

Baze podataka

Predavanja

12. ER model baze podataka – 2. dio

Svibanj, 2021.



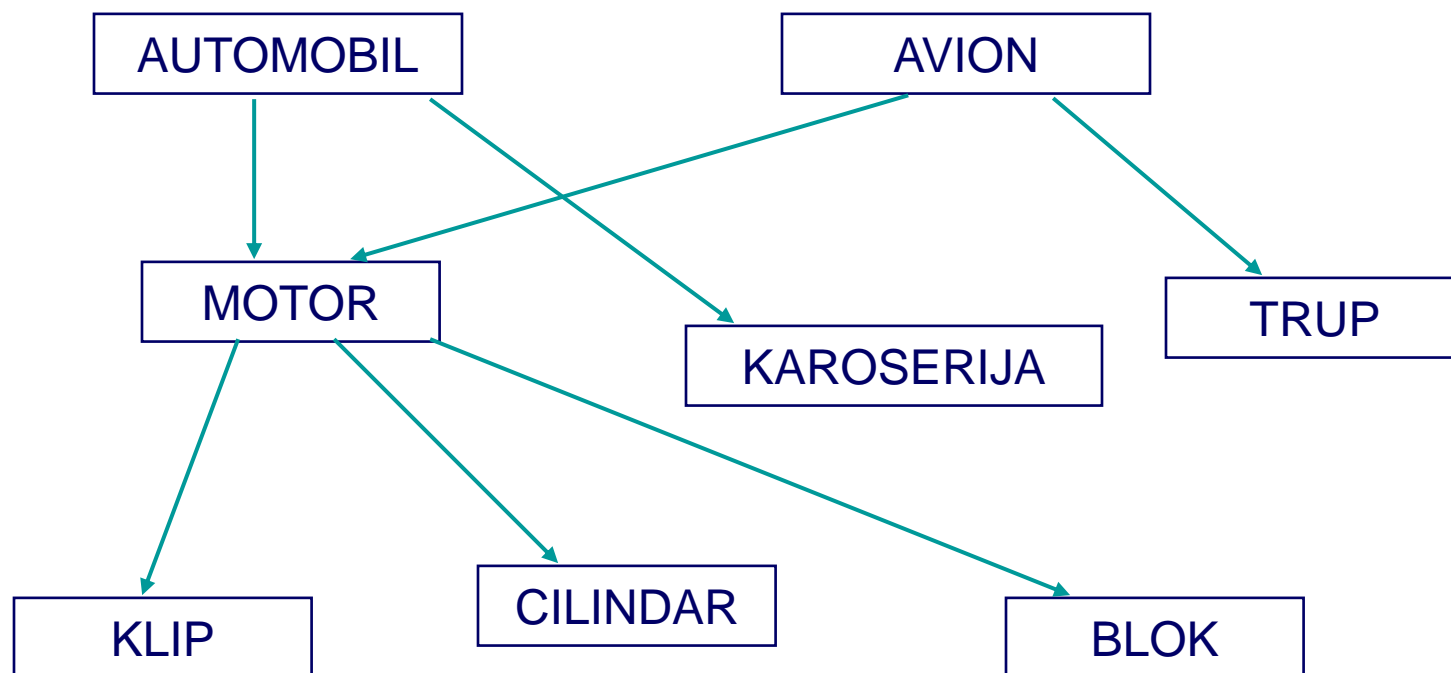
Homogena mreža

Primjer: Sastavnica

AUTOMOBIL: MOTOR, KAROSERIJA

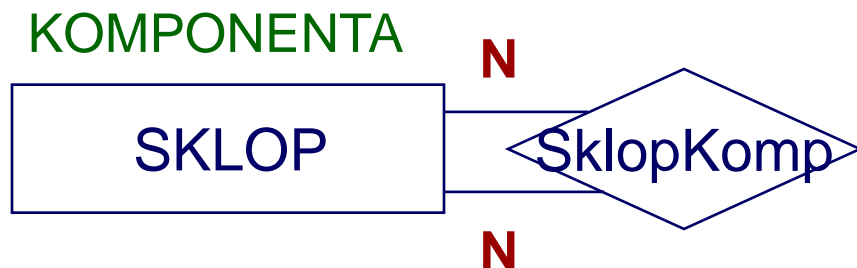
MOTOR: KLIP, CILINDAR, BLOK

AVION: MOTOR, TRUP



Čvorovi u mreži imaju jednaku strukturu: SKLOP= sifSklop, nazSklop

Refleksivne veze - preslikavanje N:N



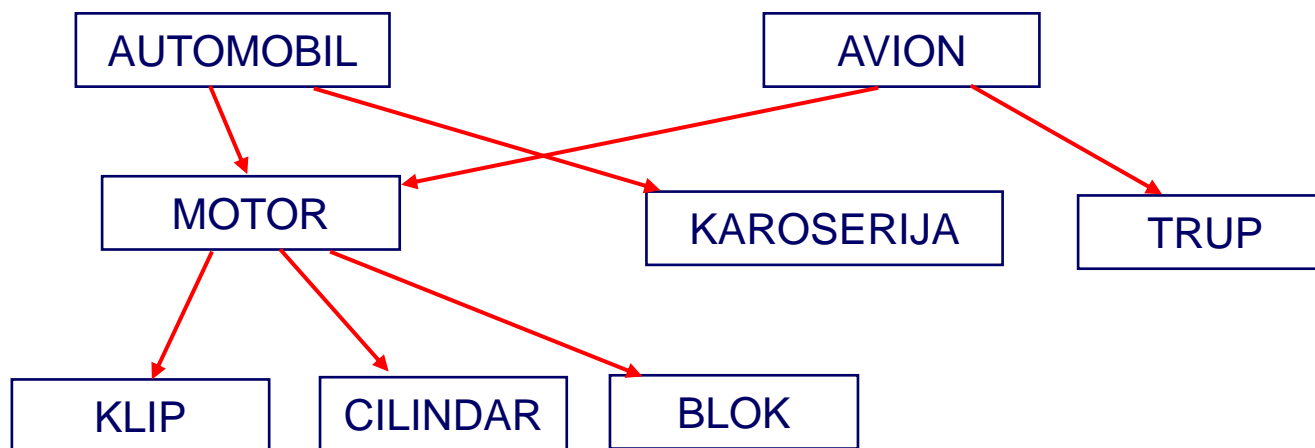
SKLOP = sifSklop, nazSklop

SklopKomp = ~~sifSklop, sifSklop~~

SklopKomp = sifSklop, sifKomp

Preimenovati
jedan od atributa !

Refleksivne veze - preslikavanje N:N



Sklop

sifSklop	nazSklop
17	Automobil
19	Motor
21	Karoserija
37	Klip
49	Cilindar
52	Blok
64	Avion
82	Trup

sklopKomp

sifSklop	sifKomp
17	19
17	21
19	37
19	49
19	52
64	19
64	82

Refleksivne veze N:N → relacijski model

SKLOP = sifSklop, nazSklop

SklopKomp = sifSklop, sifKomp

+ pravila integriteta

Zadatak: Ispisati naziv sklopa i naziv komponenti od kojih se sklop sastoji (ukoliko komponente sklopa postoje)

```
SELECT sklop.nazSklop
      , komponenta.nazSklop AS nazKomponenta
FROM sklop AS komponenta
     INNER JOIN sklopKomp
       ON komponenta.sifSklop = sklopKomp.sifKomp
     RIGHT OUTER JOIN sklop
       ON sklopKomp.sifSklop = sklop.sifSklop;
```

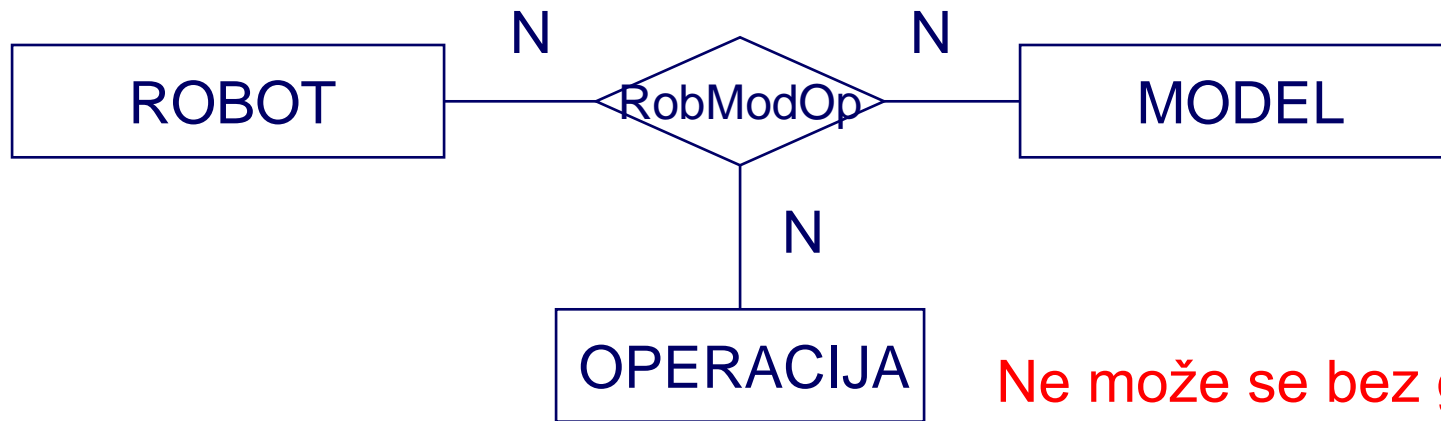
Problem

Model proizvodnje:

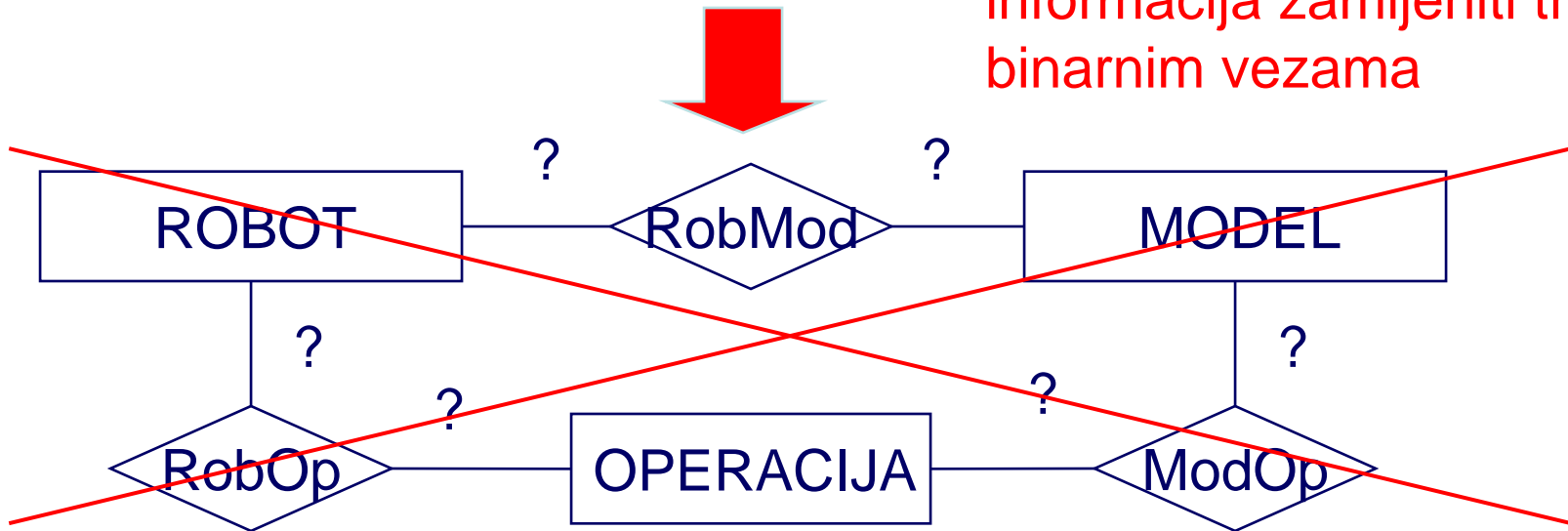
- robot R1 montira prednja lijeva vrata na modelu automobila Volvo S40 za 45 sekundi i pri tome utroši 0.8 kWh energije
- robot R2 oboji poklopac motora na modelu automobila Volvo S40 za 28 sekundi i pri tome utroši 0.4 kWh energije
- robot R1 montira prednja lijeva vrata na modelu automobila Volvo S60 za 52 sekunde i pri tome utroši 0.9 kWh energije
- robot R1 montira poklopac motora na modelu automobila Volvo S40 za 25 sekundi i pri tome utroši 0.75 kWh energije
- robot R2 montira prednja lijeva vrata na modelu automobila Volvo S40 za 40 sekundi i pri tome utroši 0.6 kWh energije
- robot R2 montira poklopac motora na modelu automobila Volvo S40 za 18 sekundi i pri tome utroši 0.7 kWh energije

Ternarne veze

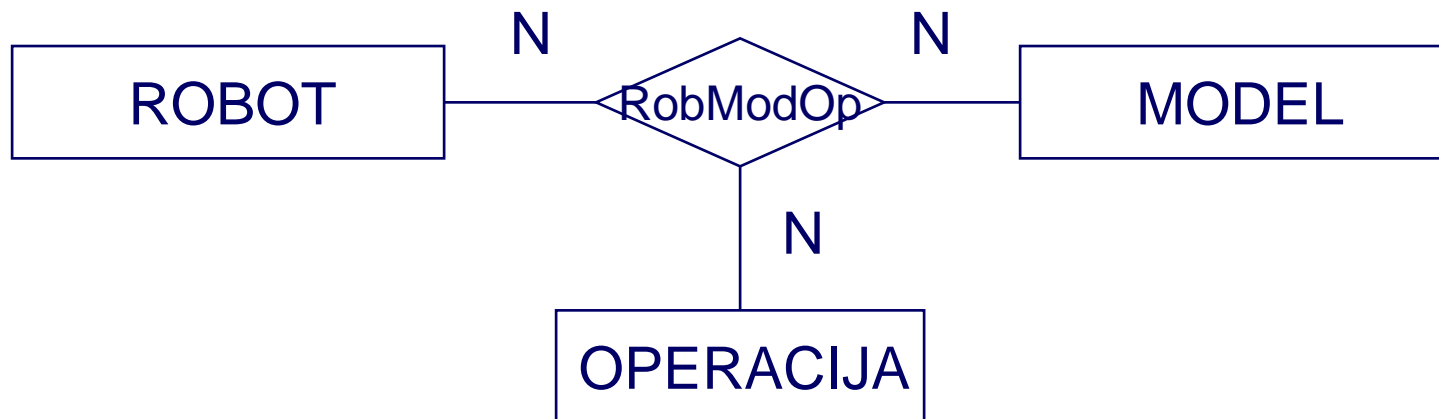
Ternarnom vezom prikazuje se istovremeni odnos triju entiteta.



Ne može se bez gubitaka informacija zamijeniti trima binarnim vezama



Ternarne veze - preslikavanje N:N:N



ROBOT = sifRobot, nazRobot, ...

MODEL = sifModel, nazModel, ...

OPERACIJA = sifOper, nazOper, ...

RobModOp = sifRobot, sifModel, sifOper, utrVrijeme, utrEnergija

Ternarne veze N:N:N → relacijski model

ROBOT = sifRobot, nazRobot, ...

MODEL = sifModel, nazModel, ...

OPERACIJA = sifOper, nazOper, ...

+ pravila integriteta

RobModOp = sifRobot, sifModel, sifOper, utrVrijeme, utrEnergija

Zadatak: Ispisati naziv robota, naziv modela automobila, naziv operacije, utrošak vremena i energije, za sve operacije koje roboti mogu obaviti

```
SELECT nazRobot, nazModel, nazOper, utrVrijeme, utrEnergija
FROM robModOp
  INNER JOIN robot
    ON robModOp.sifRobot = robot.sifRobot
  INNER JOIN model
    ON robModOp.sifModel = model.sifModel
  INNER JOIN operacija
    ON robModOp.sifOper = operacija.sifOper;
```

Definicija 1. (Teorey)

U vezi koja povezuje entitete

$E_1, \dots, E_k, \dots, E_m$,

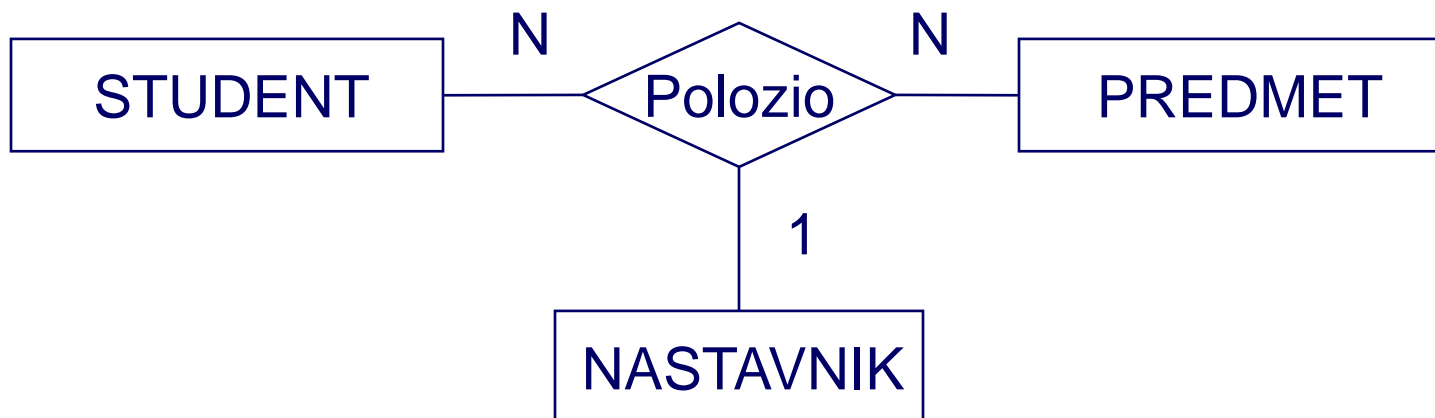
spojnost =1 entiteta E_k znači da za svaku vrijednost svih entiteta E_1, \dots, E_m , osim E_k , uvijek postoji točno jedna vrijednost od E_k .

