

**Projektovanje i eksplotacija
baza podataka kroz primjere**

Dražen Brđanin, Slavko Marić, Goran Banjac

**PROJEKTOVANJE I EKSPLOATACIJA
BAZA PODATAKA KROZ PRIMJERE**

1. izdanie

Recenzenti

Prof. dr Ivan Luković

Fakultet tehničkih nauka, Novi Sad

Prof. dr Leonid Stoimenov

Elektronski fakultet, Niš

Lektor

Dalibor Smiljić

Izdavač

**Univerzitet u Banjoj Luci
Elektrotehnički fakultet**

Štampa

MAKOPRINT, Banja Luka

Tiraž

300 primjeraka

CIP - Каталогизација у публикацији
Народна и универзитетска библиотека
Републике Српске, Бања Лука

0004.65(075.8)(076)

БРЂАНИН, Дражен, 1972-

Projektovanje i eksploracija baza podataka kroz primjere / Dražen Brđanin, Slavko Marić, Goran Banjac. - 1. izd. - Banja Luka : Univerzitet u Banjoj Luci, Elektrotehnički fakultet, 2018 (Banja Luka : Makoprint). - X, [333] str. : ilustr. ; 24 cm

Predgovor: str. V-VI. - Tiraž 300. - Prilog: str. [325-332]. - Napomene i bibliografske reference uz tekst. - Bibliografija: str. [333].

ISBN 978-99955-46-34-2

1. Марић, Славко, 1951- [автор] 2. Бањац, Горан, 1986- [автор]

COBISS.RS-ID 7787032

© Elektrotehnički fakultet Univerziteta u Banjoj Luci, 2018.

Sva prava zadržana. Nije dozvoljeno da bilo koji dio ove knjige bude snimljen, emitovan ili reproducovan na bilo koji način, uključujući, ali ne ograničavajući se na fotokopiranje, fotografiju, magnetni ili bilo koji drugi vid zapisa, bez prethodne dozvole izdavača.

**Dražen Brđanin
Slavko Marić
Goran Banjac**

PROJEKTOVANJE I EKSPLOATACIJA BAZA PODATAKA KROZ PRIMJERE

Banja Luka
2018.

Predgovor

Ova knjiga predstavlja pomoćni udžbenik za predmet *Baze podataka* u okviru studijskog programa *Računarstvo i informatika* na prvom ciklusu studija na Elektrotehničkom fakultetu Univerziteta u Banjoj Luci. Knjiga se može koristiti i kao pomoćni udžbenik za predmete u okviru kojih se izučava projektovanje i eksploatacija relacionih baza podataka na drugim studijskim programima na Elektrotehničkom fakultetu. Knjiga može biti korisna i studentima drugih fakulteta za kurseve sličnih nastavnih planova i programa, kao i svima drugima koji žele da učvrste teoretska i praktična znanja kroz raznovrsne aspekte i primjere primjene.

Knjiga je nastala kao rezultat dugogodišnjeg iskustva stečenog u podučavanju projektovanja i eksploatacije relacionih baza podataka na Elektrotehničkom fakultetu u Banjoj Luci, kao i praktičnog iskustva stečenog u industrijskim projektima.

Izložena materija organizovana je u sedam glava i pokriva proces projektovanja i eksploatacije relacionih baza podataka, što uključuje: konceptualno modelovanje, relacioni model, transformaciju konceptualnog modela u relacioni model, DDL komponentu SQL jezika za specifikaciju šeme relacione baze podataka, formalne upitne jezike, DML komponentu SQL jezika za upravljanje podacima u bazi, kao i postupke normalizacije šeme relacione baze podataka.

U prvih šest glava prikazani su relevantni teoretski koncepti, koji su ilustrovani sa preko 250 praktičnih primjera projektovanja šema relacionih baza podataka, i zatim, upravljanja relacionim bazama podataka. Posljednja, sedma glava sadrži integralne primjere projektovanja šeme i upravljanja bazom podataka, koji predstavljaju ogledne primjere za pripremanje ispita iz predmeta *Baze podataka*.

Na početku svake glave dat je kratak sažetak najvažnijih teoretskih pojmoveva i koncepata iz odnosne oblasti. Redoslijed primjera u svakoj glavi u saglasnosti je sa redoslijedom izučavanja i izlaganja gradiva u okviru predmeta *Baze podataka*. Primjeri su prezentovani na način da se postupno uvode i rješavaju novi problematski aspekti, uz sukcesivno povećanje kompleksnosti problema i izloženih rješenja. Postupak rješavanja svakog novog aspekta problema detaljno je opisan, a pri rješavanju složenijih problema prethodno opisani postupci tretirani su kao poznati.

