

Baze podataka – Završni ispit – 2006/07

Tutorial – ER model

by kaotakov

Entitet – bilo što što ima suštinu ili bit

Regularni entitet – entitet koji može postojati sam za sebe

Slabi entitet – ne postoji ukoliko ne postoji drugi entitet, koji mu je 'vlasnik',

- svi su egzistencijalno slabi
- neki su i identifikacijski slabi, kada se za identifikaciju (ključ) koriste i ključni atributi entiteta vlasnika

Veza: unarna (refleksivna), binarna, ternarna...

Spojnost veze: jedan, više



Jedan djelatnik radi na jednom projektu. Na jednom projektu radi više djelatnika.



Jedan djelatnik radi na nula ili jednom projektu. Na jednom projektu radi između nula i više djelatnika.

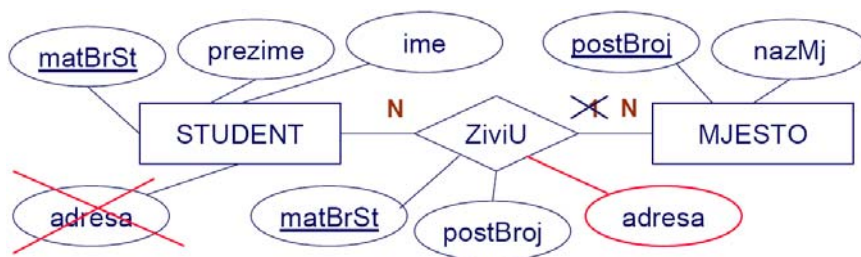


ENTITETI

Entiteti se opisuju samo vlastitim atributima. Odnosno, svi atributi koji ne opisuju isključivo entitet a imaju neke veze s njim, ne spadaju pod taj entitet, već u neku njegovu vezu.

Vlastiti atribut entiteta je atribut koji opisuje znanjao entitetu koja se pripisuju isključivo samom entitetu, a nikako vezi s drugim entitetima.

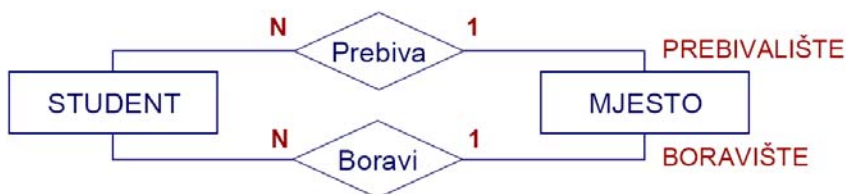
Isključivo **identifikacijski slabi** entiteti, osim svojih vlastitih entiteta, posjeduju i attribute primarnog ključa entiteta vlasnika.



dakle, tu je adresa suvišna kod entiteta STUDENT, jer adresa je u vezi sa vezom ZiviU, pa nije vlastiti atribut STUDENTA, tj. ne opisuje isključivo STUDENTA. Odmah se trpa u vezu ZiviU

VEZE

Paralelna veza



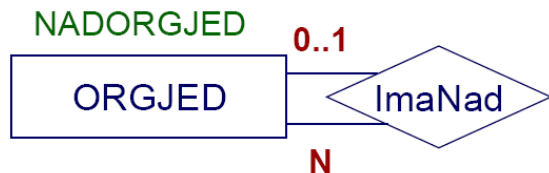
STUDENT: matBrSt, prezime, Ime

MJESTO: postBroj, nazMjesto

Prebiva: matBrSt, postBrojPreb

Boravi: matBrSt, postBrojBor

Refleksivna veza



ORGJED: sifOrgJed, nazOrgJed

Ovdje je problem što bi veza trebala sadržavati šifru dotične orgjed i nadređene orgjed, a kako sve orgjed imaju istu strukturu, to bi trebale biti dvije sifOrgJed, a pošto ne smiju biti isti nazivi atributa, moramo jednog preimenovati:

~~ImaNad = sifOrgJed, sifOrgJed~~

ImaNad = sifOrgJed, sifNadOrgJed

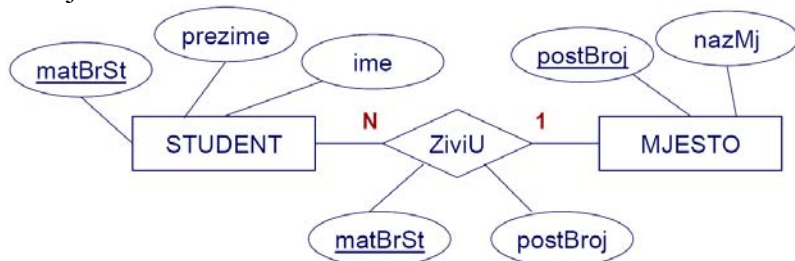
ATRIBUTI VEZE

Atributi veze čine :

- ključevi entiteta koje spaja (nemoraju nužno svi biti ključevi veze)
- neki vlastiti atributi veze

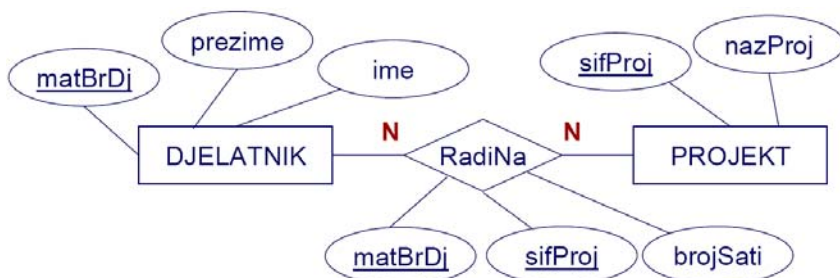
Ključ veze je određen onim ključem entiteta koje spaja, na čijoj strani je 'N' tj 'prema više'.

Primjeri:



'N' je na strani entiteta STUDENT, pa će i ključ veze biti ključ entiteta STUDENT

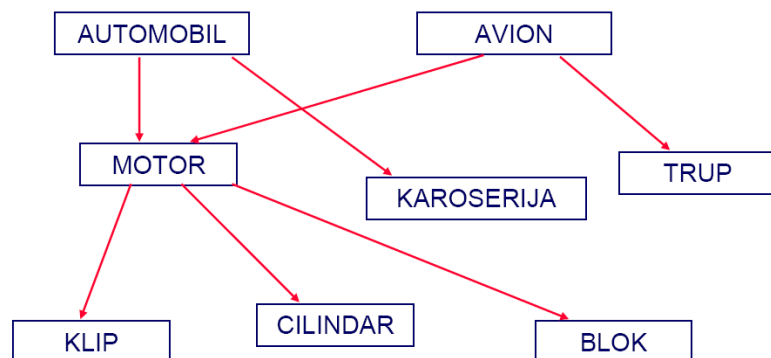
ZiviU: matBrSt, postbroj



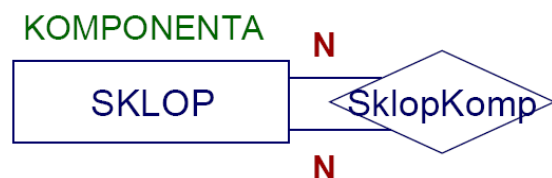
U ovom primjeru se 'N' nalazi kod oba entiteta koje povezuje veza pa se ključ sastoji od ključeva oba entiteta. Veza sadrži još jedan vlastiti atribut 'brojSati':

radiNa: matBrDj, sifProj, brojSati

Homogena mreža



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



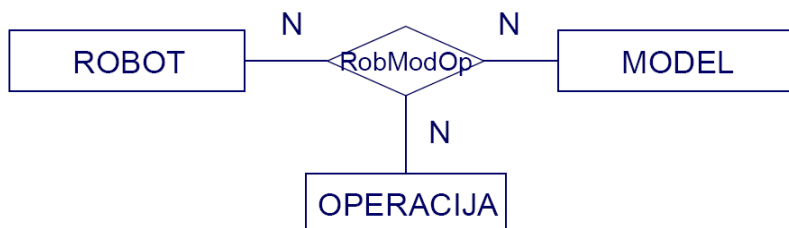
SKLOP: sifSklop, nazSklop

Primjetimo da je ovo reflektivna veza koja bi se opet sastojala od dva jednaka atributa, pa jedan trebamo preimenovati:

SklopKomp: sifSklop, SifKomp

TERNARNE VEZE

Ternarna veza prikazuje istovremeni odnos tri entiteta.



