JPA

Martin Ledvinka

martin.ledvinka@fel.cvut.cz

Winter Term 2023

Contents

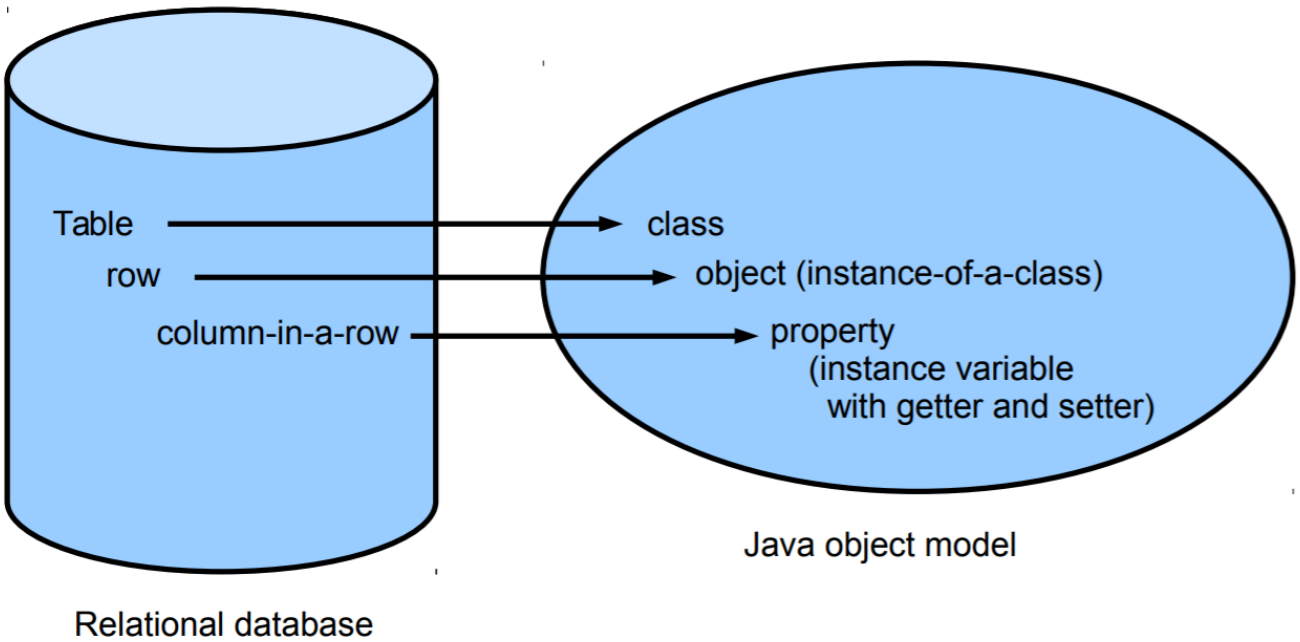
1 Introduction

2 Tasks

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

```java
@Entity
public class Person {
  @Id
  private Long id;
  private String firstName;
  private String lastName
  // setters+getters
}
```

```sql
CREATE TABLE PERSON (
  ID bigint PRIMARY KEY NOT NULL,
  FIRSTNAME varchar(255),
  LASTNAME 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

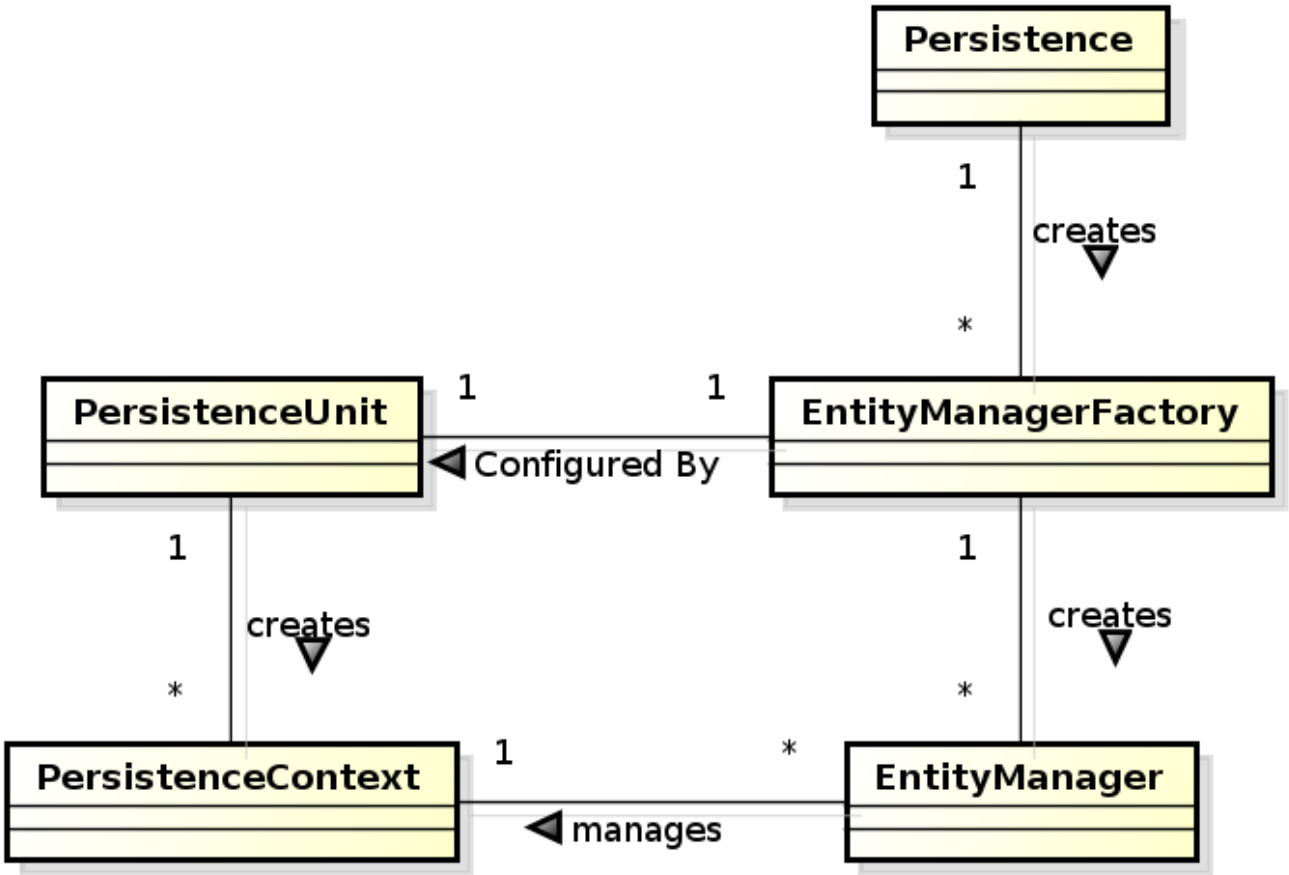


Figure: Main JPA concepts and their relationship.

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

- *Eclipselink*
  - Reference JPA implementation
- *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

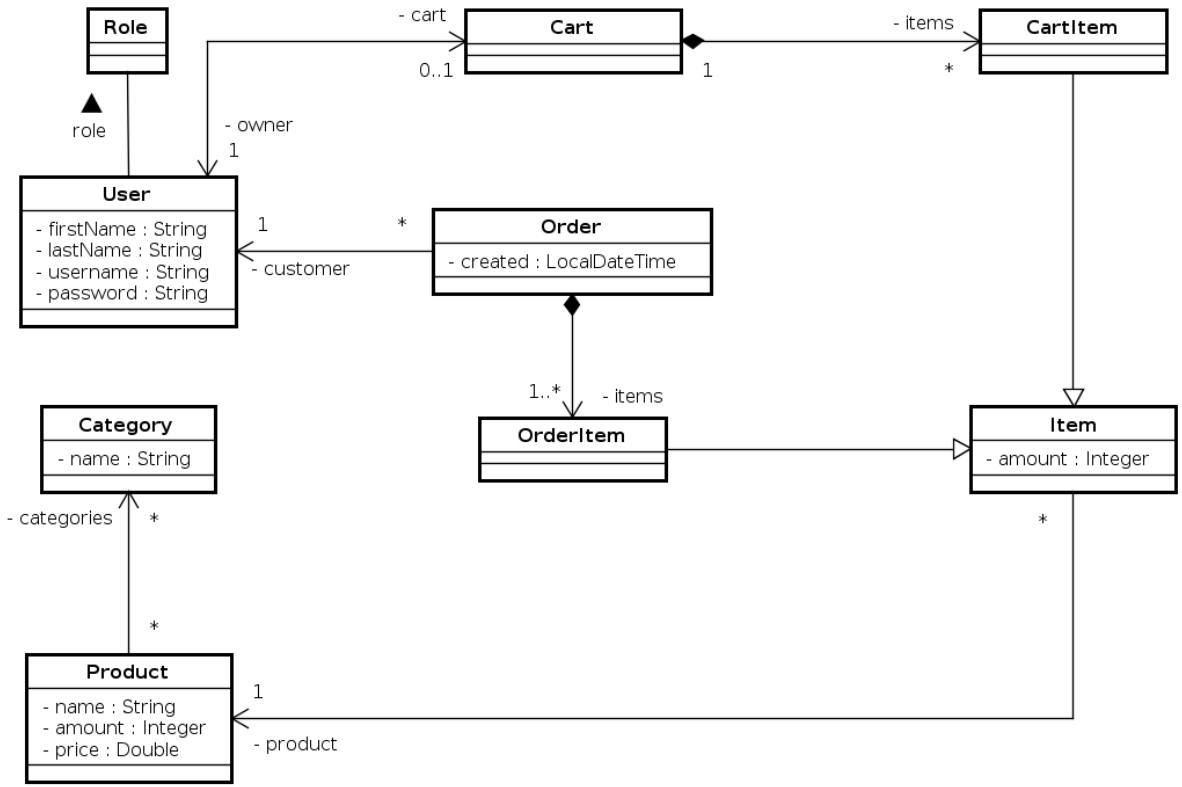
1. Add upstream remote to the local clone of your fork (if you have not done so, yet)
   - `git remote add upstream git@gitlab.fel.cvut.cz:ear/B231-eshop.git`
2. Ensure you have no uncommitted changes in the working tree
   - `git status` → your branch is up to date, nothing to commit
3. Fetch branches and commits from the upstream repository (EAR/B231-eshop)
   - `git fetch upstream`
4. Check out the task branch from **upstream** (one line!)
   - `git checkout -b b231-seminar-03-task upstream/b231-seminar-03-task`
5. Do the task
6. Commit and push the solution to **your** fork
   - `git push -u origin b231-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/b231/_media/courses/b6b36ear/lectures/lecture-03-jpa-s.pdf
- http://www.objectdb.com/api/java/jpa/annotations