Rješavanje praktičnih zadataka projektovanja baze podataka i manipulacije podacima je najvećim dijelom bazirano na primjeru pojednostavljenog informacijskog sistema univerziteta, što studentima olakšava razumijevanje materije koja se prezentuje. Pored primjera koji se odnose na pojednostavljeni informacioni sistem univerziteta, dati su primjeri projektovanja baze podataka i manipulacije podacima i za druge pojednostavljene realne sisteme (biblioteka, sistem otkupnih stanica određenih proizvoda, sportska takmičenja u različitim disciplinama i kategorijama, izborni sistem i dr.), kako bi studenti mogli da vježbaju i provjere sposobnost uočavanja, prepoznavanja i povezivanja realnih i teoretskih koncepata u različitim aplikativnim situacijama.

Pored generičkih i teoretskih modela i koncepata (model objekti-veze, relacioni model, formalni upitni jezici, normalizacija) u knjizi je prezentovano korištenje konkretnih i široko zastupljenih softverskih alata i sistema za projektovanje baze i rad sa podacima (ERwin alat za konceptualno modelovanje bazirano na IE notaciji, UML konceptualno modelovanje te MySQL DBMS), kako bi se čitaocu što više približili svi praktični aspekti rada sa relacionim bazama podataka.

Na kraju, posebnu zahvalnost izražavamo recenzentima knjige, prof. dr Ivanu Lukoviću i prof. dr Leonidu Stoimenovu, koji su korisnim sugestijama značajno doprinijeli unapređenju rukopisa.

Zahvalnost dugujemo i svim našim studentima koji su pomogli u otkrivanju grešaka u rukopisu, kao i lektoru Daliboru Smiljiću.

Banja Luka, septembar 2018.

Autori

Sadržaj

1. Konceptualno modelovanje	1
1.1. Uvod	1
1.2. Model Objekti-Veze	1
1.2.1. MOV dijagram	4
1.2.2. Proces projektovanja konceptualnog modela	5
1.3. IE notacija	35
1.4. UML dijagram klasa	77
2. Relacioni model	85
2.1. Osnovne definicije i notacija	85
2.1.1. Relacija	85
2.1.2. Osnovna notacija i konvencije	85
2.1.3. Specifikacija ograničenja u relacionom modelu	86
2.2. Šema relacione baze podataka	87
2.2.1. UML reprezentacija šeme relacione baze podataka	88
2.3. Transformacija konceptualnog modela u relacioni model	92
2.3.1. Mapiranje jakih entitetskih tipova	92
2.3.2. Mapiranje slabih entitetskih tipova	93
2.3.3. Mapiranje specijalizacija	94
2.3.4. Mapiranje umija	98
2.3.5. Mapiranje binarnih tipova veze sa kardinalnošću mapiranja M:M	100
2.3.6. Mapiranje binarnih tipova veze sa kardinalnošću mapiranja 1:M	102
2.3.7. Mapiranje binarnih tipova veze sa kardinalnošću mapiranja 1:1	104
2.3.8. Mapiranje <i>n-arnih</i> tipova veze	107
2.3.9. Mapiranje agregacije	108
2.3.10. Mapiranje više značnih atributa	110
2.3.11. Postupak mapiranja konceptualnog modela	111
2.4. Transformacija UML konceptualnog modela	135

3. SQL – DDL	139
3.1. Kreiranje šeme i objekata šeme	139
3.1.1. Kreiranje šeme	139
3.1.2. Kreiranje tabela	140
3.1.3. Kreiranje indeksa	149
3.2. Modifikacije šeme	150
3.2.1. Brisanje šeme	150
3.2.2. Brisanje tabele	150
3.2.3. Izmjena tabele	150
3.2.4. Manipulacija indeksima	154
3.3. DDL skripta za generisanje šeme	156
4. Formalni upitni jezici	171
4.1. Relaciona algebra	171
4.1.1. Definicije i kratak opis operacija relacione algebre	171
4.1.2. Primjeri upita u relacionoj algebri	176
4.2. Relacioni račun	204
4.2.1. Definicije i kratak opis relacionog računa <i>torki</i>	204
4.2.2. Definicije i kratak opis relacionog računa domena	205
4.2.3. Primjeri upita u relacionom računu	206
5. SQL – DML	217
5.1. Selekcija i prikaz podataka	217
5.1.1. Osnovna forma SQL upita	217
5.1.2. Selekcija redova u rezultatu	221
5.1.3. Operator <i>between</i>	224
5.1.4. Operator <i>in</i>	224
5.1.5. Poređenje uzoraka sa stringovima	225
5.1.6. Preimenovanje zaglavljiva kolone	228
5.1.7. Sračunavanje vrijednosti kolone	228
5.1.8. Izraz <i>case</i>	235
5.1.9. Sortiranje redova rezultata	236
5.1.10. Rad sa <i>null</i> vrijednostima	237
5.1.11. Agregatne funkcije	239
5.1.12. Upiti sa više tabела u <i>from</i> klauzuli	243
5.1.13. Spojene tabele	246
5.1.14. Gnjiježđenje SQL upita	251
5.1.15. Klauzula <i>with</i>	257
5.1.16. Setovske operacije	258