□ može se reći da tada vrijedi funkcijska zavisnost:

$$\bigcup_{j=1}^m K_j \setminus K_k \rightarrow K_k$$

gdje su skupovi K_j , ($j = 1, \dots, m$) ključevi entiteta E_1, \dots, E_m

Ternarne veze - preslikavanje N:N:1



STUDENT = matBrSt, prezSt, imeSt

PREDMET = sifPred, nazPred

NASTAVNIK = sifNast, prezNast, imeNast

Polozio = matBrSt, sifPred, sifNast, ocjena

Ternarne veze N:N:1 → relacijski model

STUDENT = matBrSt, prezSt, imeSt

PREDMET = sifPred, nazPred

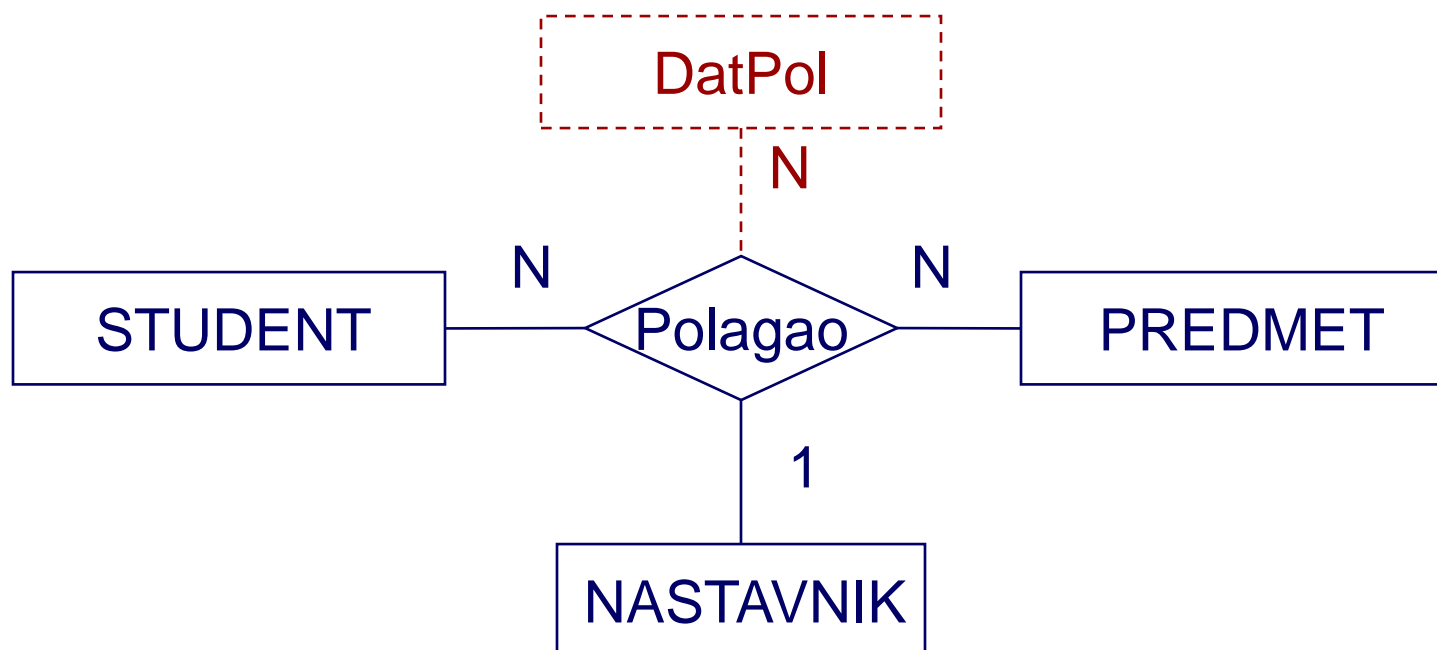
NASTAVNIK = sifNast, prezNast, imeNast

Položio = matBrSt, sifPred, sifNast, ocjena

+ pravila integriteta

```
SELECT prezSt, imeSt, nazPred, prezNast, imeNast, ocjena
FROM položio
  INNER JOIN student
    ON položio.matBrSt = student.matBrSt
  INNER JOIN predmet
    ON položio.sifPred = predmet.sifPred
  INNER JOIN nastavnik
    ON položio.sifNast = nastavnik.sifNast;
```

Ternarne veze - preslikavanje N:N:1



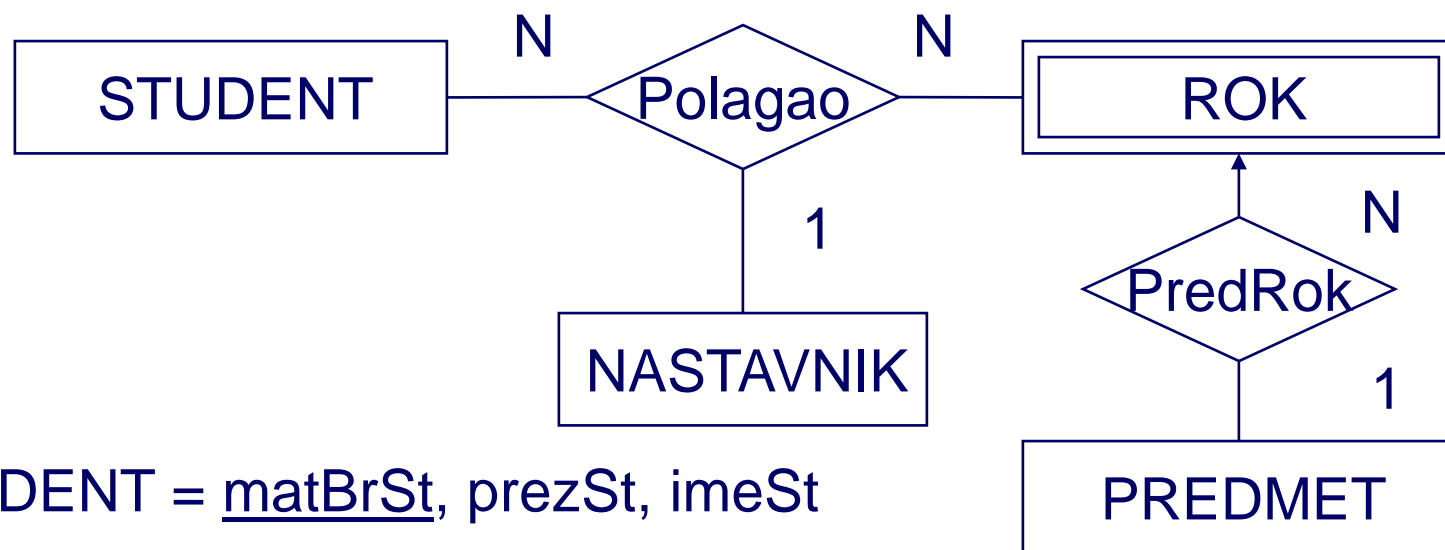
STUDENT = matBrSt, prezSt, imeSt

PREDMET = sifPred, nazPred

NASTAVNIK = sifNast, prezNast, imeNast

Polagao = matBrSt, sifPred, datPol, sifNast, ocjena

Ternarne veze - preslikavanje N:N:1



STUDENT = matBrSt, prezSt, imeSt

PREDMET = sifPred, nazPred

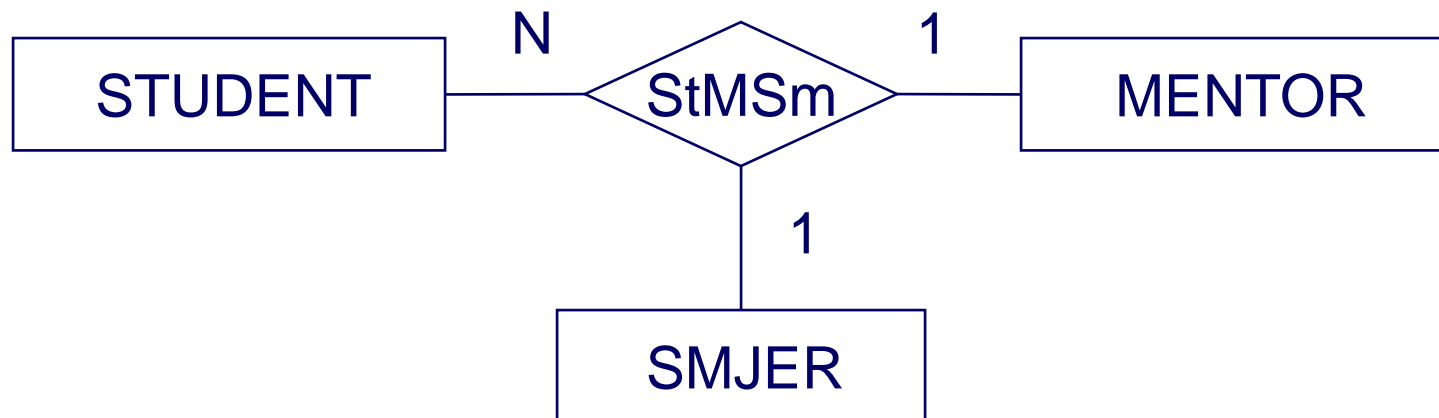
ROK = sifPred, datRok, vrstaRok

NASTAVNIK = sifNast, prezNast, imeNast

PredRok = sifPred, datRok

Polagao = matBrSt, sifPred, datRok, sifNast, ocjena

Ternarne veze - preslikavanje N:1:1



Student može studirati na više smjerova, ali na svakom smjeru mora imati različitog mentora. Student na svakom smjeru ima samo jednog mentora.

