

Beogradska akademija poslovnih i umetničkih
strukovnih studija

Projektovanje informacionih sistema

Baze podataka i MOV

dr Rade Matić

Baze podataka

- Proces projektovanja baza podataka je deo procesa razvoja informacionih sistema.
- Odnos projektovanja baze podataka i projektovanje druge komponente IS, skupa aplikacija, zavisi od načina na koji se definiše položaj baze podataka u IS.
- Ovde se mnogo šire govori o projektovanju baza podataka kao relativno nezavisnog resursa nekog IS.

Baze podataka

- Osnovu informacionog sistema čini **baza podataka**, koja se sada može definisati i kao kolekcija međusobno povezanih podataka koja modeluje (prikazuje) objekte, veze objekata i attribute objekata posmatranog realnog sistema.
- Ona zbog toga predstavlja fundamentalne, stabilne, sporo izmenljive karakteristike sistema, objekte u sistemu i njihove međusobne veze.
- Zato se projekat IS mora bazirati na bazi podataka.

Baze podataka

- Ako je baza podataka dobar model stanja realnog sistema, ako programi za održavanje dobro modeluju dejstvo ulaza na stanje realnog sistema, onda će se bilo koja informacija potrebna za upravljanje (izlazi), čak i one unapred nepredviđene, moći dobiti iz IS.
- **Baza podataka** je skup međusobno povezanih podataka o objektima od interesa kojima je moguće pristupati, unositi nove i menjati postojeće podatke i distribuirati ih korisnicima na zahtev.

Baze podataka

- Izučavanju baza podataka može se pristupiti sa dva različita, međusobno povezana aspekta u kojima se one tretiraju bilo kao:
- **sistemi za upravljanje bazom podataka (SUBP)**, specifična tehnologija obrade podataka, odnosno softverski sistem koji obezbeđuje osnovne funkcije obrade velike količine podataka: jednostavno pretraživanje i održavanje, višestruko paralelno (konkurentno) korišćenje istog skupa podataka, pouzdanost i sigurnost, ili kao
- **modeli podataka**, odnosno specifične teorije pomoću kojih se specifikuje i projektuje neka konkretna baza podataka ili informacioni sistem, uopšte.

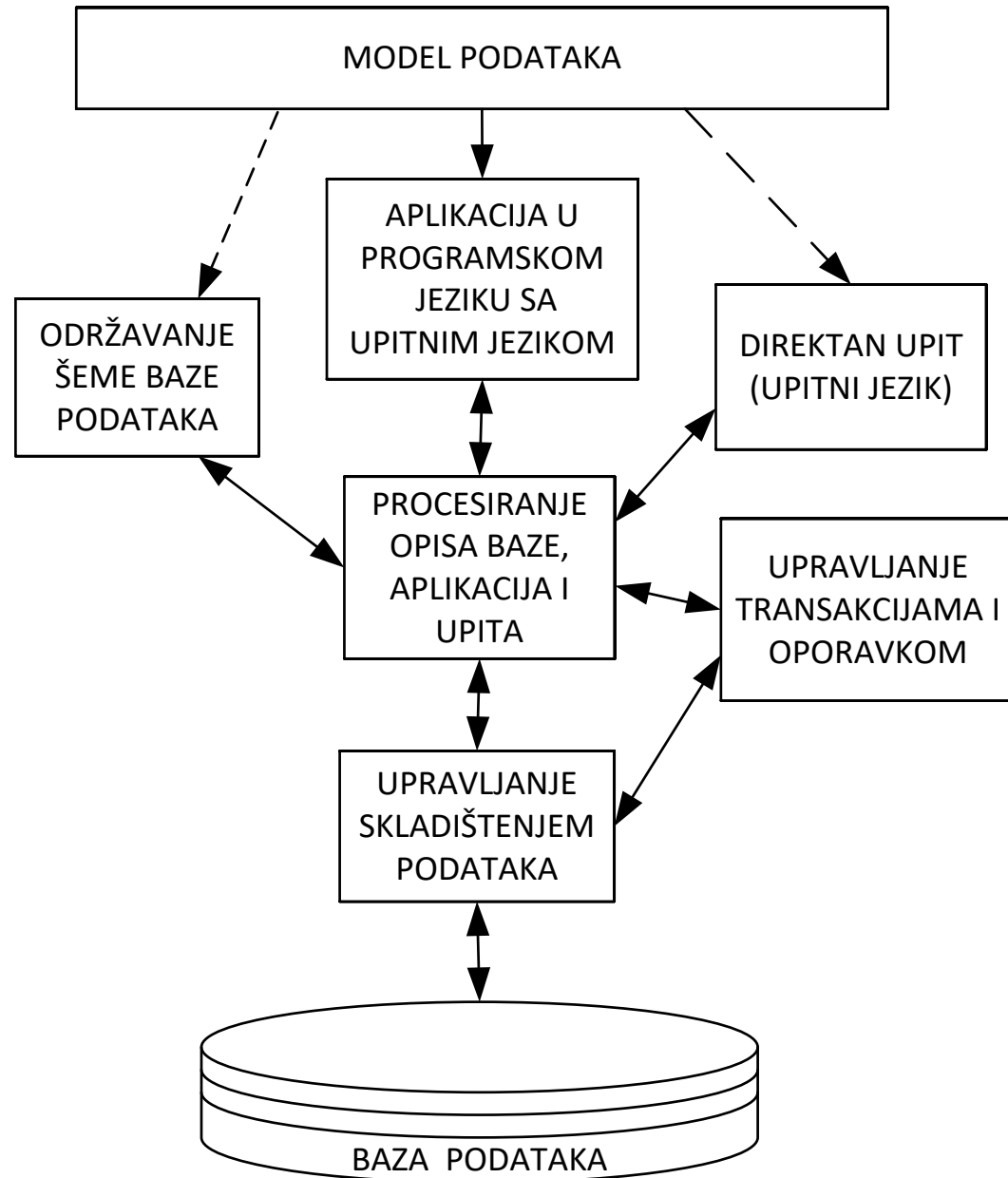
Baze podataka

- SUBP su predstavljali revoluciju u tehnologiji obrade podataka, a modeli podataka revoluciju u metodološkim pristupima razvoju IS.
- **Sistem za upravljanje bazom podataka - SUBP** (Database Management System - DBMS) je specifična vrsta softvera koja omogućava korisnicima da kreiraju, sačuvaju, organizuju i pristupaju podacima iz baze podataka.
- SUBP kao softverski sistem, u razvoju poslovnih informacionih sistema, zamenjuje sistem datoteka (engl. File System) i eliminiše nedostatke ove tehnologije koja je primenjivana u konvencionalnoj obradi podataka.

Baze podataka

- U ovakvoj tehnologiji obrade, podaci su, umesto razbacani po nezavisnim datotekama, organizovani u jedinstvenu bazu podataka. Podaci postaju jedinstveni resurs u nekom sistemu i njima se mora upravljati na jedinstven način.
- Sa stanovišta logičke strukture, baze podataka mogu biti **hijerarhijske, mrežne, relacione i objektno-orijentisane** baze podataka.

Baze podataka: Komponente sistema za upravljanje bazom podataka



Baze podataka: Komponente sistema za upravljanje bazom podataka

1. Na dnu slike je prikazana **baza podataka** koja pored podataka, sadrži i **metapodatke**, odnosno tzv. “**Rečnik podataka**”.

Meta podaci opisuju posmatranu bazu podataka (strukturu baze, pravila očuvanja integriteta podataka, prava korišćenja i slično).

Rečnik podataka je "baza podataka o bazi podataka", pa se taj deo baze podataka naziva i **metabaza** podataka.

Baze podataka: Komponente sistema za upravljanje bazom podataka

Termin **integritet baze podataka** se koristi da označi tačnost (dozvoljene vrednosti), odnosno konzistentnost (dozvoljene odnose) podataka.

