

*Projektovanje informacionih sistema*

*Logičko projektovanje*

dr Rade Matić

# Logičko projektovanje

- **Logičko projektovanje** baze podataka se svodi na transformaciju konceptualnog modela u opis baze podataka u Jeziku za opis (DDL-u) konkretnog Sistema za upravljanje bazom podataka. Najčešće se ova transformacija vrši korišćenjem CASE alata u kome se konceptualni model razvija.
- **Mnogi autori objedinjavaju fazu konceptualnog modelovanja i logičkog projektovanja i ovako objedinjene aktivnosti nazivaju logičko projektovanje.** Ako bi se govorilo samo o projektovanju relacionih baza podataka, ovo objedinjenje bi imalo smisla. Međutim, ako se govori o projektovanju objektnih i objektno relacionih modela jasno se izdvaja faza izgradnje konceptualnog modela sistema.

# Logičko projektovanje

- Logičko projektovanje relacionih baza podataka se svodi na transformaciju već definisanog konceptualnog modela u opis baze podataka u datom relacionom SUBP (SQL server ili sl.).
- **Ako je konceptualni model izgrađen već u Relacionom modelu podataka postupkom normalizacije, nikakva transformacija nije potrebna.** Dovoljno je skup relacija i ograničenja konceptualnog modela opisati u DDL-u sistema za upravljanje bazom podataka u kome se baza implementira.
- Ovde se daje najčešće korišćeni pristup, transformaciju Modela objekti-veze u Relacioni model.

# Transformacija MOV-a u Relacioni model

Transformacija MOV u Relacioni model se obavlja na osnovu sledećih **opštih jednostavnih pravila**.

1. Svaki tip objekta iz MOV postaje relacija.
2. Atributi relacije su atributi odgovarajućeg tipa objekta i identifikator tipa prema kome posmatrani tip ima preslikavanje sa kardinalnošću (1,1).

# Transformacija MOV-a u Relacioni model

3. Primarni ključ dobijenih relacija je:

a.) za nezavisan tip objekta, atribut identifikator. Međutim, U relacionom modelu se za identifikator (primarni ključ) kreira novi atribut tzv. **negovoreći identifikatori (NI)**, koji je u praksi najčešće redni broj, čiji naziv nastaje spajanjem *NazivaTabele* i *ID*, i koji se automatski inkrementira (povećava) za određeni broj (npr. za 1).

Na primer, recimo da imamo relaciju Student, njegov naziv primarnog ključa (novog atributa) bi bio StudentID, čija se vrednosti najčešće automatski popunjava i povećava za 1 i to za svaki novi zapis (n-torku) u relaciji Student.

# Transformacija MOV-a u Relacioni model

b.) Za tipove objekata slabih objekata, potpuni identifikator nadređene klase proširene sa jednim ili više atributa slabe klase koji jedinstveno identifikuju jedno pojavljivanje slabog objekta za jedno pojavljivanje nadređenog. Ovde se takođe primenjuje pravilo kreiranja **NI**.

c.) Za podtip, identifikator nadtipa. Pravilo **NI**.

d.) Za agregaciju, jedan od identifikatora komponenti koja prema agregaciji ima preslikavanje sa gornjom granicom kardinalnosti jednakom jedan ili konkatenacija identifikatora komponenti koje prema agregaciji imaju preslikavanje sa gornjom granicom kardinalnosti M. Pravilo kreiranja **NI**.

# Transformacija MOV-a u Relacioni model

4. Spoljni ključevi relacija predstavljaju attribute koji su po Pravilu 1 preuzeti od susednih objekata.
5. Ograničenja definisana u MOV prevode se u odgovarajuća ograničenja u Relacionom SUBP.
6. Veze koje imaju oba preslikavanja sa gornjom granicom kardinalnosti većom od 1 i veze u kojima bar jedno preslikavanje ima donju granicu kardinalnosti  $DG=0$ , tretiraju se kao agregacije.

