Baze podataka

Katedra za računarstvo
Elektronski fakultet u Nišu

Modeli podataka i projektovanje baza podataka

Prof.dr Leonid Stoimenov

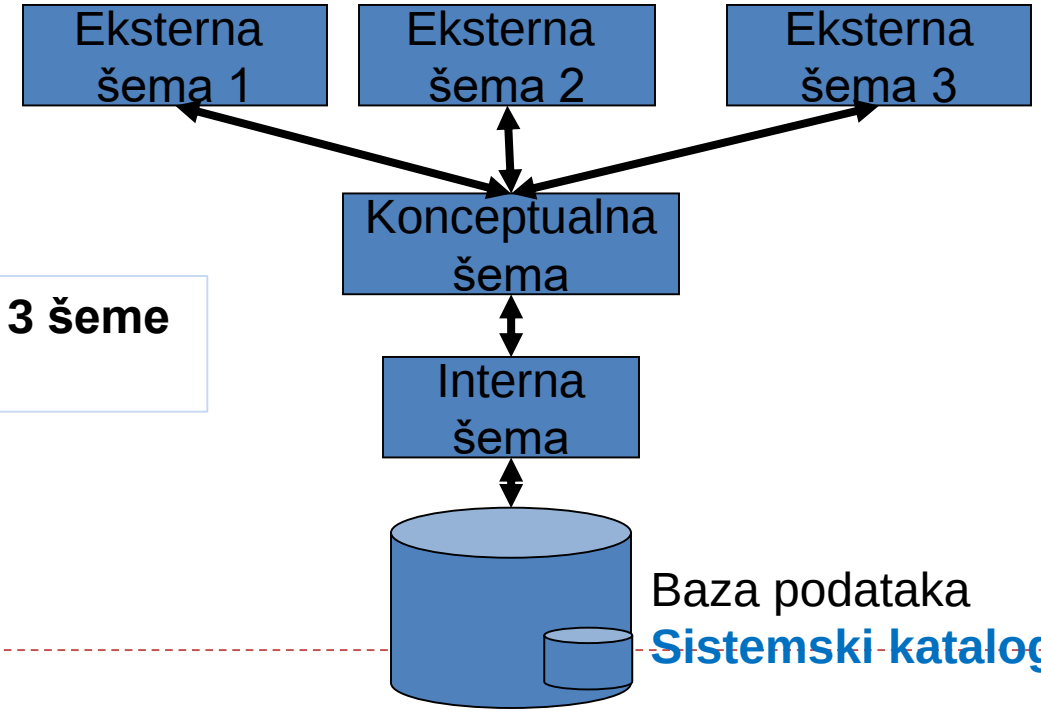
Pregled predavanja

3 termina za predavanja

- Arhitektura 3-šeme DBMSa
- Modeli podataka
- Projektovanje baza podataka
- Konceptualno modeliranje

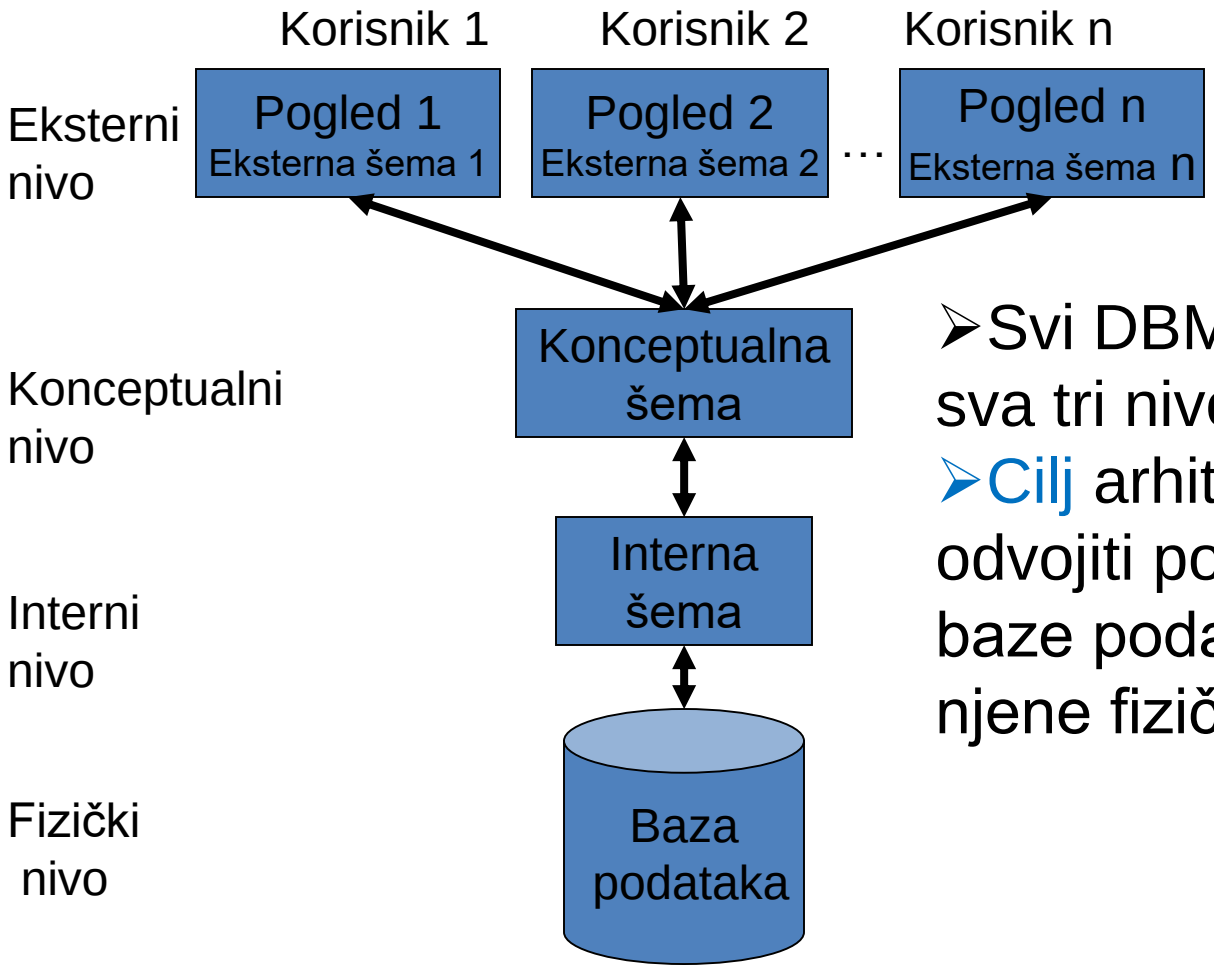
Nivoi apstrakcije u DBMS-u

- U DBMS-u podaci su opisani u **tri nivoa apstrakcije**:
  - Eksterna šema
  - Konceptualna (Logička) šema
  - Intena (Fizička) šema
- Šeme se memorišu u **sistemskom katalogu** DBMS-a



Arhitektura 3 šeme DBMSa

Arhitektura tri šeme DBMS-a
(ANSI-SPARC arhitektura)



- Svi DBMS-i ne razlikuju sva tri nivoa
- Cilj arhitekture 3-šeme je odvojiti pogled korisnika baze podataka od načina njene fizičke reprezentacije

Šeme, preslikavanja, instance

- **Šema baze podataka** je **opis baze podataka**
- U arhitekturi 3 šeme postoje
  - **Eksterna šema** (često se naziva i **podšema**): ima ih više
  - **Konceptualna šema**: postoji jedna za bazu podataka
  - **Interna šema**: postoji jedna za bazu podataka
- DBMS je odgovoran za **preslikavanja** između ovih šema
  - Preslikavanje konceptualne u internu šemu i obrnuto
  - Preslikavanje eksterne u konceptualnu šemu i obrnuto

- **Instancu baze podataka** čine **podaci** koji se trenutno nalaze u bazi podataka
- Jednoj šemi baze podataka može odgovarati više instanci

- Šema baze podataka se naziva i **intenzija** baze podataka, a instanca **ekstenzija** ili **stanje** baze podataka

Eksterni nivo (nivo pogleda)

- Korisnički pogled na bazu podataka
- Svaki pogled na bazu podataka
  - opisuje samo onaj deo baze podataka koji je od interesa za konkretnu grupu korisniku i
  - prikriva ostali deo baze podataka od ove grupe korisnika
- Sadrži veliki broj **eksternih šema** ili **pogleda korisnika**
- Različiti pogledi mogu imati različite reprezentacije istih podataka

Konceptualni (logički) nivo

- Pogled na bazu podataka **svih korisnika** baze podataka
- Opisuje koji se podaci pamte u bazi podataka, kako su ti podaci struktuirani i kako su povezani
- Sadrži **konceptualnu šemu** koja predstavlja **globalni opis baze podataka**
- Opisuje:
  - entitete, atribute i njihove veze,
  - ograničenja nad podacima,
  - semantičke informacije o podacima,
  - sigurnost i integritet informacija
- **Ne sadrži** nikakve detalje o **načinu memorisanja** podataka!

Interni nivo

- Fizička reprezentacija baze podataka na računaru
- Sadrži **internu (fizičku) šemu** koja opisuje kako su podaci memorisani u bazi podataka
- Opisuje sve detalje o memorisanju podataka, definiše strukturu i organizaciju fajlova koji se koriste za smeštanje baze podataka
- Koristi usluge fajl sistema OS-a za smeštanje podataka na memorijski medijum, za formiranje indeksa, za pretraživanje podataka itd.
- Na internom nivou se definiše:
  - Dodela prostora na disku za podatke i indekse
  - Opis slogova
  - Smeštanje slogova
  - Kompresija podataka
  - Tehnike enkripcije podataka

Fizički nivo

- Ispod internog nivoa je **fizički nivo** kojim može da upravlja OS po direktivama DBMS-a
- Funkcije OS-a i DBMS-a na fizičkom nivou nisu strogo razgraničene, pa variraju od jednog do drugog DBMS-a
- Neki DBMS-i koriste metode pristupa OS-a, dok drugi imaju sopstvene fajl sisteme

Opis i memorisanje podataka u DBMS-u

- Korisnici baze podataka
  - su u nekom realnom svetu, i
  - podaci memorisani u bazi podataka predstavljaju neke aspekte tog realnog sveta (mini svet)
  - Primer: baza podataka Fakultet
- DBMS omogućava korisnicima
  - da **opišu** (definišu) podatke koje žele memorisati u bazi podataka
  - Za opis podataka koristi se **model podataka**

Modeli podataka

- **Model podataka** je integrisana kolekcija koncepata za opis i manipulaciju podacima, vezama između podataka i ograničenjima nad podacima i vezama
  - To je matematička apstrakcija koja se koristi za projektovanje modela realnog sistema

- Model podataka ima tri **komponente**:
  - **Strukturnu** koja se sastoji od skupa pravila prema kojima se baza podataka može konstruisati
  - **Integritetnu** koja definiše ograničenja nad vrednostima atributa, veze između podataka i međusobnu uslovljenost podataka
  - **Operacijsku** koja definiše operacije nad strukturama podataka i koja modelira dinamičke osobine realnog sistema

Tipovi modela podataka

- **Model podataka je kolekcija koncepata visokog nivoa** kojima se opisuje tip, struktura i relacije među podacima

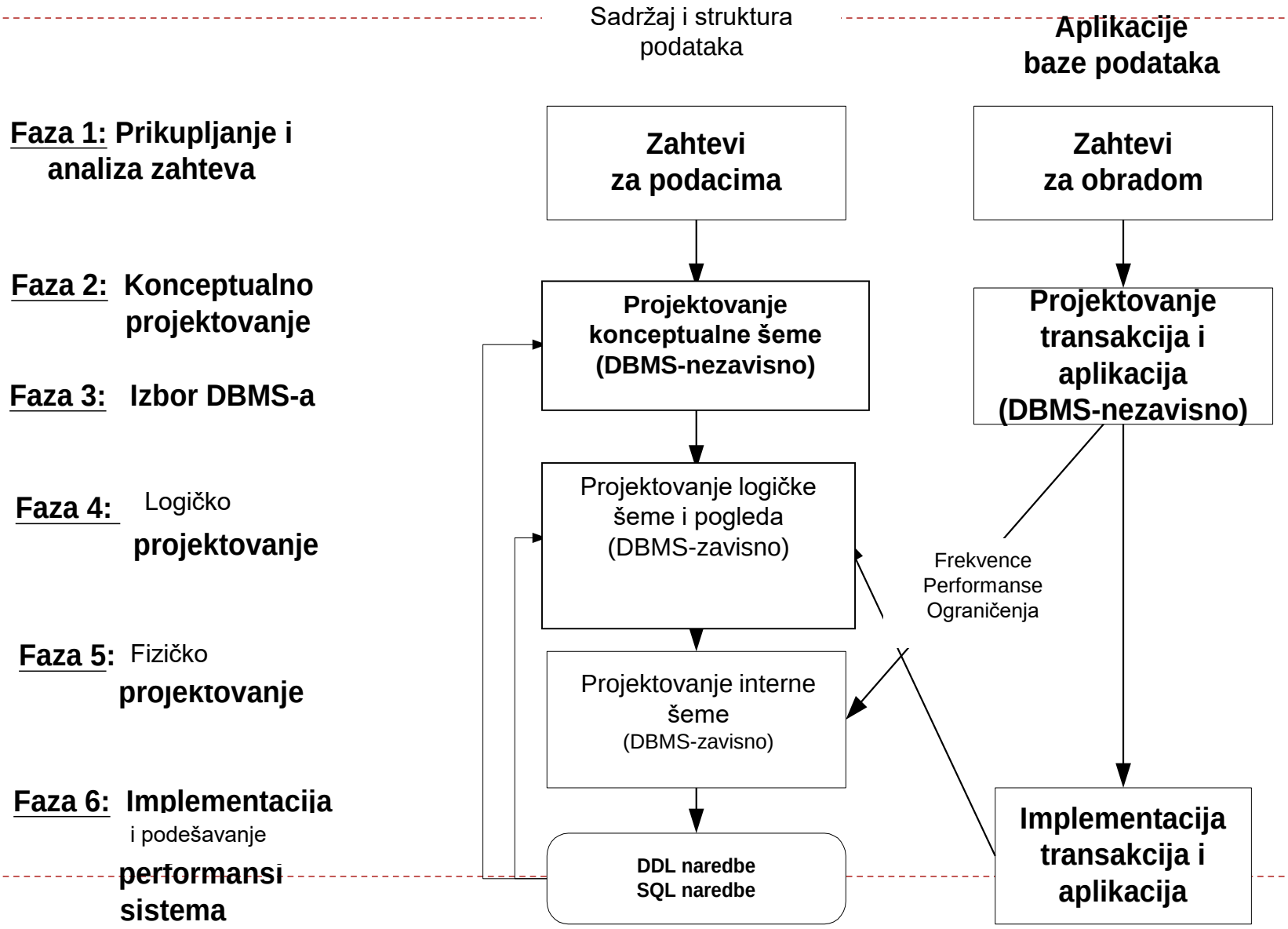
- U odnosu na ANSI-SPARC arhitekturu DBMS-a modeli podataka mogu biti:
  - Eksterni modeli podataka – služe za predstavljanje pogleda korisnika na realni svet
  - **Konceptualni modeli podataka** – služe za predstavljanje logičkog pogleda ili pogleda zajednice na realni sistem nezavisno od DBMS-a
  - Interni modeli podataka – služe za predstavljanje konceptualne šeme na takav način da je razumljiva za DBMS

