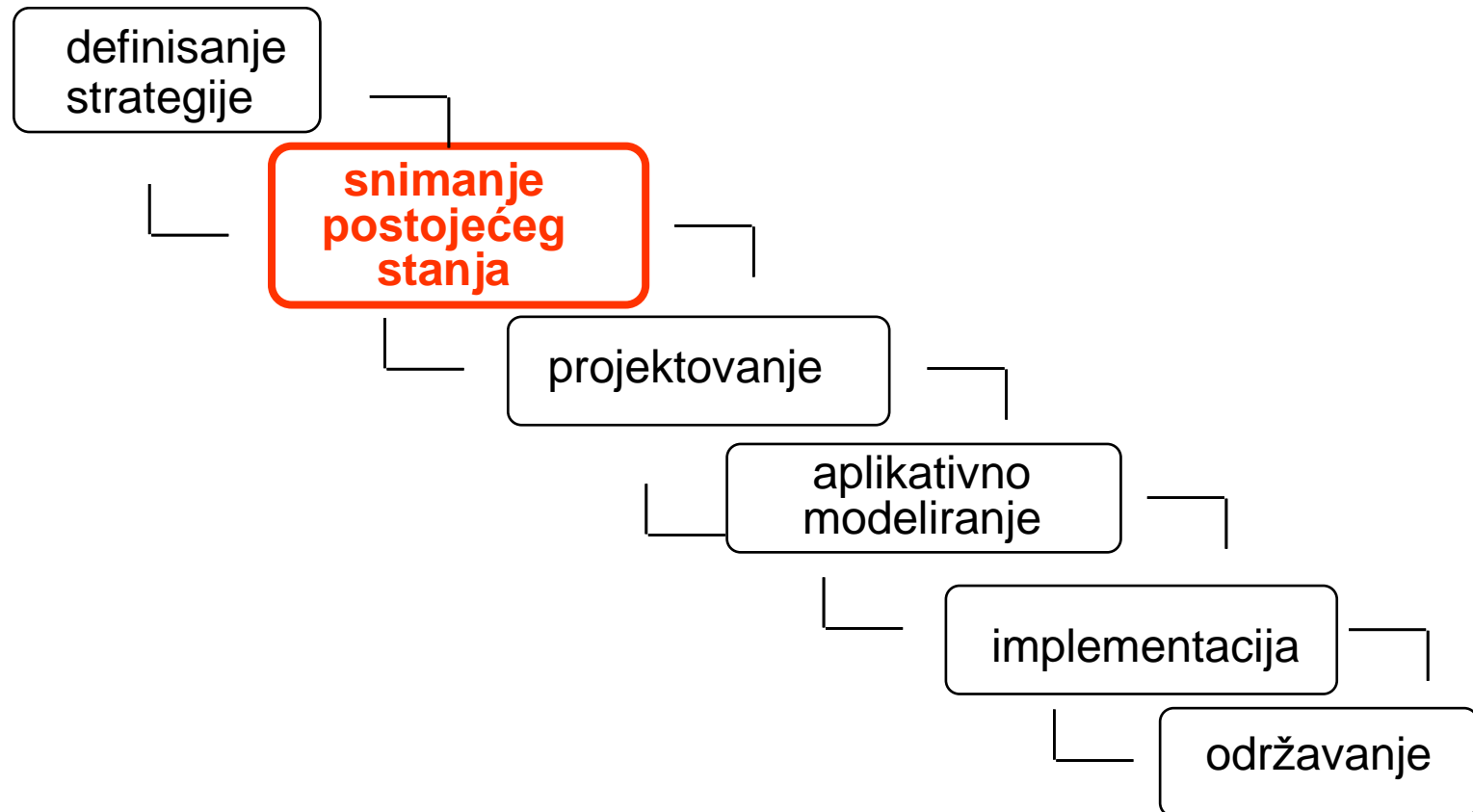


Modelovanje podataka



Model podataka - osnovne komponente

- 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.

Modeliranje sistema

- SSA - strukturna sistem analiza -
modeliranje funkcija i procesa
- Modeliranje podataka
- UML - Jedinstveni jezik modelovanja

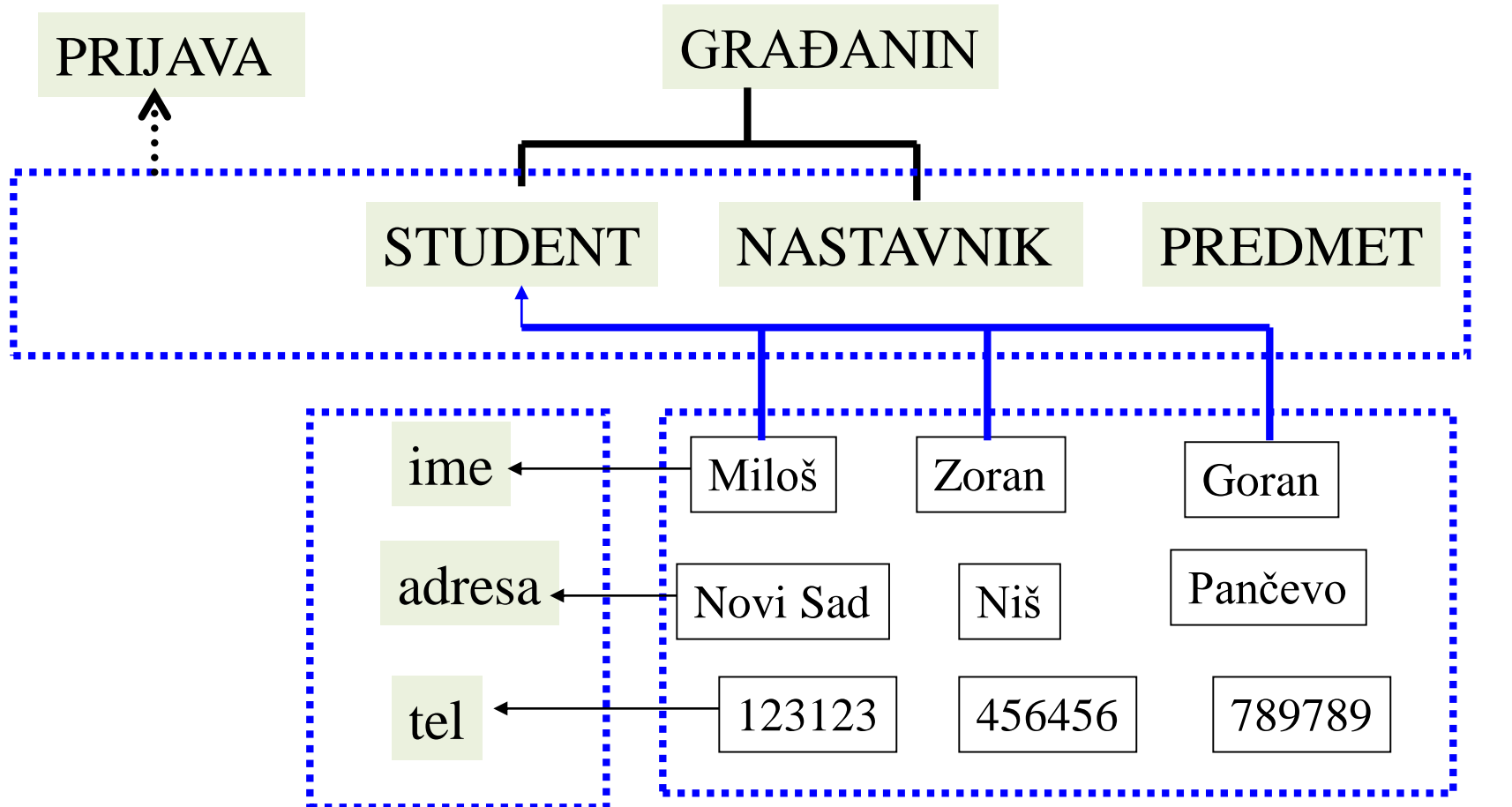
Model podataka - komponente

- Model podataka poseduje tri osnovne komponente:
 1. Struktura modela, skup koncepata za opis objekata sistema, njihovih atributa i međusobnih veza.
 2. Ograničenja na vrednosti podataka koja u svakom stacioniranom stanju moraju biti zadovoljena. Nazivaju se **statičkim pravilima integriteta** modela podataka.
 3. Operacije nad konceptima strukture.

APSTRAKCIJE U MODELU PODATAKA

- KLASIFIKACIJA (tipizacija)
- GENERALIZACIJA
- AGREGACIJA

PRIMER APSTRAKCIJA



— klasifikacija (tipizacija)
— generalizacija
... agregacija

Vrste modela

- Model Objekti - Veze
- Relacioni model
- *Istorija*
 - *Hijerarhijski model*
 - *Mrežni (CODASYL) model*

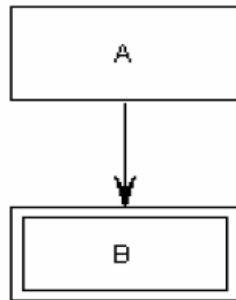
Model Objekti-Veže (MOV)

E-R Model, Entity-Relationship Model

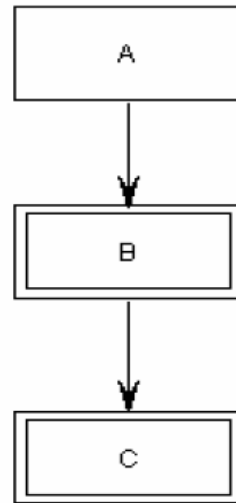
- **Objekat (entitet)** - grupa (skup) sadržaja sa karakteristikama koje su bitne za celinu.
- **kandidati za objekte:**
 - Fizički objekti (vozila, mašine,...)
 - Osobe
 - Lokacije (mesta, adrese, koordinate...)
 - Organizacije
 - Grupe/klase/tipovi (proizvoda, poslova...)
 - Dokumenta
 - Pridruženja (zadatak-osoba, vozilo-vožnja)
 - Pripadnost/članstvo

Objekti po PMOV sintaksi

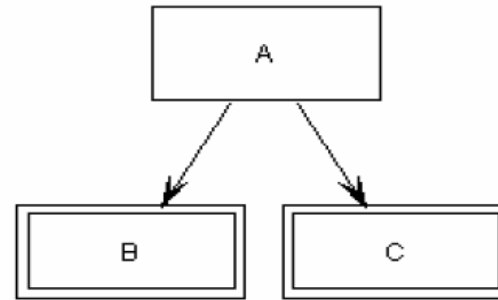
a)



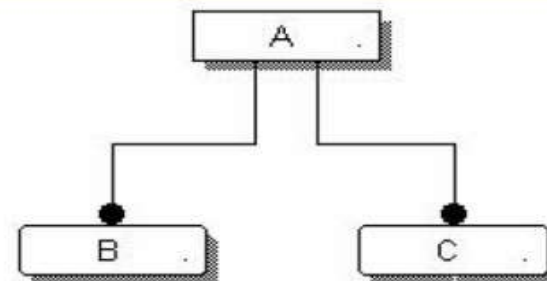
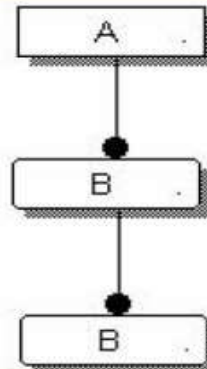
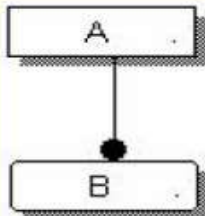
b)



c)



Objekti po IDEF1x standardu

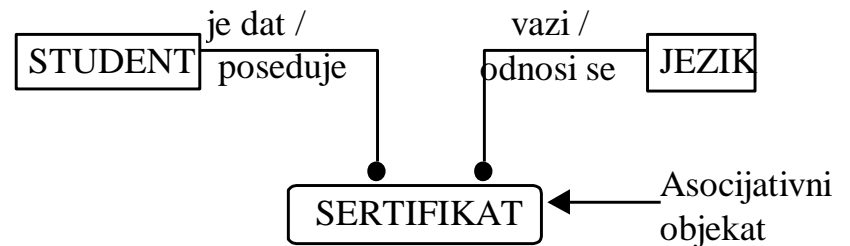


MOV – vrste objekata

- **Nezavisan objekat** ima osobinu koja ga može jednoznačno identifikovati (ne zavisi od drugih objekata).
- **Zavisan objekat** je onaj čija egzistencija i identifikacija zavise od drugog (ili drugih) objekata.

