



# DISTRIBUIRANA OBRADA I RELACIONE BAZE PODATAKA

---

mr Dušan Marković  
soba 237  
ponedeljak od 12-14h

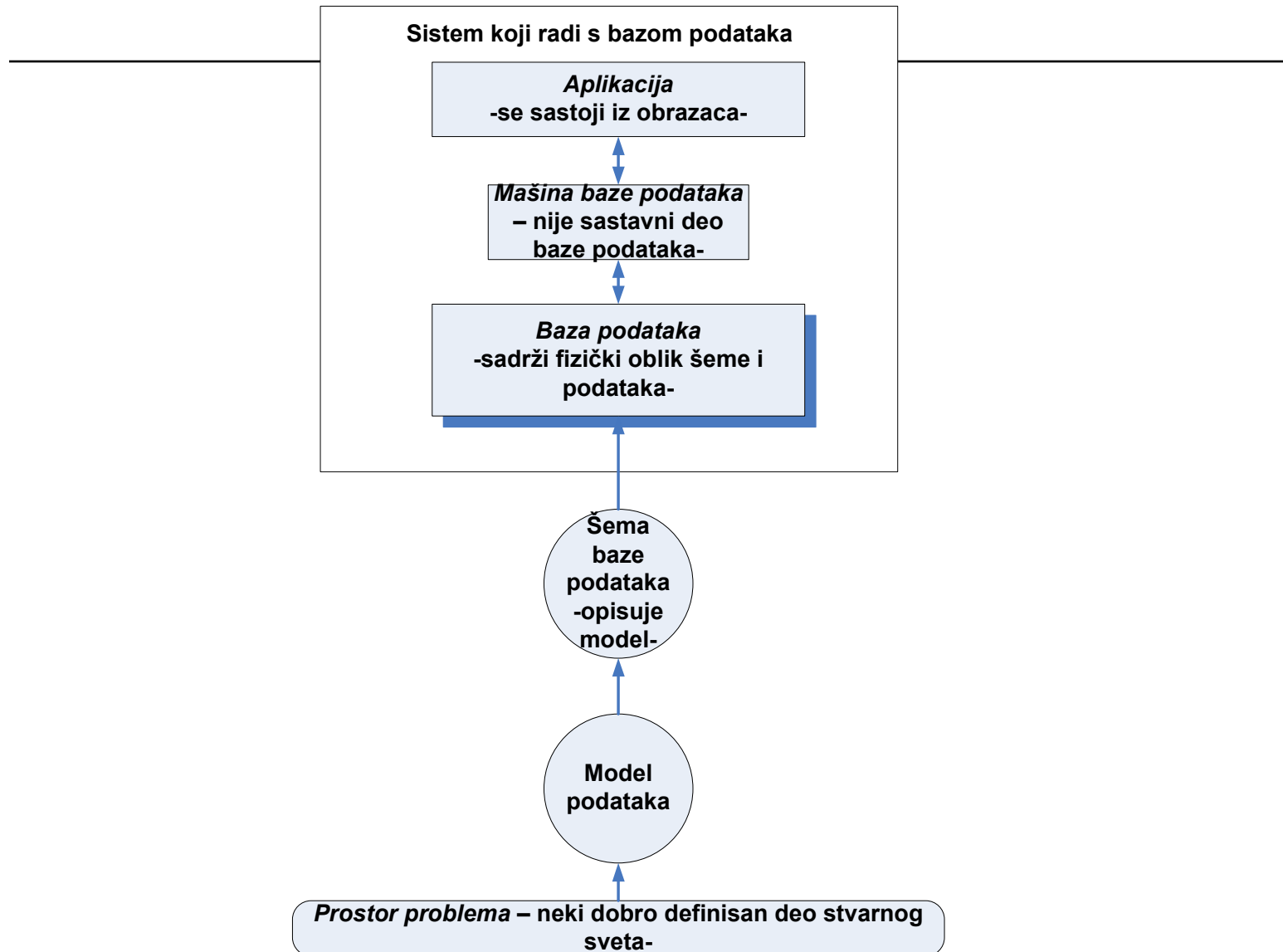


# ŠTA JE BAZA PODATA?

---

Izraz "baza podataka" koristi se za opisivanje svačega od obične grupe podataka, do složenog skupa alatki.

# TERMINOLOGIJA RELACIONIH BAZA PODATAK





# PROSTOR PROBLEMA (*problem space*)

---

**Po svojoj prirodi je zbrkan i složen.**

**Međutim za uspeh projekta je vrlo važno da sistem za koji projektujete bazu podataka bude ograničen na tačno definisani skup objekata i njihovih odnosa.**



# MODEL PODATAKA

*(data model)*

---

Model podataka obuhvata opis veza ili odnosa između pojedinih entiteta kao i ograničenja.

Model podataka ništa ne govori o fizičkoj strukturi sistema



# ŠEMA BAZE PODATAKA

*(database shema)*

---

To je preslikavanje pojmovnog modela u fizički oblik, koji se može realizovati.



# BAZA PODATAKA

---

Sadrži fizičke tabele u kojima se čuvaju podaci, upiti i uskladištene procedure.



# MAŠINA BAZE PODATAKA

*(database engine)*

---

Prepoznaje tabele, okidače i sličnih elemenata.

***Na ovom nivou ne morate da se bavite fizičkim oblikom baze podataka***





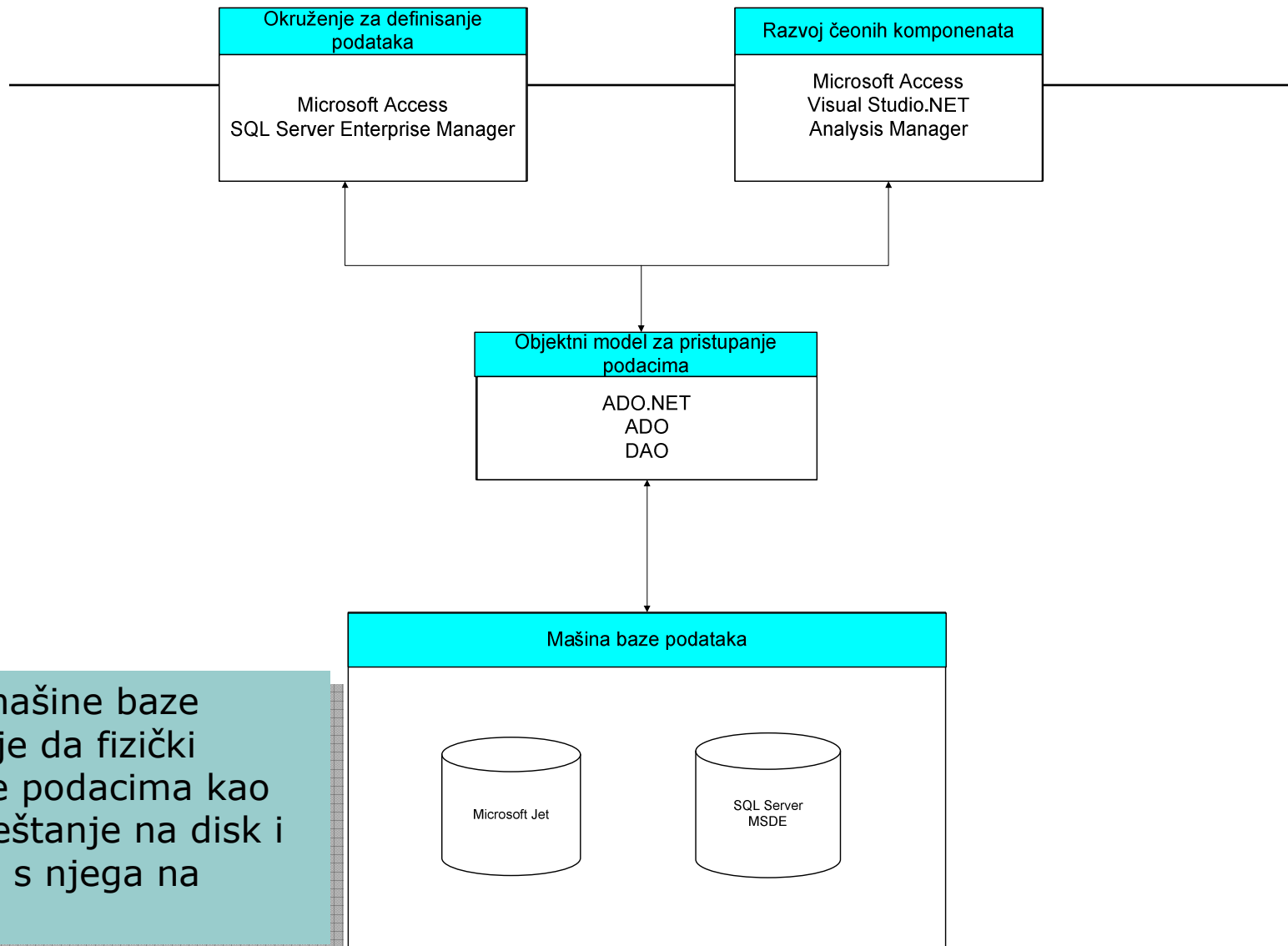
# APLIKACIJA

---

Sastoji se od obrazaca i izveštaja sa kojima radi korisnik

**Ovde spada i posrednički sloj ili IIS čija je namena da povezuje čeonu i pozadinsku komponentu sistema**

# ALATKE ZA RAD SA BAZAMA



Zadatak mašine baze podataka je da fizički manipuliše podacima kao što su smeštanje na disk i učitavanje s njega na zahtev



# MICROSOFT JET

---

Microsoft Jet je "stona" mašina baze podataka, namenjena sistemima koji se po veličini mogu svrstati u opseg od malih do srednjih



# SQL SERVER

---

SQL Server koristi klijent/server arhitekturu i namenjen je od srednjih do ogromnih sistema.



