Razvoj Softvera

dr.sc. Emir Mešković

X predavanje

Pregled

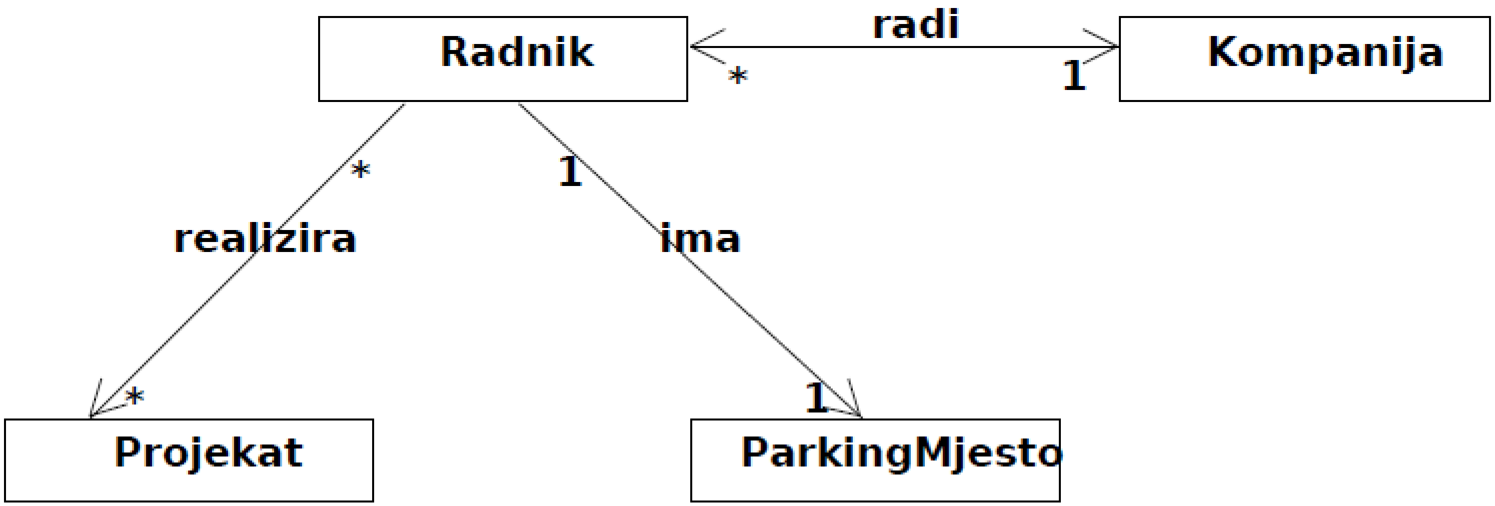
- Mapiranje identifikatora
- ORM preslikavanja
  - One to one
  - Many to one
  - One to many
  - Many to many

Mapiranje identifikatora

- Entiteti se identificiraju putem člana koji je označen @Id anotacijom, i koji se mapira u primarni ključ tabele
- Ako je uključeno id generiranje, JPA vodi računa o unikatnosti svakog kreiranog entiteta unutar PC-a
- Za uključenje ove opcije, anotaciji @Id se dodaje anotacija *@GeneratedValue*
- Polje *strategy* anotacije određuje metodologiju kojom se identifikator generiše, pri čemu postoje četiri vrijednosti za ovo polje:
  - *GenerationType.TABLE* – JPA modificira šemu i dodaje tabelu u kojoj čuva vrijednosti zadnje alociranih vrijednosti za identifikator
  - *GenerationType.SEQUENCE* – DBMS sekvence vode računa o identifikatoru
  - *GenerationType.IDENTITY* – postavlja identity ili auto_increment na PK kolonu
  - *GenerationType.AUTO*