Projektovanje baze podataka

- **Problem:**
  - Projektovati **logičku** i **fizičku** strukturu jedne ili više baza podataka da bi se zadovoljile informacione potrebe korisnika u organizaciji za definisani skup operacija

- **Ciljevi:**
  - Zadovoljiti informacione zahteve specificiranih korisnika i aplikacija
  - Obezbediti prirodno i lako za razumevanje struktuiranje informacija
  - Podržati zahteve obrade i zahteve performansi (vreme odgovora, vreme obrade i memorijski prostor)

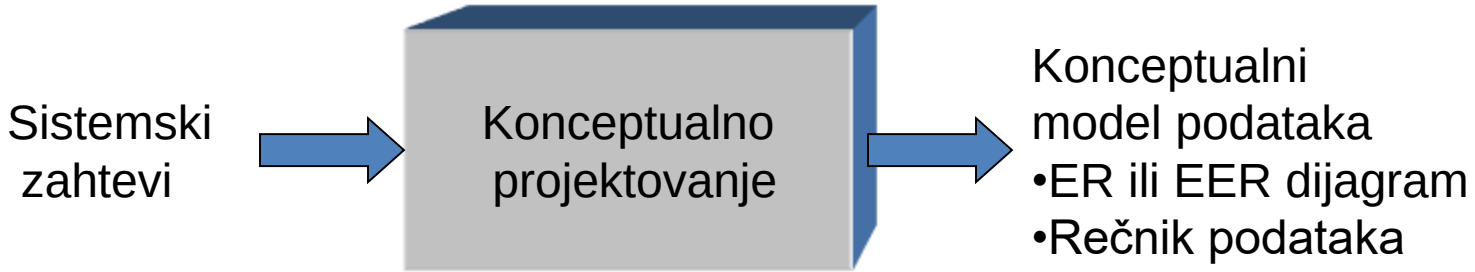
Proces projektovanja baze podataka



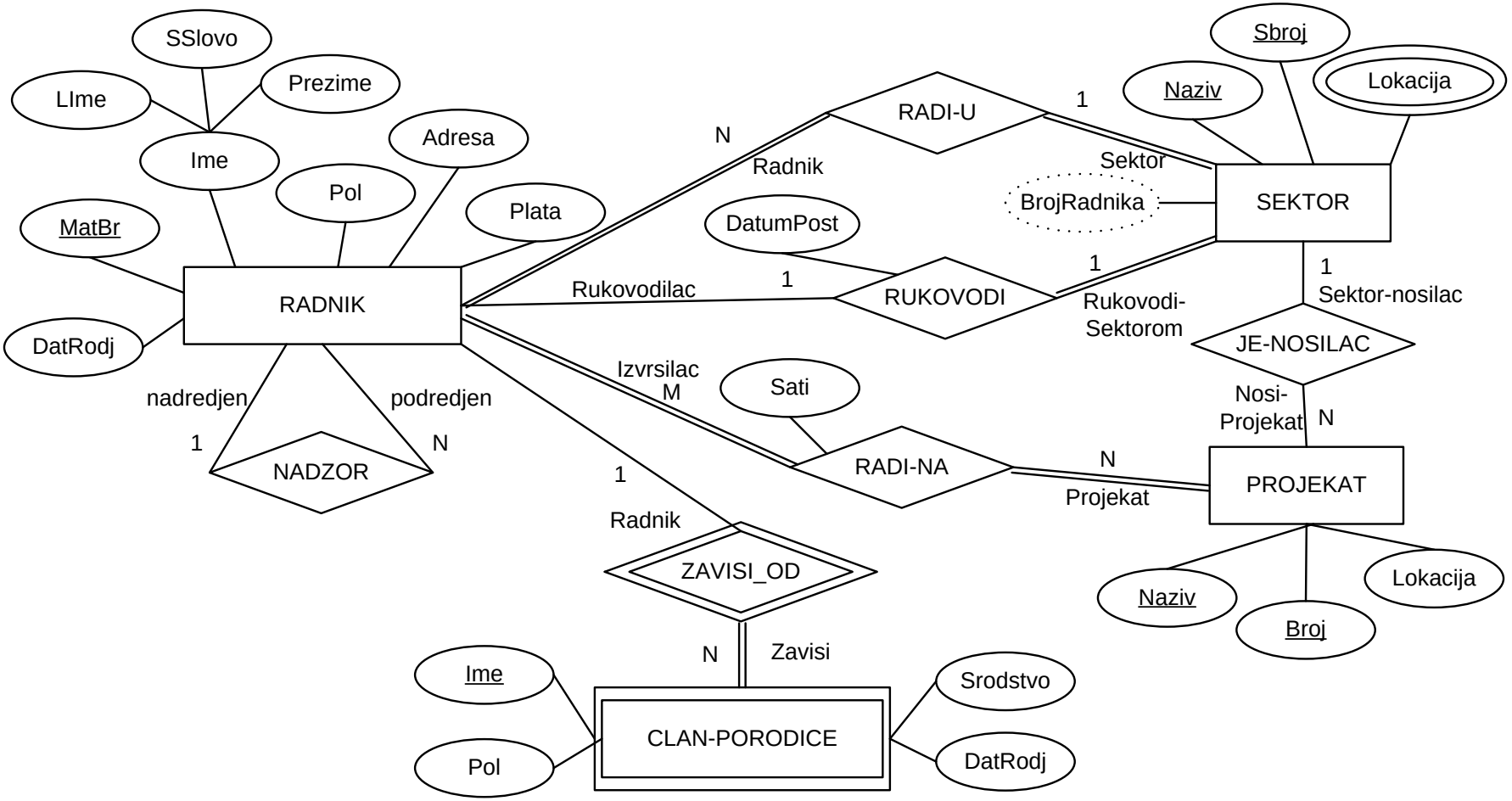
Faza	Korak
1. Konceptualno projektovanje **Cilj:** **Projektovanje ER ili EER modela baze podataka** **Izlaz:** **ER ili EER dijagram i opis dijagrama u Rečniku podataka**	1. Identifikovanje tipova entiteta 2. Identifikovanje tipova veza 3. Identifikovanje atributa za tipove entiteta i veza 4. Određivanje domena atributa 5. Određivanje ključeva kandidata i izbor primarnog ključa 6. Razmatranje upotrebe koncepata proširenog modeliranja 7. Provera da li model zadovoljava transakcije 8. Provera da li model zadovoljava postavljene zahteve
2. Logičko projektovanje **Cilj:** **Projektovanje relacionog modela baze podataka** **Izlaz:** **Šema relacione baze podataka**	1. Preslikavanje konceptualnog u relacioni model 2. Validacija relacija korišćenjem normalizacije (cilj je da svaka relacija bude bar u BCNF) 3. Definisanje ograničenja integriteta 4. Provera da li model zadovoljava transakcije 5. Provera da li model zadovoljava postavljene zahteve
3. Implementaciono projektovanje **Cilj:** **Projektovanje relacione šeme baze podataka za Oracle DBMS** **Izlaz:** **SQL opis implementacione šeme relacione baze podataka**	1. Prevođenje logičkog modela podataka u implementacioni model za ciljni DBMS (Oracle) 2. Provera da li model zadovoljava transakcije 3. Provera da li model zadovoljava postavljene zahteve 4. Opis implementacione šeme putem jezika za opis podataka (podskup SQLa) i mehanizama izabranog DBMSa

Konceptualno projektovanje baze podataka

- Cilj: Projektovati konceptualni model baze podataka koji što vernije opisuje realni sistem za koji se projektuje baza podataka
- Koristi se neki model podataka visokog nivoa
  - ER ili EER model podataka



Primer ER modela baze podataka PREDUZEĆE



(E)ER model podataka

- (Enhanced) Entity Relationship (ER) model
- Za konceptualno projektovanje baze podataka
- ER dijagram za grafičko predstavljanje
- Nazivi:
  - Model entiteta i veza
  - Model entiteta i poveznika
  - Model objekat-veze
- Autor Chen, 1976
- Postoji više verzija ER modela i više grafičkih notacija

Strukturna komponenta ER modela podataka

- Dva osnovna pojma su u osnovi ovog modela:
  - Entitet
  - Veza
- Osnovna ideja
  - Realni sistem (realni svet ili neki njegov deo - *minisvet*) se može opisati pomoću ova dva osnovna „koncepta“

- **Entitet** u realnom svetu je “nešto” što se može jednoznačno identifikovati
  - Može se odnositi na realni subjekat, objekat, događaj, pojavu ili apstraktni pojam
  - Entitet može biti objekat koji **fizički** egzistira (npr. osoba, kola, kuća ili radnik) ili objekat koji **konceptualno** egzistira (npr. kompanija, posao ili predmet na univerzitetu)
- **Veza** određuje odnos (vezu) između dva ili više entiteta

Entitet u ER modelu

Entitet je

subjekat, objekat, događaj, pojava ili apstraktni pojam o kome se prikupljaju, memorišu, obrađuju i prezentiraju podaci u automatizovanim informacionim sistemima i koji se može **jednoznačno identifikovati** i na taj način izdvojiti u skupu sličnih entiteta

Primeri Entiteta za mini svet Preduzeće i mini svet Fakultet:

- **Primer PREDUZEĆE:**
  - radnik
  - sektor (organizaciona jedinica)
  - radno mesto
  - projekat
  - plan
- **Primer FAKULTET:**
  - student
  - pofesor
  - predmet
  - laboratorija
  - udžbenik

Kandidati za entitete u realnom sistemu

- Organizacione jedinice
  - fakultet, katedra, biblioteka,…
- Lokacije
  - učionica, raskrsnica, radarska pozicija, laboratorija,…
- Uloge
  - radnik, službenik, student, profesor, stanovnik, referent,…
- Događaji koji se pamte
  - ispit, utakmica, predavanje, ispit,…
- Uređaji
  - proizvodna traka, računar, skener, radar, antena, …
- Drugi sistemi sa kojima naš sistem interaguje
  - IS ministarstva, IS univerziteta, baza podataka stanovništva,…

Skup entiteta

- Entiteti koji egzistiraju u ljudskom intelektu kao predstave realnih subjekata, objekata ili događaja mogu se klasifikovati u **skupove sličnih entiteta**

- **Primeri skupa entiteta**
  - studenti jednog fakulteta
  - radnici jednog preduzeća
  - proizvodi jednog preduzeća
  - zgrade jednog preduzeća
  - uplate na žiro račun

Tip entiteta u ER modelu

- **Tip entiteta** *(egl. Entity Type)* je **model** skupa entiteta

- Obeležavanje tipa entiteta
  - **E(A1, A2, ... , An)**
  - gde je E **ime** tipa entiteta, a Ai | i = 1,n **atributi** odabrani za modeliranje entiteta E

- **Primer :**
  - **RADNIK1** (IME, DATUM-ROĐENJA, ADRESA, TELEFON, SEKTOR, LD, ČLANOVI-PORODICE)
  - **RADNIK2** (IME, SEKTOR, LD)
  - **RADNIK3** (MATIČNI-BROJ-RADNIKA, MATIČNI-BROJ-STANOVNIKA, IME-RADNIKA)

Tipovi entiteta RADNIK1, RADNIK2, RADNIK3 modeliraju na razne načine entitet RADNIK iz realnog sistema

Skup entiteta u ER modelu

- **Skup entiteta** *(engl. entity set)* je kolekcija svih entiteta određenog tipa entiteta u bazi podataka u nekom trenutku
- Za skup entiteta se koristi **ime tipa entiteta**

- Primer:
  - Ime RADNIK se koristi i za tip entiteta i za trenutni skup svih entiteta radnik u bazi podataka kojoj taj tip entiteta pripada

- Tip entiteta opisuje **šemu** ili **intenziju** skupa entiteta koji dele istu strukturu
- Kolekcija entiteta određenog tipa entiteta grupisana u skup entiteta se takođe naziva **ekstenzija** tipa entiteta

Pojava (instanca) tipa entiteta

- Pojava ili instanca tipa entiteta se odnosi na *skup podataka o određenom entitetu* (iz skupa entiteta koji je modeliran odgovarajućim tipom entiteta)

- **Primer**
  - Dve pojave (instance) tipa entiteta RADNIK2(IME, SEKTOR, LD):

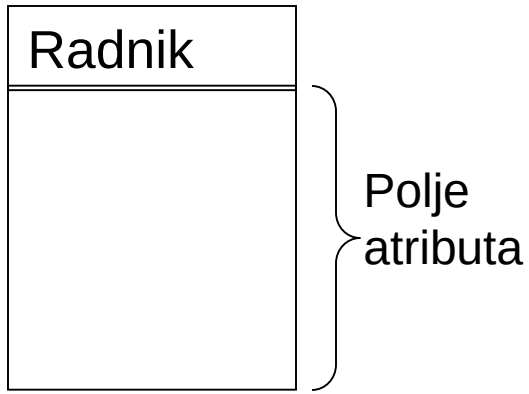
    SAVA KRSTIĆ, INSTITUT, 85000

    JOVAN RISTIĆ, TEST, 57000

Predstavljanje tipa entiteta u ER dijagramu

ERD notacija:
Pravougaonik sa upisanim imenom tipa entiteta

UML notacija



Ime entiteta: RADNIK, odnosno Radnik
Ograničenje: Imena entiteta u jednom ER dijagramu su jedinsvena

