

Upravljanje informacijama i
znanjem:

Lekcija 5: Modelovanje podataka

zima 2019/2020

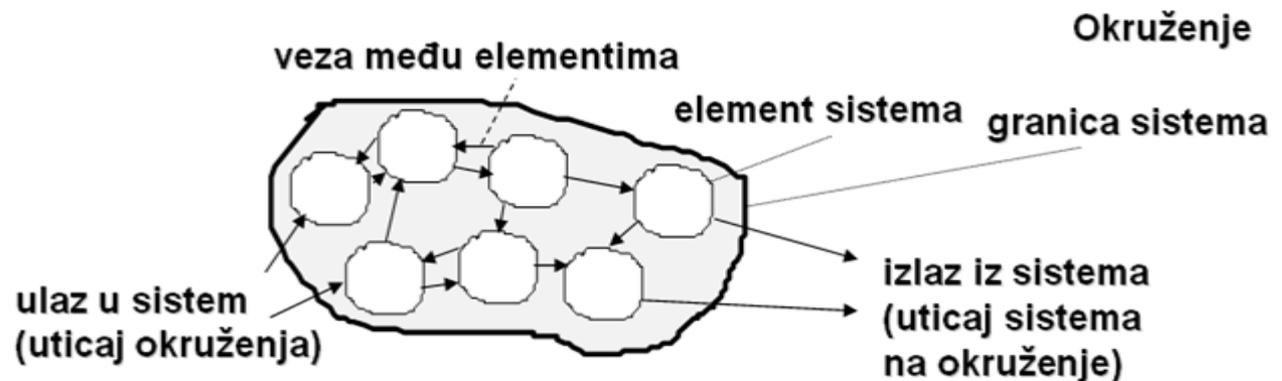
Prof. dr Branimir M. Trenkić

Model podataka – osnovni pojmovi

- **Podsetimo se** definicije sistema:

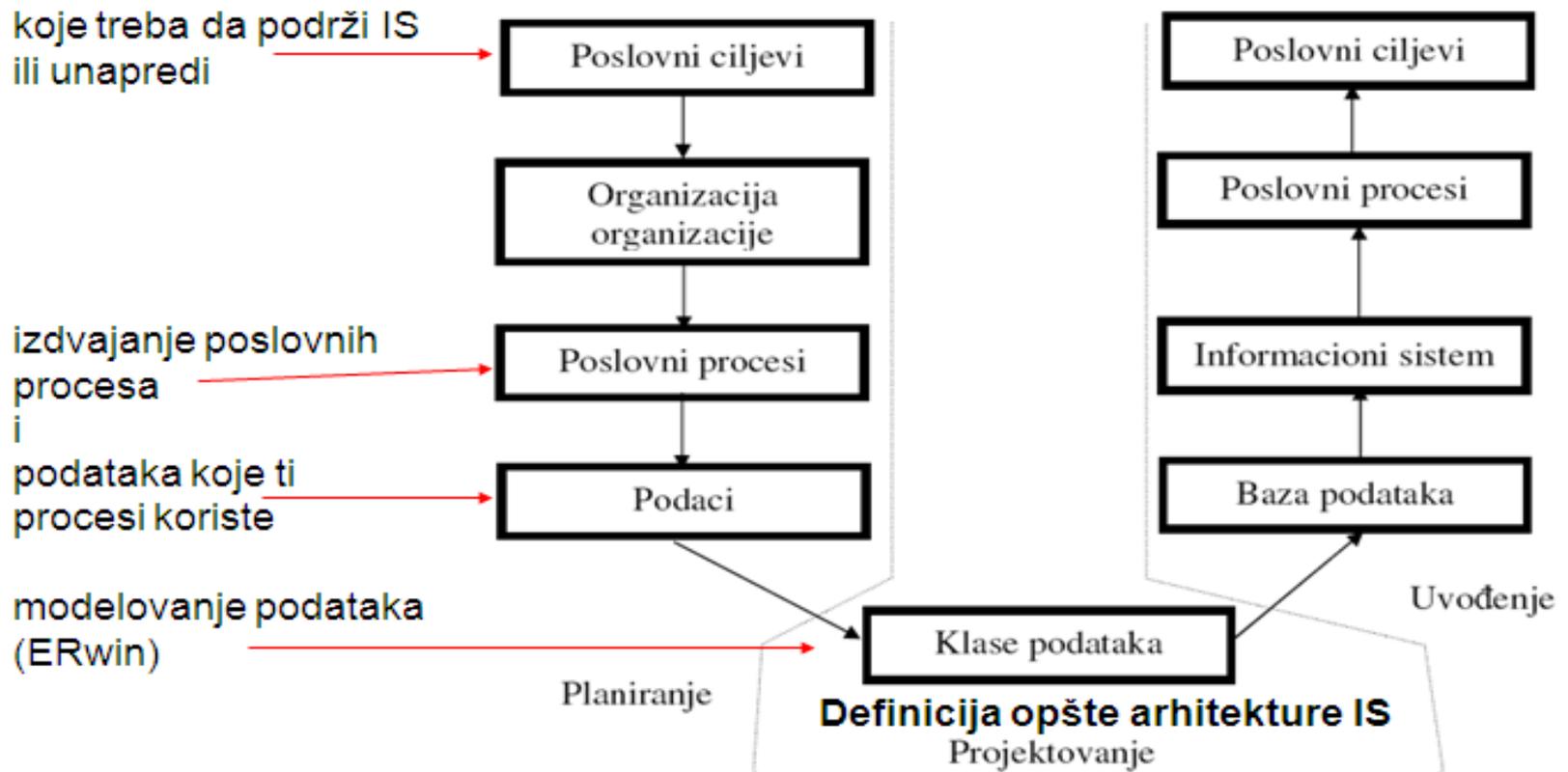
Definicije sistema:

- Izdvojena funkcionalna celina koja je sastavljena od skupa objekata, njihovih utvrđenih svojstava (atributa) i skupa relacija koje povezuju objekte, kao i od svojstava relacija.



Model podataka – osnovni pojmovi

- **Podsetimo se** planiranje i uvođenje IS primenom BSP metode:





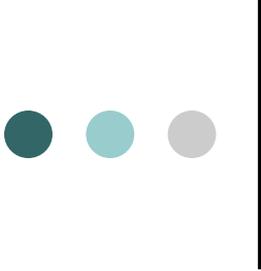
Model podataka – osnovni pojmovi

- **Podatak** je *kodirana činjenica* iz realnog sistema, on je **nosilac informacije**
- **Informacija** je *protumačeni* (***interpretirani***) **podatak**
- ***Interpretacija podataka*** se vrši na osnovu **strukture podataka**, **semantičkih ograničenja** na njihove vrednosti i preko **operacija** koje se nad njima mogu izvršiti



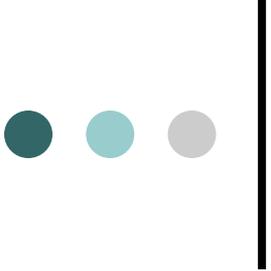
Model podataka – osnovni pojmovi

Model podataka opisuje strukturu nekog sistema (skup objekata, njihovih atributa i njihovih međusobnih veza) i njegovu dinamiku (skup operacija)



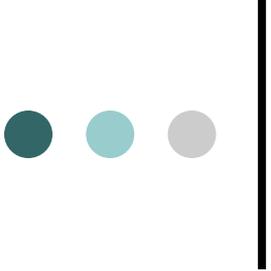
Model podataka – osnovni pojmovi

- Model podataka je osnov za dizajn baze podataka
- Model podataka
 - **Sredstvo** za prikaz objekata sistema, njihovih **atributa** i međusobnih **veza**
 - Prikazuje stanje sistema u jednom trenutku vremena
- Baza podataka
 - Kolekcija međusobno povezanih podataka koja **modeluje objekte**, **veze objekata** i **attribute** objekata posmatranog sistema



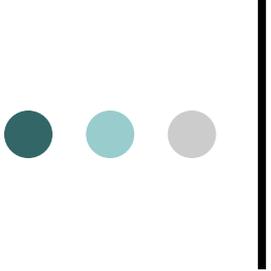
IS – Upravljanje podacima

- Jedna od osnovnih **karakteistika IS** je ***mogućnost:***
 1. **Uskladištenja** podataka i informacija
 2. **Pristupa** podacima
 3. **Obrade** podataka – ažuriranje
- Sve mogućnosti zajedno – **upravljanje podacima**
- ***Računari*** se koriste za skladištenje i obradu podataka još od 1950-tih godina



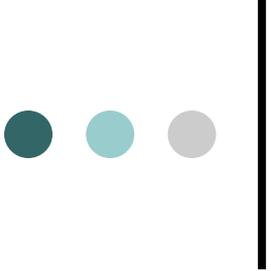
IS – Upravljanje podacima

- Primarna (glavna) **memorija** računara omogućava privremeno skladištenje podataka
- Sekundarna memorija - tip memorije za trajno skladištenje podataka
 - Magnetni i optički mediji
 - magnetna traka,
 - hard diskovi,
 - CD-ovi, DVD



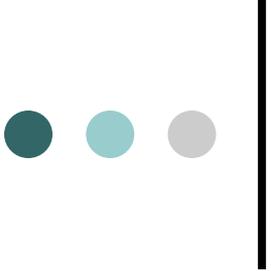
IS – Upravljanje podacima

- Od nastanka IS - *prvi zadatak* dizajnera – softver koji bi omogućio
 1. *Lociranje* određenih podataka uskladištenih u sekundarnoj memoriji i njihovo
 2. Efikasno *učitavanje* u glavnu memoriju kako bi se mogli obrađivati
 3. Nakon obrade vrši se njegovo ponovno *uskladištenje*



IS – Upravljanje podacima

- *Osnovnu strukturu jednog IS-a* čine skupovi **datoteka**
- **Datoteke** predstavljaju *skup podataka označen jedinstvenim imenom* koji je *permanently uskladištene* u nekom sekundarnom uređaju za skladištenje podataka
- Osnovni *elementi* svake ***datoteke*** – **zapisi**



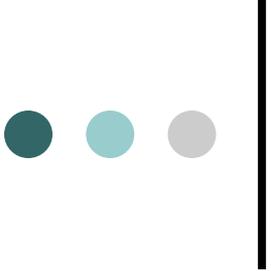
IS – Upravljanje podacima

- Osnovni *elementi* svake *datoteke* – zapisi
- Svaki zapis opisuje neku stvar, pojam (ili jedinicu)
- **Sastoji se od** određenog broja polja fiksnih dužina
- Svako **polje** daje **vrednost** određenoj **osobini** date **jedinice**



Tehnologije sistema za čuvanje i obradu podataka

- **Ključni događaj** – *pojava baze podataka* (krajem šezdesetih godina)
- **Do tada** – tradicionalni pristup (u suštini *ad hoc* pristup) koji se bazira na *datotekama*
 - Polazeći od definicije aplikacije:
 - *Kreirala bi se datoteka* (ili skup datoteka) *podataka potrebnih za* definisanu *aplikaciju*
 - *Pisali bi se programi* koji obrađuju podatke tih datoteka

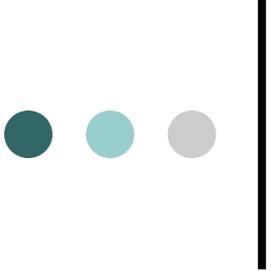


Upravljanje zasnovana na datotekama

- Struktura datoteke je *prilagođena aplikaciji* koja te podatke koristi
- ***Sistem baziran na datotekama*** predstavlja **skup aplikacija/programa**, koji obavljaju neke servise za krajnjeg korisnika – ***oslanjajući se na datotekama***
- **Svaki program definiše i upravlja "svojim" podacima!**

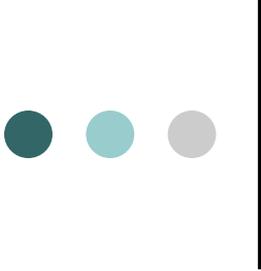
Upravljanje zasnovana na datotekama





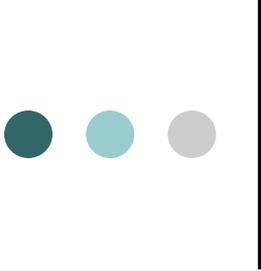
Upravljanje zasnovana na datotekama - Osobine

- **Razdvojenost** i **izolovanost** podataka
- **Ponavljjanje** istih podataka (redundansa) uz različite aplikacije
- **Nekonzistentnost podataka** – promena vrednosti koja se ne izvrši dosledno
- **Zavisnost programa** od organizacije datoteke podataka
 - Sekvencijalna
 - Direktna
 - Indeksirana...



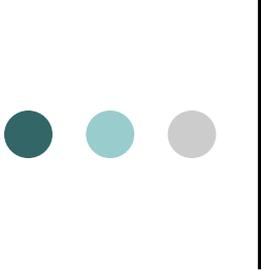
Upravljanje zasnovana na datotekama

- **Posledice:**
- **Obrada** podataka **je skupa**
- **Korišćenje istih podataka** od strane većeg broja korisnika je **otežano**
- **Neadekvatna realizacija oporavka** od pada sistema
- **Niska produktivnost** razvoja/održavanja IS



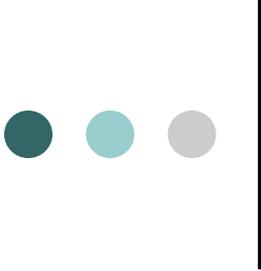
Upravljanje zasnovana na datotekama

- Utvrđeno je da su ograničenja/nedostaci sistema organizovanih na datotekama ***posledica***:
 1. **Definicija podataka** je ugnježdjena **u aplikacionom programu**
 2. **Ne postoji kontrola** nad pristupom i manipulacijom podacima, osim one koju daje aplikacioni program
- **Novi pristup**



Upravljanje zasnovana na datotekama

- **Novi pristup** – treba da obezbedi:
- **Nezavisnost** struktura podataka od programa koji ih obrađuju i programa od strukture podataka
- **Minimalnost ponavljanja** podataka
- Obrada podataka nije vezana za programski jezik opšte namene – ***“upitni” jezik***
- Korišćenje skupa podataka od strane ***većeg broja korisnika***

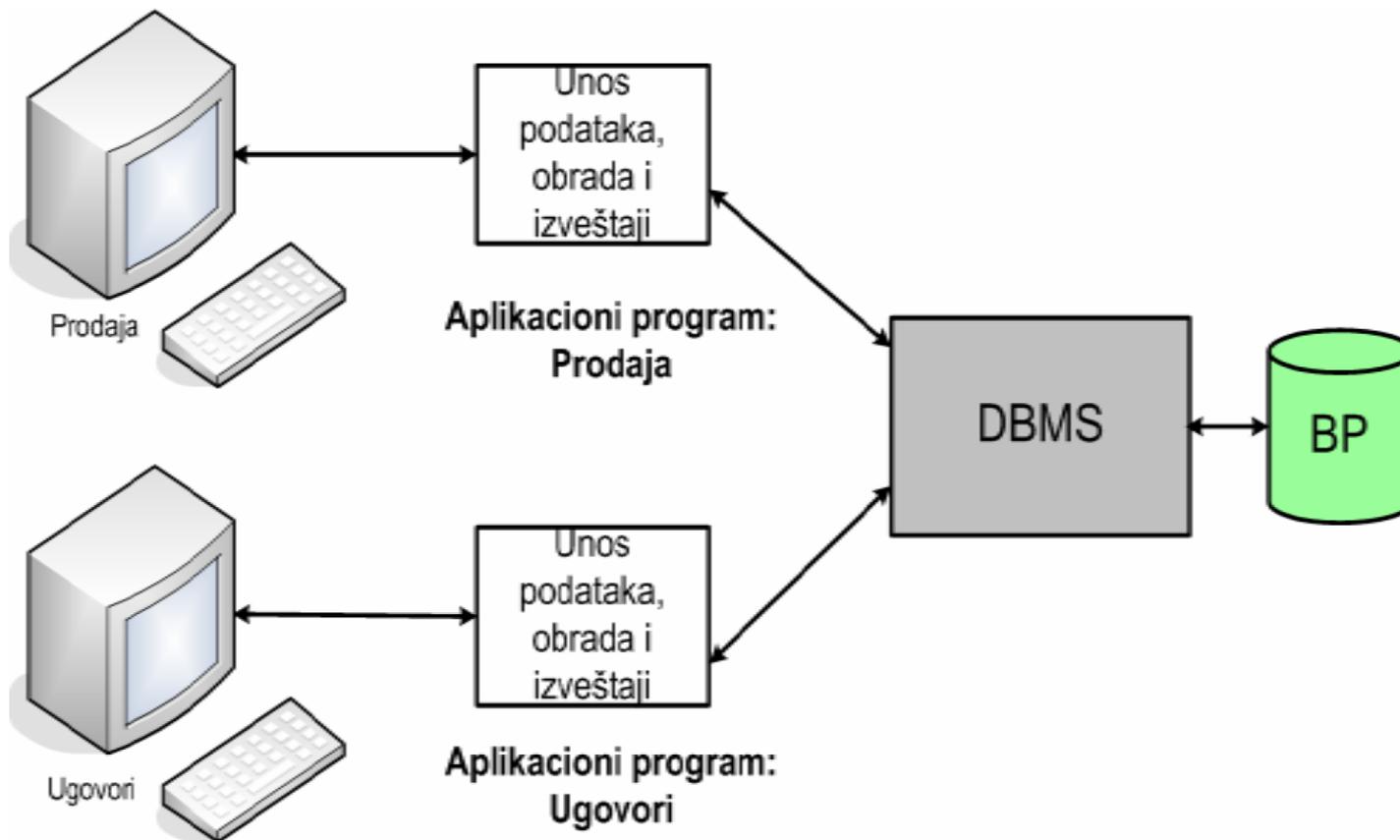


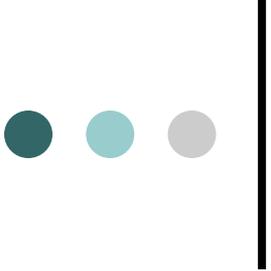
Upravljanje zasnovana na korišćenju BP

- Novi pristup baziran na:
 1. **BP**, *bazama podataka*
 2. **DBMS**, *sistemima za upravljanje BP*
(**SUBP**)
- Ovi koncepti su se pokazali **vrlo uspešnim**
- Poništava individualne datoteke
- Prevazilaze se navedena ograničenja
- Koncept BP i SUBP traje i danas

Upravljanje zasnovana na korišćenju BP

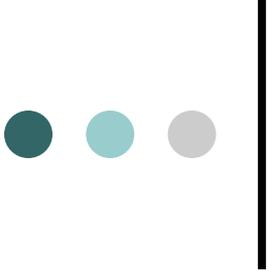
Nema pristupa BP bez prolaska kroz DBMS!





