

NAČRTOVANJE PODATKOVNIH BAZ

Iztok Savnik

Konceptualno načrtovanje podatkov

1. Klasificiraj entitete in attribute

* entitete vsebujejo opisne podatke

* večvrednostni atributi se klasificirajo kot entitete

* atributi so pripeti na entitete, ki jih najbolj direktno opisujejo

2. Identificiraj generalizacijske hierarhije

3. Definiraj razmerja – binarna, unarna in ternarna

Klasificiraj entitete in attribute

- Enostavno je definirati entiteto, atribut in razmerje...
- Ni enostavno razlikovati vloge načrtovalskih gradnikov pri modeliranju podatkovne baze.
- Kakšen podatkovni element? Entiteto, atribut ali razmerje?
- Na primer: projekti so locirani po mestih. Naj bo mesto entiteta ali atribut?
- Na primer: za vsakega zaposlenega se hrani CV. Naj bo CV entiteta ali razmerje?

Navodila za klasifikacijo entitet in atributov

- Entitete naj vsebujejo opisne podatke.
- Večvrednostni atributi naj bodo klasificirani kot entitete.
- Atributi naj bodo pripeti na entitete, ki jih najbolj direktno opisujejo.

Vsebina entitete

- Če imamo opisne podatke o podatkovnem elementu potem naj bo podatkovni element klasificiran kot entiteta.
- Primer „mesto“: običajno ima opisne podatke npr. populacija, država, itd. Torej, je entiteta.
- Če potrebujemo samo ime mesta potem je atribut.
- Primer „država“: podobno kot mesto.
- Tipični primeri entitet: zaposleni, projekt, oddelek, firma, itd.

Večvrednostni atributi

- Klasificiraj večvrednostne attribute kot entiteto.
- To velja tudi v primeru, ko vrednosti atributa nimajo identifikatorja. V tem primeru se id prenese.
- Primer:
 - Velika firma ima veliko oddelkov, ki bi jih lahko definirali kot večvrednostni atribut.
 - Oddelek raje klasificiramo kot entiteto z npr. naslovom oddelka kot identifikator.
- Načrt se velikokrat poenostavi.

Atributi

- Poveži attribute z entitetami, ki jih najbolj direktno opisujejo.
- Postopek je iterativen.
- Izberi entiteto in določi attribute. Izberi novo entiteto in izberi attribute...
- V primeru, da se krši kakšno od pravil predstavljenih prej spremeni entiteto v atribut (ali atribut v entiteto).

Identifikacija generalizacijske hierarhije

- Če obstaja generalizacijska hierarhija:
 - Postavi identifikator in druge generične lastnosti v najbolj splošno entiteto.
 - Definiraj specifične lastnosti v podrejenih entitetah.
- Primer:
 - Zaposleni z identifikatorjem id-zap in splošnimi opisnimi atributi ime-zap, naslov in datum-rojstva.
 - Vodja, ki ima poleg naštetega še naziv-dela.
 - Inženir, ki ima zraven splošnih lastnosti še stopnja-izobrazbe in naziv-dela.
 - Tehnik, ki ima posebno lastnost specializacija.
 - Tajnica, ki ima posebno lastnost najboljša-zmožnost.

Identifikacija razmerij

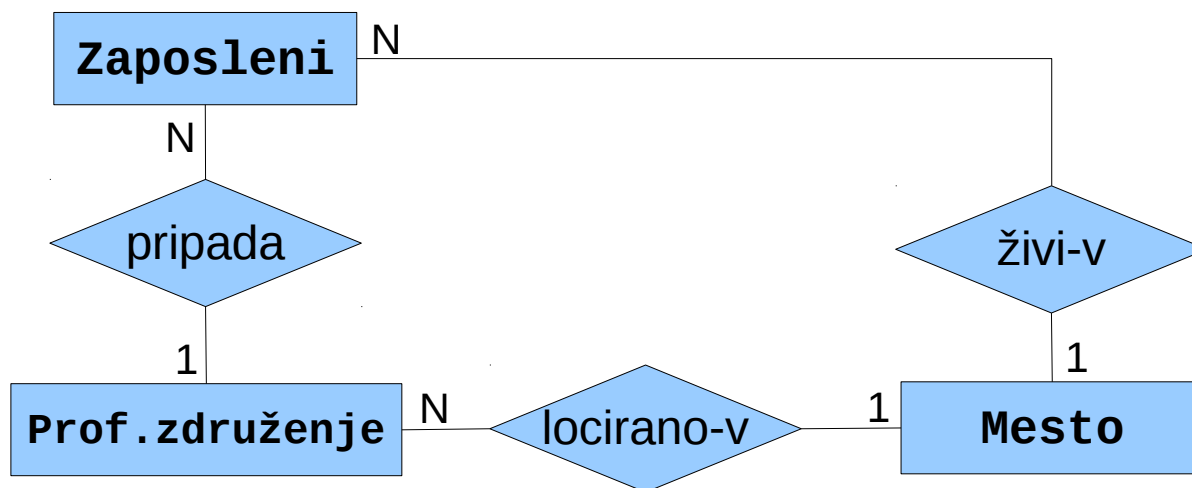
- Rekurzivna razmerja
- Binarna razmerja
- Ternarna razmerja

Redundantna razmerja

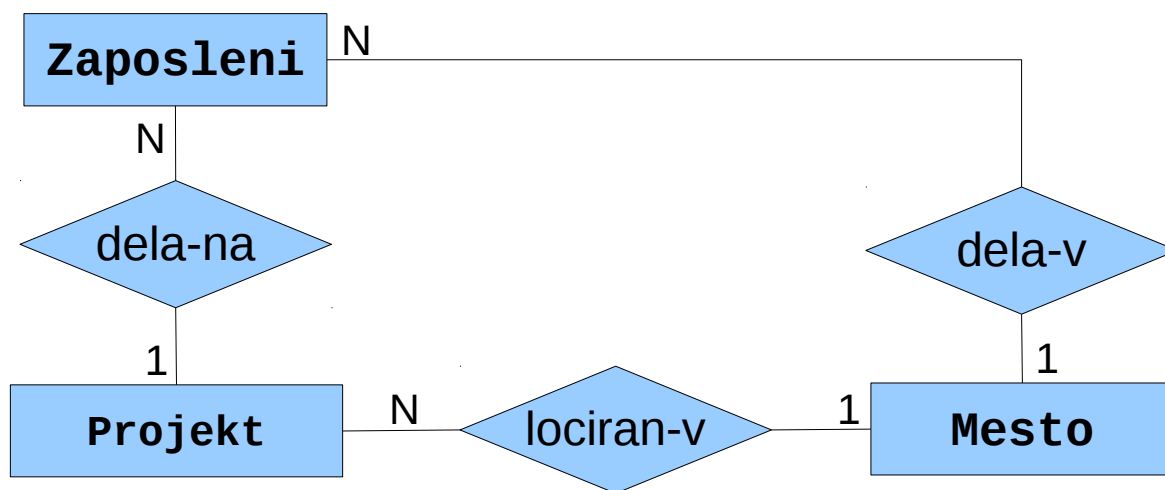
- Razmerja, ki predstavljajo isti koncept so redundantna.
- Redundantna razmerja zelo verjetno predstavljajo redundanco pri prevodu konceptualnih shem v relacijske sheme.
- Dva ali več razmerij je lahko definiranih med dvema entitetama, če imajo različen pomen.