Postoje:

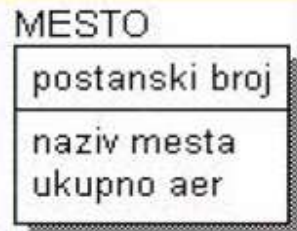
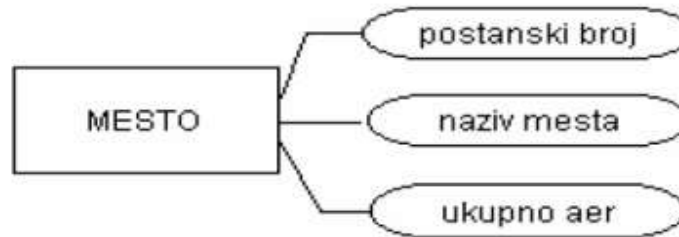
- ***karakterističan objekat*** (*slab objekat*) – onaj koji se ponavlja više puta za određeni nezavisni objekat;
- ***asocijativni objekat***, koji predstavlja vezu više objekata;



MOV - atributi

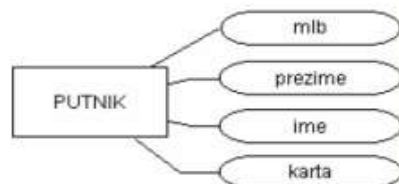
- **Atributi** su karakteristike ili osobine iskazane kao jedna ili više vrednosti koje opisuju objekat. Svaki atribut ima svoje ime.

Atributi i domen

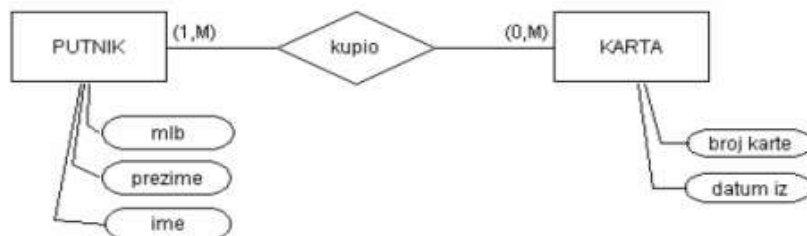


Viseznacni atributi

a)

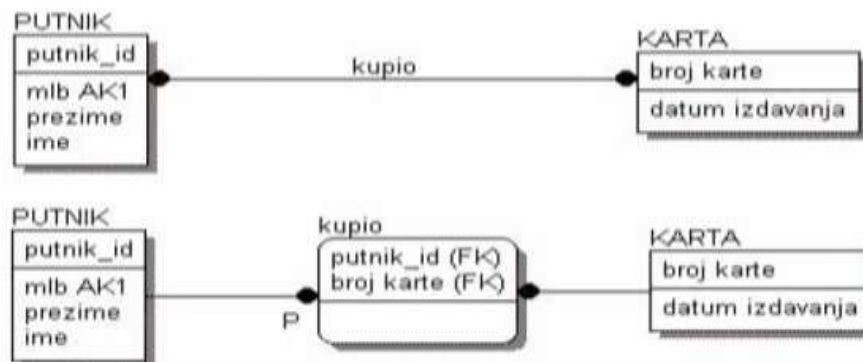


b)



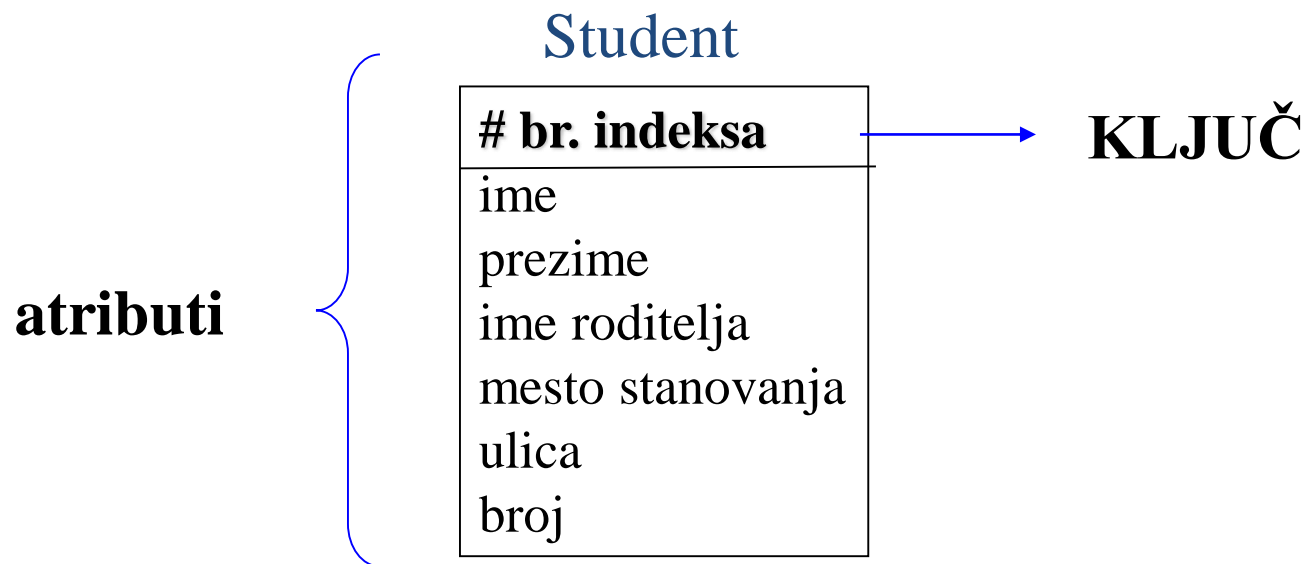
Viseznacni atributi

b)



Ključ objekta

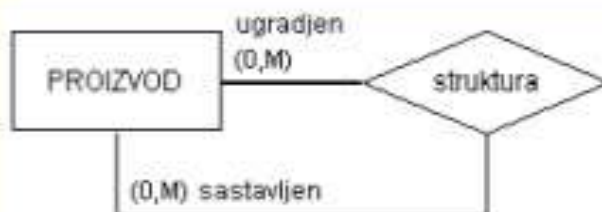
- Ključ je vrsta atributa koji jedinstveno identifikuje svaki primerak objekta.
- Od atributa - kandidata za ključeve bira se jedan koji postaje **primarni ključ**.
- Nijedan deo primarnog ključa ne može biti prazan ili nedostajući.



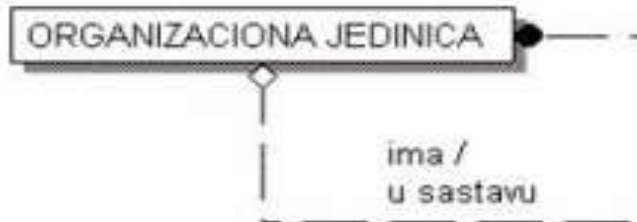
Ključevi

- Ako ključ čini samo jedan atribut, onda je **prost** ključ; u suprotnom je **složen**.
- **Alternativni ključ** predstavlja atribut ili grupa atributa koji jedinstveno identifikuju primerke entiteta, ali postoje objekti za koje taj atribut nije definisan
- **Preneseni ključ** (Foreign Key) je atribut koji povezuje objekat 'dete' sa objektom 'roditelj'

Veze po PMOV sintaksi

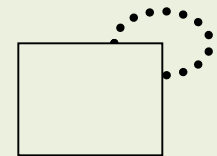


Veze po IDEF1x i IE standardu



Veze (Relationship)

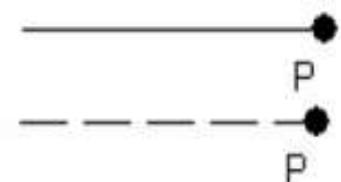
- identifikujuće (dete - roditelj)
- neidentifikujuće
 - obavezne
 - neobavezne
- rekurzivne (na sebe samog)
 - primer : radnik - šef



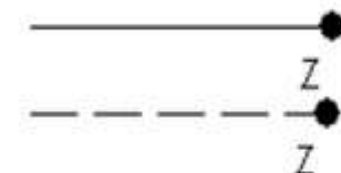
VERZIJE MOV-a: IDEF1x standard



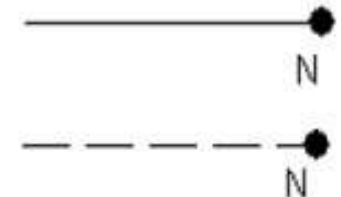
Jedan: nula ili vise



Jedan: jedan ili vise



Jedan: nula ili jedan



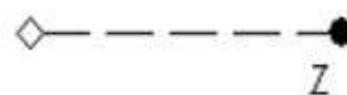
Jedan: tacno N, gde
je N konkretan broj



nula ili jedan:
nula ili vise



nula ili jedan:
jedan ili vise



nula ili jedan:
nula ili jedan



nula ili jedan:
tacno N, gde je N
konkretan broj

Kardinalnost veza

Kardinalnost veza roditelj - dete

- nula, jedan ili više
- jedan ili više
- nula ili jedan
- tačno n (*primer : godišnja doba, posada aviona*)

Kardinalnost veza dete - roditelj

- dozvoljena nula
- nije dozvoljena nula

Veza “više prema više”



Referencijalni integritet

- Omogućava korektno povezivanje objekata
- Definiše se za svaku vezu, posebno za roditelja, posebno za dete
- Dolazi do izražaja kod održavanja modela
- Realizuje se putem ograničenja, operacija i akcija

Ograničenja

- Nad strukturom
 - Integritet entiteta
 - Nad standardnim domenom
 - Tip, dužina podataka
- Nad vrednošću domena
 - Dozvoljene vrednosti
- Na kardinalnost
 - (0,1,n), (1,n), (0,1), (Exactly)

- **Specifikacija tipa ograničenja u MP**

TipO($T(t)$, TOd , TOi , TFz , TPi)

- *TipO* - oznaka tipa ograničenja
- *T(t)* - definicija tipa logičke strukture obeležja
» s uključenim kritičnim operacijama i mogućim akcijama
- *TOd* - specifikacija oblasti definisanosti
- *TOi* - specifikacija oblasti interpretacije
- *TFz* - definicija formule za zapisivanje
- *TPi* - definicija pravila za interpretaciju

- **Mogući tipovi ograničenja u RMP**

- s pridruženim oznakama (*TipO*)

- ograničenje domena (*DomCon*)
 - ograničenje vrednosti obeležja (*AttValCon*)
 - ograničenje torke (*TupleCon*)
 - prošireno ograničenje torke (*ExTupleCon*)
 - ograničenje ključa (*KeyCon*)
 - ograničenje jedinstvenosti (*UniqueCon*)
 - zavisnost sadržavanja (*InCon*)
 - proširena zavisnost sadržavanja (*ExInCon*)
 - selektivna zavisnost sadržavanja (*SelInCon*)
 - selektivna proširena zavisnost sadrž. (*SelExInCon*)