Atribut (obeležje) u ER modelu

- Svaki objekat realnog sveta ima osobine **(obeležja)** koji se kod tipa entiteta predstavljaju **atributima** – to svojstva/obeležja koja ga opisuju, i preko kojih se čuvaju podaci
  - Svi entiteti jednog skupa imaju bar jedno zajedničko svojstvo na osnovu koga su svrstani u isti skup
    - Obično postoji više takvih zajedničkih osobina koji opisuju skup entiteta
  - Ova svojstva se kod Tipa entiteta nazivaju atributima
- Primer
  - Entitet RADNIK može imati atribute
    - Matični broj radnika, ime, prezime, datum rodjenja,…
  - Entitet STUDENT može imati atribute
    - Broj indeksa, ime, prezime, datum rođenja, godina studija, …

Obeležavanje atributa

- **Stil A**
  - Obeležavaju se velikim slovom sa crticom ili znakom za podvlačenje kao poveznikom između reči
  - **Primer :**

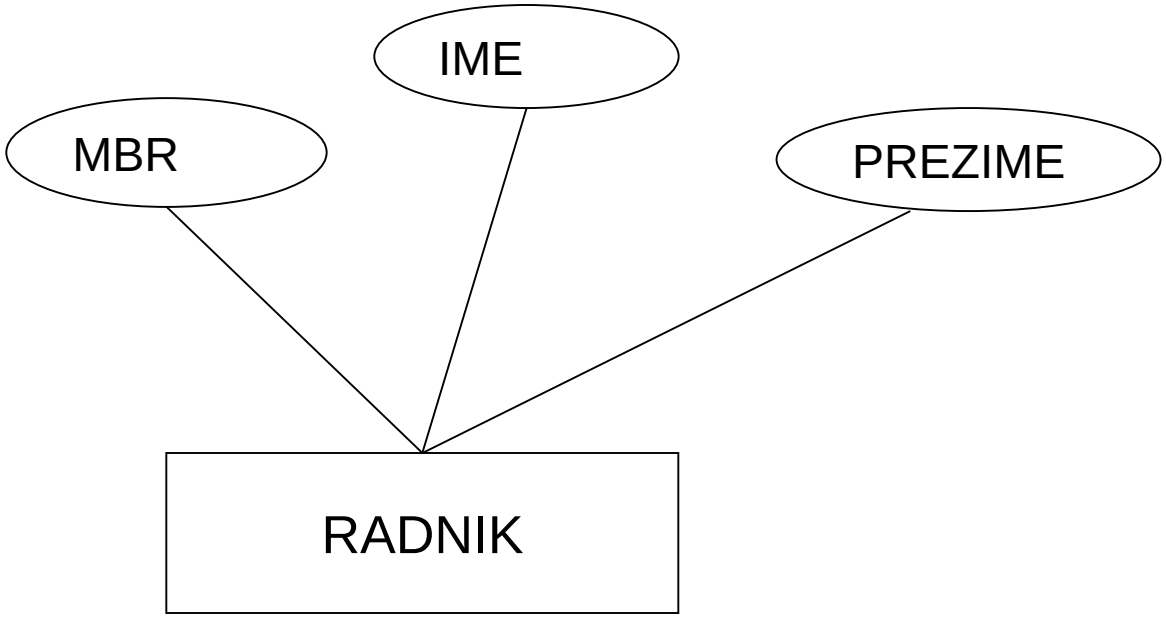
IME	IME
DATUM_UPISA	DATUM-UPISA
BOJA_KOLA	BOJA-KOLA

- Stil B (UML notacija)
  - Obeležavaju se nizom reči koje se pišu malim slovima, osim prvog slova svake reči. Nema delimitera između reči.
  - **Primer :**

Ime
DatumUpisa
BojaKola

Predstavljanje atributa u ER dijagramu

Označavanje: elipsa sa upisanim imenom atributa, povezana linijom sa pravougaonikom koji označava odgovarajući tip entiteta kome atribut pripada



Slika: Tip entiteta RADNIK sa atributima MBR, Ime i Prezime

Prosti i složeni atributi

- Atribut je **prost (elementaran)** ako se dalje ne može dekomponovati ili ako se u konretnoj situaciji ne dekomponuje na komponente koje čine atribut
- **Primer elementarnih atributa**

  OCENA-STUDENATA

  BOJA-AUTOMOBILA

  NAZIV-PROIZVODA
- Vrednost elementarnog atributa je **elementarni (prost) podatak**
- Atribut je **složen (kompozitan)** ako je sastavljen od više elementarnih atributa
- **Primer složenih atributa**

  ADRESA-STUDENTA

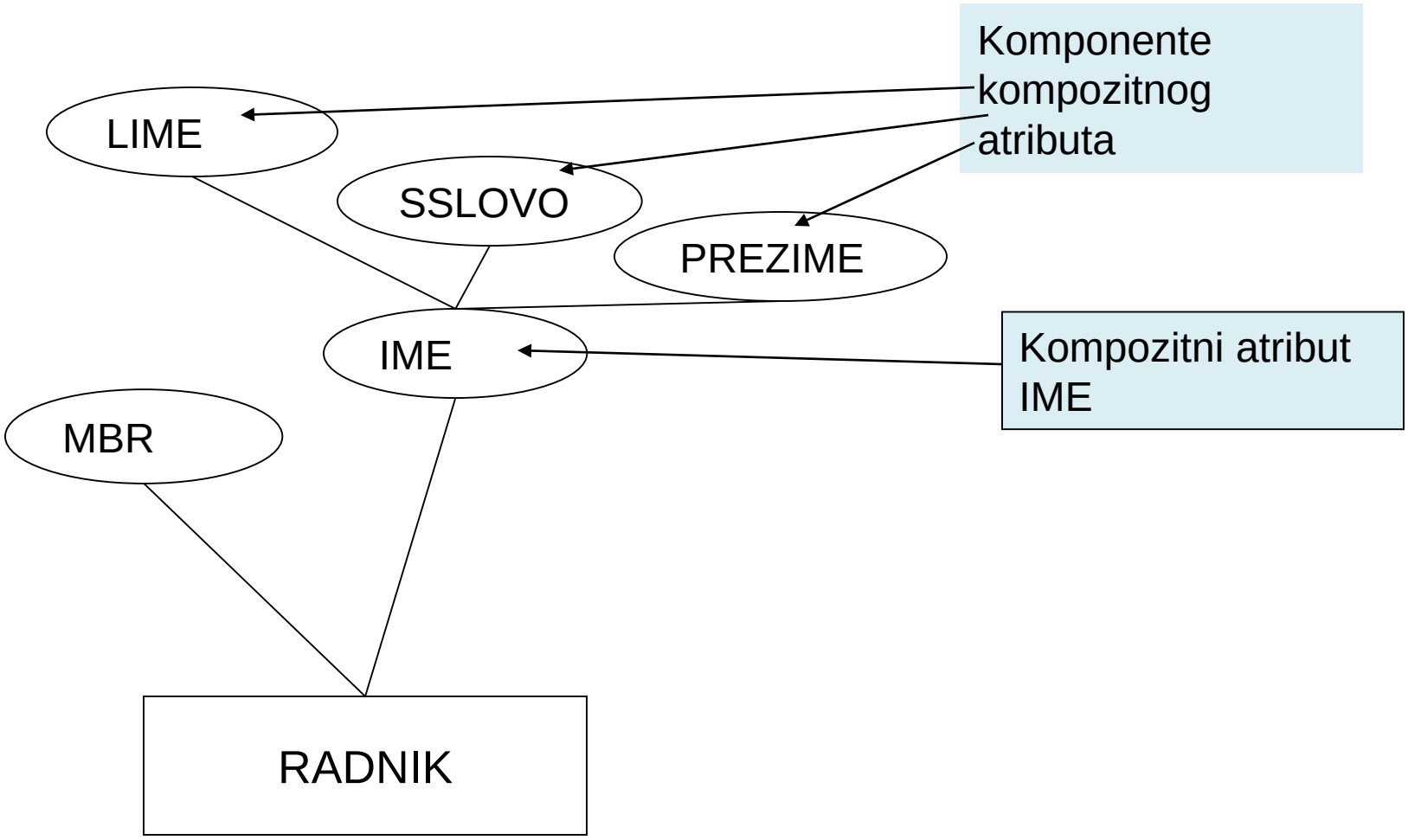
  IME-STUDENTA

  DATUM-UPISA
- Vrednost složenog atributa je **složeni (strukturni) podatak**

Dekomponovanje složenih atributa

- Složeni atribut se može podeliti (dekomponovati) na manje delove koji predstavljaju više osnovnih atributa sa nezavisnim značenjem
- Primeri:
  - **ADRESA_STUDENTA** se može dekomponovati na
    - ULICA, BROJ_ZGRADE, BROJ_STANA, GRAD
  - **IME_STUDENTA** se može dekomponovati na
    - LIME, SREDNJE_SLOVO, PREZIME
  - **DATUM_UPISA** se može dekomponovati na
    - DAN, MESEC, GODINA
- Mi ćemo dekomponovan složeni atribut označavati na sledeći način:
  - **ADRESA_STUDENTA(** ULICA, BROJ_ZGRADE, BROJ_STANA, GRAD)
  - **IME_STUDENTA** (LIME, SREDNJE_SLOVO, PREZIME)
  - **DATUM_UPISA** (DAN, MESEC, GODINA)

Predstavljanje složenih atributa u ER dijagramu



Složeni atributi – kada i zašto?

- Složeni atributi su korisni za modeliranje situacije gde se korisnici kod pretraga ponekad koriste atribut kao celinu, a ponekad koriste pojedine komponente tog atributa
- Ako se složeni atribut uvek referencira kao celina treba ga modelirati kao prost atribut

- Primer:

  ADRESA_STUDENTA(ULICA, BROJ_ZGRADE, BROJ_STANA, GRAD)

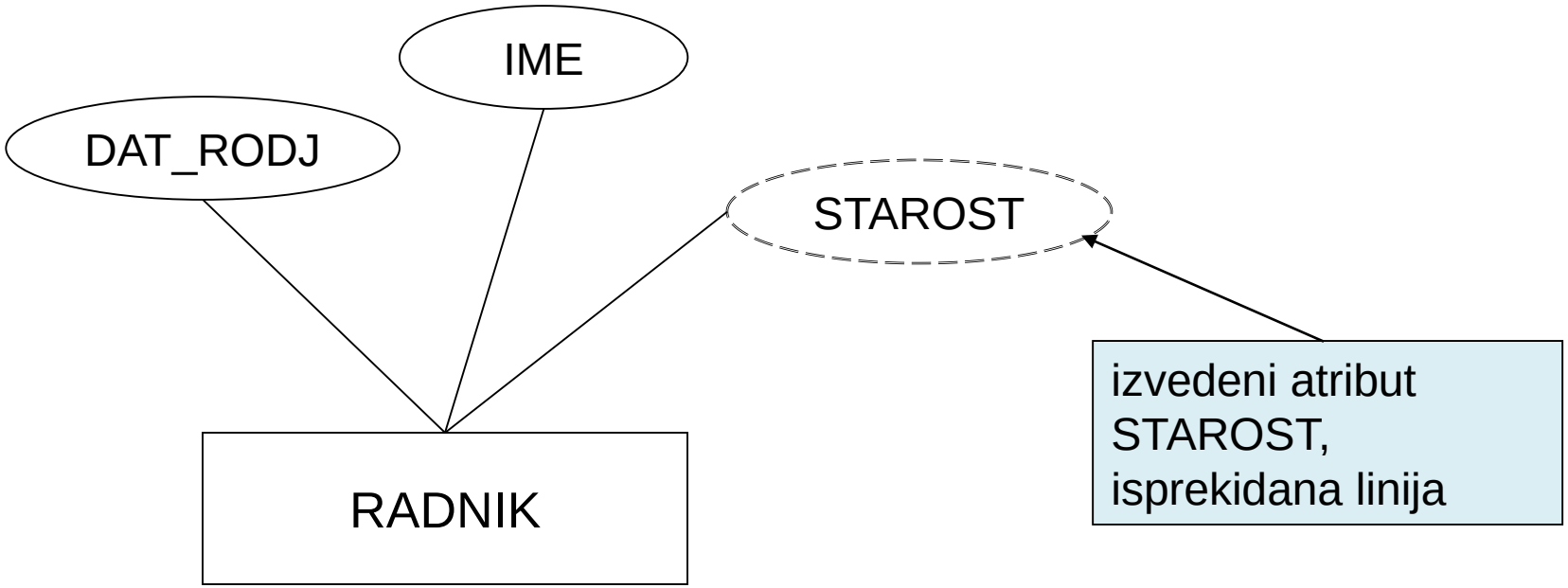
  Ako nema potrebe za nezavisnim referenciranjem pojedinih komponenti adrese, adresu studenta treba modelirati kao prost atribut

Izvedeni atribut

- To je atribut čije se vrednosti dobijaju primenom nekog algoritma na vrednosti drugih atributa (elementarnih, složenih, izvedenih)
- Vrednost izvedenog atributa je **izvedeni podatak**

- **Prmer 1**
  - Atribut SREDNJA_OCENA studenta se može dobiti primenom algoritma za izračunavanje srednje vrednosti na atribut OCENA u entitetu STUDENT
- **Primer 2**
  - Atribut BROJ_ZAPOSLENIH u preduzeću se izvodi primenom algoritma prebrojavanja pojava entiteta RADNIK

Predstavljanje izvedenih atributa u ER dijagramu



Domen atributa

- **Domen atributa** deifiniše **skup vrednosti atributa** (*engl.value set*) iz kojeg atribut uzima vrednosti
- **Obeležavanje domena**

  dom(ime_atributa)