Binarna razmerja

Neredundantno razmerje

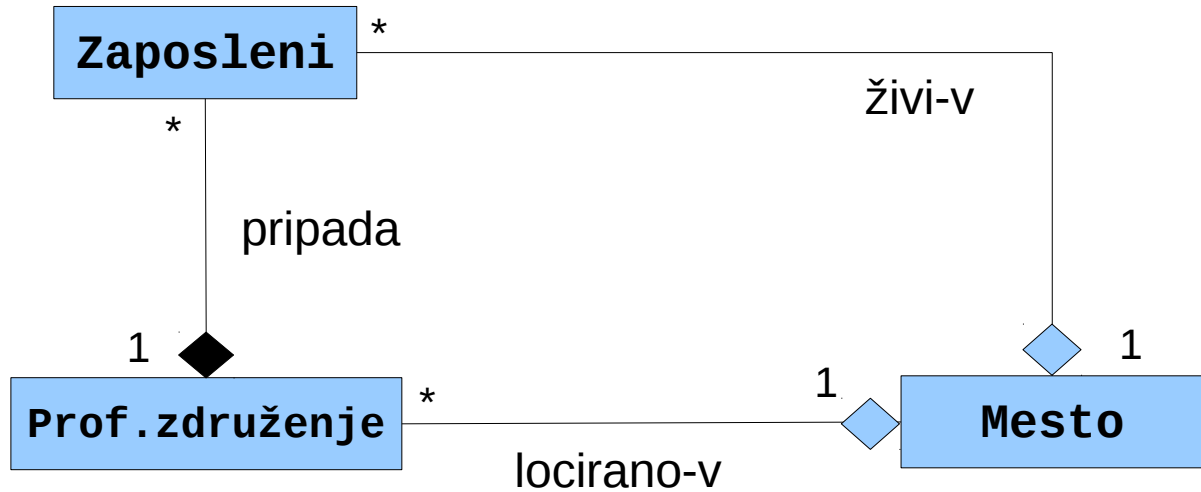


Redundantno razmerje

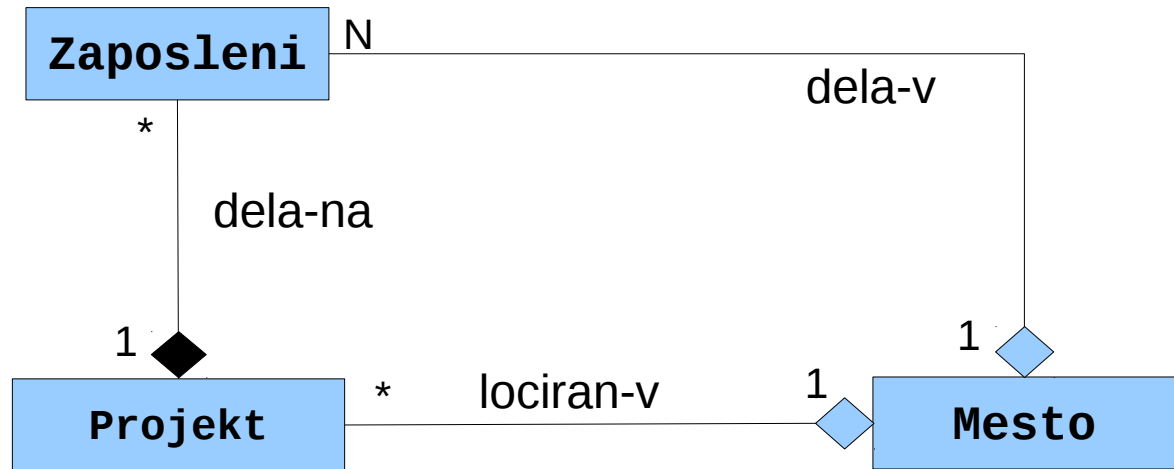


Binarna razmerja

Neredundantno razmerje



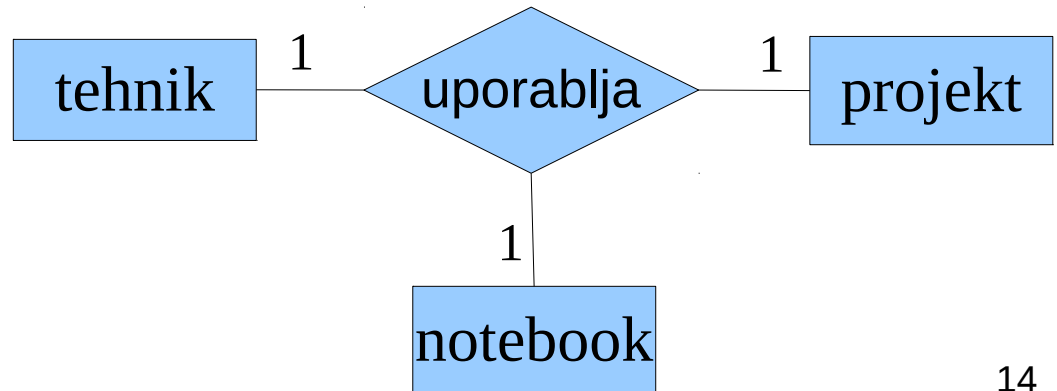
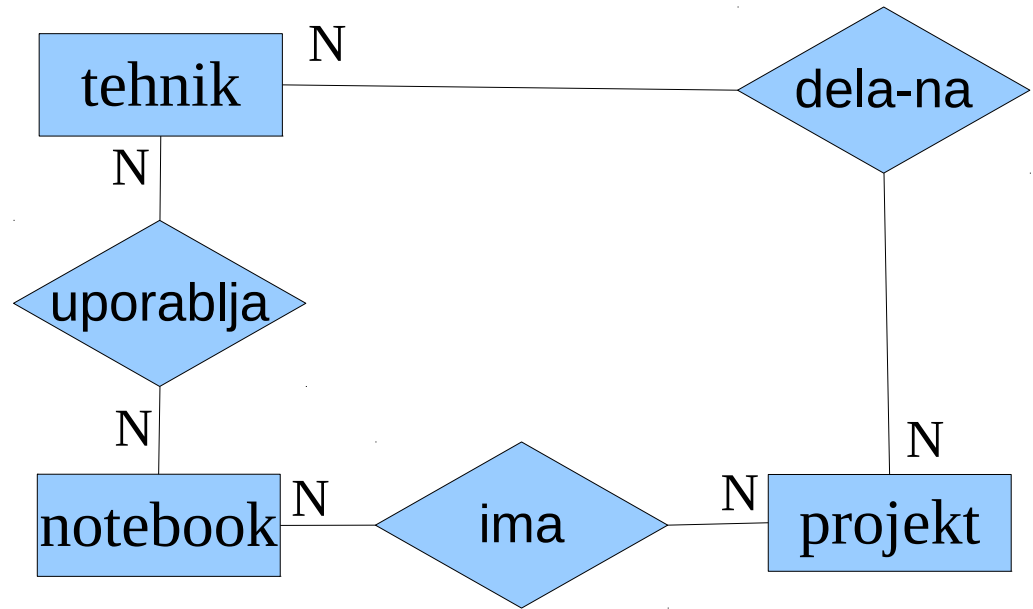
Redundantno razmerje



Ternarna razmerja

- Definiraj ternarno razmerje zelo previdno.
- Ternarno razmerje definiramo, ko koncepta ne moremo predstaviti z več binarnih razmerij.
- Primer:
 - Poglejmo si razmerje med entitetami Tehnik, Projekt in Notebook.
- Če tehnik lahko dela na večih projektih in uporablja iste notebooke na projektih potem lahko definiramo tri N-N binarna razmerja.
- Če pa tehnik uporablja en notebook na enem projektu in je ta notebook od enega tehnika potem je potrebno definirati ternarno razmerje 1:1:1.

Ternarno razmerje



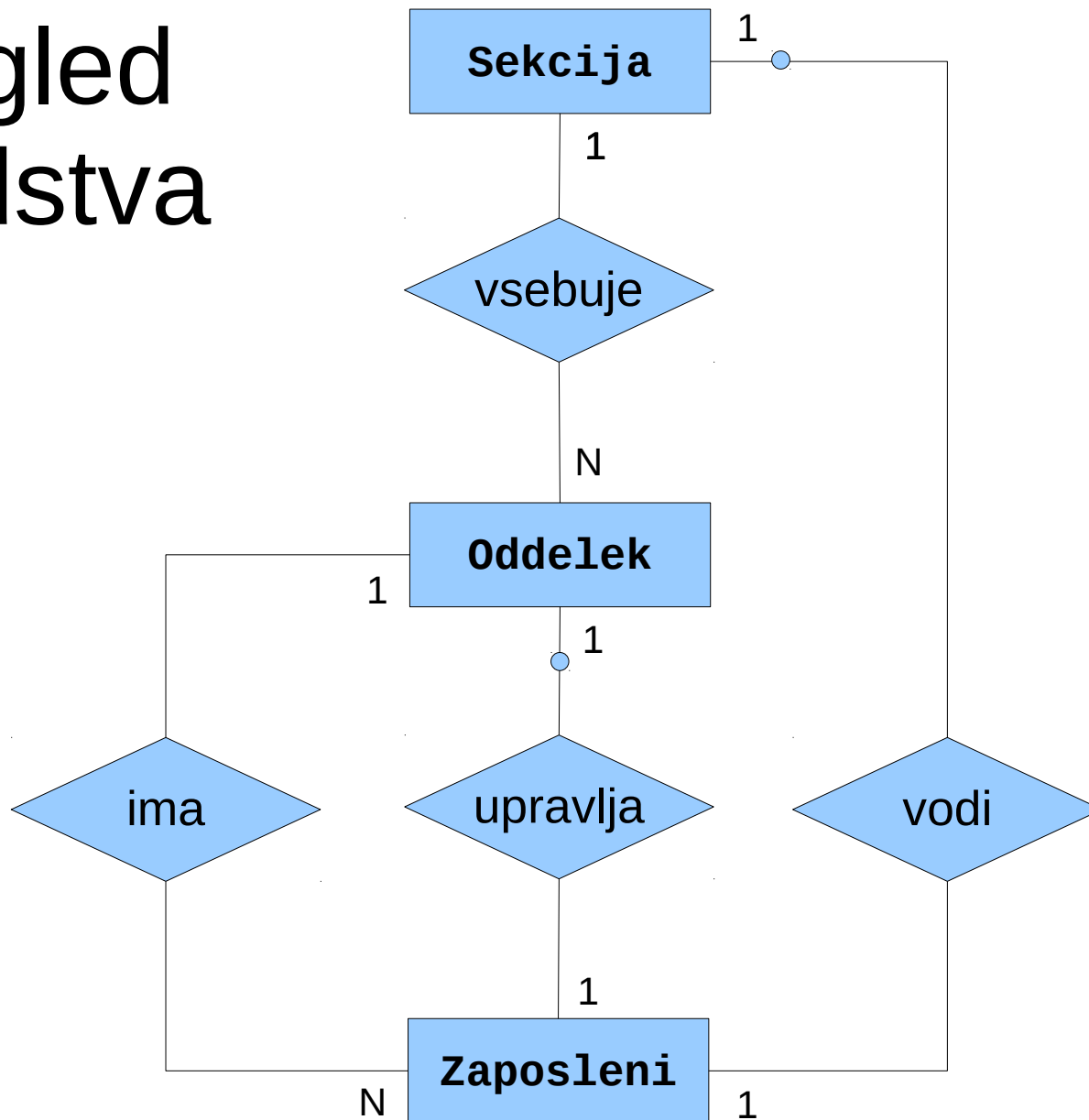
Ternarna razmerja

- Če je možno torej uporabimo binarna razmerja in v primeru, če to ni mogoče potem poskusi uporabiti ternarno razmerje.
- Kot smo že pokazali lahko isto omejitev kot z uporabo ternarnega razmerja predstavimo tudi s funkcijskimi odvisnostmi.