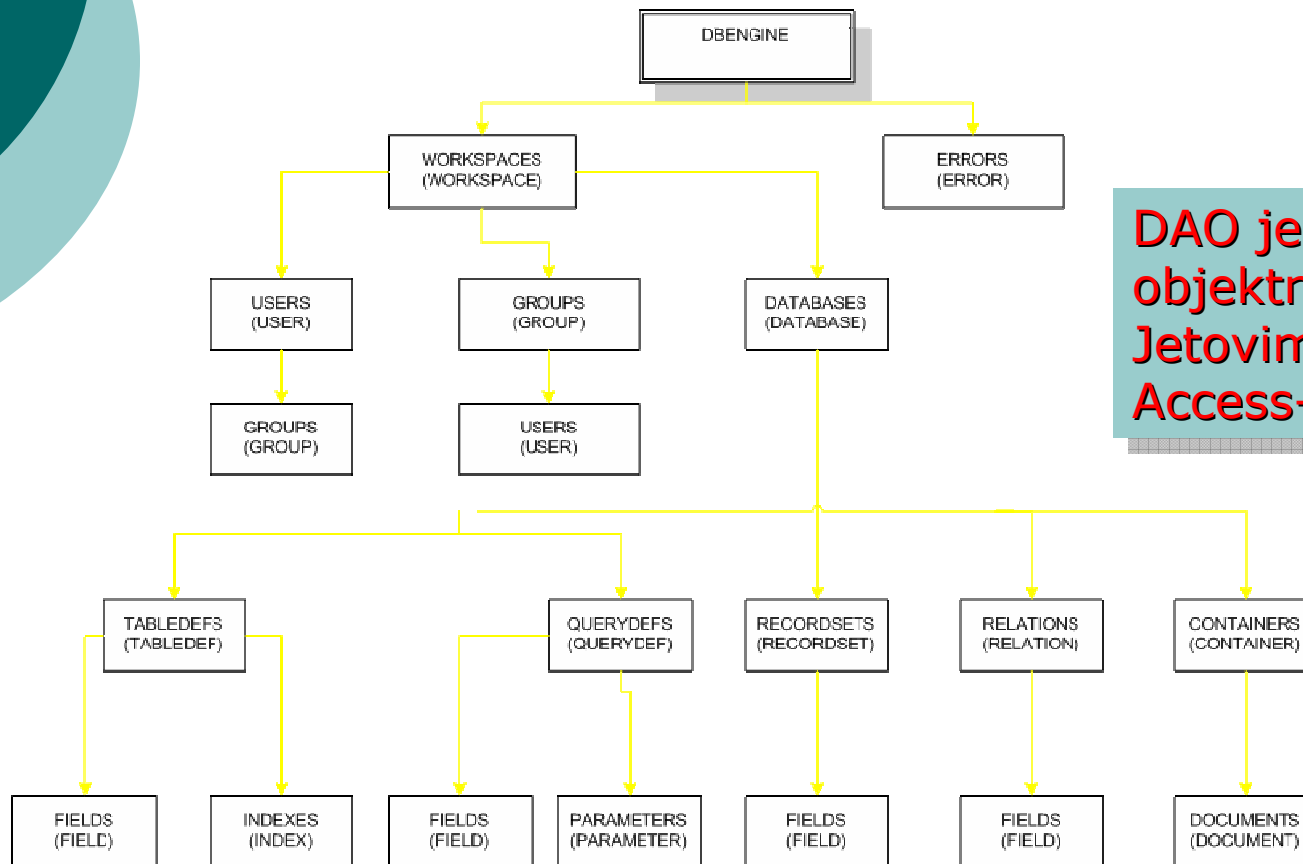
# Objektni modeli za pristupanje podacima

---

Microsoft (zasad) stavlja na raspolaganje tri objektna modela za pristupanje podacima.

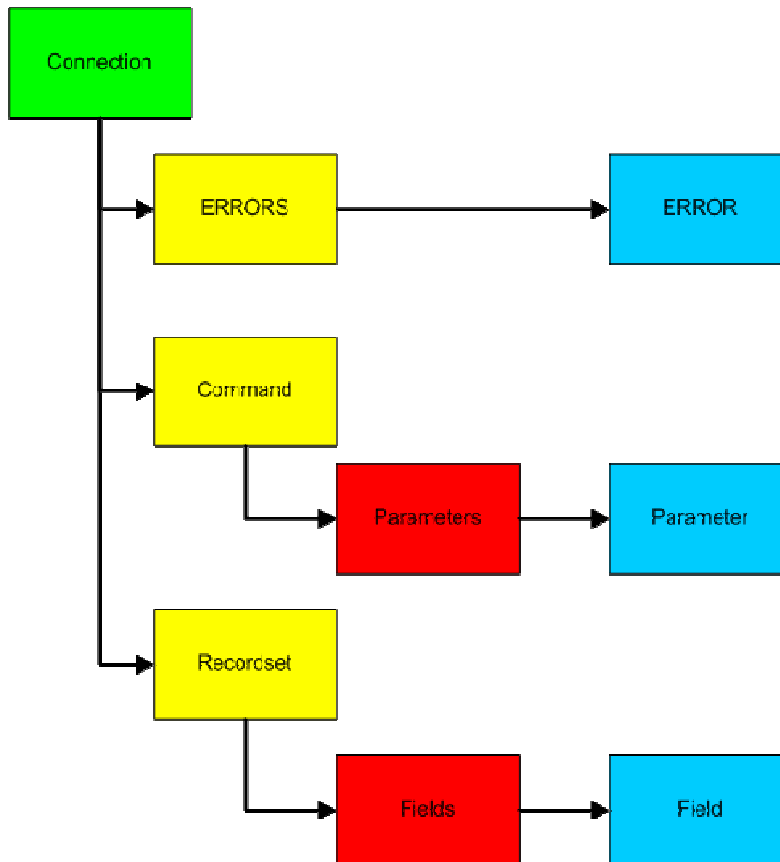
- Data Access Objects (DAO), koji postoji u dve varijante (DAO/Jet i DAO/ODBC Direct)
- Microsoft ActiveX Data Objects (ADO)
- ADO.NET

# DAO



**DAO je najefikasniji objektni model za rad sa Jetovim bazama podataka u Access-u**

# ADO



ADO je jednostavniji od DAO i donosi poboljšanje u vidu podrške za rad s nepovezanim hijerarhijskim skupovima podataka



## Okruženje za definisanje podataka

---

Ovo okruženje je Access i SQL Server Enterprise Manager i služe nam kao način za opisivanje struktuiranje podataka





# Razvoj čeone komponente aplikacije

---

Alat za korišćenje izrade obrazaca i izveštaja su Access i Visual Studio .NET



# Relacioni model

---

Relacioni model se bazira na grupi matematičkih principa izvedenih iz teorije skupova i predikatne logike

**Pravila relacionog modela definišu:**

- 1. oblik u kojem se podaci predstavljaju (struktura),**
- 2. način na koji se podaci štite (integritet podataka) i**
- 3. operacije koje se mogu izvršiti nad podacima (manipulisanje podacima)**



# Odlike relacionog sistema

---

- Svi podaci se konceptualno predstavljaju organizovani u redove i kolone; skup tako organizovanih podataka zove se **relacija** (relation)
- Sve vrednosti su skalarne. To znači da se na svakom mestu koje je određeno datim redom i kolonom, nalazi jedna i samo jedna vrednost
- Sve operacije obavljaju se nad celom relacijom, a rezultat je takođe cela relacija. Taj koncept je poznat kao **celovitost** (closure).

# Relaciona terminologija

Atributi			
ID	Task Name	Duration	Predecessors
1	Izbor delatnosti firme	1 day	
2	Istrazivanje trzista i konkurencije	3 days	1
3	Izbor lokacije	2 days	2
4	Izbor imena firme	1 day	3
5	Procena polaznih rezultata	1 day	4
6	Plan usluga	2 days	5
7	Plan kadrovanja	2 days	6
8	Plan poslovnog prostora	1 day	7
9	SWOT-analiza	3 days	8
10	Strategija razvoja	2 days	9
11	Ciljno trziste	1 day	10
12	Određivanje snabdevaca	2 days	11
13	Plan nabavke	2 days	12
14	Plan promocije	2 days	13
15	Plan cena	1 day	14
16	Plan prodaje	2 days	15
17	Finansijski plan	2 days	16
18	Registracija firme	6 days	17
19	Zakup lokala na 6 meseci	1 day	18
20	Adaptacija i opremanje lokala	16 days	19
21	Uplata reklame u Maxi katalogu za 6 meseci	1 days	19
22	Uplata reklame u Info press-u "Mirijevec" za 6 meseci	1 days	21SS
23	Uplata reklame u Halo oglasima za 3 meseca	1 day	22
24	Izrada reklamnog materijala: vizit karte, leci, upaljaci...	1 day	23SS
25	Izrada Web sajta	10 days	19
26	Placanje troskova hostovanja sajta za godinu dana	1 day	25
27	Sklapanje ugovora sa Internet dobavljačem	1 day	26SS
28	Kraj projekta	0 days	20

Torke (rows 1-4)

Zaglavlje (header row)

Telo (main body of the table)

## Formalna terminologija

Konceptualna relacija	Fizička tabela	MS Access tabela ili skup zapisa	SQL Server tabela ili skup zapisa
atributi	polje	polje	kolona
torka	zapis	zapis	red



# Model podataka

---

Model podataka je najapstraktniji nivo projektovanja baza podataka, što predstavlja konceptualni opis prostora problema.