- Primer – tip ograničenja



<i>TipO</i>	<i>ReflnCon</i>	ograničenje referencijalnog integriteta						
<i>T(t)</i>	<i>Role₁</i>	<i>referencing</i>	<i>Mult₁</i>	1	<i>AtStr₁</i>	<i>array</i>	<i>AtMult₁</i>	*
	<i>ins</i>	<i>NoAction, SetNull, SetDefault, <<UserDef></i>						
	<i>upd</i>	<i>NoAction, SetNull, SetDefault, <<UserDef></i>						
	<i>Role₂</i>	<i>referenced</i>	<i>Mult_m</i>	1	<i>AtStr_m</i>	<i>array</i>	<i>AtMult_m</i>	*
	<i>del</i>	<i>NoAction, Cascade, SetNull, SetDefault, <<UserDef></i>						
	<i>upd</i>	<i>NoAction, Cascade, SetNull, SetDefault, <<UserDef></i>						
<i>TOd</i>	2	višerelaciono ograničenje ("dvorelaciono")						
<i>TOi</i>	<i>m</i>	međurelaciono ograničenje						
<i>TFz</i>	$N_i[X] \subseteq N_j[Y], Key(N_j, Y)$							
<i>TPi</i>	$\pi_X(r(N_i)) \subseteq \pi_Y(r(N_j))$							

Operacije

- nad konceptima strukture, po ograničenjima
 - INSERT (ubacivanje)
 - REPLACE (ključ, deo ključa)
 - DELETE (objekat, veza, roditelj)

Dinamička pravila integriteta

- Dinamičkim pravilima se održava integritet podataka pri izvršenju operacija održavanja baze podataka (**insert, update, delete**).
- Jedno dinamičko pravilo integriteta čini trojka
<OPERACIJA, OGRANIČENJE, AKCIJA>
- Akcije koje se preduzimaju su:
 - Restrict
 - Cascade
 - Nullifies (SetNull)
 - SetDefault

Akcije

- Restrict - odbija operaciju koja narušava integritet
- Cascade - prosleđuje operaciju
- Default - kreira pretpostavljeni objekat
- Set null - nepoznato pojavljivanje
- None - nema ograničenja

Implementacija ograničenja šeme BP

– deklarativni mehanizmi

- aktivnosti provere važenja ograničenja i očuvanja konzistentnosti se, većim delom, podrazumevaju
 - SQL klauzula CONSTRAINT
 - CREATE DOMAIN, CREATE ASSERTION

– proceduralni mehanizmi

- aktivnosti provere važenja ograničenja i očuvanja konzistentnosti se, većim delom, programiraju
 - putem proceduralnog jezika
 - CREATE TRIGGER
 - CREATE PROCEDURE, CREATE FUNCTION
 - CREATE PACKAGE, CREATE PACKAGE BODY

- **Deklarativni mehanizmi (SQL:2006)**

- SQL klauzula CONSTRAINT

- opšti oblik sintakse

[CONSTRAINT *NazivOgr*] *SpecifikacijaTipaOgraničenja*
[INITIALLY {DEFERRED | IMMEDIATE}]
[[NOT] DEFERRABLE]

- *SpecifikacijaTipaOgraničenja*
 - NOT NULL- ograničenje nula vrednosti
 - PRIMARY KEY ... - ograničenje primarnog ključa
 - UNIQUE ... - ograničenje jedinstvenosti
 - CHECK ... - ograničenje torke
 - FOREIGN KEY ... - ograničenje stranog ključa

- Sintaksa za definisanje trigeru (PL/SQL)

```
CREATE [OR REPLACE] TRIGGER NazivTrigera
  BEFORE | AFTER | INSTEAD OF
    INSERT | DELETE | UPDATE [OF ListaObeležja]
    [ OR INSERT | DELETE | UPDATE [ OF ListaObeležja ] ... ]
  ON NazivTabele
  [ FOR EACH ROW [WHEN (LogičkiUslovPokretanjaTrigera)]]
  [ REFERENCING OLD NazivOld AS NEW AS NazivNew ]
  [ DECLARE
    Deklarativni deo - lokalne deklaracije
  ]
  BEGIN
    Izvršni deo - proceduralni deo, specifikacija aktivnosti
  [ EXCEPTION
    Deo za obradu izuzetaka ]
  END NazivTrigera
```

- Sintaksa za kreiranje procedure (PL/SQL)

CREATE [OR REPLACE] PROCEDURE *NazivProcedure*

[(*ListaFormalnihParametara*)]

AS | IS

Deklarativni deo - lokalne deklaracije procedure

- *tipovi podataka*
- *konstante i promenljive*
- *procedure i funkcije*
- *kursorska područja*
- *izuzeci*

BEGIN

Izvršni deo - proceduralni deo, specifikacija aktivnosti

[EXCEPTION

Deo za obradu izuzetaka]

END *NazivProcedure*

- Sintaksa za kreiranje funkcije (PL/SQL)

```
CREATE [OR REPLACE] FUNCTION NazivFunkcije  
  [ (ListaFormalnihParametara) ]  
  RETURN TipPodatkaPovratneVrednostiFunkcije  
  AS | IS  
    Deklarativni deo - lokalne deklaracije funkcije  
  BEGIN  
    Izvršni deo - proceduralni deo, specifikacija aktivnosti  
    /* Zahteva pojavljivanje naredbe oblika RETURN Izraz */  
  [ EXCEPTION  
    Deo za obradu izuzetaka ]  
  END NazivFunkcije
```


- Sintaksa za kreiranje paketa i tela paketa (PL/SQL)

```
CREATE [OR REPLACE] PACKAGE NazivPaketa  
AS | IS
```

Deklarativni deo – javne deklaracije paketa

- tipovi podataka
- konstante i promenljive
- zaglavlja procedura i funkcija
- kursorska područja
- izuzeci

```
[ BEGIN
```

Deo za inicijalizaciju - proceduralni, specifikacija aktivnosti

```
]
```

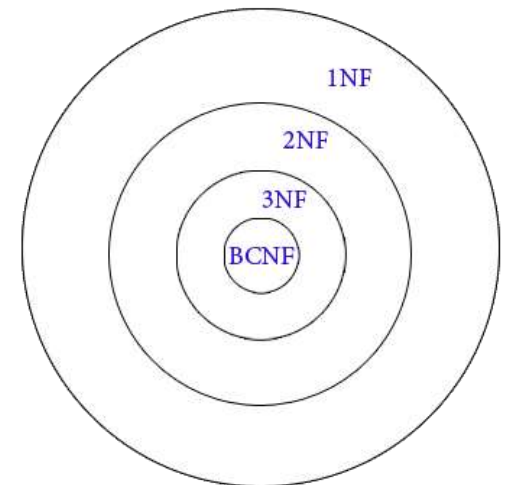
```
END NazivPaketa
```

NORMALIZACIJA

BAZE PODATAKA

- **Normalizacija** je postupak projektovanja logičke strukture baze podataka, gde se nastoji otkloniti redudansa podataka bez gubitka informacija.
- **Redundansa** predstavlja višestruko memorisanje iste informacije u bazi podataka.
- Cilj:
 - kontrolisana redundansa podataka
 - jednostavno korišćenje i menjanje podataka (održavanje podataka)

- Definisano je šest normalnih formi (NF):
 1. **prva normalna forma (1NF),**
 2. **druga normalna forma (2NF),**
 3. **treća normalna forma (3NF),**
 4. Boyce/Coddova normalna forma (BCNF),
 5. četvrta normalna forma (4NF),
 6. peta normalna forma (5NF).



Postupak normalizacije

“jedna činjenica na jednom mestu”

- uklanjanje redundanse*!

- Prva normalna forma
 - Svaki od atributa ima jedno značenje i ne više od jedne vrednosti za svaki primerak (instancu)
- Druga normalna forma
 - Svaki atribut koji nije ključ potpuno zavisi od primarnog ključa
- Treća normalna forma
 - Svaki atribut koji nije ključ mora da zavisi jedino od primarnog ključa

* redundansa – višestruko ponavljanje istog podatka u bazi

1NF - primer

RADNIK

Šifra radnika
Prezime
Ime
Kvalifikacija
Dat.zaposl. ili dat.odlaska

123	Petar Perić	Programer	01.11.1998
124	Ana Ilić	Projektant	20.05.2005
125	Milan Milić	Operater	15.09.2004

1NF : Jednoznačna upotreba atributa;
Svaki od atributa ima jedno značenje
i ne više od jedne vrednosti za svaki primerak (instancu)

2NF - Primer

ISPLATA

Šifra radnika

Br. isplate

~~Datum zaposlenja~~

Isplata

123	1	01.10.1997	20000
123	2	01.10.1997	22500
123	3	01.10.1997	22000
123	4	01.10.1997	25000

2NF : Svaki atribut koji nije ključ mora potpuno da zavisi od primarnog ključa, inače ga treba premestiti u nadređeni entitet

3NF - Primer

Radna lista

Broj liste
Br. časova
Cena časa
Iznos

321	10	100	1000
322	2	80	160
323	3	150	450

3NF : Svaki atribut koji nije ključ mora da zavisi jedino od primarnog ključa;
Ne koristiti attribute čija se vrednost može izračunati

MODEL

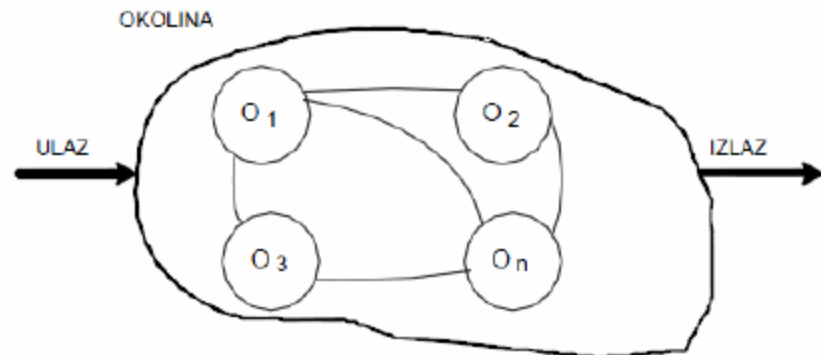
MODEL

OBJEKTI - VEZE

OBJEKTI - VEZE

Model podataka

- Model objekti-veze služi za prikazivanje **objekata** sistema, njihovih međusobnih **veza** i **atributa** koji ih opisuju.
- Predstavlja model podataka kojim se opisuju statičke karakteristike sistema i definiše se logička struktura baze podataka.



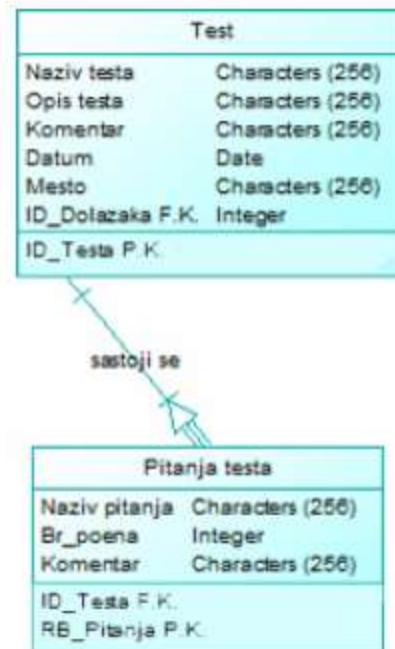
Model podataka

- Objekat (entitet) u modelu može biti:
 - ✓ fizički objekat sistema (proizvod, radnik),
 - ✓ koncept, događaj i dr. (konferencija, plata, kasa).
- Objekti u sistemu se opisuju preko svojih svojstava (atributa).
- Atribut je elementarni podatak, nosilac informacija, koji uzima vrednosti samo iz svog domena.