Integritet baze podataka može da naruši neka greška u programu ili neki sistemski otkaz.

Termin **prava korišćenja** vezan je za koncept **sigurnosti baze podataka** i odnosi se na zaštitu baze od neovlašćenog korišćenja, modifikovanja, kao i namernog oštećenja ili uništenja podataka.

SUBP održava i **bazu indeksa**. **Indeks**, najopštije, predstavlja strukturu podataka koja omogućava brz pristup “indeksiranim” podacima baze. Najčešća struktura indeksa je B-stablo.

Baze podataka: Komponente sistema za upravljanje bazom podataka

2. Sistem za upravljanje skladištenjem podataka sadrži dve osnovne komponente, **Menadžer baferima** (Buffer Manager) i **Upravljanje datotekama** (File Manager).

Upravljanje datotekama vodi računa o lokaciji datoteka preko kojih se realizuje baza podataka i o pristupima blokovima podataka na zahtev menadžera baferima.

Menadžer baferima prihvata blok podataka sa diska, dodeljuje mu izabranu stranicu centralne memorije, zadržava ga izvesno vreme, u skladu sa ugrađenim algoritmom upravljanja baferima, a zatim vraća na disk oslobađajući dodeljenu mu stranicu.

Baze podataka: Komponente sistema za upravljanje bazom podataka

3. Ulazi u SUBP su:

Upiti, “ad hoc” specifikovani zahtevi za podacima iz baze, sa navedenim uslovima koje podaci treba da zadovolje. Preko upita se može da menja i sadržaj baze podataka;

Aplikacije u nekom programskom jeziku koji se proširuje sa konstrukcijama upitnih jezika, preko kojih se, na unapred definisani način pretražuje i menja sadržaj baze;

Održavanje šeme baze podataka. Šema baze podataka opisuje strukturu baze podataka, pravila integriteta i prava korišćenja. Održavanje šeme baze podrazumeva prvo kreiranje, a zatim i modifikovanje ovog opisa koji se čuva u Rečniku podataka.

Sva tri navedena ulaza se realizuju preko tzv. **Jezika baze podataka** (Database Language).

Baze podataka: Komponente sistema za upravljanje bazom podataka

Jezik baze podataka čine dve osnovne komponente:

Jezik za opis podataka - JOP (Data Definition Language - DDL) koji se koristi za definisanje tabela, podataka, veze među njima na logičkom nivou i za održavanje šeme baze podataka,

Jezik za manipulaciju podataka - JMP (Data Manipulation Language - DML) preko koga se realizuju upiti i modifikacija baze podataka.

Jezik za manipulaciju se koristi za dodavanje, brisanje, izmenu i učitavanje podataka iz baze.

Koriste ga programeri i neposredni korisnici za interaktivno pretraživanje baze podataka.

Baze podataka: Komponente sistema za upravljanje bazom podataka

Jezici baze podataka su ili potpuno neproceduralni (relacione baze) ili u sebi sadrže proceduralne delove.

Neproceduralni jezici sadrže konstrukcije preko kojih se samo specificiraju uslovi koje treba da zadovolji željeni rezultat, a ne i procedura pomoću koje se dobija taj rezultat.

Neproceduralni jezici se često nazivaju i **upitni jezici** zato što im je osnovna namena specificovanje upita.

Baze podataka: Komponente sistema za upravljanje bazom podataka

Zbog toga je osnovni zadatak *Procesora upita (Query Processor)* da transformiše neproceduralni iskaz u sekvencu akcija koje treba da obavi Sistem za upravljanje skladištenjem podataka.

Najznačajniji i najsloženiji deo ove transformacije je **optimizacija upita**, odnosno nalaženje najpogodnije procedure (sekvence akcija) za realizaciju neproceduralnog iskaza.

Navedena transformacija koristi podatke iz Rečnika podataka: opis strukture, pravila integriteta, prava pristupa i definiciju indeksa.

Najpoznatiji upitni jezik je SQL (*Structured Query Language*), standardni jezik baze podataka za relacione baze podataka.

Baze podataka : Komponente sistema za upravljanje bazom podataka

Transakcija je niz operacija nad bazom podataka koja odgovara jednoj logičkoj jedinici posla (Logical Unit of Work – LUW) u realnom sistemu.

Baze podataka : Komponente sistema za upravljanje bazom podataka

4. Upravljanje transakcijama i oporavkom treba da obezbedi da baza podataka ostane u konzistentnom stanju u **konkurentnoj obradi** podataka, čak i pri veoma ozbiljnim otkazima sistema.

S obzirom da se jedan podatak u bazi podataka, po pravilu, pamti samo jednom, više programa istovremeno (konkurentno) zahteva pristup istim podacima.

SUBP mora obezbediti da se konfliktni zahtevi i neželjeni efekti programa, do kojih u ovoj situaciji može da dođe, uspešno razreše.

SUBP takođe treba da, preko periodičnog kopiranja baze podataka na "arhivske memorije" i registrovanja promena koje su se između dva kopiranja dogodile, omogući efikasan **oporavak baze podataka** posle nekog otkaza.

Baze podataka : Komponente sistema za upravljanje bazom podataka

Transakcija u izvršenju mora da ima tzv. ACID osobine (početna slova engleskih reči):

1. Atomnost (engl. Atomicity) - Skup aktivnosti nad bazom podataka koje se izvršavaju po principu "sve ili ništa" (ili su sve aktivnosti uspešno obavljene ili je baza podataka ostala nepromenjena) je atomski skup aktivnosti.

Očigledno je da transakcija, "logička jedinica posla", mora predstavljati atomski skup aktivnosti.

Baze podataka : Komponente sistema za upravljanje bazom podataka

Da bi se ostvarila atomnost transakcije definišu se dve specifične operacije nad bazom podataka:

COMMIT koja označava uspešan kraj transakcije i koja potvrđuje sve promene u bazi koje je posmatrana transakcija proizvela;

ROLLBACK kojom se poništavaju efekti svih prethodnih operacija nad bazom podataka u jednoj transakciji, ako ona, zbog predviđene ili nepredviđene greške (otkaza sistema) može da dovede bazu podataka u nekonzistentno stanje.

Baze podataka : Komponente sistema za upravljanje bazom podataka

2. *Konzistentnost* (engl. Consistency). Očigledno je da se transakcija može definisati i kao jedinica konzistentnosti baze podataka: pre početka i posle okončanja transakcije stanje **baze podataka mora da zadovolji uslove konzistentnosti (dozvoljene vrednosti)**.

Izvršenje transakcije treba da prevede bazu podataka iz jednog u drugo konzistentno stanje.

Baze podataka : Komponente sistema za upravljanje bazom podataka

3. *Izolacija* (engl. Isolation). Kada se dve ili više transakcija izvršavaju istovremeno, njihovi efekti moraju biti međusobno izolovani.

Drugim rečima efekti koje izazovu transakcije koje se obavljaju istovremeno moraju biti jednaki efektima nekog njihovog serijskog (jedna posle druge) izvršenja.

Po pravilu, transakcija ne treba svoje promene baze podataka da učini vidljivim drugim transakcijama pre nego što se ona okonča, odnosno promene potvrde u bazi podataka.

Baze podataka : Komponente sistema za upravljanje bazom podataka

4. *Trajnost* (engl. Durability). Kada se transakcija završi njeni efekti ne mogu biti izgubljeni, čak i ako se neposredno po njenom okončanju desi neki ozbiljan otkaz sistema.