Modeli podataka sastoje se od elemenata:

1. entiteta
2. atributa
3. domena i veze



# Entiteti (entity)

---

**Entitet je sve o čemu sistem  
treba da skladišti podatke**

Primeri entiteta:

1. Kupci
2. Prodavci
3. Studenti itd..



# Atributi

---

**Atributi, pobliže određuju entitete.**

Određivanje atributa koje ćete ugraditi u svoj model, morate doneti na osnovu značenja podataka i načina na koji će se oni koristiti



# Prva strategija određivanje atributa

---

**Krenite od rezultata i nemojte praviti složeniju strukturu nego što je zaista potrebno.**

Drugim rečima, treba naći odgovor na koja pitanja vaša baza podataka mora dati odgovor





# Druga strategija

---

## **Otkrijte izuzetke**

**Prvo - morate identifikovati sve izuzetke**  
**Drugo – sistem morate projektovati tako da**  
**obrađuje što veći broj izuzetaka ali da pri**  
**tome ne zbunjuje korisnika**

**Imajte u vidu da morate napraviti kompromis između**  
**fleksibilnosti i složenosti**



# Domen (domain)

---

Domen predstavlja skup svih prihvatljivih vrednosti koje atribut može imati



## Veze/odnosi između entiteta (relationship)

- Pod vezom ili odnosom između entiteta podrazumeva se njihova asocijacija
- Entiteti između kojih postoji veza ili odnos zovu se učesnici veze (participants)
- Broj učesnika određuje stepen veze



## Binarna veza

---

**Binarna veza predstavlja vezu između dva entiteta.**

*Specijalan slučaj binarne veze je entitet koju učestvuje u vezi sa samim sobom. To se često zove veza tipa sastavnice i najčešće se koristi za predstavljanje hijerahijskih struktura.*



# Vrsta veza između entiteta

---

- Jedan prema jedan
- Jedan prema više
- Više prema više



# Veza tipa jedan prema jedan

---

Veze tipa **jedan prema jedan** su retke, ali mogu biti korisne u nekim okolnostima



# Veza tipa jedan prema više

---

Veze tipa **jedan prema više** su verovatno najuobičajenija vrsta.



# Veza tipa više prema više

---

Veze tipa **više prema više** nisu neouobičajene i mogu se naći brojni primeri

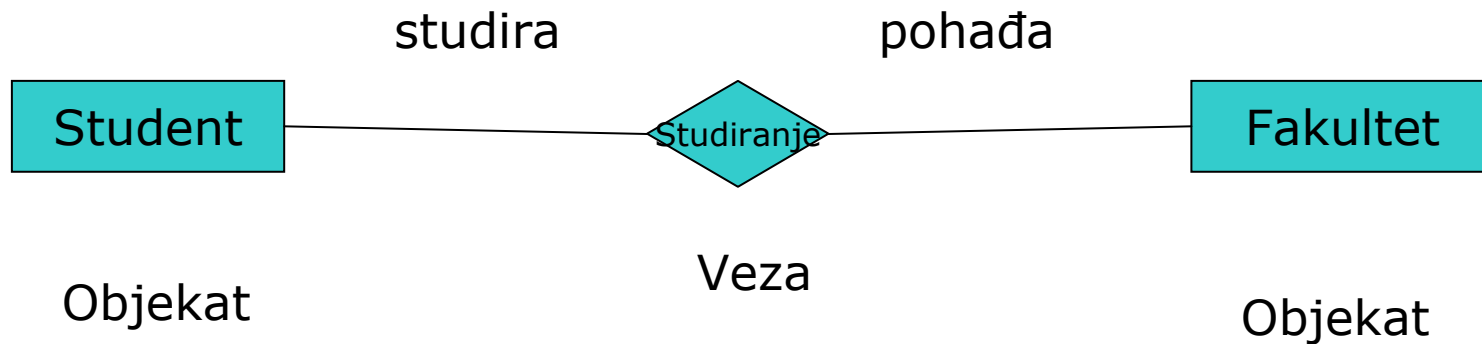


# Struktura modela objekti-veze MOV

U okviru MOV sistema, vrši se opisivanje objekata i njihovih veza

Šta su objekti?

Objekat u modelu predstavlja bilo neki fizički objekat ili koncept realnog sistema (konkretan proizvod , konkretnog radnika, vremenski trenutak ili period, smer studija i slično)



Veze u modelu opisuju način povezivanja (uzajamna dejstva) objekata



# Struktura modela objekti-veze MOV

---

U MOV-u se direktno predstavljaju samo binarne veze, veze između dva objekata

Svaki tip binarne veze tipova objekata E1 i E2 definiše dva tipa preslikavanja  $E1 \rightarrow E2$  i inverzno preslikavanje,  $E2 \rightarrow E1$

U prethodnom primeru:

1. Studira: Student  $\rightarrow$  Fakultet (Student studira fakultet)
2. Pohađa: Fakultet  $\rightarrow$  Student (Fakultet pohađa studenti)



# Konvencije za koncept veza

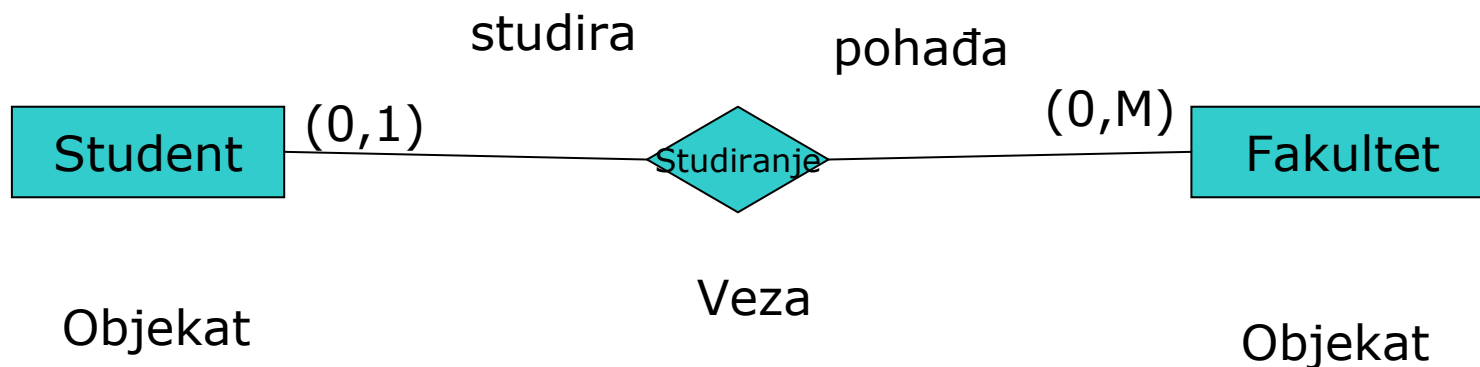
---

- Nazivi veza se uvek zadaju
- Nazivi preslikavanja se obavezno zadaju u rekurzivnim vezama (binarna veza nad jednom klasom)
- Nazivi preslikavanja u ostalim vezama su opciono i zadaju se samo kada je neophodno istaći ulogu objekata u vezi, odnosno kada se uloga oba objekta u vezi ne može dedukovati iz naziva veza.

