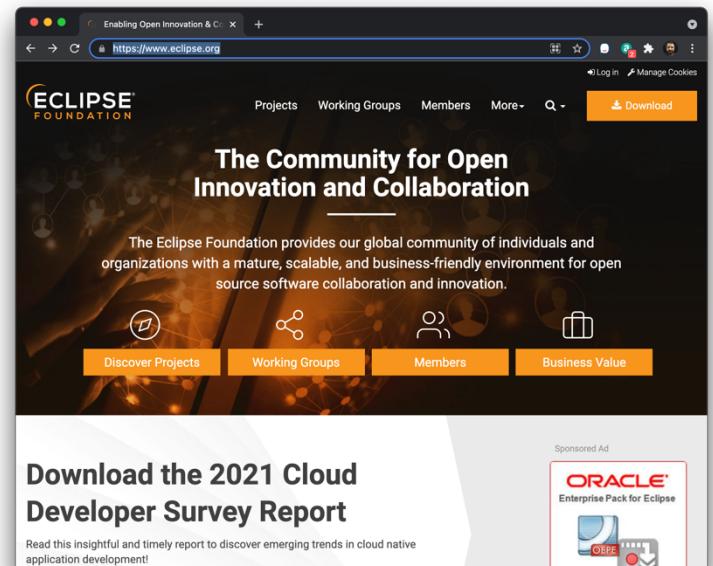


20 February 2022  
Christopher Sean Morrison  
Term Project Part 1

# TERM PROJECT PART 1

In this overview, we show the steps taken to implement a Book Publishing Enterprise Application, including setup and development steps for Assignment's 1.1 (Spring DI of Java Collections), 1.2 (Spring AOP with Before, After Returning, Throws Advices), 1.3 (Spring JDBC), and 1.4 (Spring Hibernate).

To get set up, I first installed all the specified tools including the latest Java (OpenJDK), Eclipse, Hibernate, Maven, Tomcat, and MySQL. I use Mac OS X for all development, Emacs for coding, and Eclipse IDE for interactive debugging. Homebrew was used to install most, except for Eclipse and Java which were installed manually. Maven was used for all Java dependencies.



A screenshot of the eclipseinstaller by Oomph interface. It shows a product card for "Eclipse IDE for Java Developers" with a "details" link. Below the card, there are sections for "Java 11+ VM" (set to JRE 17.0.0) and "Installation Folder" (set to /Users/morrison/eclipse/java-2021-09). At the bottom is a large orange "INSTALL" button.

A screenshot of the Eclipse IDE workspace titled "workspace - SpringAutoWiringAnnotationExample/src/main/java/com/websystique/spring/domain/ApplicationUser.java - Eclipse IDE". The code editor shows Java code for an ApplicationUser class. The Variables view on the right shows the state of the "name" field, which is currently "defaultName". Other fields like "coder" and "hash" are also listed. The workspace sidebar shows various projects and files.

## 1. ASSIGNMENT 1

For the first assignment, I was fully successful implementing the three categories and populating them. I decided to use XML configuration, and obtained the desired output shown in the assignment instructions. I created six books with two in each of the three categories.

For all assignments, I created a wrapper shell script to run Maven and facilitate development iterations.

```
(base) morrison@agua assignment1 % find .
.
./pom.xml
./run.sh
./src
./src/main
./src/main/resources
./src/main/resources/BookBeans.xml
./src/main/resources/log4j.properties
./src/main/java
./src/main/java/edu
./src/main/java/edu/morrison
./src/main/java/edu/morrison/spring
./src/main/java/edu/morrison/spring/app
./src/main/java/edu/morrison/spring/app/App.java
./src/main/java/edu/morrison/spring/app/AppConfig.java
./src/main/java/edu/morrison/spring/beans
./src/main/java/edu/morrison/spring/beans/Category.java
./src/main/java/edu/morrison/spring/beans/Book.java
(base) morrison@agua assignment1 %
```

```
#!/bin/sh
mvn clean install
mvn exec:java -Dexec.mainClass="edu.morrison.spring.app.App"
run.sh (END)
```

```
assignment1 — 103x39
/Users/morrison/git/rseg105/part1/assignment1

Users/morrison/.m2/repository/edu/morrison/spring/part1.assignment1/1.0.0/part1.assignment1-1.0.0.jar
[INFO] Installing /Users/morrison/git/rseg105/part1/assignment1/pom.xml to /Users/morrison/.m2/repository/edu/morrison/spring/part1.assignment1/1.0.0/part1.assignment1-1.0.0.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.755 s
[INFO] Finished at: 2022-02-20T06:49:26-05:00
[INFO] -----
[INFO] Scanning for projects...
[INFO] -----
[INFO] < edu.morrison.spring:part1.assignment1 >
[INFO] Building part1-assignment1 1.0.0
[INFO] [ jar ] -----
[INFO] -----
[INFO] --- exec-maven-plugin:3.0.0:java (default-cli) @ part1.assignment1 ---
2022-02-20 06:49:28,549 INFO [edu.morrison.spring.app.App] - <
=====Books Set Output Start =====
Category -- Id: 1, Name: Novels, Books List: [Book - Id: 1, ISBN: 978-0075536321, Title: Anna Karenina, Price: 17.57, Book - Id: 1, ISBN: 978-0060194994, Title: To Kill A Mockingbird, Price: 25.95]
=====Books Set Output End =====
=====Books List Output Start =====
Category -- Id: 2, Name: Java, Books List: [Book - Id: 2, ISBN: 978-0137150021, Title: Effective Java, Price: 43.86, Book - Id: 2, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33]
=====Books List Output End =====
=====Books Map Output Start =====
Category -- Id: 3, Name: Poetry, Books List: [key1=Book - Id: 3, ISBN: 978-0571096343, Title: The Waste Land by T.S. Eliot, Price: 200.0, key2=Book - Id: 3, ISBN: 978-0375505966, Title: Still I Rise by Maya Angelou, Price: 30.29]
=====Books Map Output End =====
>
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.898 s
[INFO] Finished at: 2022-02-20T06:49:28-05:00
[INFO] -----
(base) morrison@agua assignment1 %
```

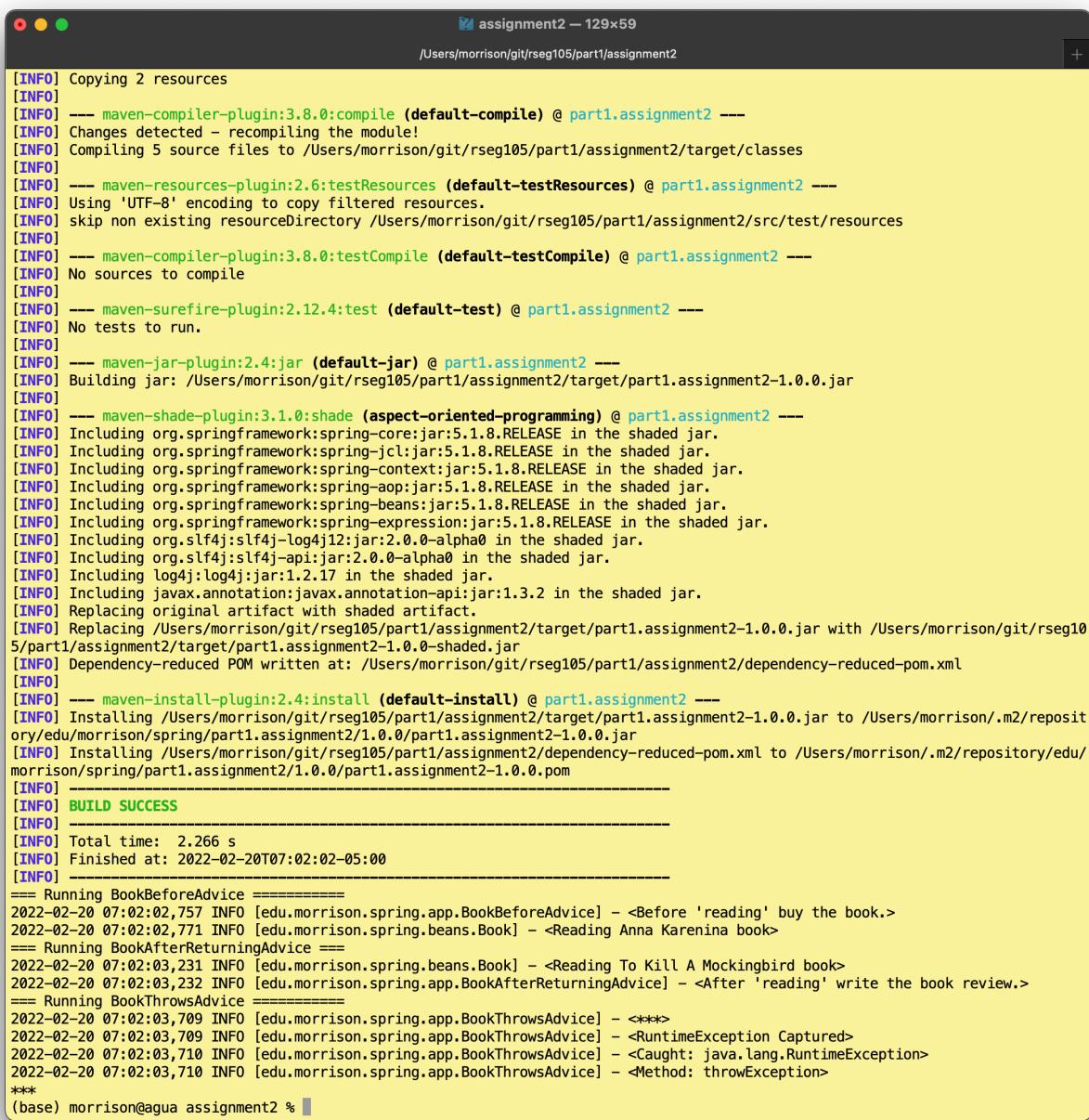
## 2. ASSIGNMENT 2

For the second Spring AOP assignment, I was also fully successful in implementing the solution as specified in the assignment. I created three main classes, and invoke them directly, obtaining the desired result. I again used XML configuration.