STUDENT = matBrSt, prezSt, imeSt

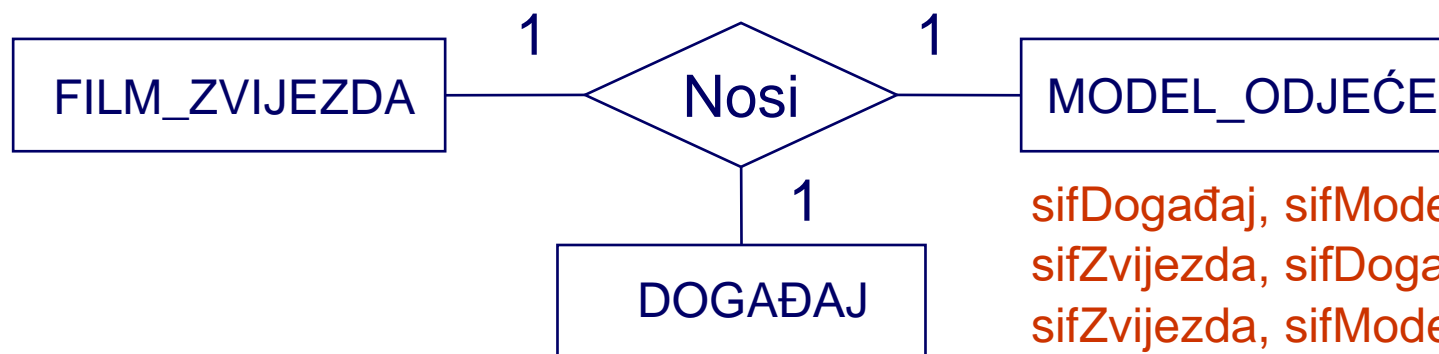
SMJER = sifSmjer, nazSmjer

MENTOR = sifMentor, prezMentor, imeMentor

StMSm = sifSmjer, matBrSt, sifMentor

Ternarne veze - preslikavanje 1:1:1

Na jednom događaju (npr. Dodjela Oscara 2017.) ne smiju dvije ili više filmskih zvijezda nositi isti model odjeće. Tijekom jednog događaja, jedna zvijezda nosi samo jedan model odjeće. Jedna zvijezda smije jedan model odjeće nositi na samo jednom događaju.



sifDogađaj, sifModel → sifZvijezda
sifZvijezda, sifDogađaj → sifModel
sifZvijezda, sifModel → sifDogađaj

FILM_ZVIJEZDA = sifZvijezda, ime, prezime

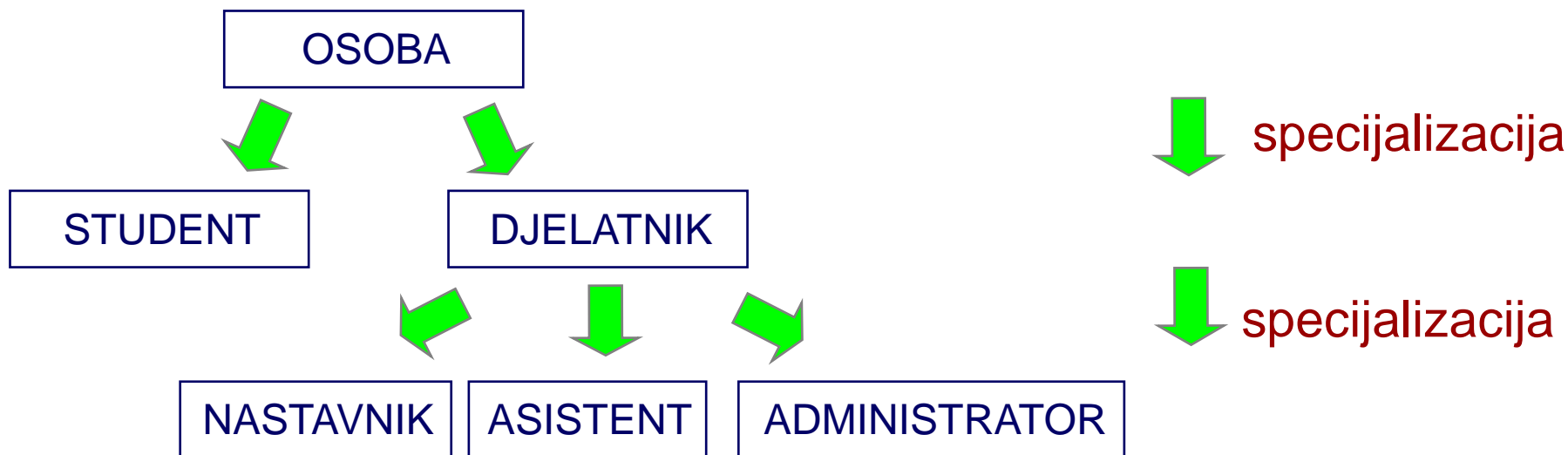
MODEL_ODJEĆE = sifModel, nazModel

DOGAĐAJ = sifDogađaj, nazDogađaj, datumDogađaj

Nosi = sifZvijezda, sifDogađaj, sifModel

Specijalizacija

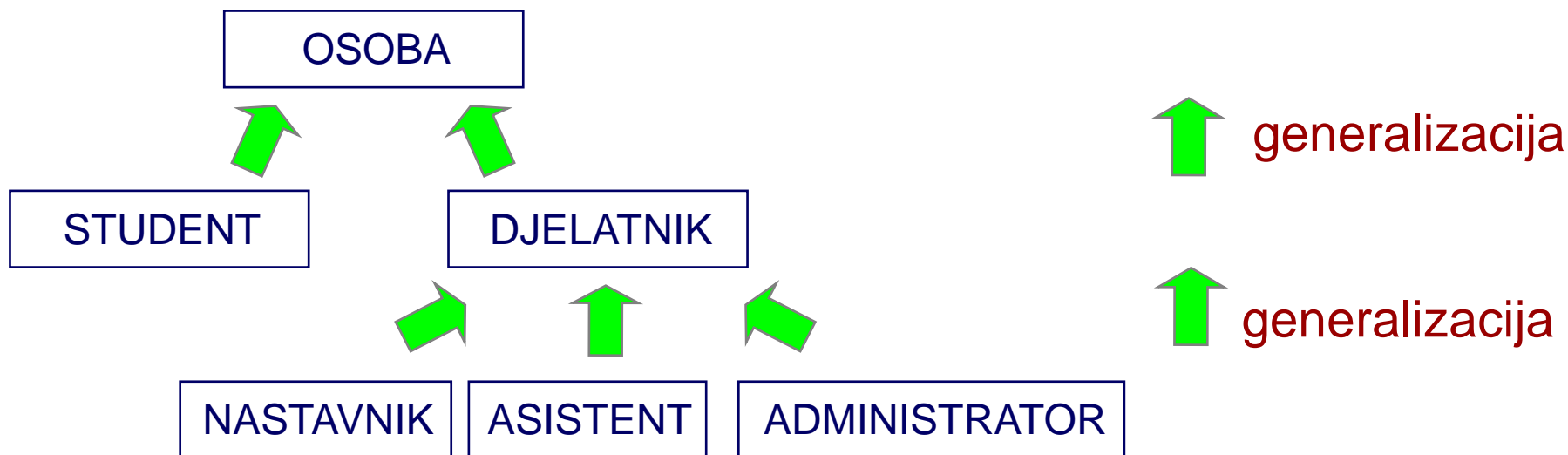
Entiteti jednog skupa entiteta mogu se temeljem njihovih karakterističnih svojstava klasificirati u zasebne skupove entiteta, postupkom koji se naziva **specijalizacija**



Skupovi entiteta dobiveni postupkom **specijalizacije** nazivaju se podklase (*subclasses*) ili specijalizacije