# Kardinalnost veza

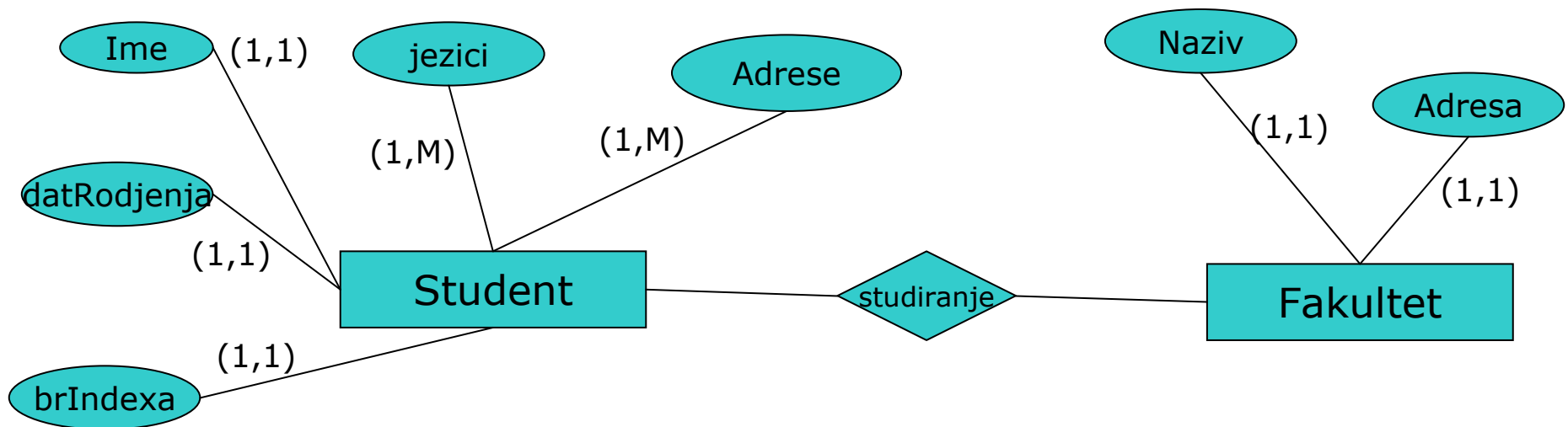
---

- Kardinalnost veza  $E1 \rightarrow E2$  definiše se parom  $(DG, GG)$
- DG – donja granica, daje najmanji mogući
- GG – gornja granica, daje najveći mogući broj pojavljivanja objekta  $E2$   $DG \leq GG$



# Atribut i domen

- Objekti se opisuju preko svojih atributa
- Atributi uzimaju vrednost iz skupa mogućih vrednosti, koji se nazivaju domeni





# Atributi i domeni

---

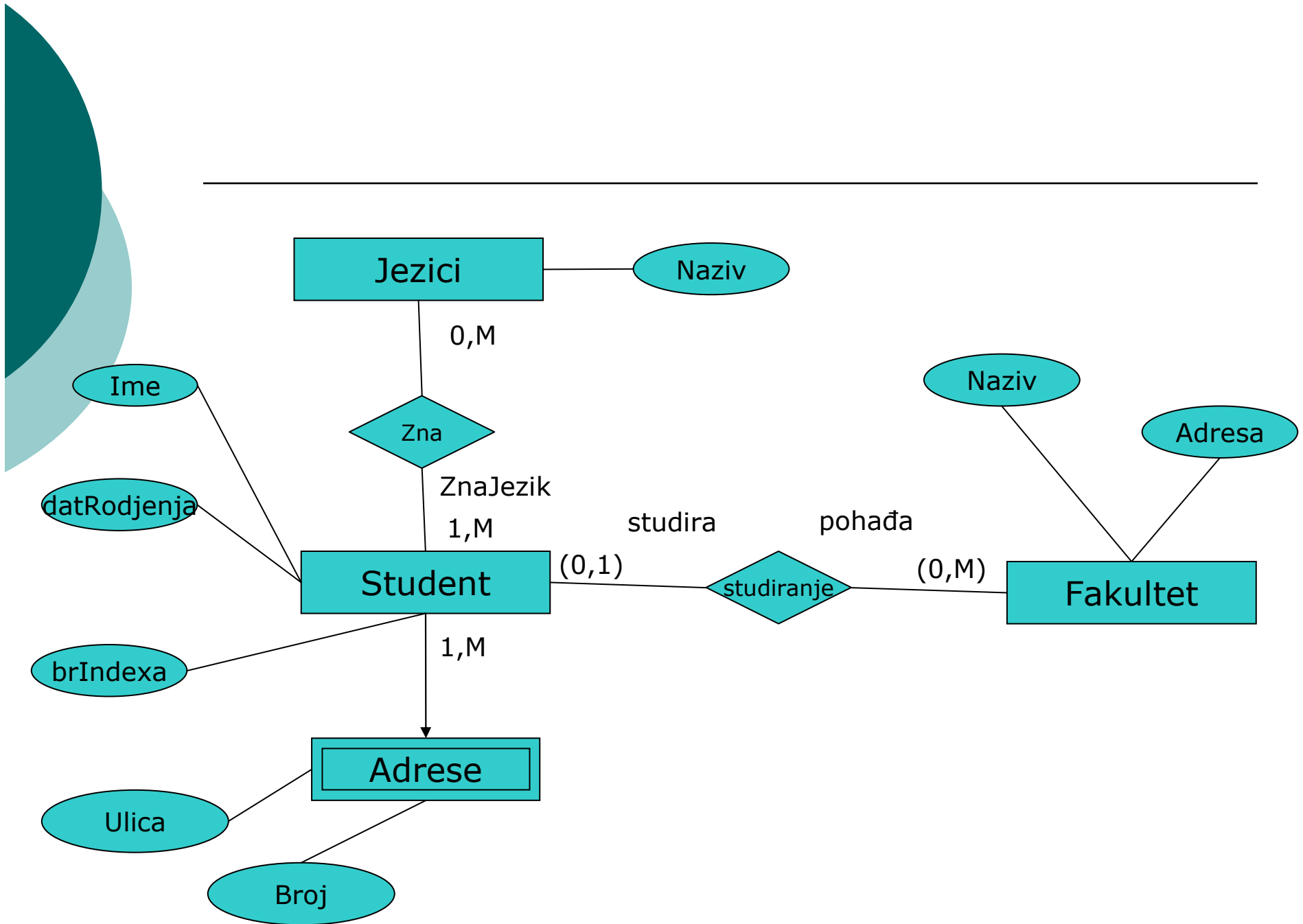
- *Jednoznačni* atributi objekta su atributi sa kardinalnošću  $DG=1$  i  $GG=1$  nazivaju se *indetifikatori objekta*
- *Višeznačni atributi* su atributi sa  $GG=M$



# Atributi i domeni

---

- Ako domen višeznačnog atributa ima unapred zadat, semantički skup vrednosti, tada se on modelira kao objekat.
- Ako domen višeznačnog atributa nema unapred zadat semantički značajan skup vrednosti, tada se predstavlja preko novog koncepta (***identifikaciono zavisnog slabog objekta***).







# Slabi objekat

---

**Slabi objekat u modelu ne može da postoji (egzistencijalno je zavisian) i ne može da se identifikuje bez veze sa njemu nadređenim objektom**



# Ograničenja

---

Jezik za definisanje ograničenja može da se bazira na konceptu **primitivnih i složenih ograničenja**

**Primitivna** ograničenja se konstruišu preko sledećih operatora

- $\theta$  konstanta ( $>, <, \leq, \geq, =, \neq$ )
- *Between* konstanta, gde su konstante vrednosti iz datog domena
- *In*(lista vrednosti), gde se lista formira od konstanti iz odgovarajućeg domena.
- *NotNull*, kada dato polje ne može da dobije "nula vrednost", odnosno mora uvek da ima vrednost

**Složena** ograničenja se formiraju od primitivnih ili drugih složenih ograničenja vezujući ih logičkim operatorima *And*, *Or* i *Not*