# **Transformacija MOV-a u Relacioni model**

Detaljnija pravila za prevođenje MOV u relacioni model:

**1. Pravila za objekte (tipove objekata)**

**2. Pravila za binarne veze**

**3. Pravila za unarne veze**



# Transformacija MOV-a u Relacioni model

## 1. Pravila za objekte (tipove objekata)

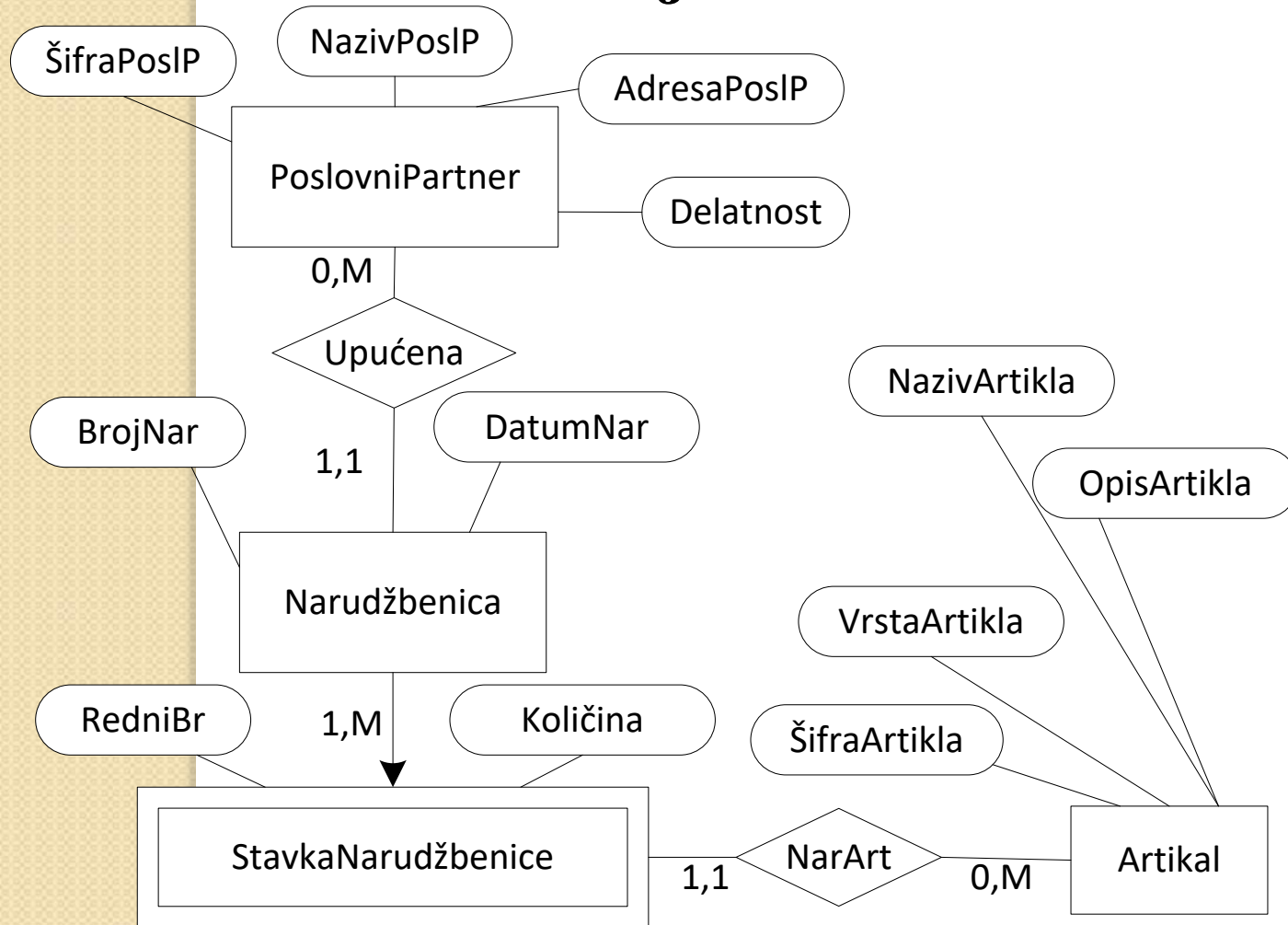
### Pravilo 1.1

Svaki tip objekat iz MOV postaje relacija.  
Ime tipa objekta postaje ime šeme relacije.  
Atributi objekta su atributi relacije.