Upravljanje zasnovana na korišćenju BP

- **BP**: zajednički korišćena kolekcija logički povezanih *podataka* i *opisa tih podataka*, projektovana da zadovolji informacione potrebe preduzeća
- **SUBP** je *s/w sistem* koji omogućava korisnicima da definišu, kreiraju, održavaju i kontrolišu *pristup BP na efikasan način*
- **SUBP** je s/w koji interaguje i sa aplikacionim programom korisnika i sa BP



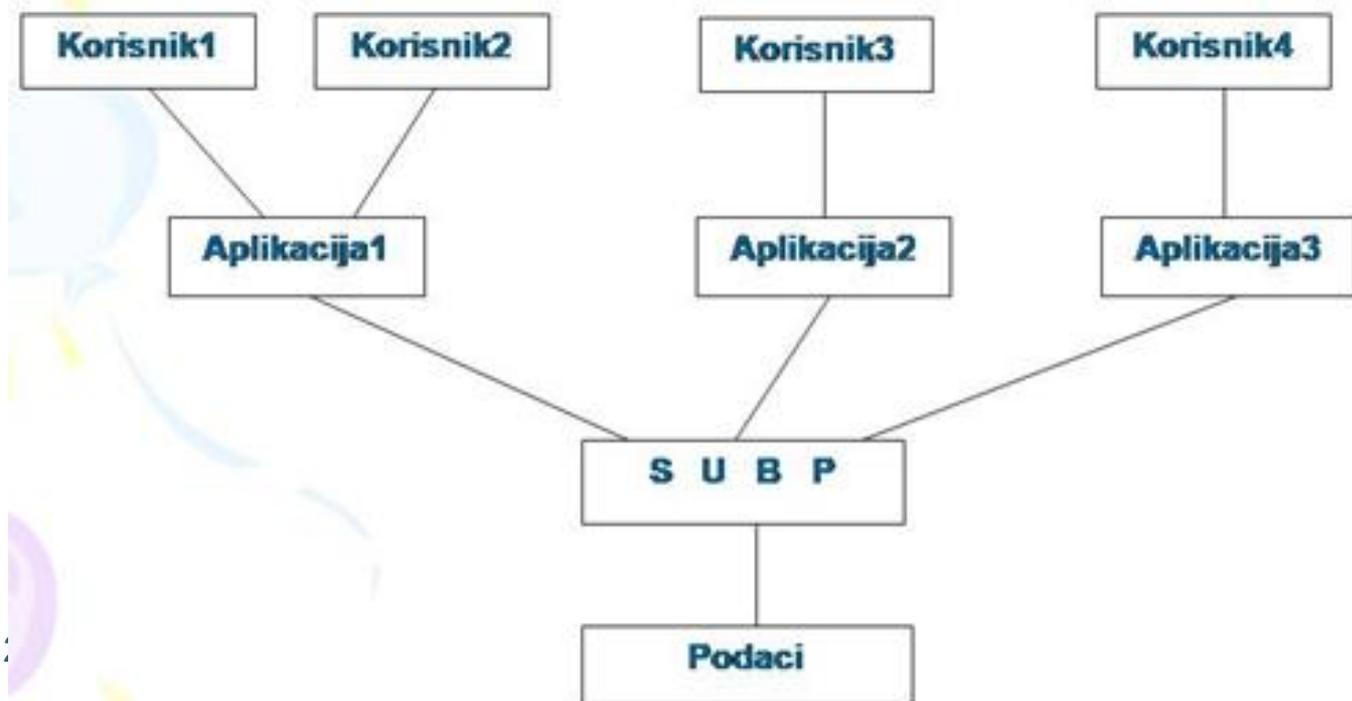
Upravljanje zasnovana na korišćenju BP

- Novine pristupa:
- **Definicija podataka** je razdvojena od aplikacionog programa i sama se čuva u BP (*katalog/rečnik*)
- Fundamentalna karakteristika pristupa – **obezbeđivanje određenog nivoa apstrakcije podataka**, skrivanjem nekih detalja skladištenih podataka koji nisu potrebni većini korisnika

Upravljanje zasnovana na korišćenju BP

SUBP – Blok šema:

Kontrolni sloj između različitih **korisnika aplikacija** i **samih podataka**

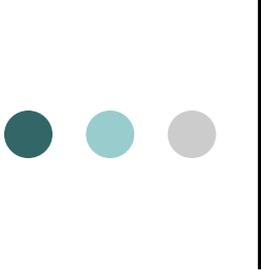




Upravljanje zasnovana na korišćenju BP

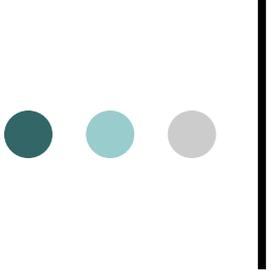
SUBP – *Pogled*:

- Svakoj aplikaciji će biti potreban samo određeni specifični podskup podataka koji se nalazi u bazi podataka
- **Pogled** - specifični podskup podataka iz baze podataka koji koristi određena aplikacija
- SUBP kontroliše pristup i omogućava ga koristeći usluge operativnog sistema



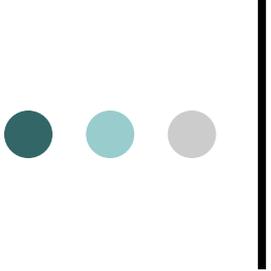
ANSI/SPARC arhitektura

- **1975.** g. **ANSI/SPARC** komitet je prepoznao potrebu za *tro-nivovskim modelom arhitekture* SUBP-a
- **Cilj** ovog modela – da se **odvoji pogled** korisnika na BP **od načina** na koji je BP **fizički predstavljana**
- Nikada **nije** formalno **postao standard**
- Model pruža osnovu za razumevanje nekih od funkcionalnosti SUBP-a



ANSI/SPARC arhitektura

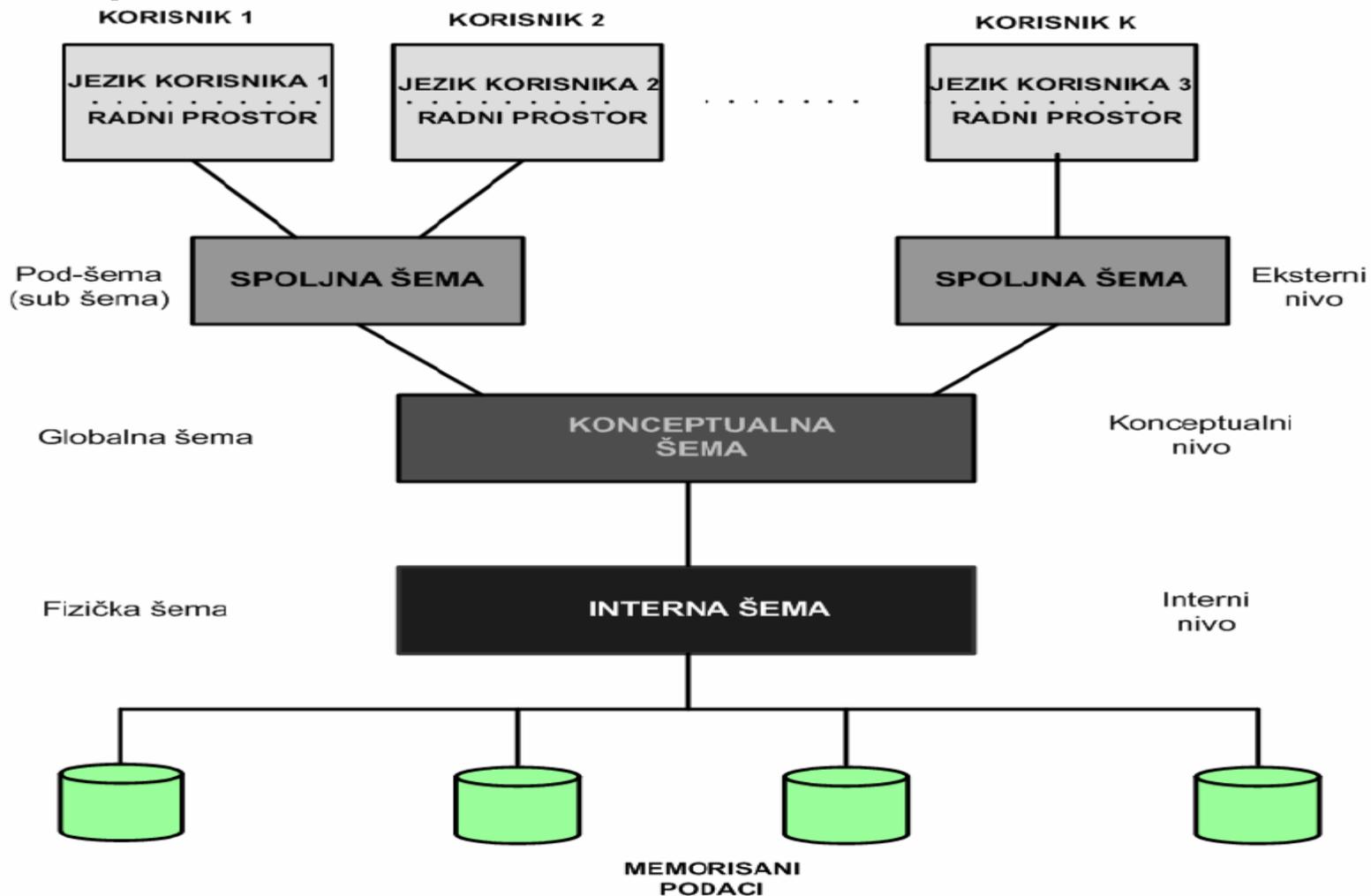
- **Arhitektura** - predstavljena hijerarhijom apstrakcija
- **3-nivovska hijerarhija**
 - Specifičan način predstavljanja objekata
 - Odnose među objektima
 - Operacije nad objektima
- Hijerarhijska struktura omogućuje
 - Prirodnu **dekompoziciju** i
 - Efikasan **razvoj** sistema upravljanja

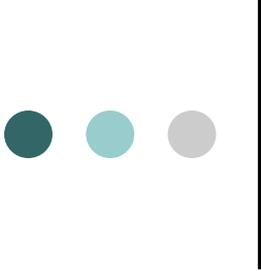


ANSI/SPARC arhitektura

- ***Tro-nivovska arhitektura*** se sastoji od:
 - ***Eksternog nivoa***
 - ***Konceptualnog nivoa***
 - ***Internog nivoa***

ANSI/SPARC arhitektura

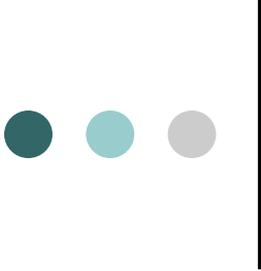




ANSI/SPARC arhitektura – Eksterni nivo

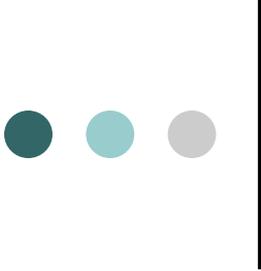
- **Koncept pogleda:**

- Korisnici mogu imati **sopstvene, specifične, “pogleda”** na bazu podataka (iz razloga **udobnosti** ili **bezbednosti**)
- Opisuje samo onaj **deo baze podataka** koji je relevantan za **pojedinih korisnika**
- Eksterni nivo – **više** različitih eksternih **pogleda** na bazu podataka



ANSI/SPARC arhitektura – Eksterni nivo

- Na eksternom nivou – **više različitih eksternih šema** – nazivaju se i **pod-šeme** – koje odgovaraju različitim **pogledima (view)** korisnika



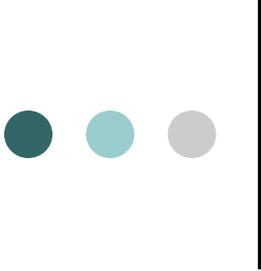
ANSI/SPARC arhitektura – Eksterni nivo

- Na primer, jedan korisnik može videti datume u formi (**dan, mesec, godina**), dok drugi može videti datume u formi (**godina, mesec, dan**)
- Neki pogledi mogu podrazumevati i određena proračunavanja
 - (**DOB -> Starost**)



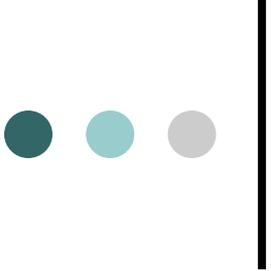
ANSI/SPARC arhitektura – Konceptualni nivo

- **Konceptualna šema**: način na koji se podaci sa fizičkog nivoa baze predstavljaju ***korisniku u opštem slučaju***
- Konceptualni nivo omogućuje (1) **preslikavanje** i (2) željenu **nezavisnost** između eksternog i internog nivoa
- Mogu se “**videti**” **svi podaci** iz fizičke baze podataka – reprezentacija **pogodnija za korisnika** od fizičke (na višem je nivou apstrakcije)



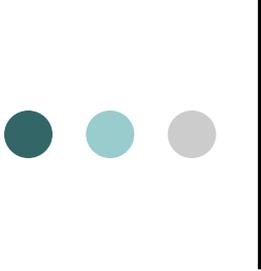
ANSI/SPARC arhitektura – Konceptualni nivo

- Reprezentacija podataka na konceptualnom nivou – ***implementacioni model podataka***
 - Logička struktura svih podataka u bazi
 - Skup operacija koje korisnik može izvršiti nad tim podacima
- tj., opisuje **sve** entitete, attribute, njihove relacije, ograničenja, informacije o sigurnosti i integritetu



ANSI/SPARC arhitektura – Konceptualni nivo

- Konceptualni nivo ***podržava sve eksterne poglede***, tako što će svaki podatak koji je raspoloživ za korisnike – mora biti **sadržan** u ili **izvodljiv** iz konceptualnog nivoa



ANSI/SPARC arhitektura – Interni nivo

- **Fizička šema**: opisuje kako su podaci memorisani u BP
 - ***Fizička struktura podataka*** - datoteke sa sekvencijalnim, indeksiranim ili direktnim pristupom
 - ***Procedure za fizičku realizaciju operacija*** koje zadaje korisnik na višem nivou

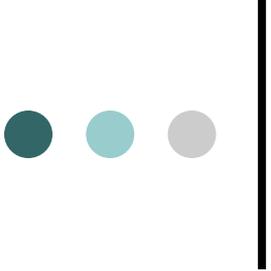
ANSI/SPARC arhitektura

Zaključak:

- Postoji samo jedan model podataka u sistemu koji se odnosi na celokupnost baze podataka, i veći broj eksternih pogleda, od kojih se svaki sastoji od *apstraktne slike dela baze podataka*

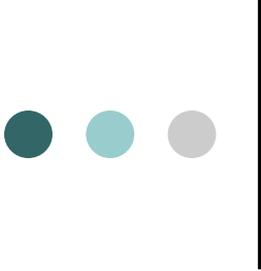
Primarni cilj modela podataka:

- Uspostavljanje formalnog sistema za predstavljanje i manipulisanje podacima u bazi podataka



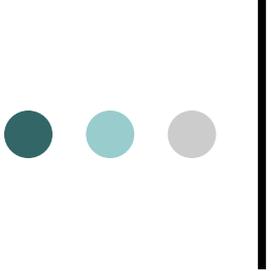
ANSI/SPARC arhitektura

- Preslikavanja:
- *Eksterno-konceptualno* preslikavanje
- *Konceptualno-interno* preslikavanje
- SUBP je odgovoran za preslikavanja između ove tri vrste šeme (modela)
- SUBP, takođe, mora da proveri šeme radi konzistentnosti (doslednosti)



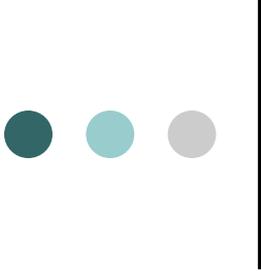
ANSI/SPARC arhitektura

- **Preslikavanja:**
- ***Svaka eksterna šema je izvedena iz *konceptualne šeme* i ona mora da koristi informacije unutar konceptualne šeme u cilju preslikavanja između svake eksterne šeme i interne šeme***



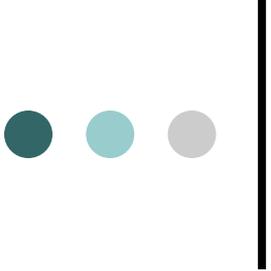
ANSI/SPARC arhitektura - ciljevi

- Svaki korisnik bi trebalo da bude u mogućnosti da **pristupi istim podacima**, ali da imaju **drugačije prilagođeni prikaz** tih podataka
- Korisnik **ne bi trebalo** da se direktno **bavi detaljima** skladištenja fizičke baze podataka
- DBA treba da bude u stanju da **promeni strukturu baze** podataka za skladištenje **bez uticaja** na **korisnikov pregled**