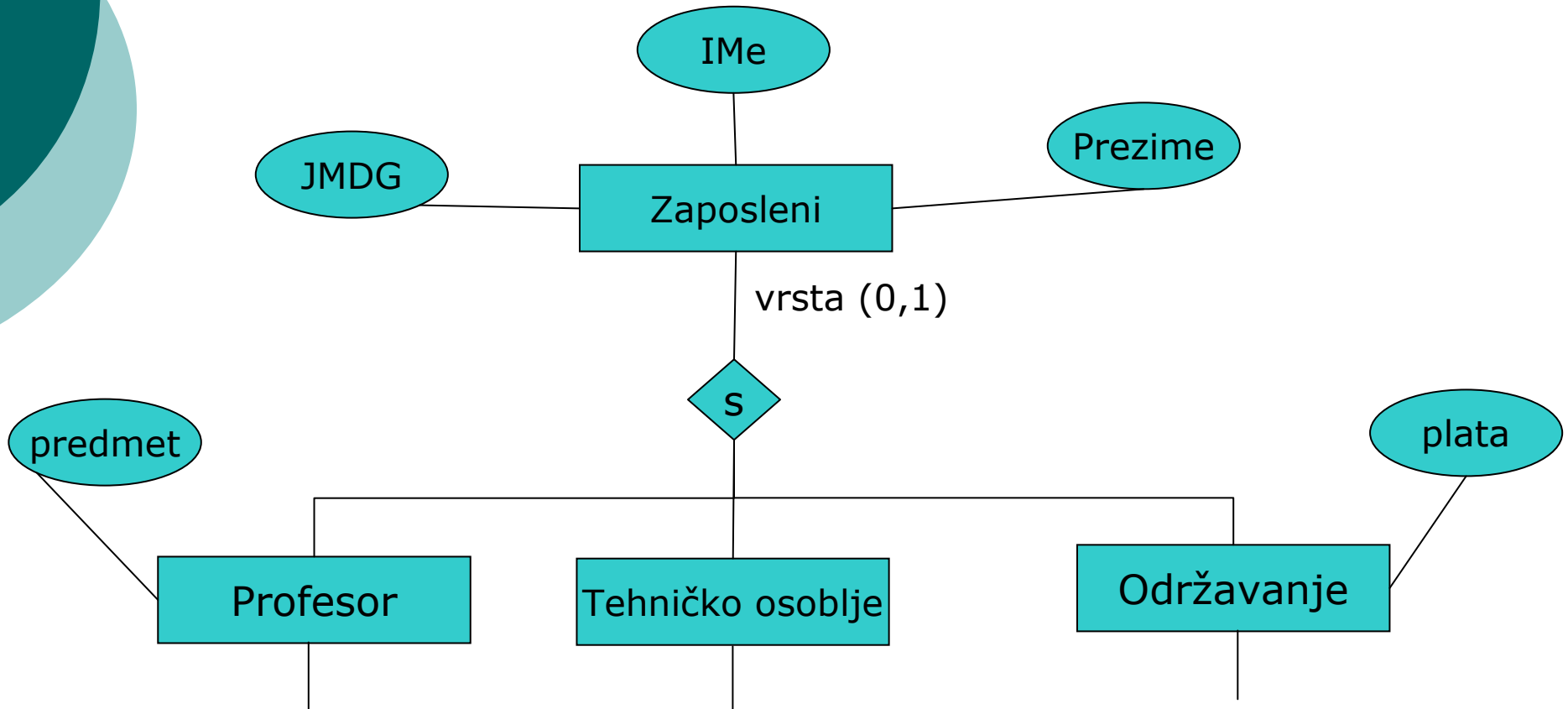


# Generalizacija i specijalizacija

---

- **Generalizacija** je apstrakcija u kojoj se skup sličnih tipova objekata tretira kao generički tip (nadtip). Slični tipovi su tipovi koji imaju neka zajednička svojstva ili veze.
- **Specijalizacija** je inverzni postupak u kome se za neki tip objekata, definišu njegovi podtipovi, koji imaju neka njima specifična svojstva.

# Primer





# Označavanje generalizacije

---

Generalizacija, odnosno specijalizacija se u MOV predstavljaju specijalnom oznakom “**S**” (potiče od Subtype) vezom, koja se često naziva i **ISA**



# Ekskluzivna specijalizacija

---

Ekskluzivna specijalizacija je da kada se jedno pojavljivanje tipa može specijalizovati u samo jedan podtip.  
 $GG=1$



# Neekskluzivna specijalizacija

---

Kada se jedno pojavljivanje tipa može specijalizovati u pojavljivanju različitih podtipova  $GG > 1$



# Obavezna specijalizacija

---

Obavezna specijalizacija je kada pojavljivanje tipa mora specijalizovati neki podtip (unija podtipova je jednaka nadtipu)  $DG=1$





# Neobavezna specijalizacija

---

Neobavezna specijalizacija je kada je  $DG=0$



# Agregacija i dekompozicija

---

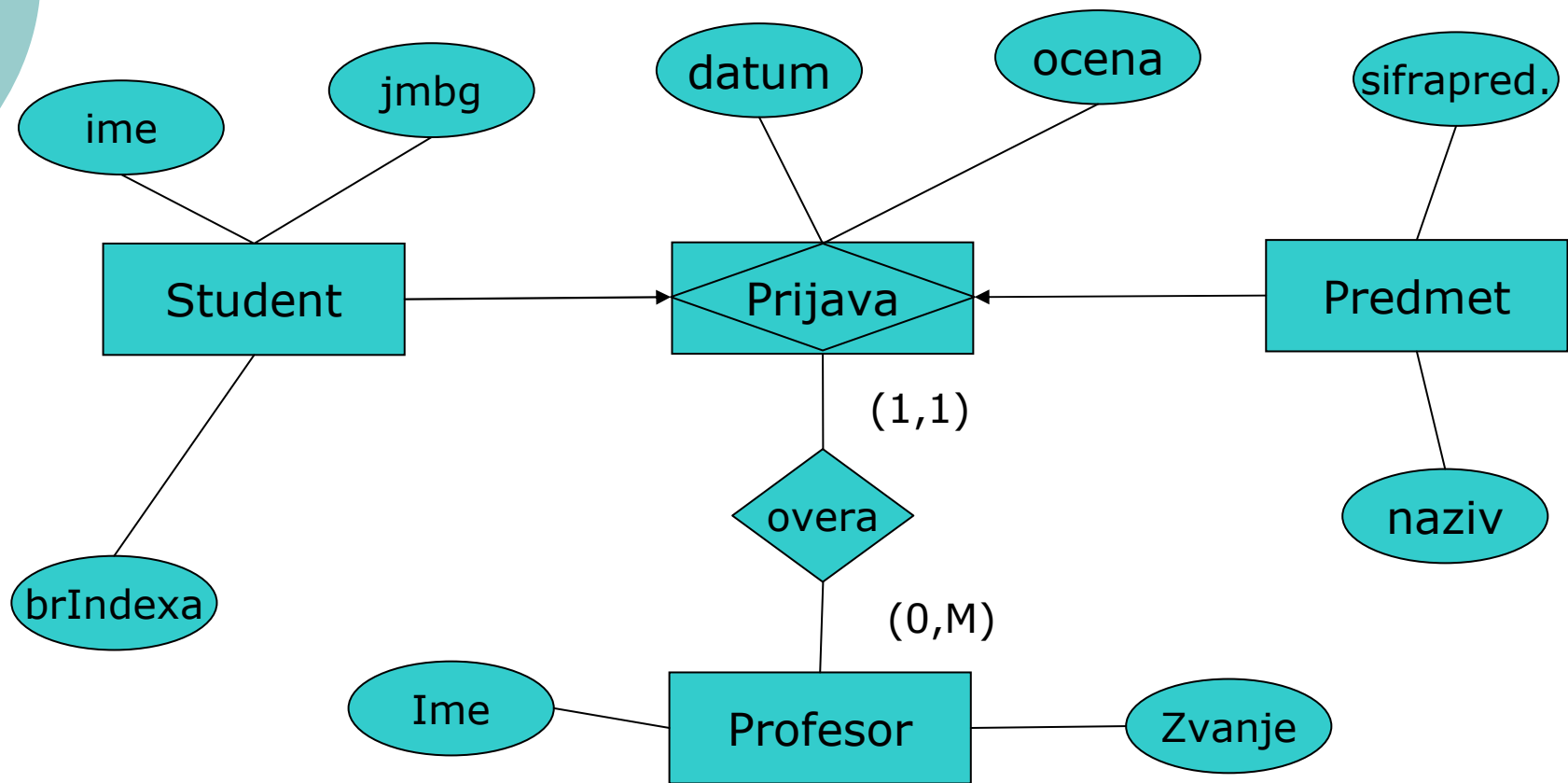
- **Agregacija** je apstrakcija u kojoj se skup povezanih objekata tretira kao jedinstveni objekat na višem nivou apstrakcije. Zbog toga što istovremeno predstavlja i jedinstveni objekat i vezu objekata koje ga čine agregacija se često zove i **mešovit tip objekata-veza**
- **Dekompozicija** je inverzna agregaciji



# Specifičnost agregacije

---

- Agregacija je egzistencijalno zavisna od svojih komponenti
- Agregirani objekat se razlikuje od ostalih objekata u sistemu po tome što nema svoj sopstveni identifikator, već ga identifikuju objekti koje on agregira.