ORM veze

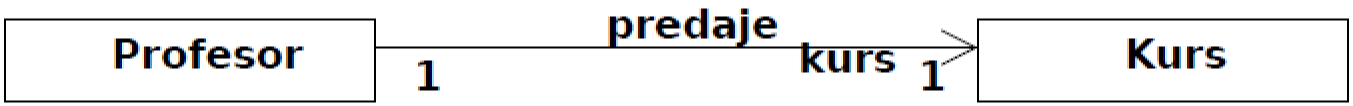
- Dva JPA entiteta mogu biti međusobno različito povezani zavisno od uloga u modelu
- Veze mogu biti jednosmjerne ili dvosmjerne
- Svaka veza ima kardinalnost, tj. broj elemenata koji učestvuje u vezi



ORM preslikavanja

- Nazivi preslikavanja se definišu kardinalnošću svake uloge u vezi
  - Many-to-one
  - One-to-one
  - One-to-many
  - Many-to-many
- Ovo su ujedno i nazivi anotacija koje se koriste za indiciranje tipa veze na atributima koji će biti preslikani
- Asocijacija od jedne instance entiteta ka instanci drugog entiteta (gdje je kardinalnost ciljnog entiteta “one”) se naziva asocijacija putem jedne vrijednosti (*single-valued association*)

One-to-one preslikavanje



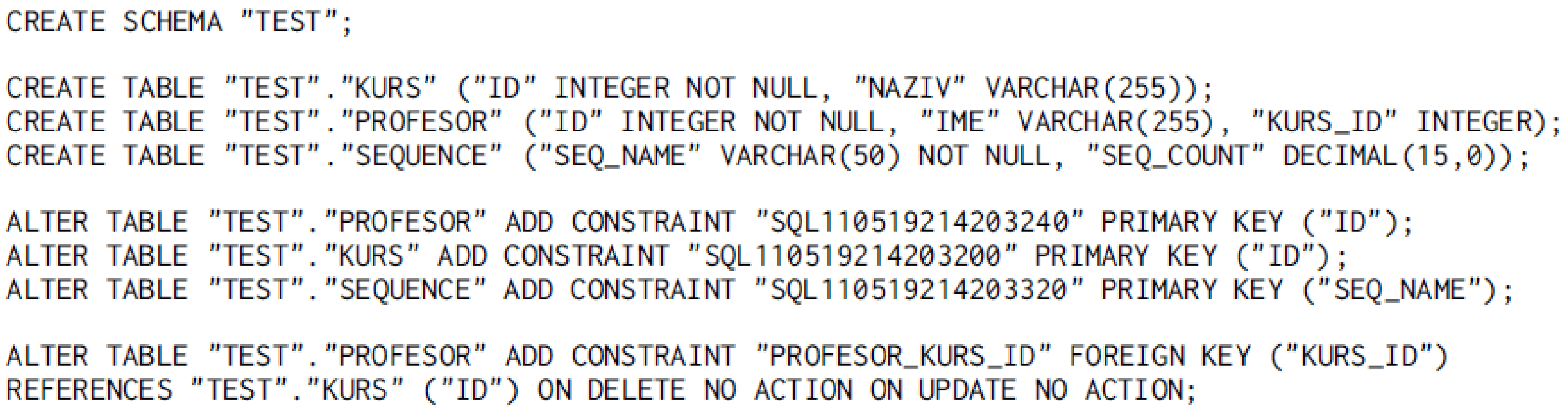
```
package model;
import javax.persistence.*;

@Entity
public class Profesor {
    @Id
    @GeneratedValue(strategy=GenerationType.TABLE)
    private int id;
    private String ime;
    @OneToOne
    private Kurs kurs;

    public Profesor() {}
    public void setKurs(Kurs kurs) {this.kurs=kurs;}
    public Kurs getKurs() {return kurs;}
    public int getId() {return id;}
    public String getIme() {return ime;}
    public void setIme(String name) {this.ime=name;}
    @Override
    public String toString() {
       return getId()+" "+getIme();
    }
}
```

```
package model;
import javax.persistence.*;

@Entity
public class Kurs {
    @Id
    @TableGenerator(name="Kurs_Gen")
    @GeneratedValue(generator="Kurs_Gen")
    private int id;
    private String naziv;

    public Kurs() {}
    public int getId() {return id;}
    public String getNaziv() {return naziv;}
    public void setNaziv(String naziv)
    {this.naziv=naziv;}
    @Override
    public String toString() {
       return getId()+" "+getNaziv();
    }
}
```