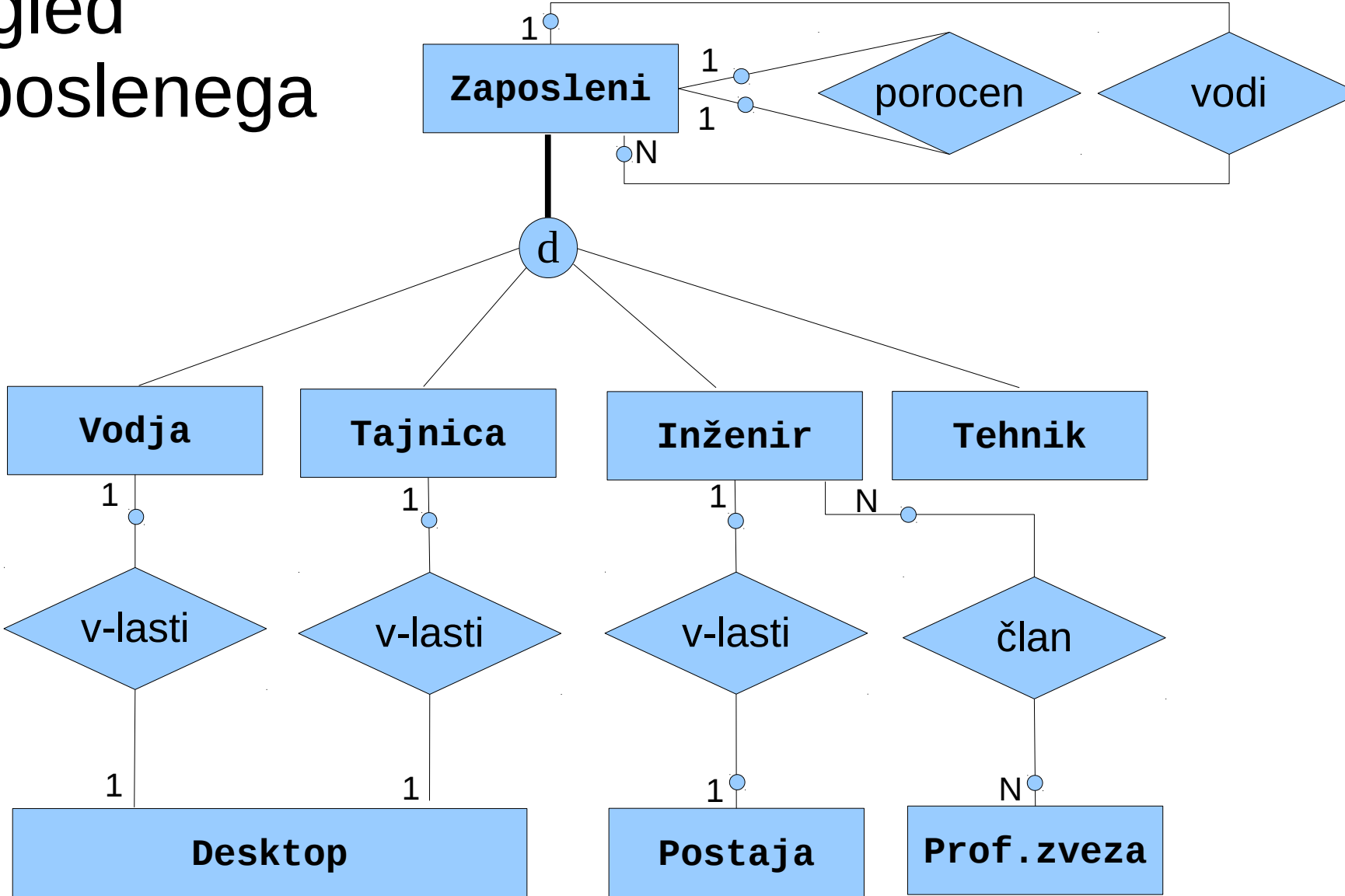
Primer: PB zaposlenih in projektov

- Velika inženirska firma
- Sledimo osebjem, zmoglostim, projektom, prireditvi osebja projektom, uporabljen material, prireditvev osebja oddelkom in sekcijam.

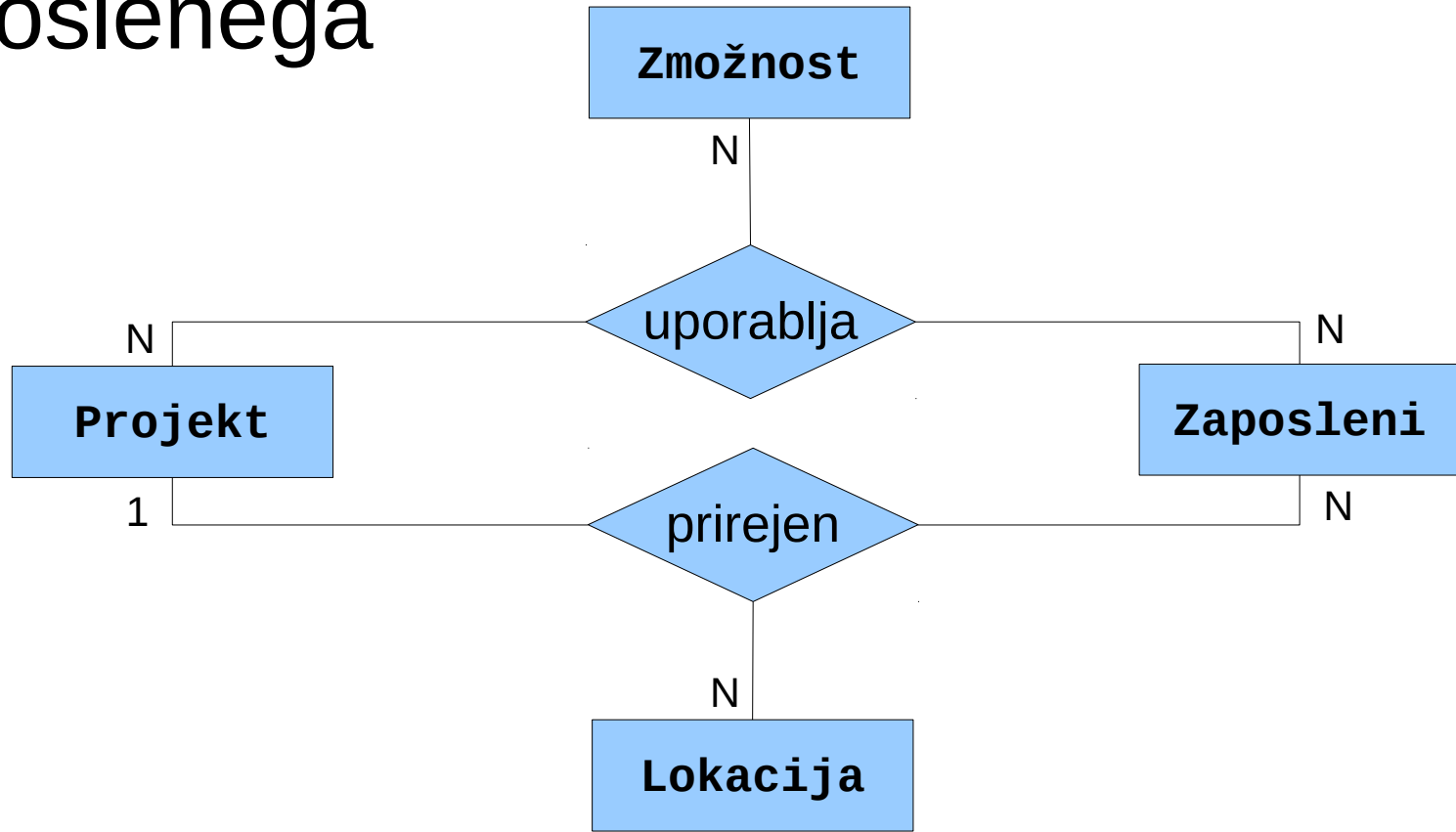
Pogled vodstva



Pogled zaposlenega



Pogled uporabe zaposlenega



Integracija shem

- kreiranje neredundantne (unificirane) globalne konceptualne sheme
 - (1) kompletnost – vse komponente se morajo nahajati v globalni shemi
 - (2) minimalnost – odstrani vse redundantne koncepte v globalni shemi
 - (3) razumljivost – ima globalna shema smisla?

koraki integracije shem

- pre-integracijska analiza
- primerjanje shem
- prilagajanje shem
- združevanje in restrukturiranje

pre-integracijska analiza

- izbira integracijske strategije
- binarni pristop
 - enostaven; majhno število operacij
- n-terni pristop
 - med 2 in številom shem
 - eno veliko združevanje
 - z n narašča kompleksnost združevanja

primerjanje shem

- poišči zveze (enakost) med entitetami
- identificiraj možne konflikte
- določi lastnosti povezav med shemami

odkrij možne konflikte

- konflikti imen
 - homonim – isto ime za različne koncepte
 - sinonim – različna imena za iste koncepte
- strukturni konflikti
 - konflikti tipov
 - različni gradniki za iste koncepte
 - (npr. “naročilo” kot entiteta, atribut, razmerje)

odkrij možne konflikte (2)