Of ancillary note, this was by far the easiest of the 4 assignments and the only one that took me less time than what was listed as the estimated hours on Moodle.

```
(base) morrison@agua assignment2 % cat run.sh
#!/bin/sh

mvn clean install
echo "==== Running BookBeforeAdvice ====="
java -cp ./target/part1.assignment2-1.0.0.jar edu.morrison.spring.app.BookBeforeAdvice
echo "==== Running BookAfterReturningAdvice ====="
java -cp ./target/part1.assignment2-1.0.0.jar edu.morrison.spring.app.BookAfterReturningAdvice
echo "==== Running BookThrowsAdvice ====="
java -cp ./target/part1.assignment2-1.0.0.jar edu.morrison.spring.app.BookThrowsAdvice
(base) morrison@agua assignment2 %
```



The screenshot shows a terminal window titled 'assignment2 – 129x59' with the path '/Users/morrison/git/rseg105/part1/assignment2'. The window contains the following Maven build log:

```
[INFO] Copying 2 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ part1.assignment2 ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 5 source files to /Users/morrison/git/rseg105/part1/assignment2/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ part1.assignment2 ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/morrison/git/rseg105/part1/assignment2/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ part1.assignment2 ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ part1.assignment2 ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ part1.assignment2 ---
[INFO] Building jar: /Users/morrison/git/rseg105/part1/assignment2/target/part1.assignment2-1.0.0.jar
[INFO]
[INFO] --- maven-shade-plugin:3.1.0:shade (aspect-oriented-programming) @ part1.assignment2 ---
[INFO] Including org.springframework:spring-core:jar:5.1.8.RELEASE in the shaded jar.
[INFO] Including org.springframework:spring-jcl:jar:5.1.8.RELEASE in the shaded jar.
[INFO] Including org.springframework:spring-context:jar:5.1.8.RELEASE in the shaded jar.
[INFO] Including org.springframework:spring-aop:jar:5.1.8.RELEASE in the shaded jar.
[INFO] Including org.springframework:spring-beans:jar:5.1.8.RELEASE in the shaded jar.
[INFO] Including org.springframework:spring-expression:jar:5.1.8.RELEASE in the shaded jar.
[INFO] Including org.slf4j:slf4j-log4j12:jar:2.0.0-alpha0 in the shaded jar.
[INFO] Including org.slf4j:slf4j-api:jar:2.0.0-alpha0 in the shaded jar.
[INFO] Including log4j:log4j:jar:1.2.17 in the shaded jar.
[INFO] Including javax.annotation:javax.annotation-api:jar:1.3.2 in the shaded jar.
[INFO] Replacing original artifact with shaded artifact.
[INFO] Replacing /Users/morrison/git/rseg105/part1/assignment2/target/part1.assignment2-1.0.0.jar with /Users/morrison/git/rseg105/part1/assignment2/target/part1.assignment2-1.0.0-shaded.jar
[INFO] Dependency-reduced POM written at: /Users/morrison/git/rseg105/part1/assignment2/dependency-reduced-pom.xml
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ part1.assignment2 ---
[INFO] Installing /Users/morrison/git/rseg105/part1/assignment2/target/part1.assignment2-1.0.0.jar to /Users/morrison/.m2/repository/edu/morrison/spring/part1.assignment2/1.0.0/part1.assignment2-1.0.0.jar
[INFO] Installing /Users/morrison/git/rseg105/part1/assignment2/dependency-reduced-pom.xml to /Users/morrison/.m2/repository/edu/morrison/spring/part1.assignment2/1.0.0/part1.assignment2-1.0.0.pom
[INFO]
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.266 s
[INFO] Finished at: 2022-02-20T07:02:05-05:00
[INFO] -----
==== Running BookBeforeAdvice ====
2022-02-20 07:02:02,757 INFO [edu.morrison.spring.app.BookBeforeAdvice] - <Before 'reading' buy the book.>
2022-02-20 07:02:02,771 INFO [edu.morrison.spring.beans.Book] - <Reading Anna Karenina book>
==== Running BookAfterReturningAdvice ====
2022-02-20 07:02:03,231 INFO [edu.morrison.spring.beans.Book] - <Reading To Kill A Mockingbird book>
2022-02-20 07:02:03,232 INFO [edu.morrison.spring.app.BookAfterReturningAdvice] - <After 'reading' write the book review.>
==== Running BookThrowsAdvice ====
2022-02-20 07:02:03,709 INFO [edu.morrison.spring.app.BookThrowsAdvice] - <***>
2022-02-20 07:02:03,709 INFO [edu.morrison.spring.app.BookThrowsAdvice] - <RuntimeException Captured>
2022-02-20 07:02:03,710 INFO [edu.morrison.spring.app.BookThrowsAdvice] - <Caught: java.lang.RuntimeException>
2022-02-20 07:02:03,710 INFO [edu.morrison.spring.app.BookThrowsAdvice] - <Method: throwException>
***
```

(base) morrison@agua assignment2 %

### 3. ASSIGNMENT 3

The third assignment was by far my favorite and challenging, with all objectives fully achieved. I first set up the SQL database, creating the database, tables, and populating with data. As shown, the database was successfully populated.

```
(base) morrison@guu sql % cat ddl.sql
CREATE USER IF NOT EXISTS 'spring'@'localhost' IDENTIFIED BY 'spring';

DROP DATABASE IF EXISTS BOOKS;
USE BOOKS;
GRANT ALL PRIVILEGES ON BOOKS.* TO 'spring'@'localhost';
FLUSH PRIVILEGES;

SET GLOBAL time_zone = '+5:00';
(base) morrison@guu sql % cat schema.sql
CREATE TABLE CATEGORY (
    ID INT NOT NULL AUTO_INCREMENT,
    NAME VARCHAR(100),
    PRIMARY KEY (ID)
);

CREATE TABLE BOOK (
    ID INT NOT NULL AUTO_INCREMENT,
    CATEGORY_ID INT(11),
    ISBN VARCHAR(14),
    TITLE VARCHAR(500),
    PRICE DECIMAL(5,2),
    PRIMARY KEY (ID)
);
(base) morrison@guu sql % cat test-data.sql
INSERT INTO CATEGORY (ID, NAME) VALUES (1, 'Novels');
INSERT INTO CATEGORY (ID, NAME) VALUES (2, 'Java');
INSERT INTO CATEGORY (ID, NAME) VALUES (3, 'Programming');
INSERT INTO CATEGORY (ID, NAME) VALUES (4, 'Poetry');

INSERT INTO BOOK (CATEGORY_ID, ISBN, TITLE, PRICE) VALUES (1, "978-0075536321", "Anna Karenina", 17.57);
INSERT INTO BOOK (CATEGORY_ID, ISBN, TITLE, PRICE) VALUES (1, "978-0060194994", "To Kill A Mockingbird", 25.95);
INSERT INTO BOOK (CATEGORY_ID, ISBN, TITLE, PRICE) VALUES (2, "978-017150021", "Effective Java", 43.86);
INSERT INTO BOOK (CATEGORY_ID, ISBN, TITLE, PRICE) VALUES (3, "978-0132350884", "Clean Code", 25.33);
INSERT INTO BOOK (CATEGORY_ID, ISBN, TITLE, PRICE) VALUES (4, "978-0571096343", "The Waste Land by T.S. Eliot", 99.99);
INSERT INTO BOOK (CATEGORY_ID, ISBN, TITLE, PRICE) VALUES (4, "978-0375505966", "Still I Rise by Maya Angelou", 30.29);

-- Test data added directly by the app
-- INSERT INTO BOOK (CATEGORY_ID, ISBN, TITLE, PRICE) VALUES (3, "978-0367505035", "Fundamentals of Computer Graphics", 126.0);
```

```
m Morrison — 103x52

mysql> delete from Category; delete from Book;
ed (0.00 sec)

Query OK, 6 rows affected (0.00 sec)

mysql> source /Users/morrison/git/rseg105/part1/assignment3/src/main/resources/sql/test-data.sql
Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

mysql> select * from category;
+----+----+
| ID | NAME |
+----+----+
| 1 | Novels |
| 2 | Java    |
| 3 | Programming |
| 4 | Poetry  |
+----+----+
4 rows in set (0.00 sec)

mysql> select * from book;
+----+----+----+----+----+
| ID | CATEGORY_ID | ISBN      | TITLE          | PRICE |
+----+----+----+----+----+
| 30 |           1 | 978-0075536321 | Anna Karenina | 17.57 |
| 31 |           1 | 978-0060194994 | To Kill A Mockingbird | 25.95 |
| 32 |           2 | 978-0137150021 | Effective Java | 43.86 |
| 33 |           3 | 978-0132350884 | Clean Code     | 25.33 |
| 34 |           4 | 978-0571096343 | The Waste Land by T.S. Eliot | 99.99 |
| 35 |           4 | 978-0375505966 | Still I Rise by Maya Angelou | 30.29 |
+----+----+----+----+----+
6 rows in set (0.00 sec)

mysql>
```