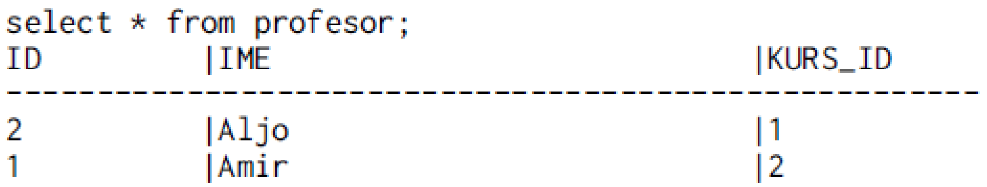
Korištenje one-to-one preslikavanja

```
import java.util.List;
import javax.persistence.*;
import model.*;

public class Main {
    private static final String PERSISTENCE_UNIT_NAME = "TestPU";
    private static EntityManagerFactory factory;

    public static void main(String[] args) {
       factory = Persistence.createEntityManagerFactory(PERSISTENCE_UNIT_NAME);
       EntityManager em = factory.createEntityManager();

       Profesor amir = new Profesor();
       amir.setIme("Amir");
       Kurs modeliranje = new Kurs();
       modeliranje.setNaziv("Modeliranje i simulacije");
       amir.setKurs(modeliranje);

       Profesor aljo = new Profesor();
       aljo.setIme("Aljo");
       Kurs elektronika= new Kurs();
       elektronika.setNaziv("Elektronika");
       aljo.setKurs(elektronika);

       em.getTransaction().begin();
       em.persist(amir);
       em.persist(aljo);
       em.persist(elektronika);
       em.persist(modeliranje);
       em.getTransaction().commit();
       em.close();
       factory.close();
    }
```

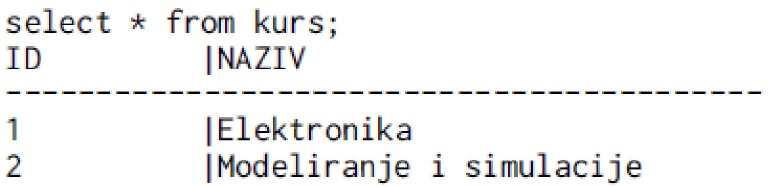
One-to-one pogled na bazu



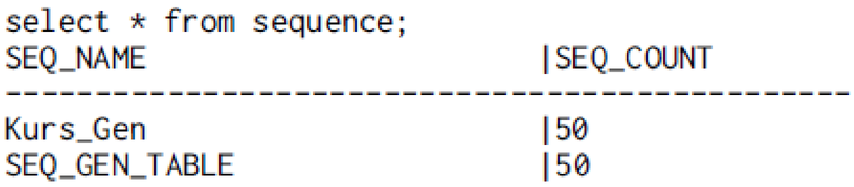
```
CREATE SCHEMA "TEST";

CREATE TABLE "TEST"."KURS" ("ID" INTEGER NOT NULL, "NAZIV" VARCHAR(255));
CREATE TABLE "TEST"."PROFESOR" ("ID" INTEGER NOT NULL, "IME" VARCHAR(255), "KURS_ID" INTEGER);
CREATE TABLE "TEST"."SEQUENCE" ("SEQ_NAME" VARCHAR(50) NOT NULL, "SEQ_COUNT" DECIMAL(15,0));

ALTER TABLE "TEST"."PROFESOR" ADD CONSTRAINT "SQL110519214203240" PRIMARY KEY ("ID");
ALTER TABLE "TEST"."KURS" ADD CONSTRAINT "SQL110519214203200" PRIMARY KEY ("ID");
ALTER TABLE "TEST"."SEQUENCE" ADD CONSTRAINT "SQL110519214203320" PRIMARY KEY ("SEQ_NAME");

ALTER TABLE "TEST"."PROFESOR" ADD CONSTRAINT "PROFESOR_KURS_ID" FOREIGN KEY ("KURS_ID")
REFERENCES "TEST"."KURS" ("ID") ON DELETE NO ACTION ON UPDATE NO ACTION;
```



```
select * from profesor;
ID        |IME                        |KURS_ID
-----------------------------------------------
2         |Aljo                       |1
1         |Amir                       |2
```