- **Primer**

  dom(IME) = Character(15)

  dom(BOJA_KOLA) = (bela, crvena, zelena)

  dom(OCENA_STUDENTA) = (5, 6, 7, 8, 9, 10)

  dom(PRELAZNA_OCENA) = (6, 7, 8, 9, 10)

Pojava (instanca) atributa

- To je konkretna vrednost koju atribut uzima iz svog domena da predstavi podatak o određenom entitetu
  - Samo instanca atributa može predstavljati podatak
- **Primer**
  - Student PERA ILIĆ ima broj indeksa RE 5034/88.
  - Tada je podatak
    - PERA ILIĆ instanca atributa IME_STUDENTA, a
    - RE5034/88 instanca atributa BROJ_INDEKSA

Vrednost atributa

- Svaki entitet za svaki atribut definisan tipom entiteta ima određenu vrednost

- **Primer**

  RADNIK(IME,ADRESA,DATUM_ROĐENJA, KUCNI_TELEFON, ZANIMANJE,SEKTOR,LD)

  - Vrednosti atributa dve instance tipa entiteta RADNIK (dva konkretna entiteta odnosno dva radnika):

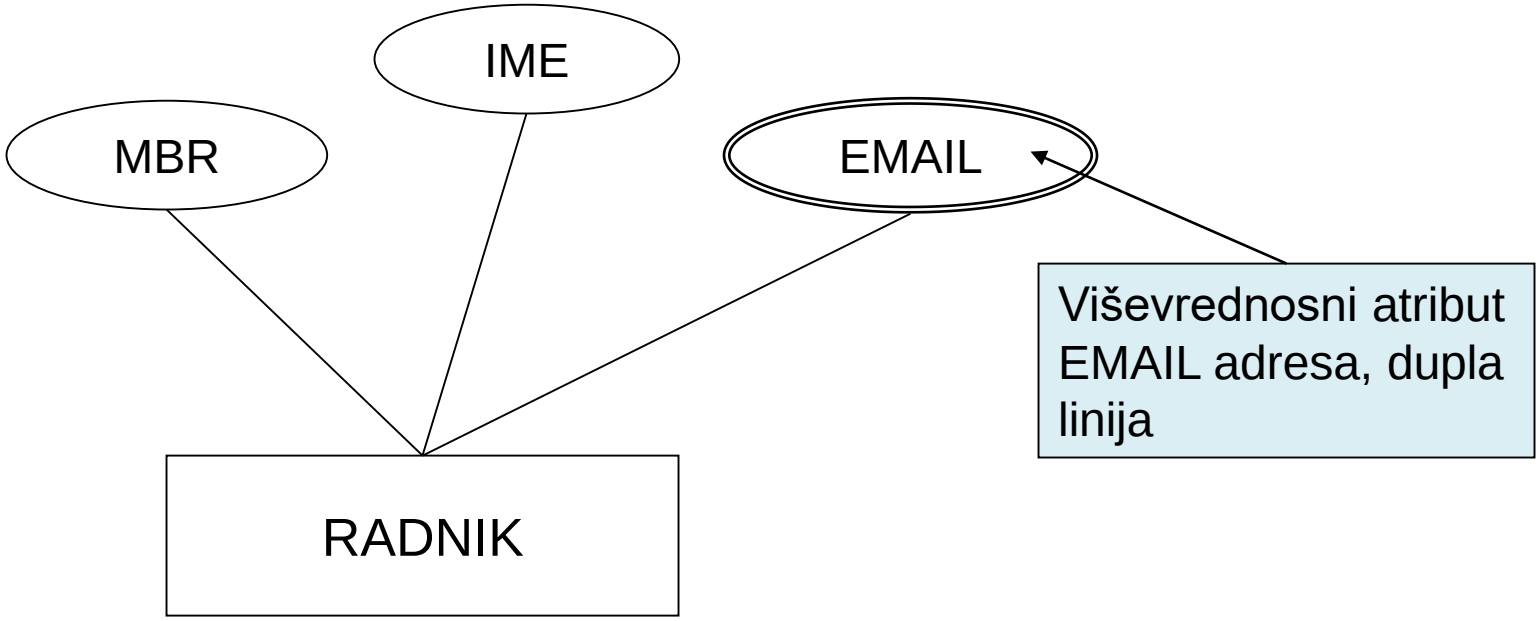
Pera Ilić, Niška16 18000 Niš YU, 05 - FEB – 68, 018 - 315 – 072, inženjer elektronike, Razvoj, 35000

Mina Stojanović, Majakovskog 29 18000 NIŠ YU, 25 - JUL – 85, 018 - 515 – 799, diplomirani pravnik, Zajedničke službe, 25000

Jednovrednosni i viševrednosni atributi

- Za određeni entitet jedan atribut može imati jednu ili više vrednosti
  - U prvom slučaju atribut je jednovrednosni, a u drugom viševrednosni
- **Primer**
  - Atribut DATUM_ROĐENJA entiteta RADNIK je jednovrednosni atribut jer radnik može imati samo jedan datum rođenja, npr.05-FEB-68
  - Atribut KUĆNI_TELEFON entiteta RADNIK je viševrenosni atribut jer radnik može imati više telefona, npr.520-505 i 224-061
  - Atribut EMAIL_ADRESA entiteta RADNIK je viševrenosni atribut jer radnik može imati više email adresa

Predstavljanje viševrednosnog atributa u ER dijagramu



NULL vrednost atributa

- Neki atributi mogu biti bez vrednosti za neki entitet
  - Tada se koristi NULL vrednost
- NULL vrednost predstavlja specijalno ograničenje domena
  - Putem ovog ograničenja se specificira da li atribut može imati nedefinisanu vrednost
- **Primer**
  1. Ako radnik IVANA GOCIĆ nema telefon u stanu, atribut TELEFON za ovog radnika će imati NULL vrednost
  2. Ako radnik IVANA GOCIĆ nije trenutno raspoređena ni u jednom sektoru, atribut SEKTOR će imati NULL vrednost
  3. Ako je adresa radnika IVANE GOCIĆ nepoznata, atribut ADRESA će imati NULL vrednost
- **Semantika NULL vrednosti**
  - Neprimerena vrednost (primeri 1 i 2)
  - Postojeća ali nepoznata vrednost (primer 3)

Ključ

- **Ključ** je atribut ili minimalni skup atributa koji ima jedinstvenu vrednost za sve pojave određenog tipa entiteta
- Svaki tip entiteta poseduje bar jedan ključ
- Tip entiteta može imati više ključeva - to su **ključevi kandidati**
  - jedan od ključeva kandidata je **primarni ključ**

- **Primer**

  U tipu entiteta

  RADNIK3 (MATIČNI_BROJ_RADNIKA, MATIČNI_BROJ_STANOVNIKA, IME_RADNIKA)

  ključ može biti MATIČNI_BROJ_RADNIKA i MATIČNI_BROJ_STANOVNIKA ukoliko važi pravilo da svi radnici imaju različite matične brojeve radnika i različite matične brojeve stanovnika

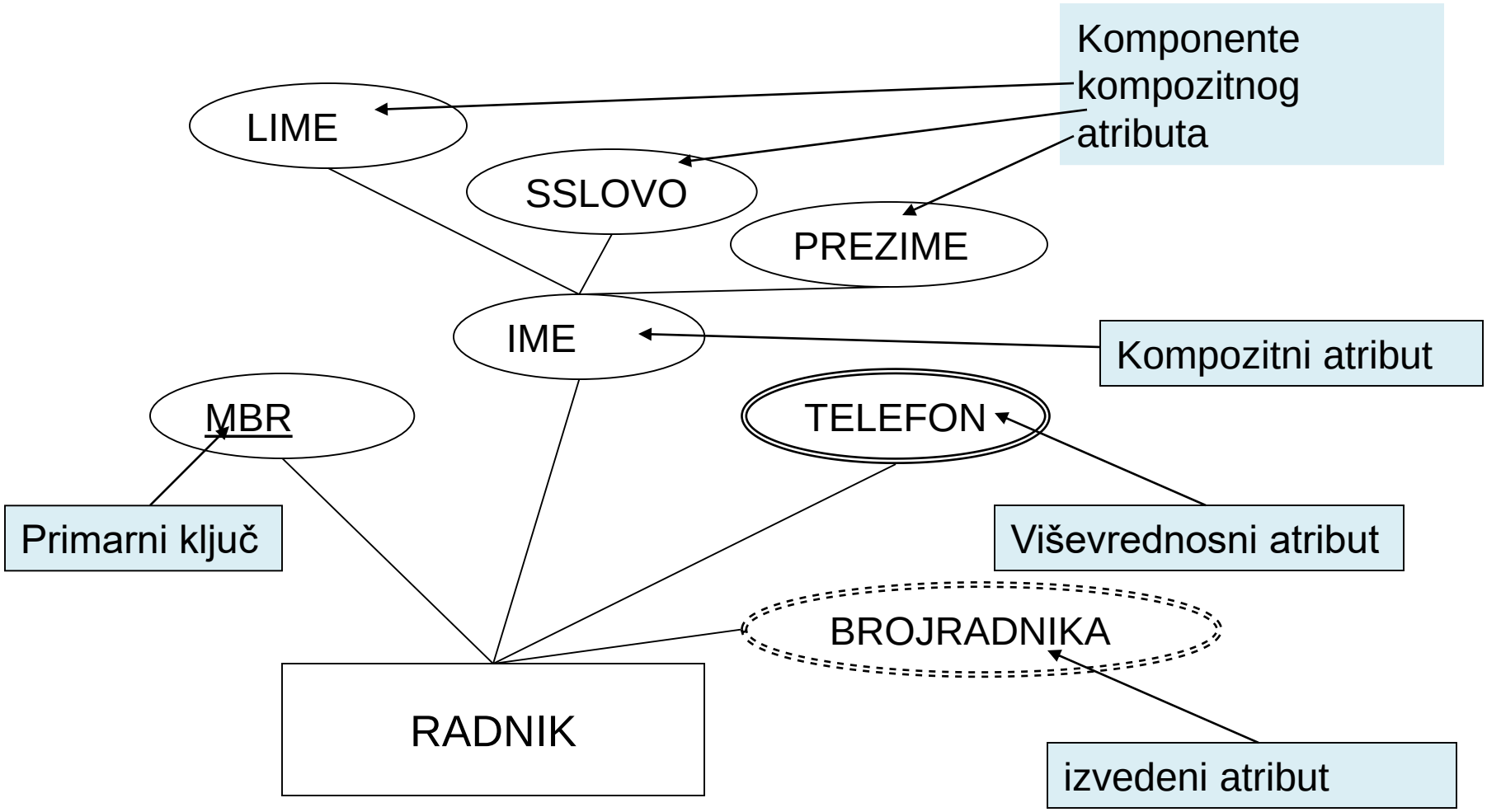
Obeležavanje primarnog ključa

- **Koriste se dva načina**
  - primarni ključ u tipu entiteta se **podvlači kontinualnom linijom**
  - iza imena primarnog ključa se stavlja povisilica (#)
- **Primer**

RADNIK3(MATIČNI_BROJ_RADNIKA, MATIČNI_BROJ_STANOVNIKA, IME)

RADNIK3(MATIČNI_BROJ_RADNIKA# , MATIČNI_BROJ_STANOVNIKA, IME)

Predstavljanje atributa u ER dijagramu



Skup poveznika

- U realnom sistemu između entiteta postoje određeni odnosi (relacije) koje treba modelirati

- **Skup poveznika** $R=\{e_1,e_2,\ldots,e_n | e_i \in E_i, i=1,\ldots,n\}$ predstavlja relaciju (u matematičkom smislu) između n ($n \geq 2$) skupova entiteta
  - Skupovi E_i ne moraju biti različiti
  - Svaka n-torka $(e_1,e_2,\ldots,e_n)$ u R predstavlja jedan poveznik
  - Svaki entitet u n-torci ima svoju ulogu
  - Ako se uloge entiteta u n-torki eksplicitno navedu tada redosled navođenja entiteta nije bitan
  - Između dva ista skupa entiteta može postojati više različitih skupova poveznika

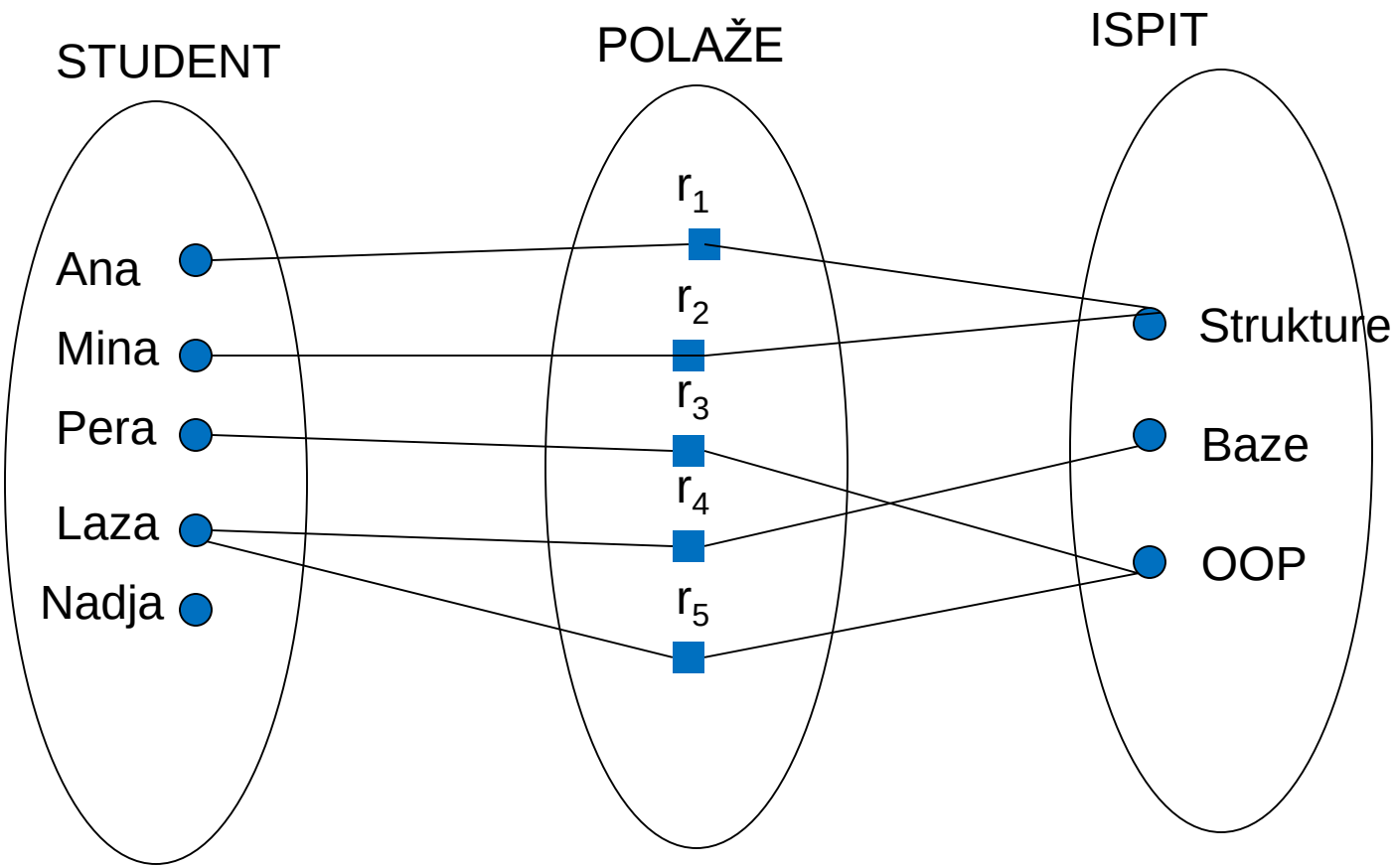
- Ako poveznik povezuje entitete istog skupa, naziva se **rekurzivnim**

