

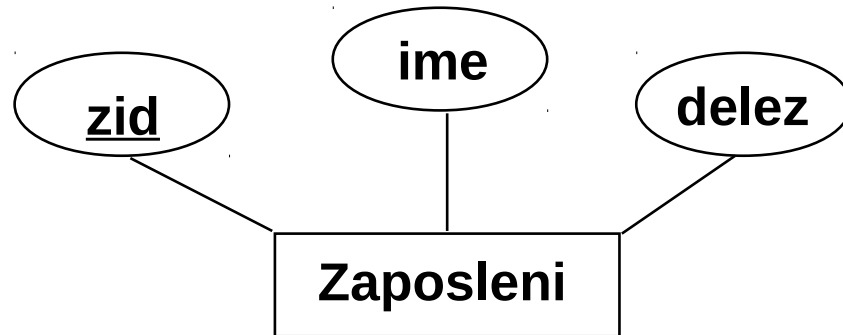
# PODATKOVNI MODEL ENTITETA-RAZMERJE

Iztok Savnik

# Osnovni elementi ER

- Entiteta
- Razmerje
- Atributi
- Odvisne entitete
- Identifikator
- Specializacija/generalizacija
- Agregacija/dekompozicija
- Vir: Ragu Ramakrishnan, DBMS

# Entitete



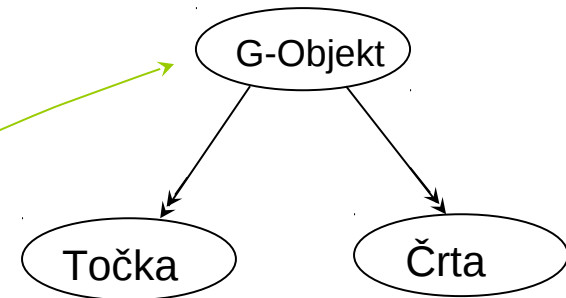
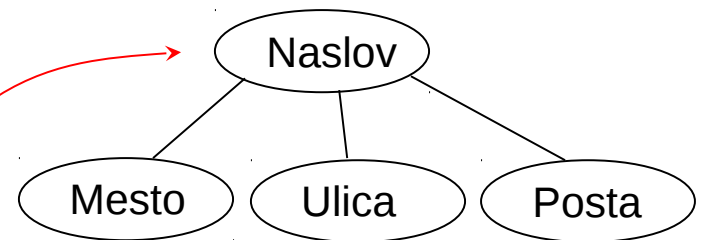
- Entiteta: Objekt iz realnega sveta, ki ga lahko ločimo od ostalih objektov. Entiteta je podatkovni bazi predstavljena z množico atributov (lastnosti).

# Entitete - formalno

- **Entitetne množice** :  $E, E_1, E_2$ 
  - $\Pi[E]=\{e_1, e_2, \dots, e_n\}$ ;  $e_1, e_2, \dots, e_n$  so entitete
- Vsaka entiteta ima **identifikator** s katerim jo lahko enolično identificiramo.
  - Identifikator entitete tvori en ali več atributov katerih vredost je unikatna znotraj entitetne množice te entitete.
  - Kandidatni ključi, ki enolično določa n-terice v neki relaciji.
  - Več identifikatorjev entitet neke entitetne množice.
  - Primarni identifikator je odločitev načrtovalca.
- Preostale attribute entitete, ki niso del primarnega identifikatorja imenujemo **opisni atributi**.

# Atributi

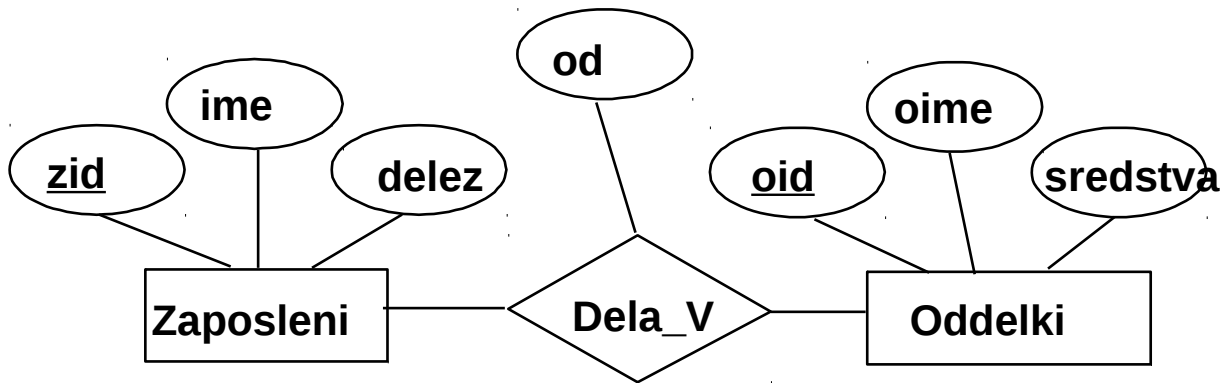
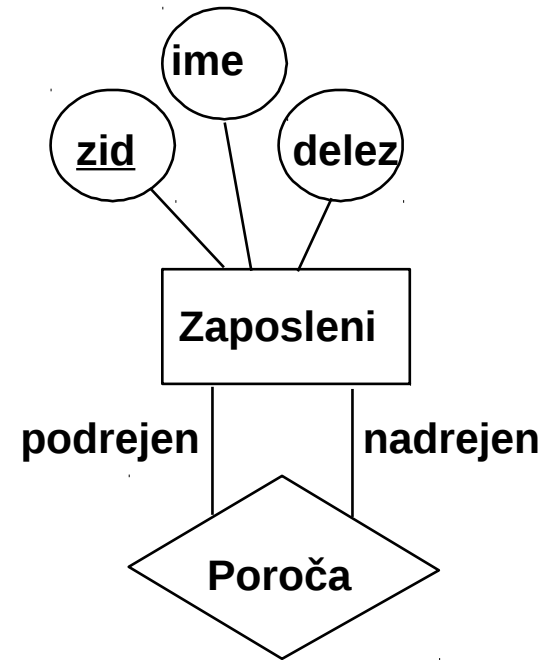
- Lastnosti entitet predstavimo z atributi.
  - Atribut je podatkovni element s katerim opišemo eno lastnost entitete.
- Vsak atribut ima **zalogo vrednosti**:
  - Določa dovoljene vrednosti posameznega atributa.
  - **Enostavni atributi**: enostavne vrednosti kot so na primer cela števila in nizi,
  - **Sestavljeni atributi**: vrednosti sestavljene iz več enostavnih vrednosti, ki so lahko različnega tipa – te imenujemo sestavljene vrednosti, ali
  - **Večvrednostni atributi**: množice vrednosti, ki so vse istega tipa.



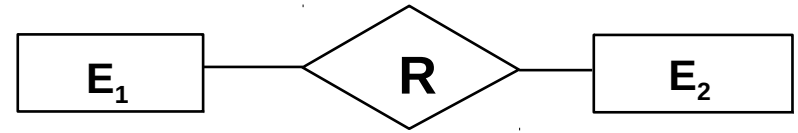
# Atributi

- Sestavljeni in večvrednostni atributi omogočajo bolj **abstraktno predstavitev** neke lastnosti entitete
  - Lastnost, ki bi jo sicer morali predstaviti z več enostavnimi atributi ali celo z razmerjem, predstavimo z enim samim konceptom
- Samo nekateri praktični modeli uporabljajo sestavljene in večvrednostne attribute
  - Extended Entity-Relationship Model
  - Večina sistemov nima teh atributov
  - **Praktična osnova v SQL3 !**

# Primer ER shem



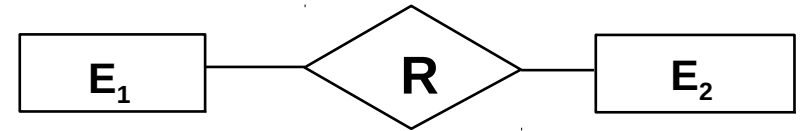
# Razmerja



- Razmerje definira **povezavo med dvema ali več entitetami**, ki lahko pripadajo različnim entitetnim množicam.
  - Na primer, razmerje Lastnik definira povezave med entitetami iz entitetnih množic Stranke ter Račun.
  - Konkretno razmerje lahko opišemo kot par entitet na primer  $(s, r)$ , kjer je  $s$  član entitetne množice Stranka,  $r$  pa član entitetne množice Račun.
- Razmerje je **n-terica**  $(e_1, e_2, \dots, e_n)$ , kjer predstavljajo oznake  $e_i$  entitete.