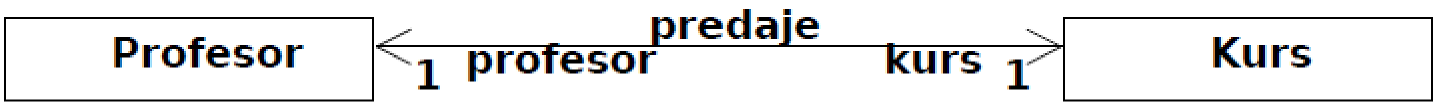


```
select * from kurs;
ID        |NAZIV
----------------------------------------
1         |Elektronika
2         |Modeliranje i simulacije
```



```
select * from sequence;
SEQ_NAME                  |SEQ_COUNT
----------------------------------------
Kurs_Gen                  |50
SEQ_GEN_TABLE             |50
```

Bidirekcioni one-to-one

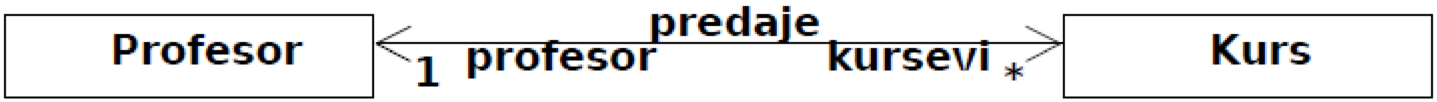


```
package model;
import javax.persistence.*;

@Entity
public class Profesor {
     @Id
     @GeneratedValue(strategy=GenerationType.TABLE)
     private int id;
     private String ime;
     @OneToOne(cascade={CascadeType.PERSIST,CascadeType.REMOVE})
     private Kurs kurs;

     public Profesor() {}
     public void setKurs(Kurs kurs) {this.kurs=kurs;}
     public Kurs getKurs() {return kurs;}
     public int getId() {return id;}
     public String getIme() {return ime;}
     public void setIme(String name) {this.ime=name;}
}
```

```
package model;
import javax.persistence.*;

@Entity
public class Kurs {
     @Id
     @TableGenerator(name="Kurs_Gen")
     @GeneratedValue(generator="Kurs_Gen")
     private int id;
     private String naziv;
     @OneToOne(mappedBy="kurs")
     private Profesor profesor;

     public Kurs() {}
     public int getId() {return id;}
     public String getNaziv() {return naziv;}
     public Profesor getProfesor() {return profesor;}
     public void setProfesor(Profesor prof)
     {this.profesor=prof;}
     public void setNaziv(String naziv) {this.naziv=naziv;}
}
```

```
EntityManager em = factory.createEntityManager();
Profesor amir = new Profesor();
amir.setIme("Amir");
Kurs modeliranje = new Kurs();
modeliranje.setNaziv("Modeliranje i simulacije");
modeliranje.setProfesor(amir);
amir.setKurs(modeliranje);
em.getTransaction().begin();
em.persist(amir);
em.getTransaction().commit();
em.close();
```