### Pravilo 1.2

Svaki slab objekat takođe postaje šema relacije.  
Ime tipa objekta postaje ime šeme relacije.  
Identifikator nadređenog objekta postaje jedno od obeležja šeme relacije koja odgovara slabom objektu.  
Identifikator slabog objekta čini identifikator nadređenog objekta i obeležja slabog objekta koja jedinstveno identifikuje pojavljivanje slabog objekta u okviru pojavljivanja njemu nadređenog objekta.

# Transformacija MOV-a u Relacioni model



**PoslovniPartner** (PoslovniPartnerID, SifraPosIP, NazivPosIP, AdresaPosIP, Delatnost)

**Narudzbenica** (NarudzbenicaID, BrojNar, DatumNar, PoslovniPartnerID)

**StavkaNarudzbenice** (NarudzbenicaID, StavkaNarudzbenice ID, BrojNar, Rbr, Kolicina, ArtikalID)

**Artikal** (ArtikalID, SifraArtikla, VrstaArtikla, NazivArtikla, OpisArtikla)

# Transformacija MOV-a u Relacioni model

## **Pravilo 1.3 (Nadtip)**

Objekat nadtip (generalizovani tip objekta) postaje šema relacije.

Ime nadtipa postaje ime šeme relacije.

Obeležja nadtipa su obeležja šeme relacije.

Identifikator nadtipa postaje ključ šeme relacije.