- konflikt odvisnosti
 - števnost (istih/podobnih) razmerij je različna po različnih shemah (npr. job-title vs. job-title-history)
- konflikt ključev
 - isti koncepti, ki imajo drugačne ključe (npr. ID-no vs. SSN)
- konflikt obnašanja
 - različne integritetne omejitve (npr. null pravila za opsijsko/obvezno vlogo: insert/delete rules)

določi lastnosti med shemami

- možna nova razmerja za povezovanje shem
- abstrakcije na obstoječih entitetah ali kreacija novih super-razredov (super-tipi)

prilagajanje shem

- razreši konflikte (velikokrat je potrebna interakcija z uporabnikom)
- prilagodi in uredi sheme tako, da bodo kompatibilne za integracijo
- transformiraj shemo z uporabo
 - preimenovanje (homonimi, sinonimi, konflikt ključa)
 - transformacija tipov (konflikti tipa ali odvisnosti)
 - spreminjanje omejitev (izjav) (konflikt obnašanja)

prilagajanje shem (2)

- primeri prilagoditev
 - entitete ali ključne je potrebno preimenovati
 - dan koncept modeliramo na en sam način (entiteta, razmerje ali atribut)
 - razmerja, ki imajo enako stopnjo, vloge in števnost je enostavno združevati
 - težje je združevati razmerja z omejitvami, ki se ne skladajo; včasih ni možno
 - relacije so včasih nekonsistentne; npr. generalizacija in ekskluziven OR
 - omejitve (izjave) se morajo prilagoditi tako, da so konsistentne

združevanje in restrukturiranje

združevanje in restrukturiranje je vodeno z naslednjimi cilji:

kompletnost

- vsi koncepti komponent se nahajajo v globalni shemi in so semantično kompletni

minimalnost

- minimalnost zahteva od načrtovalca, da odstrani vse redundantne koncepte v globalni shemi

razumljivost

- zahteva, da je globalna shema razumljiva uporabniku

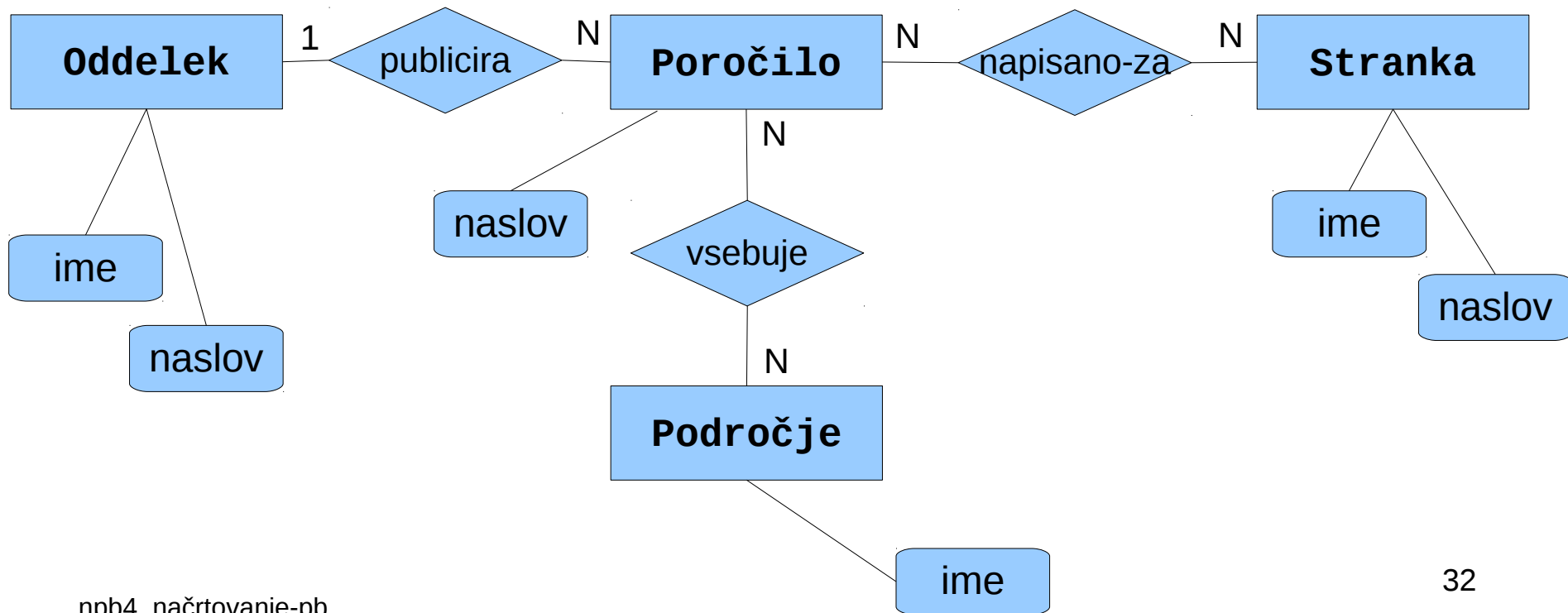
združevanje in restrukturiranje (2)

- komponentne sheme se združijo tako, da najprej prekrijemo enake koncepte s predstavnikom
- sledi re-strukturiranje sheme za boljšo razumljivost
- primer:
 - supertip/podtip relacija je rezultat združevanja
 - lastnosti podtipa lahko odstranimo zaradi dedovanja

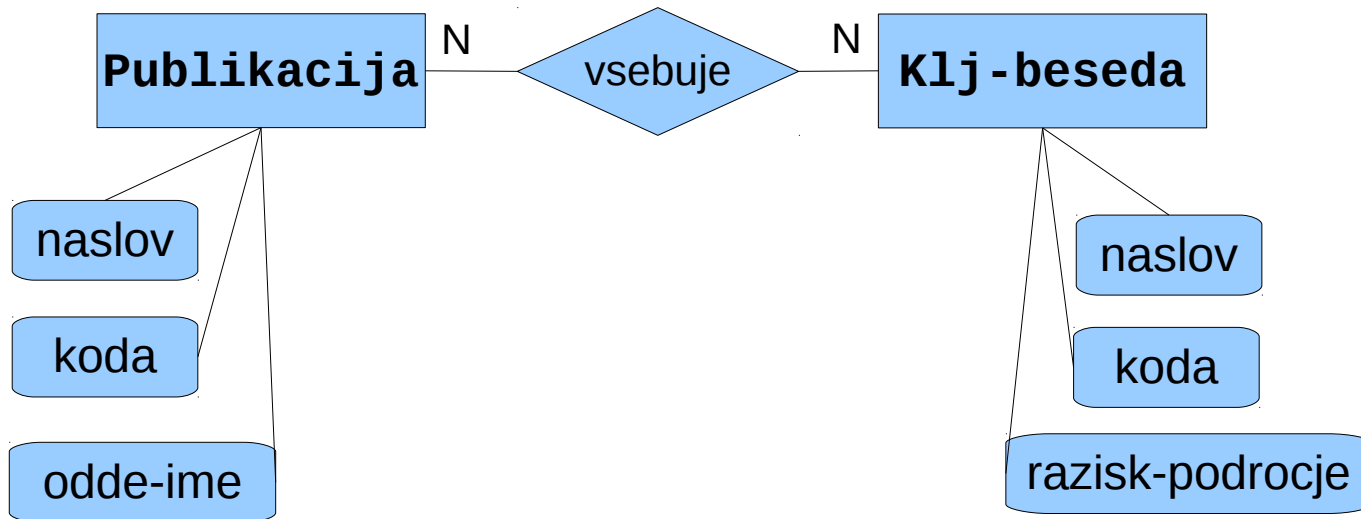
primer združevanja shem

- oglejmo si različne poglede na prekrivajoče podatke, ki temeljijo na različnih intervjujih
- prvi pogled je fokusiran na poročila
- drugi pogled je fokusiran na publikacije
- poiskati združitev shem, ki zagotavlja kompletnost, minimalnost in razumljivost