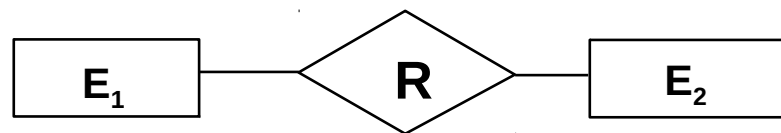


# Interpretacija



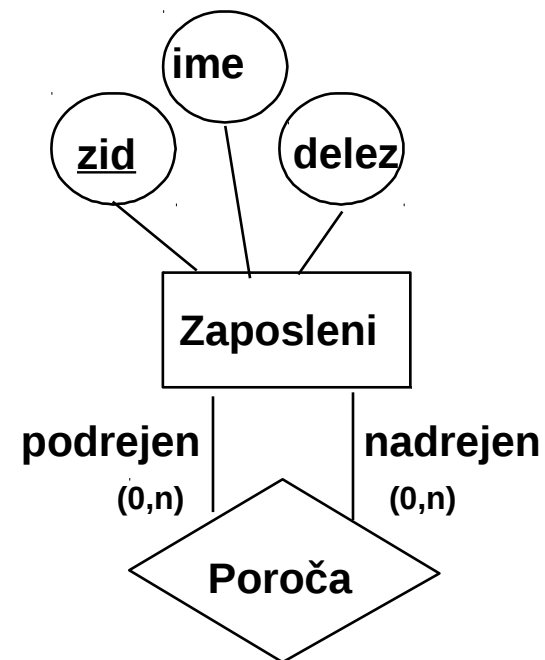
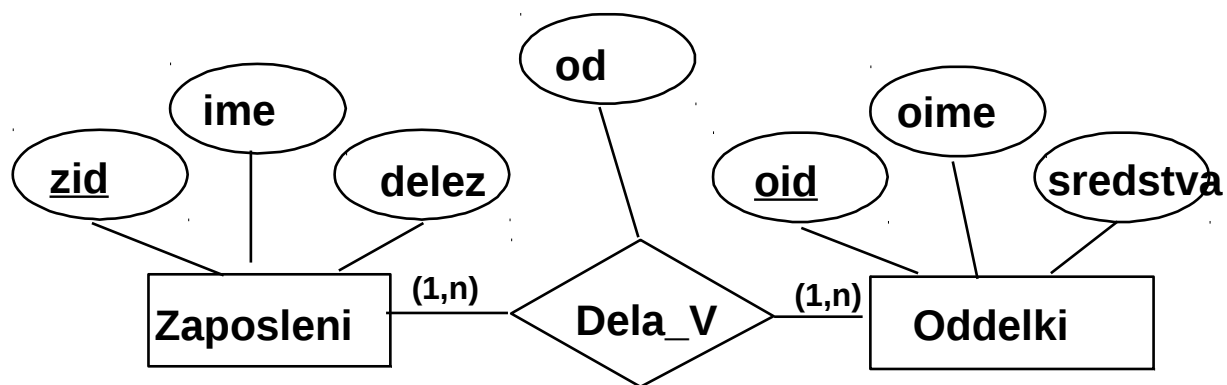
- **Množica razmerja:**
  - Razmerja pogosto klasificiramo v množice, ki vsebujejo podobna razmerja.
  - $\Pi[R] = \{(e_1, \dots, e_n) \mid e_1 \in E_1, \dots, e_n \in E_n\}$ , kjer so  $E_i$  entitetne množice.
  - $R \subseteq E_1 \times \dots \times E_n$
- Število entitet v razmeju  $n$  imenujemo tudi **stopnja razmerja**.
- Razmerje med dvema entitetama imenujemo binarno razmerje.

# Razmerja



- Množico razmerja predstavimo z grafičnim simbolom diamant, ki je povezan z Entitetnimi množicami s črtami.
  - Črto, ki povezuje entiteto in razmerje lahko tudi poimenujemo.
  - Ime predstavlja **vloga entitete v razmerju**.
  - Povezavo med entitetno množico in množico razmerja imenujemo tudi vloga saj opisuje vlogo, ki jo imajo entitete v razmerju.
- Binarno razmerje, ki povezuje dve entiteti iz iste entitetne množice imenujemo tudi **rekurzivno**
  - Rekurzivno razmerje - želimo eksplicitno ločiti med različnimi vlogami, ki jih imajo entitete v razmerju.
  - Naslednji zgled predstavlja primer rekurzivnega razmerja iz konceptualne sheme bančnega okolja.

# Primer



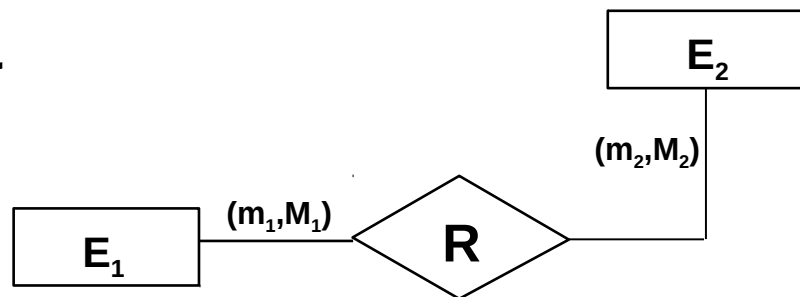
- Vloga razmerja
  - Domena je lahko ista entiteta

# Števnost razmerja

Razmerje lahko bolj natančno opišemo s števnostjo razmerja,

Razmerje samo ne govori o tipu preslikave

Števnost preslikave med entitetnimi množicami razmerja.



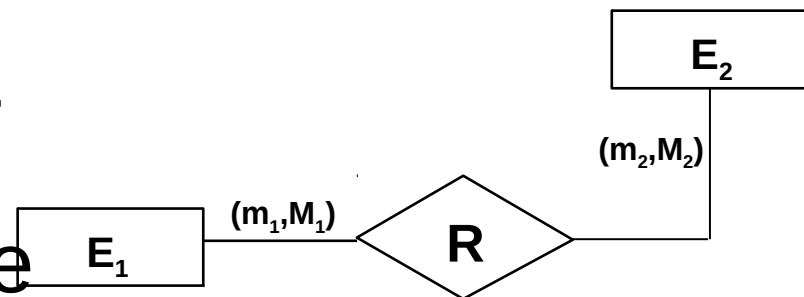
Dane imamo entitetne množice  $E_1, E_2, \dots, E_n$ , ki so povezane z razmerjem  $R$ .

Števnost definiramo za vsako posamezno entitetno množico, ki sodeluje v razmerju  $R$ .

S števnostjo entitetne množice  $E_i$  v razmerju  $R$  povemo v kolikih različnih razmerjih lahko sodeluje entiteta iz entitetne množice  $E_i$ .

# Števnost razmerja

Števnost entitetne množice



$E_i$  v razmerju  $R$  zapišemo kot funkcijo:

$$\text{card}(E_i, R) = (\text{min}, \text{max})$$

min predstavlja **minimalno števnost** entitetne množice  $E_i$  v razmerju  $R$ ,

max pa predstavlja **maksimalno števnost**  $E_i$  v razmerju  $R$ .

Vrednost minimalne i maksimalne števности:

“0” (nič), “1” (ena) “N” (“N” beremo “mnogo” ali “več”, kar v splošnem pomeni več kot ena).

min-card( $E, R$ ) in max-card( $E, R$ )

# Vrste razmerja glede na števnost

Vrste razmerij, ki so določena glede na števnost entitet v razmerju.

Vloge entitetne množice E v razmerju R:

$\text{max-card}(E,R) = 1$  - E ima **eno-vrednostno vlogo** v razmerju R.

$\text{max-card}(E,R) = N$  - entiteta E ima v razmerju R **večvrednostno vlogo**.

Binarno razmerje R med entitetnima množicama E in F označimo:

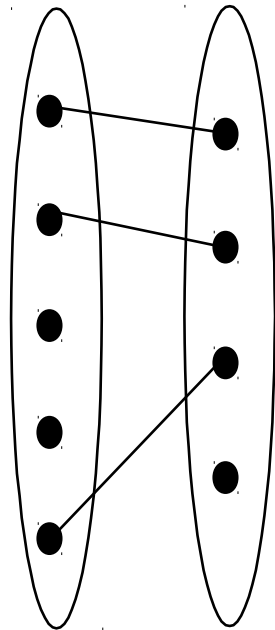
**N-N** -- mnogo-proti-mnogo -- če sta obe E in F večvrednostni v razmerju R.

