

# JPA

Jana Ahmad

[jana.ahmad@fel.cvut.cz](mailto:jana.ahmad@fel.cvut.cz)

Winter Term 2019



# Contents

1 Introduction

2 Tasks

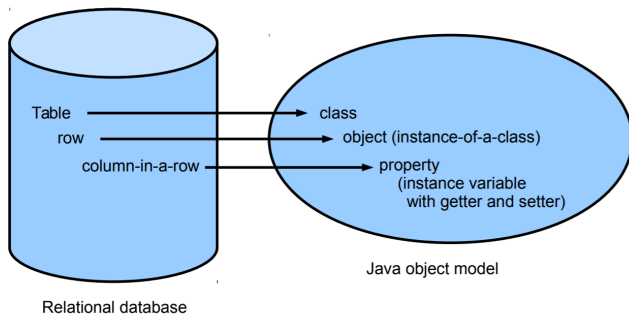


# Introduction



# Object-relational mapping

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



# JPA Basic

- idea: “map whole Java classes to database records”
- a typical system architecture with ORM (Object Relational Model)

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



## JPA – EntityManager

- **EntityManager (EM)** instance is in fact a generic DAO, while entities can be understood as DPO (*managed*) or DTO (*detached*).
- 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]`



# Tasks





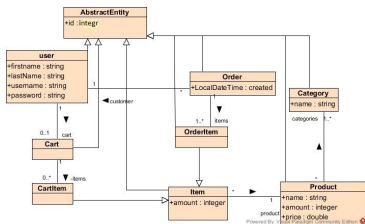
# Syncing Your Fork

- 1 Fetch branches and commits from the upstream repository (EAR/B191-eshop)
  - `git fetch upstream`
- 2 Check out local branch corresponding to the task branch
  - `git checkout -b b191-seminar-03-task`
- 3 Merge changes from the corresponding upstream branch
  - `git merge upstream/b191-seminar-03-task`
- 4 Do your task
- 5 Push the solution to your fork
  - `git push origin b191-seminar-03-task`



# JPA Entities

The package `cz.cvut.kbss.ear.model` contains JPA entities (sketched in the diagram below).



## Task 1: 1 Point

What do you have to fix?

- Some annotations are missing (entities or relationship between them, e.g., `OneToOne`, `ManyToOne`, `setter` and `getter`, etc)
  - E.g., fix the relationships between entities (`Order` and `User`, `Cart` and `CartItem`).
- Some entities are missing altogether
  - E.g., `Category` entity

**Acceptance criteria:** Project is compilable.



## Task 2: 1 Point

In this task you should write a DAO class using a simple JPQL query using Entity Manager (em).

- Create `UserDao` class.
- In `UserDao`, implement `findByUsername` which creates a query to find a user by its name.

**Acceptance criteria:** Project is buildable, i.e., all tests pass.



# The End



The End

Thank You



# Resources

- [https://cw.fel.cvut.cz/b191/\\_media/courses/b6b36ear/lectures/lecture-03-jpa-s.pdf](https://cw.fel.cvut.cz/b191/_media/courses/b6b36ear/lectures/lecture-03-jpa-s.pdf)
- <http://www.objectdb.com/api/java/jpa/annotations>