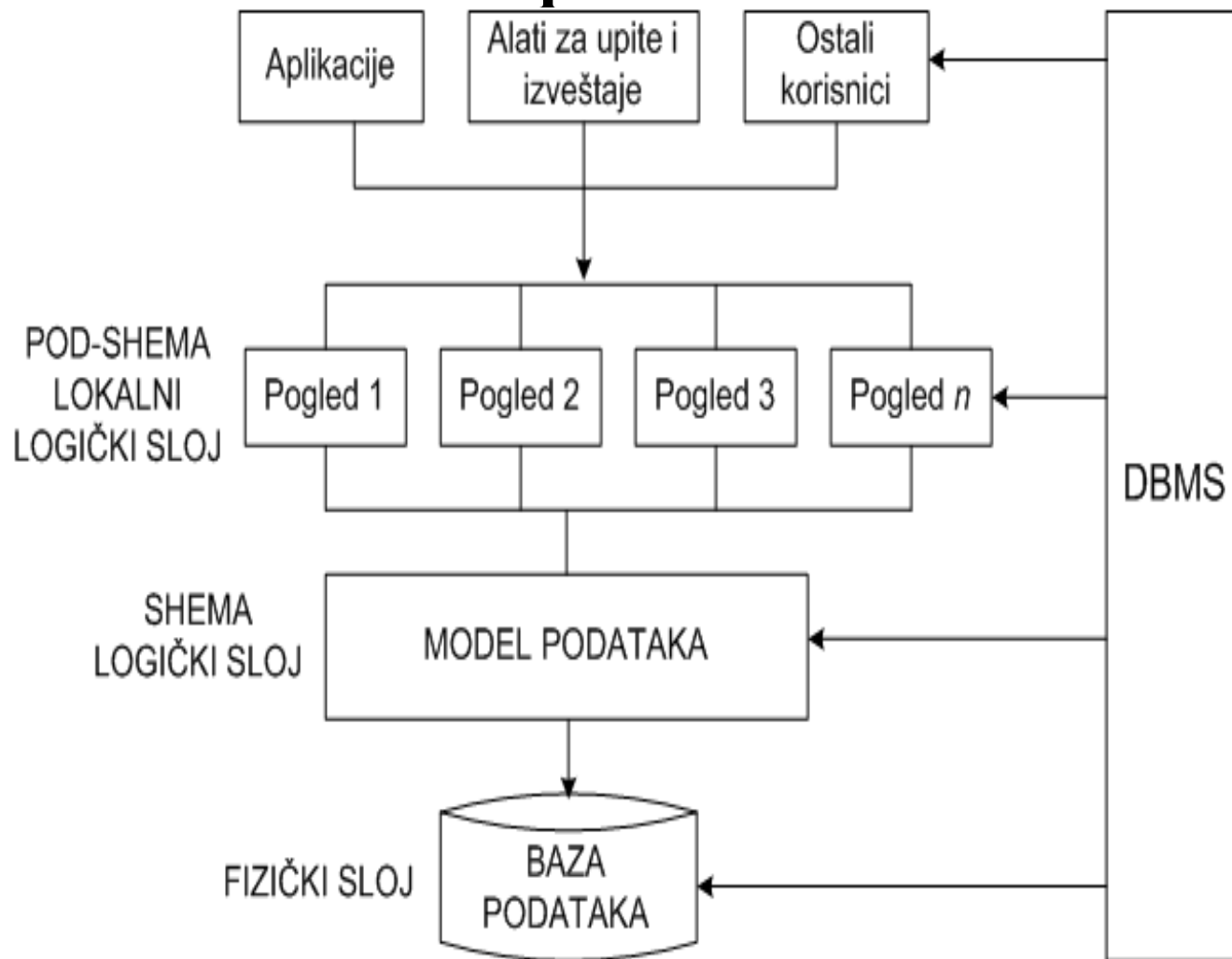
Kada su u bazi potvrđene promene koje je izvršila neka transakcija, one se više ne mogu izgubiti.

Baze podataka : Komponente sistema za upravljanje bazom podataka

Da bi se **ACID** osobine transakcije obezbedile, skup instrukcija koje predstavljaju transakciju počinje, po pravilu, sa specijalnom instrukcijom `BEGIN TRANSACTION`, a završava se bilo sa instrukcijom `COMMIT` (potvrđuju se promene u bazi podataka, ako su sve instrukcije transakcije uspešno izvršene) ili instrukcijom `ROLLBACK` (poništavaju se promene u bazi, ako sve instrukcije nisu uspešno završene).

Instrukcija `ROLLBACK` može da bude i implicitna, inicira je `SUBP` kada, iz bilo kog razloga, dođe do nekog neplaniranog završetka transakcije.

Arhitektura baze podataka



Arhitektura baze podataka se sastoji od tri sloja:

1. fizičkog (interni pogled na bazu),
2. logičkog (konceptualni pogled na bazu ili šema baze podataka) i
3. lokalnog logičkog (eksterni korisnički pogled) sloja.

Arhitektura baze podataka

- **Fizički sloj** se odnosi na organizaciju i strukturu podataka na fizičkim jedinicama spoljne memorije, kao što su diskovi, i njega koriste programeri koji razvijaju SUBP.
- Na **logičkom nivou** podaci su prikazani onako kako ih vide projektanti i administratori baze podataka. Logička struktura baze se naziva šema baze podataka. U relacionim bazama podataka šema definiše tabele i polja u svakoj tabeli i veze između polja i tabela. Šema se generalno čuva u rečniku podataka.
- Na **lokalnom logičkom nivou** delovi baze su prikazani onako kako ih vide korisnici ili aplikacije, odnosno programeri. Zapis jedne logičke definicije zove se pogled (*view*) ili pod-šema.

Modeli podataka

- Informacioni sistem treba da bude model realnog sistema u kome deluje, a razvoj informacionog sistema se u velikoj meri svodi na nalaženje odgovarajućeg modela realnog sistema.
- Intelektualni alati pomoću kojih se opisuje (modeluje) sistem kao skup objekata, njihovih atributa i njihovih međusobnih veza i njegova dinamika (skup operacija), se nazivaju **modeli podataka**.
- **Model podataka** se takođe može definisati i kao specifičan teorijski okvir pomoću koga se specifikuje, projektuje i implementira neka konkretna baza podataka.

Modeli podataka

U modelima podataka koriste se sledeće apstrakcije:

- ***Klasifikacija (tipizacija) objekata, odnosno podataka.***
Slični objekti, odnosno objekti koji imaju isti skup osobina (atributa i veza) i isto dinamičko ponašanje mogu se predstaviti nekim tipom ili klasom objekata.
- ***Generalizacija*** je apstrakcija u kojoj se skup sličnih tipova objekata predstavlja opštijim ***generičkim tipom***, odnosno ***nadtipom***.
- ***Agregacija*** je apstrakcija u kojoj se skup objekata i njihovih međusobnih veza tretira kao novi, jedinstveni, ***agregirani tip***.

Modeli podataka

- MOV je najpopularniji i u praksi najviše korišćeni model za projektovanje baza podataka, prvenstveno relacionih. Struktura MOV je predstavljena preko dijagrama objekti – veze (DOV).
- Postoji više različitih verzija ovog modela, sa gotovo istim konceptima i različitim grafičkim oznakama. Ovde se izlaže verzija koja se delimično razlikuje od originalne notacije tvorca modela Peter Chen iz 1976. god.

Modeli podataka

- Prilikom identifikacije kandidata za objekte polazi se od objekata posmatranja.
- Objekat posmatranja je sve što se može jednoznačno identifikovati, pa samim tim i izolovati iz okoline i opisati.
- Objekat je osoba, stvar, događaj, pojam (realni ili apstraktni) koji je od trajnog interesa za preduzeće, tj. nešto što se želi pojedinačno posmatrati.