Ne može se bez gubitaka informacija zamijeniti trima binarnim vezama !!!

Veza bi se opisala ovako:

Jedan robot na **jednom** modelu može raditi **više** operacija.

Na **jednom** modelu **jednu** operaciju može raditi **više** robota

Jednu operaciju **jedan** robot može raditi na **više** modela

ROBOT: sifRobot, nazRobot

MODEL: sifModel, nazModel

OPERACIJA: sifOper, nazOper

RobModOp: sifRobot, sifModel, sifOper, utrVrijeme, utrEnergija

Dakle, tu se ključ ternarne veze sastoji od ključeva svih entiteta jer je kod svakog entiteta 'N', tj prema više.



STUDENT: matBrSt, prezSt, imeSt

PREDMET: sifPred, nazPred

NASTAVNIK: sifNast, prezNast, imeNast

Polozio: matBrSt, sifPred, sifNast, ocjena

U ovom slučaju se opet (kob reko) ključ veze sastoji od ključeva onih entiteta koji na svojoj strani imaju 'N'. Primjetimo da ovdje veza ima i vlastiti atribut 'ocjena'

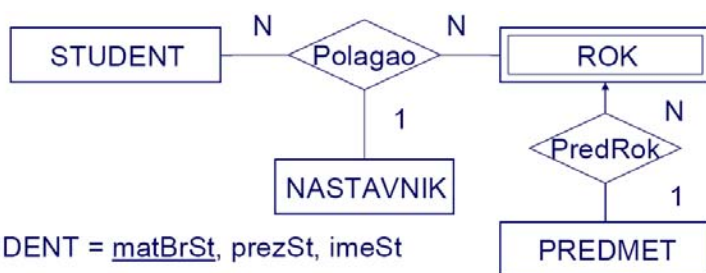
Veza se opisuje:

Jedan student može položiti **jedan** predmet kod (samo) **jednog** nastavnika.

Jedan predmet kod **jednog** nastavnika može položiti **više** studenata.

Kod **jednog** nastavnika **jedan** student može položiti **više** predmeta.

Dakle za određenog studenta i određeni predmet postoji samo jedan nastavnik kod kojeg se predmet položio.



STUDENT = matBrSt, prezSt, imeSt

PREDMET = sifPred, nazPred

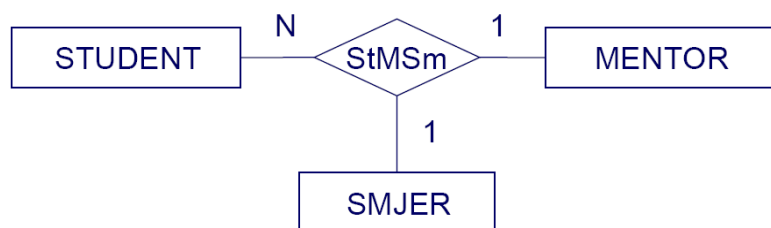
ROK = sifPred, datRok, vrstaRok

NASTAVNIK = sifNast, prezNast, imeNast

PredRok = sifPred, datRok

Polagao = matBrSt, sifPred, datRok, sifNast, ocjena

Ovdje je samo potrebno primjetiti kako se ključ veze 'Polagao' sastoji od cijelog ključa slabog entiteta ROK i ključa entiteta STUDENT. Pošto je ROK slabi, ima ključ složen od svojeg atributa 'datRok' i ključa PREDMETa 'sifPred'.



Jedan student ima **jednog** mentora na **jednom** (određenom) smjeru.

Jedan mentor na **jednom** smjeru može imati **više** studenata.

Na **jednom** smjeru **jedan** student ima točno **jednog** mentora.

Dakle, smjer je točno određen ako znamo studenta i njegovog mentora, a i mentor je točno određen ako znamo koji student je na kojem smjeru. Ako pak znamo samo mentora i smjer, nije jednoznačno određen student, što je i logično jer jedan mentor na jednom smjeru može biti mentor više studenata.

To dovodi do zaključka da ključ veze možemo odabrati na dva načina:

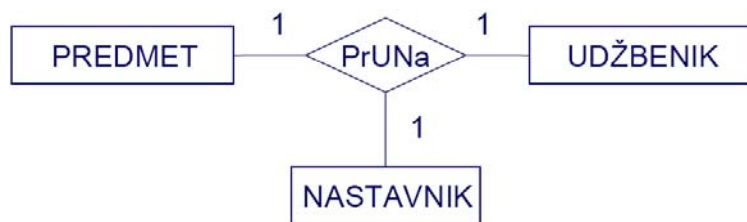
- ključ STUDENTA + ključ MENTORA
- ključ STUDENTA + ključ SMJERA

STUDENT = matBrSt, prezSt, imeSt

SMJER = sifSmjer, nazSmjer

MENTOR = sifMentor, prezMentor, imeMentor

StMSm = sifSmjer, matBrSt, sifMentor



Ovdje je udžbenik određen ako znamo predmet i nastavnika, isto tako je nastavnik određen poznavanjem predmeta i udžbenika, a i predmet je određen poznavanjem nastavnika i udžbenika. Što će reć, ključ veze 'PrUNa' mogu biti ključevi bilo koja dva entiteta:

PREDMET = sifPred, nazPred

UDŽBENIK = sifUdz, nazUdz

NASTAVNIK = sifNast, prezNast, imeNast

PrUNa = sifPred, sifUdz, sifNast

SPECIJALIZACIJA I GENERALIZACIJA

Def:

Specijalizacija – Entiteti jednog skupa entiteta mogu setemeljem njihovih karakterističnih svojstava klasificirati zasebne skupove entiteta, postupkom koji se naziva SPECIJALIZACIJA

Generalizacija – Entiteti iz nekoliko skupova entiteta sa sličnim svojstvima mogu se grupirati u zajednički skup entiteta, postupkom koji se naziva GENERALIZACIJA

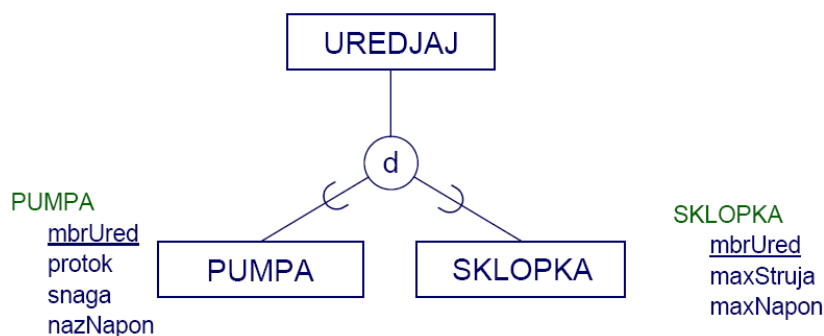


SPECIJALIZACIJA



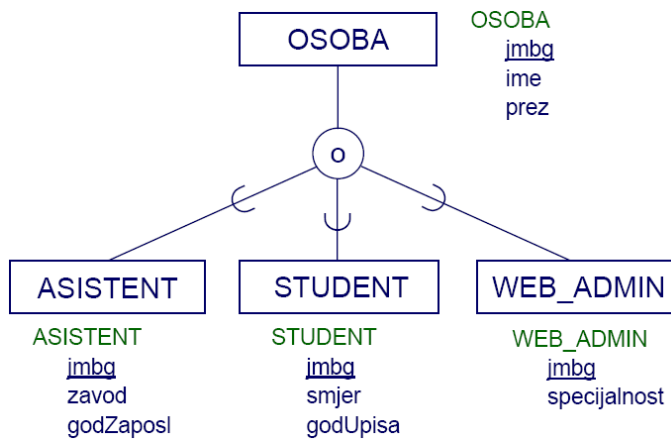
GENERALIZACIJA

Znači, specijalizacija je od općeg prema posebnom, a generalizacija grupiranje posebnih prema općem.