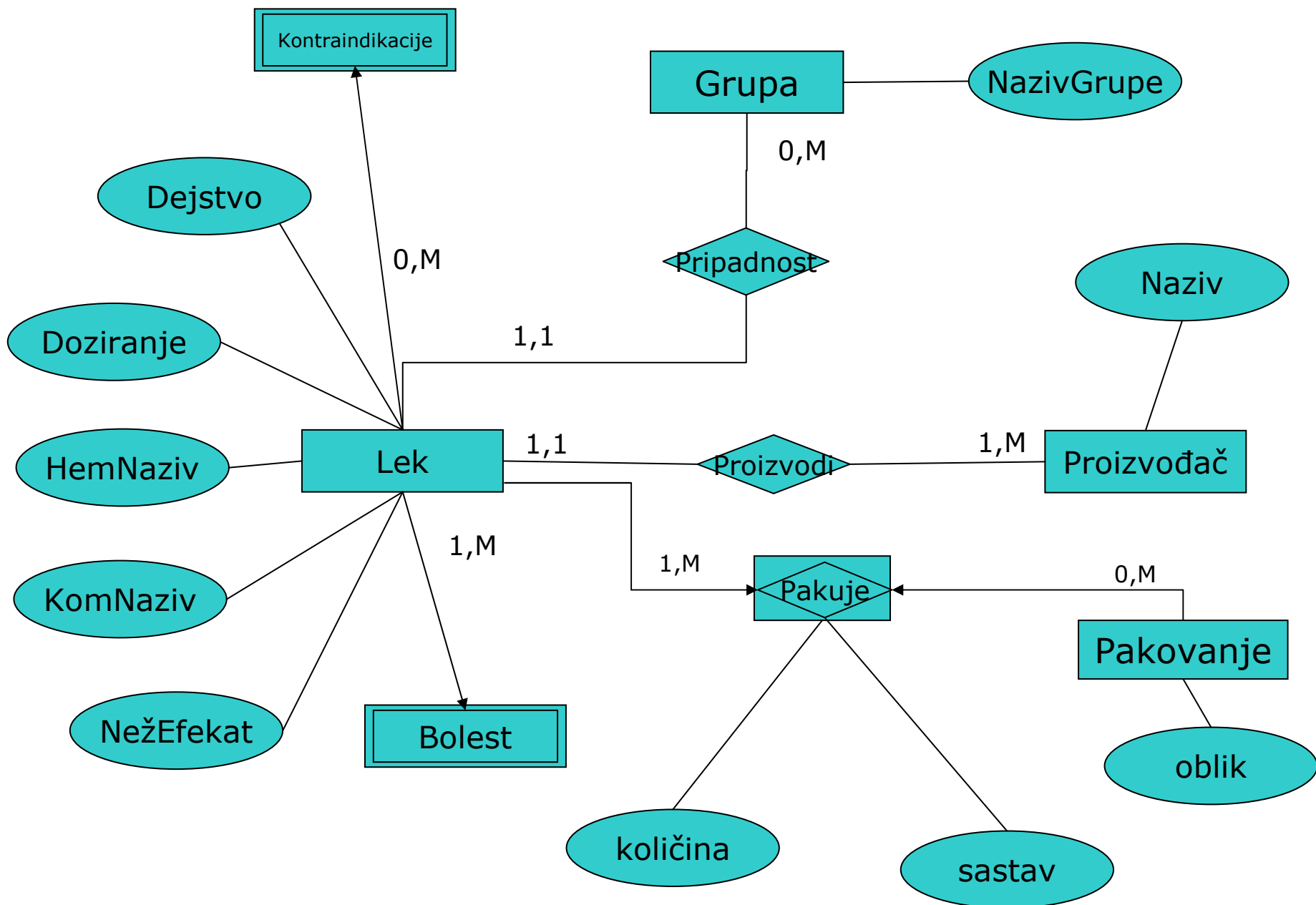




## Primer

---

Potrebno je voditi evidenciju o svim lekovima sa sledećim osnovnim atributima. Komercijalni naziv, Hemijski naziv, Doziranje, Dejstvo, Neželjeni efekti. Svaki lek se koristi za lečenje najmanje jedne vrste bolesti. S druge strane, za svaki lek potrebno je dati kontraindikacije kojih može biti više. Svaki lek pripada samo jednoj primarnoj grupi lekova (antibiotici, analgetici, antipiretici). Lek proizvodi jedan i samo jedan proizvođač. Lek se pakuje u više oblika (tableta, sirup, injekcijaprasak. Za svaku vrstu pakovanja leka potrebno je voditi evidenciju o količini i sastavu.





## Primer za vežbu

---

Nacrtati dijagram objekti-veze za video klub. Potrebno je obezbediti pretraživanje po nazivu filma, žanru, režiseru, glumcima, scenaristi i muzici. Voditi evidenciju o članu video kluba (osnovni atributi: JMBG, Ime, Prezime, Status). Video klub poseduje više kasete (kopija) istog filma. Potrebno je obezbediti praćenje istorijata izdavanja i vraćanja svake kasete. Jedan član kluba može dnevno zadužiti više kasete.



# Ograničenja

---

MOV najveću pažnju posvećuje strukturi modela, ograničenja, operacije i na dinamička pravila integriteta (ali samo ovlaš)

Ograničenja se klasifikuju na sledeće klase:

- **Strukturna ograničenja** – To su jezički iskazi koji su bitni samo u onim delovima u kojima je moguće da operacije ažuriranja baze naruše strukturu modela
- **Vrednosna ograničenja**, definišu dozvoljene vrednosti atributa i dozvoljene promene ovih vrednosti.





# Operacije

---

- Ubacivanje (Insert)
- Izbacivanje (Delete)
- Ažuriranje (Update)



# Dinamička pravila integriteta

---

**Dinamičko pravilo integriteta** čini trojka  $\langle \textit{Ograničenje}, \textit{Operacija}, \textit{Akcija} \rangle$  preko koga se iskazuje koje se akcija preduzima kada neka operacija naruši definisano ograničenje.

Postoje :

1. Prosta ograničenja
2. Složena vrednosna pravila integriteta



# Prosta ograničenja

---

Prostim ograničenjima nazivamo ograničenjima na pojedinačnim vrednostima atributa koji mogu da se naruše prilikom operacija ubacivanja i ažuriranja. Zato se formira tabela ATRIBUT, DOMEN, OGRANIČENJE, AKCIJA

Atribut	Domen	Ograničenje	Akcija
Starost	tinyint	Between 15,65	Poruka: "Starost van opsega"
NazivJ	string	In (Ruski, Engleski, Nemački, Španski)	Poruka: "Jezik ne postoji"



# Složena vrednosna pravila integriteta

---

*Uobičajeno je da je akcija koja se definiše jedinstvena za sva moguća narušavanja posmatranog ograničenja*



# Strukturalna dinamička pravila integriteta i fizički MOV

---

- Strukturalna dinamička pravila integriteta, nad MOV strukturalnom su veoma složena. Zato je potrebno da se MOV transformišu u FMOV (fizički model objekat veze).
- U FMOV-u ne postoje veze u kojima oba preslikavanja imaju gornju granicu kardinalnosti  $M$  a takođe i donju granicu kardinalnosti  $0$ . Obe ove veze se transformišu u agregaciju.



# Strukturna dinamička pravila integriteta i fizički MOV

---

U strukturnim dinamičkim pravilima integriteta, definišu se akcije koje se preduzimaju prilikom narušavanja kardinalnosti preslikavanja i to preko sledećih ključnih reči:

- **Restrict**
- **Cascade**
- **SetNull**
- **SetDefault**



# Restrict

---

Operacija se ne izvršava ako narušava odgovarajući strukturni integritet. Npr. nedozvoljava se brisanje nekog pojavljivanja ako postoji njegov preslikani deo.



# Cascade

---

Operacija se prenosi na objekat kodomen da bi se zadovoljio strukturni model.





# SetNull

---

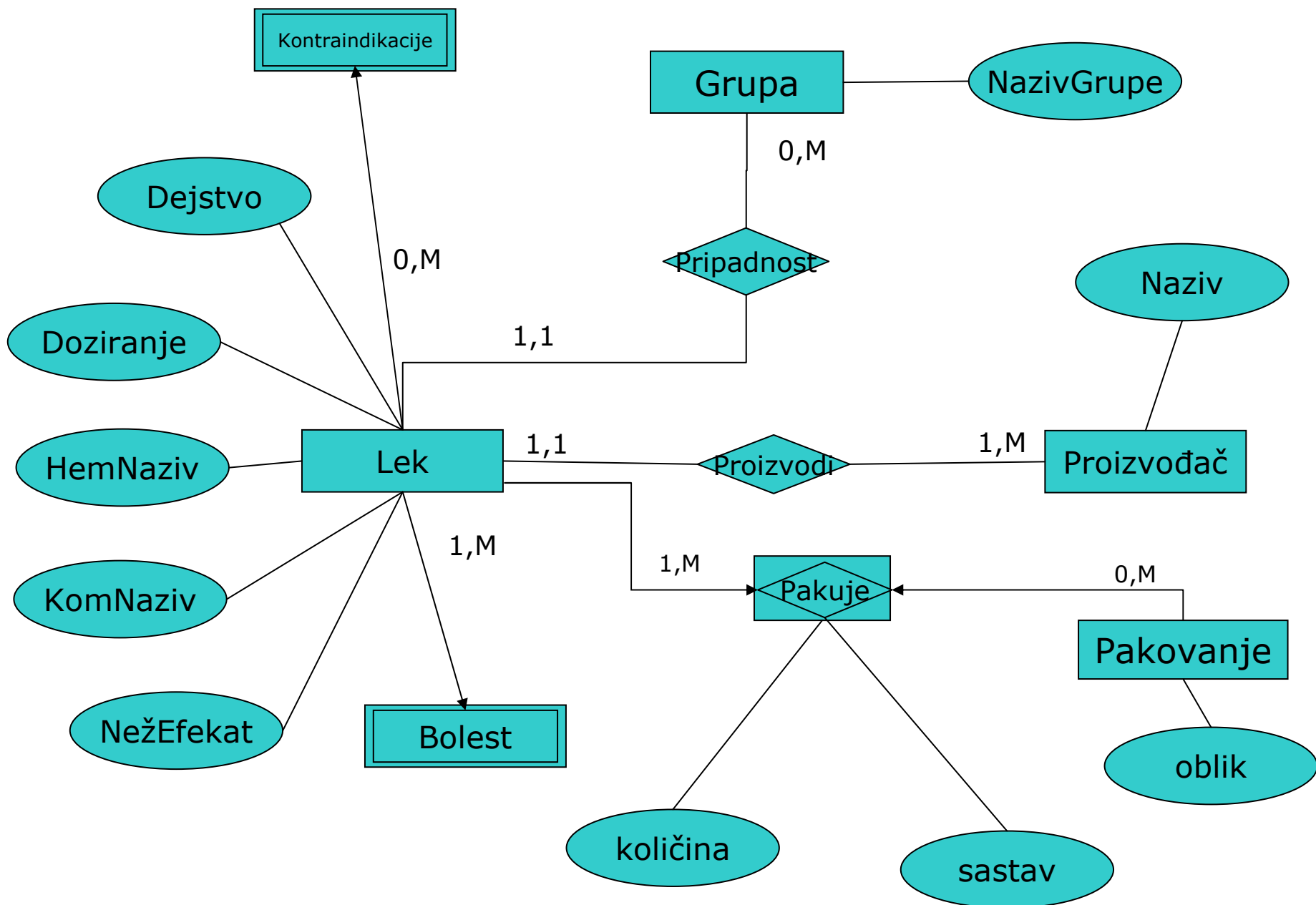
Uspostavlja se veza sa "nepoznatim objektom"  
objektom ("nula objektom")



# SetDefault

---

Uspostavlja se veza sa "default" objektom kodomena, pod pretpostavkom da je "default" pojavljivanje klase kodomena definisano.