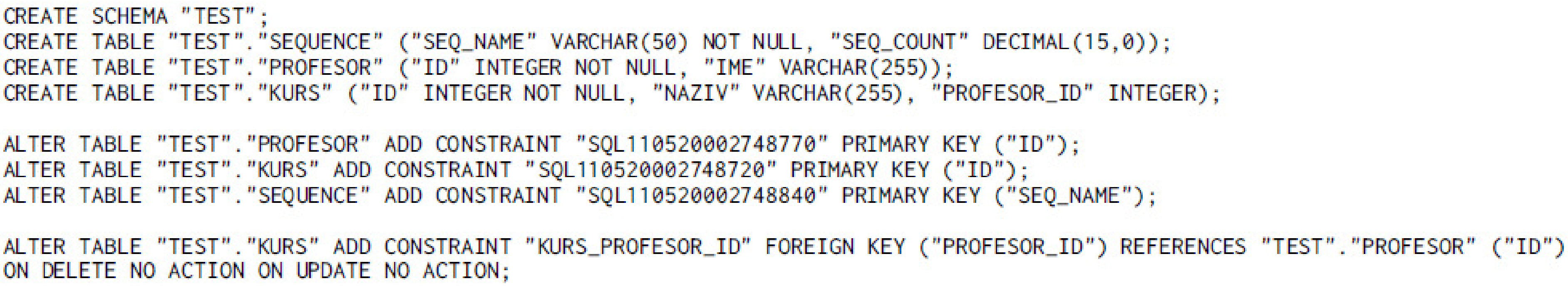
One-to-many i many-to-one preslikavanje



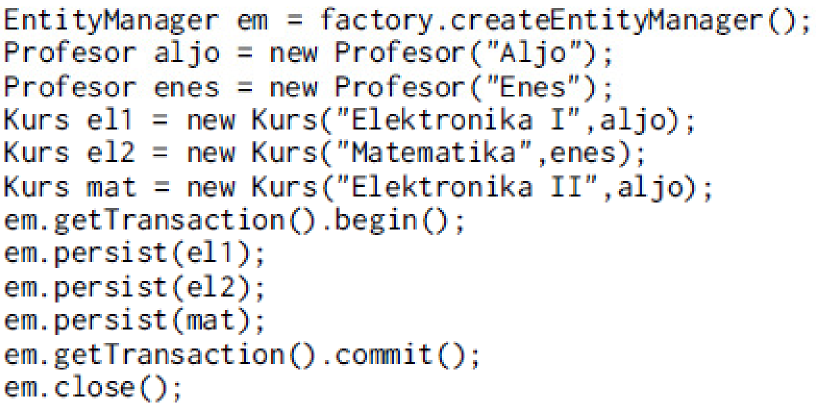
```
package model;
import javax.persistence.*;
import java.util.Collection;

@Entity
public class Profesor {
    @Id
    @GeneratedValue(strategy=GenerationType.TABLE)
    private int id;
    private String ime;
    @OneToMany(mappedBy="profesor")
    private Collection<Kurs> kursevi;

    public Profesor() {}
    public Profesor(String i) {ime=i;}

    public void setKursevi(Collection<Kurs> k) {
       kursevi=k;
    }
    public Collection<Kurs> getKursevi() {
       return kursevi;
    }
    public int getId() {return id;}
    public String getIme() {return ime;}
    public void setIme(String i) {ime=i;}
}
```

```
package model;
import javax.persistence.*;

@Entity
public class Kurs {
    @Id
    @TableGenerator(name="Kurs_Gen")
    @GeneratedValue(generator="Kurs_Gen")
    private int id;
    private String naziv;
    @ManyToOne(cascade=CascadeType.PERSIST)
    private Profesor profesor;

    public Kurs() {}
    public Kurs(String n, Profesor p) {
       profesor = p;
       naziv = n;
    }
    public int getId() {return id;}
    public Profesor getProfesor() {return profesor;}
    public String getNaziv() {return naziv;}
    public void setNaziv(String naziv)
    {this.naziv=naziv;}
}
```