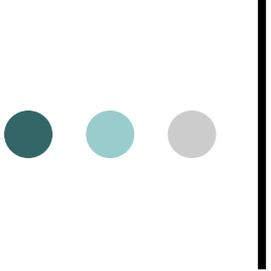
ANSI/SPARC arhitektura

- **Nezavisnost podataka:**
- **Glavni cilj** tro-nivovske arhitekture je da obezbedi **nezavisnost podataka**
- Što znači da **promene na nižim nivoima nemaju uticaja** na više nivoe
- Dve vrste nezavisnosti podataka:
 - **Logička** nezavisnost podataka
 - **Fizička** nezavisnost podataka



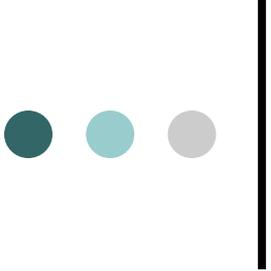
ANSI/SPARC arhitektura

- **Logička nezavisnost podataka** se odnosi na **imunitet** (zaštita) **eksternih šema na promene** u konceptualnoj šemi
- **Promene konceptualne šeme** kao što je dodavanje ili uklanjanje novih entiteta, atributa ili odnosa -
je moguće **bez promene** postojećih eksternih šema ili u aplikacionom programu



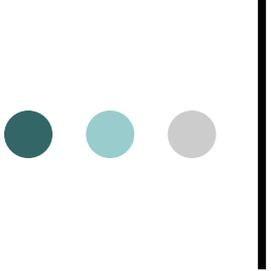
ANSI/SPARC arhitektura

- **Fizička nezavisnost podataka** se odnosi na **imunitet** konceptualne šeme na **promene** unutar interne šeme
- **Promene u internoj šemi** kao što su korišćenje različitih organizacija datoteka ili struktura skladištenja, korišćenje različitih uređaja za skladištenje, menjanje indeks ili heš algoritama bi bilo moguće **bez promene konceptualne ili eksterne šeme**



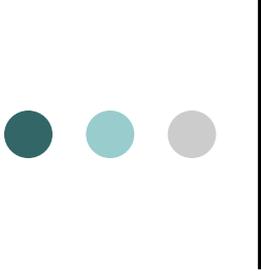
Dizajn baze podataka (BP)

- Pre kreiranja i upotrebe baze podataka potrebno ju je **dizajnirati**
- **Opis** (prikaz) strukture **baze podataka**
 - **Skup entiteta**
 - **Veze između entiteta**
- **Model podataka**



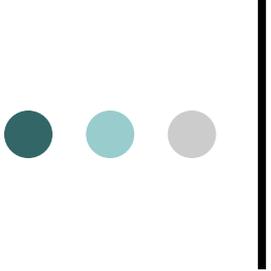
Dizajn baze podataka (BP)

- Način prikaza **interne strukture baze** podataka – **hijerarhijski**, počinje sa bitom
- **bit**(0/1) – **bajt**(karakter) – **polje**(logički povezani karakteri)
- **zapis** – logički povezana polja
- **datoteka** – logička grupa odgovarajućih polja



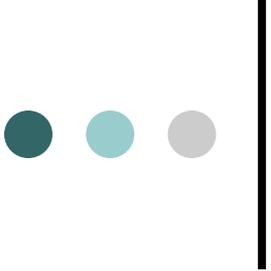
Dizajn baze podataka (BP)

- **Alternativni način** prikaza interne strukture baze podataka – bazira se na **percepciji realnog okruženja**
- **Entitet** – **objekat ili pojam** realnog sveta **od interesa** (o kome se evidentiraju podaci u IS)
- ***Atributi*** – značajnija svojstva kojima se opisuju entiteti
- ***Tip (klasa) entiteta*** – skup entiteta sa zajedničkim osobinama (definicijom)
- ***Skup entiteta*** – skup entiteta istog tipa - skup instanci istog tipa (iste klase)



Model podataka - generalno

- **Model podataka** – skup koncepata kojima se može opisati struktura baze podataka – pruža potrebna sredstva za dostizanje ciljane apstrakcije
- **Strukturu baze podataka** opisuju
 1. tipovi podataka,
 2. njihov međusoban odnos i
 3. pravila ograničenja

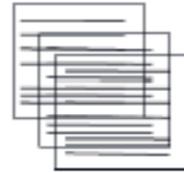
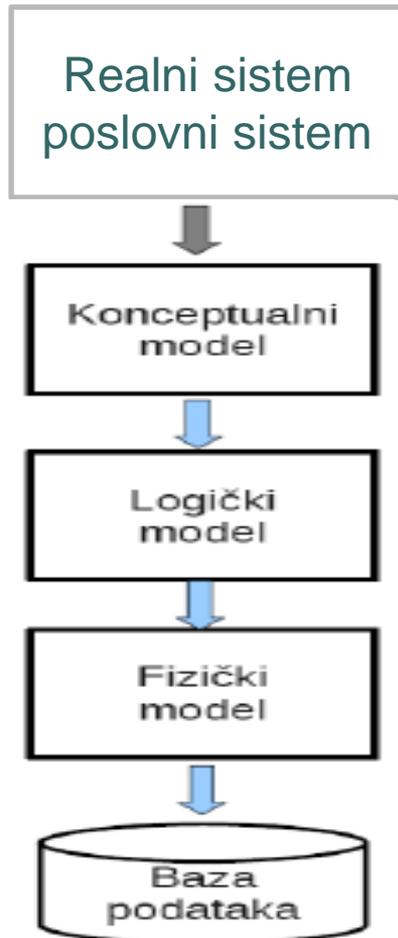


Klasifikacija modela podataka

Na osnovu:

- ***Etapa*** (koraka) ***u procesu razvoja*** baze podataka
- ***Nivoa apstrakcije*** tj. tipova koncepata koji se koriste u opisivanju strukture baze podataka

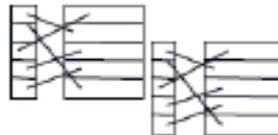
Klasifikacija modela podataka



perspektiva korisnika

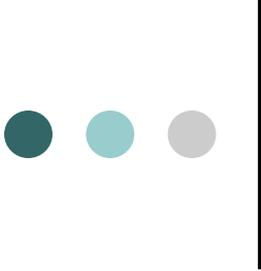


perspektiva projektanta



perspektiva izvođača

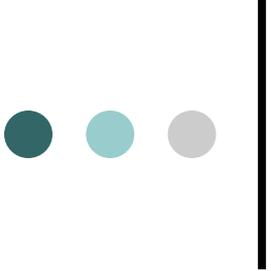




Klasifikacija modela podataka

Klasifikacija:

- **Korak 1 – Konceptualni dizajn**
 - **Konceptualni model podataka ili Model entiteta ili Model objekti/veze (MOV)**
 - Izrada modela nezavistan od izbora DBMS
- **Korak 2 - Logički dizajn**
 - **Logički (implementacioni) model podataka**
 - Kreiranje modela baze podataka u odabranom DBMS
- **Korak 3 – Fizički dizajn**



Klasifikacija modela podataka

Klasifikacija:

- **Korak 3 – Fizički dizajn**
 - **Fizički model podataka**
 - Kako je baza podataka smeštena u hardveru

Model Objekti/veze (E/R model)

- **Konceptualni model podataka ili E/R model**
 - Upotrebljava se za konceptualni dizajn
 - Obezbeđuje **koncepte** koji su bliski načinu na koji strukturu (sistema) baze podataka **vide korisnici**

Koncepti:

- ***Entiteti*** – **objekti** i **stavke** realnog sveta od interesa - o kojima se evidentiraju podaci
- ***Atributi*** – činjenice o, ili ***osobine entiteta***
- ***Povezanost*** - ***veze između entiteta***

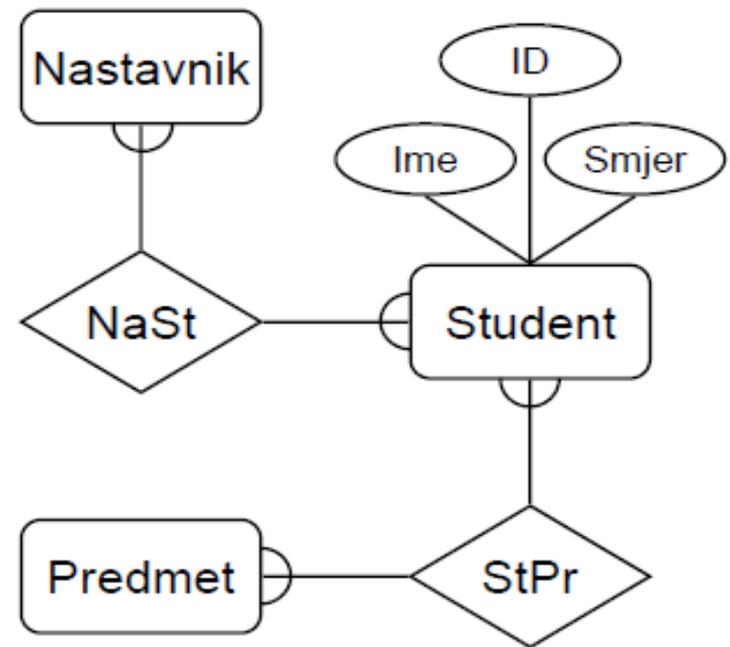
Model Objekti/veze (E/R model)

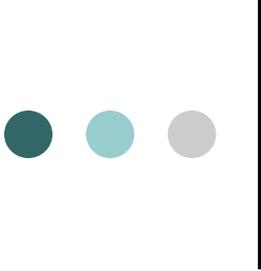
- Primer
- Na Fakultetu, baza podataka bi mogla imati **entitete** (opšte tipove – **klase entiteta**) za:
 - Studente,
 - Predmete i
 - Nastavnike
- Entitet Student može imati **attribute** kao
 - **broj indeksa (ID), ime, smer, ...**
 - može biti **povezan** sa entitetom **Predmet** i **Nastavnikom**

Model Objekti/veze (E/R model)

E/R modeli se često predstavljaju kao E/R dijagrami.

- Daju konceptualni izgled baze podataka.
- Nezavistni su od izbora DBMS.
- Pomažu u uočavanju mogućih problema u dizajnu baze podataka





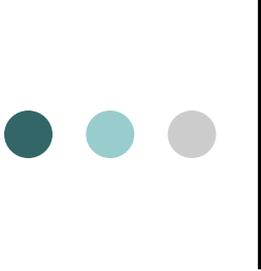
Entiteti u E/R modelu

Entiteti – objekti (od interesa) koji postoje i razlikuju se od svih drugih objekata

1. **Fizički objekti** kao studenti, nastavnici, radnici, proizvodi ...
2. Više **apstraktni objekti** kao predmeti, smerovi, projekti, ...

Entiteti imaju:

- **Opšti tip** ili **klasu**
 - Kao što su Nastavnik ili Predmet
 - Irena Orlić, Perica Štrbac su **instance opšteg tipa** Nastavnik



Entiteti u E/R modelu

Entiteti imaju:

- ***Atribute***
 - Kao što su ime, prezime, e-mail adresa, ...

Skup Entiteta (*Entity Set*) je skup entiteta ***istog tipa***

Primer: skup svih ljudi, kompanija, drveća, praznika

Model Objekti/veze (E/R model)

Skupovi entiteta *klijent* i *zajam*

customer_id customer_name customer_street customer_city loan_number amount

321-12-3123	Jones	Main	Harrison	L-17	1000
019-28-3746	Smith	North	Rye	L-23	2000
677-89-9011	Hayes	Main	Harrison	L-15	1500
555-55-5555	Jackson	Dupont	Woodside	L-14	1500
244-66-8800	Curry	North	Rye	L-19	500
963-96-3963	Williams	Nassau	Princeton	L-11	900
335-57-7991	Adams	Spring	Pittsfield	L-16	1300

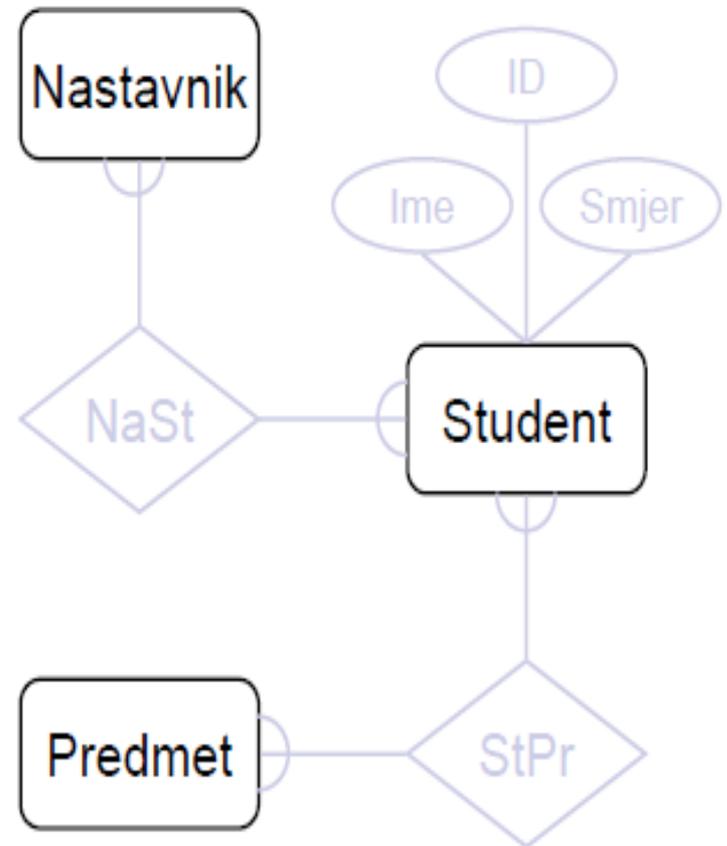
instancija

Tip entiteta

klijent **zajam**

Entiteti u E/R modelu

- U E/R dijagramu, entitet se obično crta kao pravougaonik (sa zaobljenim ivicama).
- Pravougaonik je označen sa imenom entiteta.

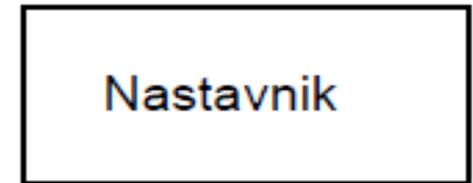


Entiteti u E/R modelu

- *Još neki načini* predstavljanja entiteta u E/R dijagramu:

- *Običan pravougaonik.*

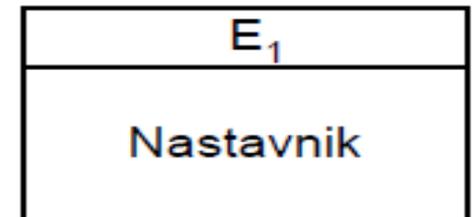
U sredini pravougaonika upisuje se ime entiteta.