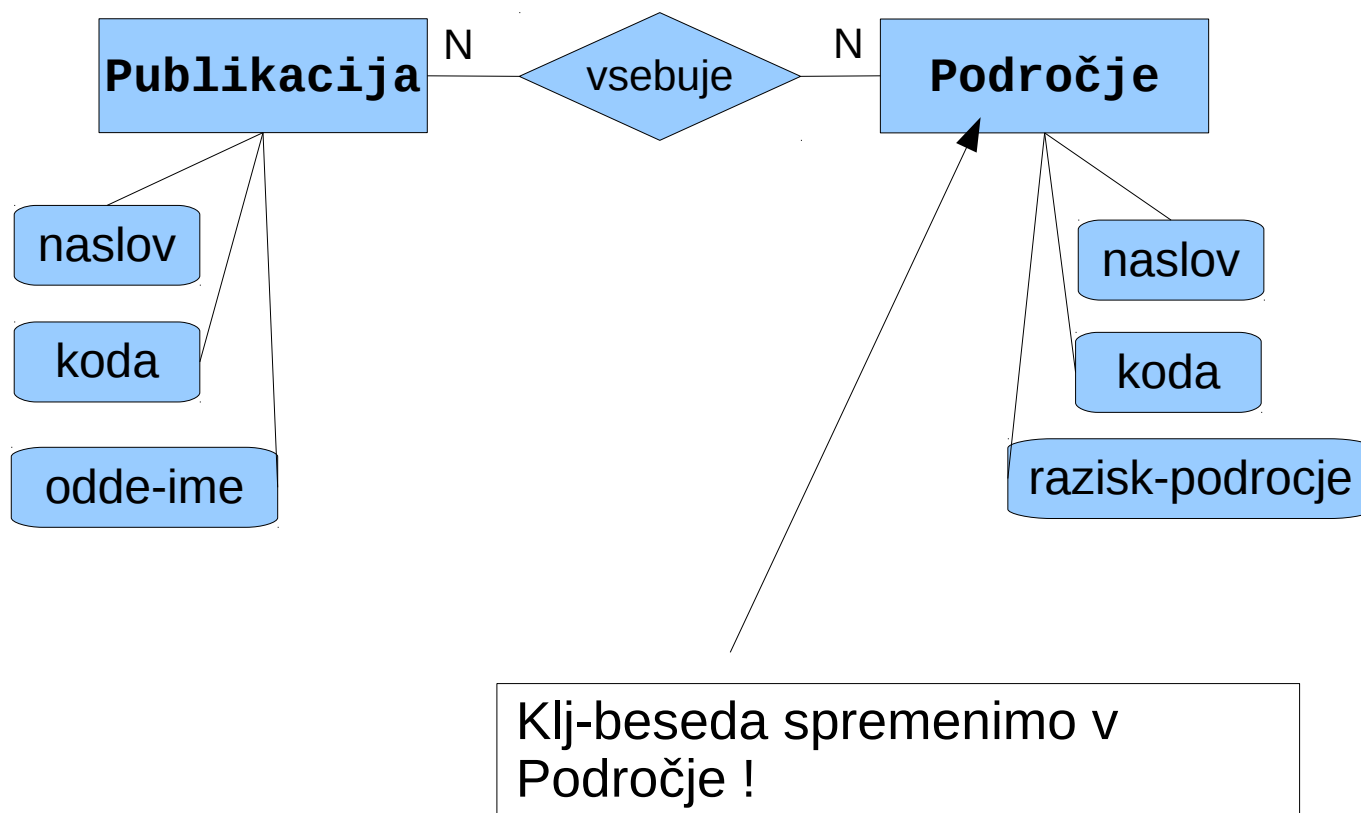
Originalna shema 1 - fokus na poročila



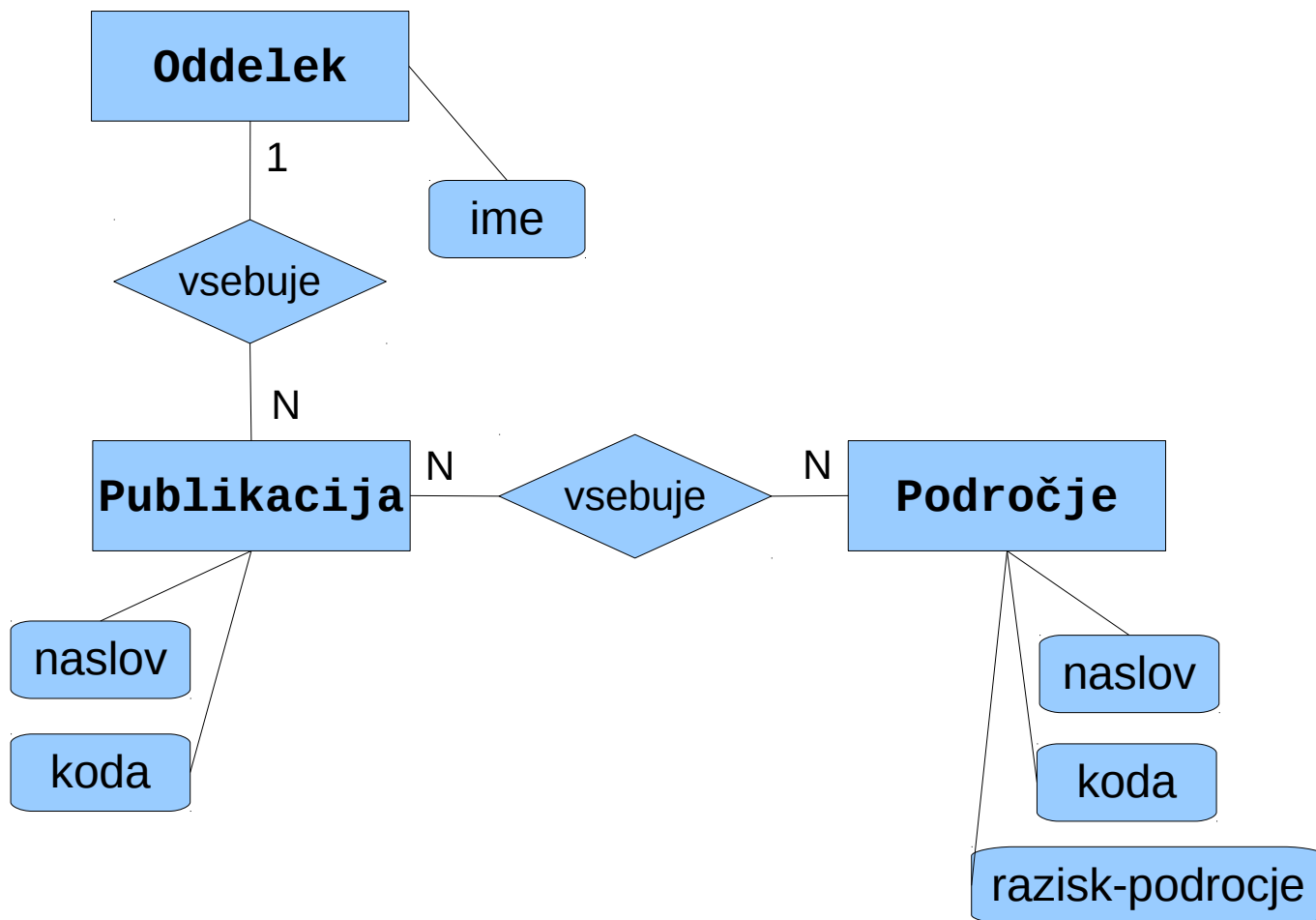
Originalna shema 2 - fokus na publikacije