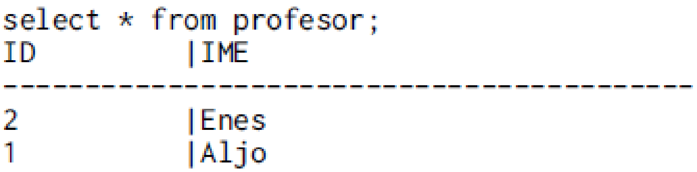
One-to-many korištenje i baza



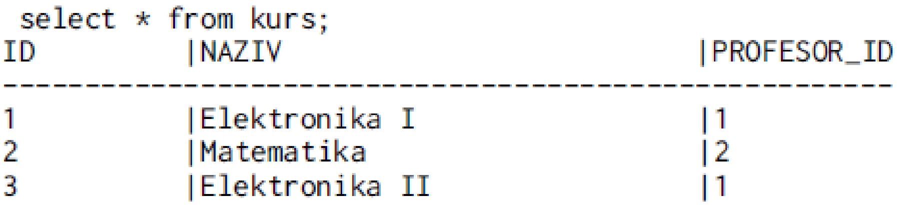
```
CREATE SCHEMA "TEST";
CREATE TABLE "TEST"."SEQUENCE" ("SEQ_NAME" VARCHAR(50) NOT NULL, "SEQ_COUNT" DECIMAL(15,0));
CREATE TABLE "TEST"."PROFESOR" ("ID" INTEGER NOT NULL, "IME" VARCHAR(255));
CREATE TABLE "TEST"."KURS" ("ID" INTEGER NOT NULL, "NAZIV" VARCHAR(255), "PROFESOR_ID" INTEGER);

ALTER TABLE "TEST"."PROFESOR" ADD CONSTRAINT "SQL110520002748770" PRIMARY KEY ("ID");
ALTER TABLE "TEST"."KURS" ADD CONSTRAINT "SQL110520002748720" PRIMARY KEY ("ID");
ALTER TABLE "TEST"."SEQUENCE" ADD CONSTRAINT "SQL110520002748840" PRIMARY KEY ("SEQ_NAME");

ALTER TABLE "TEST"."KURS" ADD CONSTRAINT "KURS_PROFESOR_ID" FOREIGN KEY ("PROFESOR_ID") REFERENCES "TEST"."PROFESOR" ("ID")
ON DELETE NO ACTION ON UPDATE NO ACTION;
```



```
EntityManager em = factory.createEntityManager();
Profesor aljo = new Profesor("Aljo");
Profesor enes = new Profesor("Enes");
Kurs el1 = new Kurs("Elektronika I",aljo);
Kurs el2 = new Kurs("Matematika",enes);
Kurs mat = new Kurs("Elektronika II",aljo);
em.getTransaction().begin();
em.persist(el1);
em.persist(el2);
em.persist(mat);
em.getTransaction().commit();
em.close();
```

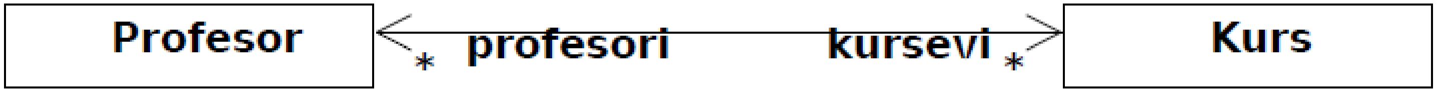


```
select * from profesor;
ID          |IME
------------------------------------------
2           |Enes
1           |Aljo
```



```
 select * from kurs;
ID          |NAZIV                           |PROFESOR_ID
-----------------------------------------------------------
1           |Elektronika I                   |1
2           |Matematika                      |2
3           |Elektronika II                  |1
```