Ovako se generalizacija i specijalizacija zapisuju u ER dijagramu.

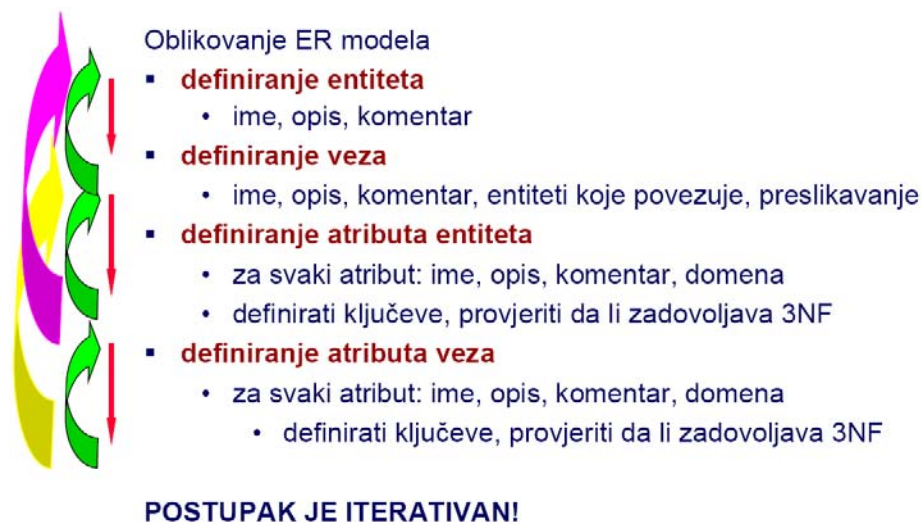
Kada je 'd' unutar kružnice, to je EKSKLUZIVNA specijalizacija/generalizacija, tj uređaj može biti ili pumpa ili sklopka, ne može biti oboje.



Ovdje se radi o NEEKSKLUZIVNOJ generalizaciji/specijalizaciji, oznažava se sa 'o' u kružnici. To znači da osoba može biti jedno od ovog troje a može biti i svo troje zajedno, znači Mirko je asistent, student i web_admin na FERu

OBLIKOVANJE ER MODELA

Znači **OVO TU** se traži na ispitu. Jedna napomena, sljedeća slika nije baš jedna od onih nevažnih sličica (nije džaba ode) koje se preskaču (pred12/45 slajd):



ER modeliranje najlakše je shvatiti na primjerima (predavanje 14), u to ime:

Primjer 1.)

MODEL BAZE PODATAKA ZA RAZREDBENI ISPIT (pred14, slajd 2)

Potrebno je evidentirati podatke o kandidatima: JMBG, prezime, ime, završenu srednju školu, mjesto rođenja i mjesto stanovanja. Pretpostavlja se da je kandidat završio samo jednu srednju školu. Za svaku srednju školu treba evidentirati šifru koja ju jedinstveno identificira, naziv, adresu i mjesto u kojem se škola nalazi. Za mjesto treba evidentirati poštanski broj, naziv mjesta i županiju u kojoj se mjesto nalazi. Županija ima svoju šifru i naziv.

Treba evidentirati podatke o zadacima na testu: redni broj zadatka, tekst zadatka, oznaku točnog odgovora (može biti A,B,C,D ili E).

Za svakog kandidata evidentirati odgovore koje je dao na zadatke (mogući odgovori kandidata su A,B,C,D,E ili ništa).

Nacrtati ER modeli opisati entitete i veze. Sve sheme moraju zadovoljavati 3NF

Opisati relacijski model u obliku SQL naredbi za kreiranje relacija s opisanim integritetskim ograničenjima. Odabrati prikladne tipove podataka.

Najvažnije je definirati entitete i veze, ostalo (atributi entiteta, atributi veza, prebacivanje u relacijski model je rutina) je rutina. Entiteti i veze se trebaju nekak skužiti iz zadatka, pa je poželjno da onaj koji sastavlja zadatak nedvosmisleno odredi, tj. natukne oblik entiteta.

Dakle, krećemo od entiteta. Prvo ćemo pobrojati sve entitete iz teksta zadatka:

'Potrebno je evidentirati podatke o kandidatima:' → evo prvog entiteta – KANDIDAT

'Za svaku srednju školu treba evidentirati šifru koja ju jedinstveno identificira...' drugi! – ŠKOLA

'Za mjesto treba evidentirati poštanski broj, ...' treći – MJESTO

'Županija ima svoju šifru i naziv.' četvrti – ŽUPANIJA

'Treba evidentirati podatke o zadacima na testu...' posljednji peti – ZADATAK

entiteti: KANDIDAT, ŠKOLA, MJESTO, ŽUPANIJA, ZADATAK

Sljedeće što moramo po onom crtežu, jest definirati veze među entitetima. Veze isto tako prepoznamo iz teksta zadatka. Sve ono što se traži u zadatku za određeni entitet a ne opisuje samo i isključivo njega, već se odnosi i na neki drugi entitet daje naslutiti da se radi o vezi, npr: 'Potrebno je evidentirati podatke o kandidatima: JMBG, prezime, ime, završenu srednju školu, mjesto rođenja i mjesto stanovanja.'

Sad gledamo te značajke:

JMBG – opisuje samo entitet KANDIDAT, nema veze sa drugim entitetima, znači JMBG će biti vlastiti entitet

prezime – isto kao JMBG

ime – isto kao JMBG

završena srednja škola – kako imamo i entitet ŠKOLA očito je da će postojati neka veza između ova dva entiteta, znači dodajemo vezu → SkolaKand

mjesto rođenja – postoji entitet mjesto, znači nova veza → MjRodKand

mjesto stanovanja – postoji entitet mjesto, znači nova veza → MjStanKand

To bi bilo to što se tiče veza sa kandidatom,

trenutno stanje veza: SkolaKand, MjRodKand, MjStanKand

Idemo dalje na veze drugih entiteta

'Za svaku srednju školu treba evidentirati šifru koja ju jedinstveno identificira, naziv, adresu i mjesto u kojem se škola nalazi.'

Opet isto:

Šifra škole – karakteristična samo za entitet ŠKOLA, pa postaje vlastiti atribut sifSkola

naziv – isto kao i šifra

mjesto u kojem se nalazi – imamo entitet MJESTO, nova veza → MjestoSkola

adresa – e ovo je specifično za samu školu ali i ima veze sa mjestom u kojem se nalazi, pošto imamo već novu vezu MjestoSkola, utrat ćemo adresu kao vlastiti atribut te veze

trenutno stanje veza: SkolaKand, MjRodKand, MjStanKand, MjestoSkola

Idemo još dalje, na ostale entitete

'Za mjesto treba evidentirati poštanski broj, naziv mjesta i županiju u kojoj se mjesto nalazi' poštanski broj - karakterističan samo za entitet MJESTO, postaje vlastiti atribut tog entiteta

naziv – isto kao poštanski broj

županija u kojoj se nalazi – imamo entitet ŽUPANIJA, nova veza → ZupMjesto

trenutno stanje veza: SkolaKand, MjRodKand, MjStanKand, MjestoSkola, ZupMjesto

Još dalje

'Treba evidentirati podatke o zadacima na testu: redni broj zadatka, tekst zadatka, oznaku točnog odgovora. Za svakog kandidata evidentirati odgovore koje je dao na zadatke.'

redni broj zadatka – očito vlastiti atribut entiteta ZADATAK

tekst – isto vlastiti

oznaka točnog odgovora – isto vlastiti jer ne ovisi u nikojem pogledu sa nekim drugim entitetom, samo označava točan

odgovori svakog kandidata na zadatke – e ovo jako smrdi na vezu s KANDIDATom i ZADATAKom pa ćemo ju i staviti → OdgKandZad

