JPA

Martin Ledvinka

martin.ledvinka@fel.cvut.cz

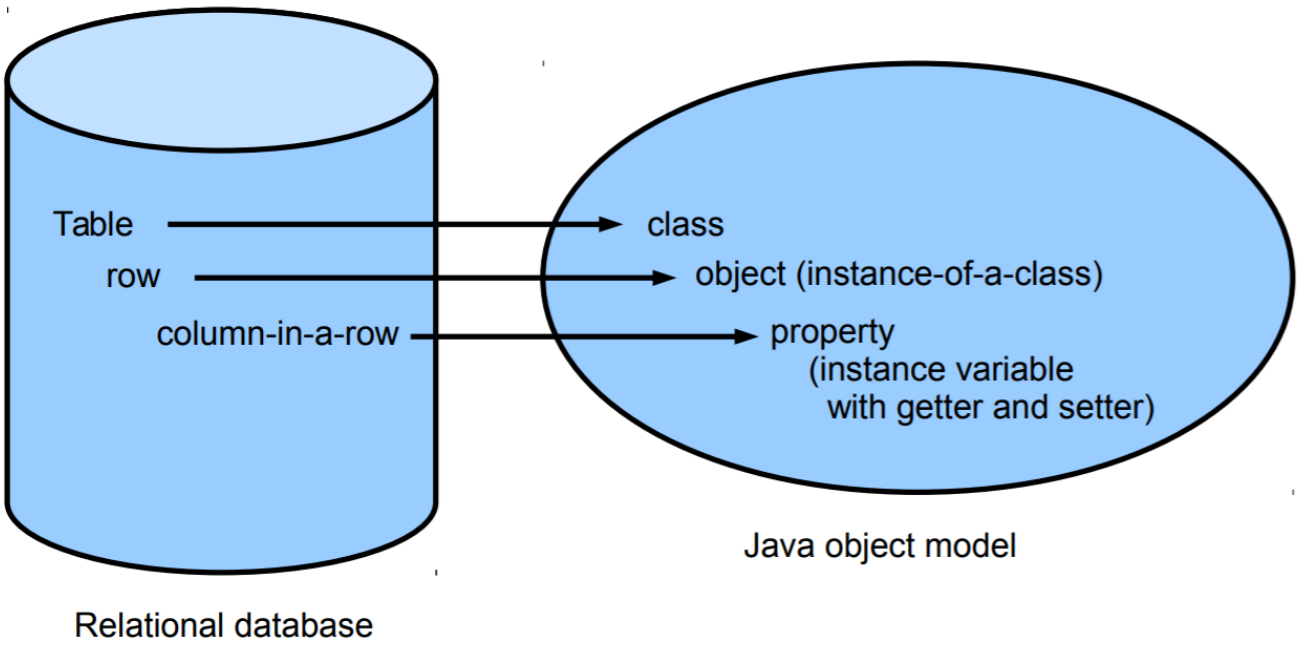
Winter Term 2020

Contents

Introduction

Object-relational mapping

Mapping between the database (declarative) schema and the data structures in an object oriented language.



JPA Basics

- The idea: "map Java classes to database records"
- Object-relational mapping in Java

```
@Entity
public Person {
  @Id
  private Long id;
  private String hasName;
  // setters+getters
}
```

```
CREATE TABLE PERSON (
  ID bigint PRIMARY KEY NOT NULL,
  HASNAME varchar(255)
);
```

JPA Main Concepts

- Entity class: a Java class representing a set of persistent objects mapped onto a relational table
- Persistence Unit: the set of all entity classes that are persistently mapped to one database
- Persistence Context: the set of all entities defined in the persistence unit being used at a given time
- Entity manager: the interface for interacting with a Persistence Context