Many-to-many preslikavanje



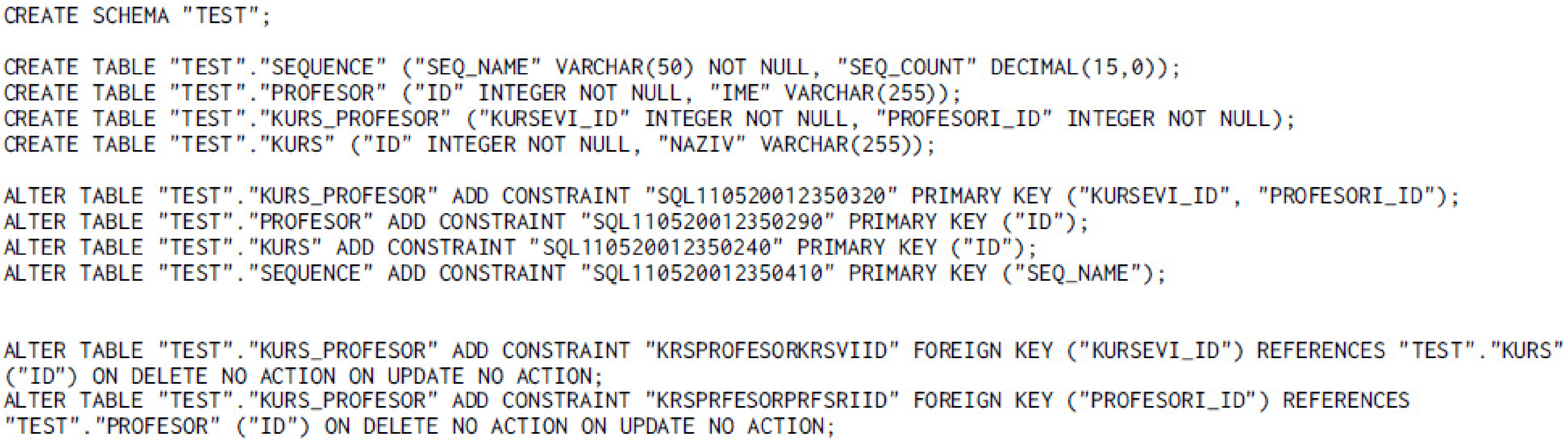
```
package model;
import javax.persistence.*;
import java.util.Collection;

@Entity
public class Profesor {
    @Id
    @GeneratedValue(strategy=GenerationType.TABLE)
    private int id;
    private String ime;
    @ManyToMany(mappedBy="profesori")
    private Collection<Kurs> kursevi;

    public Profesor() {}
    public Profesor(String i) {ime=i;}
    public void setKursevi(Collection<Kurs> k) {
       kursevi=k;
    }
    public Collection<Kurs> getKursevi() {
       return kursevi;
    }
    public int getId() {return id;}
    public String getIme() {return ime;}
    public void setIme(String i) {ime=i;}
}
```

```
package model;
import java.util.Collection;
import java.util.ArrayList;
import javax.persistence.*;
@Entity
public class Kurs {
    @Id
    @TableGenerator(name="Kurs_Gen")
    @GeneratedValue(generator="Kurs_Gen")
    private int id;
    private String naziv;
    @ManyToMany(cascade=CascadeType.PERSIST)
    private Collection<Profesor> profesori;
    public Kurs() {}
    public Kurs(String naziv ) {
       this.naziv = naziv;
    }
    public int getId() {return id;}
    public Collection<Profesor> getProfesori() {
       if (profesori == null) {
           profesori = new ArrayList<Profesor>();
       }
       return profesori;
    }
    public void setProfesori(Collection<Profesor> p) {
       this.profesori = p;
    }
    public String getNaziv() {return naziv;}
}
```

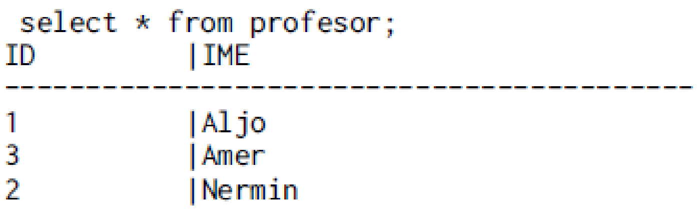
Korištenje many-to-many preslikavanja

```
import javax.persistence.*;
import model.*;

public class Main {
    private static final String PERSISTENCE_UNIT_NAME = "TestPU";
    private static EntityManagerFactory factory;

    public static void main(String[] args) {
       factory = Persistence.createEntityManagerFactory(PERSISTENCE_UNIT_NAME);
       EntityManager em = factory.createEntityManager();
       Profesor aljo = new Profesor("Aljo");
       Profesor amer = new Profesor("Amer");
       Profesor nermin = new Profesor("Nermin");

       Kurs mreze = new Kurs("Mreze");
       Kurs signali = new Kurs("Signali i sistemi");
       mreze.getProfesori().add(aljo);
       mreze.getProfesori().add(amer);
       mreze.getProfesori().add(nermin);
       signali.getProfesori().add(aljo);
       signali.getProfesori().add(nermin);

       em.getTransaction().begin();
       em.persist(signali);
       em.persist(mreze);
       em.getTransaction().commit();
       em.close();
       factory.close();
    }
}
```