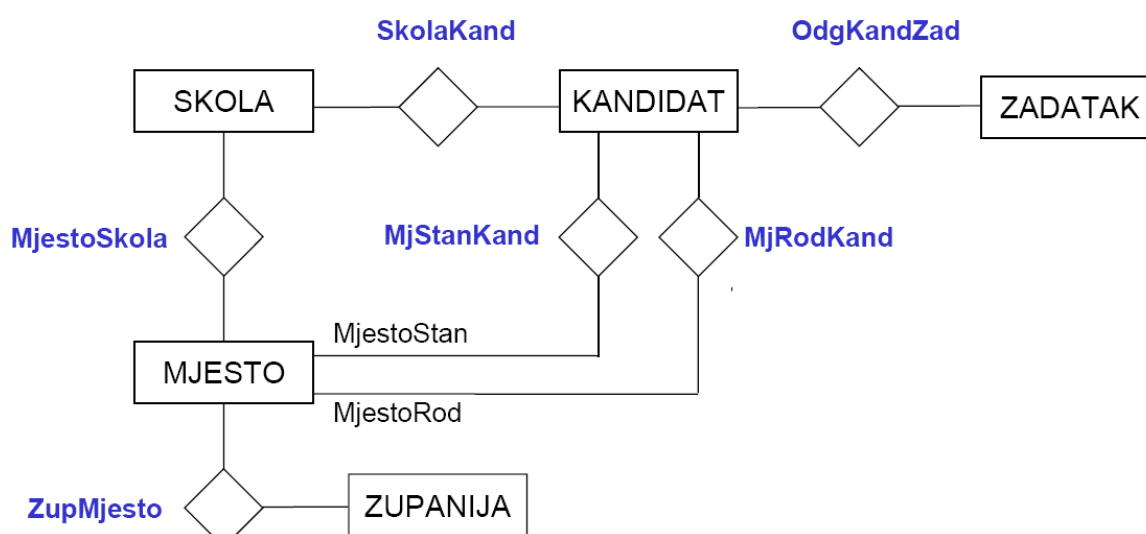
trenutno (i konačno) stanje veza: SkolaKand, MjRodKand, MjStanKand, MjestoSkola, ZupMjesto, OdgKandZad

to je to što se tiče veza, iscrpili smo sve rečenice zadatka, konačno stanje entiteta i veza:

entiteti: KANDIDAT, ŠKOLA, MJESTO, ŽUPANIJA, ZADATAK

veze: SkolaKand, MjRodKand, MjStanKand, MjestoSkola, ZupMjesto, OdgKandZad

sad nam slijedi crtanje radne verzije ER modela:



Sad treba definirati spojnosti veza, što ne bi trebao biti problem. Odrednice za neke spojnosti su zadane u zadatku, a ove za koje nisu pišemo po intuiciji, npr

MjStanKand:

jedan kandidat može stanovati u jednom mjestu → znači pišemo 1 na stranu MJESTO

u jednom mjestu može stanovati više kandidata → znači pišemo 'N' na stranu KANDIDAT

U zadatku piše:

'Pretpostavlja se da je kandidat završio samo jednu srednju školu'

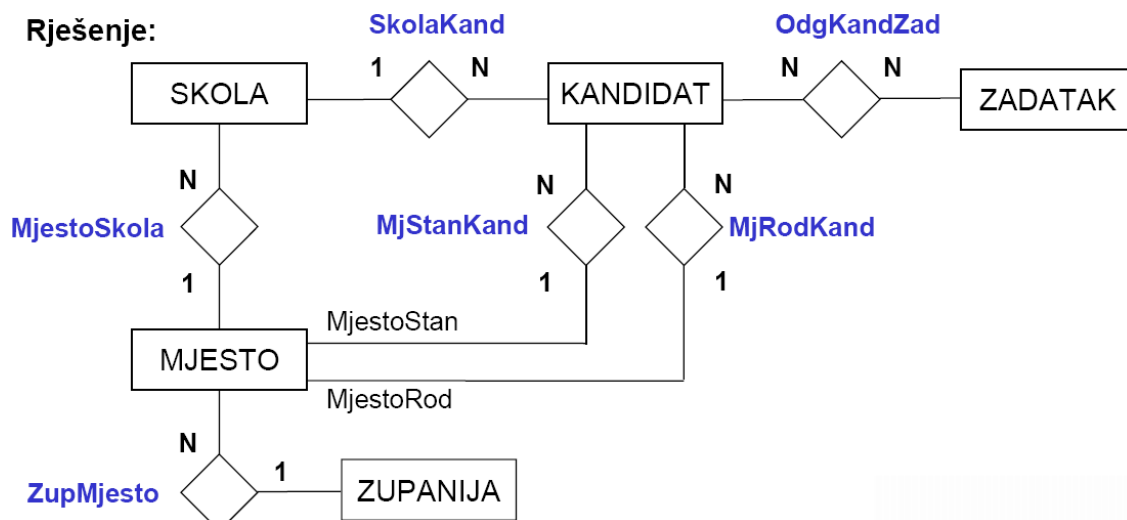
što će reći:

jedan kandidat može završiti jednu srednju školu → 1 kod SKOLA

jednu srednju školu može završiti više kandidata → N kod KANDIDAT

Na sličan način napišemo i za ostale spojnosti, jedino za vezu OdgKandZad imamo poseban slučaj, jer jedan kandidat može riješiti više zadataka, a i jedan zadatak može riješiti više kandidata, pa je 'N' na obje strane.

Crtez sa spojnostima:



Sad kad smo defilirali spojnosti, možemo definirati attribute entiteta i veza. U attribute entiteta trpamo samo one koji opisuju isključivo te entitete (osim u slučaju slabog entiteta, kojeg nemamo u ovom primjeru), sve ostale attribute trpamo u attribute veza ako možemo. Atributi veza će se sastojati od ključnih atributa entiteta koje spajaju i nekih vlastitih. Pokazat ćemo na par entiteta i veza:

iz teksta zadatka se zaključuje što opisuje posebno entitet a što ne, kod KANDIDATa npr JMBG, ime i prezime opisuju isključivo njega, a recimo mjesto stanovanja smo stavili u vezu, a njegov odgovor na pojedino pitanje ima veze sa zadacima pa ćemo taj atribut utrpati u vezu za zadacima. JMBG će očitito biti ključ. Dakle,

KANDIDAT: JMBG, imeKand, prezKand

Isto tako i kod zadatka, redni broj, tekst i oznaka tocnog opisuju isključivo ZADATAK, i logično je da redni broj bude ključ

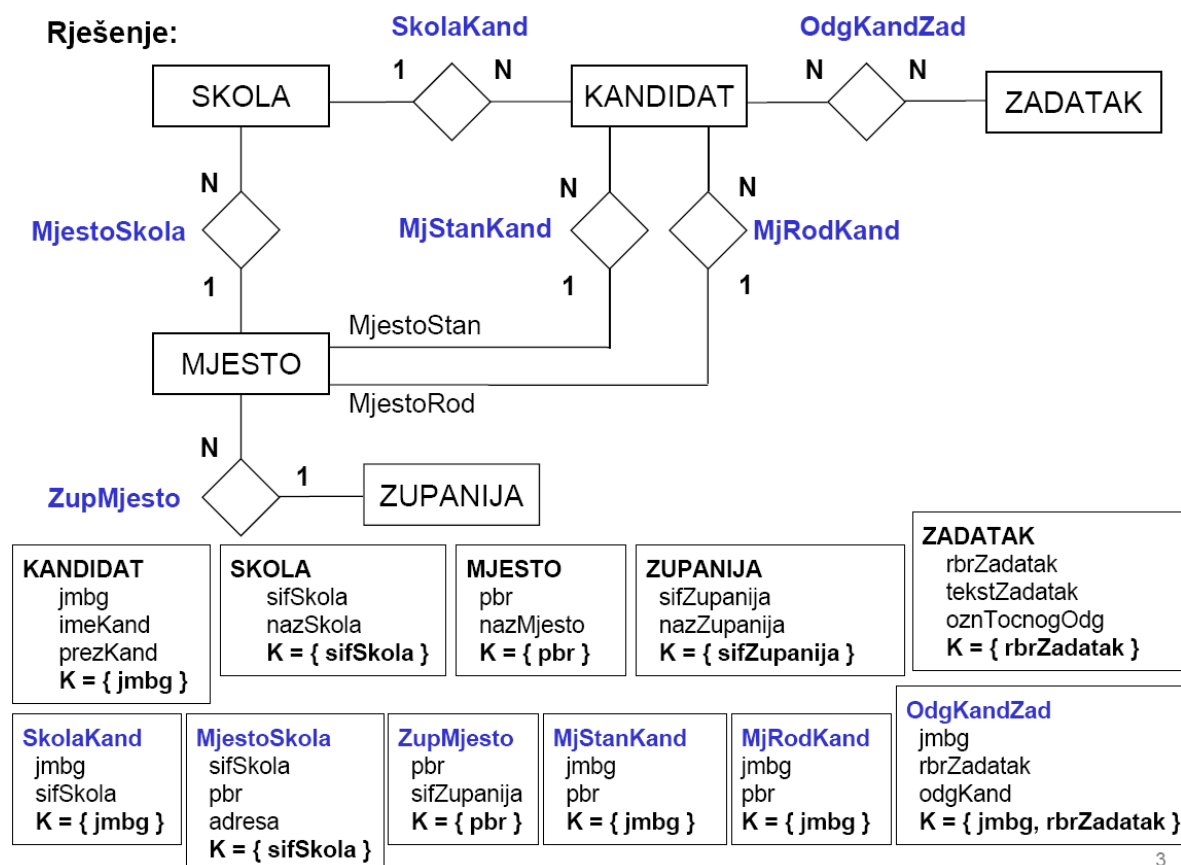
ZADATAK: rbrZadatak, tekstZadatak, oznTocnogOdg