Tip poveznika

- Tip poveznika je **model** skupa poveznika.
  - Obično se za naziv tipa poveznika koristi naziv skupa poveznika, tj. $R(E_1,E_2,\ldots,E_n;B_1,B_2,\ldots,B_m)$
  - Za svaki tip entiteta $E_i \mid i=1,n$ se kaže da participira u tipu poveznika R

- Primeri tipa poveznika
  - Modeliranje veze između radnika i projekata, radnik radi na projektima
    RADI-NA(RADNIK,PROJEKAT)
  - Modeliranje veze između radnika i projekata, radnik rukovodi projektom
    RUKOVODI(RADNIK,PROJEKAT)
  - Modeliranje veze između radnika i radnika, odnos nadređeni-podređeni
    JE-RUKOVODILAC(RADNIK,RADNIK)

Neke pojave tipa poveznika POLAŽE



Stepen tipa poveznika

- Stepen tipa poveznika je broj tipova entiteta koji participiraju u tipu poveznika
  - Tip poveznika stepena 1 naziva se rekurzivni
  - Tip poveznika stepena 2 se naziva binarni
  - Tip poveznika stepena 3 se naziva ternarni
  - ....
- Primer
  - Rekurzivni tipovi poveznika

    JE-RUKOVODILAC(RADNIK,RADNIK)

    U-BRAKU(OSOBA,OSOBA)

  - Binarni tipovi poveznika

    RADI-NA(RADNIK,PROJEKAT)

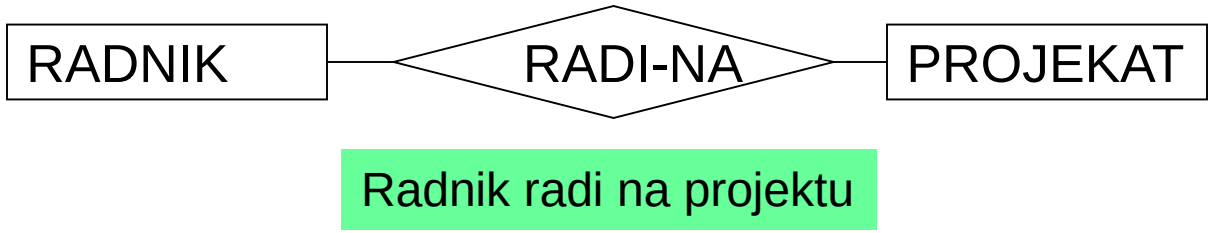
    RUKOVODI(RADNIK,PROJEKAT)

    JE-RUKOVODILAC(RADNIK,RADNIK)
  - Ternarni tip poveznika

    SNABDEVA(DOBAVLJAČ,DEO,PROJEKAT)

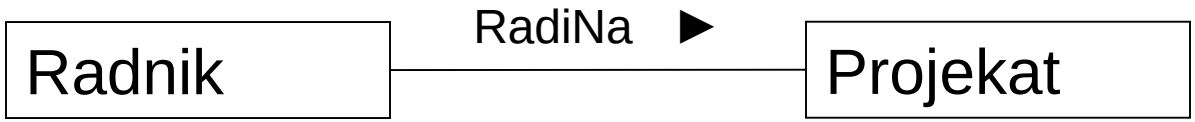
Predstavljanje tipa poveznika u ER dijagramu