Generalizacija

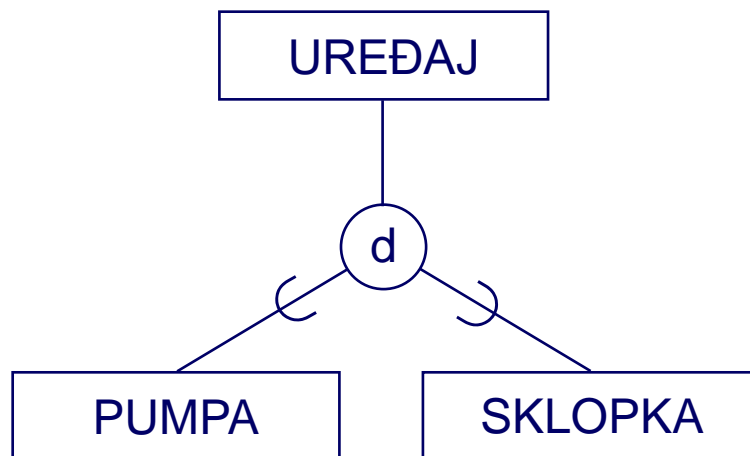
Entiteti iz nekoliko skupova entiteta sa sličnim svojstvima mogu se grupirati u zajednički skup entiteta, postupkom koji se naziva **generalizacija**



Skupovi entiteta dobiveni postupkom **generalizacije** nazivaju se nadklase (*superclasses*) ili generalizacije

Postupak generalizacije je inverzan postupku specijalizacije

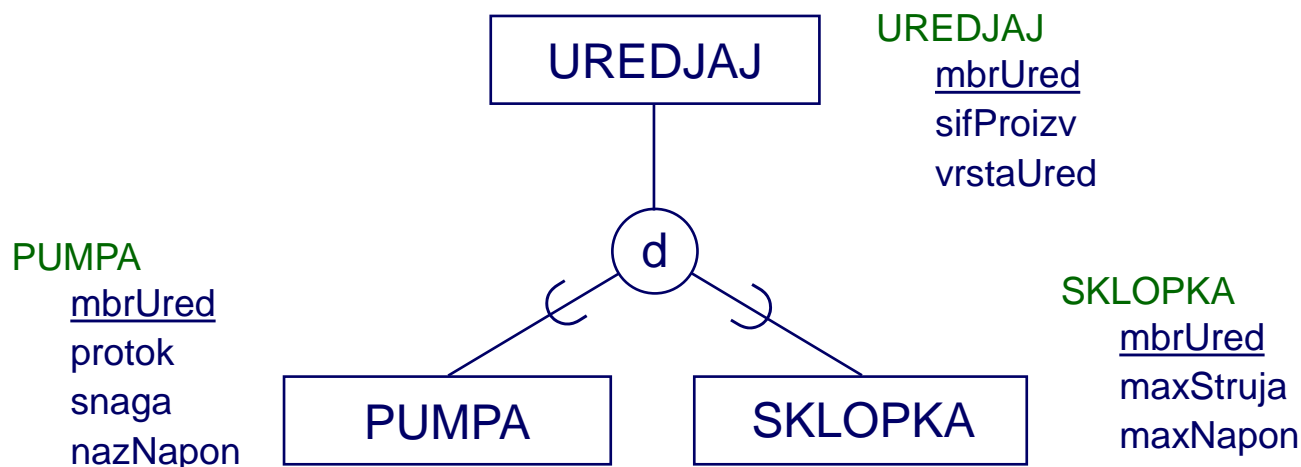
Generalizacija i specijalizacija - ER dijagram



- u kružnicu se upisuje slovo **d** (*disjoint*) ukoliko se radi o ekskluzivnoj generalizaciji/specijalizaciji
 - uređaj može biti **ili** pumpa **ili** sklopka (ekskluzivni ili)
- pomoću vrijednosti atributa *vrstaUred* (npr. 'p' ili 's') može se odrediti podklasa (specijalizacija) kojoj entitet pripada

Preslikavanje u relacijski model

- specijalizacije nemaju vlastite ključeve



UREDJAJ = mbrUred, sifProizv, vrstaUred

PUMPA = mbrUred, protok, snaga, nazNapon

SKLOPKA = mbrUred, maxStruja, maxNapon

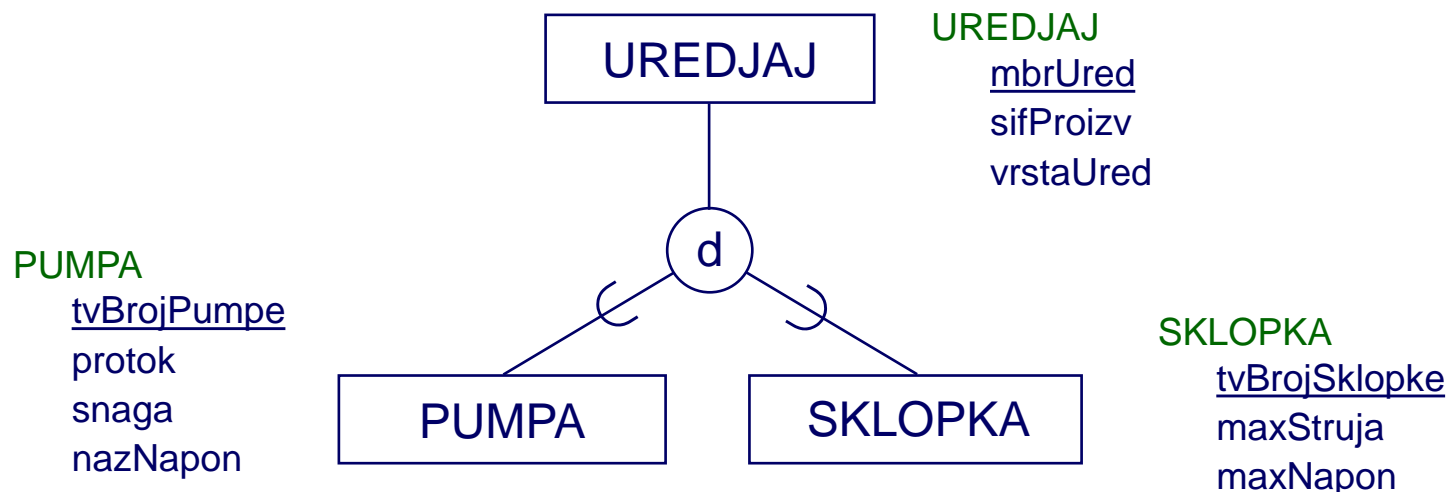
ILI

UREDJAJ = mbrUred, sifProizv, vrstaUred, protok, snaga, nazNapon, maxStruja, maxNapon

Za vježbu: napisati SQL naredbe za kreiranje relacija. Voditi računa o primarnim i stranim ključevima.

Preslikavanje u relacijski model

- specijalizacije imaju vlastite ključeve



UREDJAJ = mbrUred, sifProizv, vrstaUred

PUMPA = tvBrojPumpe, protok, snaga, nazNapon, mbrUred

SKLOPKA = tvBrojSklopke, maxStruja, maxNapon, mbrUred

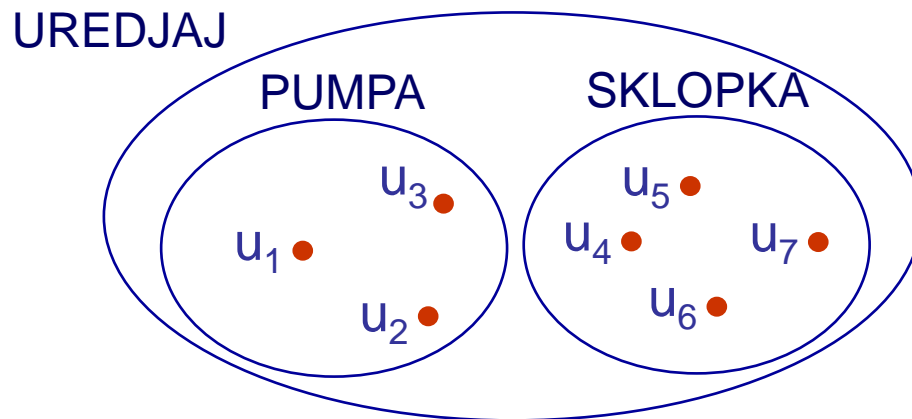
Alternativni
ključevi

Za vježbu: napisati SQL naredbe za kreiranje relacija. Voditi računa o primarnim, alternativnim i stranim ključevima.