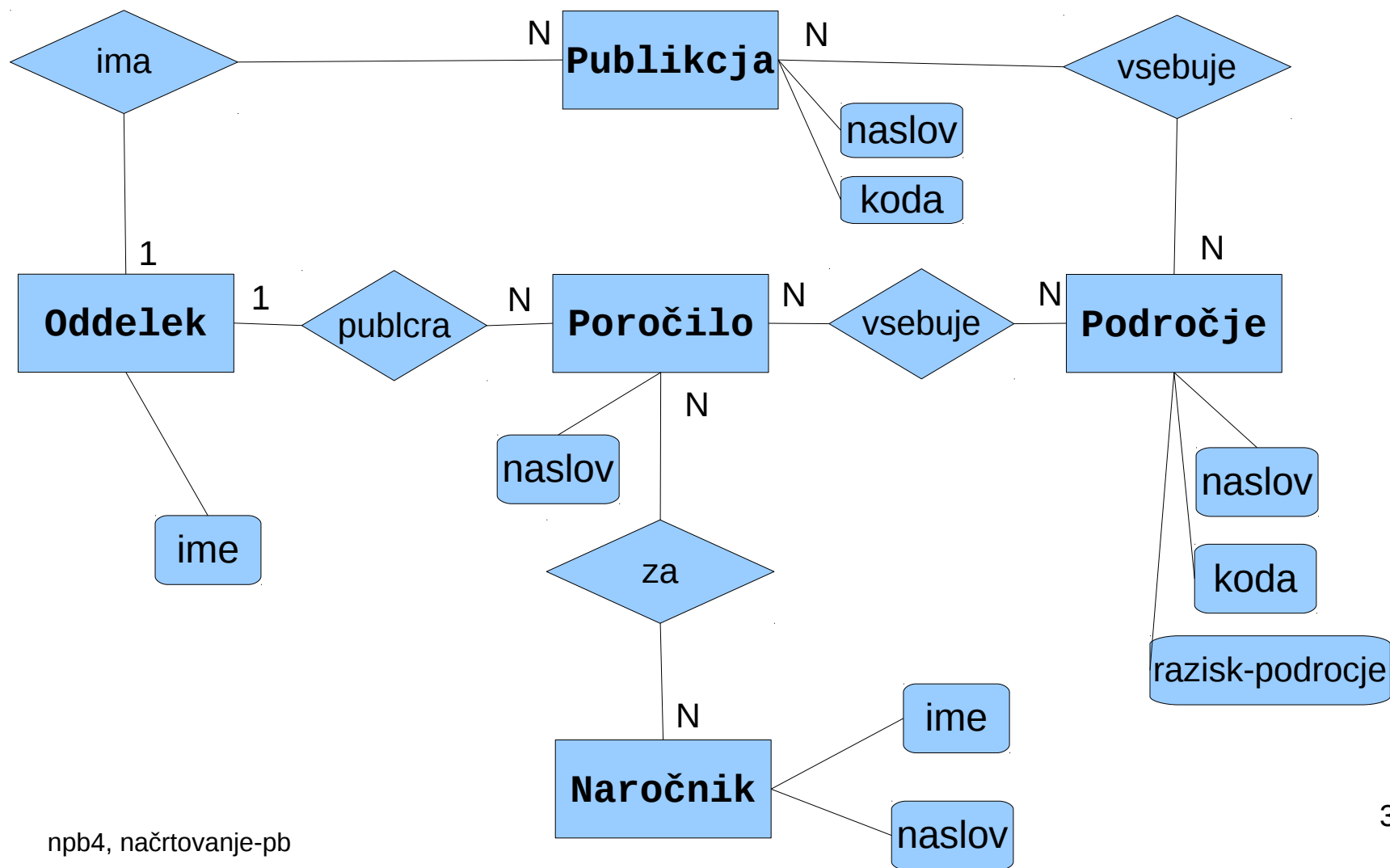
Shema 2.1 - fokus na publikacije



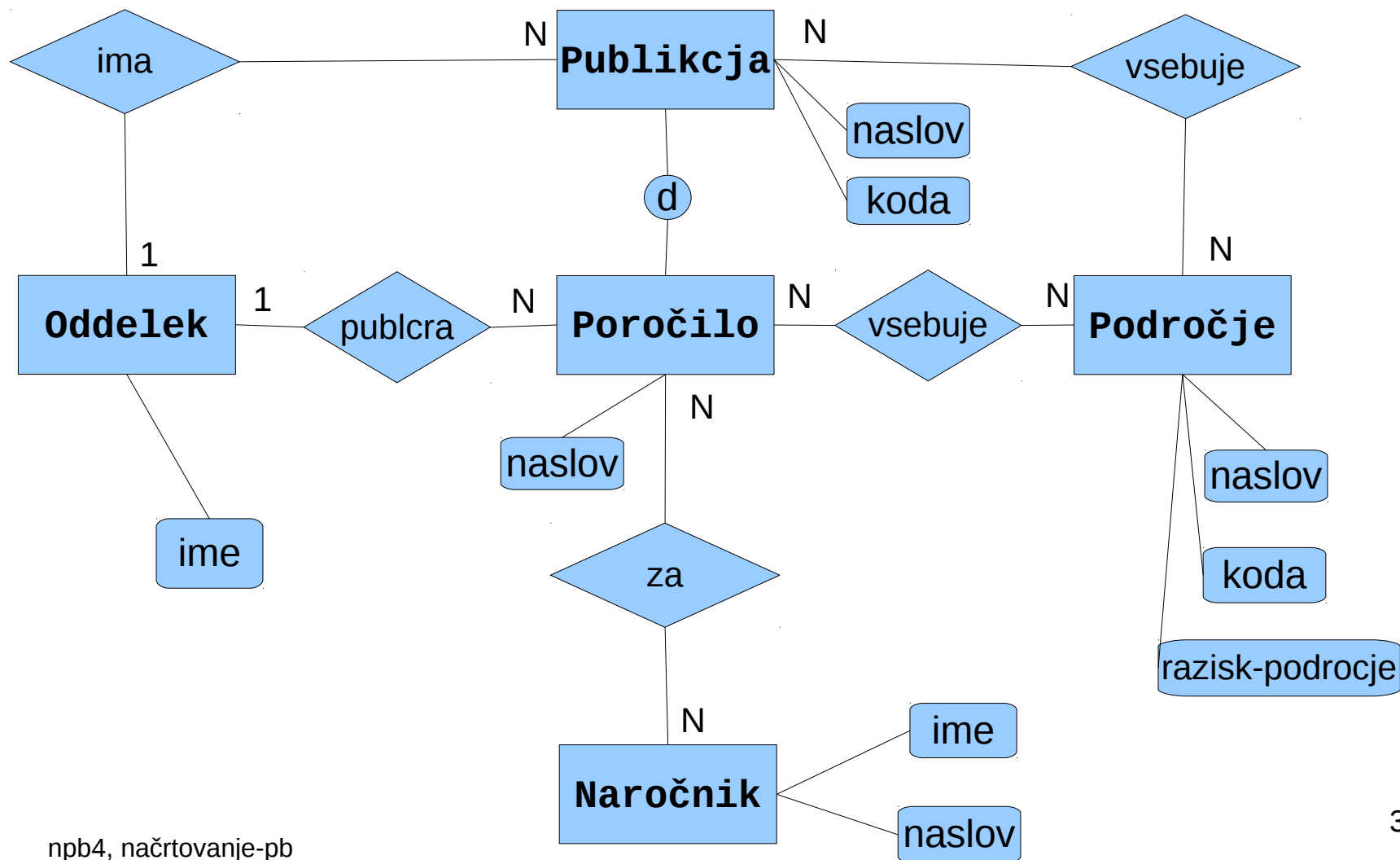
Shema 2.2 - fokus na publikacije



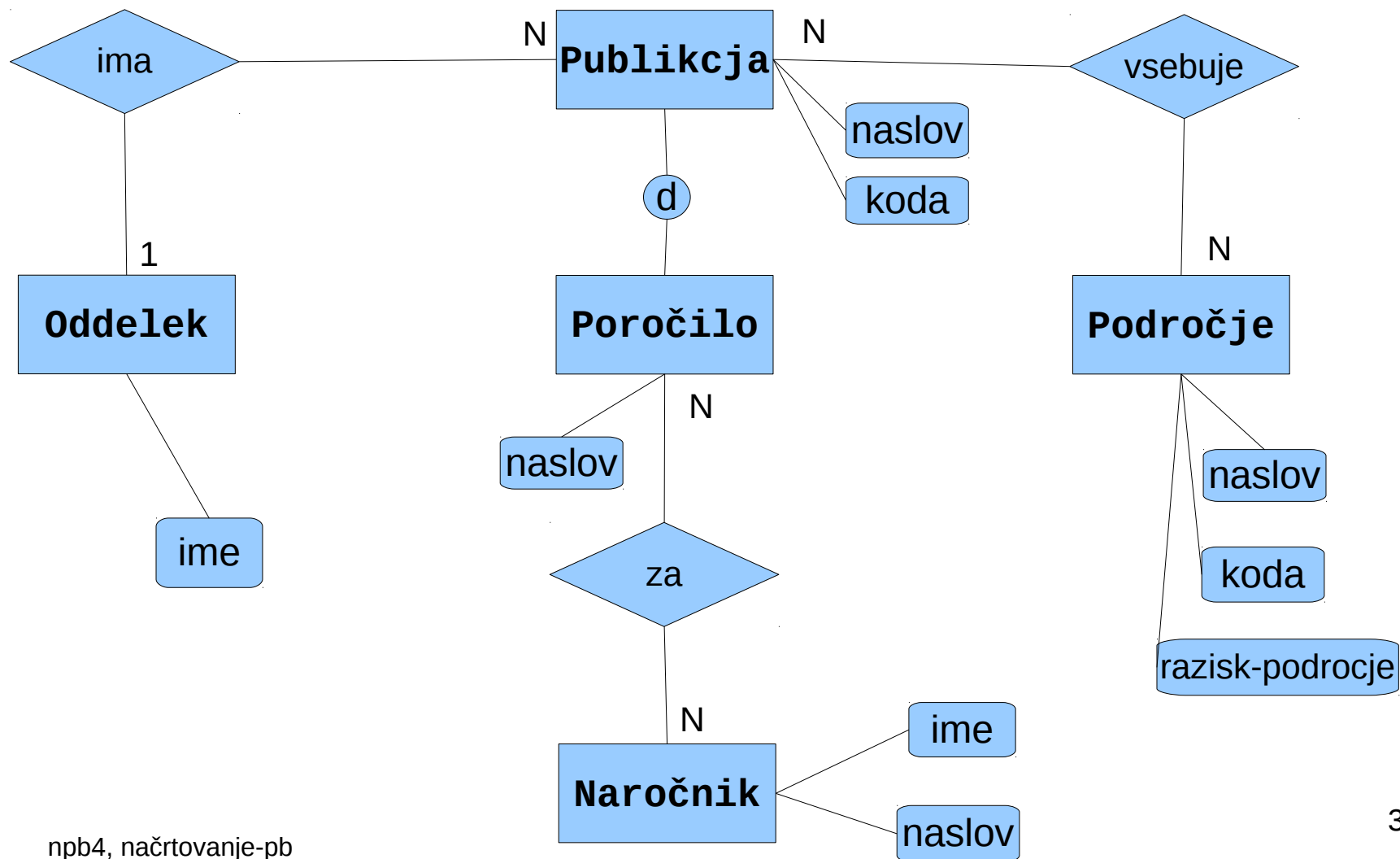
Shema 3 – združitev 1 in 2.2



Shema 3.1 – nova generalizacija



Shema 3.2 – redundantna razmerja



grupiranje entitet v ER

- abstrakcije ER shem
- celotna shema aplikacije pride na eno samo stran
- posledice za končnega uporabnika
 - razumljivost se poveča
 - formalna dokumentacija konceptualnega modela