**1-1** -- ena-proti-ena -- če sta obe E in F v razmerju R enovrednostni

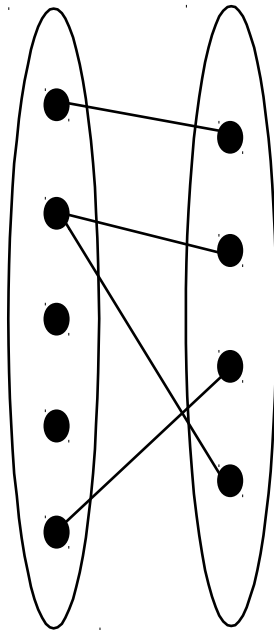
**1-N (N-1)** – ena-proti-mnogo -- če ima ena izmed entitetnih množic enovrednostno vlogo in druga večvrednostno vlogo v razmerju R

Pri klasifikaciji razmerij v tipe “1-1”, “1-N” in “N-N” smo uporabljali izključno samo funkcijo max-card!

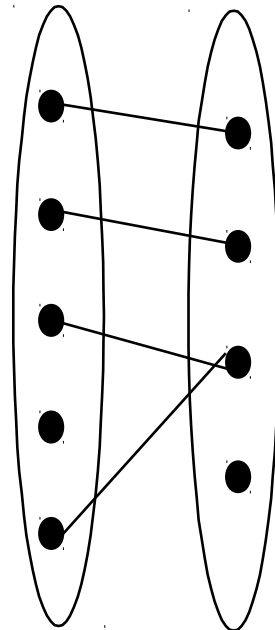
# Vrste binarnih razmerij



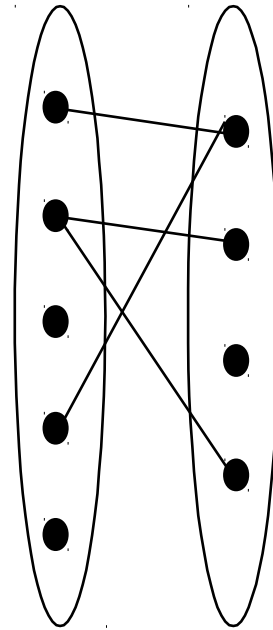
1--1



1--N



N--1



N--N

# Vrste razmerja glede na števnost

Minimalna števnost nam lahko služi za drugo vrsto klasifikacije razmerij: **obveznost**

$$\text{min-card}(E,R) = 1$$

Za vsako entiteto iz množice E mora obstajati vsaj eno razmerje iz množice razmerja R, ki vsebuje to entiteto.

Entitete iz entitetne množice E v razmerju R so **obvezne**.

$$\text{min-card}(E,R) = 0$$

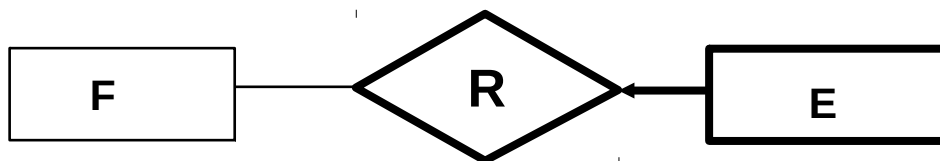
Ni nujno, da vsaki entiteti iz množice E pripada eno razmerje iz množice razmerja R.

V tem primeru so entitete iz entitetne množice E v razmerju R **opcijske**.



# Šibke entitete

Naj bosta E in F entitetni množici, ter R razmerje, ki povezuje entitete iz množic E in F.



Entitetna množica **E je šibka**, če je obstoj entitet iz entitetne množice E pogojen z obstojem entitet iz entitetne množice F.

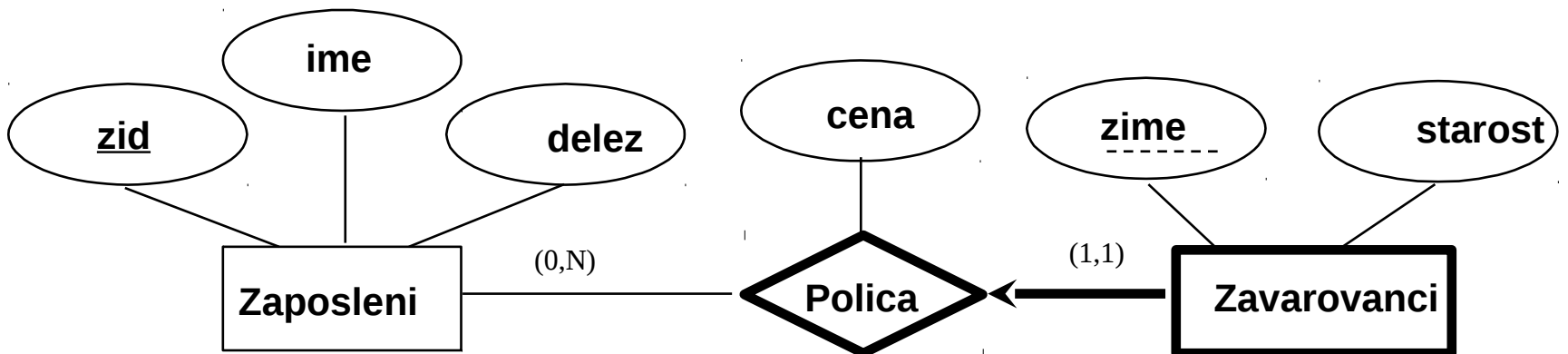
Z drugimi besedami, entiteta iz množice E lahko obstaja samo, če obstaja pripadajoča entiteta iz množice F.

Identifikator entitetne množice E je tako relevanten samo v okviru povezave z F.

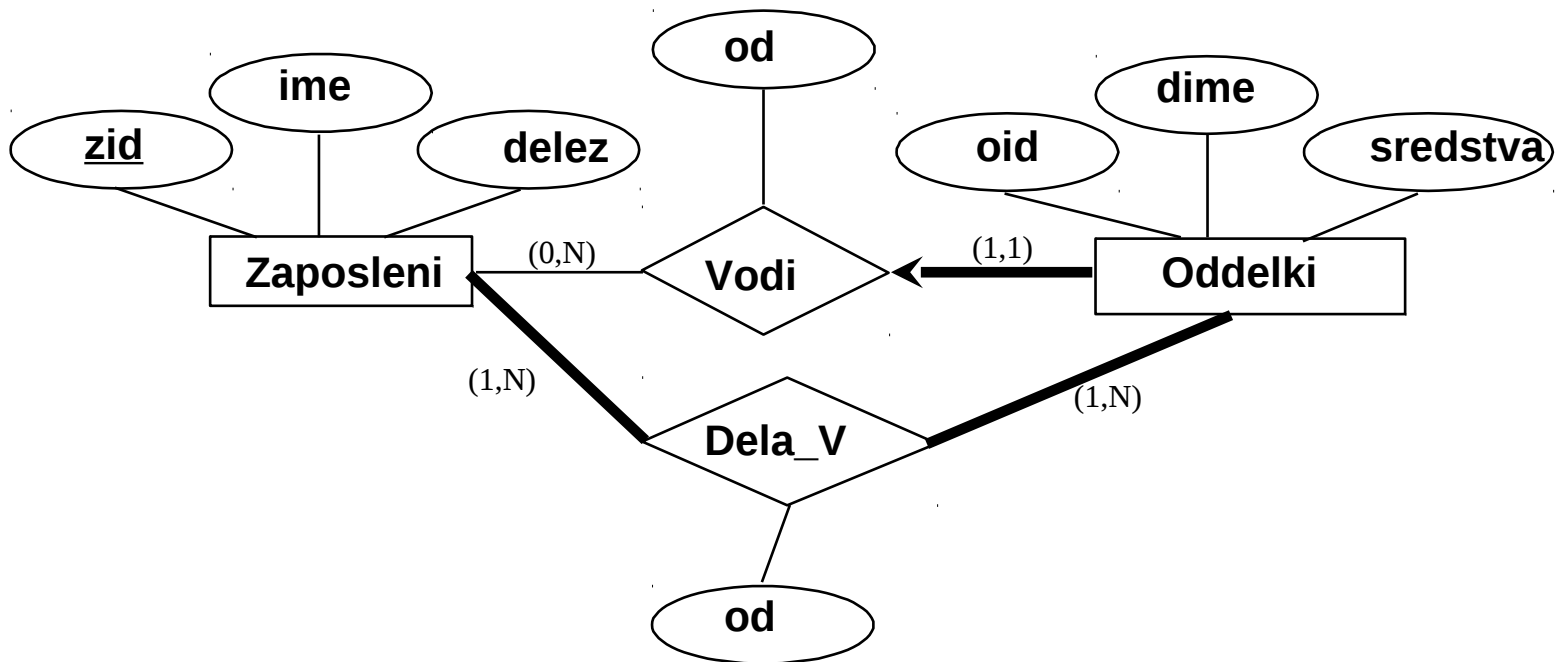
Entitetno množico F v tej vlogi imenujemo **močna entitetna množica**.