Neekskluzivna generalizacija/specijalizacija

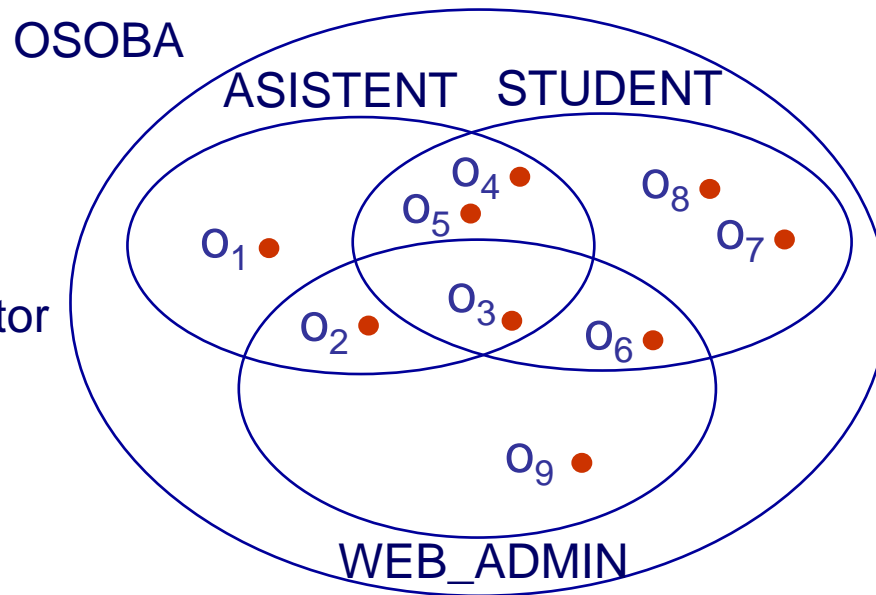
- ekskluzivna generalizacija/specijalizacija

Uređaj može biti ili pumpa ili sklopka

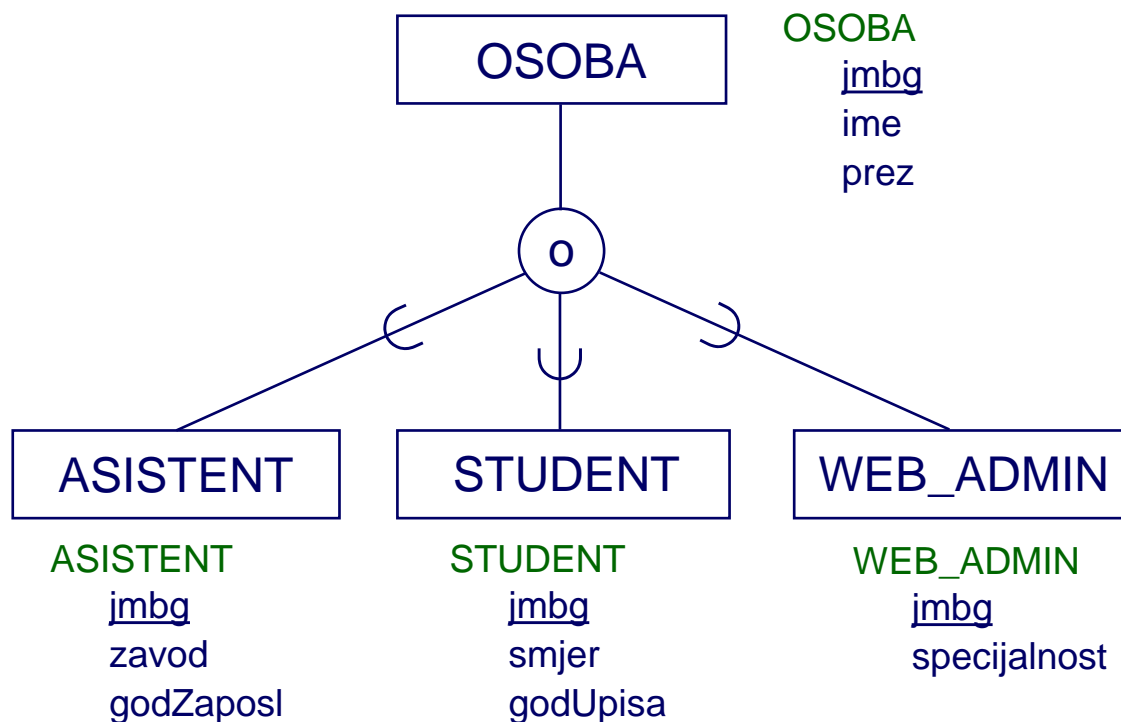


- neekskluzivna generalizacija/specijalizacija

Osoba može biti asistent i/ili student na poslijediplomskom studiju i/ili administrator web stranica fakulteta



Neekskluzivna generalizacija/specijalizacija



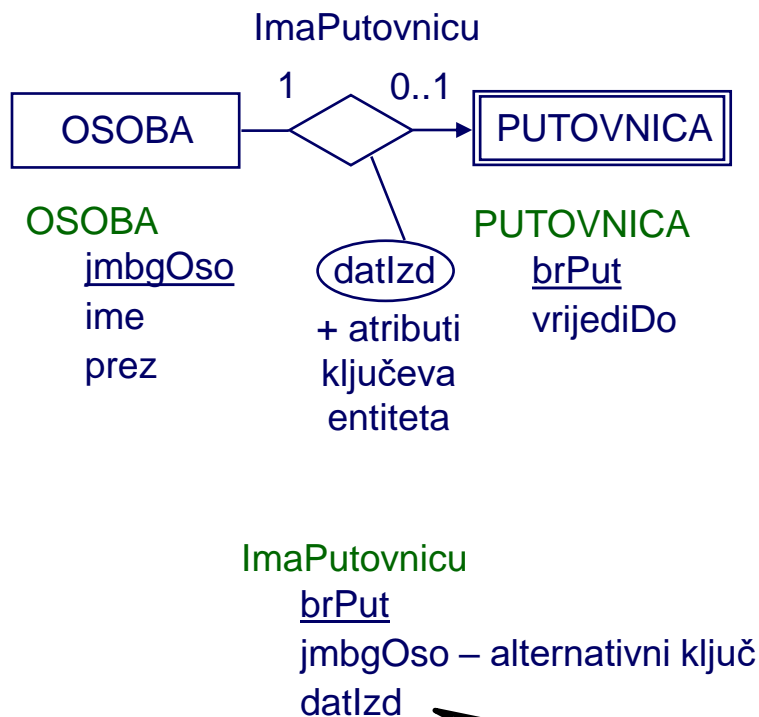
- ukoliko se radi o neekskluzivnoj generalizaciji/specijalizaciji u kružnicu se upisuje slovo **o** (*overlapping*)
 - osoba može biti asistent **i/ili** student **i/ili** administrator web stranica
- preslikavanje u relacijski model: jednako kao kod ekskluzivne generalizacije/specijalizacije (također su moguće specijalizacije s vlastitim i bez vlastitih ključeva)

Preslikavanje ER modela u relacijski model - rekapitulacija

- Osim ER dijagrama ER model sadrži i sheme entiteta i veza
- Pri opisu shema treba:
 - Voditi računa o vlastitim atributima entiteta,
 - Ispravno definirati ključeve entiteta i ključeve veza,
 - Definirati alternativne ključeve entiteta i veza (ako postoje)
- Preslikavanje u relacijski model:
 - Entiteti i veze iz ER modela postaju entiteti (relacije),
 - Sheme s jednakim ključevima zamijeniti njihovom unijom (jednaki ključ + unija atributa iz zavisnih dijelova) - osim ako su posljedica generalizacije i specijalizacije (tada se ne provodi unija),
 - Definirati pravila integriteta.

Na sljedećim prikaznicama nalaze se dodatni primjeri za vježbu.

Primjeri preslikavanja u relacijski model

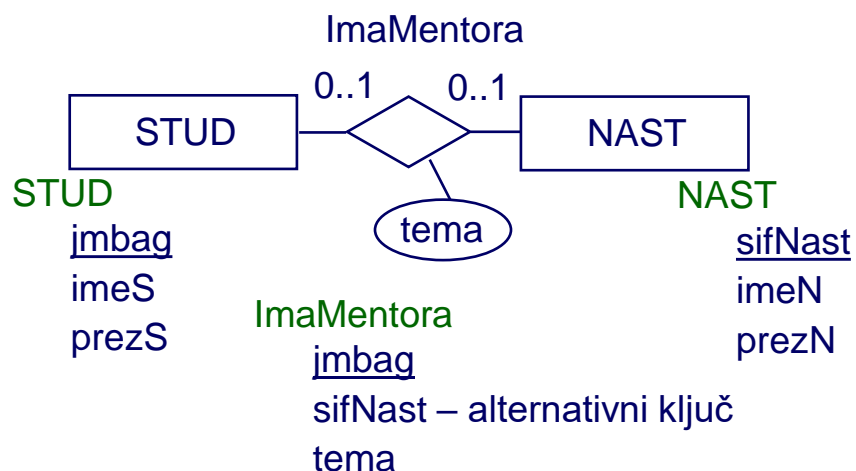


```
CREATE TABLE osoba (  
    jmbgOso    ...  
    , ime      ...  
    , prez     ...  
    , PRIMARY KEY (jmbgOso));
```

```
CREATE TABLE putovnica (  
    brPut      ...  
    , vrijediDo ...  
    , datIzd   ...  
    , jmbgOso  ... NOT NULL  
    , PRIMARY KEY (brPut)  
    , UNIQUE (jmbgOso)  
    , FOREIGN KEY (jmbgOso)  
      REFERENCES osoba (jmbgOso));
```

Bilo bi pogrešno kao **primarni** ključ odabrati jmbgOso.

Primjeri preslikavanja u relacijski model



ILI

Pretpostavlja se da jedan nastavnik može biti mentor najviše jednom studentu (ili niti jednom), te da student može imati najviše jednog mentora (ili niti jednog).

