Exercise 2.1 – Using MySQL and Hibernate Annotations

The main objective of this exercise is to practice entity creation, value types and collections.

1. Open MySQL Workbench (at the end of the installation there is a check box for that)
   a) Select **Local Instance MySQL56** connection
   b) On the Query execute the following SQL

```
CREATE SCHEMA cs544 ;
```

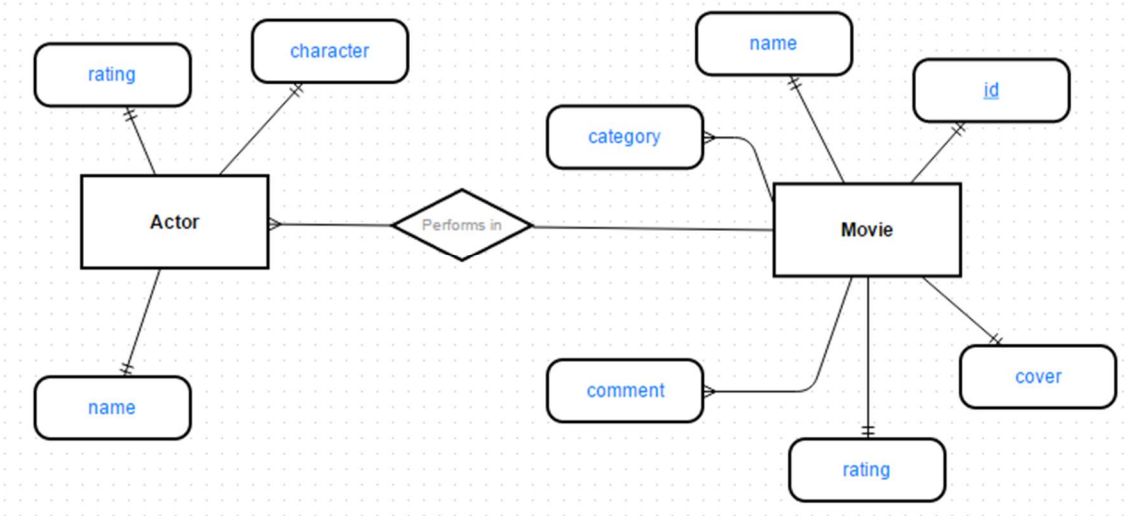
   You can do the same using the right click menu options over the existing schema

   Note that you only have to create this database schema only once for the entire course

2. Import the provided project to your workspace: hw2-1
3. Add the **Hibernate dependencies** to the projectôs pom.xml.

```
hibernate-entitymanager : 5.1.0.Final
hibernate-jpa-2.1-api : 1.0.0.Final
mysql-connector-java : 5.1.37
```

4. Create the classes to represents the following diagram



   A Movie has a set of categories (comedy, drama, é), a list of comments, and a list of Actors. The cover is binary field containing an image.

5. Set the JPA annotations

Please note: Character is a MySQL reserved keyword

6. Add a few Movies with their actors. You can use any image file for the cover

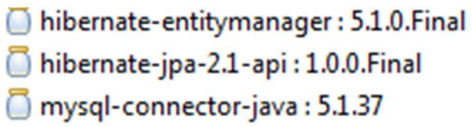
```
import java.nio.file.FileSystems;
import java.nio.file.Files;
import java.nio.file.Path;

// ...
        Path p = FileSystems.getDefault().getPath("", "myFile");
        byte [] fileData = Files.readAllBytes(p);
```

7. List all the movie names with corresponding actors

Exercise 2.2 – Inheritance

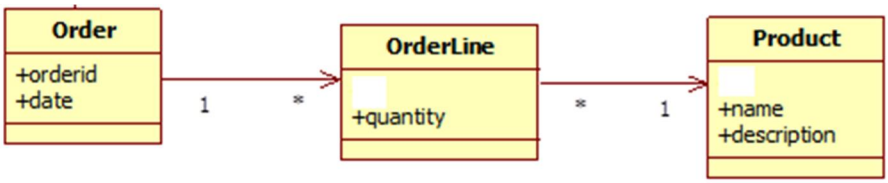
The objective of this exercise is to practice mapping inheritance. The choice of how you wish to implement your inheritance mapping, and which collections you wish to use for your association mappings is left up to you.

Start this exercise by creating a new maven project called **exercise02-2.** Add the **Hibernate dependencies** to the projectõs pom.xml.



This exercise consists out of 2 parts. In the first part you will create a basic application that does not use inheritance. Once you have tested to see that the application works, the second part asks you to add inheritance, and test it again.

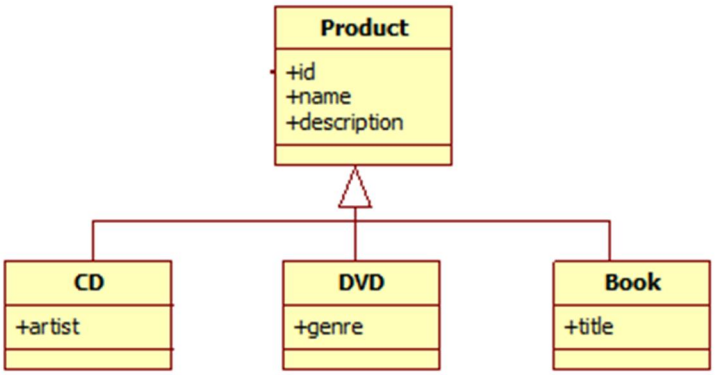
a) Create a small application that uses the following entities:



Please note: Order is a SQL reserved keyword; you can create an `Order` table by escaping it with back-tick. Alternately you can use the @Entity annotation to provide a different name for the entity.

Test the application by creating an object tree of the given domain classes, saving this tree to the database, and then retrieving the objects from the database to check if everything was persisted correctly.

b) Create another application to use the following updated domain:



Test the application by creating an object tree (now containing CDs, DVDs, and Books), saving it to the database, and then retrieving it to check if everything was persisted correctly.