# Šibka entiteta

- **Šibko entiteto** lahko identificiramo samo s pomočjo primarnega ključa neke druge (*lastnik*) entitete.
  - Entitetna množica lastnika in šibka entiteta morajo biti povezani v razmerju tipa **ena—več** (en lastnik, več šibkih entitet).
  - Šibka entiteta mora imeti **obvezno članstvo**, lastnik je del ključa šibke entitete.



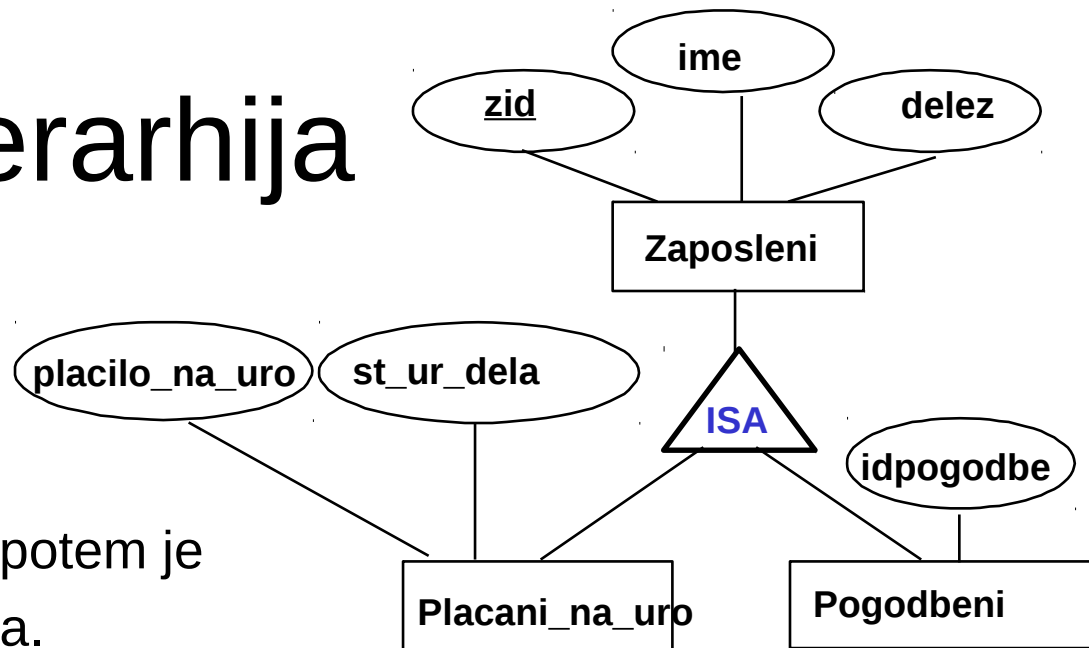
# Primer



(1,1)

# ISA ('is a') Hierarhija

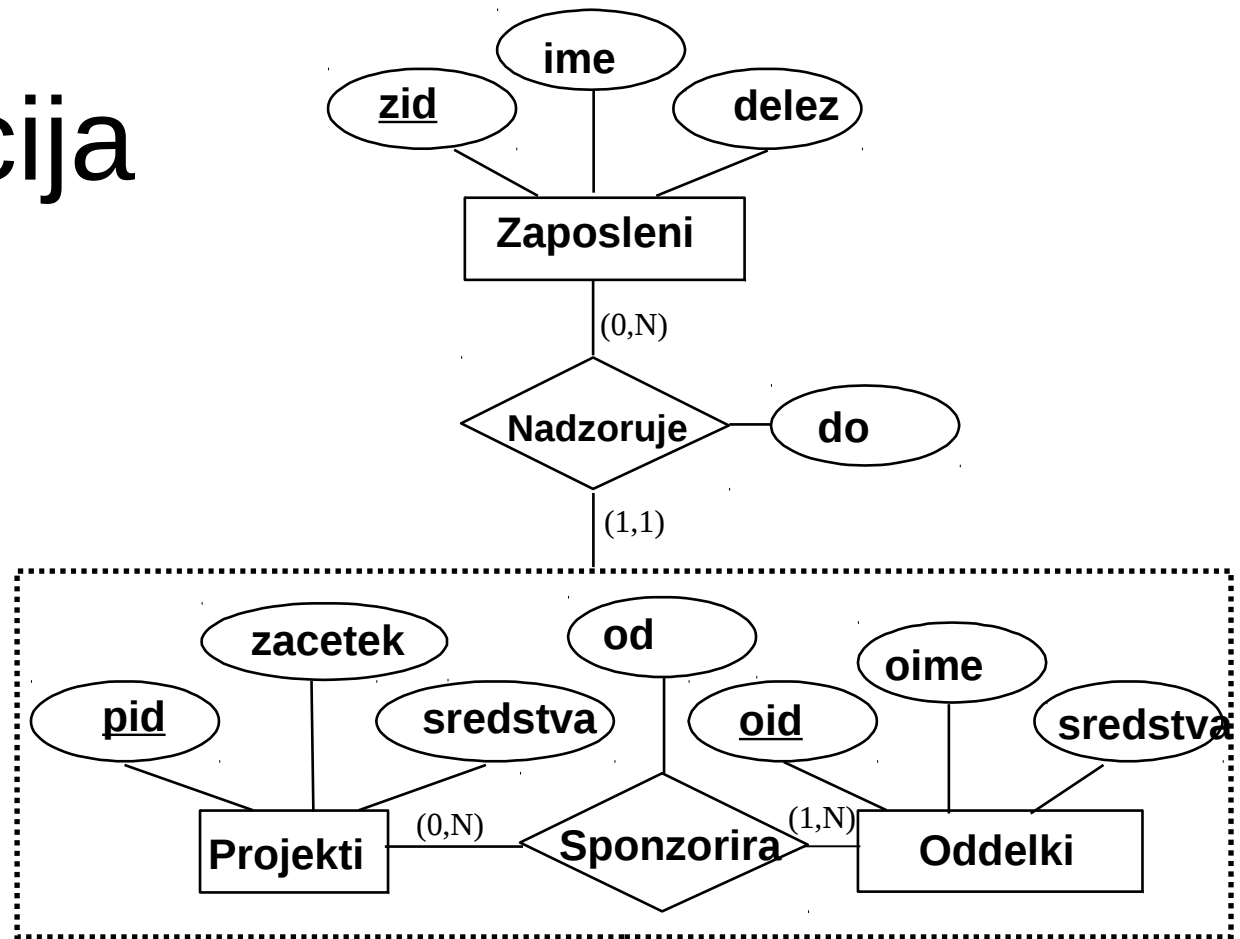
- ❖ Kot v C++, ali ostalih PL; atributi se dedujejo.
- ❖ Če deklariramo, da A **ISA** B, potem je vsaka A entiteta tudi B entiteta.



- **Prekrivanje pod-entitetnih množic:** Je lahko Tone v množici Placan\_na\_uro kot tudi v množici Pogodbeni? (*Da/ne*)
- **Pokrivanje nad-entitetne množice:** Mora vsak zaposleni nujno biti član tudi ene izmed podrejenih entitet? (*Da/ne*)
- Razlogi za uporabo ISA:
  - Dodajanje opisnih atributov direktno k določenem podrazredu.
  - Identificiranje entitet, ki sodelujejo v razmerju.

# Agregacija

- Uporablja se ko moramo modelirati razmerje, ki povezuje množice razmerij.
  - *Agregacija* nam omogoča, da obravnavamo množice razmerij kot entitetne množice, ki lahko sodelujejo v drugih razmerjih.



## ☛ *Agregacija vs. razmerje:*

- ❖ *Nadzoruje* je razmerje, ki ima opisni atribut.
- ❖ Lahko tudi rečemo, da je vsako sponzorstvo nadzorovano z največ enim zaposlenim.

# Druge notacije ER

- Chenova notacija
- Vranja notacija
- Generalizacija/specializacija
- Agregacija/dekompozicija
- Omejitve
- Vir
  - Toby Teory: Database design

# Entiteta

- Razred objektov iz realnega sveta, ki imajo skupne značilnosti in lastnosti o katerih želimo shranjevati informacije.

# Razmerje

- Asociacija med dvema ali večimi entitetami.