JPA Main Concepts – Visual

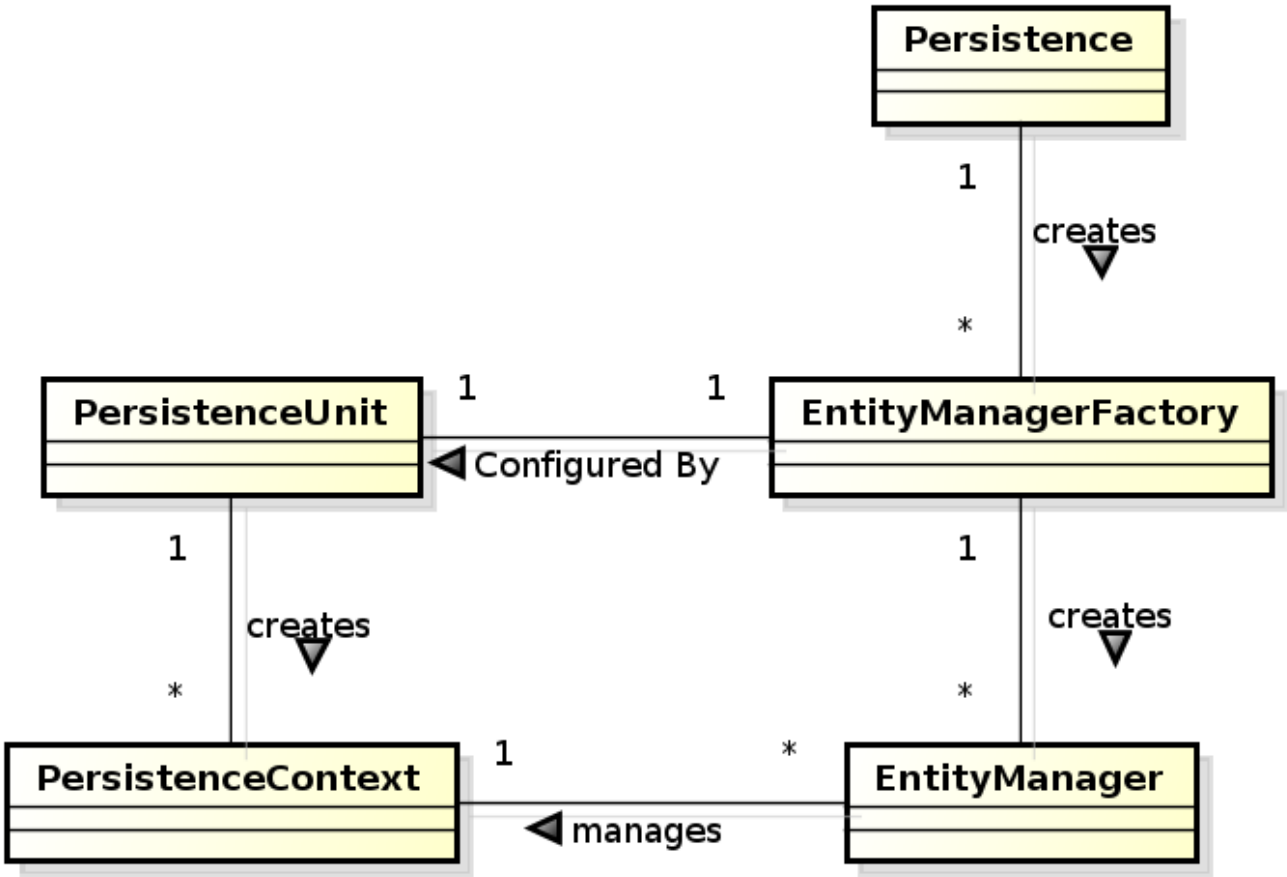


Figure: Main JPA concepts and their relationship.

JPA – EntityManager

- **EntityManager (EM)** instance is in fact a generic DAO, while entities can be understood as DTO's.
- Selected operations on EM (CRUD):

Create : `em.persist`
Read : `em.find`, `em.refresh`
Update : `em.merge`
Delete : `em.remove`
Native/JPQL queries : `em.createNativeQuery`, `em.createQuery`, `em.createNamedQuery`, etc.
Resource-local transactions : `em.getTransaction.[begin,commit,rollback]`

Implementations

- We will be using *Eclipselink*
  - Reference JPA implementation
- Another (arguably more popular) solution is *Hibernate*
  - JPA implementation + its own, `Session`-based API
  - `Session` ∼ `EntityManager`
  - Provided by the Spring Boot `spring-boot-starter-data-jpa` project by default
- More: *Apache OpenJPA*, *DataNucleus*, etc.
- Each implementation has its quirks, additional options/specific API, and problems