Kadrovi	
Ime	Characters (256)
Prezime	Characters (256)
Stručna sprema	Characters (256)
Godine iskustva	Integer
Datum zaposlenja	Date
ID_Kadra	P.K.

MOV objekti

- Razlikujemo **jake** i **slabe** objekte.
- Slabi objekat u sistemu je zavisan egzistencijalno (ne može da postoji) i identifikaciono (ne može da se identifikuje) od njemu nadređenog objekta.

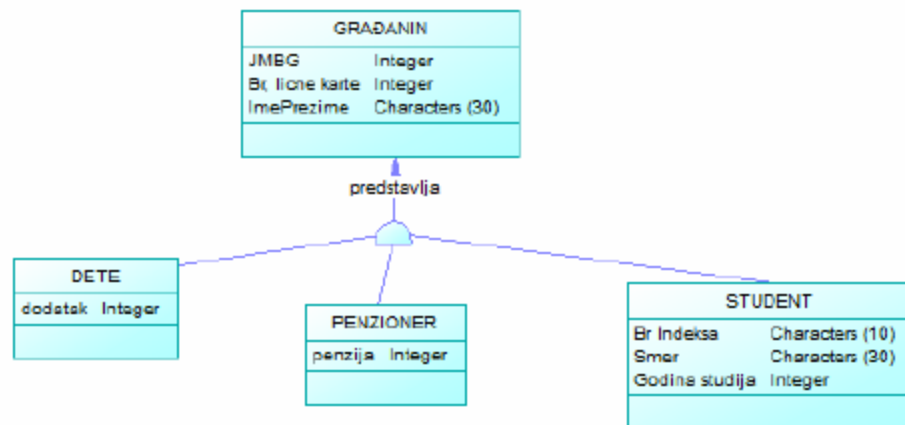


Apstrakcija podataka

- Apstrakcija je kontrolisano uključivanje detalja, odnosno "izvlačenje" opštih karakteristika u opisivanju nekog sistema.
- Apstrakcije podataka:
 1. **Klasifikacija (tipizacija)**
 2. **Generalizacija i specijalizacija**
 3. **Agregacija i dekompozicija**

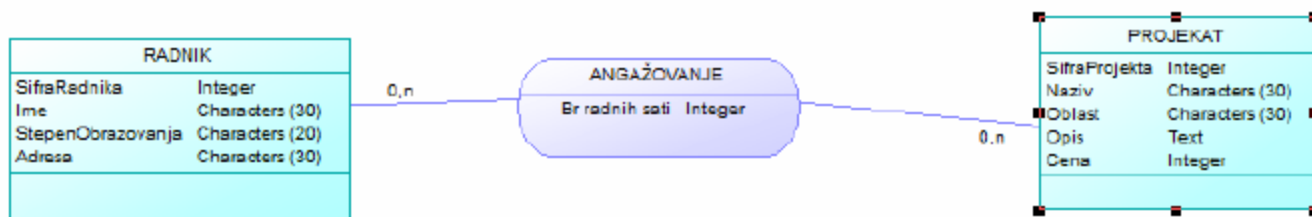
- **Klasifikacija ili tipizacija** je apstrakcija u kojoj se skup sličnih objekata predstavlja jednom klasom objekata, odnosno svaki objekat iz posmatranog skupa odgovarajućim tipom objekta.
- Primer klasifikacije:
 - Matematika, Uvod u informacione sisteme, Operaciona istraživanja su Ispiti.

- **Generalizacija** je apstrakcija u kojoj se skup sličnih tipova objekata predstavlja opštijim generičkim tipom (nadtipom).
- Slični tipovi objekata su oni tipovi koji imaju jedan broj istih (zajedničkih) atributa, veza i/ili operacija.



* Primer bez ključeva

- Agregacija je apstrakcija u kojoj se skup tipova objekata i njihovih veza tretira kao jedinstveni **agregirani tip objekta**.



MOV - veze

- Veza opisuje međusoban odnos objekata, odnosno učešće jednog objekta u drugom.

- Tipovi veza:

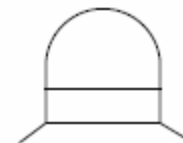
1. Neidentifikujuća



2. Identifikujuća



3. Nasleđivanje



4. Asocijacija



Ključni atributi

- **Primarni ključ** je atribut ili skup atributa koji jedinstveno identifikuje jedno pojavljivanje tipa objekta.
- **Spoljni ključ** je atribut koji predstavlja identifikator entiteta prema kome dati entitet ima preslikavanje.
- **Kandidati za ključ** su atributi identifikatori koji jedinstveno određuju jedno pojavljivanje tipa objekta.

Nastavnik (SifraNast, Ime, Titula)

Predmet (SifraPredmeta, Naziv, Opis, NacinPolaganja, SifraNast)

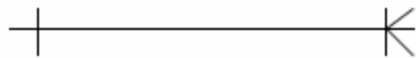
Kardinalnost

- Svaka binarna veza definiše dva preslikavanja.
- Preslikavanja određuju uloge objekta u vezi.
- **Kardinalnost preslikavanja** ($E1 \rightarrow E2$) definiše najmanji mogući (DG) i najveći mogući (GG) broj pojavljivanja tipa objekta E2, za jedno pojavljivanje tipa objekta E1.

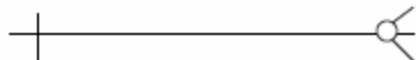
$DG \in [0, 1, 2, 3, \dots]$, $GG \in [1, \dots, M] \rightarrow DG \leq GG$

NEIDENTIFIKUJUĆA VEZA

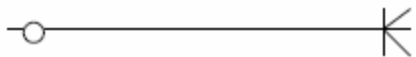
- $(1,1) - (1,M)$



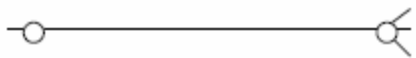
- $(1,1) - (0,M)$



- $(0,1) - (1,M)$



- $(0,1) - (0,M)$



- $(0,1) - (0,1)$

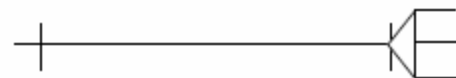


- $(1,1) - (0,1)$

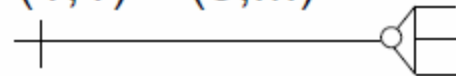


IDENTIFIKUJUĆA VEZA

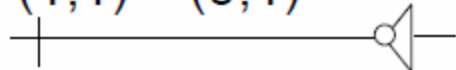
- $(1,1) - (1,M)$



- $(1,1) - (0,M)$



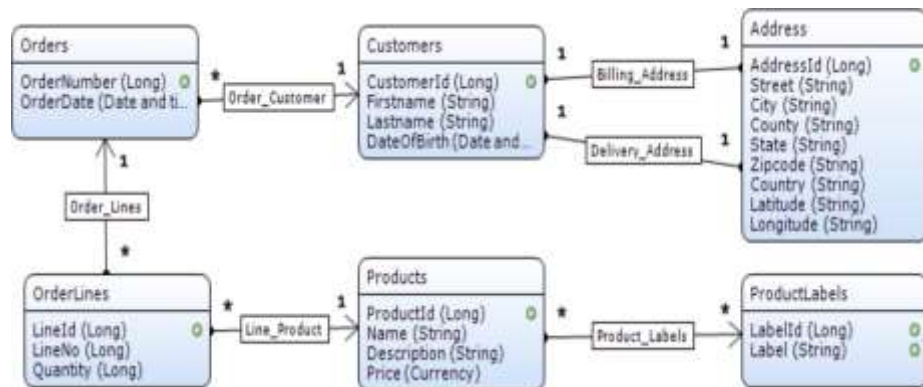
- $(1,1) - (0,1)$



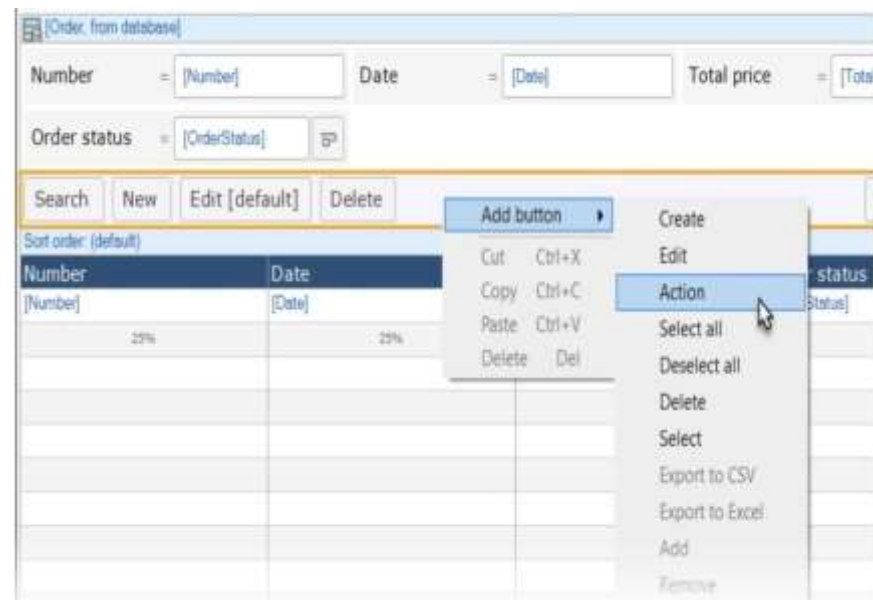
MODELIRANJE IS

- Tradicionalne metode:
 - modelovanje podataka
 - modelovanje procesa
- Objektno-orijentisane metode
 - podaci i procesi enkapsulirani u okviru klasa (atributi i metode)
- U praksi se ravnopravno koriste oba pristupa