Modeli podataka

- Objekti se opisuju preko svojih svojstava, odnosno atributa. Na primer, atributi tipa Radnik su Mlb (matični lični broj), ImeR, Starost itd. Svaki atribut u jednom trenutku vremena ima neku vrednost.
- Atributi uzimaju vrednost iz skupa mogućih vrednosti. Ovi skupovi se nazivaju **domenima**.
- Formalno se atribut objekta može definisati kao preslikavanje iz tipa objekta u domen. Na primer,

Mlb: Radnik ----> Matbr (1,1)

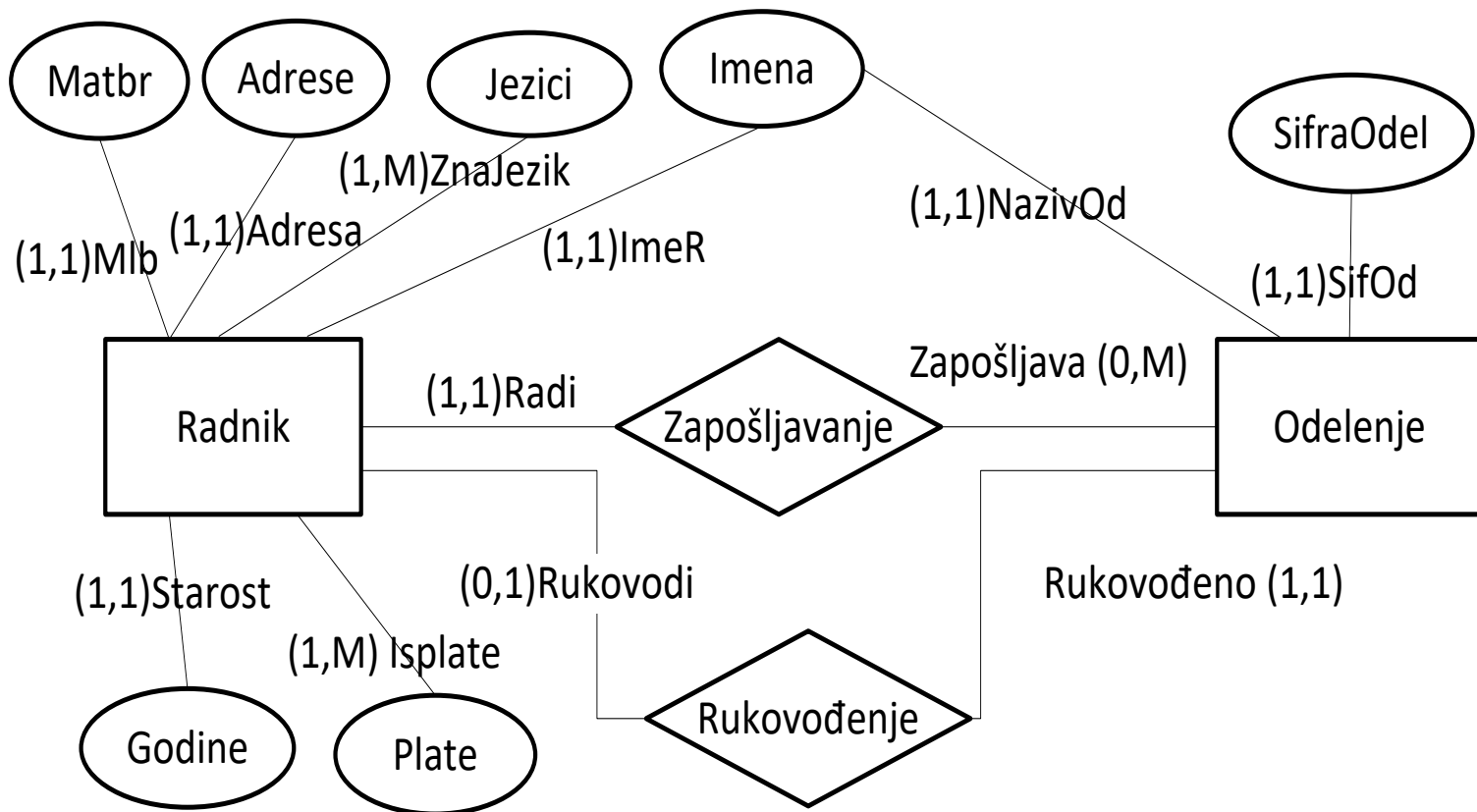
ImeR: Radnik ----> Imena (1,1)

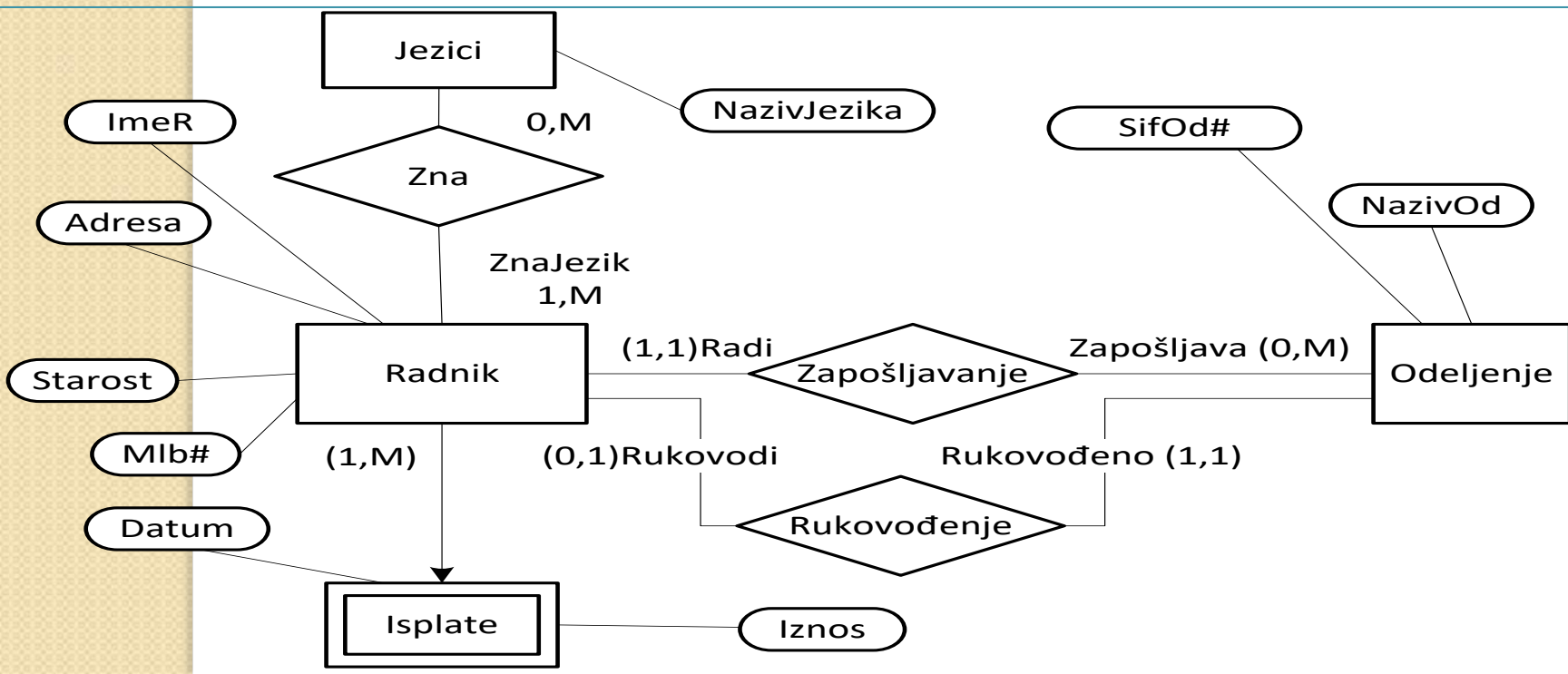
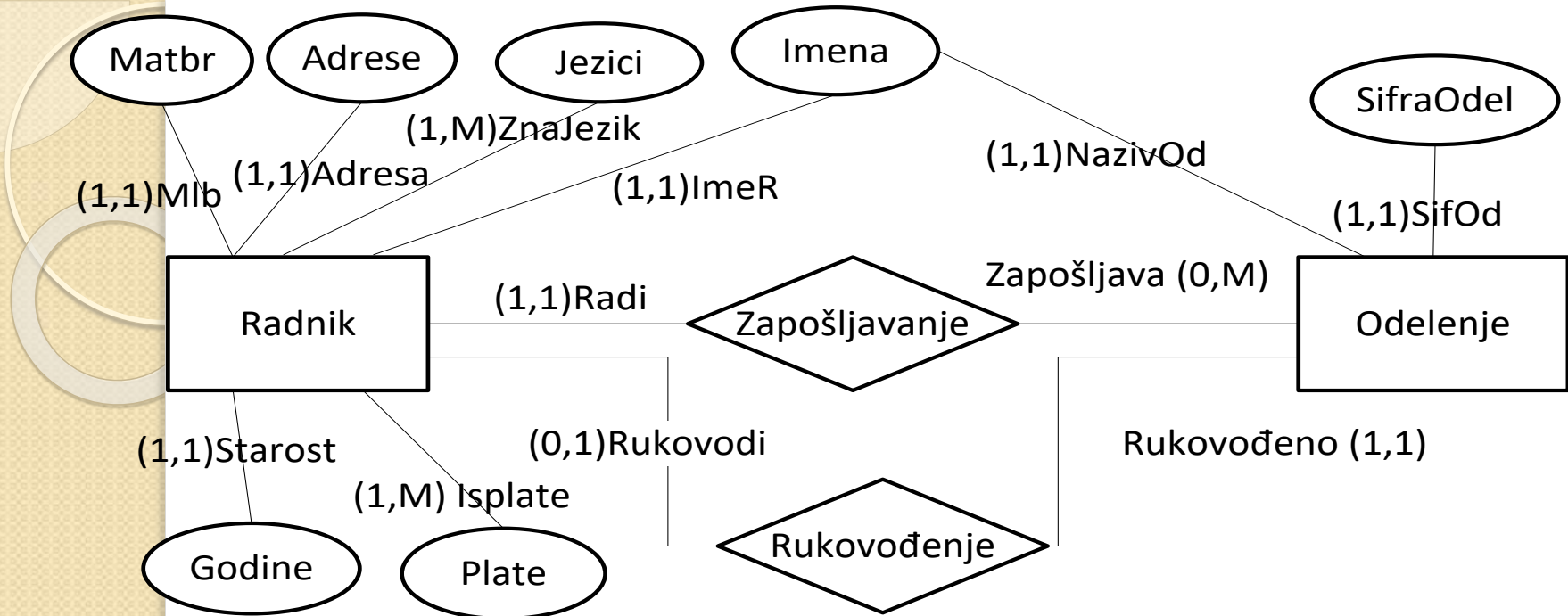
ZnaJezik: Radnik ----> Jezici (1,M)