Tasks

Syncing Your Fork

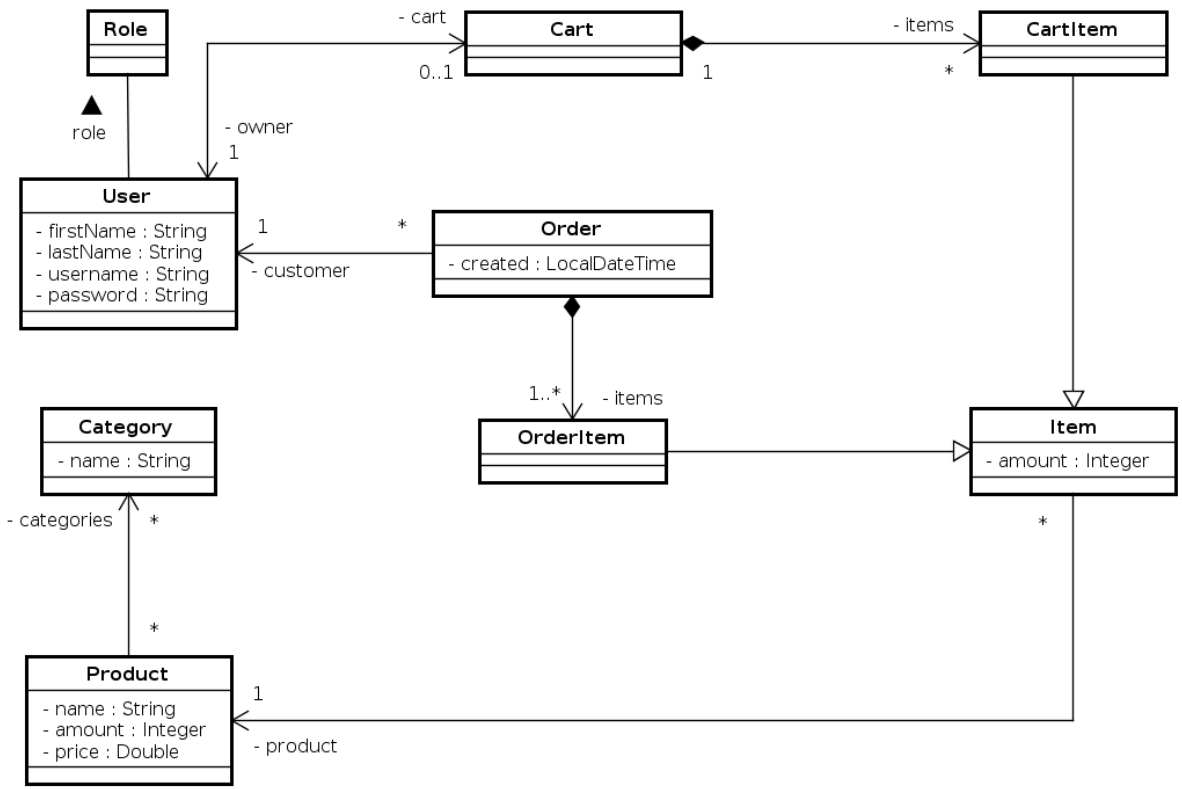
1. Fetch branches and commits from the upstream repository (EAR/B201-eshop)
   - `git fetch upstream`
2. Check out local branch corresponding to the task branch
   - `git checkout -b b201-seminar-03-task`
3. Merge changes from the corresponding upstream branch
   - `git merge upstream/b201-seminar-03-task`
4. Do your task
5. Push the solution to your fork
   - `git push origin b201-seminar-03-task`

Task: 1 Point

The application is missing an object model (JPA entity classes). Create an object model corresponding to the class diagram below.

We will start together, your task is to finish it.

Acceptance criteria: Project is compilable and the object model corresponds to the class diagram below.



The End

The End

Thank You

Resources

- https://cw.fel.cvut.cz/b201/_media/courses/b6b36ear/lectures/lecture-03-jpa-s.pdf
- http://www.objectdb.com/api/java/jpa/annotations