For the assignment, I created a main runner application that can either run all four demos at once, or each individually. Here are all run at once and their respective outputs:

```
[INFO] --- maven-install-plugin:2.4:install (default-install) @ part1.assignment3 ---
[INFO] Installing /Users/morrison/git/rseg105/part1/assignment3/target/part1.assignment3-1.0-SNAPSHOT.jar to /Users/morrison/.m2/repository/edu/morrison/spring/part1.assignment3/1.0-SNAPSHOT/part1.assignment3-1.0-SNAPSHOT.jar
[INFO] Installing /Users/morrison/git/rseg105/part1/assignment3/pom.xml to /Users/morrison/.m2/repository/edu/morrison/spring/part1.assignment3/1.0-SNAPSHOT/part1.assignment3-1.0-SNAPSHOT.pom
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.824 s
[INFO] Finished at: 2022-02-20T07:08:38-05:00
[INFO] -----
== Running FindBookCategoryJdbcApp ======
07:08:39,577 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 27 - Find Book by Category Spring JDBC Demo
07:08:39,578 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 28 - v0.0.1 by C.S. Morrison
07:08:39,578 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 29 -
07:08:39,579 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 30 - Usage: edu.morrison.spring.app.FindBookCategoryJdbcApp [1234]
07:08:39,579 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 31 -
07:08:39,579 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 32 - This is a demonstration of reading and writing to a database using Spring JDBC.
07:08:39,579 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 33 - Specify a number to only run one specific demo which allows database to be inspected.
07:08:39,579 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 44 - ===== Start of Find Books By Category =====
07:08:39,579 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 127 - ----- Demo 1: Printing books and categories -----
07:08:39,579 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 128 - All Books:
07:08:39,607 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 87, Category Id: 1, ISBN: 978-0075536321, Title: Anna Karenina, Price: 17.57
07:08:39,607 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 88, Category Id: 1, ISBN: 978-0060194994, Title: To Kill A Mockingbird, Price: 25.95
07:08:39,607 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 89, Category Id: 2, ISBN: 978-0137150201, Title: Effective Java, Price: 43.86
07:08:39,608 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 90, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
07:08:39,608 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 91, Category Id: 4, ISBN: 978-0571096343, Title: The Waste Land by T.S. Eliot, Price: 99.99
07:08:39,609 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 92, Category Id: 4, ISBN: 978-0375505966, Title: Still I Rise by Maya Angelou, Price: 30.29
07:08:39,609 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 131 - Poetry Books:
07:08:39,611 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 91, Category Id: 4, ISBN: 978-0571096343, Title: The Waste Land by T.S. Eliot, Price: 99.99
07:08:39,611 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 92, Category Id: 4, ISBN: 978-0375505966, Title: Still I Rise by Maya Angelou, Price: 30.29
07:08:39,611 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 134 - All Categories:
07:08:39,613 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 154 - Novels
07:08:39,614 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 154 - Java
07:08:39,614 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 154 - Programming
07:08:39,614 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 154 - Poetry
07:08:39,614 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 107 - ----- Demo 2: Adding a book -----
07:08:39,614 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 111 - Programming books BEFORE add:
07:08:39,616 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 90, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
07:08:39,635 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 121 - Programming books AFTER add:
07:08:39,637 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 90, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
07:08:39,637 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 112, Category Id: 3, ISBN: 978-0367505035, Title: Fundamentals of Computer Graphics, Price: 126.0
07:08:39,637 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 83 - ----- Demo 3: Modifying a book -----
07:08:39,637 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 87 - Programming books BEFORE modify:
07:08:39,638 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 90, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
07:08:39,639 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 112, Category Id: 3, ISBN: 978-0367505035, Title: Fundamentals of Computer Graphics, Price: 126.0
07:08:39,642 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 101 - Programming books AFTER modify:
07:08:39,644 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 90, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
07:08:39,644 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 112, Category Id: 3, ISBN: 978-0367505035, Title: Fundamentals of Computer Graphics, v5, Price: 125.99
07:08:39,644 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 62 - ----- Demo 4: Deleting a book -----
07:08:39,644 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 66 - Programming books BEFORE delete:
07:08:39,645 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 90, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
07:08:39,645 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 112, Category Id: 3, ISBN: 978-0367505035, Title: Fundamentals of Computer Graphics, v5, Price: 125.99
07:08:39,649 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 77 - Programming books AFTER delete:
07:08:39,650 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 90, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
07:08:39,650 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 55 - ----- End of Find Books By Category -----
(base) morrison@agua assignment3 %
```

In order to avoid creating separate main(), I decided to create a single runner that took an argument representing which demo to run, and allowing them to run individually.

```
(base) morrison@agua assignment3 % cat run.sh
#!/bin/sh

mvn clean install
echo "==== Running FindBookCategoryJdbcApp ====="
java -jar ./target/part1-assignment3.jar edu.morrison.spring.app.FindBookCategoryJdbcApp $*
```

Running the demo programs one at a time, we can verify in MySQL that they're performing as specified.

```
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.307 s
[INFO] Finished at: 2022-02-20T07:21:39+05:00
[INFO] -----
==== Running FindBookCategoryJdbcApp ======
07:21:40,426 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 27 - Find Book by Category Spring JDBC Demo
07:21:40,427 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 28 - v0.0.1 by C.S. Morrison
07:21:40,427 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 29 -
07:21:40,427 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 30 - Usage: edu.morrison.spring.app.FindBookCategoryJdbcApp [1234]
07:21:40,428 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 31 -
07:21:40,428 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 32 - This is a demonstration of reading and writing to a database using Spring JDBC.
07:21:40,428 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 33 - Specify a number to only run one specific demo which allows database to be inspected.
07:21:40,428 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 34 -
07:21:40,428 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 44 ===== Start of Find Books By Category =====
07:21:40,428 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 101 - ===== Demo 2: Adding a book =====
07:21:40,428 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 101 - Programming books BEFORE add:
07:21:40,456 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 98, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
07:21:40,479 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 98, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
07:21:40,479 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 113, Category Id: 3, ISBN: 978-0367505035, Title: Fundamentals of Computer Graphics, Price: 126.0
07:21:40,479 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 55 ===== End of Find Books By Category =====
```

```
[mysql]> select * from book;
+----+-----+-----+-----+-----+
| ID | CATEGORY_ID | ISBN          | TITLE           | price |
+----+-----+-----+-----+-----+
| 87 |           1 | 978-0075536321 | Anna Karenina   | 17.57 |
| 88 |           1 | 978-0060194994 | To Kill A Mockingbird | 25.95 |
| 89 |           2 | 978-0137150021 | Effective Java   | 43.86 |
| 90 |           3 | 978-0132350884 | Clean Code        | 25.33 |
| 91 |           4 | 978-0571096343 | The Waste Land by T.S. Eliot | 99.99 |
| 92 |           4 | 978-0375505966 | Still I Rise by Maya Angelou | 30.29 |
| 113 |          3 | 978-0367505035 | Fundamentals of Computer Graphics | 126.0 |
+----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

And after modification, the change is persisted. Since the instructions said to save the book with a category name, I took that literally:

```
categoryDao.addBook(book, "Programming");
```

and implemented a lookup to obtain the corresponding category ID for the book being added (instead of assuming/using a number).