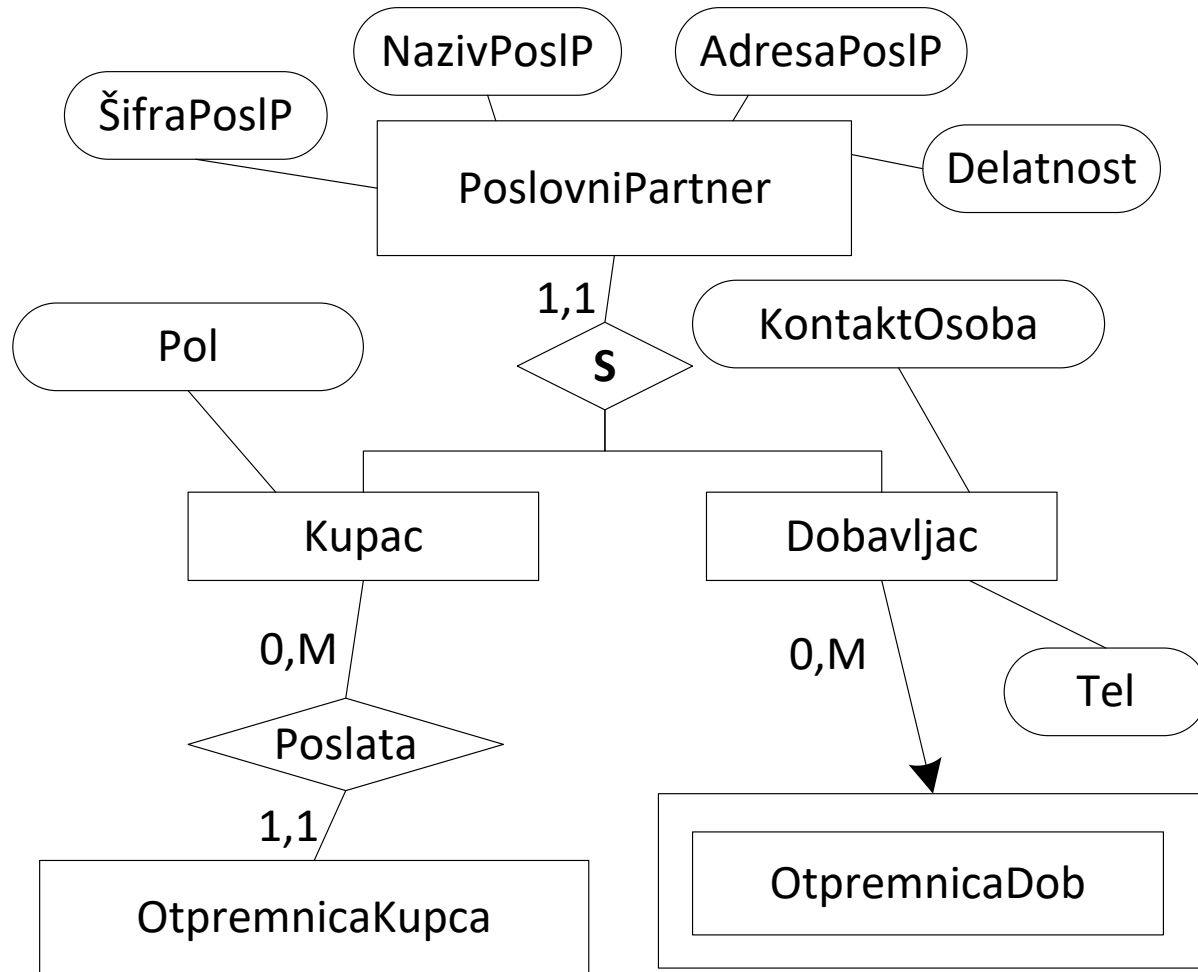
## **Pravilo 1.4 (Podtip)**

Objekat podtip postaje šema relacije.

Ime podtipa postaje ime šeme relacije.

Identifikator nadtipa postaje ključ (identifikatora) podtipa.

# Transformacija MOV-a u Relacioni model



**PoslovniPartner** (PoslovniPartnerID, SifraPosIP, NazivPosIP, AdresaPosIP, Delatnost)  
**Kupac** (PoslovniPartnerID, Pol) i **Dobavljac** (PoslovniPartnerID, KontaktOsoba, Tel)

# Transformacija MOV-a u Relacioni model

## 2. Pravila za binarne veze

### 2.1 Veza sa kardinalnošću (1,1)

Kad se kaže Veze sa kardinalnošću (1,1) onda se misli na sve veze objekata sa sledećim kardinalnostima: (1,1)-(1,1);(0,1)-(1,1);(0,1)-(0,1). Ovde se vidi da su sve gornje granice GG=1. Slede detaljna pravila svih prethodno navedenih kardinalnosti.

#### 2.1.1 Veza sa kardinalnošću (1,1) – (1,1)

Oba objekta koji u njoj učestvuju prevodimo u jednu šemu relacije, čija su obeležja sva obeležja jednog i drugog objekta. Kandidat za ključ u ovoj šemi relacije su identifikatori jednog i drugog objekta koji su u vezi.

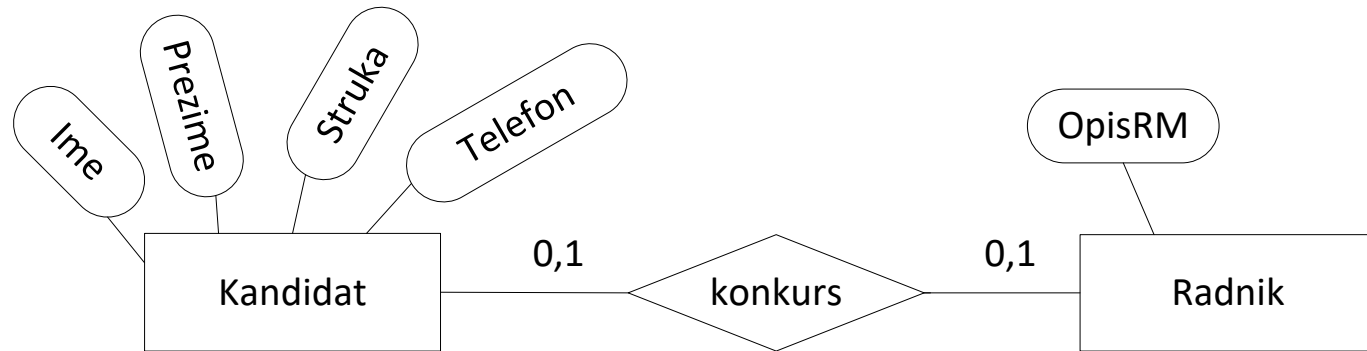
#### 2.1.2 Veza sa kardinalnošću (0,1) – (1,1)