# Razmerje

- primerek – instanca razmerja je kolektivna instanca vseh povezanih entitet
- stopnja – število entitet povezanih v razmerje (binarno, ternarno, n-arno)
- števnost – ena-ena, ena-več, več-več
- obstoj razmerja (omejitev) – opcijska / obvezna

# Atribut

- Značilnost entitete ali razmerja

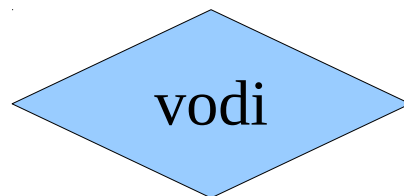
# Atribut

- identifikator – enolično določa instanco entitete
- odvisnost identitete – del identifikatorja je prevzet od druge entitete
- več-vrednosten – isti atribut, ki ima več vrednosti za eno entiteto
- nadomestni identifikator – sistemsko kreiran in kontroliran unikaten ključ (npr. “create sequence” v Oracle)

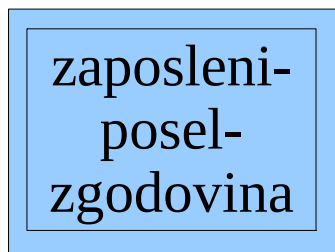
# Notacija



entiteta



razmerje



šibka entiteta



identifikator



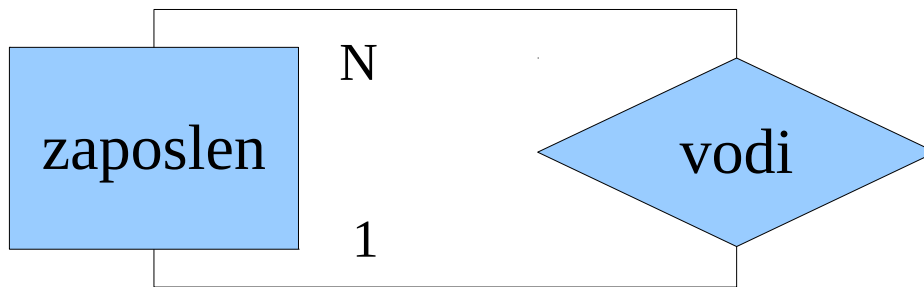
opisni  
atribut

več-vrednostni  
atribut

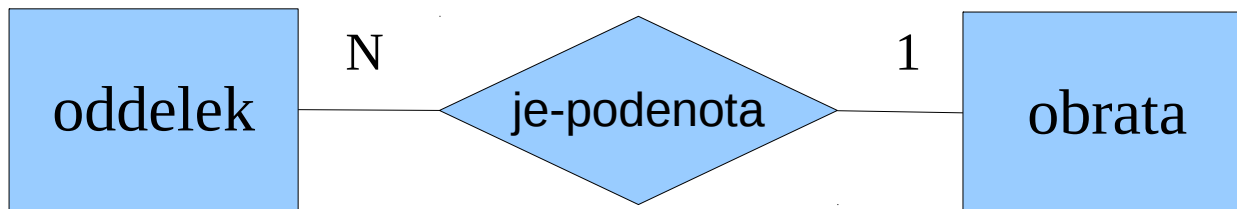
kompleksni  
atribut



# Stopnja razmerja

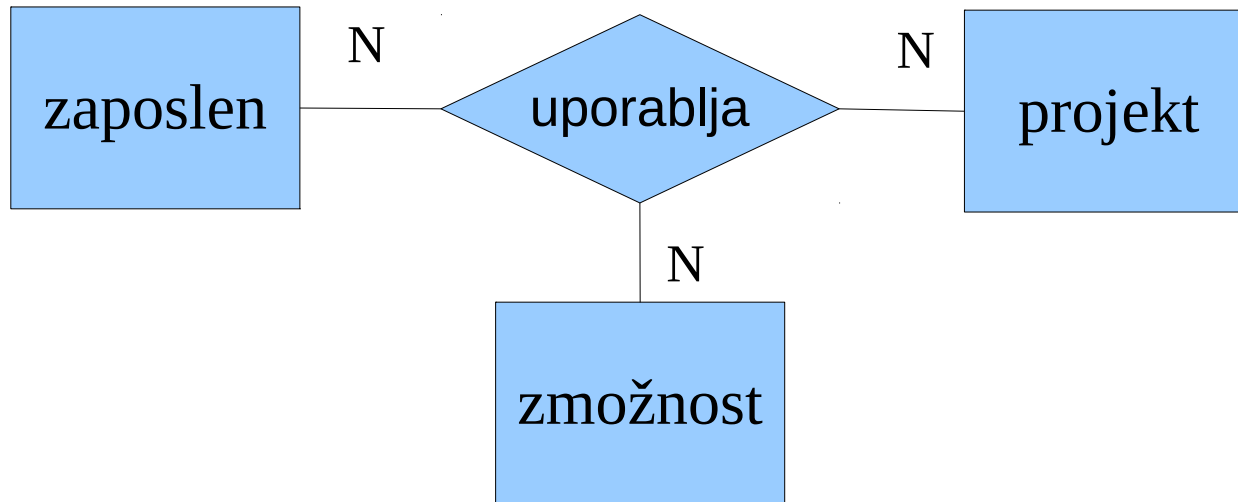


Rekurzivno  
binarno razmerje



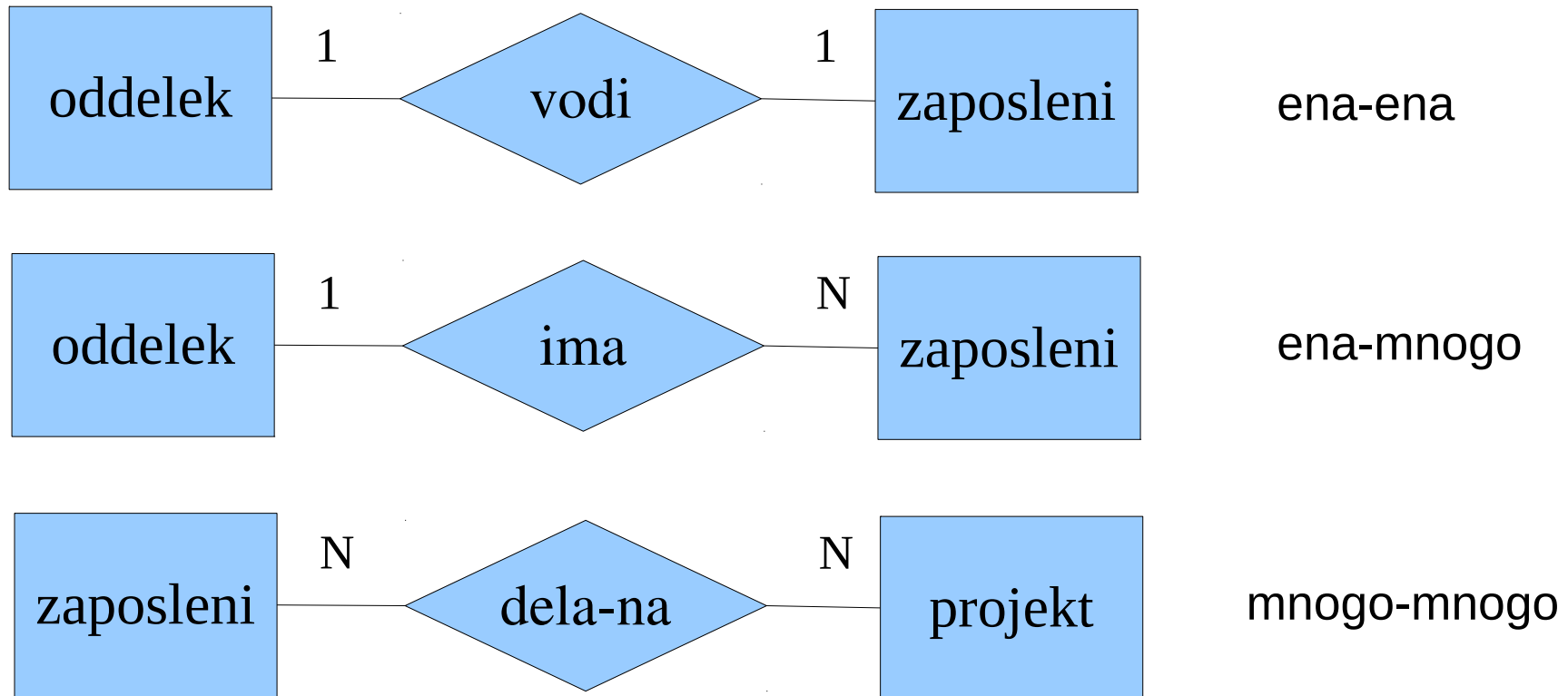
Binarno razmerje

# Stopnja razmerja

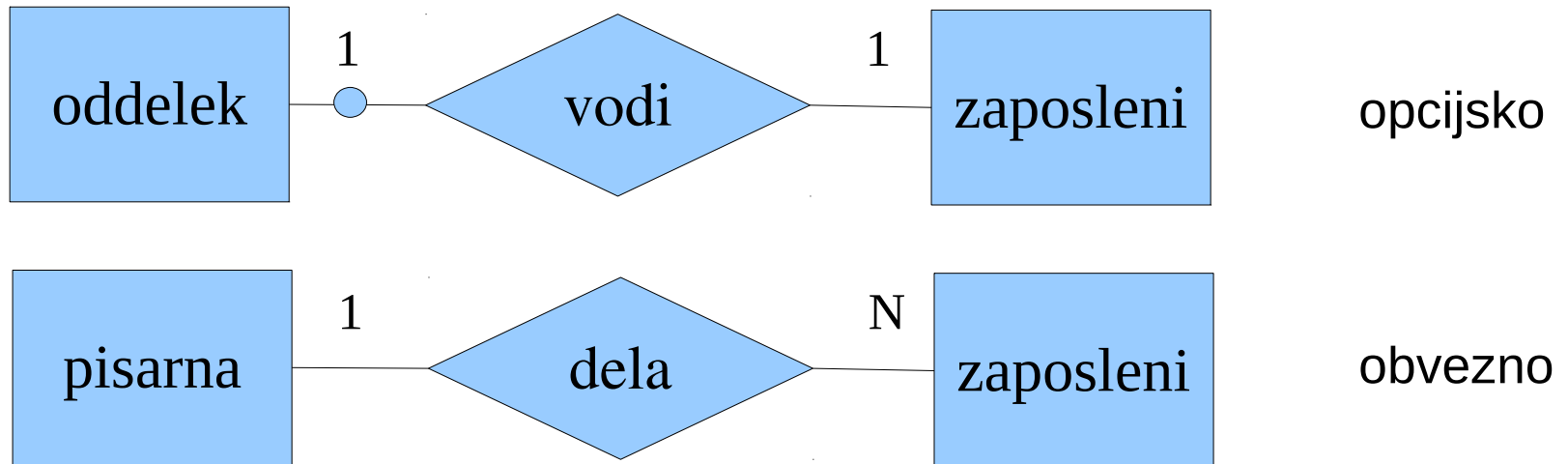


Ternarno razmerje

# Števnost razmerja

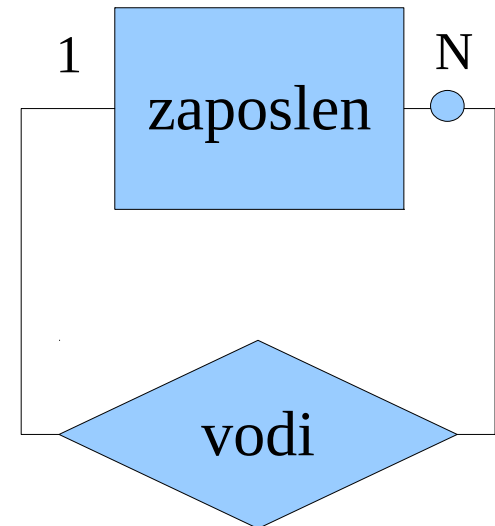
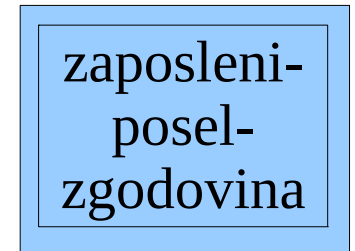
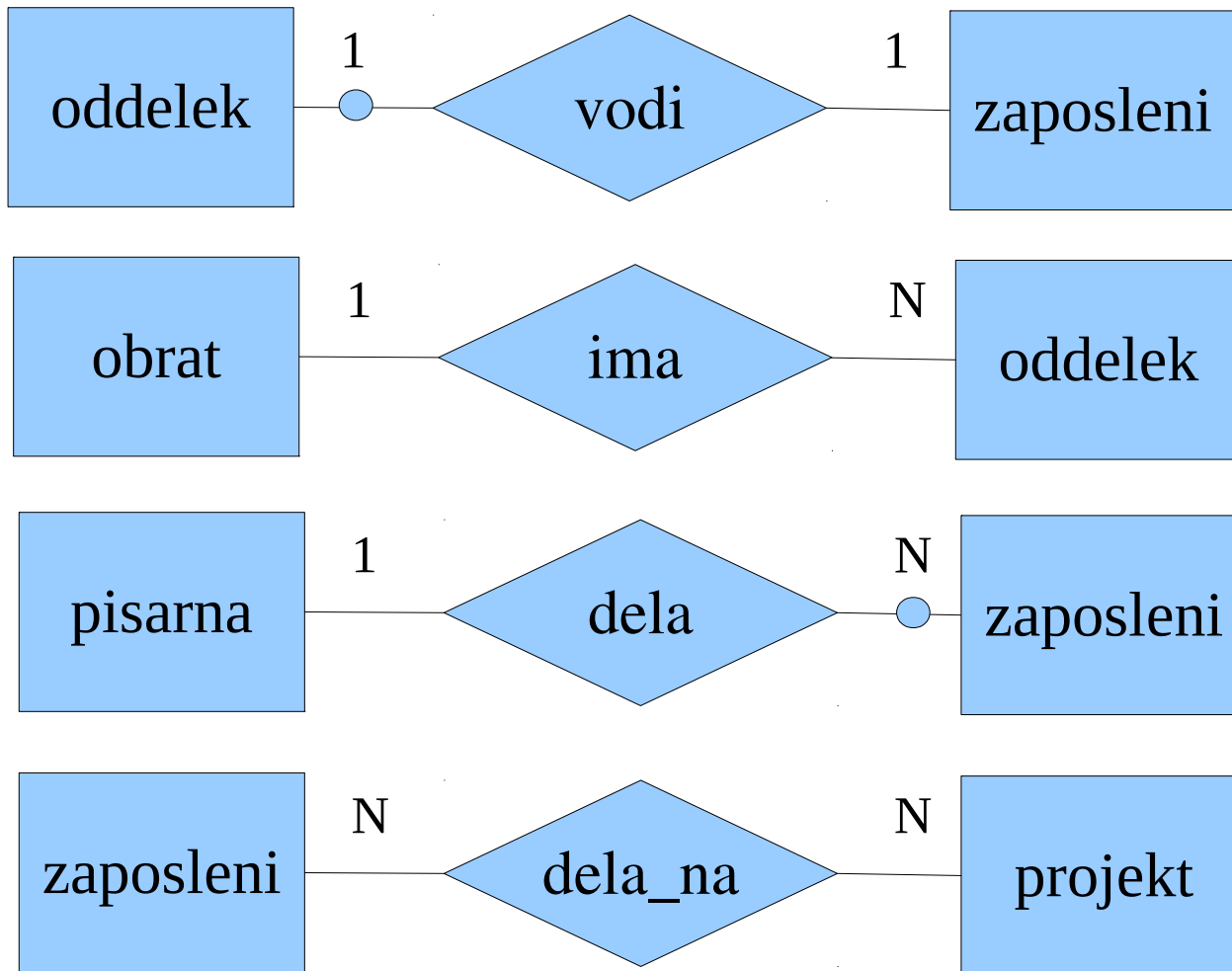