Modeli podataka

- S obzirom da se atribut definiše kao preslikavanje, može se reći da atribut ima svoju kardinalnost.
- Donja (DG) i gornja granica (GG) prethodno definisanih atributa prikazane su u zagradi.
- Ako je kardinalnost atributa ($DG = 1$, $GG = 1$) onda se takav atribut naziva **jednoznačni atribut** objekata.
- Ako i inverzno preslikavanje jednoznačnog atributa (preslikavanje Domen \rightarrow Tip objekta) ima kardinalnost ($DG = 0$ (ili 1), $GG = 1$), tada se takav atribut naziva **identifikator objekta**, jer jedno pojavljivanje takvog atributa jedinstveno određuje jedno pojavljivanje objekta u tipu.
- Ako je gornja granica kardinalnost atributa $GG = M$, onda se takav atribut naziva **višeznačni atribut** objekta.

Modeli podataka





UML 2.0

ATRIBUT	DOMEN	OGRANIČENJE
Mlb	long	NotNull And Substring (1,2) Between 1,31 And Substring (3,4) Between 1,12
NazivJezika	string	In (Srpski, Ruski, Engleski, Nemački)
Adresa	string	
Datum	date	
Starost	short	Between 15,65
SifOd	short	Modul_11

Atributa sa ograničenjima

Modeli podataka

- **Nezavisni tip objekat ili jak tip objekta** je objekat koji ima jednu osobinu koja ga može jednoznačno identifikovati, tj. nezavisni objekti imaju vlastitu identifikaciju (ne zavise od drugih objekata).
- Grafički se nezavisni tip objekat prikazuje pravougaonikom u okviru koga se upisuje naziv tipa objekta u jednini.

Modeli podataka

- **Zavisni tip objekat** je objekat čija egzistencija i identifikacija zavisi od drugog ili drugih tipova objekata. Dele se na:
 - **karakteristične**, tj. objekte koji se ponavljaju više puta za određeni nezavisni objekat;
 - **asocijativne** objekte, koji predstavljaju vezu više objekata;
 - **projektne**, koji su slični asocijativnim objektima, samo što nemaju sopstvene attribute;
 - objekte **kategorije**, koji predstavljaju podkategoriju objekata.

Modeli podataka

- ***Karakteristični tip objekta ili slab tip objekta*** predstavlja grupu atributa koji se ponavljaju više puta za jedan tip objekta i koji se identifikuju preko nezavisnog tipa objekta. Npr. to su tipovi objekata *Konkurs* i *Intervju*. Za tip objekta *Intervju* se kaže da je karakterističan tip objekta, jer zavisi od tipa *Konkurs*.
- ***Asocijativni objekti*** su sastavljeni od više veza između dva ili više tipova objekata. Npr. *Prijava* je asocijativni objekat koji opisuje vezu između objekata: *Konkurs* i *Kandidat*. Asocijativni objekti nose informaciju o višeznačnoj vezi, kao što je atribut *DatumPrijava*.

Modeli podataka

- **Projektni objekat** (*designative*) sličan je asocijativnom objektu, s tim što nema sopstvene attribute. Videti sliku gde *Predavac* sastavlja *ProgramObuke* i gde ova veza (tip objekta) ne nosi nikakvu informaciju (atribut), izuzev identifikacionih oznaka za *Predavaca* i *ProgramObuke*.
- **Objekat kategorija** (*category*) je zavisian tip objekta koji ima tzv. vezu tipa podkategoriju (*subcategory*). Kod objekata tipa kategorija definišu se: nadređeni objekat koji ima zajedničke osobine (npr. objekat *Osoba*) i podređeni objekti (objekti: *Student* i *Predavac*) koji se identifikuju ključem nadređenog i poseduju svoje specifične osobine.

Modeli podataka

- ✓ Na osnovu definisanih kandidata za objekte pristupa se identifikaciji mogućih veza između objekata.
- ✓ Veza predstavlja interakciju među objektima, tj. znanje o njihovoj međusobnoj povezanosti. Vezama se definišu zavisnosti između objekata, odnosno uzajamna dejstava objekata.
- ✓ Evo nekoliko načina identifikacija veza, i to:
 - povezivanje objekata na osnovu odgovarajućih interesa;
 - definisanje zavisnosti objekata (gde se definišu nezavisni i zavisni objekti);
 - definisanje značenja zavisnosti (definisanjem referencijalnog integriteta);
 - izvođenje varijacija zavisnosti (nalaženjem optimalne);
 - izbor objekata 'roditelj' i objekata 'dete'.