Oba objekta u vezi prevodimo u dve šeme relacije. Za svaki objekat u vezi po jedna šema. Veza se predstavlja spoljnim ključem. Pravilo za prevođenje veze sa kardinalnošću (0,1) : (1,1) je njeno predstavljanje spoljnim ključem u šemi relacije objekta sa strane (1,1).

# Transformacija MOV-a u Relacioni model

## 2.1.3 Veza sa kardinalnošću (0,1) – (0,1)

Kreiraju se tri šeme. Po jedna za svaki objekat i jedna za vezu. Obeležja u šemi relacije koja odgovaraju vezi su i identifikatori objekata koji su u vezi i oba su kandidati za ključ.



**Kandidat** (**KandidatID**, Ime, Prezime, Struka, Telefon)

**Konkurs** (**KandidatID**, RadnikID)

**Radnik** (**RadnikID**, OpisRM)

ILI

**Kandidat** (**KandidatID**, Ime, Prezime, Struka, Telefon)

**Konkurs** (**RadnikID**, KandidatID)

**Radnik** (**RadnikID**, OpisRM)

# Transformacija MOV-a u Relacioni model

## 2.2 Veza sa kardinalnošću (1,1) – (0,M) , (1,1) – (1,M)

### 2.2.1 Veza sa kardinalnošću (1,1) – (0,M)

Ne postaju posebne šeme relacija. Identifikator objekta sa strane za koju je gornja granica kardinaliteta preslikavanja  $GG = M$  postaje obeležje šeme relacije koja odgovara objektu sa strane za koju je  $GG = 1$ .

### 2.2.2 Veza sa kardinalnošću (1,1) – (1,M)

Ne postaju posebne šeme relacija. Identifikator objekta sa strane za koju je gornja granica kardinaliteta preslikavanja  $GG = M$  postaje obeležje šeme relacije koja odgovara objektu sa strane za koju je  $GG = 1$ .

# Transformacija MOV-a u Relacioni model

## 2.3 Veza sa kardinalnošću (0,1) – (0,M) , (0,1) – (1,M)

### 2.3.1 Veza sa kardinalnošću (0,1) – (0,M)

Postaju posebne šeme relacije. Obeležja ove šeme relacije su identifikatori objekata koji su u vezi, a ključ šeme relacije je identifikator objekta za koji je  $GG = 1$ .

### 2.3.2 Veza sa kardinalnošću (0,1) – (1,M)

Postaju posebne šeme relacije. Obeležja ove šeme relacije su identifikatori objekata koji su u vezi, a ključ šeme relacije je identifikator objekta za koji je  $GG = 1$ .



# Transformacija MOV-a u Relacioni model

## 2.4 Veza sa kardinalnošću $(0,M) - (0,M)$ , $(0,M) - (1,M)$ , $(1,M) - (1,M)$

Postaju posebne šeme relacija. Obeležja ove šeme relacije su identifikatori objekata koji su u vezi i oni postaju složeni ključ šeme relacije.

## 2.5 Agregirani objekti

Aggregirani objekat (mešoviti tip objekat-veza, gerund) se posmatra na isti način kao veza sa kardinalnošću  $(0,M) - (0,M)$ ,  $(0,M) - (1,M)$ ,  $(1,M) - (1,M)$ .