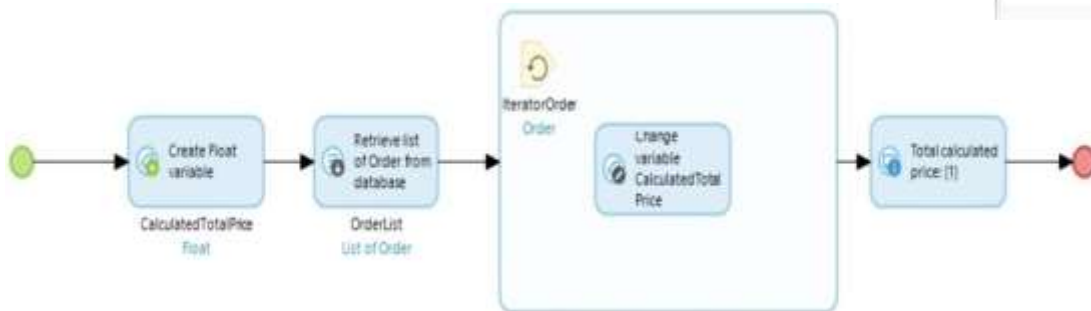
Mendix – *low-code* platforma



Model podataka



Izgradnja view-a

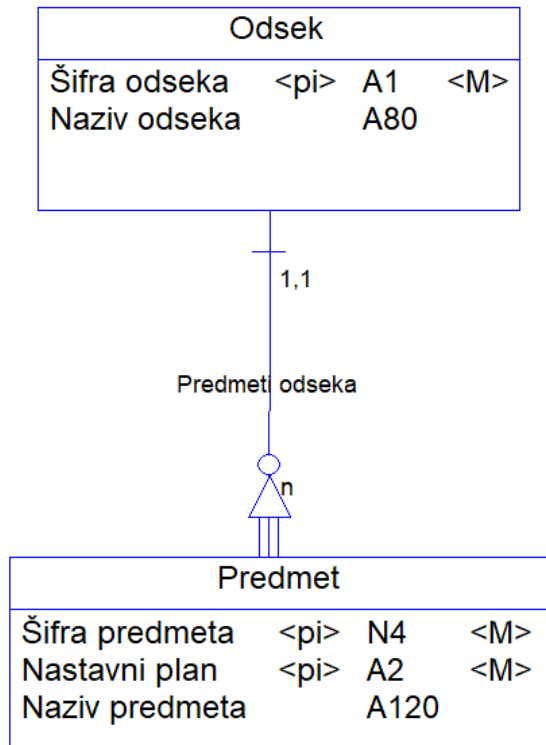


Definisanje logike

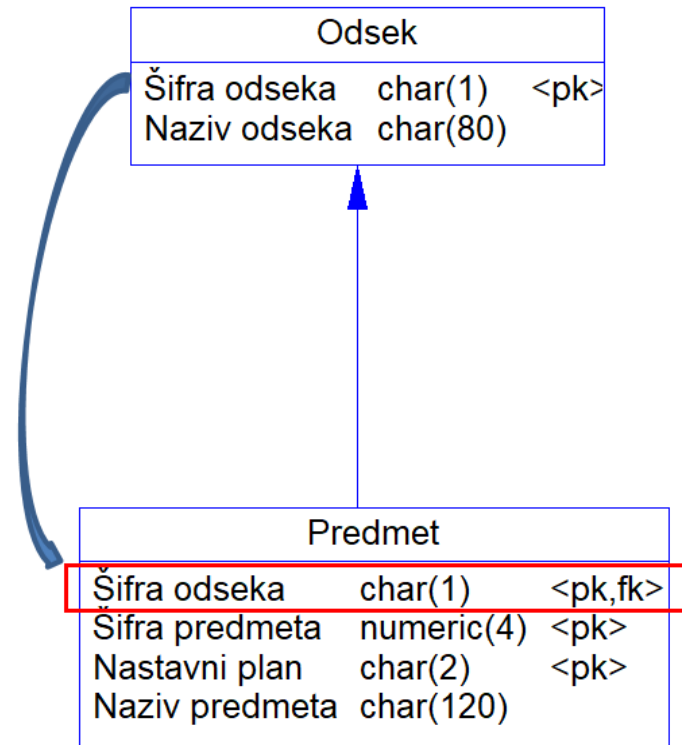
Modelovanje podataka

- Obično na:
- logičkom (konceptualnom) nivou
 - nezavisno od korišćene platforme
 - ne mora čak ni imati fizičku implementaciju
 - postoje različite notacije
- fizičkom nivou
 - unose se i detalji konkretne platforme (Oracle, MySQL, MS-SQL server, Progress...)

Primer konceptualnog i fizičkog modela



Logičko-Konceptualni model



Fizički model

Kako se formira model podataka

✓ izolovati tipove entiteta

- bitni pojmovi koje korisnici spominju, imenice u rečenicama...
 - Student, predmet, kurs, preduzeće, narudžbenica, mašina, pogon...
- mogu biti fizički objekti u realnom sistemu ali i dokumenti

✓ izolovati attribute

- obično se prvo otkriju samo osnovni, kasnijom analizom se vrši dopunjavanje

Kako se formira model podataka

- Atributi
 - naziv
 - tip podatka
 - dužina
 - preciznost
 - obaveznost unosa vrednosti (mandatory)
 - inicijalna vrednost
 - ograničenja na vrednosti (D:G - donje i gornje granice, nabrojani skup vrednosti)
- ✓ Odrediti ključeve

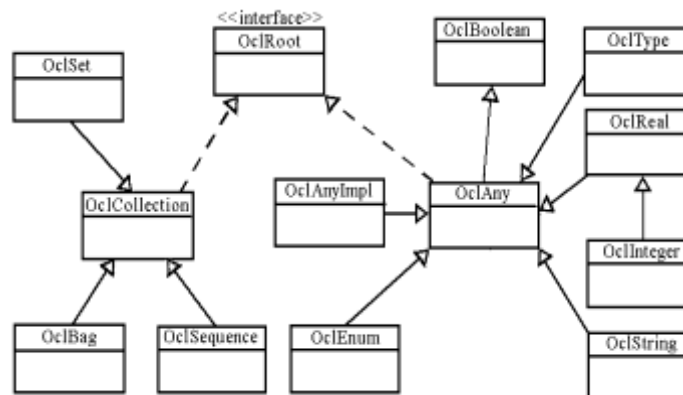
Kako se formira model podataka

- ✓ Odrediti veze i njihove osobine
 - glagoli u rečenicama
 - Proizvod **se sastoji** od materijala i poluproizvoda
 - Student **pohađa** kurs
 - vrsta veze (jaka, slaba)
 - kardinalitet
 - atributi veze

Kako se formira model podataka

✓ Odrediti ograničenja

- neka ograničenja se mogu projektovati sredstvima modela podataka
 - nabrojane vrednosti
 - donja i gornja granica vrednosti atributa
 - obaveznost unosa vrednosti nekog atributa
- neka se mogu samo zabeležiti i kasnije programski implementirati
- postoje i specijalni jezici za modelovanje ograničenja
 - npr. OCL



Dileme kod projektovanja entiteta

- Da li je neki pojam entitet ili atribut?
 - Npr. državljanstvo, mesto rođenja, opština....
- Da li uvesti atribut ili definisati nabrojani tip
 - Npr. Status radnika: zaposlen, prekinuo rad, u penziji

[illegible]

Dilema kod izbora ključeva - zaključak

○ **Prirodni**

- bliži realnom sistemu
- lakše sprovesti reinženjering (prenos postojećih podataka)
- korisnici poslovnih sistema su često navikli na prirodne ključeve
- prirodni ključevi sa puno komponenti mogu da loše utiču na performansu baze podataka

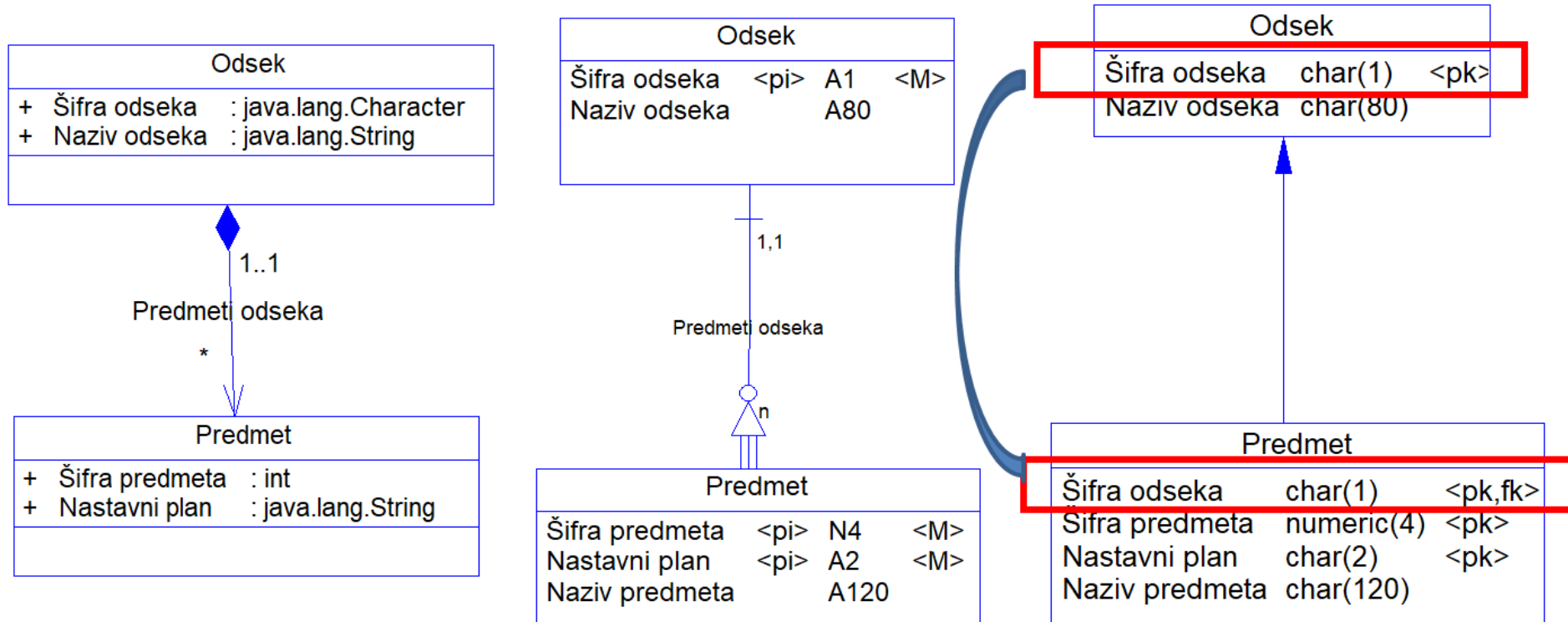
○ **Surogat ključevi**

- većina sistema za implementaciju srednjeg sloga preporučuje korišćenje surogat ključeva

Dileme kod projektovanja veza

- Identifikaciona (jaka veza) ili ne?
- veza više-na-više ili entitet?
- generalizacija ili ne?
- kada koristiti vezu sa kardinalitetom 1-1?

Identifikaciona veza (utiče na ključ)