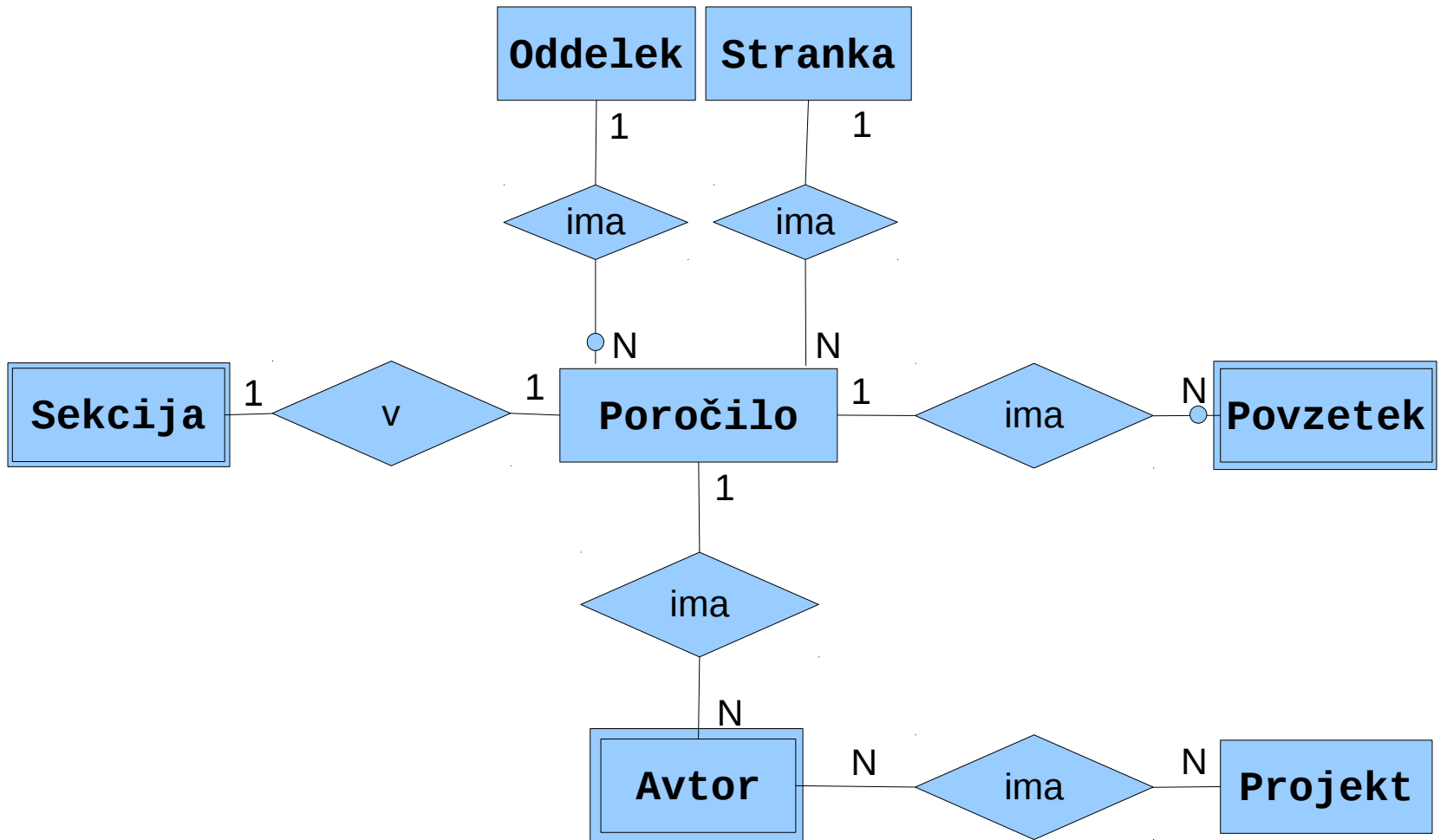
motivacija

- konceptualni modeli so težko razumljivi za velike in kompleksne podatkovne baze (>100,>1000,...)
- grupiranje entitet je metoda s katero organiziramo kompleksno konceptualno shemo v nivoje abstrakcije
- orodje za abstrakcijo kompleksnih konceptualnih shem
 - potencialne aplikacije
 - komunikacija s končnim uporabnikom
 - komunikacija z aplikacijsko ekipo
 - dokumentacija sheme podatkovne baze (v skladu s podatkovnim slovarjem)

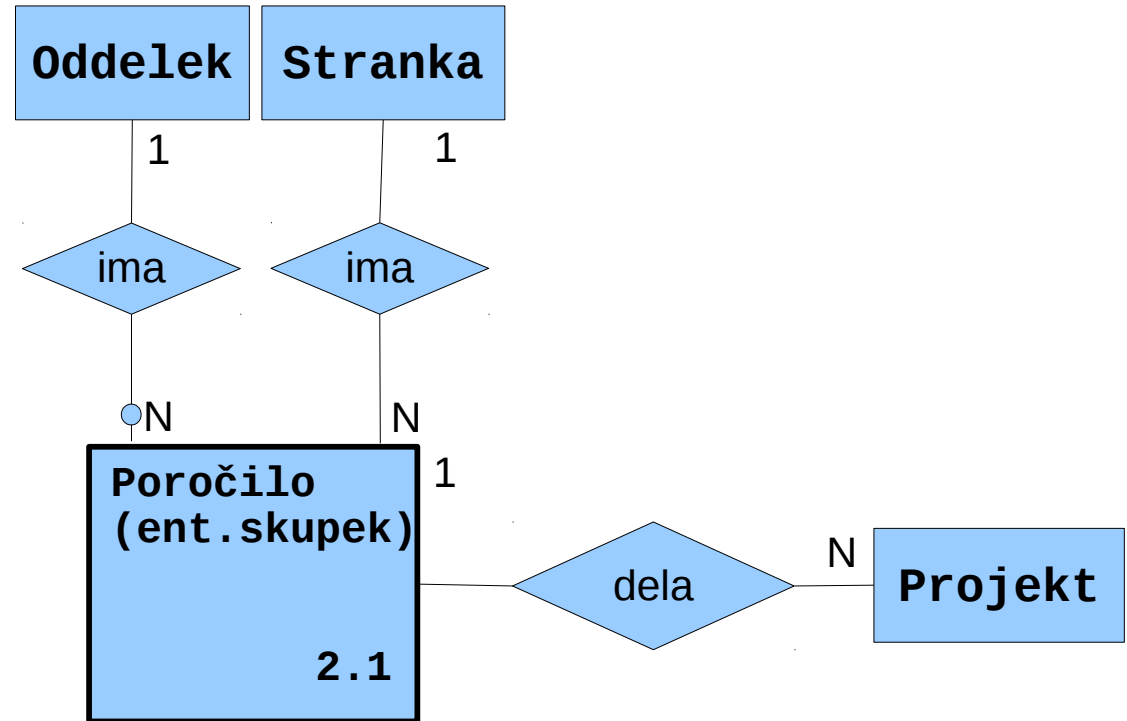
koncepti grupiranja

- kombiniranje entitet in razmerij za kreacijo bolj abstraktnega koncepta
 - rezultat = skupek entitet
- operacija grupiranja tvori bolj abstraktno entiteto iz skupka entitet (skupkov) in razmerij
- najvišji skupek predstavlja kompletno podatkovno bazo – korenski skupek entitet

koncepti grupiranja (2)



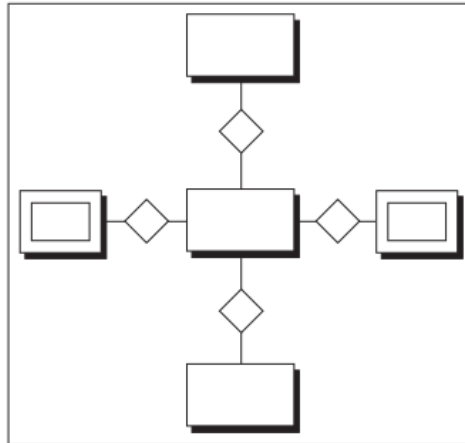
koncepti grupiranja (3)



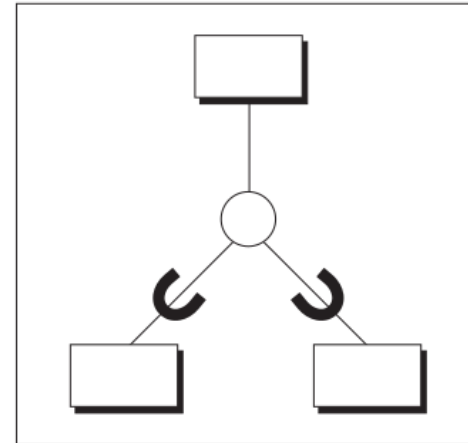
operacije grupiranja

- osnovne vrste grupiranja
 - dominanca
 - klasifikacija
 - omejitev
 - relacija

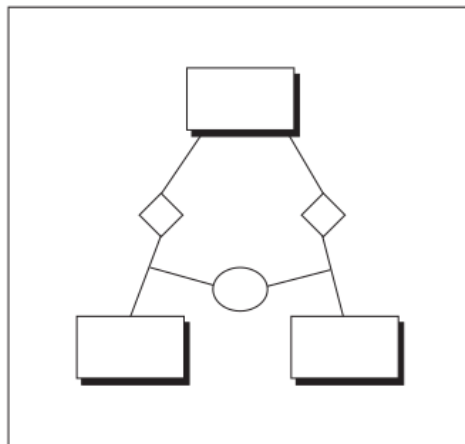
operacije grupiranja (2)



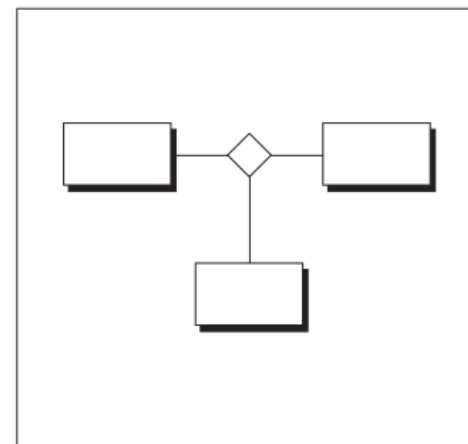
(a) Dominance grouping



(b) Abstraction grouping



(c) Constraint grouping



(d) Relationship grouping

metodologija grupiranja

- kako kreirati korenski skupek iz elementarnih entitet?
- predpostavljamo, da je bila narejena top-down analiza
- višje-nivojske strukture uporabljene za vodenje grupiranja na nižjih nivojih
- funkcijske enote so pogosto definirane v skladu z organizacijsko strukturo organizacij, poslovnimi procesi ali primarno aplikacijsko področje
- funkcijska področja se lahko prekrivajo

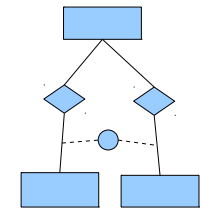
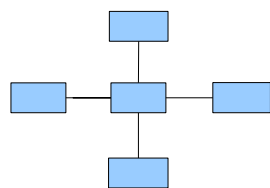
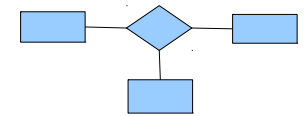
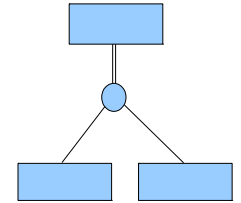
metodologija grupiranja (2)

za dan ER diagram podatkovne baze.....