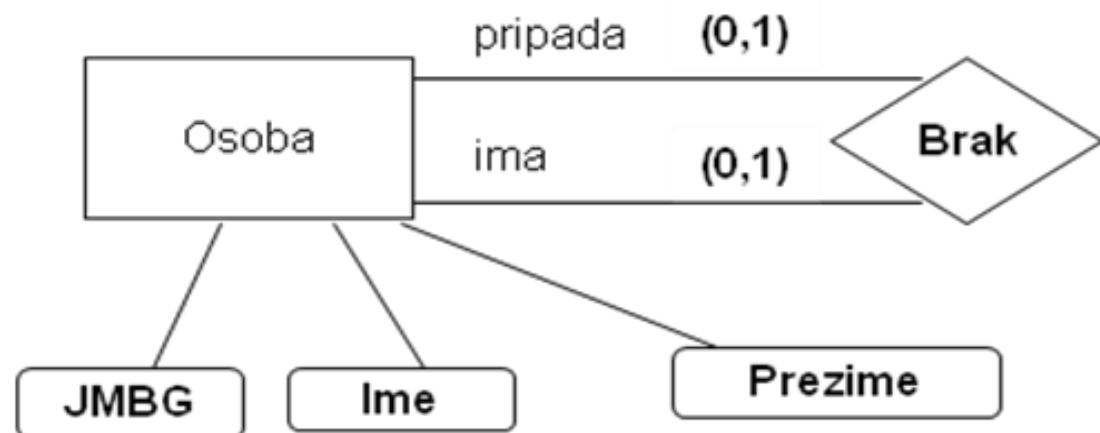
# Transformacija MOV-a u Relacioni model

## 3. Pravila za unarne veze

Prevođenje unarnih veza (unarnom nazivamo vezu između dva objekta istog tipa) u relacioni model podataka zavisi od kardinalnosti tipa veze i izvodi se kao i za druge tipove ranije opisanih binarnih veza.

### 3.1 Unarne veze (0,1) – (0,1)

Pri prevođenju unarnih veza s obzirom da bi spoljni ključ u šemi relacije imao isto ime kao i primarni ključ, vršimo njegovog preimenovanje.



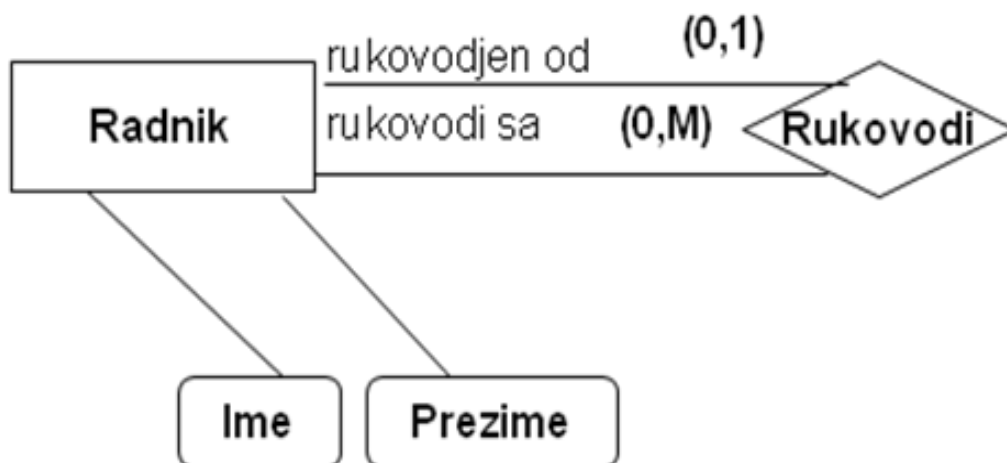
**Osoba** (OsobaID, JMBG, Ime, Prezime)

**Brak** (OsobaMuskaID, OsobaZenskaID)

# Transformacija MOV-a u Relacioni model

## 3.2 Unarne veze (0,1) – (0,M)

Jedan radnik može da rukovodi sa više radnika i može imati jednog nadređenog rukovodioca. Svaki radnik ne mora imati nadređenog rukovodioca i svaki radnik ne mora biti rukovodilac.