ERD: romb, sa navedenim imenom tipa veze

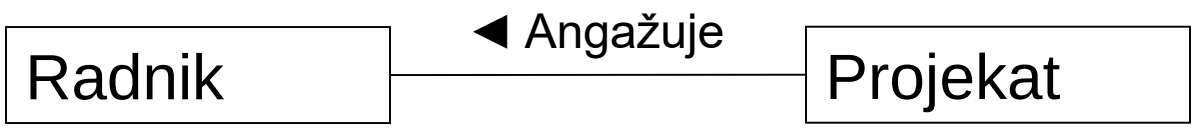


Radnik radi na projektu

UML notacija

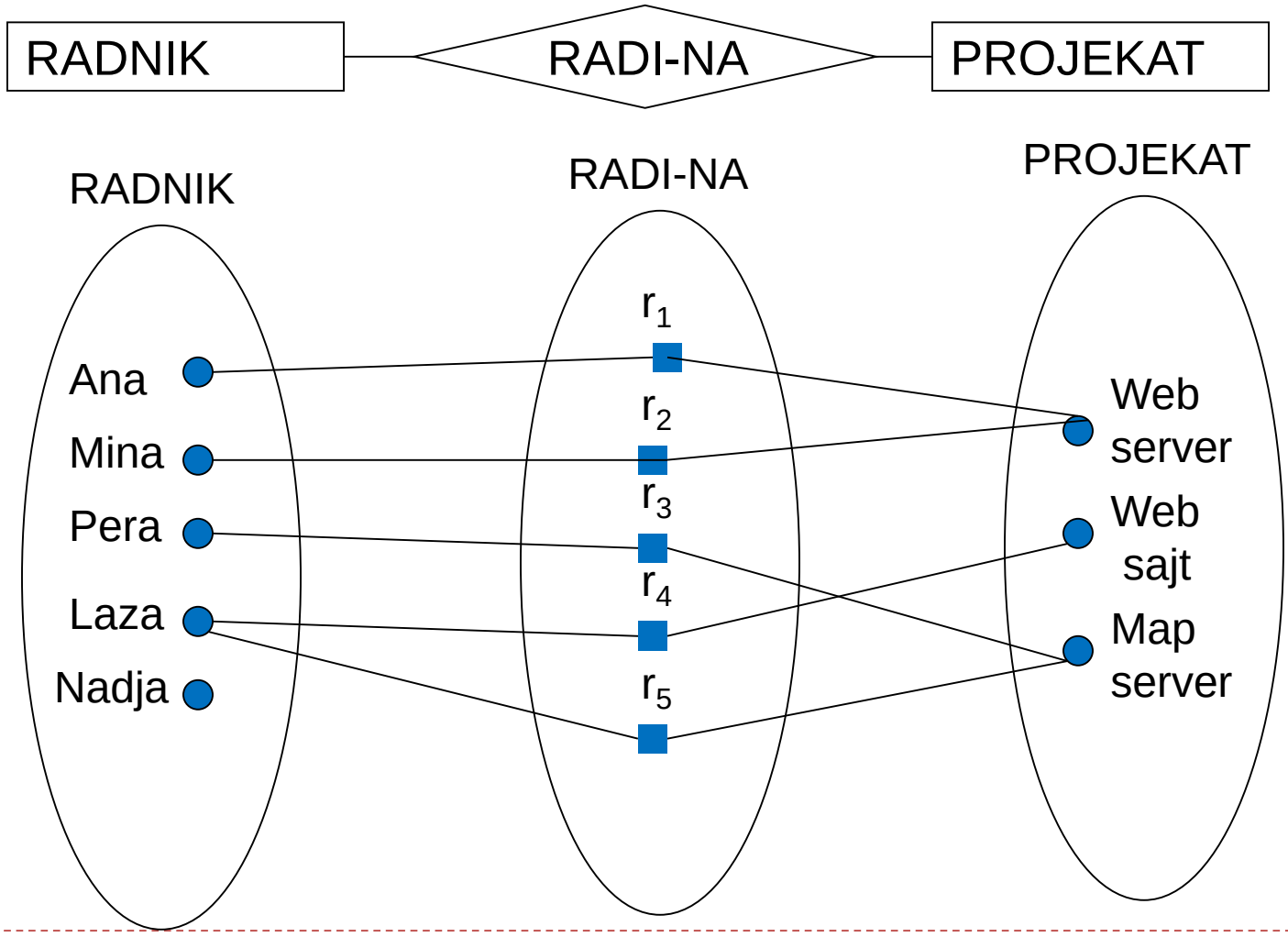


Radnik radi na projektu

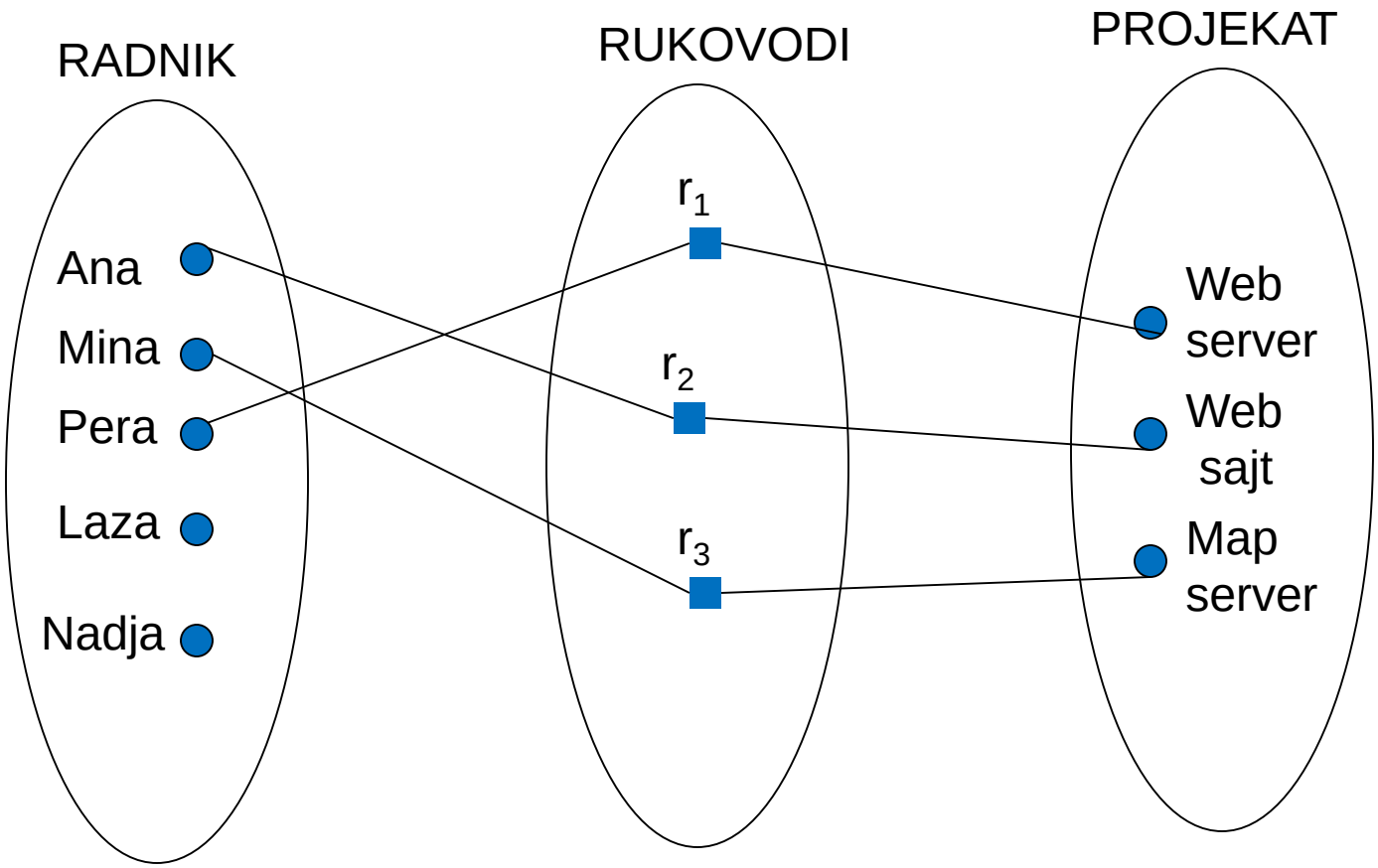


Projekat angažuje radnike

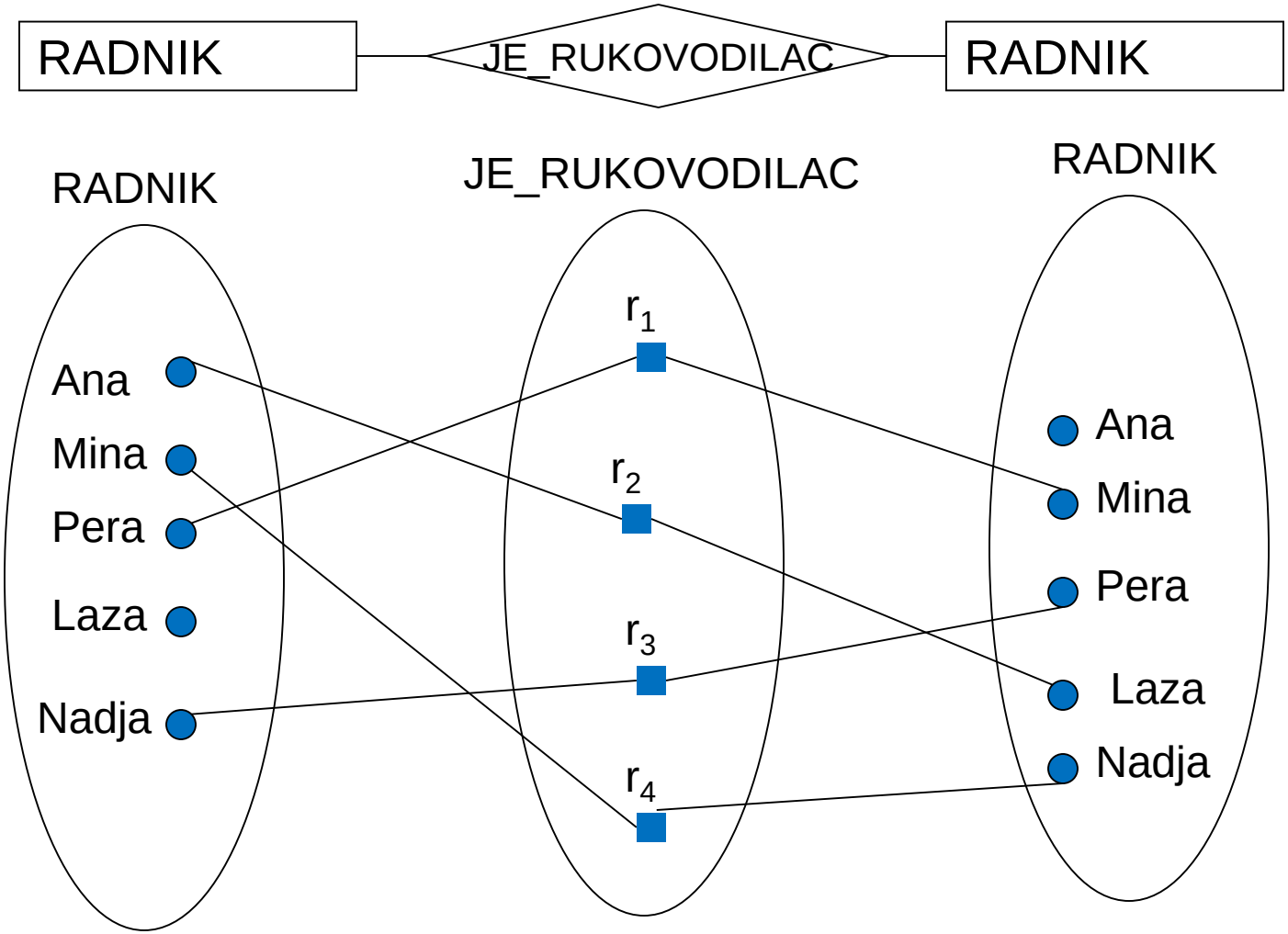
Neke pojave tipa poveznika RADI-NA



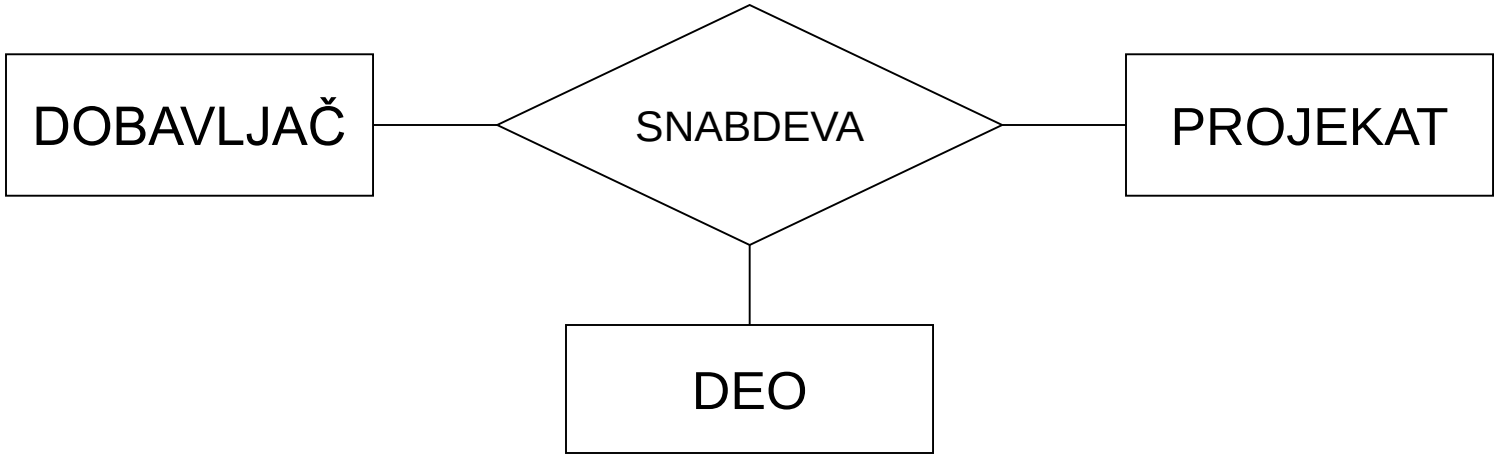
Neke pojave tipa poveznika RUKOVODI



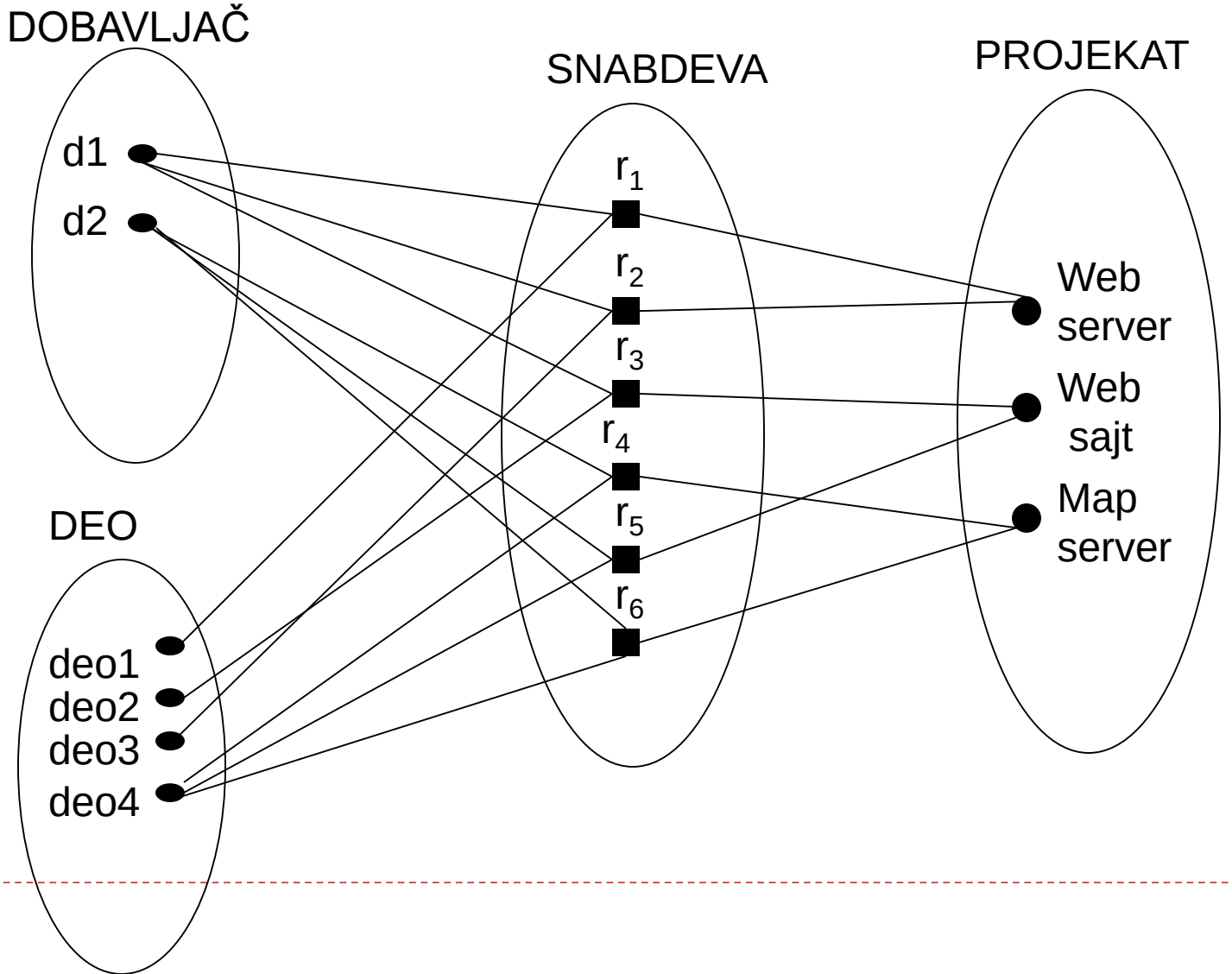
Neke pojave tipa poveznika
JE_RUKOVODILAC



Ternarni tip poveznika SNABDEVA



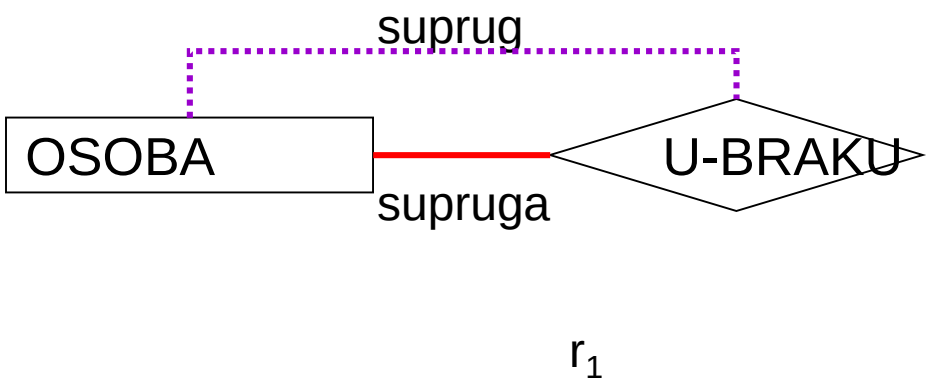
Neke pojave ternarnog tipa poveznika SNABDEVA