Kod veza, kao što rekoah, stavljamo ključne attribute entiteta koje povezujei ako postoji neki vlastiti atribut. Ključ veze je onaj ključni atribut nekog entiteta na čijoj je strani 'N', može bit jedan, mogu biti oba, a mogu biti i sva tri ako je veza ternarna:

MjStanKand: JMBG, pbr ← na strani kandidata je 'N' pa je JMBG i ključ veze

MjestoSkola: sifSkola, pbr, adresa ← ovdje je adresa vlastiti atribut veze jer ima veze sa oba entiteta koje povezuje. Kod veze 'MjStanKand' nemamo adresu jer adresa kansasata nije tražena u zadatku.

Na isti način definiramo ostale attribute entiteta i veza



3

Prebacivanje u relacijski model

Prije nastavka malo o prebacivanju

Problem kod relacijskog modela je što baš i nemamo definirane veze, pa njih i njihove vlastite attribute moramo opisati na neki durgi način (FOREIGN KEY, UNIQUE, itd...).

Glavno pravilo je UNIJA RELACIJSKIH SHEMA S JEDNAKIM KLJUČEVIMA, oće reć, veze koje imaju jednake ključeve kao neki entiteti, spajaju se s njima u jednu relaciju, a one koje nemaju, postaju nove relacije:

ovdje se veza Stanuje spaja sa entitetom DJELATNIK



DJELATNIK = matBrDj, prezime, ime

MJESTO = postBr, nazMjesto

Stanuje = matBrDj, postBr, adresa

Relacijske sheme opisuju entitete (veze postaju entiteti)

DJELATNIK = matBrDj, prezime, ime

MJESTO = postBr, nazMjesto

Stanuje = matBrDj, postBr, adresa

Unija relacijskih shema s jednakim ključevima

DJELATNIK = matBrDj, prezime, ime, postBr, adresa

MJESTO = postBr, nazMjesto

ovdje vezaRadiNa nema s nijednim entitetom zajednički ključ pa postaje novi entitet



DJELATNIK= matBrDj, prezime, ime

PROJEKT= sifProj, nazProj

RadiNa = matBrDj, sifProj, brojSati

Relacijske sheme opisuju entitete (veze postaju entiteti)

DJELATNIK= matBrDj, prezime, ime

PROJEKT= sifProj, nazProj

RadiNa = matBrDj, sifProj, brojSati

Vraćamo se na zadatak:

Sa entitetom KANDIDAT zajedničke ključeve imaju veze SkolaKand, MjRodKand, MjStanKand. Znači u entitetu KANDIDAT će biti sadržane te veze. To se ostvaruje na sljedeći način (screenshot, neda mi se pisat, a nemogu kopirat nikak)

```

CREATE TABLE kandidat (
    jmbg          CHAR(13)
  , imeKand      CHAR(20)
  , prezKand     CHAR(20)
  , pbrRod       INTEGER NOT NULL
  , pbrStan      INTEGER NOT NULL
  , sifSkola     INTEGER NOT NULL
  , PRIMARY KEY (jmbg)
  , FOREIGN KEY (pbrRod) REFERENCES mjesto (pbr)
  , FOREIGN KEY (pbrStan) REFERENCES mjesto (pbr)
  , FOREIGN KEY (sifSkola) REFERENCES skola (sifSkola));
  
```

Definiramo attribute i primarni ključ.

Kako imamo dvije veze sa poštanskim brojevima

MjRodKand: JMBG, pbr

MjStanKand: JMBG, pbr

moramo ih preimenovati pa dobijem attribute pbrRod i pbrStan

Atribut sifSkola smije ostat takav kakav je. Primjetimo da sve te tri veze nisu imale vlastitih atributa pa ih ni nemamo u entitetu KANDIDAT.

Sve te novopridosle attribute u KANDIDATu referencirano sa entitetom kojeg je veza u ER modelu povezivala i time simuliramo ER model običnim relacijskim modelom, npr:

FOREIGN KEY (sifSkola) REFERENCES skola(sifSkola)

Nijedan od tih atributa veza ne smije biti NULL jer su tako definirane veze, 'jedan prema više', znači najmanje jedan (i najviše u ovom slučaju).

Entitet MJESTO ima zajednički ključ sa vezom ZupMjesto, pa i njih spojimo.

Entitet SKOLA ima zajednički ključ sa vezom MjestoSkola, pa i njih spojimo. Tu nailazimo prvi put na vlastiti atribut veze 'adresa' koji također stavljamo pod attribute entiteta SKOLA:

```
CREATE TABLE skola (
    sifSkola INTEGER
, nazSkola CHAR(40)
, pbr INTEGER NOT NULL
, adresa CHAR(40)
, PRIMARY KEY (sifSkola)
, FOREIGN KEY (pbr) REFERENCES mjesto(pbr));
```

Entitet ZUPANIJA nema s nijednom vezom nijedan zajednički ključni atribut pa ga ostavljamo u tom obliku.

Entitet ZADATAK ključni atribut 'rbrZadatak' koji je dio ključa veze 'OdgKandZad' ali nisu jednaki u potpunosti, pa ih zato nemožemo spojiti, pa ZADATAK ostavljamo takav kakav je. Time smo riješili sve entitete, sad ćemo još provjeriti sve veze. Zasad smo pobrali veze SkolaKand, MjRodKand, MjStanKand, ZupMjessto, MjestoSkola.

Ostaje nam samo veza 'OdgKandZad' koja nema zajednički ključ s nijednim od entiteta, pa ta veza postaje novi entitet (s vlastitim atributom odgKand):

```
CREATE TABLE odgKandZad (
    jmbg CHAR(13)
, rbrZadatak INTEGER
, odgKand CHAR(1)
, PRIMARY KEY (jmbg, rbrZadatak)
, FOREIGN KEY (jmbg) REFERENCES kandidat (jmbg)
, FOREIGN KEY (rbrZadatak) REFERENCES zadatak (rbrZadatak));
```