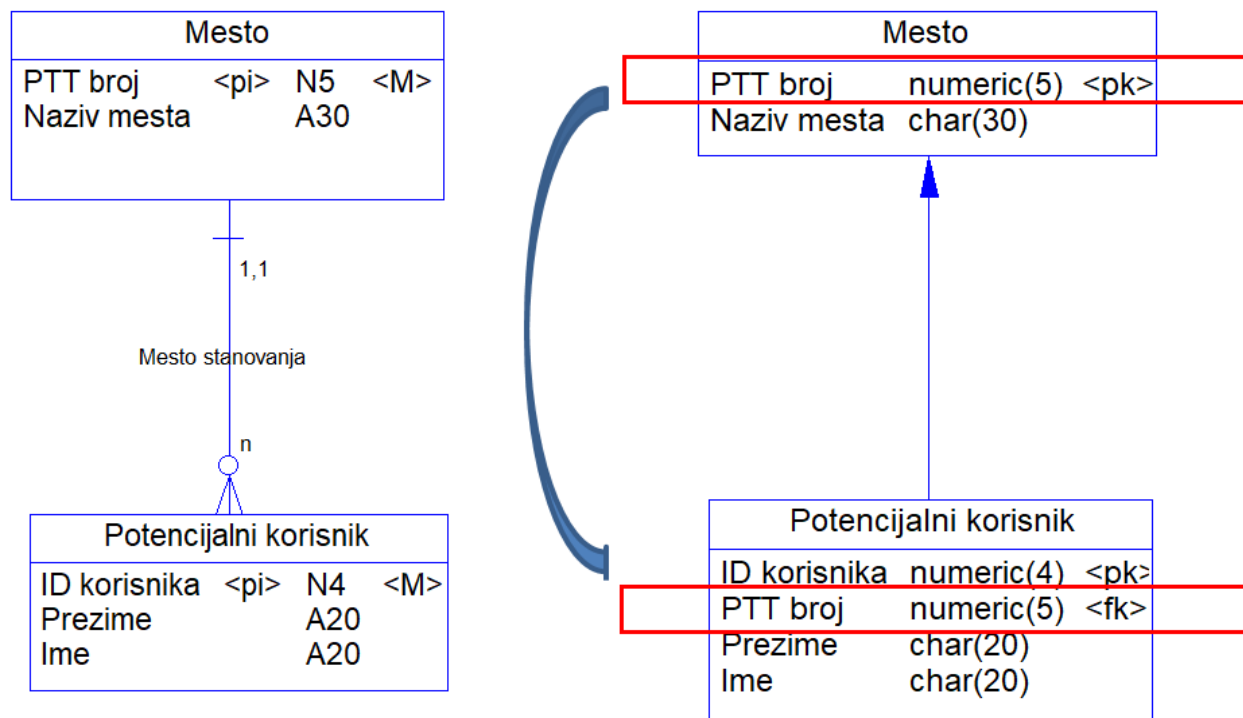


Dijagram klasa

Konceptualni

Fizički

Veza koja nije identifikaciona

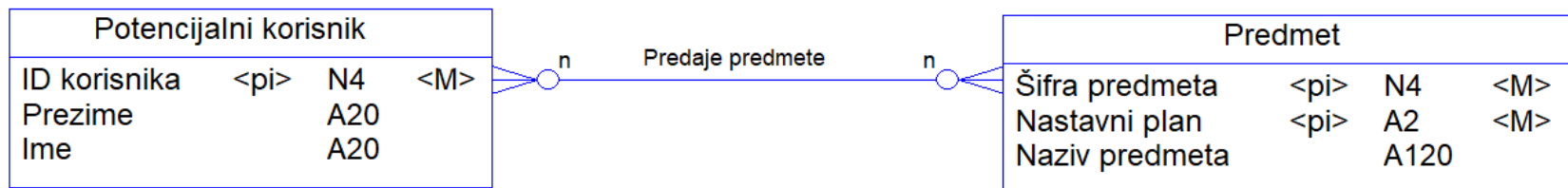


Konceptualni

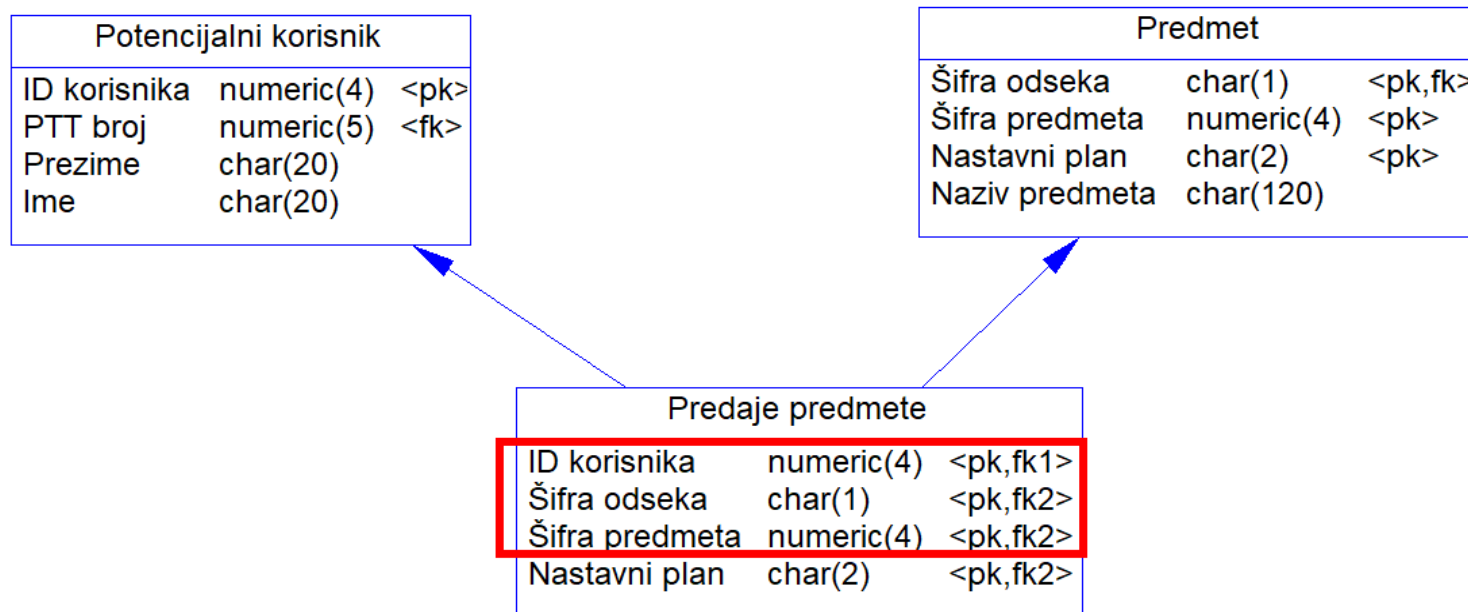
Fizički

Veza “više-na-više”

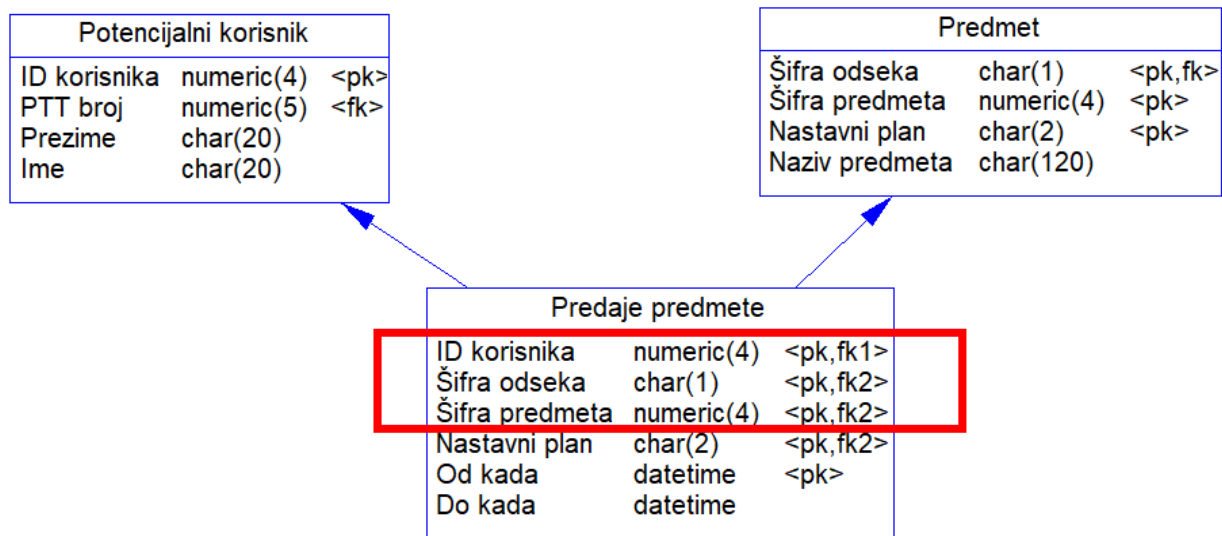
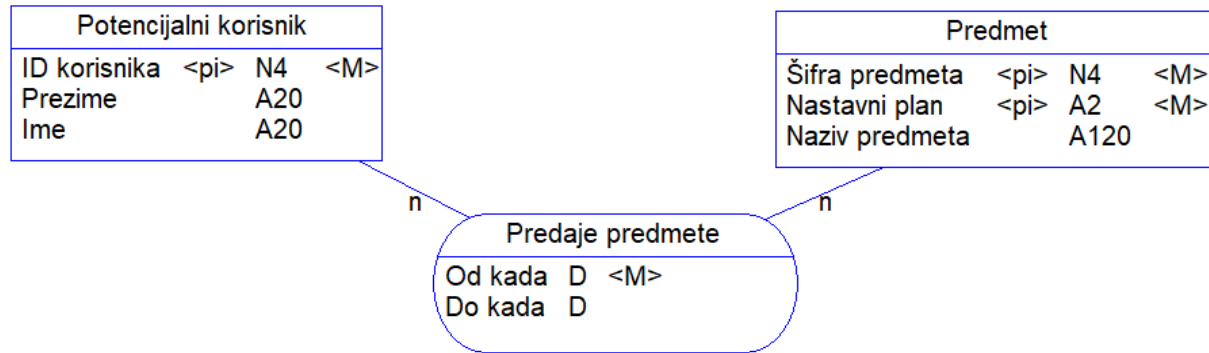
Konceptualni



Fizički



Veza “više-na-više”

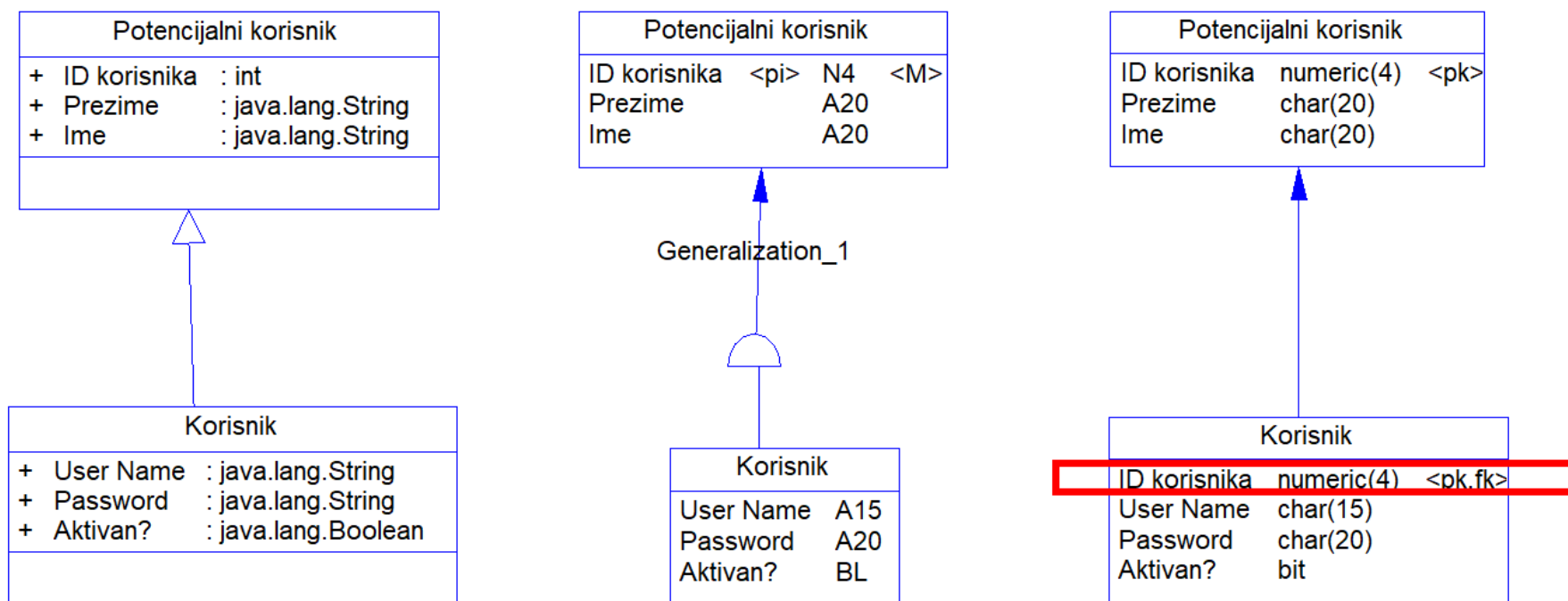


Veza 1-1

- Kada ima smisla koristiti je?
- Da li je ovo dobro?