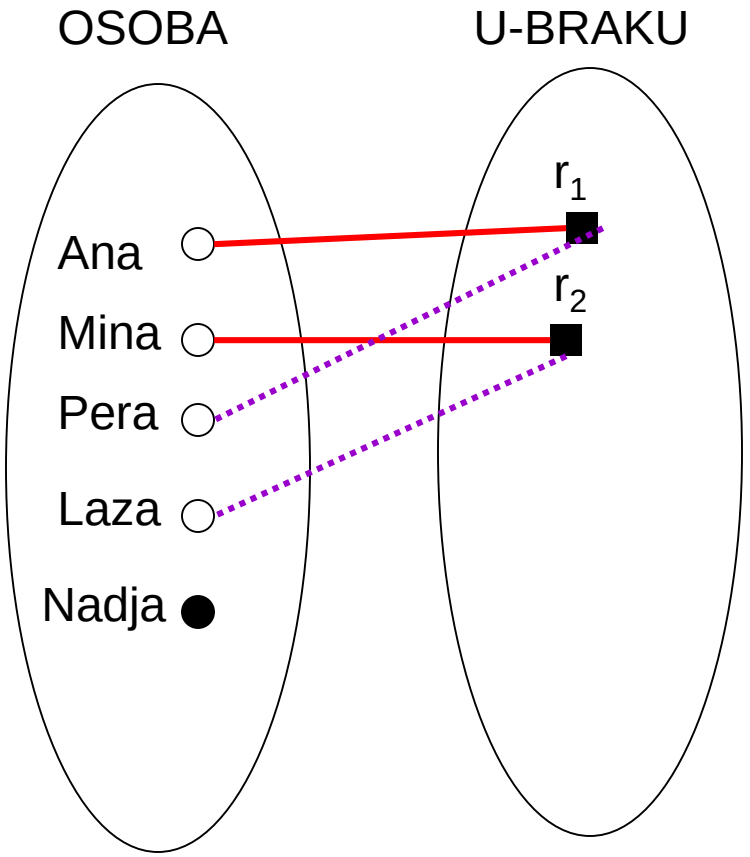


Ime uloge

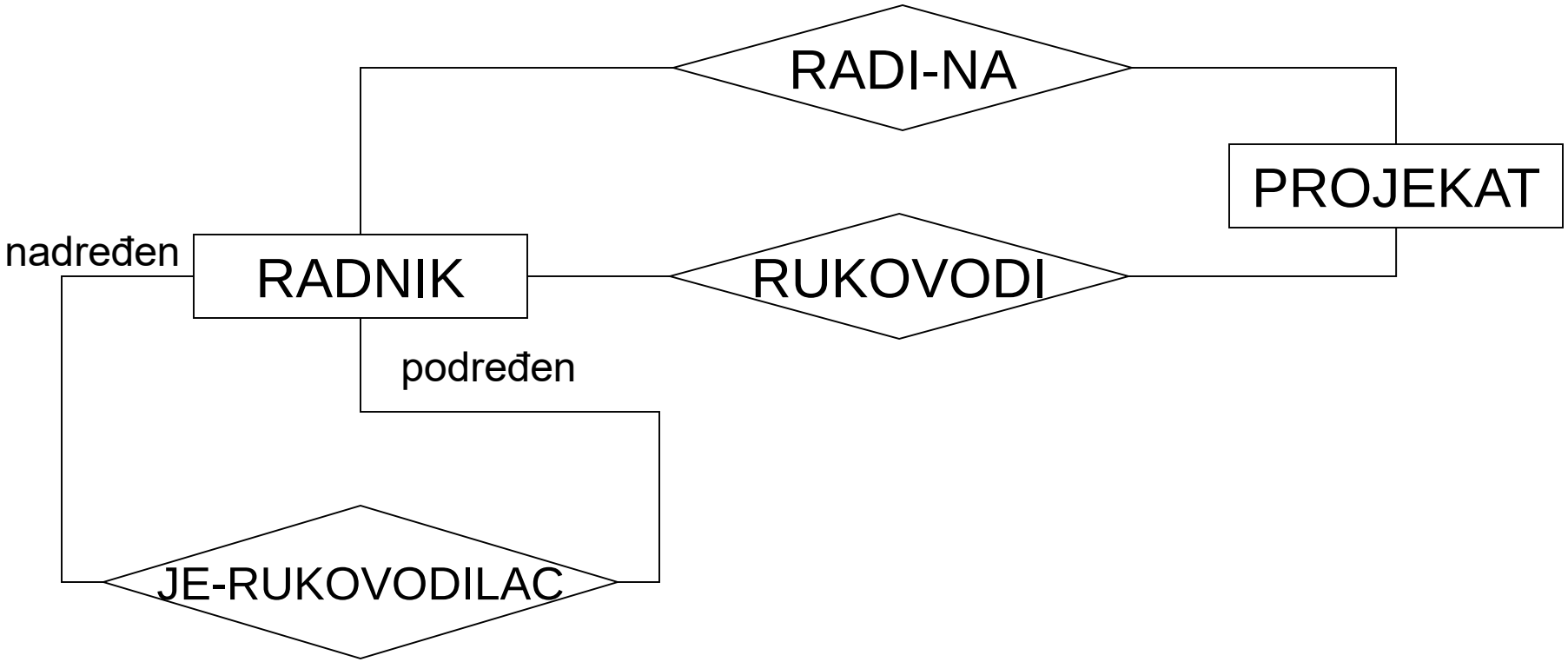
- Svaki tip entiteta koji participira u tipu poveznika ima posebnu **ulogu** u tom odnosu
- **Ime uloge** označava ulogu koju igra entitet koji participira u povezniku u svakoj instanci poveznika
- Pomaže u razumevanju odnosa koji poveznik modelira
- Ime uloge je posebno važno kod rekurzivnih poveznika
- Primer
  - U tipu poveznika RADI-NA(RADNIK,PROJEKAT) tip entiteta RADNIK je u ulozi **zaposlen**, a PROJEKAT u ulozi **poslodavac**
  - U tipu poveznika JE-RUKOVODILAC(RADNIK,RADNIK) tip entiteta RADNIK igra dvojaku ulogu, kao **nadređeni** i **podređeni** radnik u hijerarhiji rukovođenja

Neke pojave rekurzivnog tipa poveznika U-BRAKU





ER dijagram, označavanje uloga



Ograničenja nad tipom poveznika

- Tipovi poveznika imaju izvesna ograničenja koja limitiraju moguće kombinacije entiteta koji mogu participirati u skupu poveznika
- Ova ograničenja omogućavaju modeliranje prirode odnosa koji postoje među entitetima u realnom svetu
- Primer
  - Radnik **može** raditi na više projekata
  - Radnik **može** rukovoditi samo jednim projektom
  - Svaki radnik **mora** raditi bar na jednom projektu
  - Svaki radnik, osim glavnog menadžera preduzeća, **mora** imati rukovodioca

- Dva glavna tipa ograničenja nad tipom poveznika su
  - **Odnos kardinaliteta**
  - **Participacija**

Kardinalnost binarnog tipa poveznika

- Odnos kardinaliteta binarnog tipa poveznika R(E1,E2) specificira **maksimalni broj** instanci poveznika u kojima entiteti E1 i E2 mogu participirati
- Mogući odnosi kardinaliteta tipa poveznika R(E1,E2)
  - 1:1, 1:N, N:1, M:N, gde je N ≥0 i M≥0

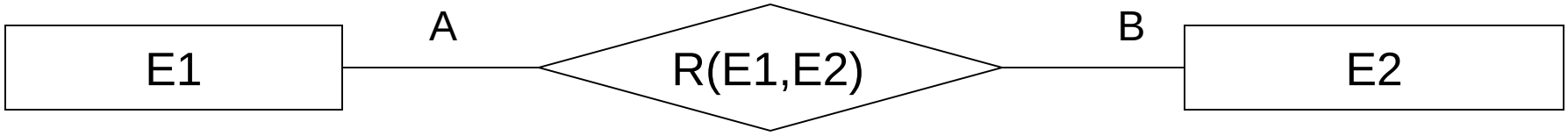
- Primer
  - Tip poveznika

    RADI-U(RADNIK,SEKTOR) ima odnos kardinaliteta N:1

    što znači da:
    - jedan radnik može biti u vezi (u ulozi zaposlenog) samo sa jednim sektorom,
    - ali svaki sektor (u ulozi poslodavca) može biti u vezi sa proizvoljnim brojem radnika (u ulozi zaposlenih)

Predstavljanje kardinaliteta tipa poveznika u ER dijagramima

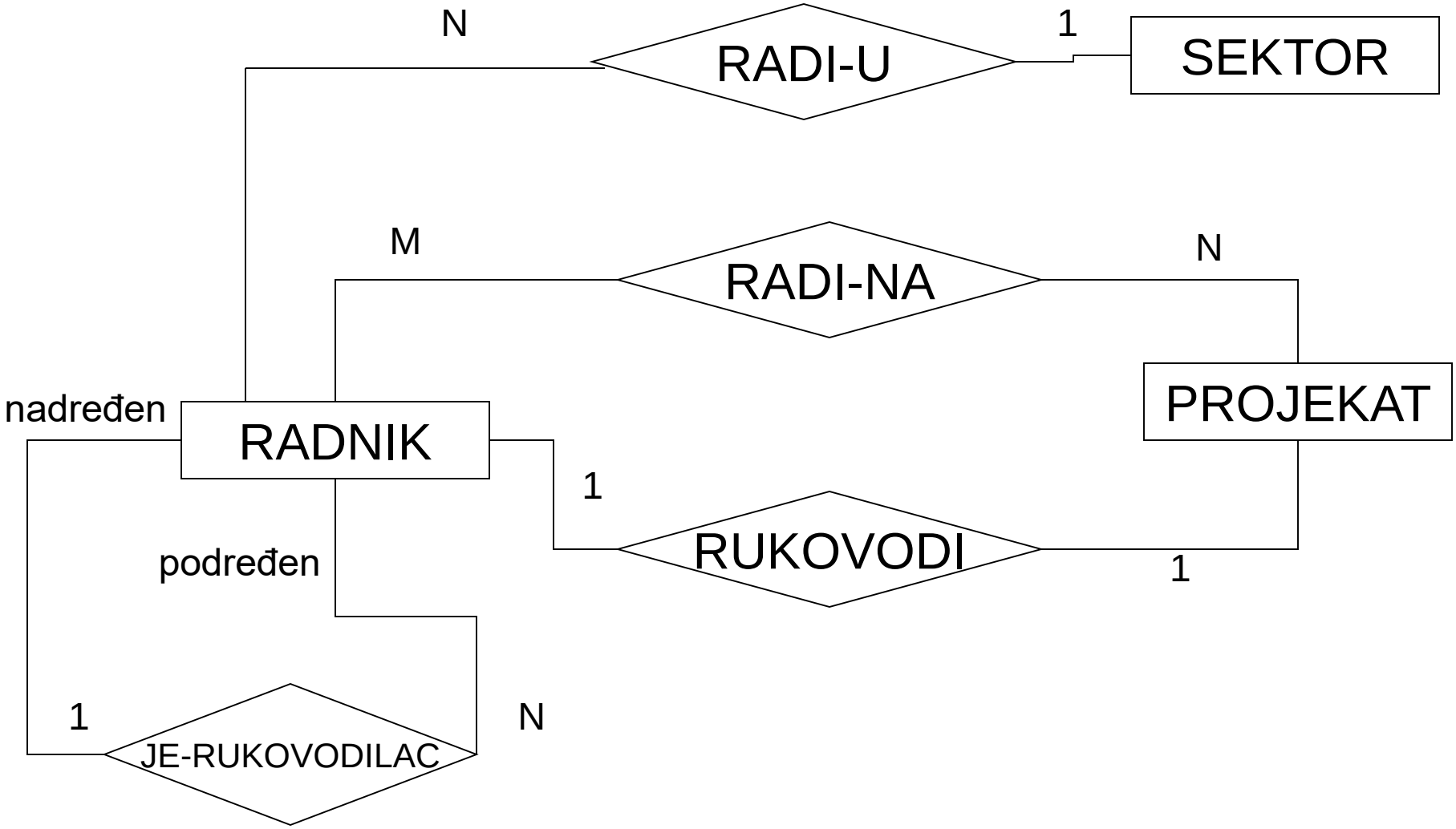
- Označavanje kardinaliteta **A:B** tipa poveznika **R(E1,E2),**
- A se navodi uz tip poveznika E1, a B uz tip poveznika E2 (ili obrnuto u nekim notacijama)



Primer predstavljanja kardinaliteta tipa poveznika u ER dijagramima

- Niže su prikazana ova dva načina označavanja kardinaliteta za tip poveznika RASPOREĐEN(RADNIK,RADNO_MESTO) za realni sistem u kojem:
  - **Radnik** može biti raspoređen samo na jedno radno mesto, a
  - Na jedno **radno mesto** može biti raspoređeno više radnika

ER dijagram sa označenim kardinalitetima tipa poveznika



Ograničenje participacije

- Ograničenje participacije specificira **minimalni broj** instanci poveznika u kojima entitet može participirati
  - Ponekad se naziva **minimalno ograničenje kardinaliteta**

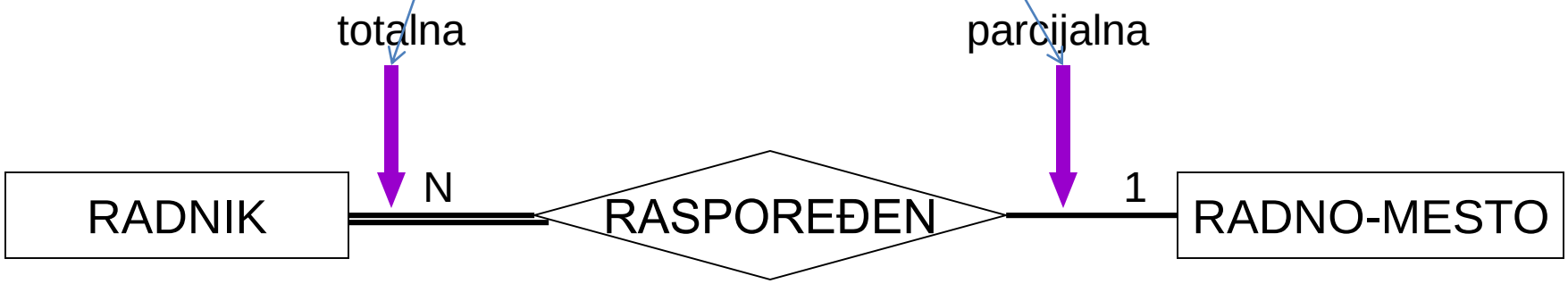
- Postoje dva tipa ograničenja participacije
  - **Totalna (egzistencijalna)** – svaki entitet mora participirati u nekoj instanci tipa poveznika
  - **Parcijalna** – neki entiteti mogu participirati u instancama ipa poveznika

- Ograničenje participacije specificira da li postojanje (egzistencija) entiteta zavisi od toga da li je u vezi sa drugim entitetom preko tipa poveznika

Predstavljanje participacije tipa poveznika u ER dijagramima

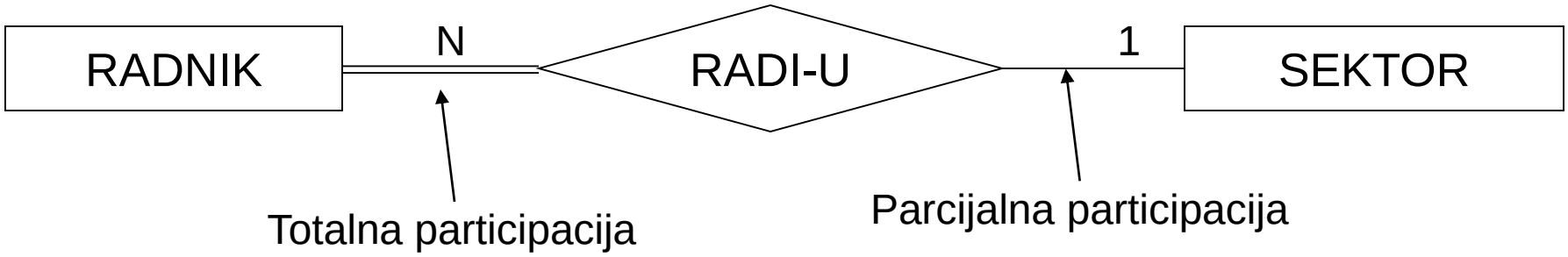
- Za označavanje participacije tipa poveznika R(E1,E2) koristićemo dva različita tipa linija koje spajaju tip entiteta sa tipom poveznika
  - **Jednostruku liniju** za parcijalnu participaciju i
  - **Dvostruku liniju** za totalnu participaciju

- Na primeru su prikazana dva načina za realni sistem u kojem:
  - **Radnik mora biti raspoređen** na tačno jedno radno mesto, a
  - Na jedno radno mesto **može biti raspoređeno** više radnika, ali mogu postojati radna mesta na koja niko nije raspoređen



Primer ograničenja participacije

- U tipu poveznika RADI-U(RADNIK,SEKTOR)
  - tip entiteta **RADNIK ima totalnu participaciju** ako u preduzeću vlada pravilo da svaki radnik mora biti zaposlen u nekom sektoru
  - Tip entiteta **SEKTOR ima parcijalnu participaciju** ako u preduzeću vlada pravilo da mogu postojati sektori bez zaposlenih



Strukturalna ograničenja tipa poveznika ili (min,max) ograničenja

- Posmatra se binarna relacija R između skupova pojava dva tipa entiteta E1 i E2
- Ova relacija se može predstaviti putem dva preslikavanja

  R1: E1 $\rightarrow \mathcal{P}$(E2)