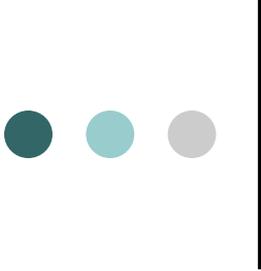


- *Pravougaonik* u čiji *donji*

deo se upisuje ime tipa entiteta, a u *gornji* redni

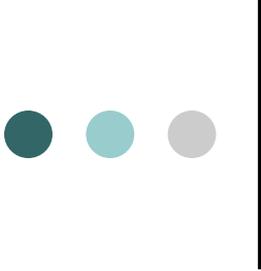
58 broj tipa entiteta





Atributi u E/R modelu

- **Atributi** su činjenice, aspekti, osobine, ili detalji koji **služe za opis entiteta**
 - **Studenti** imaju **broj indeksa** (ID), **ime**, **smer studija**, **adresu**, ...
 - **Predmeti** imaju šifru, naziv, ESPB kredite,
- **Atributi** imaju:
 - **ime**;
 - **entitet** kojem pripadaju;
 - **domen** mogućih vrednosti;
 - **vrednost** iz domena za svaki **entitetsku instancu**, na koju se odnosi atribut;



Atributi u E/R modelu

Tipovi Atributa:

- Prosti i Kompozitni atributi
- Sa **Jednom** (Single-valued) i **Više** (Multi-valued) **vrednosti**

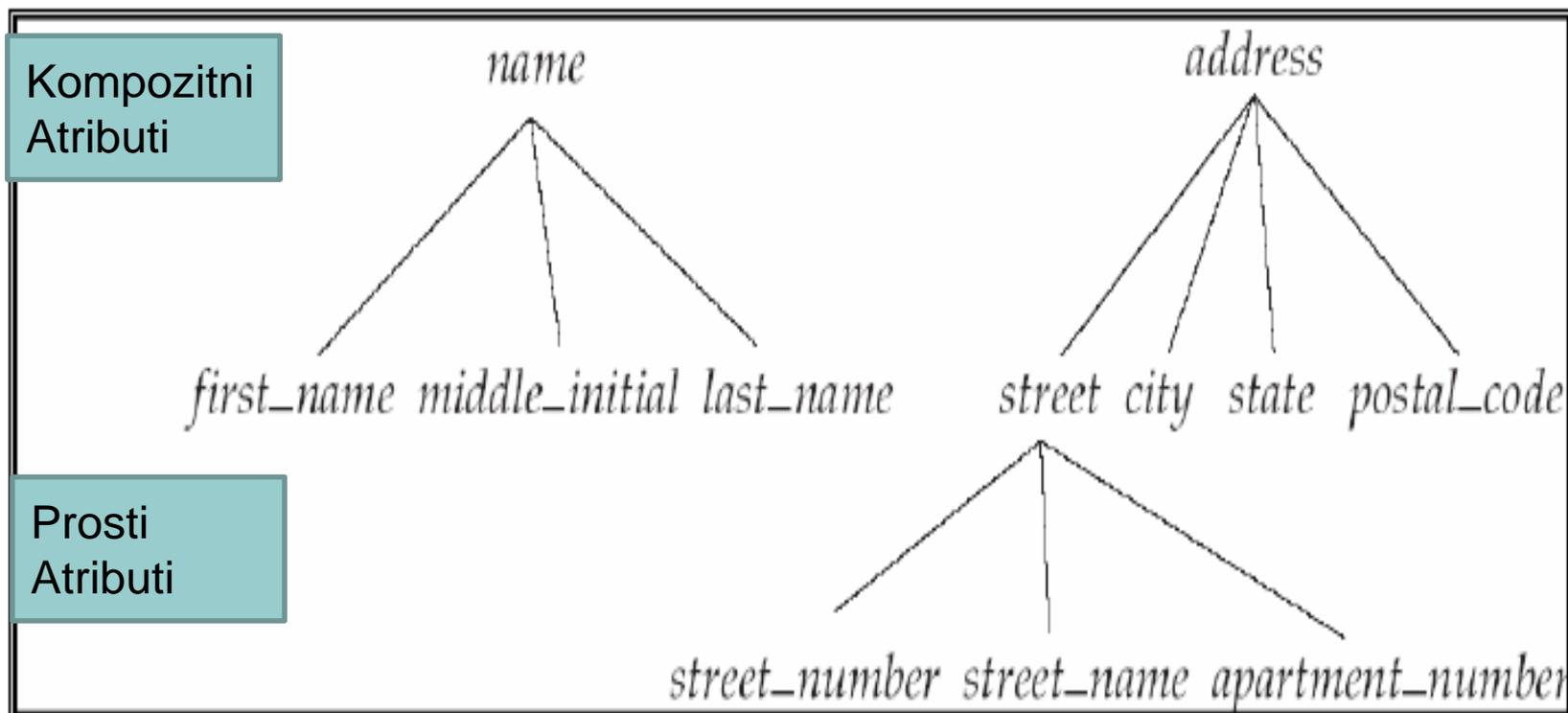
Primer:

- **više-vrednosni** (multi-valued) **atribut:**
broj_telefona
- Sračunljivi (*Derived*) **atributi**
- Mogu se sračunati iz drugih atributa

Primer: **starost**, sračunava se iz **datum_rođenja**

Atributi u E/R modelu

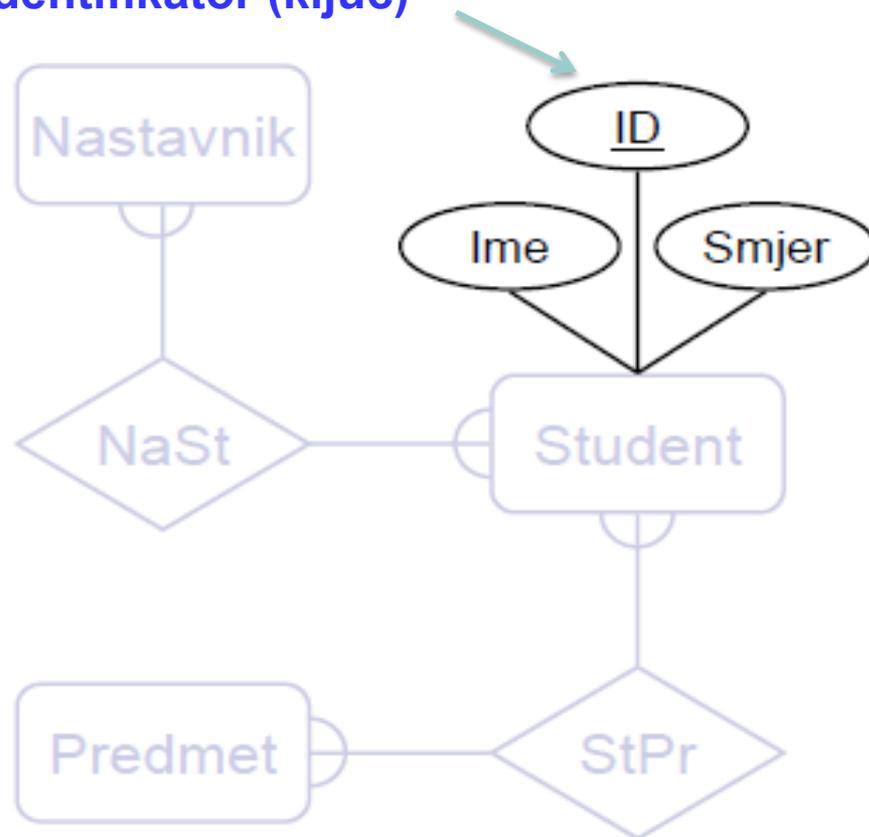
Tipovi Atributa - Primer:

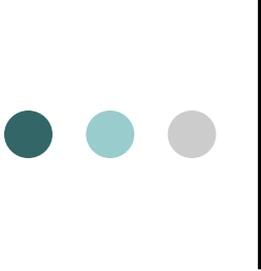


Atributi u E/R modelu

- U E/R diagramu atributi su ovalnog (elipsastog) oblika.
- Svaki atribut je pravom linijom povezan sa entitetom kojem pripada.
- Ime atributa zapisuje se unutar ovala.

Identifikator (ključ)





Veze (*Relationships*)

- Veza (*odnos*) je koncept koji oslikava neku *interakciju* (povezanost) *između dva* ili više *entiteta (objekata)*

Primer:

Hayes

klijent

entitet

ulaganje

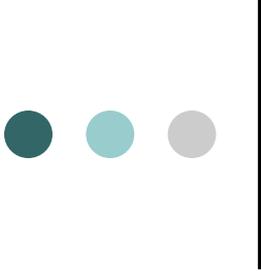
skup

veza

A-102

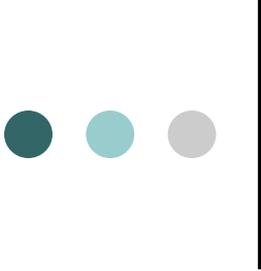
račun

entitet



Veze (*Relationships*)

- Veze poseduju sledeća obeležja:
 - ***Ime***
 - ***Atribute veze***
 - ***Stepen***
 - ***Kardinalnost***



Veze (*Relationships*)

- Veze imaju:
 - **Ime** – nazivi veza oslikavaju semantiku povezanosti (tj. prirodu interakcije) među objektima
 - Pojedini atributi (**primarni ključevi**) entiteta koji su povezani su i **atributi veze**
 - **Stepen** – broj skupova (tipova) entiteta koje povezuje (najčešće je stepan = 2)
 - **Kardinalnost** predstavlja **odnos broja objekata** koji se povezuju

Veze (*Relationships*)

- **Skup Veza** (*Relationship set*) je matematička relacija između $n \geq 2$ entiteta, od kojih svaki pripada jednom od povezanih skupova entiteta

$\{(e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$

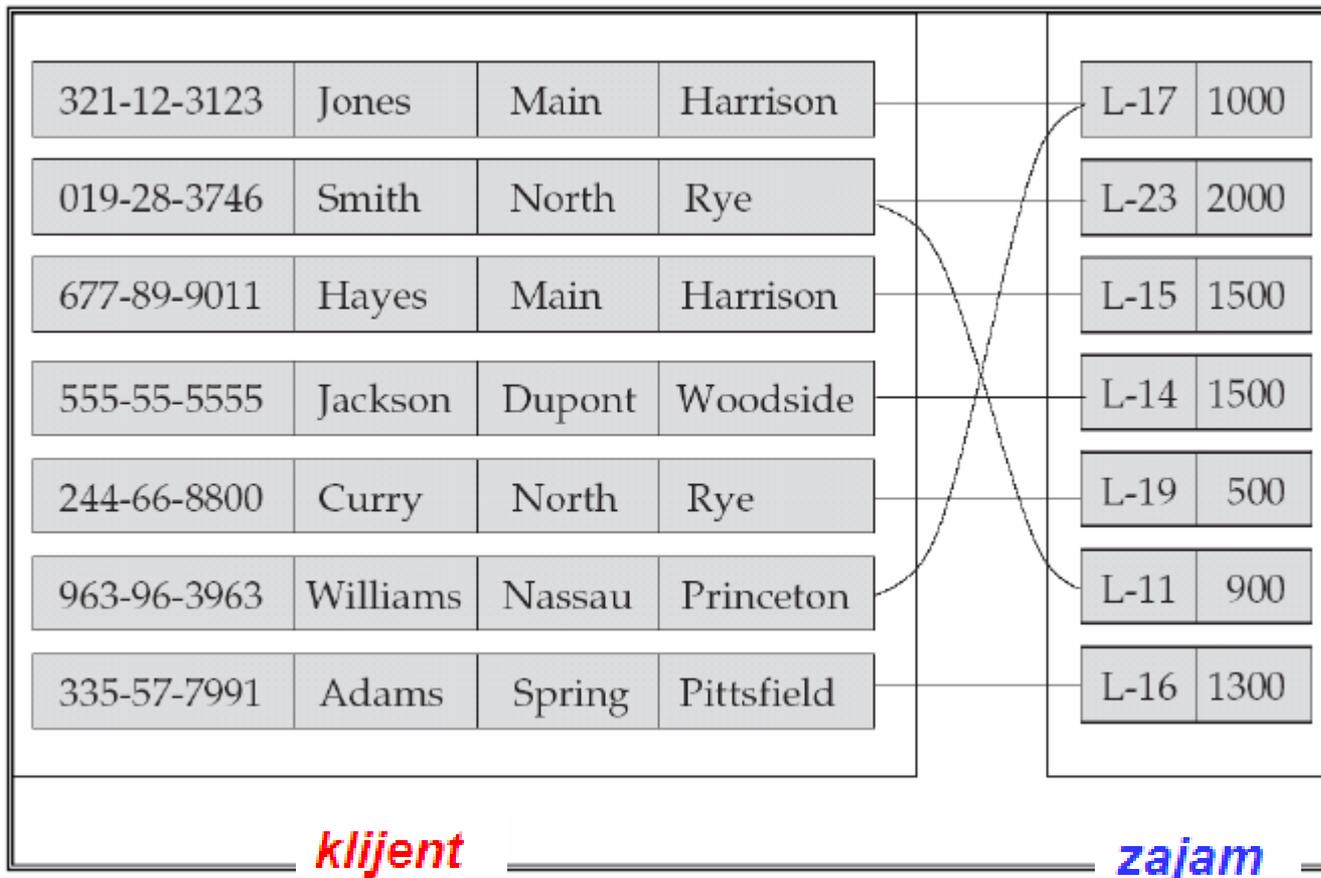
gde je (e_1, e_2, \dots, e_n) veza

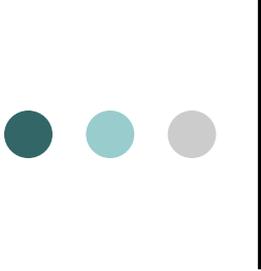
- Primer:

(Smith, L-11) ∈ pozajmiti

Veze (*Relationships*)

- Skup veza *pozajmiti*





Skup veza

- Skup Veza može takođe imati ***attribute***

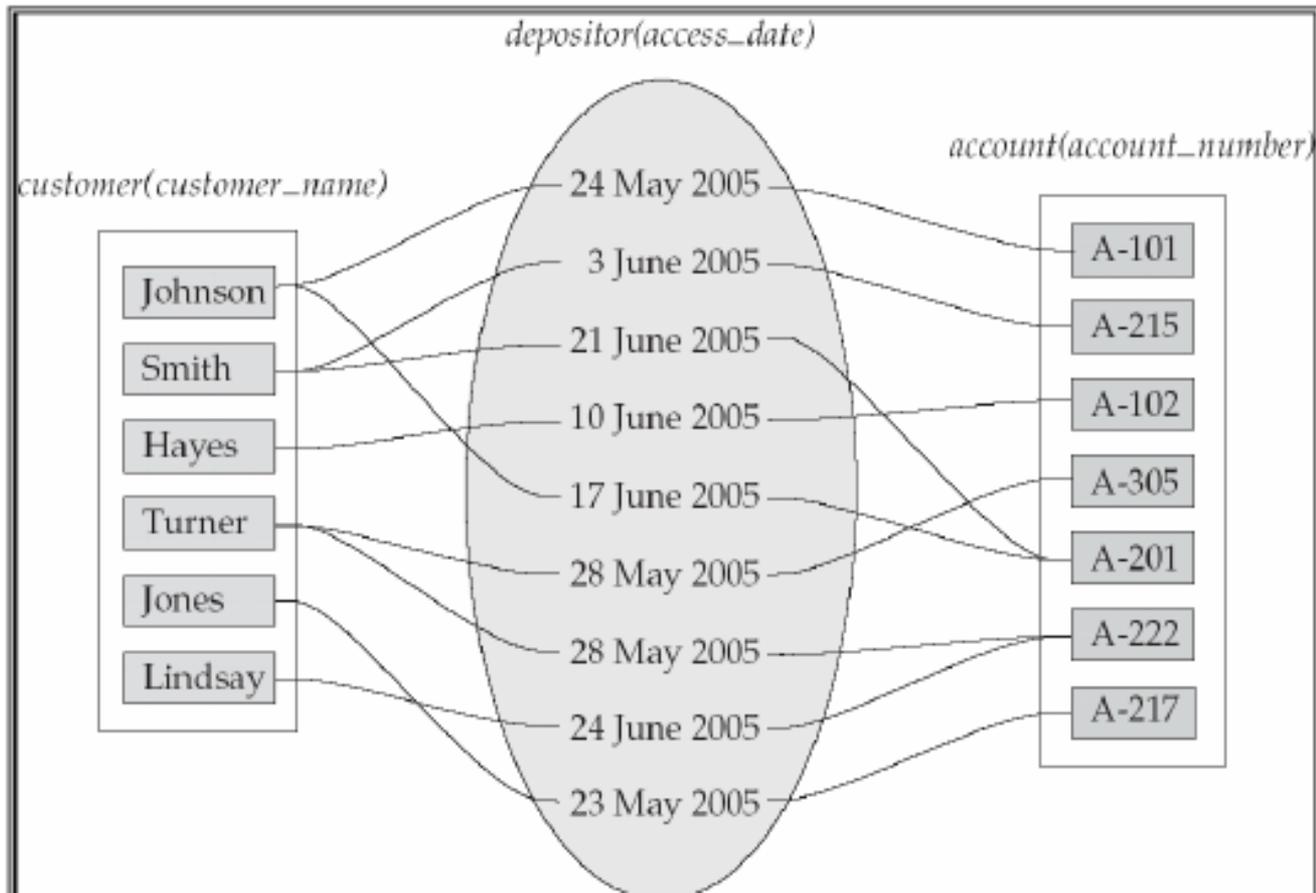
Na primer,

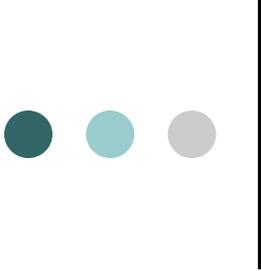
- Skup veza ***ulaganje*** (***depositor***) između skupova entiteta ***klijent*** (***customer***) i ***račun*** (***account***) može imati **atribut** ***datum-pristupa*** (***access-date***)

Skup veza

- **Skup veza** može takođe imati **atribute**

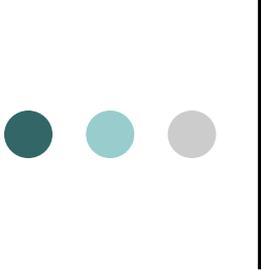
Na primer,





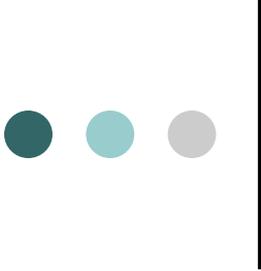
Stepen Skupa veza

- Odnosi se na broj skupova entiteta koji učestvuju u skupu veza
- Skupovi veza u koji uključuju dva skupa entiteta su **binarni** (ili stepena dva)
- Većina skupova veza u bazama podataka su **binarni**
- Skupovi veza mogu uključivati više nego dva skupa entiteta – ali su vrlo retki



Stepen Skupa veza

- Uobičajeni stepeni odnosa su:
- **Binarni**
 - Odnos između **dva skupa** (tipa) entiteta
- **Ternarni**
 - Odnos između **tri skupa** (tipa) entiteta
- **Rekurzivni**
 - Odnos koji uključuje samo **jedan skup** (tip) entiteta



Rekurzivni tip veze

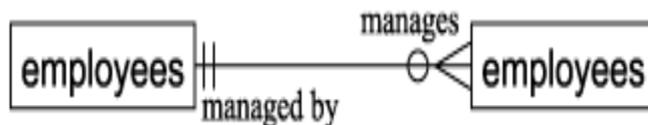
- Verovatno ***najkomplikovaniji*** je ***rekurzivni odnos***
- To je **odnos** koji entitet gradi **sa samim sobom**
- Može se zamisliti i ***kao binarni odnos*** nad ***dva ista skupa (tipa)*** entiteta

Primer:

- ***Zaposleni*** koji je ***direktori*** zaposlenima

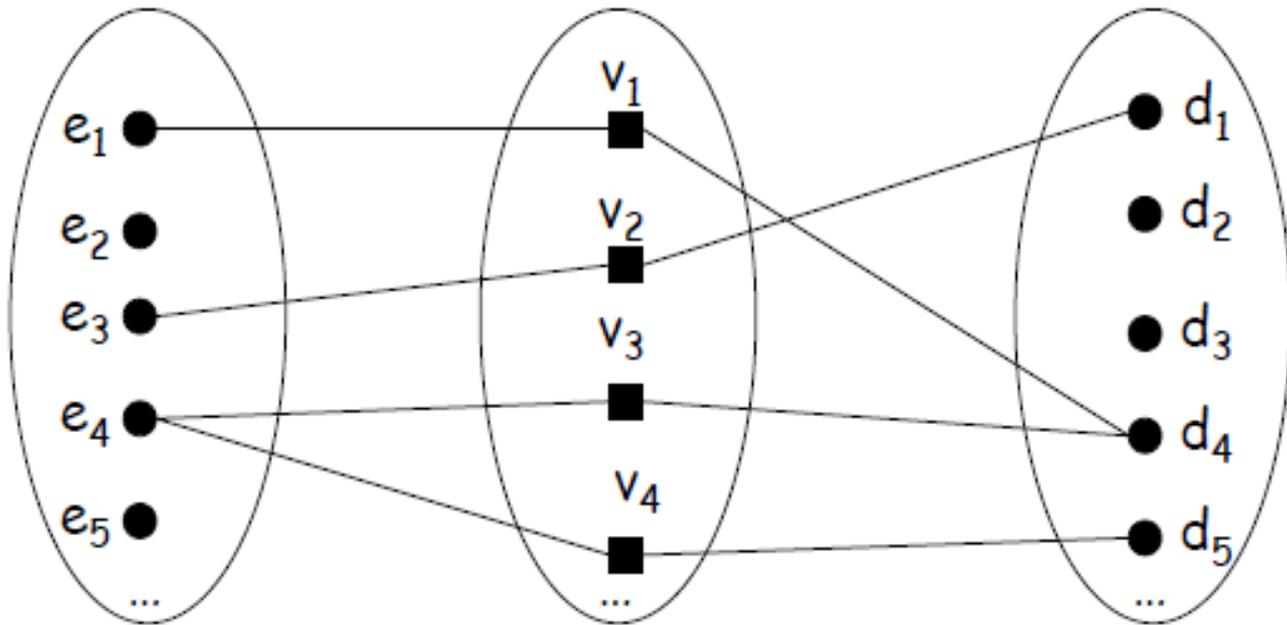
Rekurzivni tip veze - Primer

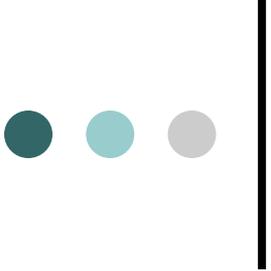
Jedan direktor upravlja sa više zaposlenih, a jedan zaposleni ima tačno jednog neposrednog direktora. Ali direktor je samo još jedno ime za zaposlenog. Zato se *direktori* (managers) može zameniti sa *zaposleni* (employees) u ovom dijagramu:



Veze (*Relationships*)

- Ponekad je **veze zgodno prikazati** pomoću dijagrama:

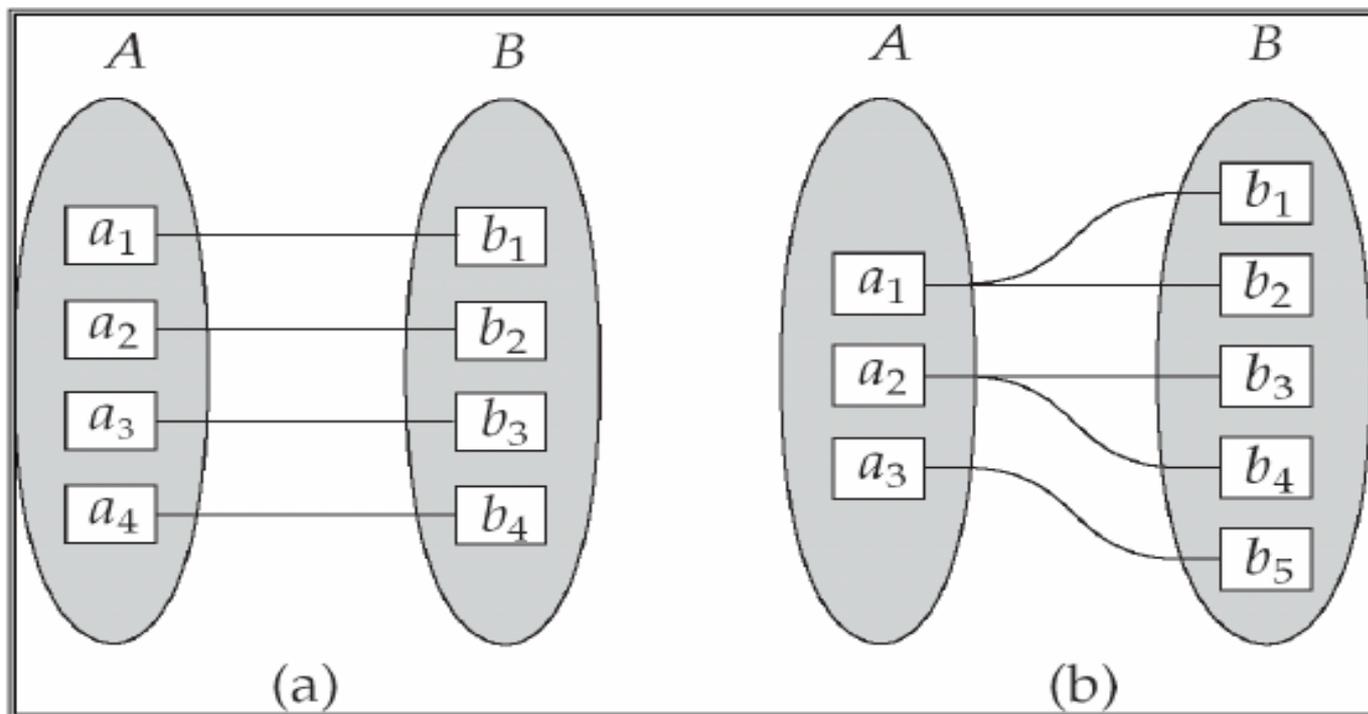




Kardinalnost preslikavanja

- Izražava **broj entiteta** koji se mogu dovesti **u vezu sa drugim entitetom**
- Najupotrebljivija za opisivanje **binarnog** skupa veza
- Za binarni skup veza **kardinalnost preslikavanja** može biti:
 - **Jedan-na-jedan** (*One to one*)
 - **Jedan-na-više** (*One to many*)
 - **Više-na-više** (*Many to many*)

Kardinalnost preslikavanja



jedan-na-jedan

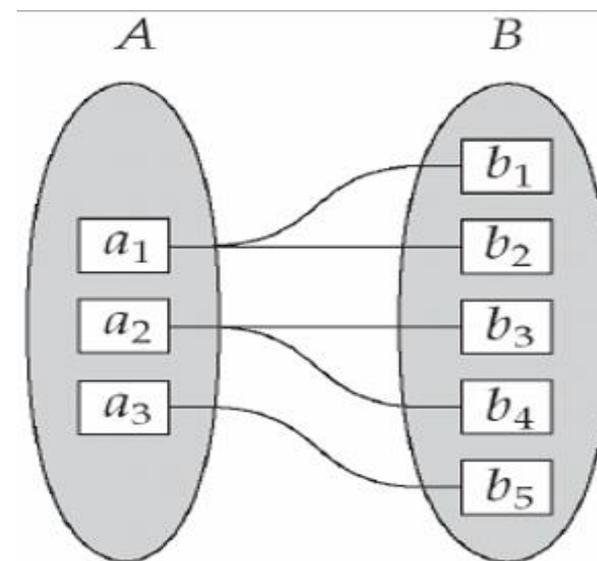
jedan-na-više

Primedba: Neki elementi u A i B ne moraju biti mapirani ni na jedan element drugog skupa

Kardinalnost preslikavanja

Jedan na više (1:M)

Jednom entitetu drugog skupa entiteta pridružuje se **jedan entitet prvog** skupa entiteta, ali **jednom entitetu prvog** skupa entiteta može se pridružiti **više entiteta drugog** skupa entiteta

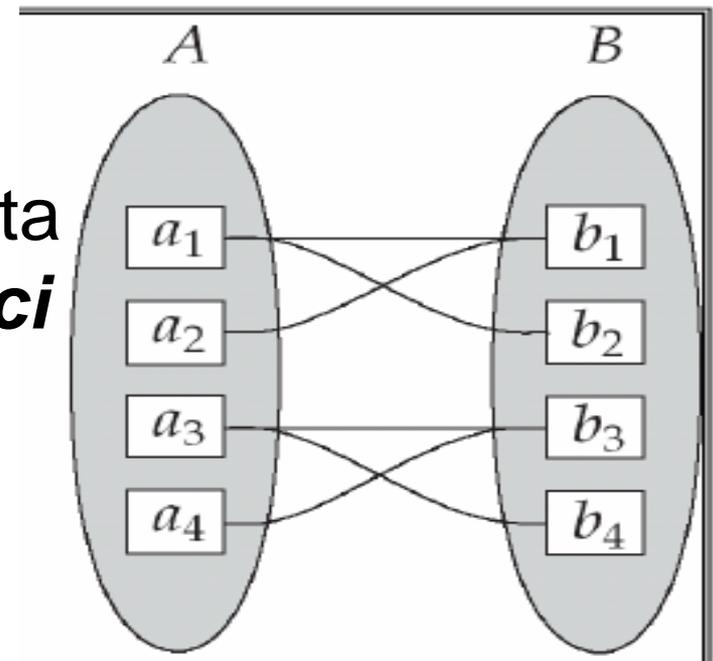


Primedba: Neki elementi u A i B ne moraju biti mapirani ni na jedan element drugog skupa

Kardinalnost preslikavanja

Više na više (M:M)

Jednoj instanci prvog entiteta pridružuje se **više instanci drugog** entiteta, i obrnuto

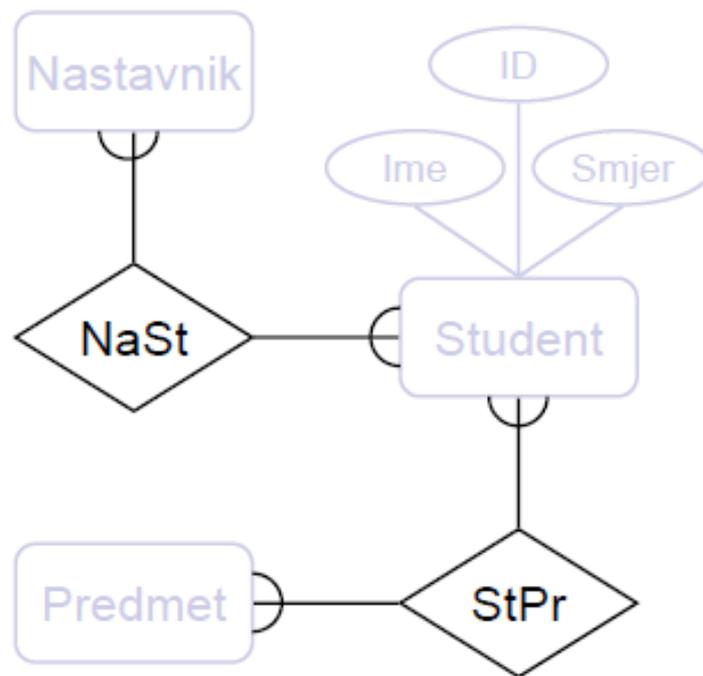


više na više (M:M)

Primedba: Neki elementi u A i B ne moraju biti mapirani ni na jedan element drugog skupa

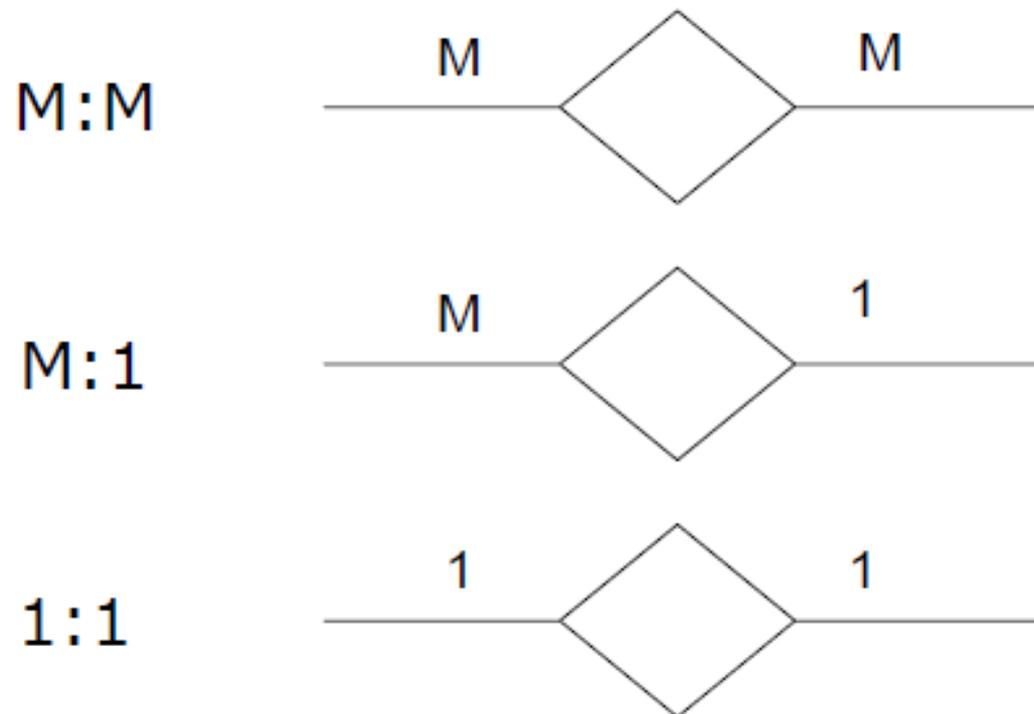
Veza u E/R modelu

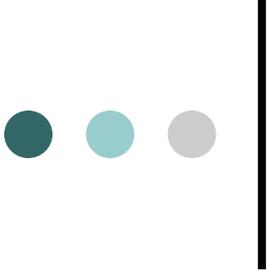
- U E/R diagramu veze su oblika romba.
- Unutar romba upisano je ime veze
- Kraj linije prikazuje kardinalnost preslikavanja



Veza u E/R modelu

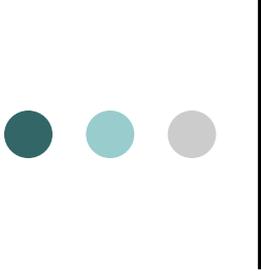
- Još neki **načini označavanja kardinalnosti preslikavanja** veze u E/R dijagramu:





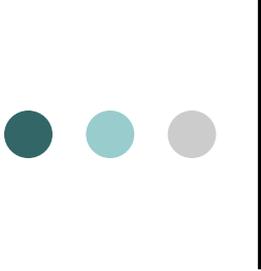
Definisanje ključeva u modelu

- **Super ključ** (*super key*) skupa entiteta je skup jednog ili više atributa čije vrednosti **jednoznačno određuju** svaki entitet
- **Kandidat ključ** (*candidate key*) skupa entiteta je **minimalni super ključ**
- **Customer_id** je kandidat ključ za **customer**
- **account_number** je kandidat ključ za **account**
- **Primarni ključ** (*primary key*) je **izabrani** kandidat ključ



Ključ skupa veza

- **Skup primarnih ključeva** skupova entiteta u nekoj vezi je **super ključ skupa veza**
 - (**customer_id**, **account_number**) je super ključ za depositor (vezu ulaganje)
- Mora se uzeti u obzir **kardinalnost preslikavanja** skupa veza pri odluci o **kandidat ključu**
- Potrebno je uzeti u obzir **semantiku skupa veza** pri **izboru primarnog ključa** u slučaju više od jednog kandidat ključa



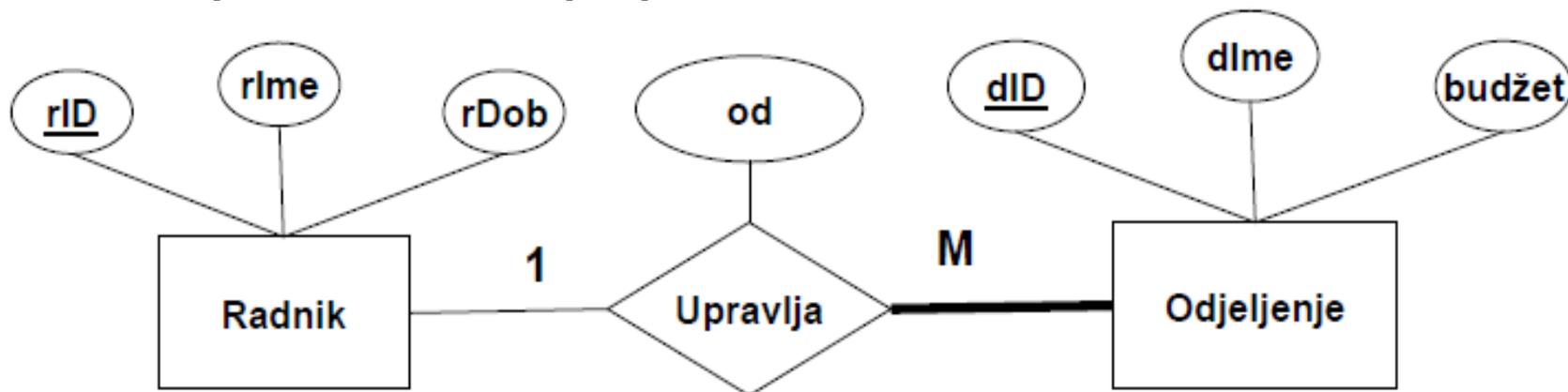
Ograničenje učešća

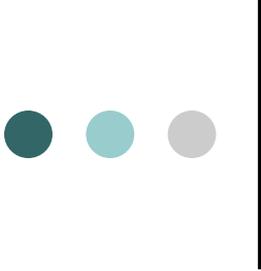
Svako **odjeljenje** mora imati **rukovodioca**