Time smo riješili i sve veze. Konačni izgled svih entiteta u relacijskom modelu izgleda:

```
CREATE TABLE zupanija (
    sifZupanija SMALLINT
, nazZupanija CHAR(40)
, PRIMARY KEY (sifZupanija));

CREATE TABLE mjesto (
    pbr INTEGER
, nazMjesto CHAR(20)
, sifZupanija SMALLINT NOT NULL
, PRIMARY KEY (pbr)
, FOREIGN KEY (sifZupanija) REFERENCES zupanija(sifZupanija));

CREATE TABLE skola (
    sifSkola INTEGER
, nazSkola CHAR(40)
, pbr INTEGER NOT NULL
, adresa CHAR(40)
, PRIMARY KEY (sifSkola)
, FOREIGN KEY (pbr) REFERENCES mjesto(pbr));

CREATE TABLE zadatak (
    rbrZadatak INTEGER
, tekstZadatak CHAR(512)
, oznTocnogOdg CHAR(1)
, PRIMARY KEY (rbrZadatak));
```

```
CREATE TABLE kandidat (
    jmbg          CHAR(13)
  , imeKand      CHAR(20)
  , prezKand     CHAR(20)
  , pbrRod       INTEGER NOT NULL
  , pbrStan      INTEGER NOT NULL
  , sifSkola     INTEGER NOT NULL
  , PRIMARY KEY (jmbg)
  , FOREIGN KEY (pbrRod) REFERENCES mjesto (pbr)
  , FOREIGN KEY (pbrStan) REFERENCES mjesto (pbr)
  , FOREIGN KEY (sifSkola) REFERENCES skola (sifSkola));
```

```
CREATE TABLE odgKandZad (
    jmbg          CHAR(13)
  , rbrZadatak   INTEGER
  , odgKand      CHAR(1)
  , PRIMARY KEY (jmbg, rbrZadatak)
  , FOREIGN KEY (jmbg) REFERENCES kandidat (jmbg)
  , FOREIGN KEY (rbrZadatak) REFERENCES zadatak (rbrZadatak));
```

To je to. Zadatak je riješen crtanjem ER dijagrama, opisivanjem entiteta i veza te prebacivanjem u relacijski model. Najveći problem je zapravo prepoznat entitete (koji mogu biti i slabi) i veze (koje mogu biti i refleksivne i ternarne), jer ako se tu zaje* griješite i u ostatku zadatka. Makar ćete i dobit neke bodove, nećete bome sve.

Sad ćemo proć kroz neke tipične situacije drugih zadataka jer tog ima previše da bi baš svaki obrađivali, a takve situacije će sigurno bit na testu.

Primjer 2.) MODEL BAZE PODATAKA ZA VIDEOTEKU (pred 14/slajd 7)

Napomena: 'Ista osoba može u istom filmu imati različite funkcije'.

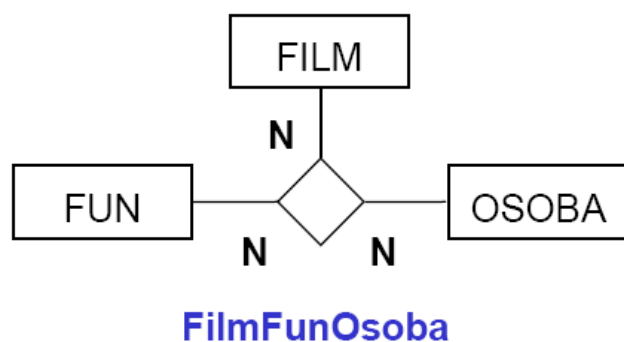
relacija VIDEOTEKA

sif Film	naslovFilm	ozn Fun	nazFun	sif Osoba	ime	prezime
1	Nepomirljivi	RED	redatelj	10	Clint	Eastwood
1	Nepomirljivi	GL	glumac	10	Clint	Eastwood
1	Nepomirljivi	GL	glumac	20	Morgan	Freeman
2	Mostovi okruga Madison	RED	redatelj	10	Clint	Eastwood
2	Mostovi okruga Madison	GL	glumac	40	Meryl	Streep
2	Mostovi okruga Madison	GL	glumac	10	Clint	Eastwood
3	Prljavi Harry	RED	redatelj	30	Don	Siegel
3	Prljavi Harry	GL	glumac	10	Clint	Eastwood

Iz tablice se vidi da

jedna osoba može na jednom filmu imati više uloga
na jednom filmu jednu ulogu mogu imati više osoba
jednu ulogu jedna osoba može imati na više filmova

→ TERNARNA VEZA jeeeeeeeeeee



Ovakva veza se **NE MOŽE** zamijeniti sa tri binarne veze bez gubitaka informacija (objasnjeno je na slajdovima).

entiteti:

FILM: sifFilm, naslovFilm

OSOBA: sifOsoba, ime, prezime

FUN: oznFun, nazFun

veza:

FilmFunOsoba: sifFilm, sifOsoba, oznFun

-ključ veze mora imati ključ svih entiteta jer je 'N' kod svih

Prebacivanje u relacijski model:

Kako veza 'FilmFunOsoba' nema zajednički ključ sa nijednim entitetom, postaje novi entitet. Probajte sami napisati kako bi to izgledalo, znači svaki entitet je posebna tablica a veza je novi entitet koji se sastoji samo od tri atributa gdje su sva tri ključ i sva tri se referenciraju na odgovarajuću tablicu.

Primjer 3.) MODEL BAZE PODATAKA ZA PODUZEĆE ZA ODRŽAVANJE PLINSKIH INSTALACIJA (pred 14/slajd 14)

Iz rečenice

'Za svaki pojedini uređaj treba evidentirati vrstu uređaja (V, B, R), proizvođača uređaja, tvornički broj i godinu proizvodnje uređaja.'

se vidi da uređaji imaju zajedničke atribute.

Iz sljedeće rečenice:

'Dodatno, ovisno o vrsti uređaja, treba evidentirati njima svojstvene posebne ili specijalističke podatke'

se vidi da određeni uređaji imaju vlastite atribute.

Ako su im neki atributi jednaki, a neki posebni, zaključujemo da se radi o specijalizaciji/generalizaciji. Dakle, uređaj će biti jedan entitet URED, ali će i VENTIL, BROJILO i REDUKTOR također biti specijalizirani entiteti.

Proizvođač će očitito biti zaseban entitet PROIZV.

Ključ entiteta URED određujemo iz rečenice:

'Ne postoje dva uređaja istog proizvođača koji imaju jednake tvorničke brojeve.'

→ ključ je tvornički broj, sifra proizvođača, tj (tvBroj, sifProizv)

Prema ključu entiteta URED zaključujemo da je slabiji jer sadrži ključ drugog entiteta (PROIZV).

Pošto su VENTIL, BROJILO i REDUKTOR specijalizirani entiteti entiteta URED, oni nasljeđuju isti ključ (tvBroj, sifProizv).

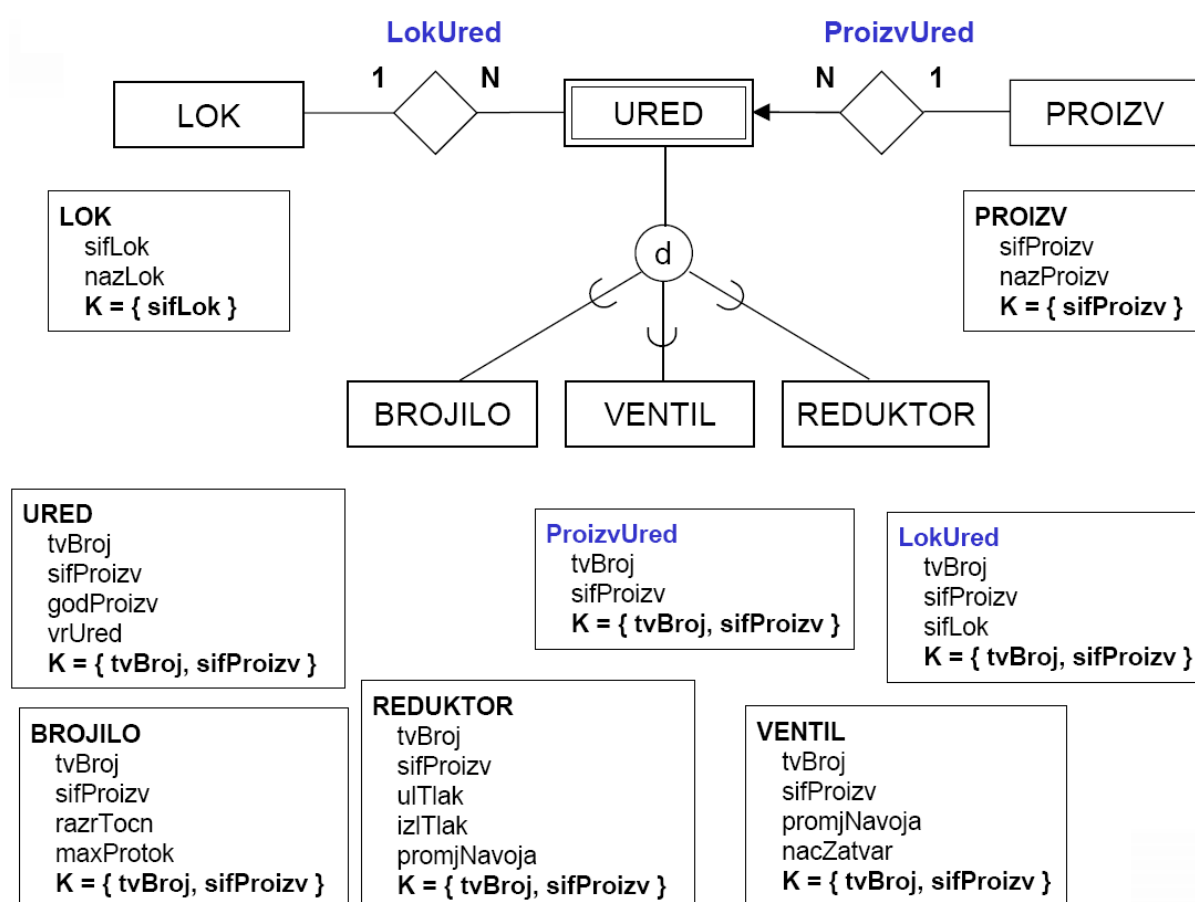
Lokacija je isto jedan zasebni entitet LOK.