  R2: E2 $\rightarrow \mathcal{P}$(E1)

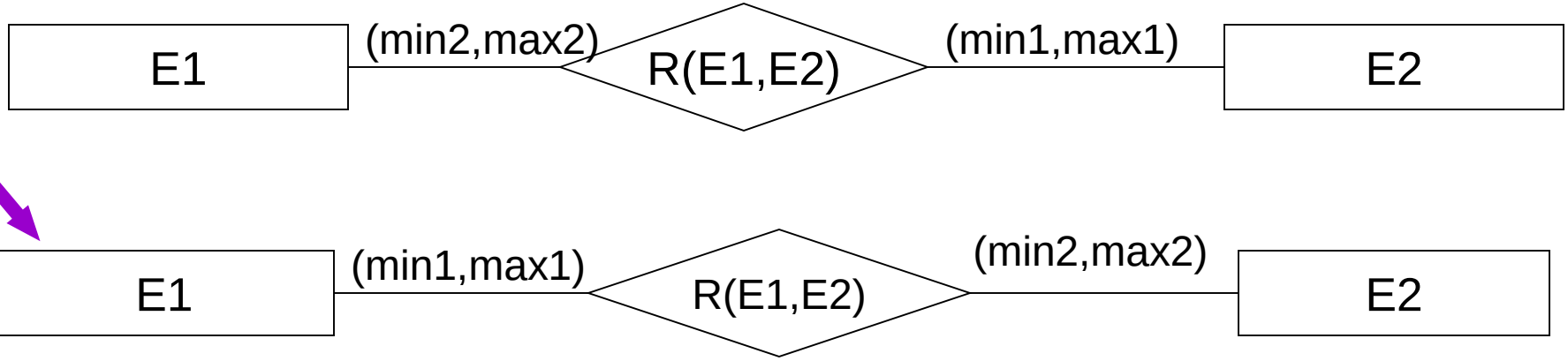
  gde je $\mathcal{P}$(E) partitivni skup skupa E
- Za svako od ovih preslikavanja se definiše minimalni i maksimalni kardinalitet preslikavanja
- Preslikavanjima R1 i R2 se može dati sledeća semantička interpretacija:
  - R1 je uloga entiteta iz skupa E1, a R2 uloga entiteta iz skupa E2 u njhovoj vezi opisanoj relacijom R(E1,E2)

(min,max) ograničenja

- Minimalna vrednost definiše participaciju entiteta u tipu poveznika R(E1,E2)
  - Ako je min=1 preslikavanje je totalno
  - Ako je min=0 preslikavanje je parcijalno
- Maksimalne vrednosti definišu odnos kardinaliteta tipa poveznika R(E1,E2)
  - max1:max2
  - 1:1
  - 1:N
  - N:M

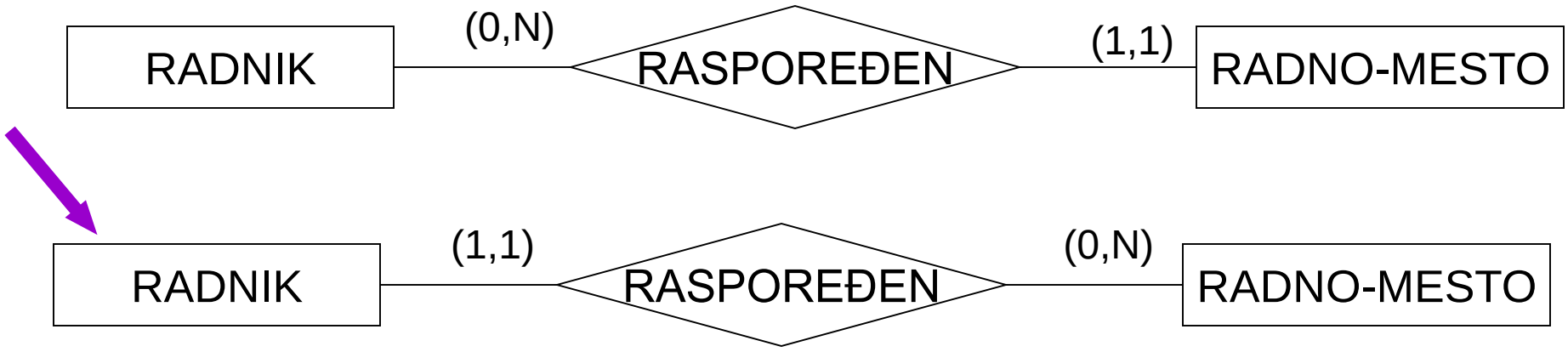
Predstavljanje (min,max) ograničenja tipa poveznika u ER dijagramima

- Za označavanje (min,max) ograničenja tipa poveznika R(E1,E2) postoje dva načina:
  - Brojnost preslikavanja se upisuje uz tip entiteta u koji se on preslikava. Par (min1,max1) se navodi na potegu uz tip poveznika E2, a par (min2,max2) na potegu uz tip poveznika E1 ili
  - Brojnost preslikavanja se upisuje uz sam tip entiteta. Par (min1,max1) se navodi na potegu uz tip poveznika E1, a par (min2,max2) na potegu uz tip poveznika E2



Primer predstavljanje (min,max) ograničenja tipa poveznika u ER dijagramima

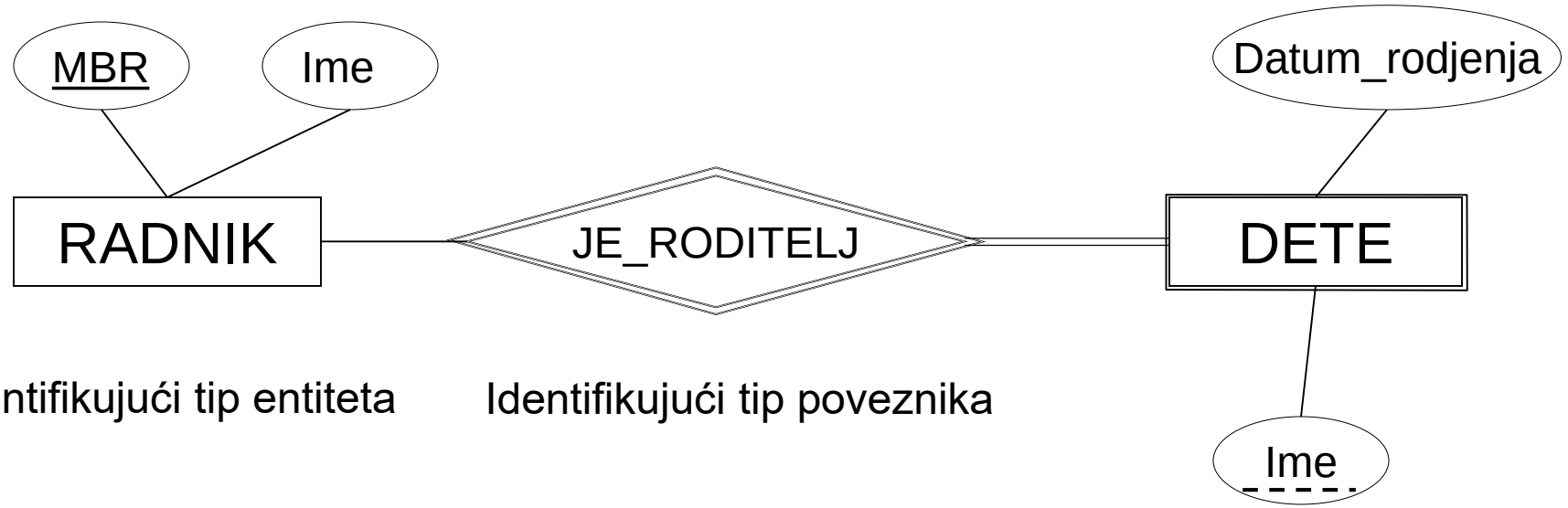
- Na primeru niže prikazana su ova dva načina za isti realni sistem u kojem:
  - Radnik može biti raspoređen samo na jedno radno mesto, a
  - Na jedno radno mesto može biti raspoređeno više radnika, a može radno mesto biti slobodno



Slabi tip entiteta

- Tip etiteta može biti
  - Regularni
  - Slabi
- Tip entiteta koji nema sopstvene ključne atribute je slabi tip entiteta
- Entiteti slabog tipa entiteta se identifikuju preko veze sa nekim drugim tipom entiteta (regularnim ili slabim) u kombinaciji sa nekim svojim atributom
  - Taj drugi tip entiteta je identifikujući (vlasnički, roditeljski ili dominantni) tip entiteta
  - Poveznik koji povezuje slabi tip entiteta sa vlasnikom se naziva identifikujući tip poveznika
  - Slabi tip entiteta uvek totalno participira u identifikujućoj vezi
- Slabi tip entiteta ima **parcijalni ključ** (diskriminator) koji predstavlja podskup skupa atributa slabog tipa entiteta koji na jedinstven način identifikuje slabe entitete koji su u vezi sa istim vlasničkim tipom entiteta

ER dijagram: slabi tip entiteta



Identifikujući tip entiteta

Identifikujući tip poveznika

Parcijalni ključ

Integritetna komponenta ER modela podataka

- ER model ima dobro definisana ograničenja koja se uglavnom eksplicitno definišu
- To su:
  - Integritet domena
  - Kardinalnost tipa poveznika
  - Participacija tipa poveznika
  - Slabi tip entiteta
  - Ključ tipa entiteta

Integritet (ograničenje) domena

- Ovaj tip ograničenja imaju gotovo svi modeli podataka
- Opšti oblik ograničenja domena:
  (tip podatka, dužina podatka, uslov)
- Tip podatka
  - Definiše vrstu znakova putem kojih se izražava vrednost stributa
- Dužina podatka
  - Definiše maksimalni broj znakova koji mogu biti upotrebljeni za izražavanje vrednosti atributa
- Tip podatka i dužina podatka su standardna ograničenja domena
- Često nisu dovoljno precizna, pa se dodaje treća komponenta uslov koja preciznije definiše ograničenje domena

Standardna ograničenja domena

- To su tipovi podataka sa dužinama tih podataka:
  - INTEGER(broj cifara)
  - REAL(broj cifara ispred zapete, broj cifara iza zapete
  - CHARACTER(broj znakova)
  - DATE
  - LOGICAL
  - ...
- Primer:
  - Ako je dom(OCENA) pridruženo standardno ograničenje INTEGER(2), tada važi dom(OCENA)={0,1,2,...,99}
  - Ako je dom(NAZIV_PREDMETA) pridruženo standardno ograničenje CHARACTER(15), tada važi da naziv predmeta može biti bilo koji niz dužine 15 znakova

Null vrednost kao specijalni tip ograničenja domena

- Ovim ograničenjem se specificira da li atribut može imati nedefinisanu vrednost
- Ima dva značenja:
  - **Postojeća, ali nepoznata vrednost**
  - **Neprimereno svojstvo**

Pregled metodologije konceptualnog projektovanja korišćenjem ER/EER modela podataka

1. Identifikovati tipove entiteta
2. Identifikovati tipove poveznika
3. Identifikovati i pridružiti atribute tipovima entiteta i poveznika
4. Utvrditi domene atributa
5. Utvrditi atribute koji čine kandidate za ključeve, primarni ključ i alternativne ključeve
6. Razmotriti korišćenje naprednih koncepata modeliranja - EER (opciono)
7. Proveriti model podataka na redundancu
   - Radi provere da li ima redundance u modelu
8. Validirati konceptualni model u odnosu na transakcije korisnika
   - Radi provere da projektovani model podržava zahteve transakcija korisnika
9. Pregledati konceptualni model podataka sa korisnikom
   - Radi provere da su zahtevi korisnika podržani

Projektovanje baze podataka

Korak 1: Izgraditi konceptualni model podataka

- **Korak 1.1** Identifikovati tipove entiteta
- **Korak 1.2** Identifikovati tipove poveznika
- **Korak 1.3** Identifikovati i pridružiti atribute tipovima entiteta i poveznika
- **Korak 1.4** Utvrditi domene atributa
- **Korak 1.5** Utvrditi atribute koji čine kandidate za ključeve, primarni ključ i alternativne ključeve
- **Korak 1.6** Razmotriti korišćenje naprednih koncepata modeliranja (opciono)
- **Korak 1.7** Proveriti model podataka na redundancu
- **Korak 1.8** Validirati konceptualni model u odnosu na transakcije korisnika - radi provere da projektovani model podržava zahteve transakcija korisnika
- **Korak 1.9** Pregledati konceptualni model podataka sa korisnikom - radi provere da su zahtevi korisnika podržani

Projektovanje baze podataka

Korak 2: Izgraditi i proveriti logički model podataka

Korak 2.1 Pronaći relacije za logički model podataka
- Preslikati konceptualni u relacioni model podataka koristeći poznati algoritam

Korak 2.2 Validirati relacije korišćenjem normalizacije
- Prevesti šeme relacija u što je moguće višu normalnu formu

Korak 2.3 Validirati relacije u odnosu na transakcije korisnika

Korak 2.4 Proveriti ograničenja integriteta

Korak 2.5 Pregledati logički model podataka sa korisnikom

Korak 2.6 Integrisati logičke modele podataka u globalni logički model podataka *(opciono)*

Korak 2.7 Proveriti model podataka na budući rast

Projektovanje baze podataka

Korak 3: Prevesti logički model podataka na ciljni DBMS

- **Korak 3.1** Projektovati bazne relacije
- **Korak 3.2** Projektovati reprezentaciju izvedenih podataka
- **Korak 3.3** Projektovati opšta ograničenja

Korak 4: Projektovati organizaciju fajlova i indekse

- **Korak 4.1** Analizirati transakcije
- **Korak 4.2** Izabrati organizacije fajlova
- **Korak 4.3** Izabrati indekse
- **Korak 4.4** Proceniti potreban prostor na diskovima

Korak 5: Projektovati poglede korisnika

Korak 6: Projektovati bezbedonosne mehanizme

Korak 7: Razmotriti uvođenje kontrolisane redundance

Korak 8: Pratititi (*monitoring*) sistem koji je pušten u eksploataciju i vršiti njegovo podešavanje (tuning)