# Obstoj razmerja

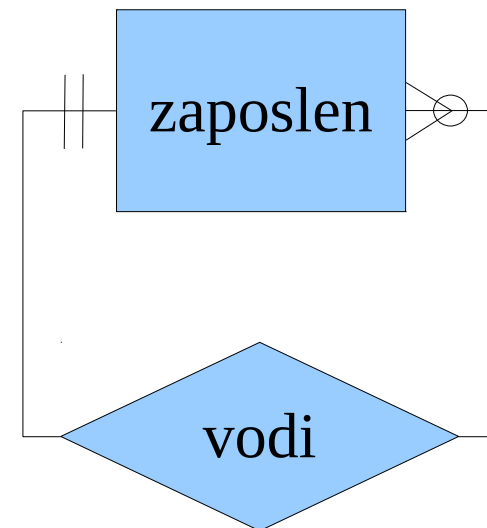
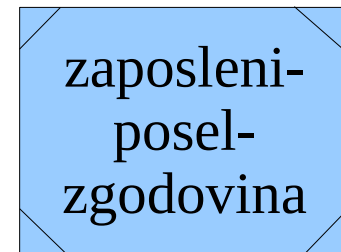
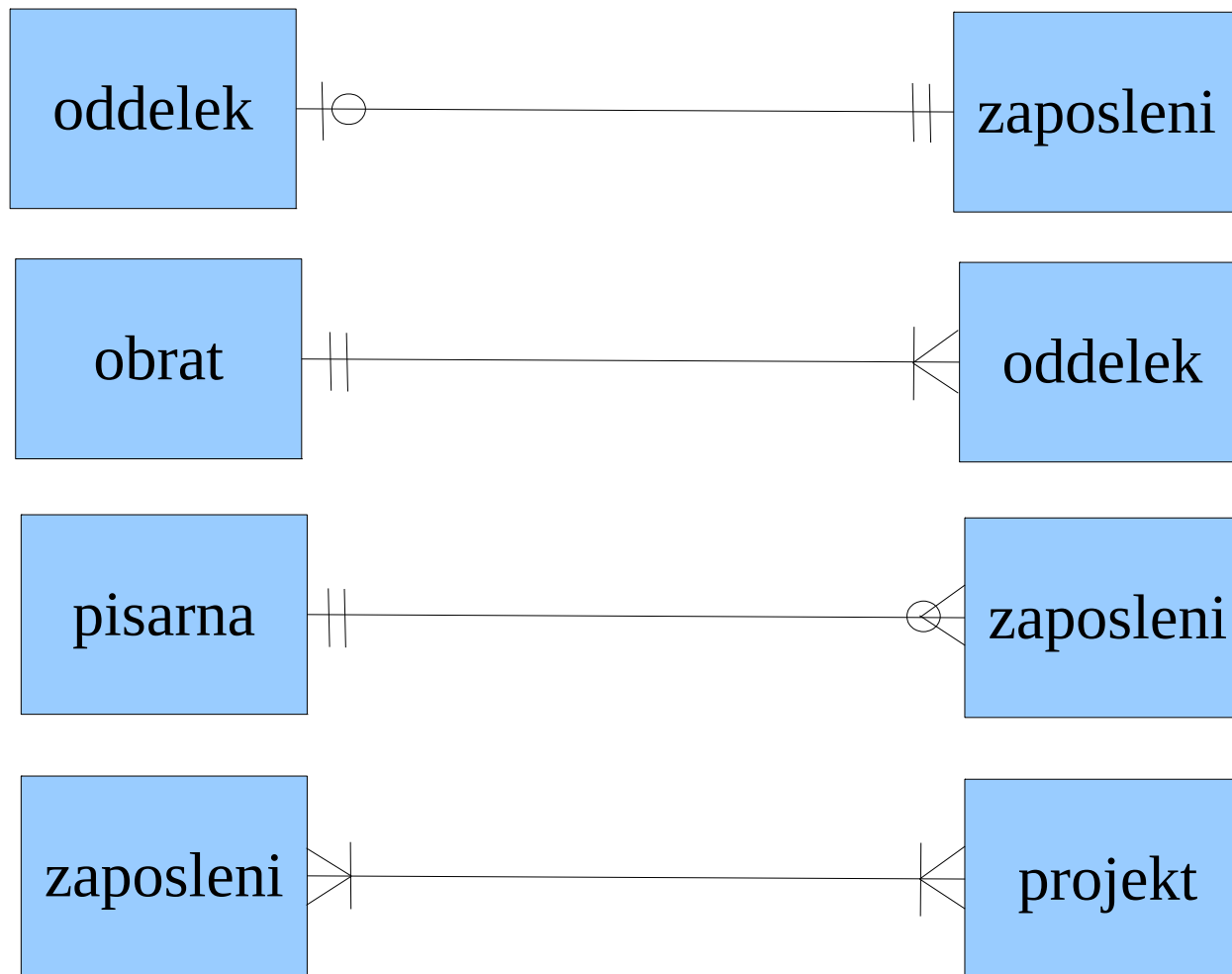




# Chenova ER notacija



# Vranja ER notacija



# Generalizacija/specializacija

- Podobnosti so posplošene na nivo entitete super-razreda.
- Razlike so specializirane na entitete podrazredov.
- Relacijo med entitetama pod in nad razredov imenujemo “ISA” relacija.

# Generalizacija/speciaizacija

- omejitev prekrivanja – ni prekrivanja med pod-razredi
- omejitev kompletnosti – omeji pod-razrede na kompletno pokrivanje nad-razreda ali ne
- totalno ali delno pokrivanje nad-razredov

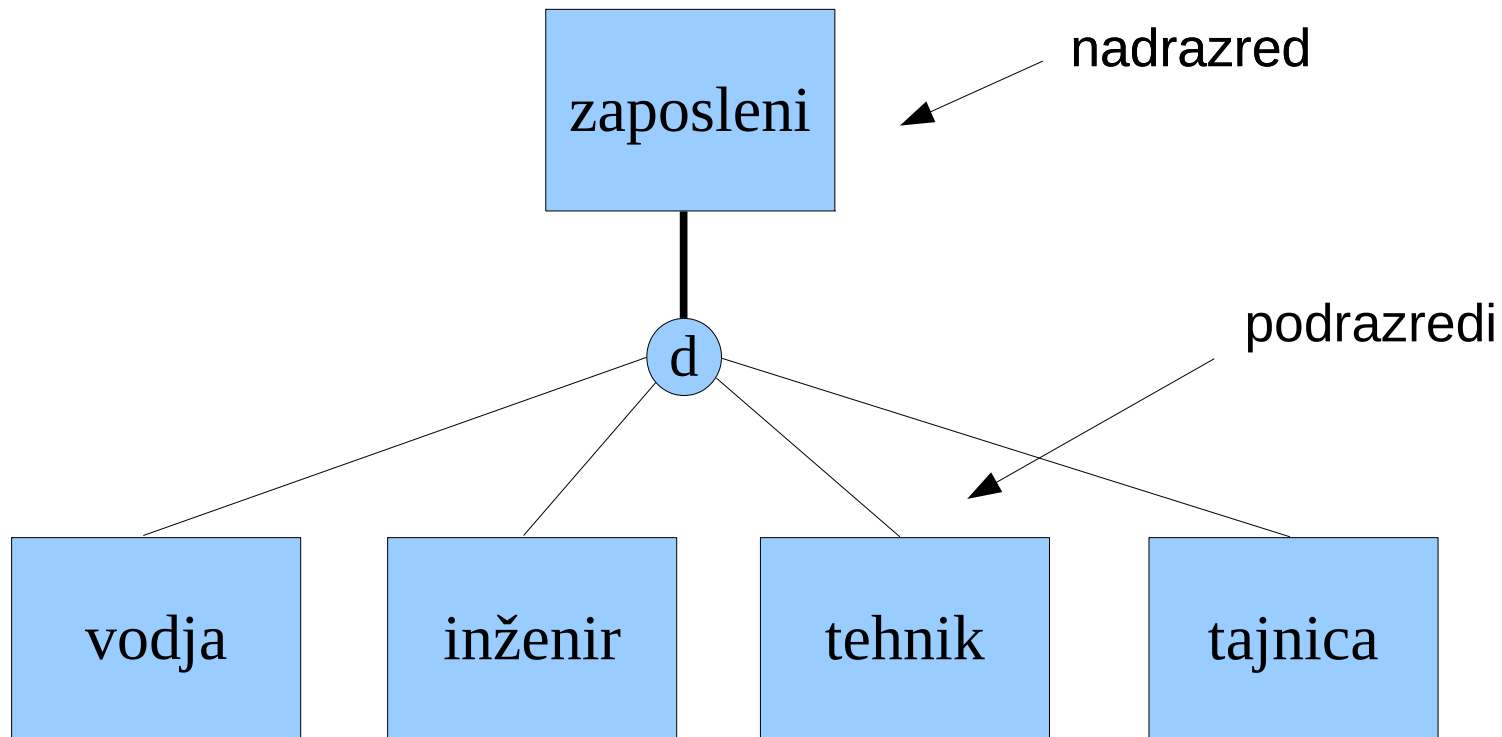
# Generalizacija/speciaizacija

- relacija ISA: hirarhična po naravi
- relacija ISA: dedovanje – podrazred podeduje primarni ključ nadrazreda
- skupni opisni atributi se dedujejo
- vsak podrazred ima svoje opisne attribute

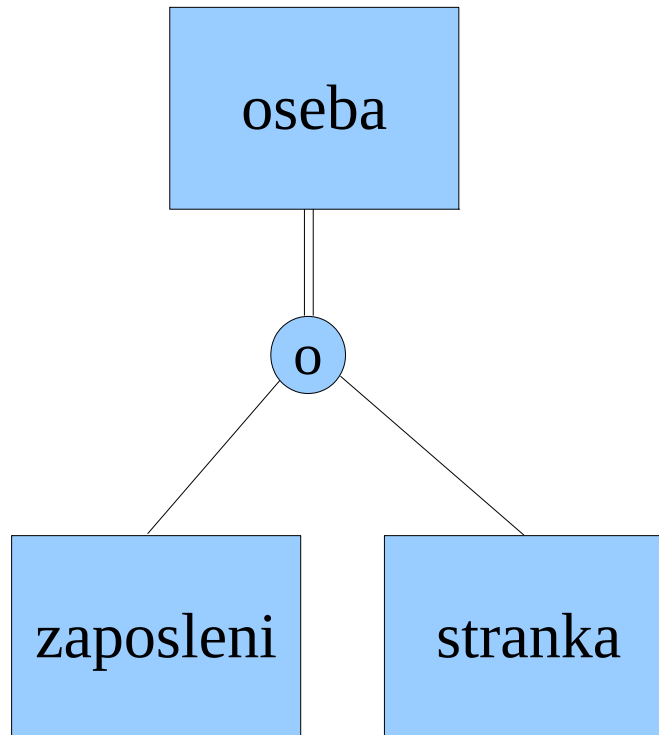
# Agregacija

- relacija kompozicija / dekompozicija
- relacija entitet do nadrejenih agregacijskih entitet
- razmerje imenujemo “vsebuje”
- atributi, ki sestavljajo agregacijsko entiteto: npr. mesec-dan-leto
- agregacija entitet: članstvo ali relacija „je-član“