Veze će biti:

LokUred – povezuje uređaj s lokacijom, ključ je opet (tvBroj, sifProizv), jer je to ključ entiteta URED koji ima 'N' na svojoj strani

ProizvUred – povezuje uređaj i proizvođača, proizvođač je vlasnik entiteta URED, ključ veze je opet (tvBroj, sifProizv), jer je to ključ entiteta URED koji ima 'N' na svojoj strani

ER dijagram:



Kod pretvaranja u relacijski model, veze 'ProizvUred' i 'LokUred' stavljamo u entitet URED jer imaju iste ključeve, i referenciramo ih na entitete koje veza u ER modelu povezuje.

Tu je samo potrebno paziti na ključeve jer ključ koji se često javlja u ovom primjeru se sastoji od dva atributa (tvBroj, sifProizv), pa da ne bi doslo do greske, oprezno s tim.

Ekvivalentan relacijski model:

```
CREATE TABLE proizv (
    sifProizv INTEGER
, nazProizv CHAR(20)
, PRIMARY KEY (sifProizv));

CREATE TABLE lok (
    sifLok INTEGER
, nazLok CHAR(40)
, PRIMARY KEY (sifLok));

CREATE TABLE ventil (
    tvBroj CHAR(20)
, sifProizv INTEGER
, promjNavoja DECIMAL(3,1)
, nacZatvar DECIMAL(5,4)
, PRIMARY KEY (tvBroj, sifProizv)
, FOREIGN KEY (tvBroj, sifProizv)
  REFERENCES ured (tvBroj, sifProizv));

CREATE TABLE ured (
    tvBroj CHAR(20)
, sifProizv INTEGER
, godProizv SMALLINT
, vrUred CHAR(1)
, sifLok INTEGER NOT NULL
, PRIMARY KEY (tvBroj, sifProizv)
, FOREIGN KEY (sifProizv) REFERENCES proizv (sifProizv)
, FOREIGN KEY (sifLok) REFERENCES lok (sifLok));

CREATE TABLE brojilo (
    tvBroj CHAR(20)
, sifProizv INTEGER
, razrToen DECIMAL(3,1)
, maxProtok DECIMAL(5,4)
, PRIMARY KEY (tvBroj, sifProizv)
, FOREIGN KEY (tvBroj, sifProizv)
  REFERENCES ured (tvBroj, sifProizv));

CREATE TABLE reduktor (
    tvBroj CHAR(20)
, sifProizv INTEGER
, ulTlak DECIMAL(6,2)
, izlTlak DECIMAL(6,2)
, promjNavoja DECIMAL(3,1)
, PRIMARY KEY (tvBroj, sifProizv)
, FOREIGN KEY (tvBroj, sifProizv)
  REFERENCES ured (tvBroj, sifProizv));
```

Primjer 4.) MODEL BAZE PODATAKA AUTOMEHANIČARSKE RADIONICE (pred 14/sljajd 19)

Pročitajte si tekst zadatka više puta i probajte skuziti što više, zajeban je malo.
U ovom primjeru je korisno primjetiti slabi entitet MODEL i ternarnu vezu N-N-1.

Slabi entitet:

Iz rečenice:

'Modeli automobila identificirani su proizvođačem i nazivom modela'
i

'Proizvođač ima naziv a identificiran je svojom šifrom'

Zaključuje se da proizvođač ima svoj entitet, model također, ali će model biti slabi entitet čiji je vlasnik entitet proizvođača, jer u ključu entiteta modela će biti i sifra proizvođača.

Ternarna veza N-N-1:

Iz teksta:

'Poslove koji su zadani na radnom nalogu može obaviti jedan ili nekoliko mehaničara, ali jedan zadani posao će jedan mehaničar obaviti sam od početka do kraja. Mehaničari na raznim Radnim nalogima mogu obavljati različite vrste poslova.'

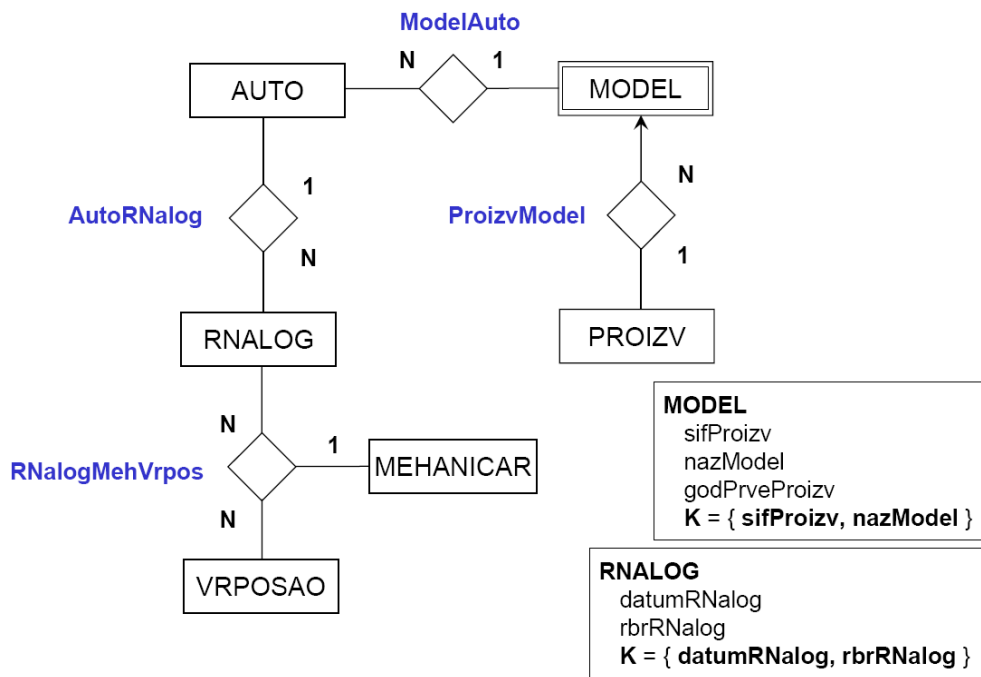
Oće reć,

za jedan radni nalog jednu vrstu posla će obaviti jedan mehaničar

jednu vrstu posla jedan mehaničar može obavljati za više radnih naloga

jedan mehaničar za jedan radni nalog može obavljati više vrsta poslova

Dobivamo ternarnu vezu N-N-1 čiji će ključ biti oni ključni atributi entiteta kdo kojih je 'N', a to su entiteti RNALOG i VRPOSAO, tj atributi (datumRNalog, rbrRNalog, sifVrstaPos)