```
==== Running FindBookCategoryJdbcApp ======
07:23:59,961 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 27 - Find Book by Category Spring JDBC Demo
07:23:59,962 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 28 - v0.0.1 by C.S. Morrison
07:23:59,962 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 29 -
07:23:59,963 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 30 - Usage: edu.morrison.spring.app.FindBookCategoryJdbcApp [1234]
07:23:59,963 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 31 -
07:23:59,963 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 32 - This is a demonstration of reading and writing to a database using Spring JDBC.
07:23:59,963 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 33 - Specify a number to only run one specific demo which allows database to be inspected.
07:23:59,963 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 34 -
07:23:59,964 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 44 ===== Start of Find Books By Category =====
07:23:59,964 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 83 - ===== Demo 3: Modifying a book =====
07:23:59,964 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 87 - Programming books BEFORE modify:
07:23:59,994 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 98, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
07:23:59,994 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 113, Category Id: 3, ISBN: 978-0367505035, Title: Fundamentals of Computer Graphics, Price: 126.0
07:24:00,001 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 101 - Programming books AFTER modify:
07:24:00,001 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 98, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
07:24:00,004 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 113, Category Id: 3, ISBN: 978-0367505035, Title: Fundamentals of Computer Graphics, v5, Price: 125.99
07:24:00,006 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 55 ===== End of Find Books By Category =====
```

```
[mysql]> select * from book;
+----+-----+-----+-----+-----+
| ID | CATEGORY_ID | ISBN          | TITLE           | price |
+----+-----+-----+-----+-----+
| 87 |           1 | 978-0075536321 | Anna Karenina   | 17.57 |
| 88 |           1 | 978-0060194994 | To Kill A Mockingbird | 25.95 |
| 89 |           2 | 978-0137150021 | Effective Java   | 43.86 |
| 90 |           3 | 978-0132350884 | Clean Code        | 25.33 |
| 91 |           4 | 978-0571096343 | The Waste Land by T.S. Eliot | 99.99 |
| 92 |           4 | 978-0375505966 | Still I Rise by Maya Angelou | 30.29 |
| 113 |          3 | 978-0367505035 | Fundamentals of Computer Graphics, v5 | 125.99 |
+----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

Finally, we demonstrate delete working – it's gone.

```
[INFO] -----
== Running FindBookCategoryJdbcApp =====
07:25:02,912 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 27 - Find Book by Category Spring JDBC Demo
07:25:02,913 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 28 - v0.0.1 by C.S. Morrison
07:25:02,913 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 29 - 
07:25:02,914 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 30 - Usage: edu.morrison.spring.app.FindBookCategoryJdbcApp [1234]
07:25:02,914 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 31 - 
07:25:02,914 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 32 - This is a demonstration of reading and writing to a database using Spring JDBC.
07:25:02,914 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 33 - Specify a number to only run one specific demo which allows database to be inspected.
07:25:02,914 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 34 - 
07:25:02,914 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 44 - ===== Start of Find Books By Category =====
07:25:02,914 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 62 - ===== Demo 4: Deleting a book =====
07:25:02,915 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 66 - Programming books BEFORE delete:
07:25:02,941 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 90, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
07:25:02,941 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 113, Category Id: 3, ISBN: 978-0367505035, Title: Fundamentals of Computer Graphics, v5, Price: 125.99
07:25:02,947 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 77 - Programming books AFTER delete:
07:25:02,949 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 144 - Book - Id: 90, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
07:25:02,949 INFO edu.morrison.spring.app.FindBookCategoryJdbcApp: 55 - ===== End of Find Books By Category =====
```

```
[mysql]> select * from book;
+----+-----+-----+-----+-----+-----+
| ID | CATEGORY_ID | ISBN           | TITLE          | price |
+----+-----+-----+-----+-----+
| 87 |           1 | 978-0075536321 | Anna Karenina | 17.57 |
| 88 |           1 | 978-0060194994 | To Kill A Mockingbird | 25.95 |
| 89 |           2 | 978-0137150021 | Effective Java | 43.86 |
| 90 |           3 | 978-0132350884 | Clean Code     | 25.33 |
| 91 |           4 | 978-0571096343 | The Waste Land by T.S. Eliot | 99.99 |
| 92 |           4 | 978-0375505966 | Still I Rise by Maya Angelou | 30.29 |
+----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

## 4. ASSIGNMENT 4

Shortly into my coding on assignment 4, after having established the database and a DAO interface. I ran into an exception on application startup that I spent nearly an entire day debugging. Certain the issue was in my DAO or in the Hibernate setup, I spent hours exploring changes such as changing my Java-based configuration, trying different file locations, changing my AppConfig class to look in different paths, utilizing different main() drivers, and more. Here is the top of the exception stack I was presented with:

```
(base) morrison@agu hibernate-crud % java -jar target/find-all-demo.jar com.apress.prospring5.ch7.FindAllDemo
05:41:06,371 INFO    org.hibernate.Version: 45 - HHH000412: Hibernate Core {[WORKING]}
05:41:06,375 INFO    org.hibernate.cfg.Environment: 211 - HHH000206: hibernate.properties not found
05:41:06,415 INFO  org.hibernate.annotations.common.Version: 66 - HCANN000001: Hibernate Commons Annotations {5.0.1.Final}
05:41:07,117 INFO  org.hibernate.dialect.Dialect: 157 - HHH000400: Using dialect: org.hibernate.dialect.MySQL5Dialect
05:41:07,476 WARN ation.AnnotationConfigApplicationContext: 557 - Exception encountered during context initialization - cancelling refresh attempt:
org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'instrumentDao': Injection of resource dependencies failed; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'sessionFactory' defined in
com.apress.prospring5.ch7.config.AppConfig: Bean instantiation via factory method failed; nested exception is
org.springframework.beans.BeanInstantiationException: Failed to instantiate [org.hibernate.SessionFactory]: Factory method 'sessionFactory' threw exception;
nested exception is org.hibernate.MappingException: Could not get constructor for org.hibernate.persister.entity.SingleTableEntityPersister
Exception in thread "main" org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'instrumentDao': Injection of resource
dependencies failed; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'sessionFactory' defined in
com.apress.prospring5.ch7.config.AppConfig: Bean instantiation via factory method failed; nested exception is
org.springframework.beans.BeanInstantiationException: Failed to instantiate [org.hibernate.SessionFactory]: Factory method 'sessionFactory' threw exception;
nested exception is org.hibernate.MappingException: Could not get constructor for org.hibernate.persister.entity.SingleTableEntityPersister
at
org.springframework.context.annotation.CommonAnnotationBeanPostProcessor.postProcessProperties(CommonAnnotationBeanPostProcessor.java:324)
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.populateBean(AbstractAutowireCapableBeanFactory.java:1411)
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:592)
at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:515)
at org.springframework.beans.factory.support.AbstractBeanFactory.lambda$doGetBean$0(AbstractBeanFactory.java:320)
at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:222)
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:318)
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:199)
at org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBeanFactory.java:845)
at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:877)
at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:549)
at org.springframework.context.annotation.AnnotationConfigApplicationContext.(AnnotationConfigApplicationContext.java:88)
at com.apress.prospring5.ch7.FindAllDemo.main(FindAllDemo.java:23)
Caused by: org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'sessionFactory' defined in
com.apress.prospring5.ch7.config.AppConfig: Bean instantiation via factory method failed; nested exception is
org.springframework.beans.BeanInstantiationException: Failed to instantiate [org.hibernate.SessionFactory]: Factory method 'sessionFactory' threw exception;
nested exception is org.hibernate.MappingException: Could not get constructor for org.hibernate.persister.entity.SingleTableEntityPersister
at org.springframework.beans.factory.support.ConstructorResolver.instantiate(ConstructorResolver.java:627)
at org.springframework.beans.factory.support.ConstructorResolver.instantiateUsingFactoryMethod(ConstructorResolver.java:456)
... more followed ...
```

From that exception, I assumed there was something wrong with my “instrumentDao” bean and/or “sessionFactory” that I had implemented, as that appeared to be what the exception was indicating. After nearly giving up, I returned to the provided hibernate-crud example only to find that it had the exact same issue, which led me to realize it must be a dependency issue or bug of some sort. As such, I began to look at each of the dependencies and their versions. Upon upgrading the javassist dependency to version 3.25.0, the exception went away and the beans that I previously through weren’t working were suddenly working just fine.

While it’s still not really clear to me exactly why and how an older javassist was resulting an exception being thrown, I was able to proceed from there with the assignment, very thankfully with an assignment extension as my time-table for the week was blown. Regardless, with the exception finally gone, I proceeded with the remaining assignment implementation as specified.

The first task that was already working and verified was database setup. My dataset was an expansion of the previous assignments and included 8 books, 4 categories, and 4 authors total with 2 categories containing more than 1 book and 2 authors having written 2 books. Here is the summary from MySQL:

```
m Morrison - 121x78
mysql> show tables;
+-----+
| Tables_in_morebooks |
+-----+
| AUTHOR           |
| AUTHOR_BOOK      |
| BOOK             |
| CATEGORY         |
+-----+
4 rows in set (0.00 sec)

mysql> select * from CATEGORY;
+----+----+
| ID | NAME |
+----+----+
| 1  | Novels |
| 2  | Java   |
| 3  | Programming |
| 4  | Poetry |
+----+----+
4 rows in set (0.00 sec)

mysql> select * from BOOK;
+----+----+----+----+
| ID | CATEGORY_ID | ISBN        | TITLE          | PRICE |
+----+----+----+----+
| 1  | 1            | 978-0075536321 | Anna Karenina    | 17.57 |
| 2  | 1            | 978-0060194994 | To Kill A Mockingbird | 25.95 |
| 3  | 2            | 978-0137150021 | Effective Java    | 43.86 |
| 4  | 3            | 978-0132350884 | Clean Code        | 25.33 |
| 5  | 4            | 978-0571096343 | The Waste Land by T.S. Eliot | 99.99 |
| 6  | 4            | 978-0375505966 | Still I Rise by Maya Angelou | 30.29 |
| 7  | 4            | 978-0156332255 | Four Quartets     | 8.99  |
| 8  | 4            | 978-1400079988 | War and Peace     | 21.00 |
+----+----+----+----+
8 rows in set (0.00 sec)

mysql> select * from AUTHOR;
+----+----+----+----+
| ID | FIRST_NAME | LAST_NAME | DESCRIPTION
+----+----+----+----+
| 1  | Leo        | Tolstoy    | Count Lev Nikolayevich Tolstoy, usually referred to in English as Leo Tolstoy, was a Russian writer who is regarded as one of the greatest authors of all time. He received nominations for the Nobel Prize in Literature every year from 1902 to 1906 and for the Nobel Peace Prize in 1901, 1902, and 1909. |
| 2  | Harper     | Lee        | Nelle Harper Lee was an American novelist best known for her 1960 novel To Kill a Mockingbird. It won the 1961 Pulitzer Prize and has become a classic of modern American literature. |
| 3  | Thomas     | Eliot      | Thomas Stearns Eliot OM was a poet, essayist, publisher, playwright, literary critic and editor. Considered one of the 20th century's major poets, he is a central figure in English-language Modernist poetry. |
| 4  | Maya       | Angelou    | Maya Angelou was an American poet, memoirist, and civil rights activist. She published seven autobiographies, three books of essays, several books of poetry, and is credited with a list of plays, movies, and television shows spanning over 50 years. She received dozens of awards and more than 50 honorary degrees. |
+----+----+----+----+
4 rows in set (0.00 sec)

mysql> select * from AUTHOR_BOOK;
+----+----+
| AUTHOR_ID | BOOK_ID |
+----+----+
| 1          | 1        |
| 2          | 2        |
| 3          | 5        |
| 4          | 6        |
| 3          | 7        |
| 1          | 8        |
+----+----+
6 rows in set (0.00 sec)

mysql>
```

Next, we demonstrate the ability to find a book, its category, and its authors given a book ID:

```

== Running PublishingApp =====
00:08:51,773 INFO org.hibernate.Version: 45 - HHH000412: Hibernate Core {[WORKING]}
00:08:51,775 INFO org.hibernate.cfg.Environment: 211 - HHH000206: hibernate.properties not found
00:08:51,810 INFO org.hibernate.annotations.common.Version: 66 - HCANN000001: Hibernate Commons Annotations {5.0.1.Final}
00:08:52,277 INFO org.hibernate.dialect.Dialect: 157 - HHH000400: Using dialect: org.hibernate.dialect.MySQL5Dialect
00:08:52,616 INFO rnatne.hql.internal.QueryTranslatorFactoryInitiator: 47 - HHH000397: Using ASTQueryTranslatorFactory
00:08:52,716 INFO edu.morrison.spring.app.PublishingApp: 24 - Book Publishing Hibernate Demo
00:08:52,716 INFO edu.morrison.spring.app.PublishingApp: 25 - v0.0.1 by C.S. Morrison
00:08:52,717 INFO edu.morrison.spring.app.PublishingApp: 26 -
00:08:52,717 INFO edu.morrison.spring.app.PublishingApp: 27 - Usage: edu.morrison.spring.app.PublishingApp
00:08:52,717 INFO edu.morrison.spring.app.PublishingApp: 28 -
00:08:52,717 INFO edu.morrison.spring.app.PublishingApp: 29 - This is a demonstration of persisting entities via Spring Hibernate.
00:08:52,717 INFO edu.morrison.spring.app.PublishingApp: 30 - Specify a number to only run one specific demo which allows database to be inspected.
00:08:52,717 INFO edu.morrison.spring.app.PublishingApp: 31 -
00:08:52,717 INFO edu.morrison.spring.app.PublishingApp: 41 - ===== Start of Publishing Demo =====
00:08:52,718 INFO edu.morrison.spring.app.PublishingApp: 110 - ----- Demo 1: Printing books and categories -----
00:08:52,718 INFO edu.morrison.spring.app.PublishingApp: 125 - Looking up book ID: 3
Hibernate:
/* Book.findBookWithAuthorCategoryById */ select
    distinct book0_.id as id1_2_0_,
    author2_.id as id1_0_1,
    category3_.id as id1_3_2,
    book0_.CATEGORY_ID as CATEGORY5_2_0_,
    book0_.isbn as isbn2_2_0_,
    book0_.price as price3_2_0_,
    book0_.title as title4_2_0_,
    author2_.description as description2_0_1,
    author2_.FIRST_NAME as FIRST_NAM3_0_1,
    author2_.LAST_NAME as LAST_NAM4_0_1,
    authors1_.BOOK_ID as BOOK_ID1_1_0_,
    authors1_.AUTHOR_ID as AUTHOR_I2_1_0_,
    category3_.name as name2_3_2_
from
    BOOK book0_
left outer join
    AUTHOR_BOOK authors1_
        on book0_.id=authors1_.BOOK_ID
left outer join
    Author author2_
        on authors1_.AUTHOR_ID=author2_.id
left outer join
    CATEGORY category3_
        on book0_.CATEGORY_ID=category3_.id
where
    book0_.id=?
00:08:52,808 INFO edu.morrison.spring.app.PublishingApp: 127 - Book - Id: 3, Category Id: 2, ISBN: 978-0137150021, Title: Effective Java, Price: 43.86
00:08:52,808 INFO edu.morrison.spring.app.PublishingApp: 130 - Category - id: 2, Name: Java
00:08:52,811 INFO edu.morrison.spring.app.PublishingApp: 52 - ===== End of Publishing Demo =====
(base) morrison@agu assignment4 %

```

The demo shown above is set up to pull one of the first 8 random books, and we get “lucky” with it picking the one book (#4) that does not have an author in the database, and it’s handled correctly. For the remainder of output listings, we turn off the helpful “show\_sql” debugging information that was used in development. Running again, we get book #6 which has an author:

```

== Running PublishingApp =====
00:23:39,589 INFO org.hibernate.Version: 45 - HHH000412: Hibernate Core {[WORKING]}
00:23:39,591 INFO org.hibernate.cfg.Environment: 211 - HHH000206: hibernate.properties not found
00:23:39,620 INFO org.hibernate.annotations.common.Version: 66 - HCANN000001: Hibernate Commons Annotations {5.0.1.Final}
00:23:40,043 INFO org.hibernate.dialect.Dialect: 157 - HHH000400: Using dialect: org.hibernate.dialect.MySQL5Dialect
00:23:40,369 INFO rnatne.hql.internal.QueryTranslatorFactoryInitiator: 47 - HHH000397: Using ASTQueryTranslatorFactory
00:23:40,464 INFO edu.morrison.spring.app.PublishingApp: 24 - Book Publishing Hibernate Demo
00:23:40,464 INFO edu.morrison.spring.app.PublishingApp: 25 - v0.0.1 by C.S. Morrison
00:23:40,464 INFO edu.morrison.spring.app.PublishingApp: 26 -
00:23:40,465 INFO edu.morrison.spring.app.PublishingApp: 27 - Usage: edu.morrison.spring.app.PublishingApp
00:23:40,465 INFO edu.morrison.spring.app.PublishingApp: 28 -
00:23:40,465 INFO edu.morrison.spring.app.PublishingApp: 29 - This is a demonstration of persisting entities via Spring Hibernate.
00:23:40,465 INFO edu.morrison.spring.app.PublishingApp: 30 - Specify a number to only run one specific demo which allows database to be inspected.
00:23:40,465 INFO edu.morrison.spring.app.PublishingApp: 31 -
00:23:40,465 INFO edu.morrison.spring.app.PublishingApp: 41 - ===== Start of Publishing Demo =====
00:23:40,465 INFO edu.morrison.spring.app.PublishingApp: 110 - ----- Demo 1: Printing books and categories -----
00:23:40,466 INFO edu.morrison.spring.app.PublishingApp: 125 - Looking up book ID: 6
00:23:40,554 INFO edu.morrison.spring.app.PublishingApp: 127 - Book - Id: 6, Category Id: 4, ISBN: 978-0375505966, Title: Still I Rise by Maya Angelou, Price: 30.29
00:23:40,554 INFO edu.morrison.spring.app.PublishingApp: 130 - Category - id: 4, Name: Poetry
00:23:40,557 INFO edu.morrison.spring.app.PublishingApp: 132 - Author - Id: 4, First name: Maya, Last name: Angelou, Description: Maya Angelou was an American poet, memoirist, and civil rights activist. She published seven autobiographies, three books of essays, several books of poetry, and is credited with a list of plays, movies, and television shows spanning over 50 years. She received dozens of awards and more than 50 honorary degrees.
00:23:40,559 INFO edu.morrison.spring.app.PublishingApp: 52 - ===== End of Publishing Demo =====

```

Next, we see all books by author ID. Here we see our two books by author ID = 3:

```
== Running PublishingApp =====
03:12:11,231 INFO org.hibernate.Version: 45 - HHH000412: Hibernate Core {[WORKING]}
03:12:11,234 INFO org.hibernate.cfg.Environment: 211 - HHH000206: hibernate.properties not found
03:12:11,263 INFO org.hibernate.annotations.common.Version: 66 - HCANN000001: Hibernate Commons Annotations {5.0.1.Final}
03:12:11,692 INFO org.hibernate.dialect.Dialect: 157 - HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
03:12:12,022 INFO rnate.hql.internal.QueryTranslatorFactoryInitiator: 47 - HHH000397: Using ASTQueryTranslatorFactory
03:12:12,119 INFO edu.morrison.spring.app.PublishingApp: 26 - Book Publishing Hibernate Demo
03:12:12,119 INFO edu.morrison.spring.app.PublishingApp: 27 - v0.0.1 by C.S. Morrison
03:12:12,120 INFO edu.morrison.spring.app.PublishingApp: 28 -
03:12:12,120 INFO edu.morrison.spring.app.PublishingApp: 29 - Usage: edu.morrison.spring.app.PublishingApp
03:12:12,120 INFO edu.morrison.spring.app.PublishingApp: 30 -
03:12:12,120 INFO edu.morrison.spring.app.PublishingApp: 31 - This is a demonstration of persisting entities via Spring Hibernate.
03:12:12,120 INFO edu.morrison.spring.app.PublishingApp: 32 - Specify a number to only run one specific demo which allows database to be inspected.
03:12:12,120 INFO edu.morrison.spring.app.PublishingApp: 33 -
03:12:12,120 INFO edu.morrison.spring.app.PublishingApp: 43 - ===== Start of Publishing Demo =====
03:12:12,120 INFO edu.morrison.spring.app.PublishingApp: 106 - ----- Demo 2: Listing all books by author ID -----
03:12:12,121 INFO edu.morrison.spring.app.PublishingApp: 111 - Looking up author ID: 3
03:12:12,206 INFO edu.morrison.spring.app.PublishingApp: 133 - Book - Id: 5, Category Id: 4, ISBN: 978-0571096343, Title: The Waste Land by T.S. Eliot, Price: 99.99
03:12:12,206 INFO edu.morrison.spring.app.PublishingApp: 136 - Category - id: 4, Name: Poetry
03:12:12,209 INFO edu.morrison.spring.app.PublishingApp: 138 - Author - Id: 3, First name: Thomas, Last name: Eliot, Description: Thomas Stearns Eliot OM was a poet, essayist, publisher, playwright, literary critic and editor. Considered one of the 20th century's major poets, he is a central figure in English-language Modernist poetry.
03:12:12,209 INFO edu.morrison.spring.app.PublishingApp: 133 - Book - Id: 7, Category Id: 4, ISBN: 978-0156332255, Title: Four Quartets, Price: 8.99
03:12:12,209 INFO edu.morrison.spring.app.PublishingApp: 136 - Category - id: 4, Name: Poetry
03:12:12,209 INFO edu.morrison.spring.app.PublishingApp: 138 - Author - Id: 3, First name: Thomas, Last name: Eliot, Description: Thomas Stearns Eliot OM was a poet, essayist, publisher, playwright, literary critic and editor. Considered one of the 20th century's major poets, he is a central figure in English-language Modernist poetry.
03:12:12,212 INFO edu.morrison.spring.app.PublishingApp: 54 - ===== End of Publishing Demo =====
```

Next, we create a new book with multiple authors that do not yet exist in the book. We specify a category that already exists (i.e., “Programming”) whose category ID we first look up in order to set that as the book’s category ID. Similarly, we create the books Authors, write them to the database, and then save out our book. This action automatically updates our book and author\_book table accordingly!

```
== Running PublishingApp =====
05:25:44,499 INFO org.hibernate.Version: 45 - HHH000412: Hibernate Core {[WORKING]}
05:25:44,501 INFO org.hibernate.cfg.Environment: 211 - HHH000206: hibernate.properties not found
05:25:44,530 INFO org.hibernate.annotations.common.Version: 66 - HCANN000001: Hibernate Commons Annotations {5.0.1.Final}
05:25:44,964 INFO org.hibernate.dialect.Dialect: 157 - HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
05:25:45,281 INFO rnate.hql.internal.QueryTranslatorFactoryInitiator: 47 - HHH000397: Using ASTQueryTranslatorFactory
05:25:45,387 INFO edu.morrison.spring.app.PublishingApp: 30 - Book Publishing Hibernate Demo
05:25:45,387 INFO edu.morrison.spring.app.PublishingApp: 31 - v0.0.1 by C.S. Morrison
05:25:45,387 INFO edu.morrison.spring.app.PublishingApp: 32 -
05:25:45,387 INFO edu.morrison.spring.app.PublishingApp: 33 - Usage: edu.morrison.spring.app.PublishingApp
05:25:45,387 INFO edu.morrison.spring.app.PublishingApp: 34 -
05:25:45,387 INFO edu.morrison.spring.app.PublishingApp: 35 - This is a demonstration of persisting entities via Spring Hibernate.
05:25:45,387 INFO edu.morrison.spring.app.PublishingApp: 36 - Specify a number to only run one specific demo which allows database to be inspected.
05:25:45,387 INFO edu.morrison.spring.app.PublishingApp: 37 -
05:25:45,388 INFO edu.morrison.spring.app.PublishingApp: 47 - ===== Start of Publishing Demo =====
05:25:45,388 INFO edu.morrison.spring.app.PublishingApp: 96 - ----- Demo 3: Adding a book -----
05:25:45,388 INFO edu.morrison.spring.app.PublishingApp: 100 - All books BEFORE add:
05:25:45,482 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 1, Category Id: 1, ISBN: 978-0075536321, Title: Anna Karenina, Price: 17.57
05:25:45,483 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 2, Category Id: 1, ISBN: 978-0060194994, Title: To Kill A Mockingbird, Price: 25.95
05:25:45,483 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 3, Category Id: 2, ISBN: 978-0137150021, Title: Effective Java, Price: 43.86
05:25:45,483 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 4, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
05:25:45,483 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 5, Category Id: 4, ISBN: 978-0571096343, Title: The Waste Land by T.S. Eliot, Price: 99.99
05:25:45,483 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 6, Category Id: 4, ISBN: 978-0375505966, Title: Still I Rise by Maya Angelou, Price: 30.29
05:25:45,483 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 7, Category Id: 4, ISBN: 978-0156332255, Title: Four Quartets, Price: 8.99
05:25:45,483 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 8, Category Id: 4, ISBN: 978-1400079988, Title: War and Peace, Price: 21.0
05:25:45,483 INFO edu.morrison.spring.app.PublishingApp: 110 - Looking up category by name: Programming
05:25:45,488 INFO edu.morrison.spring.app.PublishingApp: 112 - Category - id: 3, Name: Programming
05:25:45,499 INFO edu.morrison.spring.dao.AuthorDaoImpl: 23 - Author saved with id: 9
05:25:45,500 INFO edu.morrison.spring.dao.AuthorDaoImpl: 23 - Author saved with id: 10
05:25:45,508 INFO edu.morrison.spring.app.PublishingApp: 132 - Adding book [Fundamentals of Computer Graphics]
05:25:45,511 INFO edu.morrison.spring.dao.BookDaoImpl: 50 - Book saved with id: 12
05:25:45,517 INFO edu.morrison.spring.app.PublishingApp: 135 - All books AFTER add:
05:25:45,529 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 1, Category Id: 1, ISBN: 978-0075536321, Title: Anna Karenina, Price: 17.57
05:25:45,529 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 2, Category Id: 1, ISBN: 978-0060194994, Title: To Kill A Mockingbird, Price: 25.95
05:25:45,529 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 3, Category Id: 2, ISBN: 978-0137150021, Title: Effective Java, Price: 43.86
05:25:45,529 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 4, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
05:25:45,530 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 5, Category Id: 4, ISBN: 978-0571096343, Title: The Waste Land by T.S. Eliot, Price: 99.99
05:25:45,530 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 6, Category Id: 4, ISBN: 978-0375505966, Title: Still I Rise by Maya Angelou, Price: 30.29
05:25:45,530 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 7, Category Id: 4, ISBN: 978-0156332255, Title: Four Quartets, Price: 8.99
05:25:45,530 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 8, Category Id: 4, ISBN: 978-1400079988, Title: War and Peace, Price: 21.0
05:25:45,530 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 12, Category Id: 3, ISBN: 978-0367505035, Title: Fundamentals of Computer Graphics, Price: 126.0
05:25:45,530 INFO edu.morrison.spring.app.PublishingApp: 139 - Book actually written:
05:25:45,537 INFO edu.morrison.spring.app.PublishingApp: 173 - Book - Id: 12, Category Id: 3, ISBN: 978-0367505035, Title: Fundamentals of Computer Graphics, Price: 126.0
05:25:45,538 INFO edu.morrison.spring.app.PublishingApp: 176 - Category - id: 3, Name: Programming
05:25:45,541 INFO edu.morrison.spring.app.PublishingApp: 178 - Author - Id: 10, First name: Peter, Last name: Shirley, Description: American computer scientist and computer graphic s researcher.
05:25:45,541 INFO edu.morrison.spring.app.PublishingApp: 178 - Author - Id: 9, First name: Steve, Last name: Marschner, Description: Professor of Computer Science, Cornell Universi ty
05:25:45,542 INFO edu.morrison.spring.app.PublishingApp: 58 - ===== End of Publishing Demo =====
```

To confirm our book addition, we inspect the database to find our book and its respective authors correctly inserted (book ID #12 and authors ID #9 and #10).