Korak 1. definiraj točke grupiranja preko funkcijskih področij.

Korak 2. kreiraj skupke entitet

- uporabi operacije grupiranja
- entitete lahko pripadajo različnim skupkom
- definirati je potrebno prioritete
 - grupiraj entitete znotraj funkcijskih področij
 - entiteta v skupnku ne sme biti na drugih fun. področjih
 - v primeru večih skupkov znotraj fun. področja
 - razreši konflikte s kombiniranjem višjih funkcionalnih grupiranj



metodologija grupiranja (2)

Korak 3. kreiraj višjenivojske skupke entitet rekurzivno do korena

- razreši konflikte na isti način kot opisano
- skupki in entitete, ki ostanejo na eni strani so koren

Korak 4. preveri diagrame skupkov

- preveri konsistenco vmesnikov
- končni uporabnik mora potrditi vsak nivo.

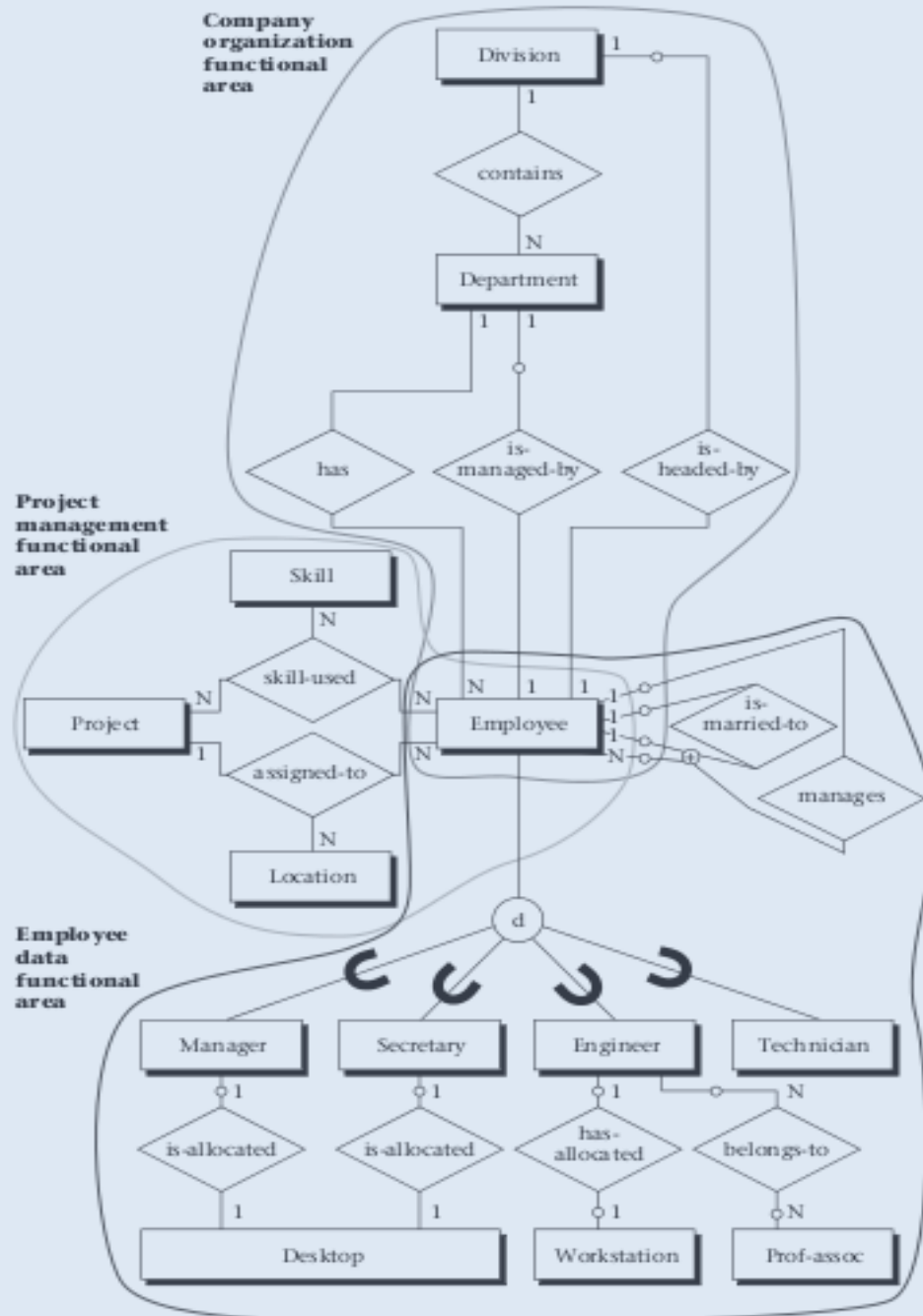


Figure 4.10 ER diagram: clustering technique

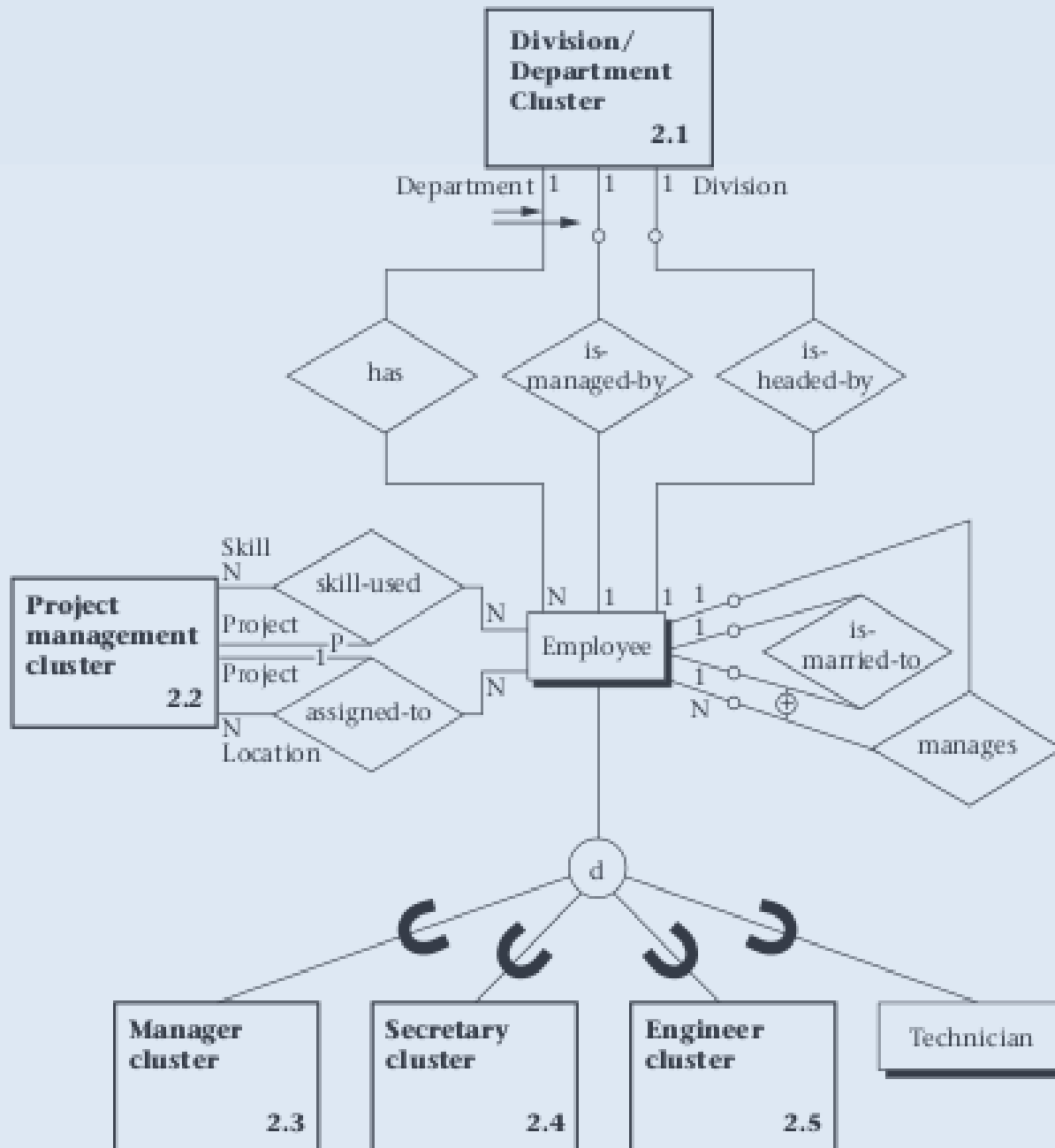


Figure 4.11 Clustering results