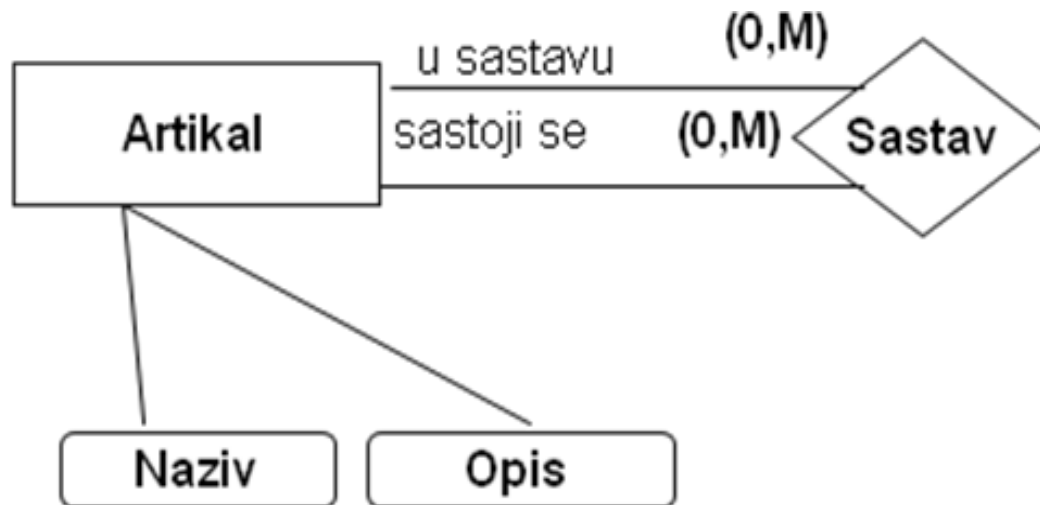
**Radnik** (RadnikID, Ime, Prezime)

**Rukovodi** (RadnikID, RadnikRukovodiID)

# Transformacija MOV-a u Relacioni model

## 3.3 Unarne veze (0,M) – (0,M)

Jedan artikal može da se sastoji iz više sastavnih delova. Svaki artikal ne mora imati sastavni deo.



**Artikal** (ArtikalID, Naziv, Opis)

**Sastav** (ArtikalID, ArtikalUSastavuID)

# Fizičko projektovanje

- **Fizičko projektovanje** baze podataka je projektovanje interne, fizičke strukture baze podataka, na osnovu logičkog modela i specifikacija svih aplikacija i nefunkcionalnih specifikacija.
- Dok logički model baze podataka definiše strukture pogodne za razvoj aplikacija, fizička struktura baze podataka treba da omogući ostvarivanje zadovoljavajućih performansi sistema.

# Fizičko projektovanje baza podataka

- Osnovni cilj fizičkog projektovanja baza podataka je da se zadovolje **nefunkcionalne specifikacije** prikupljene u fazi analize sistema.
- **Nefunkcionalne specifikacije** najčešće definišu performanse koje ceo sistem treba da zadovolji, kao i posebne zahteve za neke pojedinačne ili grupe aplikacija.
- Stalno se primenjuju ekspertska znanja u fazi fizičkog projektovanja, a zatim se stalno prate performanse bitnih aplikacija i vrši se povremeno podešavanje (engl. tuning) fizičke strukture baze.

# **Fizičko projektovanje baza podataka obuhvata:**

**1. Projektovanje distribucije baze podataka.** Ponekad se baza podataka projektovane logičke strukture fizički distribuira na više nezavisnih SUBP.

**2. Prilagođavanje logičke strukture konkretnom skupu aplikacija - denormalizacija.** Nisu sve aplikacije u nekom IS podjednako značajne. Obično se zahteva da se jedan manji broj aplikacije obavlja sa zadovoljavajućim performansama, makar zbog toga narušili performanse ostalih.

**3. Klasterovanje** – više servera, a jedna baza.

**4. Izbor pristupa pojedinim tabelama.** U pojedinim relacionim SUBP moguće je izabrati način pristupa po primarnom ključu neke tabele: indeksni ili hešing.

*Projektovanje informacionih sistema*

***HVALA !***

dr Rade Matić