- Ovo je **primer ograničenja učešća** (*participation constraint*)
- **Totalno učešće**: **svaki entitet** u skupu entiteta **učestvuje** u **najmanje jednoj vezi** u skupu veza
- Ako nije tako za **učešće** se kaže da je **parcijalno** (*partial*)

Učešće u E/R dijagramima

- **Totalno učešće** se u dijagramu prikazuje kao **deblja (bold) linija** između entiteta i relacije
- Ponekad se totalno učešće prikazuje i kao **dupla linija** ili tačka na rombu veze na početku linije prema entitetu



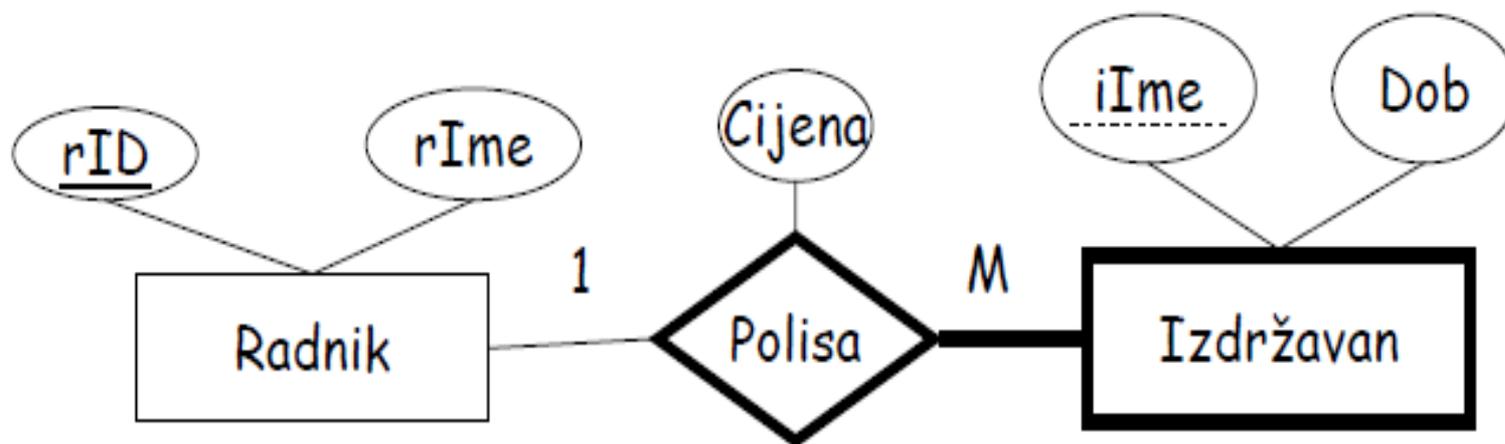


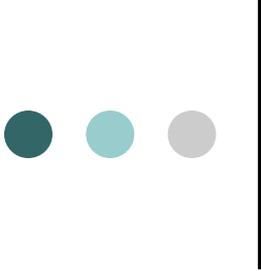
Skup slabih entiteta

- **Skup entiteta** koji **nema dovoljno atributa** za formiranje **primarnog ključa**
- Takav skup entiteta se naziva **skup slabih entiteta** (***weak entity type***)
- Slabi entitet može biti **jedinstveno indentifikovan** samo uzimajući u obzir **primarni ključ drugog tzv. identifikacionog skupa entiteta (vlasničkog entiteta)**

Skup slabih entiteta

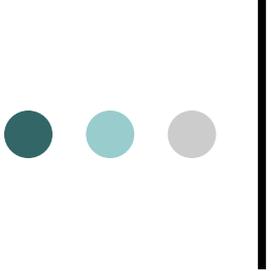
- **Skupovi identifikacionih i slabih entiteta** moraju biti povezani **1:M vezom**
- Slabi entitet mora imati **totalno učešće** u ovoj **identifikacionoj vezi**





Prošireni model Objekti/Veze

- Pored osnovnog, **postoji** i **prošireni model objekti/veze**
- Omogućava **detaljnije definisanje veza** između objekata
- Postoje **specifične veze kojima se izražava hijerarhija** i komponovanje objekata
 - Pored **asocijativnih veza** koje **oslikaju semantiku udruživanja** objekata u sistemu



Prošireni model Objekti/Veze

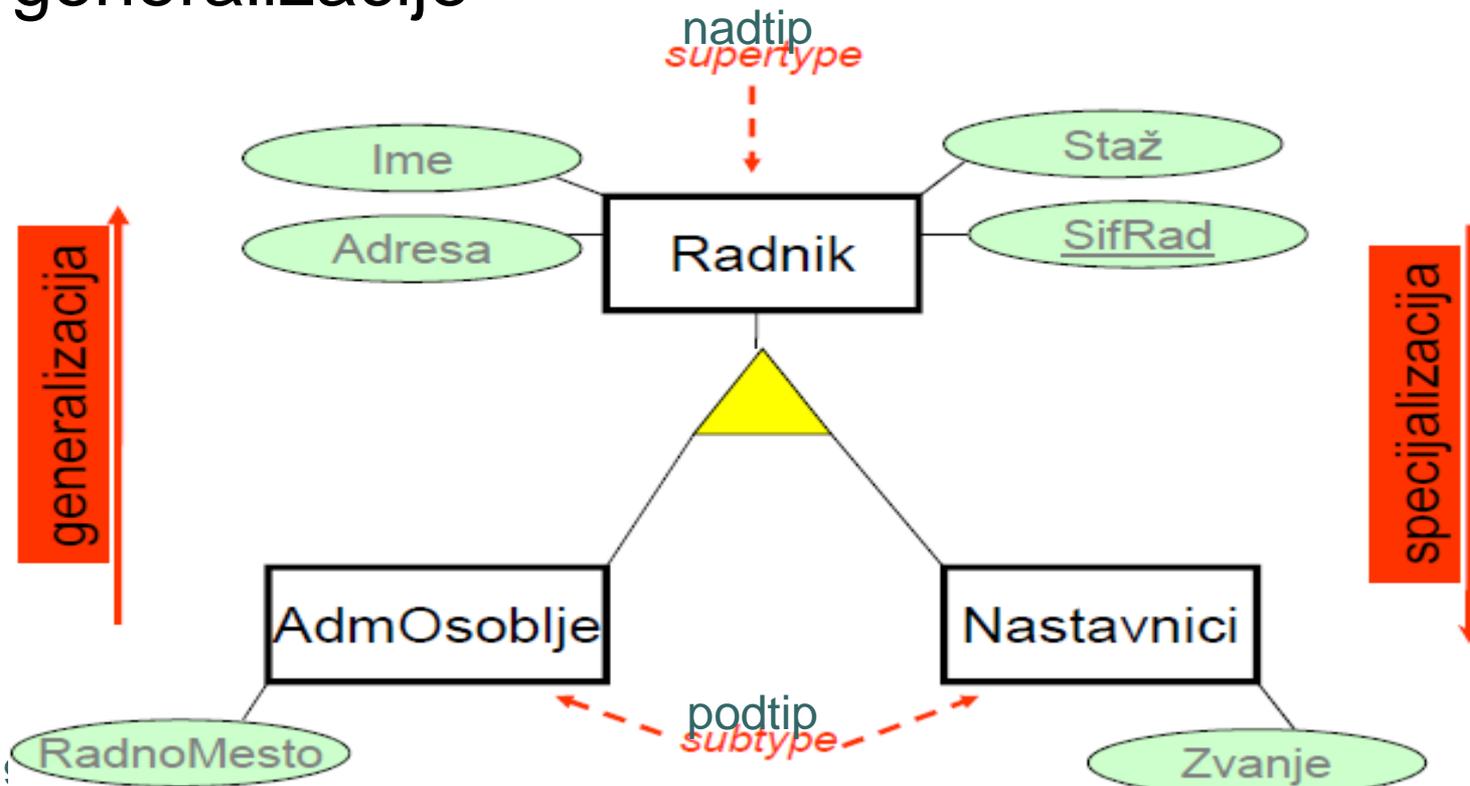
- ***Postoje dve reprezentativne vrste ovakvih veza:***
 - Specijalizacija/generalizacija***
 - Agregacija***

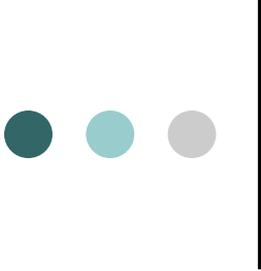
Generalizacija (Nadtip/Podtip)

- **Generalizacija** je **apstrakcija** u kojoj se **skup sličnih tipova objekata** **predstavlja** **opštim generičkim tipom** ili **nadtipom**
- **Slični tipovi objekata** su oni koji imaju **zajedničke osobine i ponašanje**
 - Npr: **klasa** Nastavnik i AdmOsoblje se može **generalizovati** u klasu Radnik
- **Podtip** **nasleđuje osobine** (atribute) nadtipa
- **Podtip** može imati i neke svoje **posebne osobine**

Specijalizacija

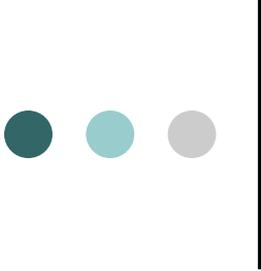
- **Specijalizacija** je **obrnuti postupak** od generalizacije





Agregacija

- Agregacija je skup veza koja se ponaša kao skup entiteta (objekata) i **može da učestvuje u drugim vezama**
- Agregacija je apstrakcija u kojoj se:
 - skup tipova entiteta i
 - njihovih vezatretira kao jedinstveni agregirani **tip entiteta**

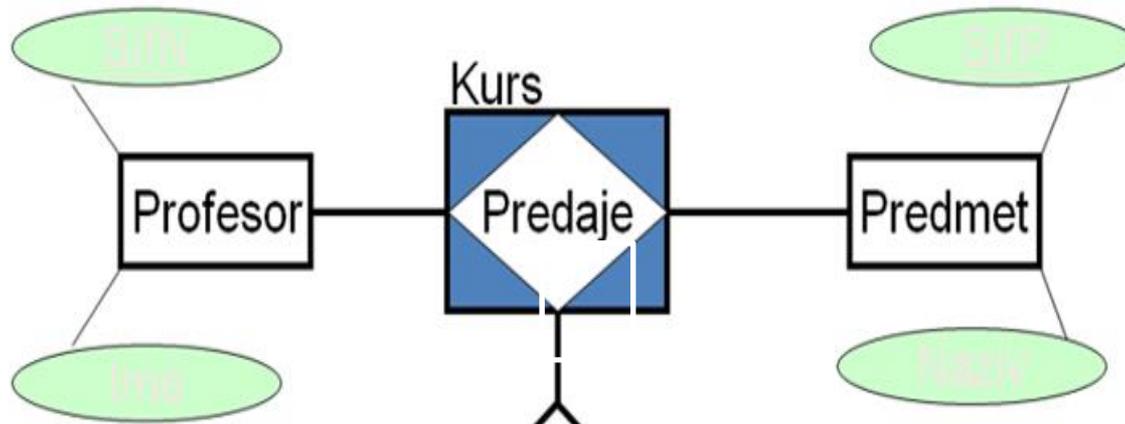


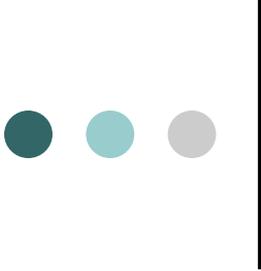
Agregacija

- Zbog toga što ***istovremeno predstavlja*** i ***entitet*** i ***vezu*** agregacija se **često naziva** i ***mešoviti tip objekta-veza***

Agregacija

- Na primer
- ***Skup entiteta*** Nastavnik i Predmet ***povezane su skupom veza*** Predaje



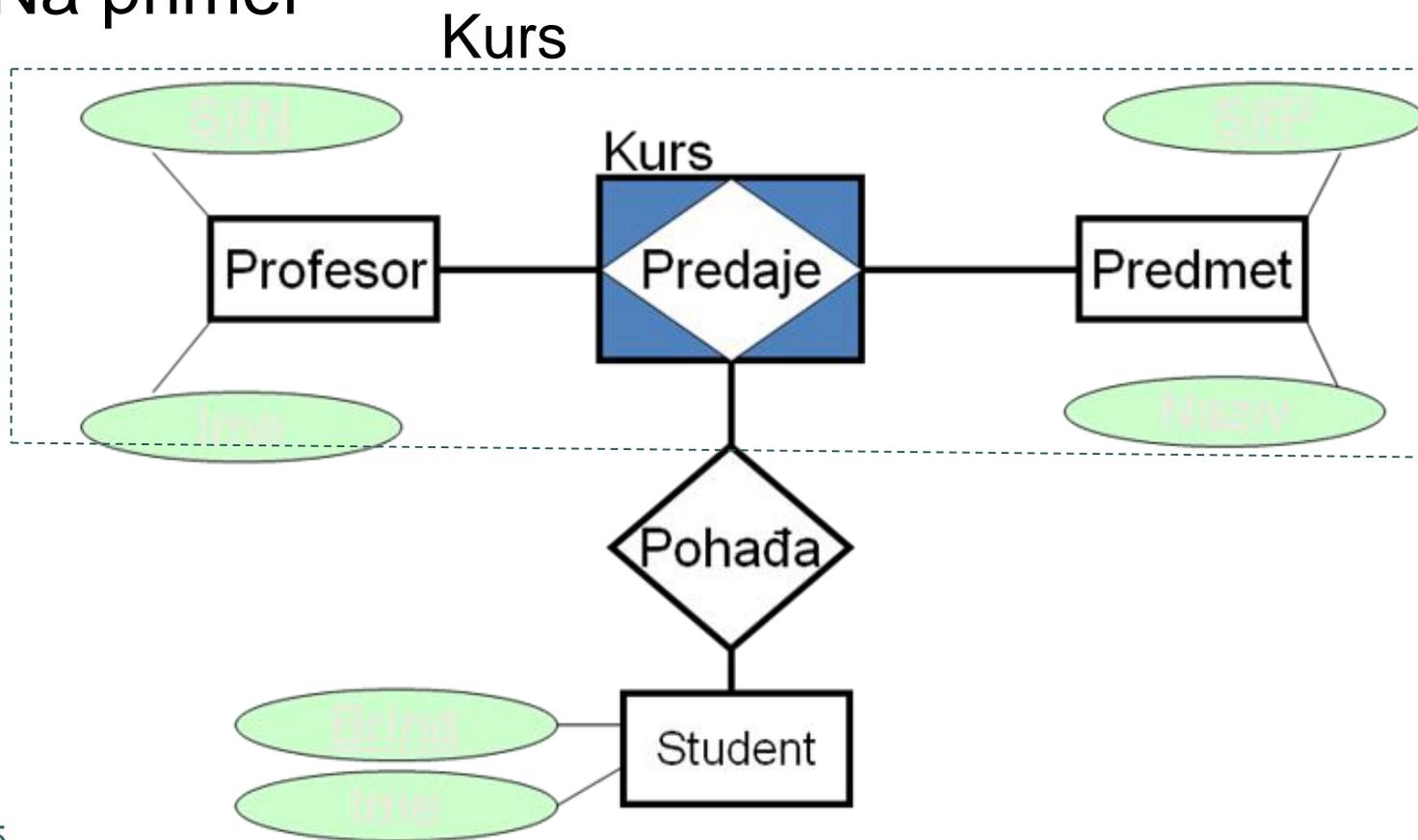


Agregacija

- Na primer
- **Studenti** slušaju predavanja iz **određenih predmeta**, ali kod **određenih nastavnika**
- **Skup veza** (Predaje) se posmatra kao agregacija (**skup entiteta**) - **Kurs**
- Između skupa objekata **Kurs** i **Student** može se uspostaviti veza **Pohađa**