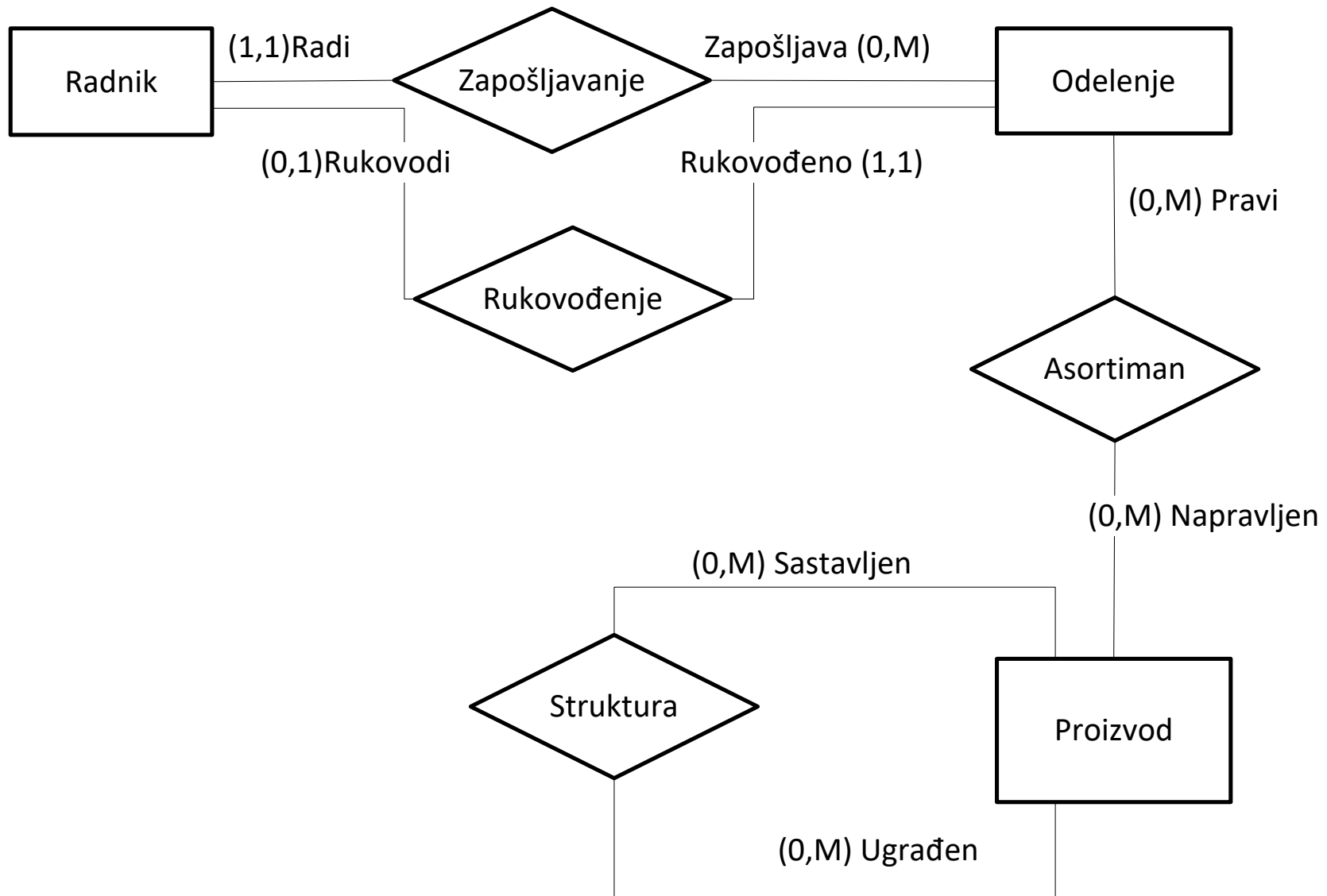
Modeli podataka

- Veze u rečenici su glagolske fraze koje pokazuju kakav je odnos među objektima.
- Iako glagolske fraze ne opisuju pravila precizno, one dozvoljavaju osobi koja posmatra model da stekne osnovni osećaj o povezanosti objekata.
- **Dobra je praksa kad čitanje modela daje valjane rečenice (iskaze).**

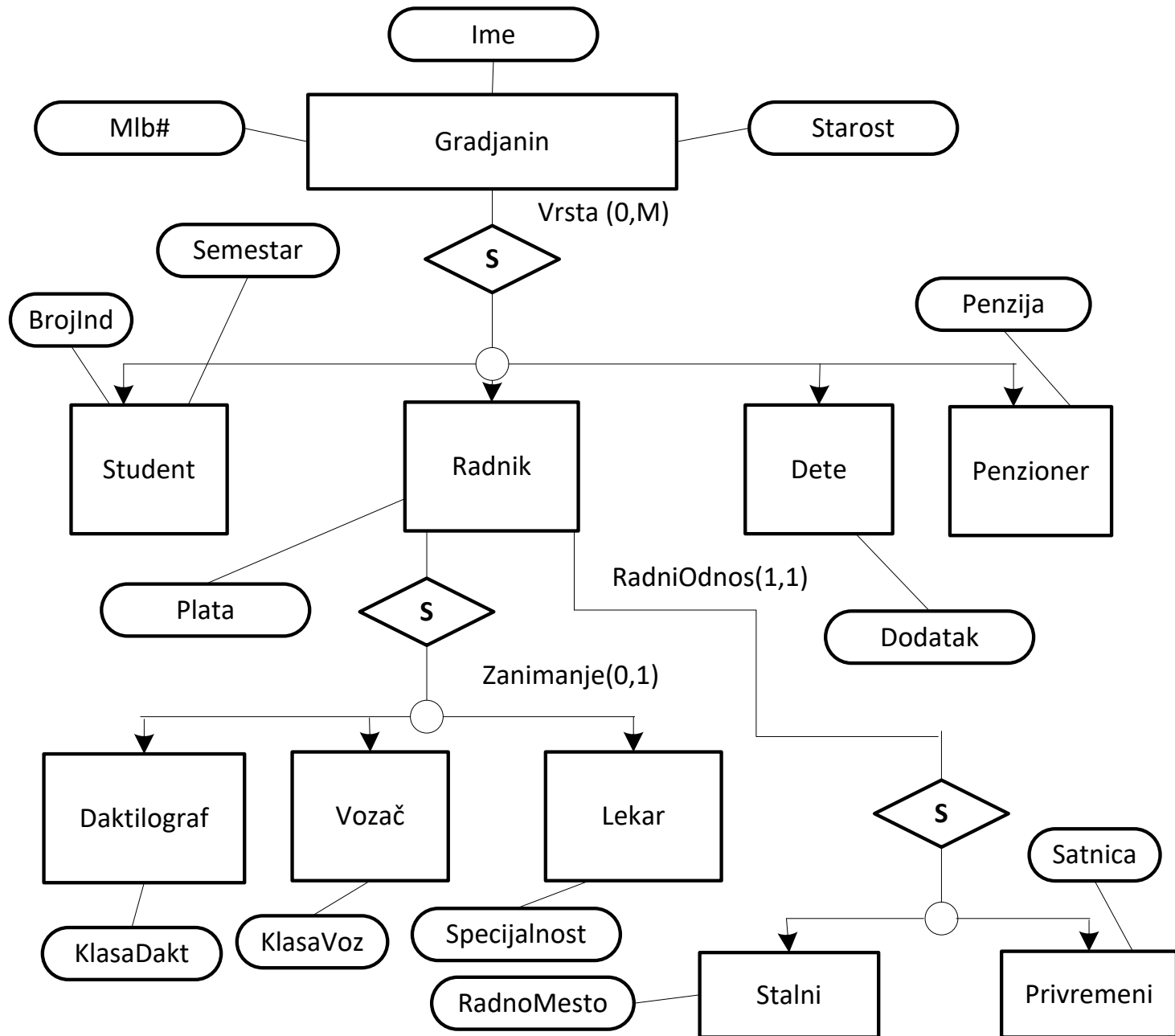
Modeli podataka

- U MOV se direktno predstavljaju samo **binarne veze**, veza između dva objekata. Višestruke veze objekata se predstavljaju posredno, na način koji će biti kasnije opisan.
- Svaki tip **binarne veze** tipova objekata E1 i E2 definiše dva tipa preslikavanja, preslikavanje $E1 \dashrightarrow E2$ (E1 je domen, a E2 kodomen preslikavanja) i inverzno preslikavanje, $E2 \dashrightarrow E1$. Na primer, u tipu veze Zapošljavanje, definišu se sledeća dva preslikavanja:
 - Radi: Radnik \dashrightarrow Odeljenje (sa semantikom "radnik radi u odeljenju")
 - Zapošljava: Odeljenje \dashrightarrow Radnik (sa semantikom "odeljenje zapošljava radnike")