Veza generalizacije

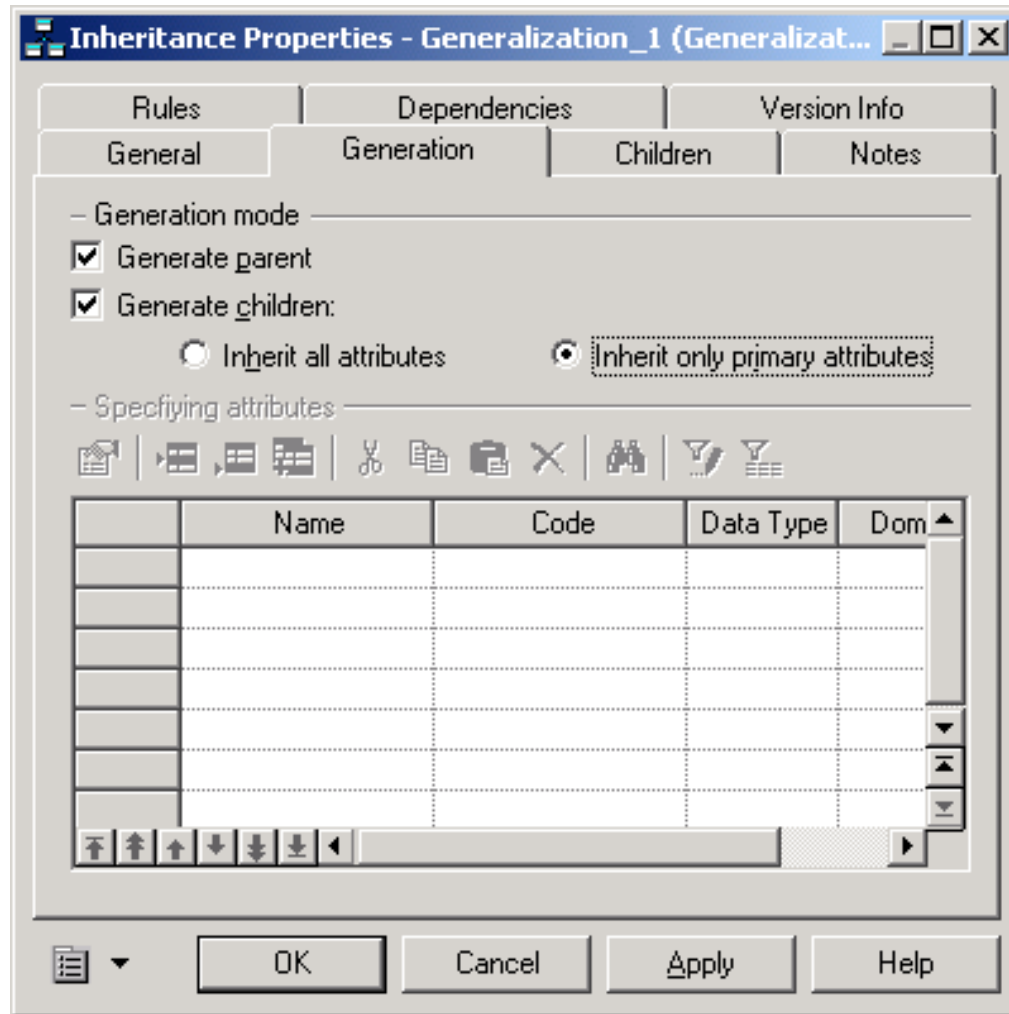


Dijagram klasa

Konceptualni

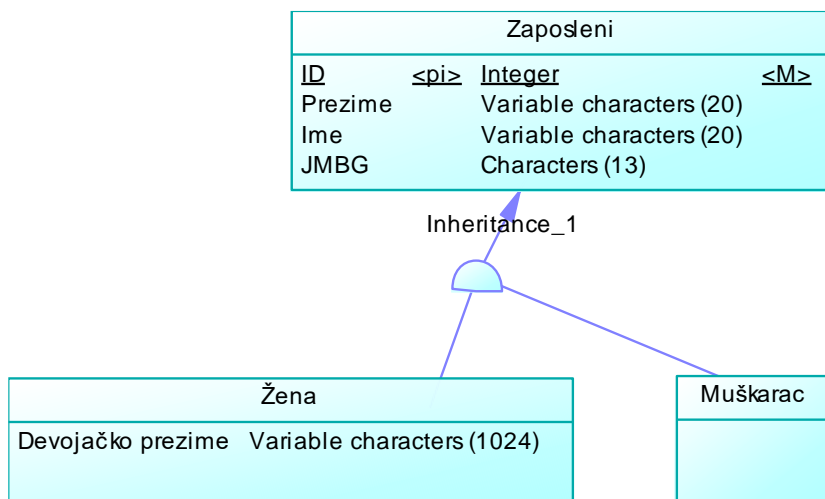
Fizički

PowerDesigner – dva načina za specificiranje generalizacije



Dilema kod modelovanja - primer

Sa nasleđivanjem:

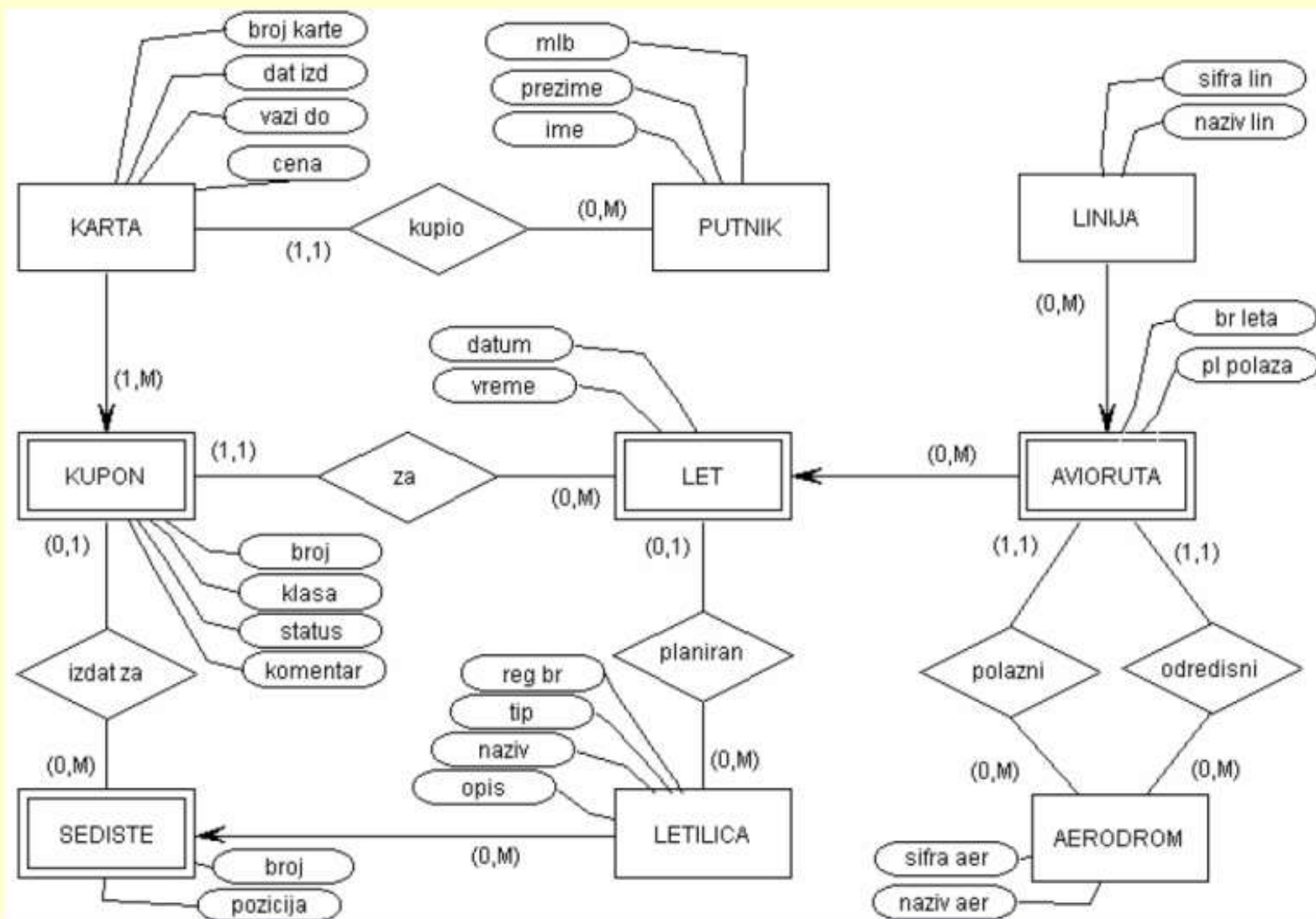


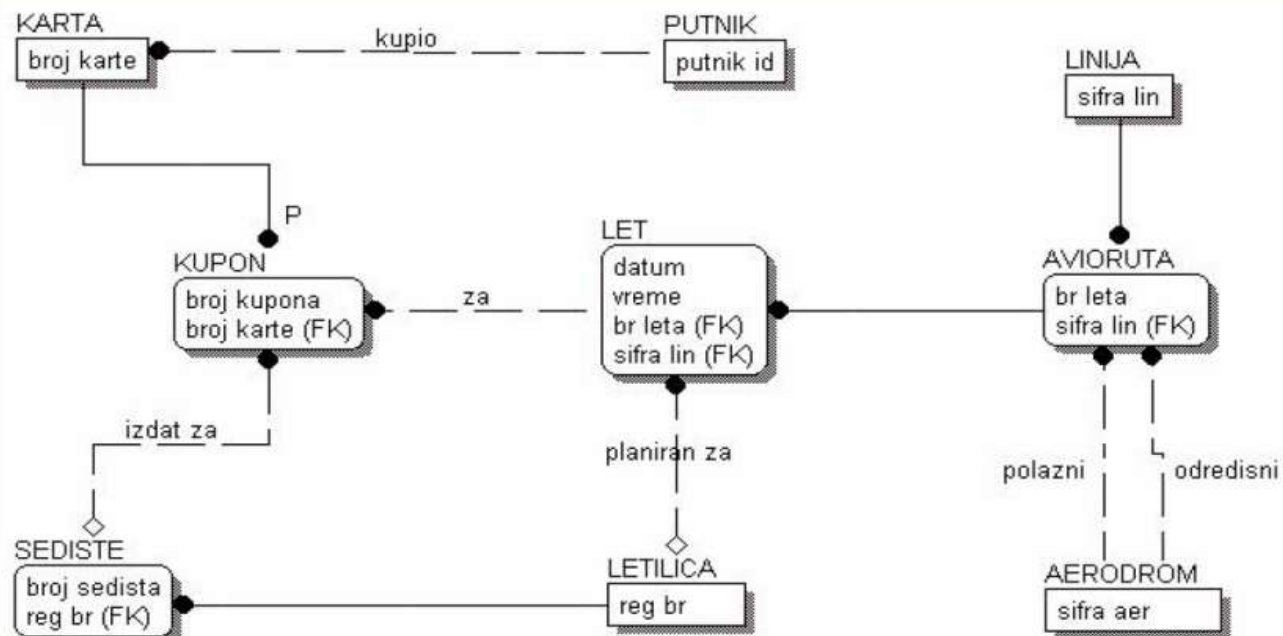
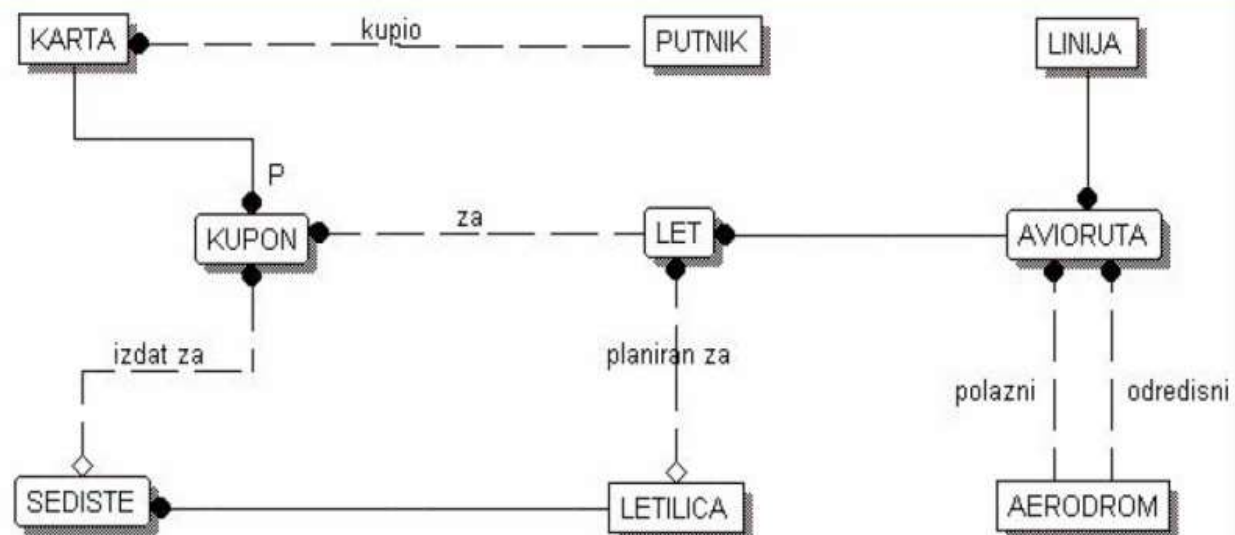
Bez nasleđivanja:

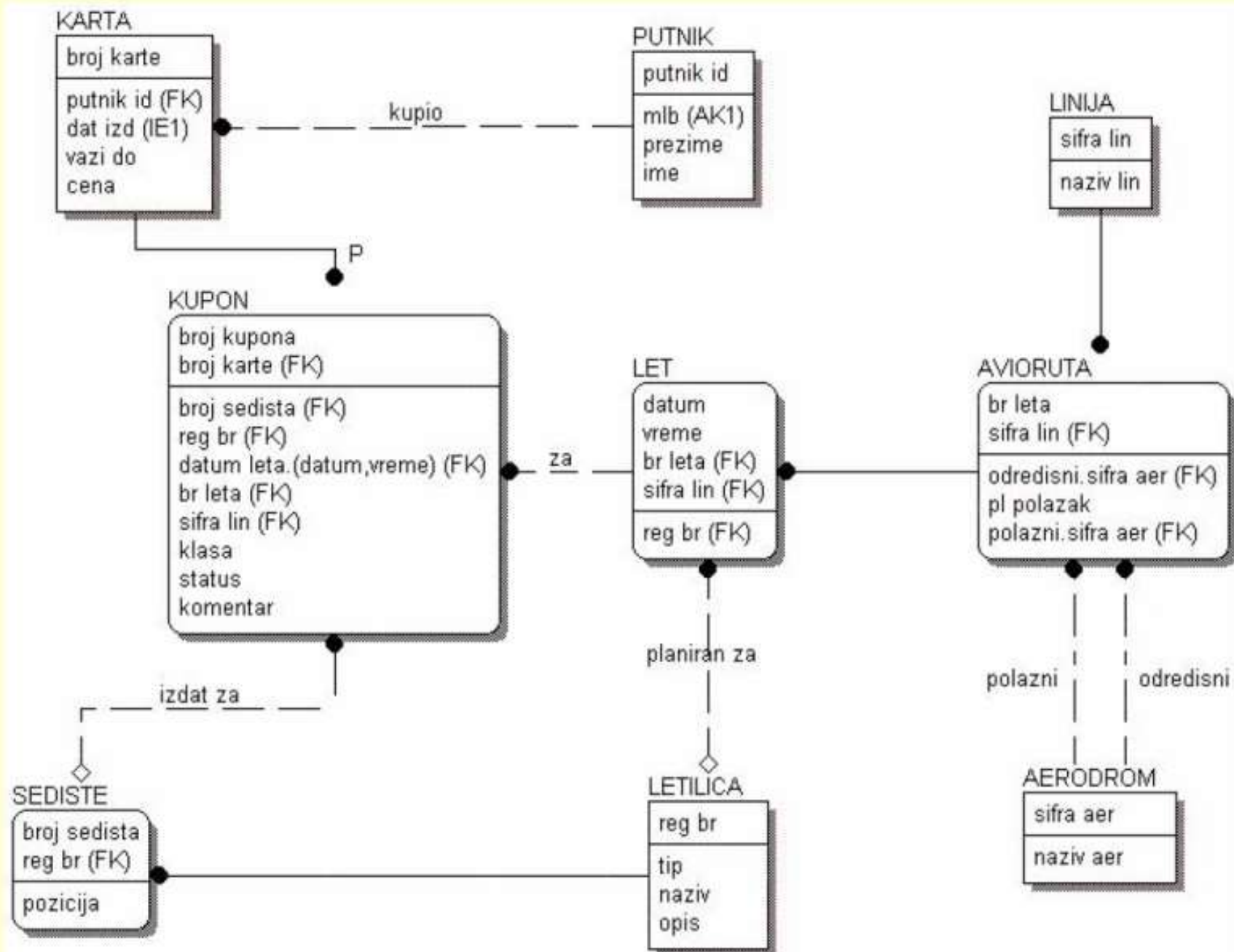
Zaposleni			
<u>ID</u>	<pi>	Integer	<M>
Ime		Variable characters (20)	
Prezime		Variable characters (20)	
JMBG		Characters (13)	
Pol		Characters (1)	
Devojačko prezime		Variable characters (20)	

Dilema kod modelovanja – zaključak

- Ako naslednici nemaju puno atributa, više se “isplati” (sa stanovišta performanse, lakoće implementacije klijentske aplikacije, lakoće korišćenja)
 - sve attribute staviti u jednu klasu/entitet
 - dodati atribut koji omogućava klasifikaciju
 - na prethodnom primeru, to je bio pol







REČNIK PODATAKA

Rečnik podataka - uvod

- Rečnik podataka opisuje **sadržaj i strukturu** svih tokova i skladišta podataka.
- Koncepti:
 1. Polje i domen
 2. Struktura

IspitnaPrijava <brIndeksa, ImeStudenta, NazivPredmeta,
IspitniRok, Datum, Ocena, NazivNastavnika >

Rečnik podataka – Polje i domen

- Polje je elementarna (atomska) struktura koja se dalje ne dekomponuje i ima svoju vrednost.
Na primer: BrIndeksa, Status, Ocena.
- Polja svoje vrednosti uzimaju iz skupova vrednosti koji se nazivaju domenima.

naziv polja : domen [ograničenje]

Rečnik podataka – Polje i domen

- Domeni mogu biti:
 1. “**predefinisani**”- standardni programsko-jezički domen, kao što su: INTEGER, REAL, CHARACTER, DATE i LOGICAL.
 2. “**semantički**”- definišu se posebno preko svoga imena, predefinisanog domena i, eventualno, ograničenja na mogući skup vrednosti predefinisanog domena.

SEMESTRI DEFINED_AS INTEGER (2)

- Dva polja su semantički slična samo ako su definisana nad istim domenom.

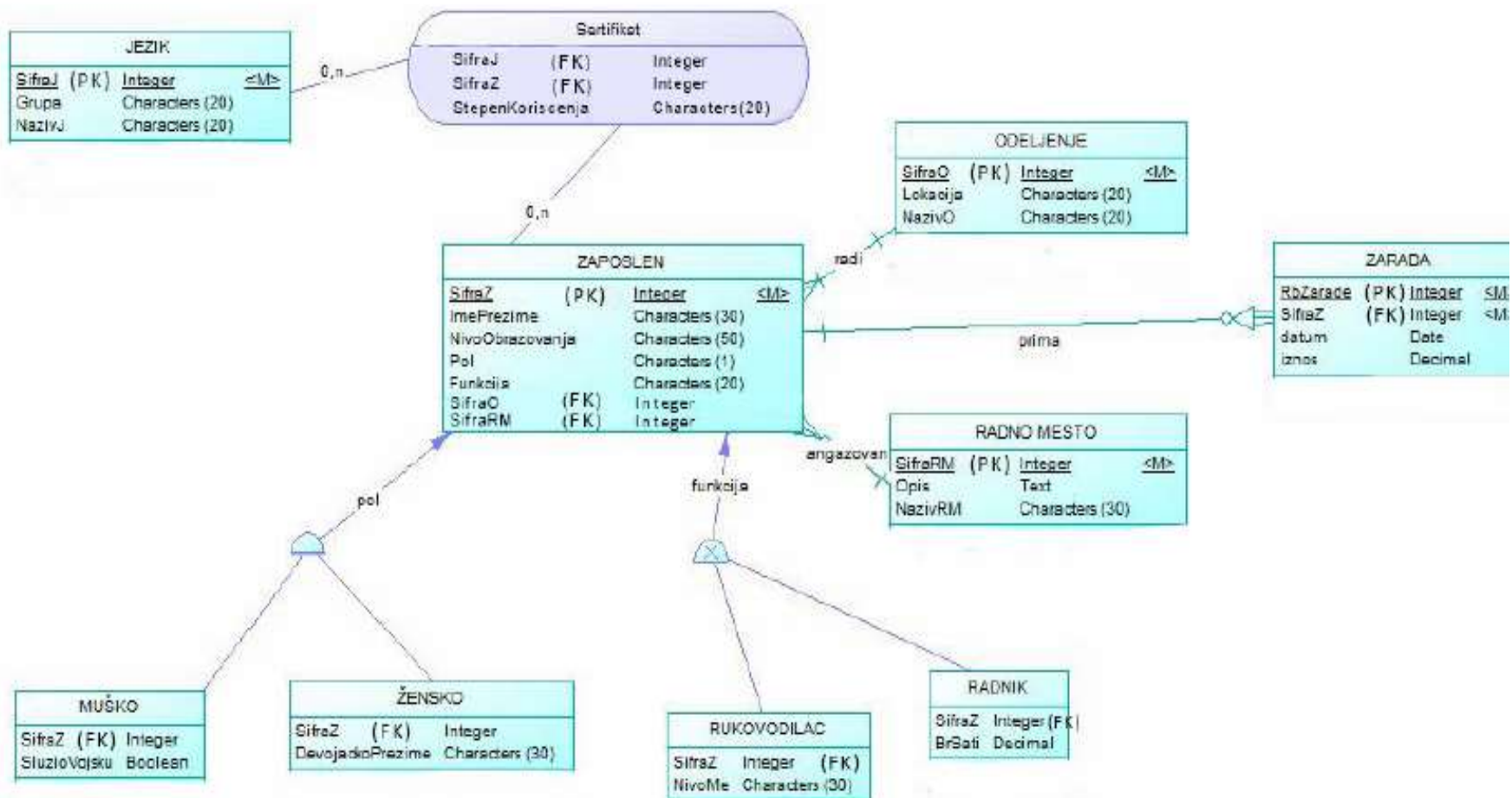
Rečnik podataka – Polje i domen

- Primeri polja i vrednosti koje polje uzima iz nekog domena:
 - NazivPredmeta: CHAR (20)
 - Ocena: INT(2) IN (5,6,7,8,9,10)
 - Prosek: REAL (2,2) < 10.00
 - Semestar: SEMESTRI

MOV PRIMER

MOV - primer

- Verbalni opis:
 - ✓ Po proceduri i politici preduzeća Deko u sektoru za ljudske resurse, između ostalog, vodi se evidencija o zaposlenim osobama. Zaposleni su kategorizovani po polu i funkciji (rukovodilac, radnik, itd.).
 - ✓ Vodi se evidencija o odeljenju u kome zaposleni radi, kao i o radnom mestu na kome je zadužen (radno mesto nije striktno određeno odeljenjem u kome se zaposleni nalazi).
 - ✓ Čuvaju se informacije o isplatama (zaradama) zaposlenih, a bitno je da se zna stepen korišćenja stranog jezika.



KRAJ