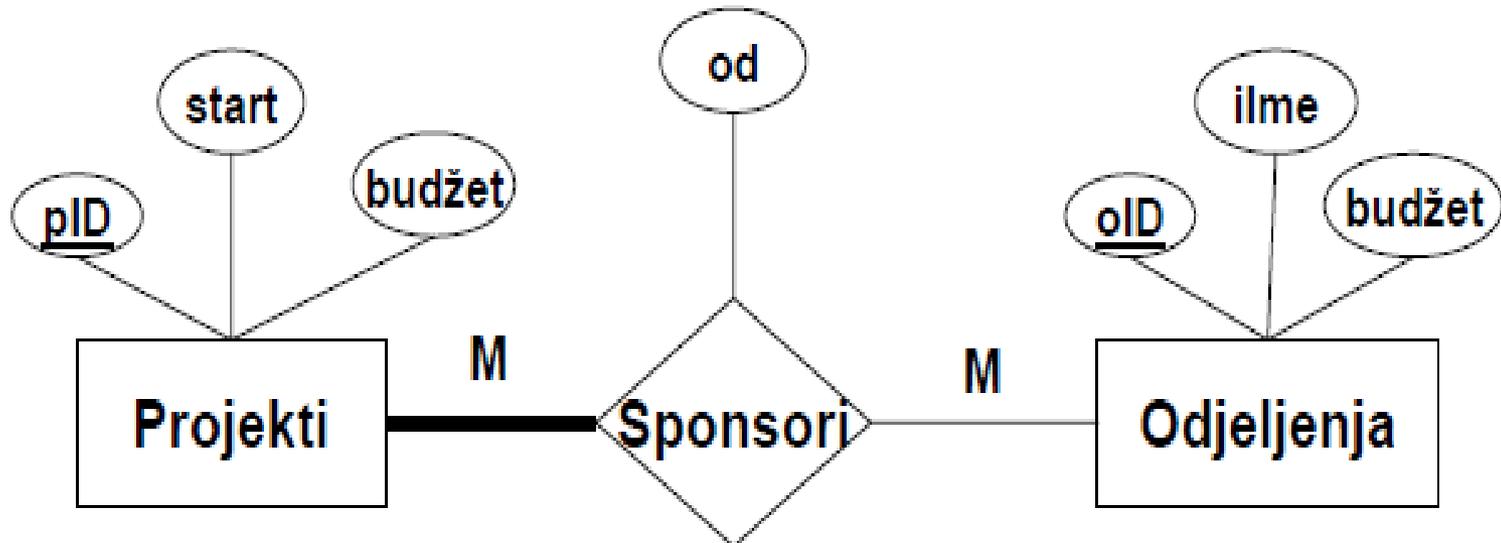
Agregacija

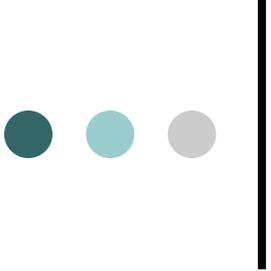
- Na primer



Agregacija

- Drugi primer,
- Pretpostavimo da imamo skup entiteta *Projekti* i da je svaki projekat **sponzorisan** od strane jednog ili više *odjeljenja*



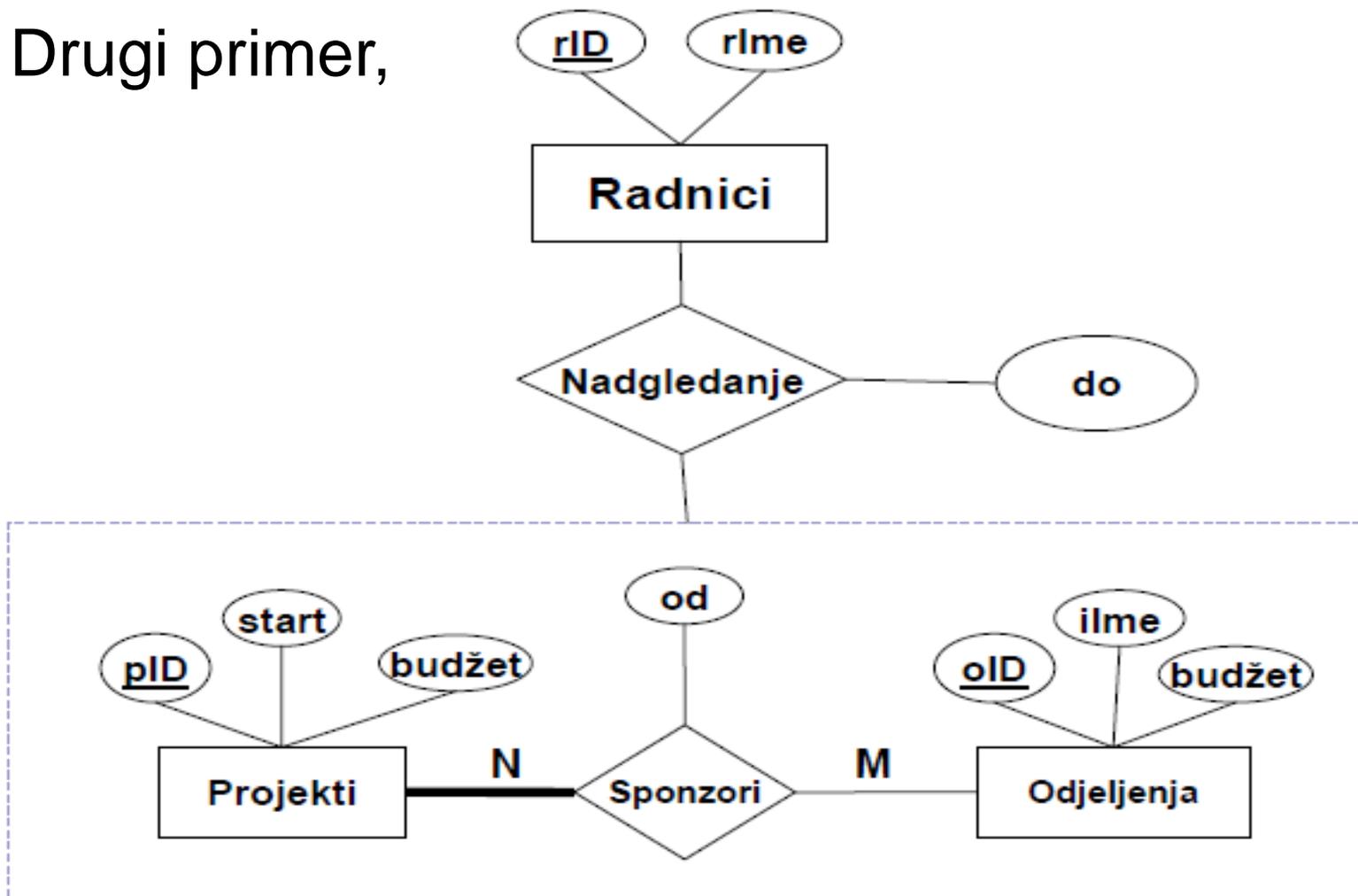


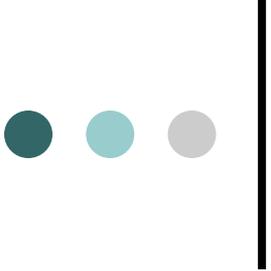
Agregacija

- Drugi primer,
- Neka *postoje radnici* koji su zaduženi da *nadgledaju sponzorisanje*
- Nedgledanje bi bila veza između Radnika i Sponzori **veze**
- *Agregacija* se koristi da se ukaže da jedna veza učestvuje u drugoj vezi
- Koristiti isprekidane linije

Aggregacija

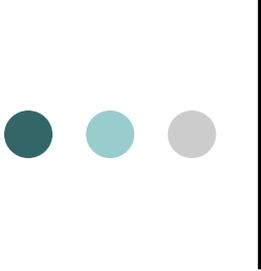
Drugi primer,





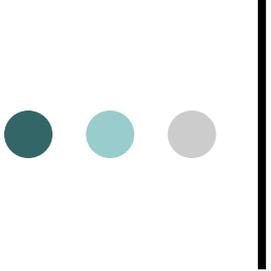
Pravljenje E/R modela

- Prilikom pravljenja E/R modela, ***iz opisa nemene***, potrebno je ***prepoznati***
 1. Entitete
 2. Atribute
 3. Veze
 4. Kardinalnost preslikavanja



Pravljenje E/R modela

- Opšte smernice:
- Kako su **entiteti** uglavnom bića, stvari ili objekti oni su najčešće imenice u opisu
- **Atributi** su činjenice ili **osobine** pa su i oni često imenice
- **Glagoli** često opisuju veze između entiteta



Pravljenje E/R modela - Primer

Univertitet se sastoji iz više odseka.

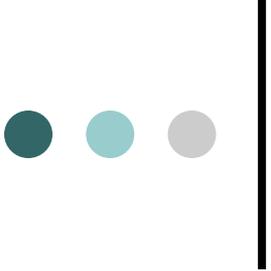
Svaki odsek nudi više smerova.

Više predmeta sačinjavaju svaki smer.

Studenti upisuju određeni smer i uzimaju predmete koje taj smer nudi.

Svaki predmet predaje nastavnik iz odgovarajućeg odseka.

Svaki nastavnik drži nastavu grupi studenata.



Pravljenje E/R modela - Primer

Univertitet se sastoji iz više **odseka**.

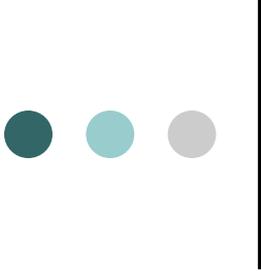
Svaki odsek nudi više **smerova**.

Više **predmeta** sačinjavaju svaki smer.

Studenti upisuju određeni smer i uzimaju predmete koje taj smer nudi.

Svaki predmet predaje **nastavnik** iz odgovarajućeg odseka.

Svaki nastavnik drži nastavu grupi studenata.



Pravljenje E/R modela - Primer

Univertitet se sastoji iz više odseka.

Svaki odsek **nudi** više smerova.

Više predmeta **sačinjavaju** svaki smer.

Studenti **upisuju** određeni smer i **uzimaju** predmete koje taj smer nudi.

Svaki predmet **predaje** nastavnik **iz** odgovarajućeg odseka.

Svaki nastavnik **drži** nastavu grupi studenata.