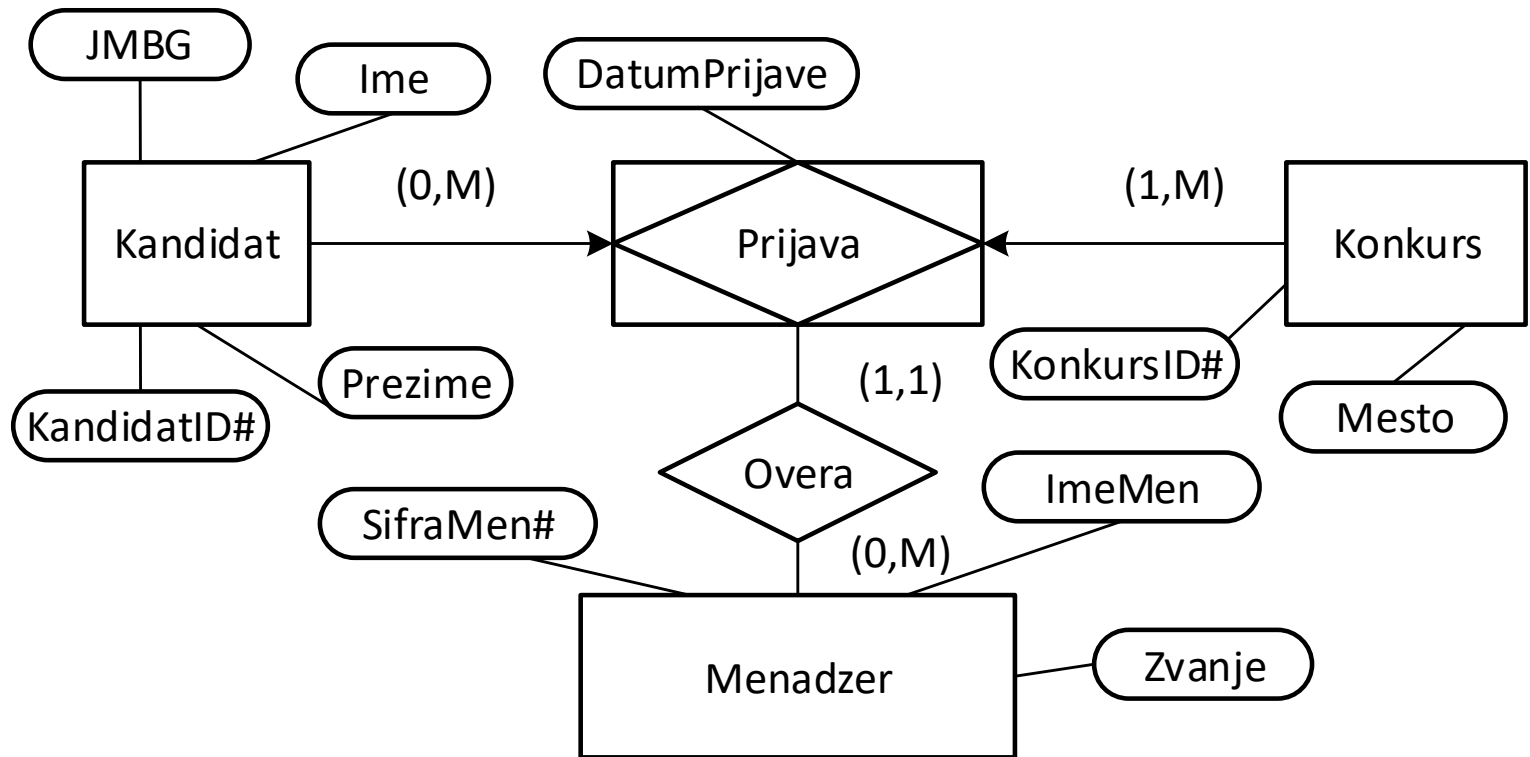
Modeli podataka



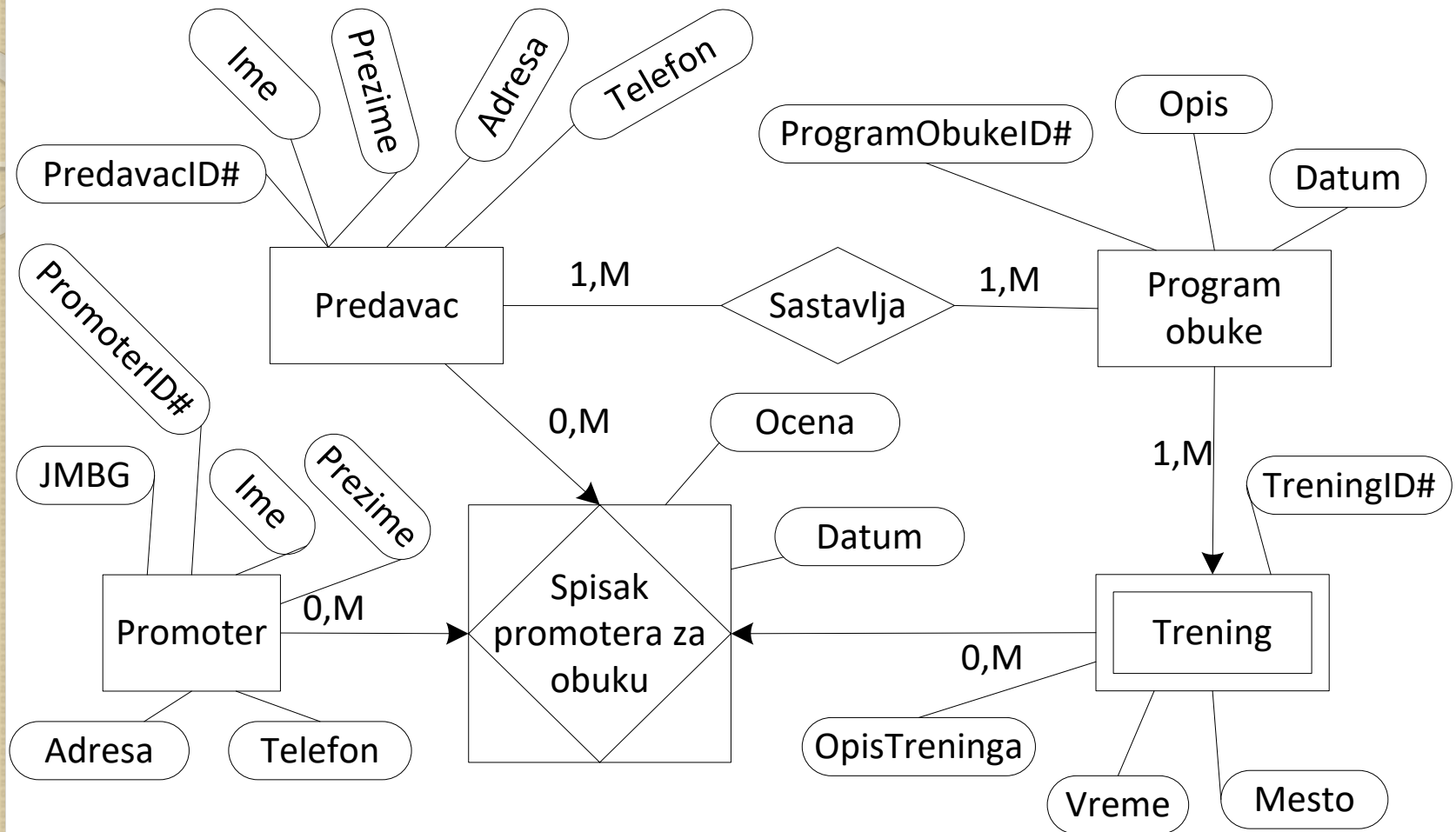
Modeli podataka



Modeli podataka

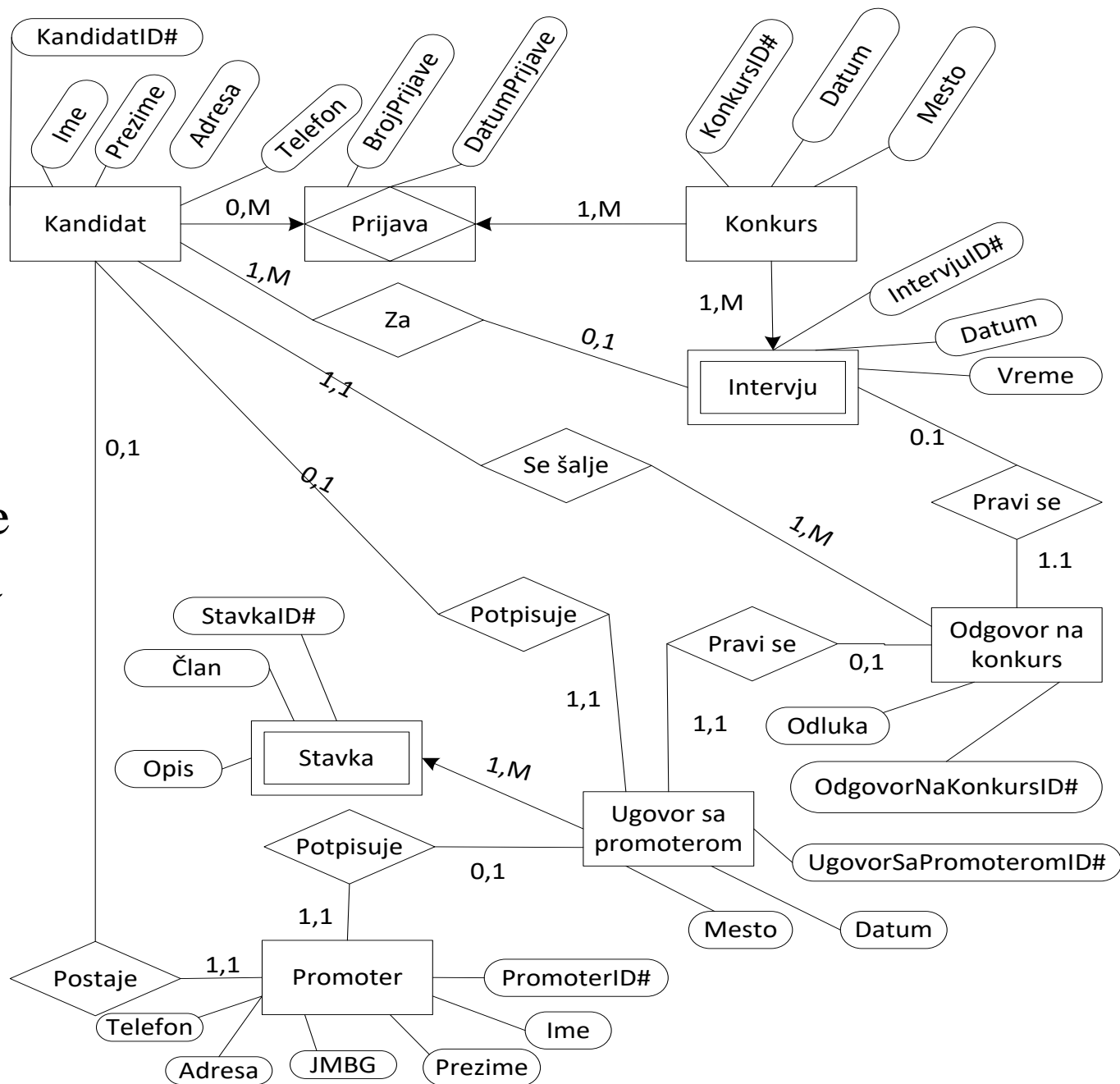


Modeli podataka