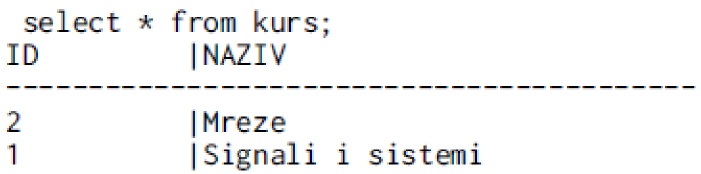
One-to-many korištenje i baza



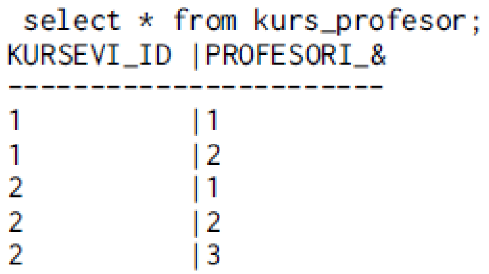
```
CREATE SCHEMA "TEST";

CREATE TABLE "TEST"."SEQUENCE" ("SEQ_NAME" VARCHAR(50) NOT NULL, "SEQ_COUNT" DECIMAL(15,0));
CREATE TABLE "TEST"."PROFESOR" ("ID" INTEGER NOT NULL, "IME" VARCHAR(255));
CREATE TABLE "TEST"."KURS_PROFESOR" ("KURSEVI_ID" INTEGER NOT NULL, "PROFESORI_ID" INTEGER NOT NULL);
CREATE TABLE "TEST"."KURS" ("ID" INTEGER NOT NULL, "NAZIV" VARCHAR(255));

ALTER TABLE "TEST"."KURS_PROFESOR" ADD CONSTRAINT "SQL110520012350320" PRIMARY KEY ("KURSEVI_ID", "PROFESORI_ID");
ALTER TABLE "TEST"."PROFESOR" ADD CONSTRAINT "SQL110520012350290" PRIMARY KEY ("ID");
ALTER TABLE "TEST"."KURS" ADD CONSTRAINT "SQL110520012350240" PRIMARY KEY ("ID");
ALTER TABLE "TEST"."SEQUENCE" ADD CONSTRAINT "SQL110520012350410" PRIMARY KEY ("SEQ_NAME");


ALTER TABLE "TEST"."KURS_PROFESOR" ADD CONSTRAINT "KRSPROFESORKRSVIID" FOREIGN KEY ("KURSEVI_ID") REFERENCES "TEST"."KURS"
("ID") ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE "TEST"."KURS_PROFESOR" ADD CONSTRAINT "KRSPRFESORPRFSRIID" FOREIGN KEY ("PROFESORI_ID") REFERENCES
"TEST"."PROFESOR" ("ID") ON DELETE NO ACTION ON UPDATE NO ACTION;
```



```
 select * from profesor;
ID          |IME
-------------------------------------------
1           |Aljo
3           |Amer
2           |Nermin
```



```
 select * from kurs;
ID          |NAZIV
-------------------------------------------
2           |Mreze
1           |Signali i sistemi
```



```
 select * from kurs_profesor;
KURSEVI_ID |PROFESORI_&
-----------------------
1           |1
1           |2
2           |1
2           |2
2           |3
```