<b>Operacije</b>	<b>Preslikavanje</b>	<b>Opcija</b>
Insert Lek	Lek -->Grupa	SetDefault
	Lek -->Proizvođač	SetDefault
	Lek -->Pakuje	Cascade
	Lek -->Bolest	Cascade
	Lek -->Kontaindikacije	Cascade
Delete Lek	Lek -->Grupa	-
	Lek -->Proizvođač	-
	Lek -->Pakuje	Cascade
	Lek -->Bolest	Cascade
	Lek -->Kontaindikacije	Cascade
Insert Grupa	Grupa -->Lek	--
Delete Grupa	Grupa -->Lek	Restrict
Insert Proizvođač	Proizvođač -->Lek	-
Delete Proizvođač	Proizvođač -->Lek	Cascade
Insert Pakovanje	Pakovanje -->Pakuje	-
Delete Pakovanje	Pakovanje -->Pakuje	Restrict
Insert Bolest	Bolest -->Lek	Restrict
Delete Bolest	Bolest -->Lek	-
Insert Kontraindikacije	Kontraindikacije -->Lek	Restrict
Delete Kontraindikacije	Kontraindikacije -->Lek	-



# IDEF1X standard za modelovanje podataka

---

U IDEF1X definišu se dve vrste klasa

- Klasa nezavisnih (jakih) objekata
- Klasa zavisnih (slabih) objekata



# Klasa nezavisnih (jakih) objekata

---

To su objekti koji se mogu identifikovati i mogu da postoje bez veze sa drugim objektima u sistemu i nisu ni slabi, ni agregacija ni podtip.

## Klasa nezavisnih (jakih) objekata

**Deo za atribute primarnog ključa**

Deo za ostale atribute



# Klasa zavisnih (slabih) objekata

---

To su objekti koji se ne mogu identifikovati niti postojati u modelu bez veze sa njima "nadređenim objektima". Klasa zavisnih objekata u IDEF1X verziji odgovaraju klase slabih objekata, agregacije i podtipovi u verziji MOV

## Klasa zavisnih (slabih) objekata

Atributi primarnog ključa

Ostali atributi



# Kategorije klase objekta

---

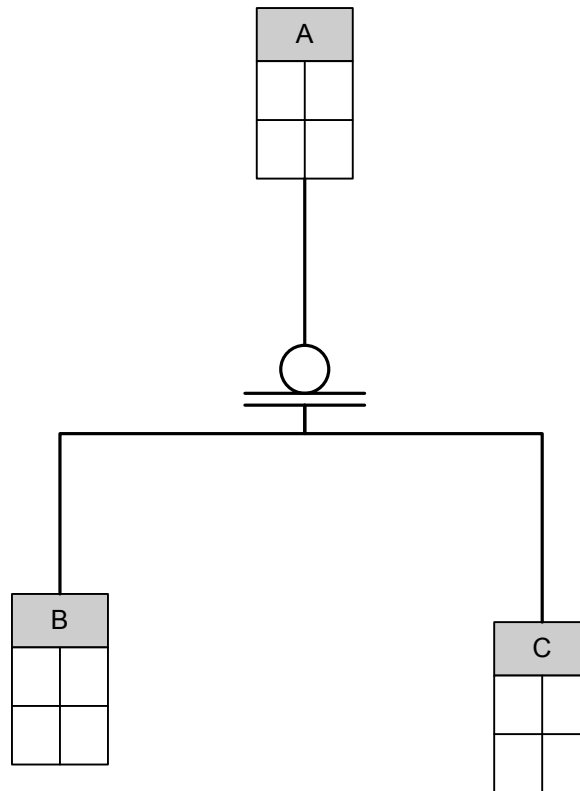
Jedna kategorija neke klase objekata predstavlja podskup pojavljivanja njoj nadređene generičke klase i nasleđuje sve atribute i veze svoje generičke klase. On je ekvivalentan pojmu *generalizacije, odnosno specijalizacije*.

- Kompletna kategorija
- Nekompletna kategorija



# Kompletne kategorije

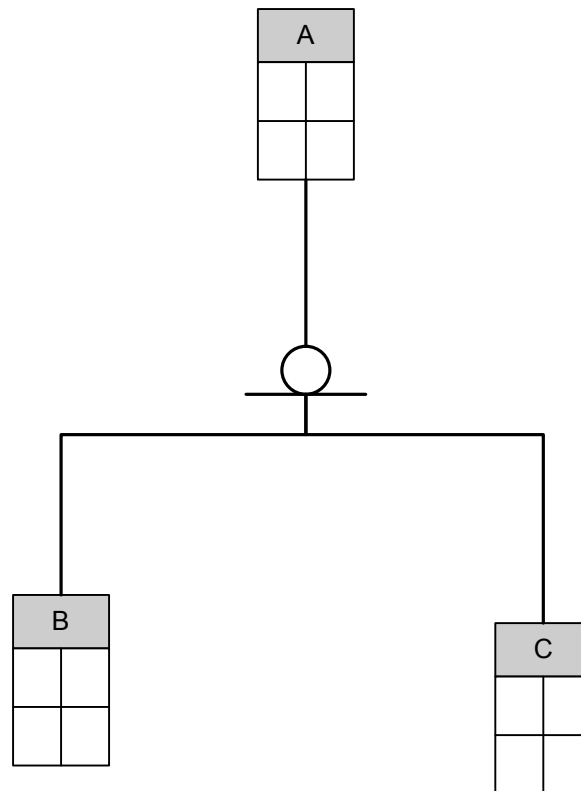
---



Odgovara ranije definisanom pojmu obavezne specijalizacije

# Nekompletne kategorije

---



Odgovara ranije definisanom pojmu neobavezne specijalizacije



# Ključevi

---

Po definiciji svaki pripadnik skupa  $\pi$  torki je jedinstven.

Posledično je da u svakoj relaciji mora postojati određena kombinacija atributa koja na nedvosmislen način identifikuje svaku torku.

Taj skup ili kombinacija jednog ili više atributa zove se kandidat za ključ (*candidate key*)



# Kandidat za ključ

---

- Kandidat za ključ može se sastojati od samo jednog atributa (**prost ključ** *simple key*)
- Ili od više njih (**složeni ključ** *composite key*)




# Uslove koje mora da ispuni kandidat za ključ

---

Osim jedinstvenosti koje mora ključ da obezbedi, neophodno je da ključ ne može da se razbije na prostije delove; pa iz toga sledi da skup atributa relacije ne mora obavezno biti kandidat za ključ

Kada je jedini mogući kandidat suviše nezgrapan – na primer, zato što zahteva previše atributa ili je preveliki – možete dodati polje s veštačkim ključevima čiji je tip podataka takav da sistem automatski generiše vrednosti ključa.

Polja tog tipa, u Microsoftovom Jetu zove se *AutoNumber*, a u SQL Serveru *Identity*



Ponekad – mada ne često – događa se da relacija ima više od jednog kandidata za ključeve.

U takvim slučajevima, jedan od kandidata se određuje kao **primarni ključ** (*primary key*), a drugi kandidati se smatraju **alternativnim ključevima** (*alternate keys*).



# Primarni ključ

---

## Osnovne osobine primarnog ključa

1. Mora da jedinstveno identifikuje svaku entitetsku instancu.
2. Ni jedan deo primarnog ključa ne može biti **NULL** ili prazan (**empty**) ili nedostajući (**missing**)
3. Ključ treba biti razumno mali
4. Treba izbegavati upotrebu "**intelligentnih ključeva**" (na primer gde struktura brojeva identifikuje grupisanje, lociranje, klasifikaciju, datume.





# Alternativni i inverzni ključevi

---

Kandidati za ključev koji nisu izabrani za primarne ključeve mogu se definisati kao alternativni ključ (*Akn*)

Alternativni ključ (*Akn*) predstavlja atribut ili grupu atributa koji jedinstveno identifikuje ntorke. Ali za razliku od primarnog ključa može imati i *NULL* vrednost



# Inverzni ključ

---

Atributi koji nemaju jednoznačnost a služe nam za brže pretraživanje, zovu se inverzni ključevi sa oznakom *IE*.

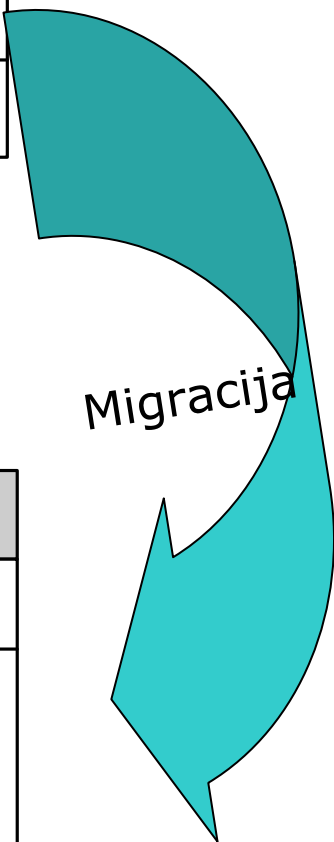
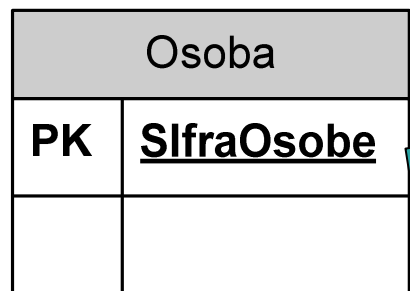


## Preneseni ključ

---

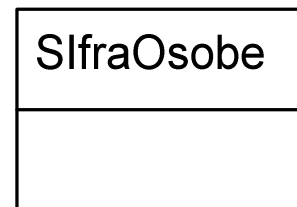
Preneseni ključ je kolekcija atributa koji u posmatranom entitetu nisu ključ, ali su zato ključ u nekom drugom entitetu.