ImaMentora

sifNast

jmbag – alternativni ključ

tema

ILI

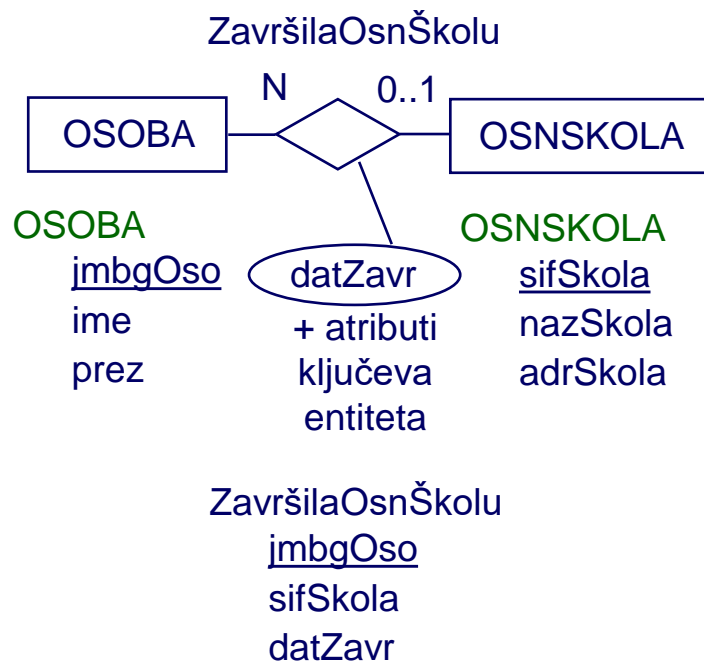
```
CREATE TABLE stud (
    jmbag ...
, imeS ...
, prezS ...
, sifNast ...
, tema ...
, PRIMARY KEY (jmbag)
, FOREIGN KEY (sifNast)
  REFERENCES nast(sifNast));
```

```
CREATE TABLE nast (
    sifNast ...
, imeN ...
, prezN ...
, PRIMARY KEY (sifNast));
```

```
CREATE TABLE stud (
    jmbag ...
, imeS ...
, prezS ...
, PRIMARY KEY (jmbag));
```

```
CREATE TABLE nast (
    sifNast ...
, imeN ...
, prezN ...
, jmbag ...
, tema ...
, PRIMARY KEY (sifNast)
, FOREIGN KEY (jmbag)
  REFERENCES stud (jmbag));
```

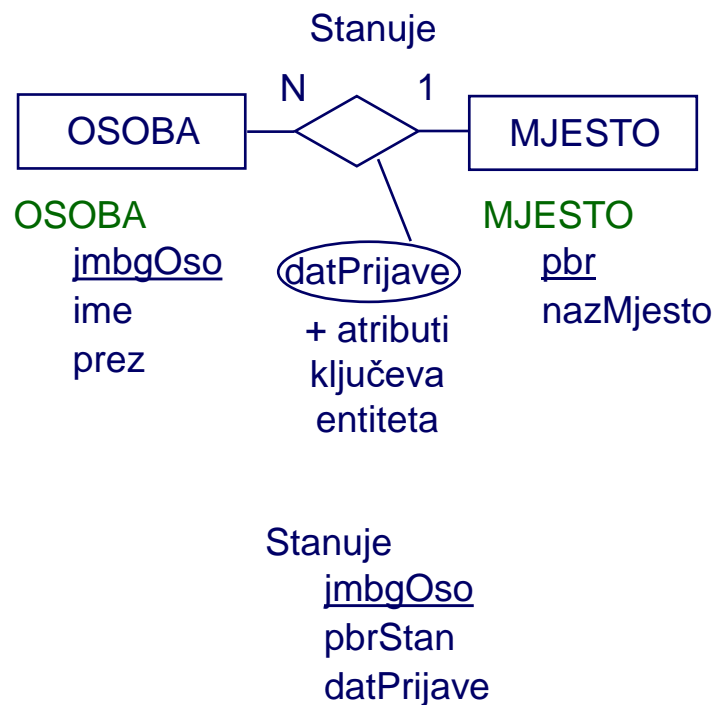
Primjeri preslikavanja u relacijski model



```
CREATE TABLE osoba (  
    jmbgOso ...  
    , ime ...  
    , prez ...  
    , sifSkola ...  
    , datZavr ...  
    , PRIMARY KEY (jmbgOso)  
    , FOREIGN KEY (sifSkola)  
      REFERENCES osnSkola(sifSkola));
```

```
CREATE TABLE osnSkola (  
    sifSkola ...  
    , nazSkola ...  
    , adrSkola ...  
    , PRIMARY KEY (sifSkola));
```

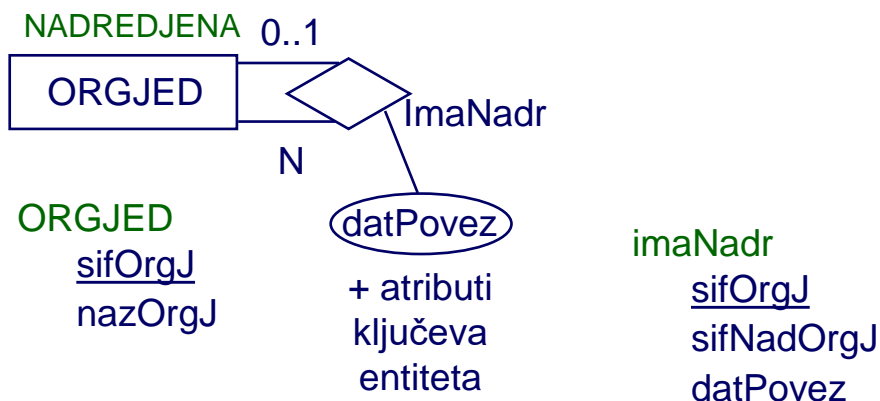
Primjeri preslikavanja u relacijski model



```
CREATE TABLE mjesto (  
    pbr ...  
    , nazMjesto ...  
    , PRIMARY KEY (pbr));
```

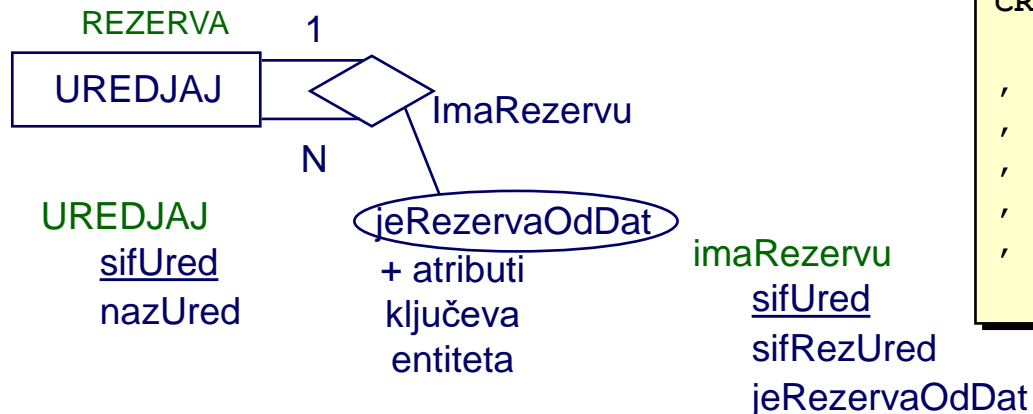
```
CREATE TABLE osoba (  
    jmbgOso ...  
    , ime ...  
    , prez ...  
    , pbrStan ... NOT NULL  
    , datPrijave ...  
    , PRIMARY KEY (jmbgOso)  
    , FOREIGN KEY (pbrStan)  
      REFERENCES mjesto (pbr);
```

Primjeri preslikavanja u relacijski model



```
CREATE TABLE orgJed (  
    sifOrgJ      ...  
    , nazOrgJ    ...  
    , sifNadOrgJ ...  
    , datPovez   ...  
    , PRIMARY KEY (sifOrgJ)  
    , FOREIGN KEY (sifNadOrgJ)  
      REFERENCES orgjed (sifOrgJ));
```

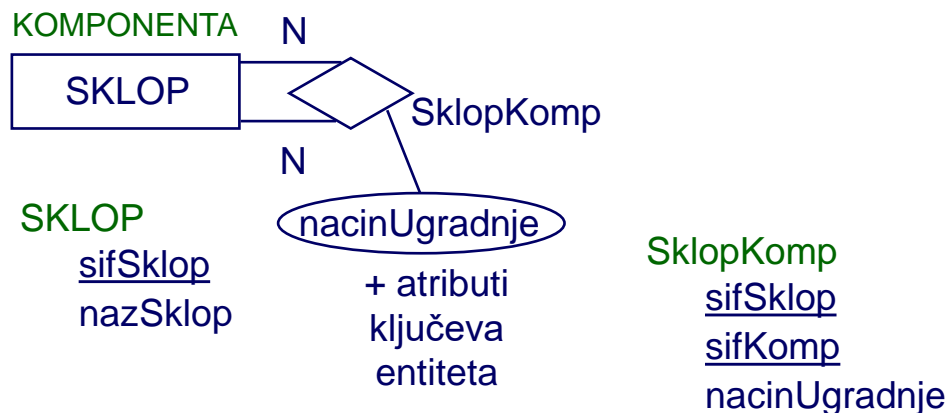
Za svaki uređaj mora biti evidentiran točno jedan rezervni uređaj. Jedan uređaj može biti rezerva za nekoliko uređaja. Evidentira se datum kad je uređaj postao rezerva nekog drugog uređaja



```
CREATE TABLE uredjaj (  
    sifUred      ...  
    , nazUred    ...  
    , sifRezUred ... NOT NULL  
    , jeRezervaOdDat ...  
    , PRIMARY KEY (sifUred)  
    , FOREIGN KEY (sifRezUred)  
      REFERENCES uredjaj (sifUred));
```

Zadatak: kako osigurati da uređaj ne bude sam sebi rezerva?