# Generalizacija z neprekrivajočimi pod-razredi



# Generalizacija s prekrivajočimi podrazredi in kompletnim pokrivanjem





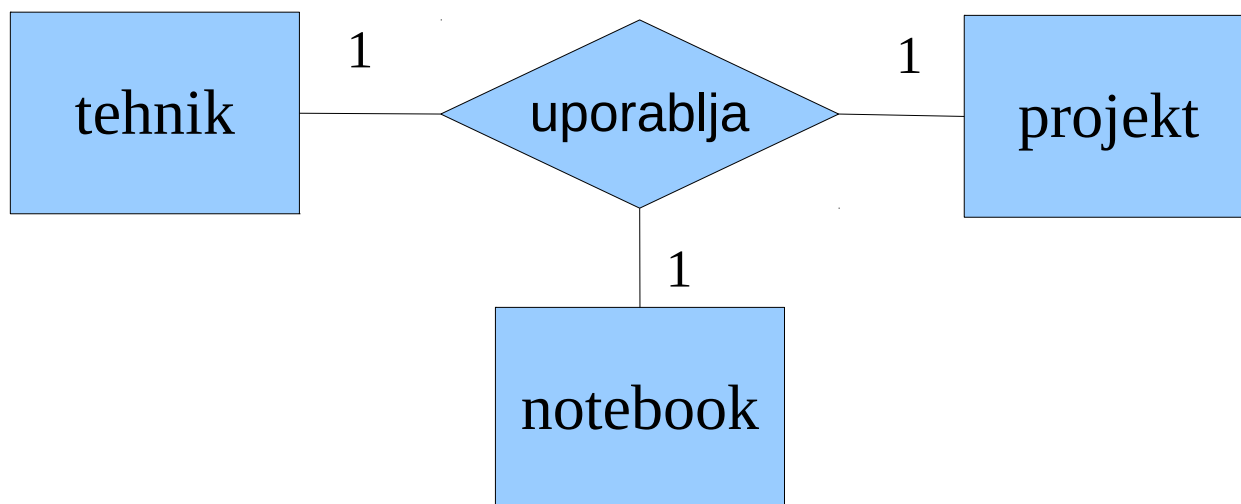
# Omejitve pri ER načrtovanju

- omejitev pokrivanja – omeji razmerje entitete do določenih pod-razredov
- entitete v danem trenutku:
  - obvezno / opsijsko
  - določi spodnjo mejo povezanosti instanc entitet
  - sodelovanje v razmerju s števnostjo 1 ali 0
- omejitev unikatnosti – ena-ena funkcijska odvisnost med atributi ključev v razmerju: binarnem, ternarnem in višjih.

# Primer: ena-ena-ena

- Tehnik uporablja natančno en notebook za en projekt.
- Vsak notebook pripada enemu tehniku za en projekt.
- Tehnik lahko dela na večih projektih in uporablja različne notebooke za različne projekte.

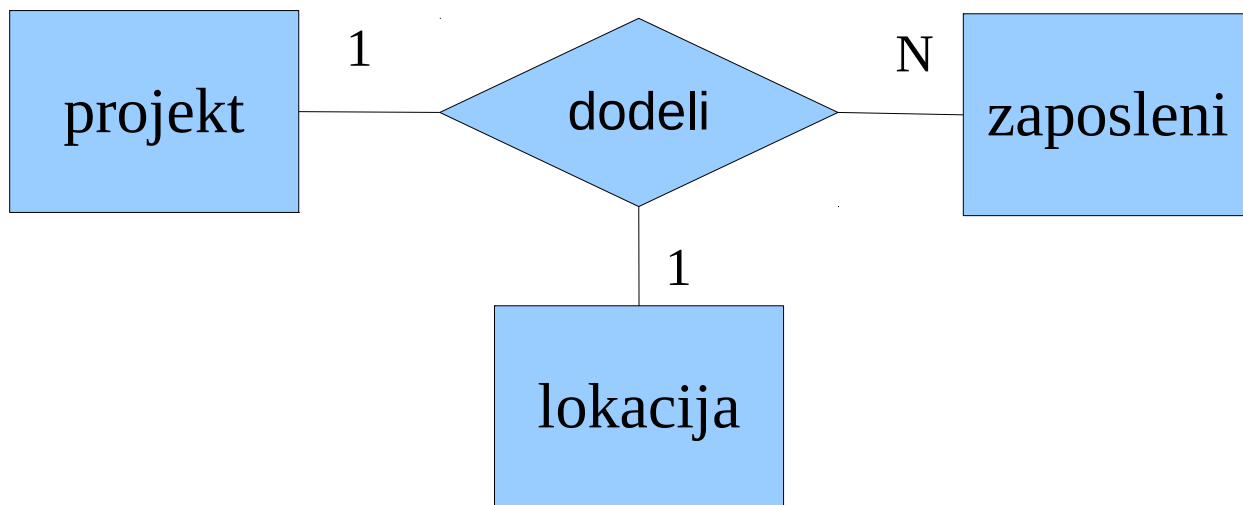
zaposleni-id, projekt-ime --> notebook-id  
zaposleni-id, notebook-id --> projekt-ime  
projekt-ime, notebook-id --> zaposleni-id



# Primer: ena-ena-več

- Vsak zaposleni, ki je dodeljen na projekt dela na eni sami lokaciji.
- Zaposleni so lahko na različnih lokacijah za različne projekte.
- Na dani lokaciji dela en zaposleni na enem projektu.
- Na dani lokaciji je lahko več zaposlenih na danem projektu.

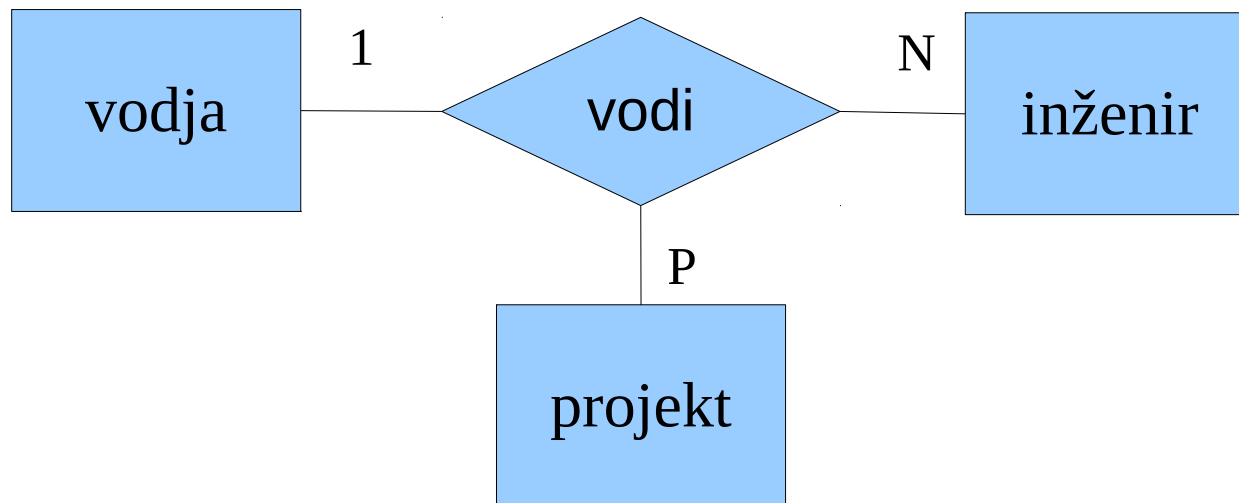
zaposleni-id, projekt-ime --> lokacija-id  
zaposleni-id, lokacija-id --> projekt-ime



# Primer: ena-več-več

- Vsak inženir, ki dela na določenem projektu ima natančno enega vodjo.
- Vsak vodja projekta lahko vodi več inženirjev.
- Vsak vodja inženirja je lahko vodja istega inženirja na večih projektih.

projekt-ime, zaposleni-id --> vodja-id



# Primer: N-N-N

- Zaposleni uporabljajo več znanj na kateremkoli od večih projektov.
- Vsak projekt ima več zaposlenih z raznovrstnimi znanji.