**Preneseni ključ (*Foreign Key*) jeste atribut koji povezuje entitet "dete" sa entitetom "roditelj" i određuje se oznakom **FK****

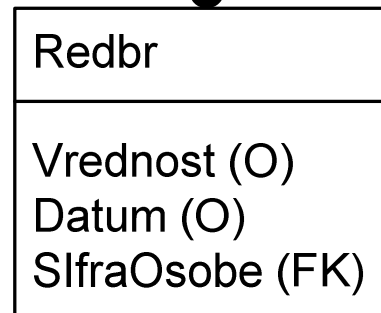


Relaciona notacija

Osoba

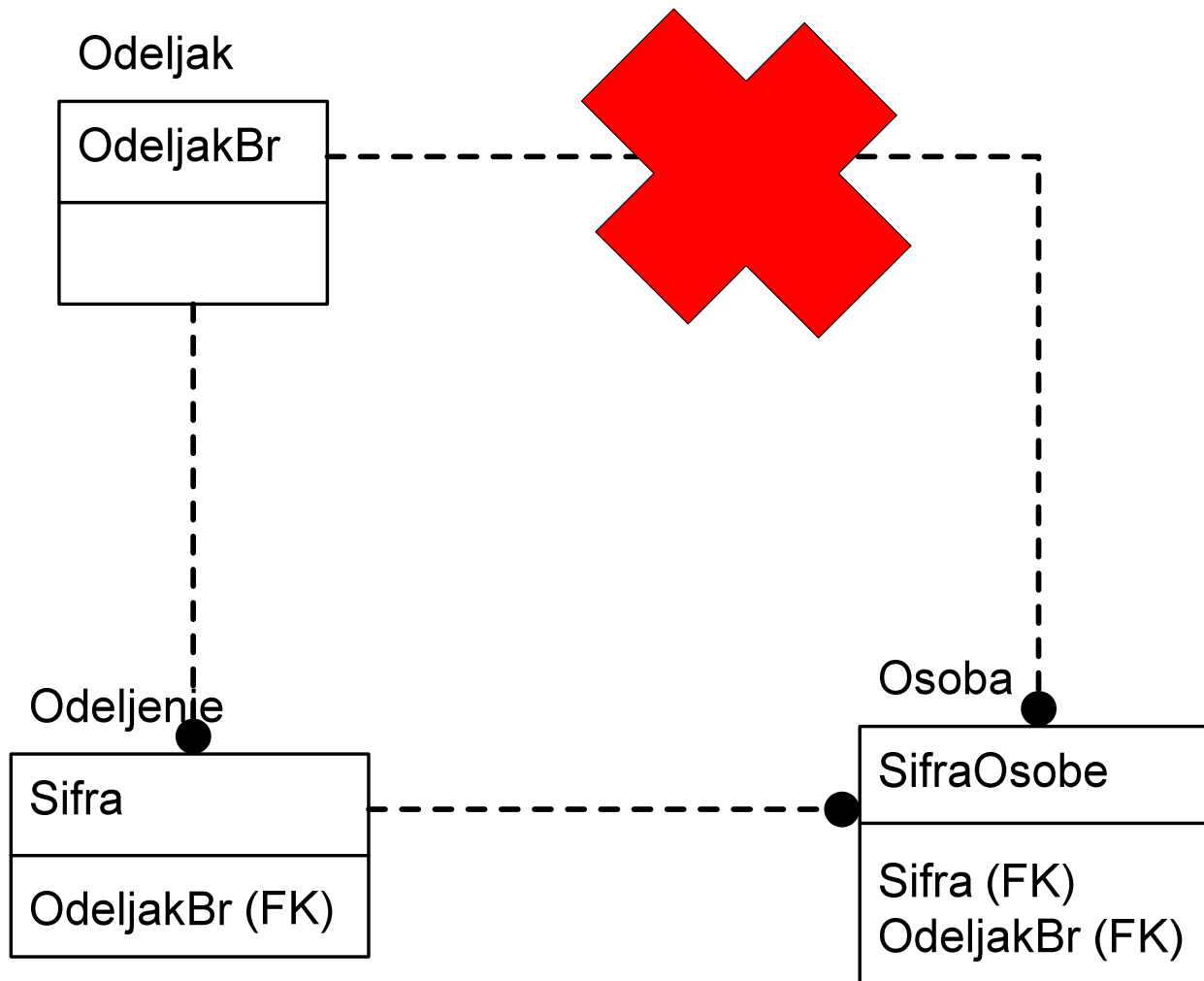


Isplata



IDEF1X - notacija

# Moguća greška





## Osobina prenesenog ključa

---

Preneseni ključevi mogu imati i drugo ime, što se definiše kao uloga atributa u entitetu. Ime uloge (*Rolename*) predstavlja novo ime za prenesene ključne attribute koji definišu ulogu koju tu atributi igraju u tom entitetu. Ime uloge definiše novi atribut, čije ime treba da opiše poslovno pravilo.



# Veze

---

Veza u IDEF1X metodologiji se prikazuje kao linija koja povezuje dva entiteta, sa tačkom na jednom kraju i glagolskom frazom napisanom duž te linije

Entitet od koga je uspostavljena veza zove se "roditelj" (parent), a entitet ka kome je uspostavljena veza zove "dete" (child)



# Tipovi veza

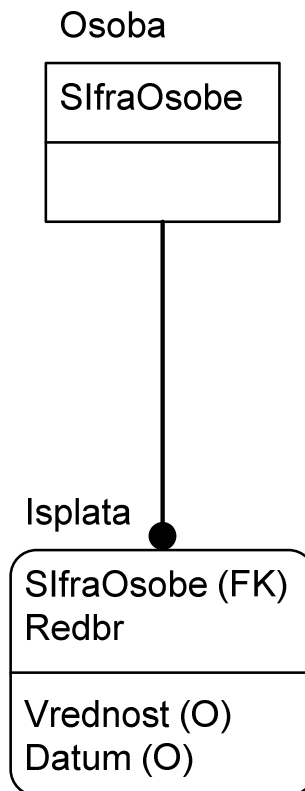
---

- Identifikujuća veza
- Neidentifikujuća veza
- Veza kategorije
- Neodređujuća veza



# Identifikujuća veza

Veza se zove *identifikujuća* zato što ključevi entiteta "roditelj" predstavljaju deo identiteta entiteta "dete".



Entitet "dete" ima složeni ključ



# Neidentifikujuća veza

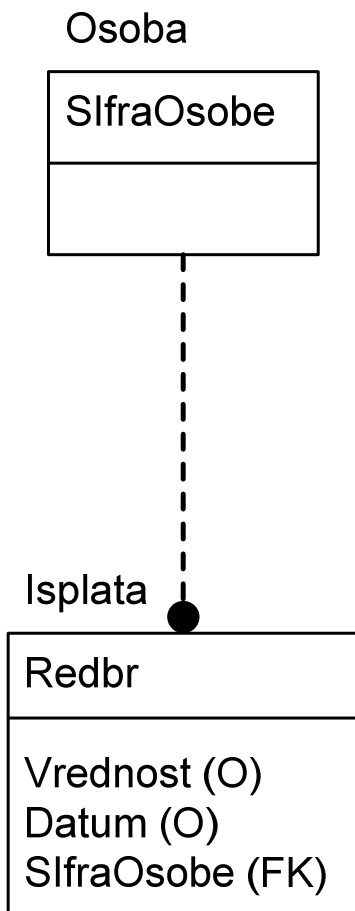
---

Ako se svaka entiteta "dete" može jedinstveno identifikovati, bez znanja veze sa primerkom entiteta "roditelj", onda se takva veza definiše kao **neidentifikujuća**

Neidentifikujuća ili slaba veza zavisi od načina definisanja ključeva od "roditelj" ka "detetu" na dva načina:

1. kao obavezna neidentifikujuća veza
2. kao neobavezna neidentifikujuća veza

# Obavezna neidentifikujuća veza



Obavezujuća veza (No Nulls ili Mandatory) iz perspektive "roditelj", onda je dete egzistencijalno zavisno od "roditelj". No Nulls ili Mandatory znači da je obavezan unos prenesenog ključa entiteta "roditelj" u entitet "dete"

# Neobavezna neidentifikujuća veza

---

Imena

id: int

ime: varchar(20)

prezime: varchar(20)

pol: varchar(10)

deca

id\_1: int

ime: varchar(20)

id: int

Ako je veza neobavezna (Nulls Allowed ili Optional), tada "dete" niti je egzistencijalno niti identifikaciono zavisno ali poštuje vezu.