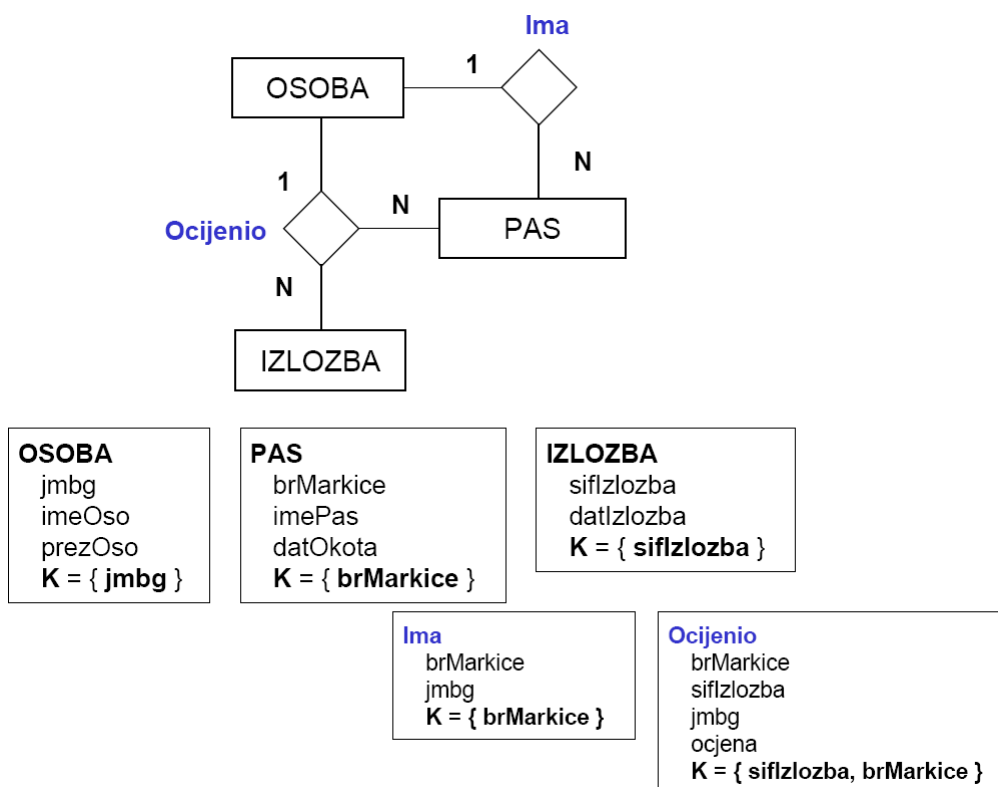
Primjer 5.) MODEL BAZE PODATAKA ZA IZLOŽBE PASA (pred 14/slajd 22)

Opet ternarna veza N-N-1, nešto jednostavniji zadatak od prošlog.

Ternarna veza se čita iz teksta:

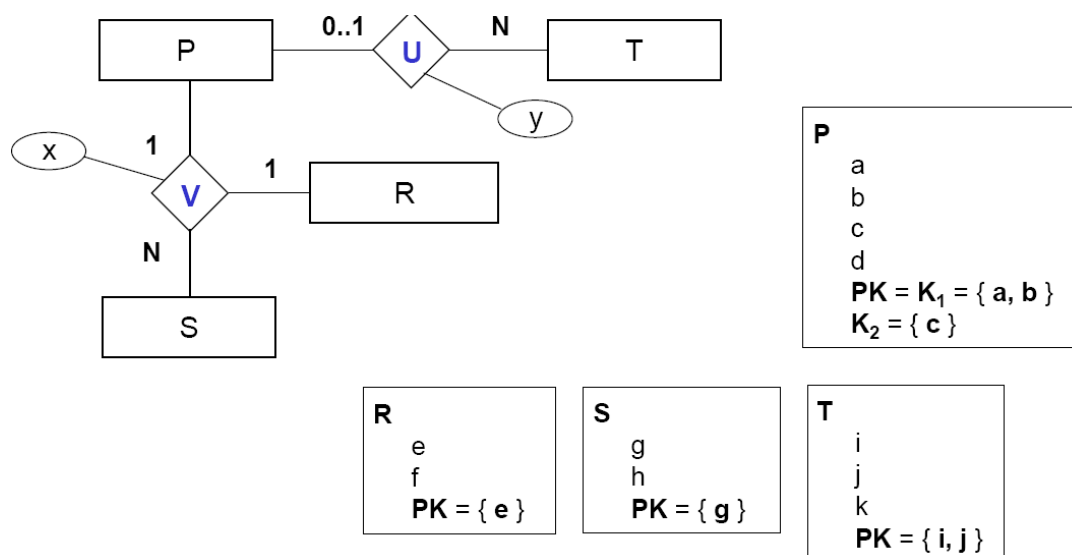
'Za jednog psa na jednoj izložbi treba evidentirati samo jednu ocjenu i osobu koja ga je ocjenjivala. Ista osoba na jednoj izložbi može ocijeniti više pasa. Ista osoba može ocjenjivati istog psa na više različitih izložbi'

Kod ovog primjera je posebno to što ta ternarna veza sadrži i vlastiti atribut 'ocjena'. To je zato jer 'ocjena' ne opisuje nijedan entitet isključivo, već ima veze sa sva tri entiteta.



Primjer 6.) ZADANI ER MODEL (pred 14/slajd 24)

Ovdje je zadan ER model, te se traži opisivanje entiteta i veza, te, nešto važnije, prebacivanje u relacijski model. Zašto je to važnije? E pa zato jer još nismo prebacivali N-1-1 ternarnu vezu u relacijski model, a to ima svojih par caka. Rekli smo prije nekad davno da u N-1-1 ternarnoj vezi ključ se može definirati na 2 načina, zato kad odaberemo jedan način, moramo osigurati onaj drugi način UNIQUE naredbom.

Prebacivanje u relacijski model

Dakle ključ novog entiteta V mogu biti

ključevi entiteta S i P, tj (g, a, b) ili

ključevi entiteta S i R, tj (g, e)

Mi ćemo odabrati za ključ entiteta V atribut (g, a, b)

Kad bi samo stavili

PRIMARY KEY (g,a,b)

za jednu kombinaciju (g, a, b) možemo dobiti više (e), što nije dozvoljeno u zadatku. Zato ćemo se osigurati sa

UNIQUE (e, g)

Isto tako kod entiteta P imamo dva potencijalna ključa (a,b) i (c) te nakon što jednog postavimo za PRIMARY KEY mora osigurati da je i drugi atribut jedinstven sa UNIQUE

Va
b
e
g
x $PK = K_1 = \{a, b, g\}$ $K_2 = \{e, g\}$ **U**i
j
a
b
y $PK = \{i, j\}$

Relacijski model

```
CREATE TABLE p (
  a ...
, b ...
, c ...
, d ...
, PRIMARY KEY (a, b)
, UNIQUE (c));
```

```
CREATE TABLE r (
  e ...
, f ...
, PRIMARY KEY (e));
```

```
CREATE TABLE s (
  g ...
, h ...
, PRIMARY KEY (g));
```

```
CREATE TABLE t (
  i ...
, j ...
, k ...
, y ...
, a ...
, b ...
, PRIMARY KEY (i, j)
, FOREIGN KEY (a, b)
  REFERENCES p (a, b));
```

```
CREATE TABLE v (
  a ...
, b ...
, e ...
, g ...
, x ...
, PRIMARY KEY (a, b, g)
, UNIQUE (e, g)
, FOREIGN KEY (a, b)
  REFERENCES p (a, b)
, FOREIGN KEY (e)
  REFERENCES r (e)
, FOREIGN KEY (g)
  REFERENCES s (g));
```


Primjer 7.) ZADANI ER MODEL (pred 14/slajd 26)

To ćete sami skužiti, jedina novost je da je u pitanju refleksivna veza pa se pri prebacivanju u relacijski model mora jedan atribut preimenovati