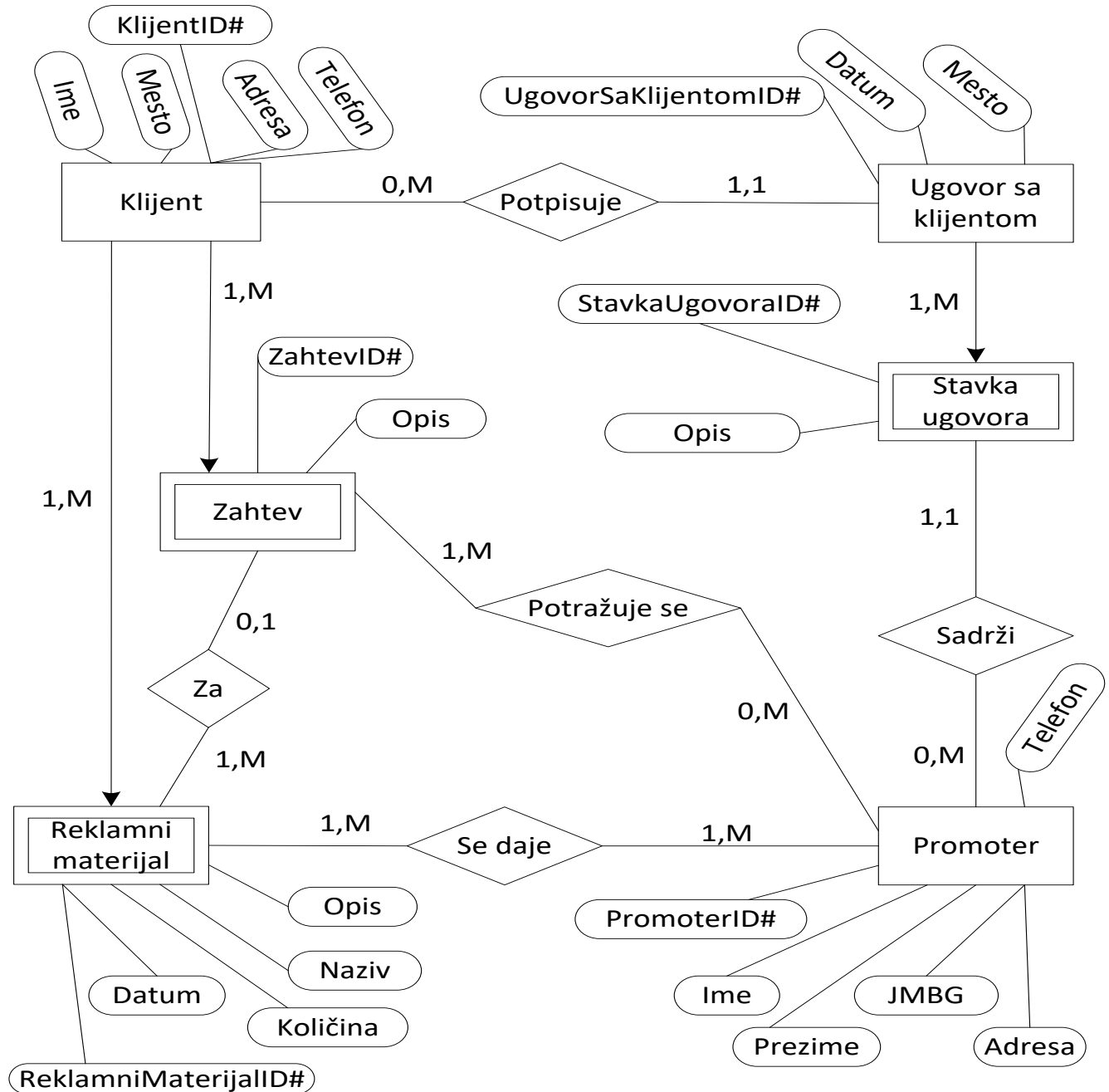
Obuka promotera

Modeli podataka



Konkurisanje
za promotera

Modeli podataka



Angažovanje promotera

Projektovanje informacionih sistema

HVALA !

dr Rade Matić