

Arbeitsblatt JPA: JPQL & JOINS

Ziele

Ziel dieses Arbeitsblattes ist es, dass sie ein JOIN mit JPA formuliert haben.

Ausgangslage

Im Gitlab-Projekt <https://gitlab.fhnw.ch/eaf/hs21/04> steht das Projekt lab-jpa-jpql zur Verfügung, welches für diese Aufgaben verwendet werden soll. Importieren Sie dieses Projekt in ihre IDE.

Wie im Arbeitsblatt von letzter Woche verwenden wir auch in diesem Projekt Spring um den Entity-Manager zu erzeugen. Die Parameter in der Datei `application.properties` sind so gesetzt, dass eine in-memory Datenbank verwendet wird:

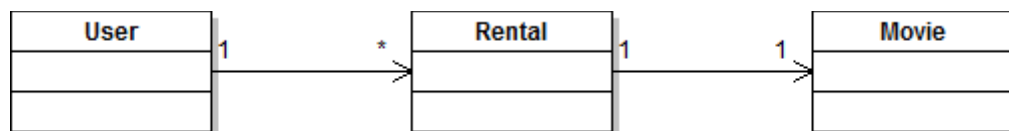
```
spring.datasource.url=jdbc:h2:mem:lab-jpa-jpql
```

Das Datenbankschema lassen wir von JPA generieren. Dies wird durch den Wert `update` oder `create` bei der Eigenschaft

```
spring.jpa.hibernate.ddl-auto=create
```

erreicht. Die Modellklassen haben wir *nicht* mit `@Column`-Annotationen markiert, daher werden die Default-Werte für die Tabellen- und Kolonnennamen übernommen.

Im Gegensatz zur Übung werden nur *unidirektionale* Assoziationen verwendet, d.h. das Modell vereinfacht sich zu



Aufgaben:

- 1) In der Klasse `Test1` wird der Film mit `ID=1` geladen und dessen Titel ausgegeben. Ergänzen Sie dieses Programm mit einem Query das ausgibt, wer diesen Film ausgeliehen hat.

```
"SELECT u from User u ..."
```

Wenn Sie die ausgegeben SQL-Statements formatiert sehen möchten, dann können Sie in der Datei `application.properties` folgende Eigenschaften setzen:

```
spring.jpa.properties.hibernate.format_sql=true
```

- 2) Wie könnte das Query aus Aufgabe 1) formuliert werden wenn die Assoziation zwischen User und Rental bidirektional modelliert wäre?
Hinweis: In diesem Fall muss *kein* explizites JOIN mehr formuliert werden!