

ORM. Hibernate

Užduotis

Sukurkime dar vieną lentelę: **vartotojai**, turinčią laukus:

id, vardas

Lentelėje uzduotys pridėkime vieną papildomą lauką: `vartotojas_id`

Sukurkime dar vieną POJO klasę **Vartotojas** ir sudėkime Hibernate anotacijas

Asociacijos

Objektai gali būti susieti ryšiais:

@ManyToOne

@OneToMany

@OneToOne

@ManyToMany

Taip pat ryšiai gali būti dvikrypčiai arba vienkrypčiai

Plačiau:

https://www.tutorialspoint.com/jpa/jpa_entity_relationships.htm

Užduotis

Task klasėje sukurkime lauką:

```
private Status status;
```

Ir nurodykime anotaciją:

```
@ManyToOne
```

```
@JoinColumn(name="status")
```

Užduotis

Pamėginkime atvaizduoti užduotis ir šalia jų statusus (tekstu iš lentelės statuses)

Užduotis

Klasėje Statuses sukurkime naują atributą **tasks**

```
private Set<Task> tasks=new Set<Tasks>();
```

Nurodykite anotacijas jog šis atributas bus paimamas pagal atgalinį ryšį:

```
@OneToMany(mappedBy="status")
```

Užduotis

Atspausdinkime visus statusus (kokias galime turėti) ir šalia jų visas užduotis

Užduotis

Sukurkime papildomą konstruktorių:

```
public Task(int id, String name, Status status)
```

Išmėginkime sukurti naują statusą, tuomet jam skirtą naują užduotį

Norint išsaugoti įrašus į DB mums reikės įvykdyti du kartus save metodą:

```
session.save(status);
```

```
session.save(task);
```


Įrašų cascade

Tam, kad nereikėtų keletą kartų kviesti save metodo, mes galime nurodyti jog išsaugant vienoje pusėje esantį įrašą būtų išsaugti ir susiję įrašai:

```
@ManyToOne(cascade= {CascadeType.PERSIST})
```

Tuomet atnaujinant šį įrašą bus atnaujintas ir susijęs įrašas.

Jei norėtumėme taip pat galime nurodyti ir tai, jog pašalinus įrašą būtų pašalinti visi susiję įrašai:

```
@ManyToOne(cascade= {CascadeType.PERSIST, CascadeType.REMOVE})
```

Užduotis

Pamėginkime sukurti vartotoją ir užduotį ir juos išsaugoti vien tik saugant užduotį.

Earge ir Lazy Loading

„Lazy loading“ būdu užsikraus pirmiausia užsikraus tik pagrindiniai įrašai (pagrindinėje lentelėje), visi kiti įrašai bus kraunami tik tuomet kai jų paprašysime

„Earge loading“ būdu visi įrašai užsikraus iš karto

Turėtumėme stengtis pagrinde naudoti Lazy Loading, o Earg loading naudoti tik tuomet kai reikia

Užkrovimo būdai pagal nutylėjimą

Mapping	Default Fetch Type
@OneToOne	FetchType.EAGER
@OneToMany	FetchType.LAZY
@ManyToOne	FetchType.EAGER
@ManyToMany	FetchType.LAZY

Užkrovimo tipo nurodymas

Nurodant santykį tarp įrašų, mes galime nurodyti ir užkrovimo būdą:

```
@Entity
@Table(name="instructor")
public class Instructor{
    ...
    @OneToMany(fetch=FetchType.LAZY, mappedBy="instructor")
    private List<Course> courses;
    ...
}
```

Užkrovimo tipo nurodymas

```
@Entity
```

```
@Table(name="courses")
```

```
public class Course{
```

```
    @ManyToOne(fetch=FetchType.LAZY)
```

```
    @JoinColumn(name="instructor_id")
```

```
    private Instructor instructor;
```

```
}
```

Lazy load būdas

Lazy load būdu duomenys užkraunami tik pareikalavus

Norint duomenis paimti, Hibernate sesija privalo būti įjungta

Norėdami jog duomenys būtų užkrauti į kintamuosius Lazy load būdu galėtume:

1. iškvieti `session.get` visiems reikiamiems įrašams
2. iškvieti per Hibernate query (HQL)