```
m Morrison - 121x78

9 rows in set (0.00 sec)

mysql> select * from category;
+----+-----+
| ID | NAME |
+----+-----+
| 1 | Novels
| 2 | Java
| 3 | Programming
| 4 | Poetry
+----+-----+
4 rows in set (0.00 sec)

mysql> select * from book;
+----+-----+-----+-----+-----+
| ID | CATEGORY_ID | ISBN      | TITLE          | PRICE   |
+----+-----+-----+-----+-----+
| 1  | 1           | 978-0075536321 | Anna Karenina    | 17.57  |
| 2  | 1           | 978-0060194994 | To Kill A Mockingbird | 25.95  |
| 3  | 2           | 978-0137150021 | Effective Java    | 43.86  |
| 4  | 3           | 978-0132350884 | Clean Code        | 25.33  |
| 5  | 4           | 978-0571096343 | The Waste Land by T.S. Eliot | 99.99  |
| 6  | 4           | 978-0375505966 | Still I Rise by Maya Angelou | 30.29  |
| 7  | 4           | 978-0156332255 | Four Quartets     | 8.99   |
| 8  | 4           | 978-1400079988 | War and Peace     | 21.00  |
| 12 | 3           | 978-0367505035 | Fundamentals of Computer Graphics | 126.00 |
+----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

mysql> select * from author;
+----+-----+-----+-----+
| ID | FIRST_NAME | LAST_NAME | DESCRIPTION
+----+-----+-----+-----+
| 1  | Leo       | Tolstoy   | Count Lev Nikolayevich Tolstoy, usually referred to in English as Leo Tolstoy, was a Russian writer who is regarded as one of the greatest authors of all time. He received nominations for the Nobel Prize in Literature every year from 1902 to 1906 and for the Nobel Peace Prize in 1901, 1902, and 1909.
| 2  | Harper    | Lee       | Nelle Harper Lee was an American novelist best known for her 1960 novel To Kill a Mockingbird. It won the 1961 Pulitzer Prize and has become a classic of modern American literature.
| 3  | Thomas    | Eliot     | Thomas Stearns Eliot OM was a poet, essayist, publisher, playwright, literary critic and editor. Considered one of the 20th century's major poets, he is a central figure in English-language Modernist poetry.
| 4  | Maya      | Angelou   | Maya Angelou was an American poet, memoirist, and civil rights activist. She published seven autobiographies, three books of essays, several books of poetry, and is credited with a list of plays, movies, and television shows spanning over 50 years. She received dozens of awards and more than 50 honorary degrees.
| 9  | Steve     | Marschner | Professor of Computer Science, Cornell University
| 10 | Peter     | Shirley   | American computer scientist and computer graphics researcher.
+----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from author_book;
+----+-----+
| AUTHOR_ID | BOOK_ID |
+----+-----+
| 1          | 1        |
| 2          | 2        |
| 3          | 5        |
| 4          | 6        |
| 3          | 7        |
| 1          | 8        |
| 10         | 12       |
| 9          | 12       |
+----+-----+
8 rows in set (0.00 sec)

mysql>
```

Last but not least, we delete the saved book and authors from the database by deleting our book. To demonstrate that this is working correctly, we dynamically look up the book by title first, and then issue the delete calls.

```
== Running PublishingApp =====
05:29:30,514 INFO org.hibernate.Version: 45 - HHH000412: Hibernate Core {[WORKING]}
05:29:30,516 INFO org.hibernate.cfg.Environment: 211 - HHH000206: hibernate.properties not found
05:29:30,546 INFO org.hibernate.annotations.common.Version: 66 - HCANN000001: Hibernate Commons Annotations {5.0.1.Final}
05:29:31,158 INFO org.hibernate.dialect.Dialect: 157 - HHH000400: Using dialect: org.hibernate.dialect.MySQL5Dialect
05:29:31,483 INFO rnatne.hql.internal.QueryTranslatorFactoryInitiator: 47 - HHH000397: Using ASTQueryTranslatorFactory
05:29:31,597 INFO edu.morrison.spring.app.PublishingApp: 30 - Book Publishing Hibernate Demo
05:29:31,597 INFO edu.morrison.spring.app.PublishingApp: 31 - v0.0.1 by C.S. Morrison
05:29:31,597 INFO edu.morrison.spring.app.PublishingApp: 32 -
05:29:31,598 INFO edu.morrison.spring.app.PublishingApp: 33 - Usage: edu.morrison.spring.app.PublishingApp
05:29:31,598 INFO edu.morrison.spring.app.PublishingApp: 34 -
05:29:31,598 INFO edu.morrison.spring.app.PublishingApp: 35 - This is a demonstration of persisting entities via Spring Hibernate.
05:29:31,598 INFO edu.morrison.spring.app.PublishingApp: 36 - Specify a number to only run one specific demo which allows database to be inspected.
05:29:31,598 INFO edu.morrison.spring.app.PublishingApp: 37 -
05:29:31,598 INFO edu.morrison.spring.app.PublishingApp: 47 - ===== Start of Publishing Demo =====
05:29:31,598 INFO edu.morrison.spring.app.PublishingApp: 64 - ----- Demo 4: Deleting a book -----
05:29:31,598 INFO edu.morrison.spring.app.PublishingApp: 66 - All books BEFORE delete:
05:29:31,691 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 1, Category Id: 1, ISBN: 978-0075536321, Title: Anna Karenina, Price: 17.57
05:29:31,691 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 2, Category Id: 1, ISBN: 978-0060194994, Title: To Kill A Mockingbird, Price: 25.95
05:29:31,691 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 3, Category Id: 2, ISBN: 978-0137150021, Title: Effective Java, Price: 43.86
05:29:31,691 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 4, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
05:29:31,691 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 5, Category Id: 4, ISBN: 978-0571096343, Title: The Waste Land by T.S. Eliot, Price: 99.99
05:29:31,691 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 6, Category Id: 4, ISBN: 978-0375505966, Title: Still I Rise by Maya Angelou, Price: 30.29
05:29:31,691 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 7, Category Id: 4, ISBN: 978-0156332255, Title: Four Quartets, Price: 8.99
05:29:31,691 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 8, Category Id: 4, ISBN: 978-1400079988, Title: War and Peace, Price: 21.0
05:29:31,691 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 12, Category Id: 3, ISBN: 978-0367505035, Title: Fundamentals of Computer Graphics, Price: 126.0
05:29:31,699 INFO edu.morrison.spring.app.PublishingApp: 75 - Found our book to delete, id: 12
05:29:31,699 INFO edu.morrison.spring.app.PublishingApp: 173 - Book - Id: 12, Category Id: 3, ISBN: 978-0367505035, Title: Fundamentals of Computer Graphics, Price: 126.0
05:29:31,699 INFO edu.morrison.spring.app.PublishingApp: 176 - Category - id: 3, Name: Programming
05:29:31,702 INFO edu.morrison.spring.app.PublishingApp: 178 - Author - Id: 10, First name: Peter, Last name: Shirley, Description: American computer scientist and computer graphic s researcher.
05:29:31,702 INFO edu.morrison.spring.app.PublishingApp: 178 - Author - Id: 9, First name: Steve, Last name: Marschner, Description: Professor of Computer Science, Cornell Universi ty
05:29:31,708 INFO edu.morrison.spring.dao.AuthorDaoImpl: 29 - Author deleted with id: 10
05:29:31,718 INFO edu.morrison.spring.dao.AuthorDaoImpl: 29 - Author deleted with id: 9
05:29:31,723 INFO edu.morrison.spring.dao.BookDaoImpl: 56 - Book deleted with id: 12
05:29:31,730 INFO edu.morrison.spring.app.PublishingApp: 83 - All books AFTER delete:
05:29:31,740 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 1, Category Id: 1, ISBN: 978-0075536321, Title: Anna Karenina, Price: 17.57
05:29:31,740 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 2, Category Id: 1, ISBN: 978-0060194994, Title: To Kill A Mockingbird, Price: 25.95
05:29:31,740 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 3, Category Id: 2, ISBN: 978-0137150021, Title: Effective Java, Price: 43.86
05:29:31,740 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 4, Category Id: 3, ISBN: 978-0132350884, Title: Clean Code, Price: 25.33
05:29:31,740 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 5, Category Id: 4, ISBN: 978-0571096343, Title: The Waste Land by T.S. Eliot, Price: 99.99
05:29:31,741 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 6, Category Id: 4, ISBN: 978-0375505966, Title: Still I Rise by Maya Angelou, Price: 30.29
05:29:31,741 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 7, Category Id: 4, ISBN: 978-0156332255, Title: Four Quartets, Price: 8.99
05:29:31,741 INFO edu.morrison.spring.app.PublishingApp: 186 - Book - Id: 8, Category Id: 4, ISBN: 978-1400079988, Title: War and Peace, Price: 21.0
05:29:31,744 INFO edu.morrison.spring.app.PublishingApp: 58 - ===== End of Publishing Demo =====
```

Finally, we query the database yet again to confirm. Our book and authors are indeed deleted:

```
morrison — 121x78

| 1 | 1 |
| 2 | 2 |
| 3 | 5 |
| 4 | 6 |
| 3 | 7 |
| 1 | 8 |
| 10 | 12 |
| 9 | 12 |
+---+---+
8 rows in set (0.00 sec)

mysql> select * from category;
+----+-----+
| ID | NAME   |
+----+-----+
| 1  | Novels |
| 2  | Java    |
| 3  | Programming |
| 4  | Poetry  |
+----+-----+
4 rows in set (0.00 sec)

mysql> select * from book;
+----+-----+-----+-----+
| ID | CATEGORY_ID | ISBN      | TITLE          | PRICE |
+----+-----+-----+-----+
| 1  | 1           | 978-0075536321 | Anna Karenina  | 17.57 |
| 2  | 1           | 978-0060194994 | To Kill A Mockingbird | 25.95 |
| 3  | 2           | 978-0137150021 | Effective Java | 43.86 |
| 4  | 3           | 978-0132350884 | Clean Code     | 25.33 |
| 5  | 4           | 978-0571096343 | The Waste Land by T.S. Eliot | 99.99 |
| 6  | 4           | 978-0375505966 | Still I Rise by Maya Angelou | 30.29 |
| 7  | 4           | 978-0156332255 | Four Quartets | 8.99  |
| 8  | 4           | 978-1400079988 | War and Peace  | 21.00 |
+----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> select * from author;
+----+-----+-----+
| ID | FIRST_NAME | LAST_NAME | DESCRIPTION
+----+-----+-----+
| 1  | Leo        | Tolstoy   | Count Lev Nikolayevich Tolstoy, usually referred to in English as Leo Tolstoy, was a Russian writer who is regarded as one of the greatest authors of all time. He received nominations for the Nobel Prize in Literature every year from 1902 to 1906 and for the Nobel Peace Prize in 1901, 1902, and 1909.
| 2  | Harper     | Lee       | Nelle Harper Lee was an American novelist best known for her 1960 novel To Kill a Mockingbird. It won the 1961 Pulitzer Prize and has become a classic of modern American literature.
| 3  | Thomas     | Eliot     | Thomas Stearns Eliot OM was a poet, essayist, publisher, playwright, literary critic and editor. Considered one of the 20th century's major poets, he is a central figure in English-language Modernist poetry.
| 4  | Maya       | Angelou   | Maya Angelou was an American poet, memoirist, and civil rights activist. She published seven autobiographies, three books of essays, several books of poetry, and is credited with a list of plays, movies, and television shows spanning over 50 years. She received dozens of awards and more than 50 honorary degrees.
+----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from author_book;
+----+-----+
| AUTHOR_ID | BOOK_ID |
+----+-----+
| 1  | 1 |
| 2  | 2 |
| 3  | 5 |
| 4  | 6 |
| 3  | 7 |
| 1  | 8 |
+----+-----+
6 rows in set (0.00 sec)

mysql>
```

The only remaining item in all of the above that I didn't yet demonstrate was creating and calling a task using `getNamedQuery()` without parameters. This is pretty straightforward but simply wasn't a use case needed to satisfy the other assignment criteria, so I decided to implement a 5<sup>th</sup> demo case that simply prints all authors using that method. The results are thus as follows:

```
== Running PublishingApp =====
06:05:54,868 INFO org.hibernate.Version: 45 - HHH000412: Hibernate Core { [WORKING]}
06:05:54,870 INFO org.hibernate.cfg.Environment: 211 - HHH000206: hibernate.properties not found
06:05:54,899 INFO org.hibernate.annotations.common.Version: 66 - HCANN000001: Hibernate Commons Annotations {5.0.1.Final}
06:05:55,333 INFO org.hibernate.dialect.Dialect:Dialect: 157 - HHH000400: Using dialect: org.hibernate.dialect.MySQL5Dialect
06:05:55,657 INFO rnatr.hql.internal.QueryTranslatorFactoryInitiator: 47 - HHH000397: Using ASTQueryTranslatorFactory
06:05:55,758 INFO edu.morrison.spring.app.PublishingApp: 30 - Book Publishing Hibernate Demo
06:05:55,758 INFO edu.morrison.spring.app.PublishingApp: 31 - v0.0.1 by C.S. Morrison
06:05:55,758 INFO edu.morrison.spring.app.PublishingApp: 32 -
06:05:55,758 INFO edu.morrison.spring.app.PublishingApp: 33 - Usage: edu.morrison.spring.app.PublishingApp[12345]
06:05:55,759 INFO edu.morrison.spring.app.PublishingApp: 34 -
06:05:55,759 INFO edu.morrison.spring.app.PublishingApp: 35 - This is a demonstration of persisting entities via Spring Hibernate.
06:05:55,759 INFO edu.morrison.spring.app.PublishingApp: 36 - Specify a number to only run one specific demo which allows database to be inspected.
06:05:55,759 INFO edu.morrison.spring.app.PublishingApp: 37 -
06:05:55,759 INFO edu.morrison.spring.app.PublishingApp: 47 - ===== Start of Publishing Demo =====
06:05:55,759 INFO edu.morrison.spring.app.PublishingApp: 66 - ----- Demo 5: Printing authors -----
06:05:55,836 INFO edu.morrison.spring.app.PublishingApp: 72 - All authors:
06:05:55,841 INFO edu.morrison.spring.app.PublishingApp: 73 - Author - Id: 1, First name: Leo, Last name: Tolstoy, Description: Count Lev Nikolayevich Tolstoy, usually referred to in English as Leo Tolstoy, was a Russian writer who is regarded as one of the greatest authors of all time. He received nominations for the Nobel Prize in Literature every year from 1902 to 1906 and for the Nobel Peace Prize in 1901, 1902, and 1909.
06:05:55,841 INFO edu.morrison.spring.app.PublishingApp: 73 - Author - Id: 2, First name: Harper, Last name: Lee, Description: Nelle Harper Lee was an American novelist best known for her 1960 novel To Kill a Mockingbird. It won the 1961 Pulitzer Prize and has become a classic of modern American literature.
06:05:55,841 INFO edu.morrison.spring.app.PublishingApp: 73 - Author - Id: 3, First name: Thomas, Last name: Eliot, Description: Thomas Stearns Eliot OM was a poet, essayist, publisher, playwright, literary critic and editor. Considered one of the 20th century's major poets, he is a central figure in English-language Modernist poetry.
06:05:55,841 INFO edu.morrison.spring.app.PublishingApp: 73 - Author - Id: 4, First name: Maya, Last name: Angelou, Description: Maya Angelou was an American poet, memoirist, and civil rights activist. She published seven autobiographies, three books of essays, several books of poetry, and is credited with a list of plays, movies, and television shows spanning over 50 years. She received dozens of awards and more than 50 honorary degrees.
06:05:55,844 INFO edu.morrison.spring.app.PublishingApp: 60 - ===== End of Publishing Demo =====
```

## Summary:

Assignment #4 was by far the most challenging \*and\* interesting to date. I did my best to approach this implementation as I would a real testing application development, hence implementing a single main with runners instead of multiple main and relying on caller scripting. That is also the reason for collocating all database logic in one location (`resources/sql`) instead of spreading them out across multiple folders.

All assignments are accompanied by a “run.sh” script that demonstrates how to call the program. For assignments #3 and #4 at present time, they both optionally take an argument representing the demo number with #3 having 4 demos and #4 having 5 demos, e.g.: `./run.sh 3`

There's undoubtedly room for improvement as I attempted different approaches to object management. I also strived to match the spirit of the assignment output without attempting to match output exactly for all assignments (although assignment #1 is nearly a perfect match). That all said, I believe I did achieve all of the specified functionality and implementation requirements for all assignments. Thank you.