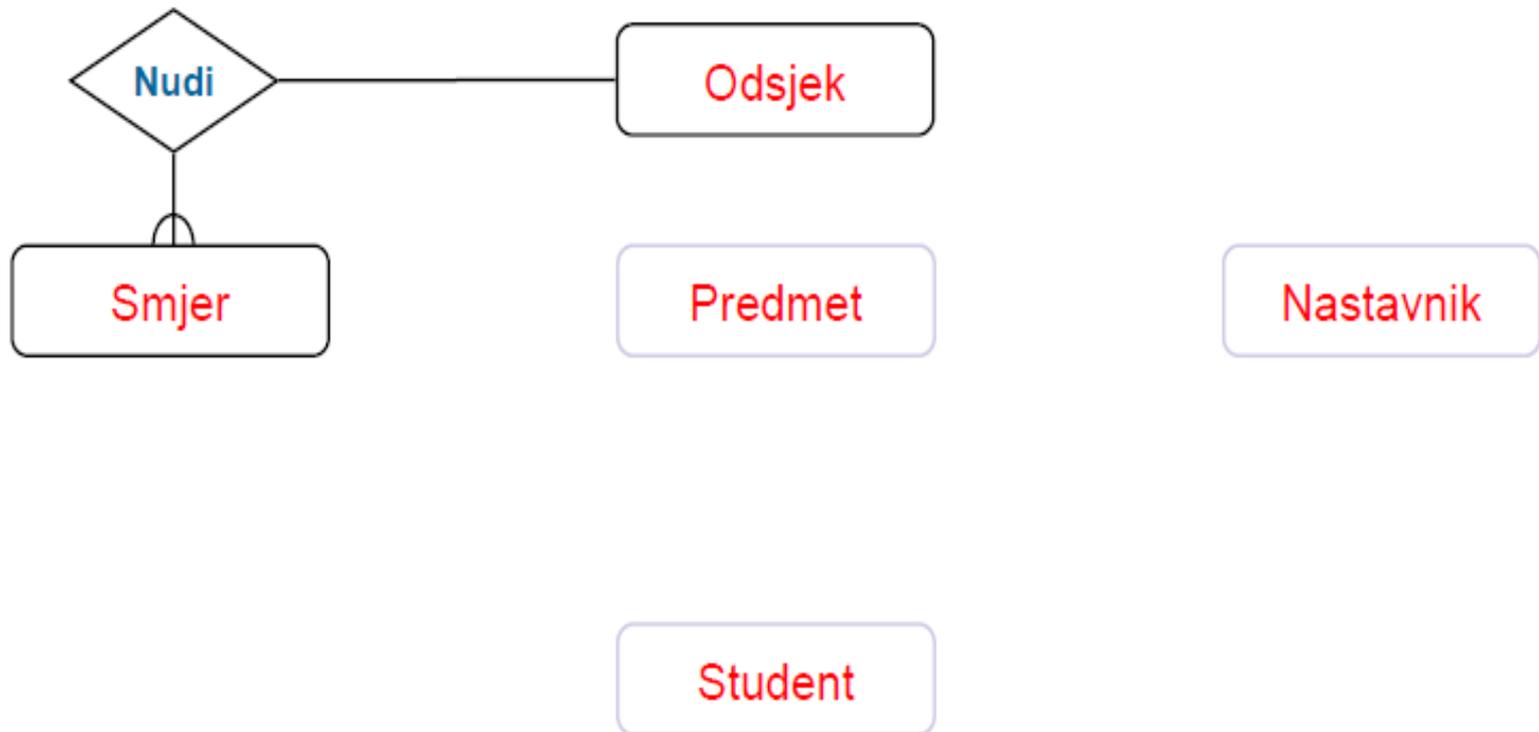
Pravljenje E/R modela - Primer

- *Entiteti*: Odsek, Smer, Predmet, Nastavnik, Student



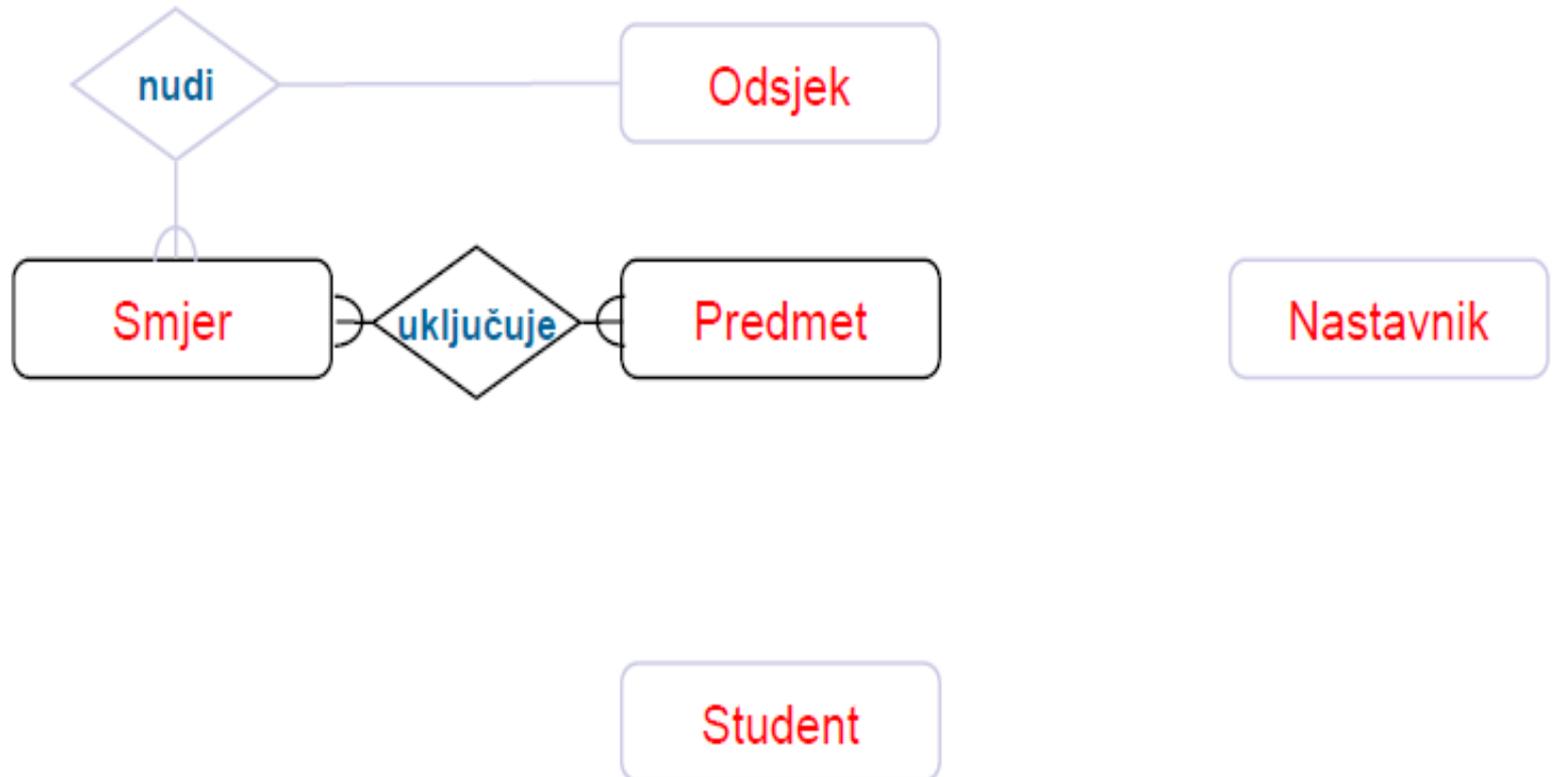
Pravljenje E/R modela - Primer

- Svaki **odsek nudi više smerova**



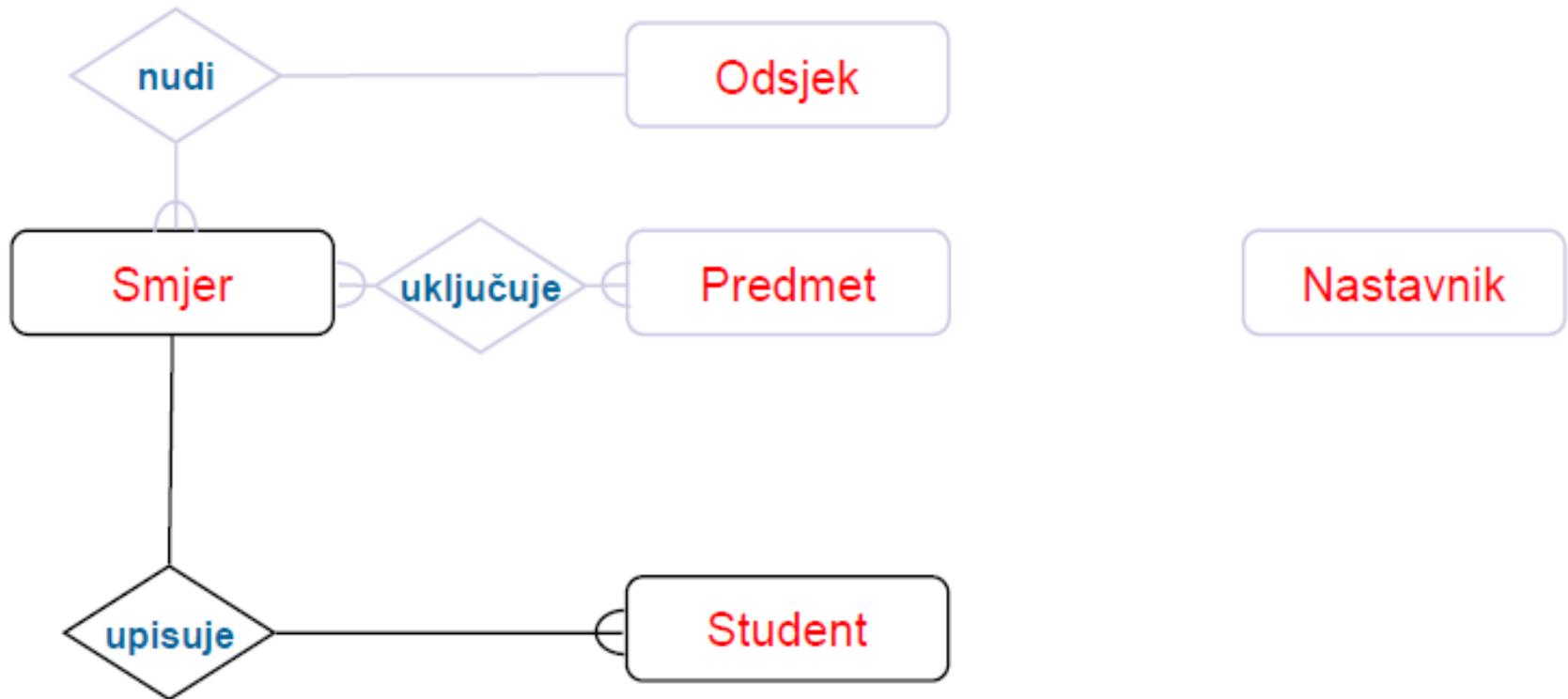
Pravljenje E/R modela - Primer

- Više **predmeta uključuje** svaki **smjer**



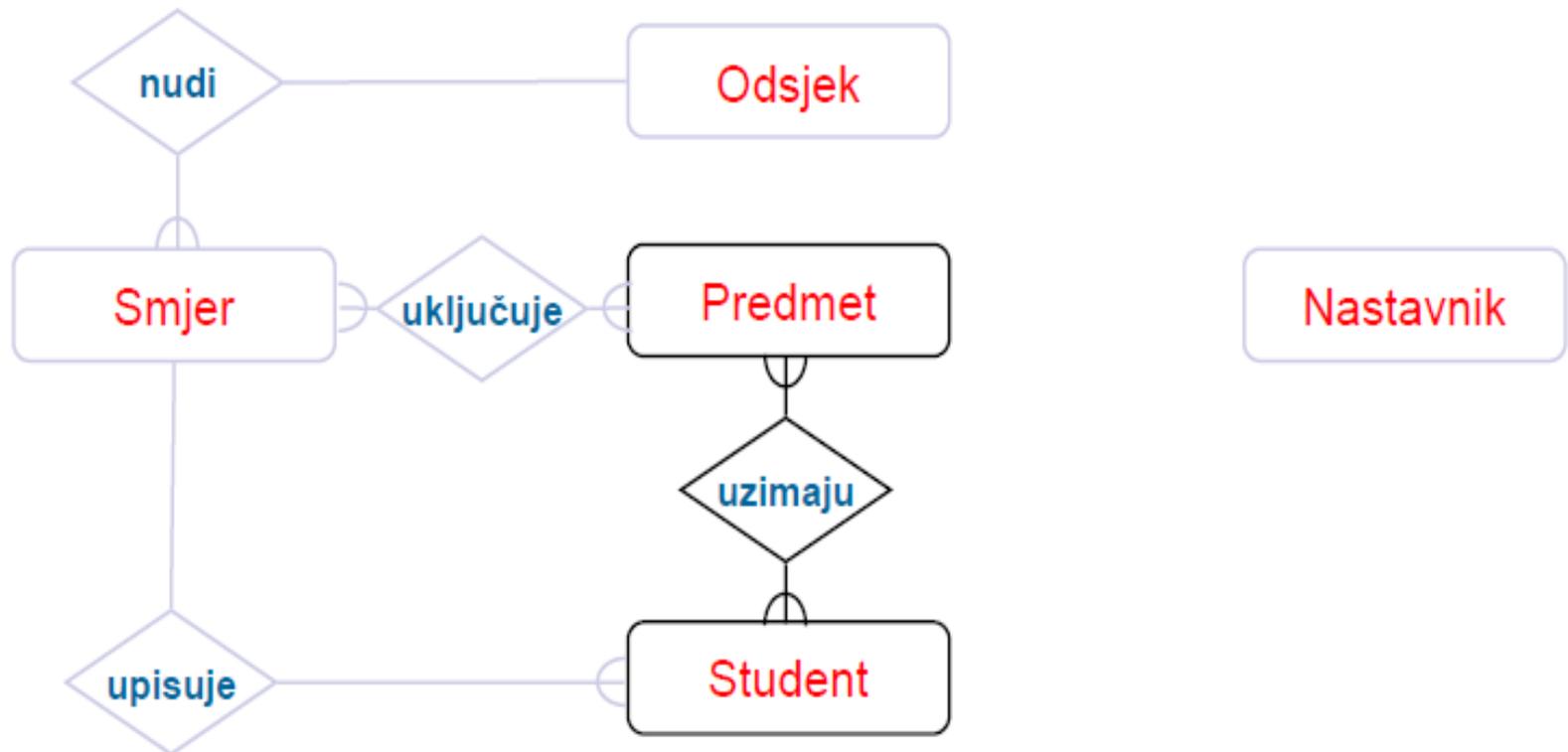
Pravljenje E/R modela - Primer

- **Studenti upisuju** određeni **smjer**



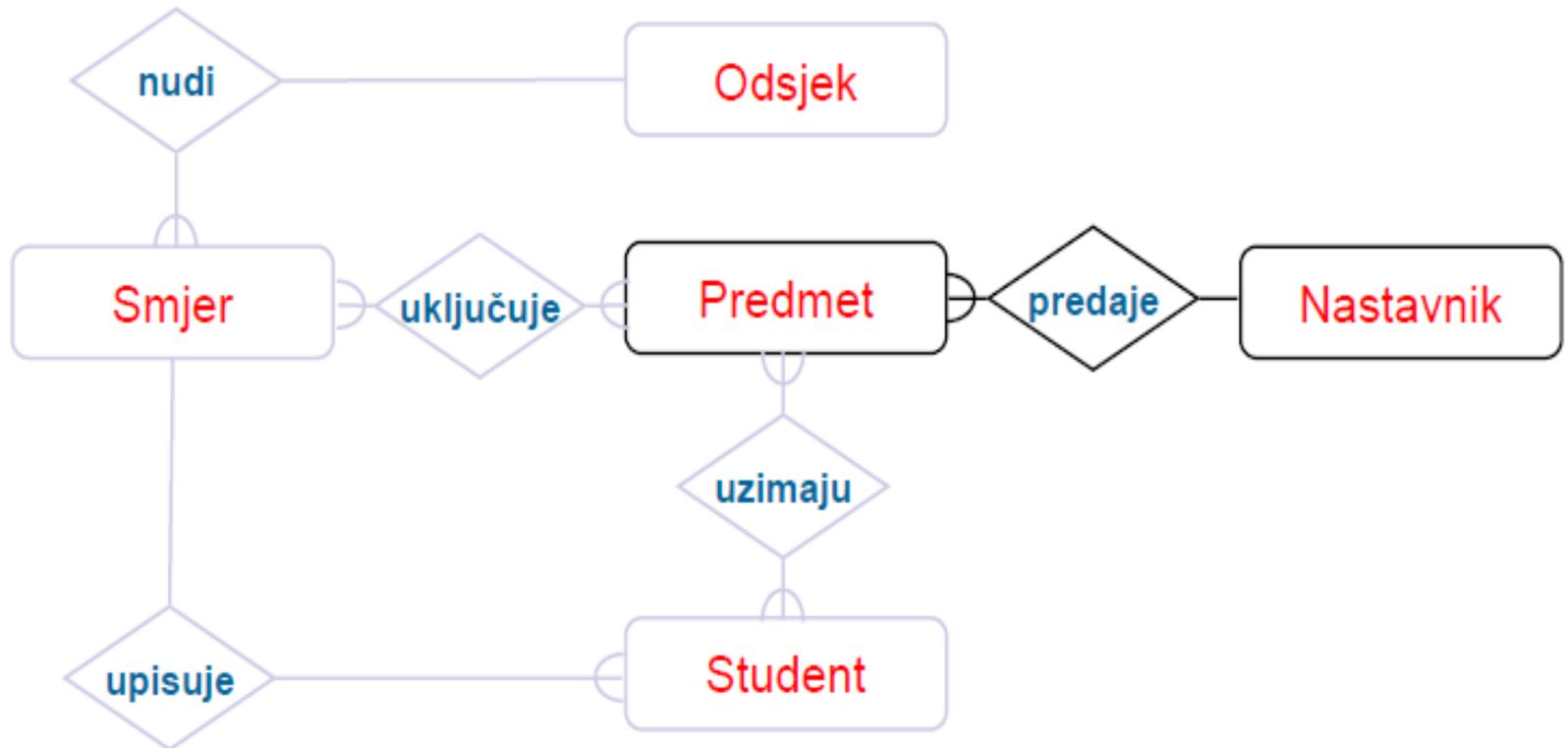
Pravljenje E/R modela - Primer

- **Studenti uzimaju predmete**



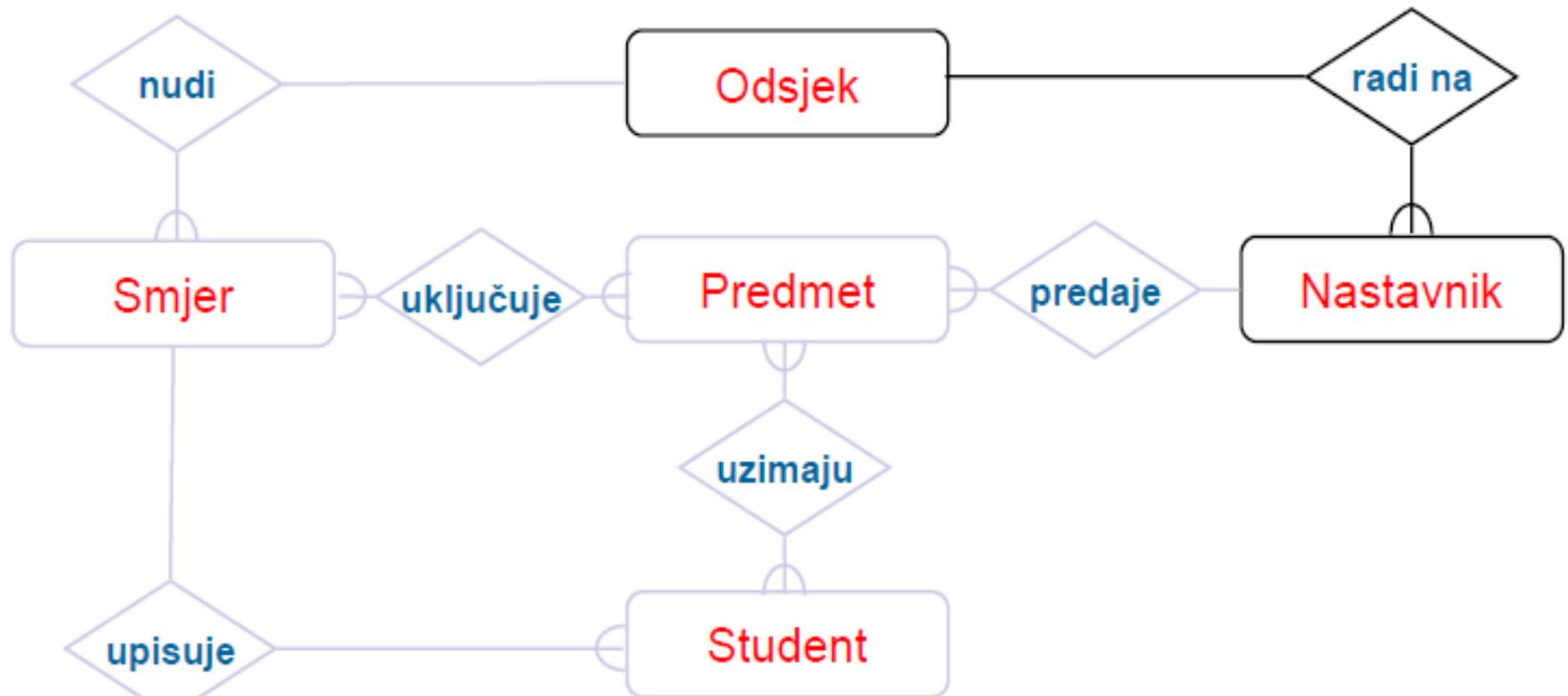
Pravljenje E/R modela - Primer

- Svaki **predmet** **predaje** **nastavnik**



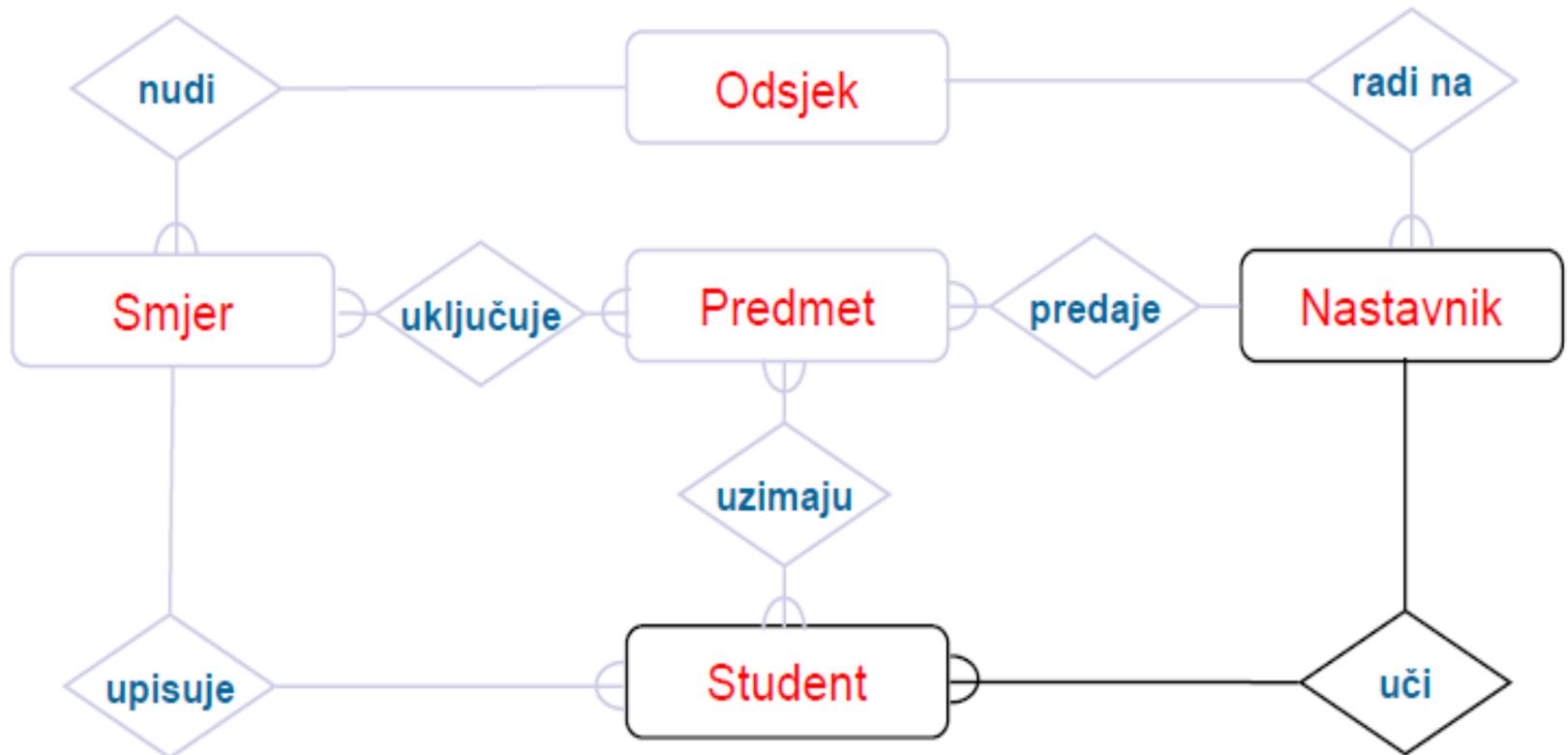
Pravljenje E/R modela - Primer

- **Nastavnik pripada** određenom **odseku**



Pravljenje E/R modela - Primer

- Svaki **nastavnik uči** grupu **studentata**



E/R dijagram - Primer