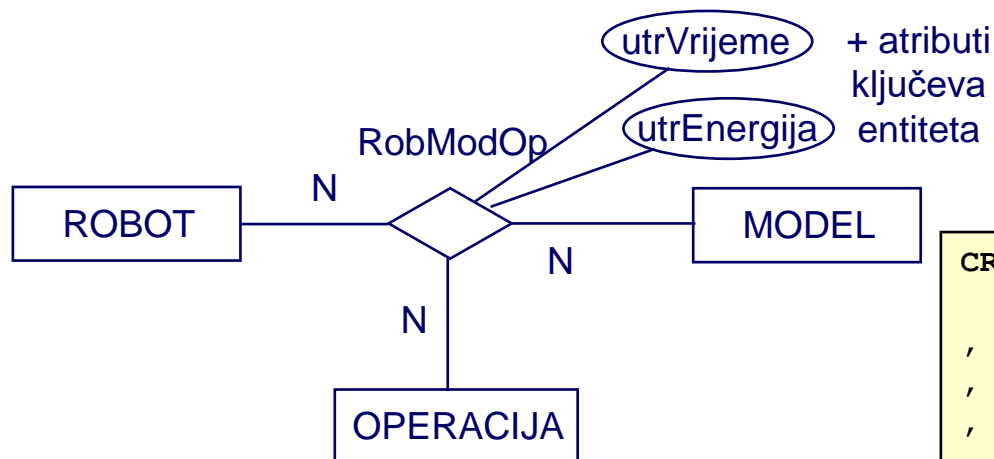
Primjeri preslikavanja u relacijski model



```
CREATE TABLE sklop (  
    sifSklop    ...  
    , nazSklop    ...  
    , PRIMARY KEY (sifSklop));
```

```
CREATE TABLE sklopKomp (  
    sifSklop    ...  
    , sifKomp    ...  
    , nacinUgradnje ...  
    , PRIMARY KEY (sifSklop, sifKomp)  
    , FOREIGN KEY (sifSklop)  
        REFERENCES sklop(sifSklop)  
    , FOREIGN KEY (sifKomp)  
        REFERENCES sklop (sifSklop));
```

Primjeri preslikavanja u relacijski model



ROBOT
sifRobot
nazRobot

MODEL
sifModel
nazModel

OPERACIJA
sifOper
nazOper

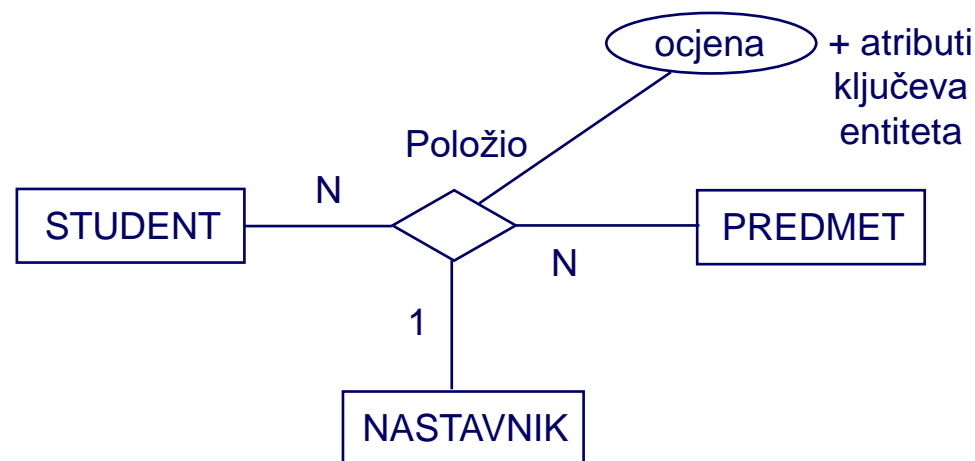
```
CREATE TABLE robModOp (  
    sifRobot    ...  
    , sifModel  ...  
    , sifOper   ...  
    , utrVrijeme ...  
    , utrEnergija ...  
    , PRIMARY KEY (sifRobot, sifModel, sifOper)  
    , FOREIGN KEY (sifRobot)  
        REFERENCES robot (sifRobot)  
    , FOREIGN KEY (sifModel)  
        REFERENCES model (sifModel)  
    , FOREIGN KEY (sifOper)  
        REFERENCES operacija (sifOper));
```

```
CREATE TABLE robot (  
    sifRobot    ...  
    , nazRobot  ...  
    , PRIMARY KEY (sifRobot));
```

```
CREATE TABLE model (  
    sifModel    ...  
    , nazModel  ...  
    , PRIMARY KEY (sifModel));
```

```
CREATE TABLE operacija (  
    sifOper     ...  
    , nazOper   ...  
    , PRIMARY KEY (sifOper));
```

Primjeri preslikavanja u relacijski model



STUDENT

mbrSt
prezSt
imeSt

PREDMET

sifPred
nazPred

NASTAVNIK

sifNast
prezNast
imeNast

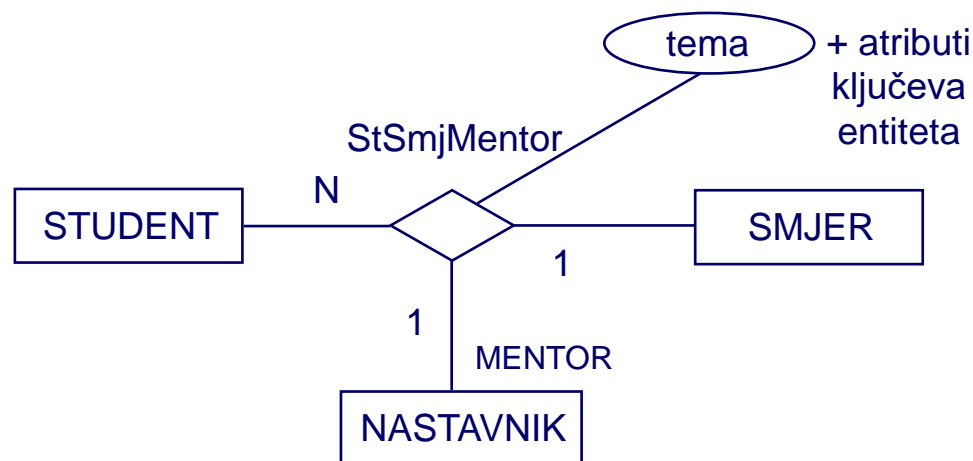
```
CREATE TABLE položio (  
    mbrSt      ...  
    , sifPred  ...  
    , sifNast  ... NOT NULL  
    , ocjena   ...  
    , PRIMARY KEY (mbrSt, sifPred)  
    , FOREIGN KEY (mbrSt)  
        REFERENCES student (mbrSt)  
    , FOREIGN KEY (sifPred)  
        REFERENCES predmet (sifPred)  
    , FOREIGN KEY (sifNast)  
        REFERENCES nastavnik (sifNast));
```

```
CREATE TABLE student (  
    mbrSt      ...  
    , prezSt   ...  
    , imeSt    ...  
    , PRIMARY KEY (mbrSt));
```

```
CREATE TABLE predmet (  
    sifPred    ...  
    , nazPred  ...  
    , PRIMARY KEY (sifPred));
```

```
CREATE TABLE nastavnik (  
    sifNast    ...  
    , prezNast ...  
    , imeNast  ...  
    , PRIMARY KEY (sifNast));
```


Primjeri preslikavanja u relacijski model



STUDENT

mbrSt
prezSt
imeSt

SMJER

sifSmjer
nazSmjer

NASTAVNIK

sifNast
prezNast
imeNast

```
CREATE TABLE stSmjMentor (  
    mbrSt      ...  
    , sifSmjer  ...  
    , sifNast   ... NOT NULL  
    , tema     ...  
    , PRIMARY KEY (mbrSt, sifSmjer)  
    , UNIQUE (mbrSt, sifNast)  
    , FOREIGN KEY (mbrSt)  
        REFERENCES student(mbrSt)  
    , FOREIGN KEY (sifSmjer)  
        REFERENCES smjer(sifSmjer)  
    , FOREIGN KEY (sifNast)  
        REFERENCES nastavnik(sifNast));
```

```
CREATE TABLE student (  
    mbrSt      ...  
    , prezSt   ...  
    , imeSt    ...  
    , PRIMARY KEY (mbrSt));
```

```
CREATE TABLE smjer (  
    sifSmjer   ...  
    , nazSmjer ...  
    , PRIMARY KEY (sifSmjer));
```

```
CREATE TABLE nastavnik (  
    sifNast    ...  
    , prezNast ...  
    , imeNast  ...  
    , PRIMARY KEY (sifNast));
```