5.2. Modifikacija baze	260
5.2.1. Brisanje podataka iz tabele	260
5.2.2. Umetanje podataka u tabelu	261
5.2.3. Ažuriranje podataka u tabeli	262
5.3. Pogledi	263
5.3.1. Modifikacija baze kroz poglede	266
5.4. Trigeri	269
5.5. Uskladištene procedure	270
5.5.1. Definicija uskladištene procedure	271
5.5.2. Kontrolne strukture	272
5.5.3. Primjeri uskladištenih procedura	274
6. Normalizacija	277
6.1. Osnovne definicije i pojmovi	277
6.1.1. Funkcionalne zavisnosti	277
6.1.2. Dekompozicija relacionih šema	278
6.1.3. Normalne forme	278
6.2. Primjeri primjene teorije normalizacije	279
7. Rješavanje integralnih zadataka	299
Prilog	325
Literatura	333

Glava 1

Konceptualno modelovanje

1.1. Uvod

Projektovanje baze podataka jeste jedan od najznačajnijih segmenta razvoja sistema sa bazama podataka. Cilj je da se dođe do detaljne specifikacije sveukupne strukture baze podataka (modela) za odgovarajući DBMS (sistem za upravljanje bazom podataka), odnosno do **logičke šeme baze podataka** na implementacionom nivou. Zbog kompleksnosti problema potreban je sistematizovan pristup koji podrazumijeva korištenje odgovarajuće metodologije.

Početna faza projektovanja baze podataka je **konceptualno modelovanje**, koje omogućava specifikaciju sveukupne strukture baze podataka na visokom nivou apstrakcije (konceptualna šema), za šta se najčešće koristi **model objekti-veze¹** (**MOV**). U praksi se često koriste i alternativne, konceptualno slične notacije, kao što je npr. **IE** (*Information Engineering*) notacija. Sa intenzivnim razvojem UML-a posljednjih godina, sve širu primjenu u konceptualnom modelovanju baze podataka ima i **UML dijagram klasa** (*class diagram*). U ovoj glavi ilustrovana je primjena sve tri prethodno pomenute notacije za konceptualno modelovanje.

1.2. Model Objekti-Veze

Koncept modela objekti-veze je jednostavan i intuitivan, zasniva se na percepciji realnog svijeta, identifikaciji klase sličnih entiteta i specifikaciji veza između njih putem odgovarajućih grafičkih koncepta. Model objekti-veze predstavlja **semantički model** podataka, jer omogućava ne samo reprezentaciju strukture podataka i veza između njih, nego i specifikaciju njihovog značenja kao i različitih ograničenja. Osnovni MOV koncepti su entiteti i veze.

Entitet (objekat) je nešto što postoji u realnom svijetu i što se razlikuje od drugih postojećih objekata. Svaki entitet realnog svijeta karakterišu određena svojstva koja se u MOV modelu reprezentuju **atributima**. Određeno svojstvo

¹ U literaturi se uobičajeno koristi engleski naziv *Entity-Relationship* (E-R) model.

entiteta se iskazuje odgovarajućom kvantitativnom vrijednošću odnosnog atributa. Skup dopuštenih vrijednosti nekog atributa, predstavlja njegov **domen**. Atributi mogu biti: *prosti ili složeni, jednoznačni ili više značni, bazni ili izvedeni*.

Određene entitete realnog svijeta karakterišu ista svojstva, odnosno ti entiteti se mogu reprezentovati istim skupom atributa. Za takve entitete kažemo da su slični, odnosno da su istog tipa. **Tip entiteta** je određen imenom tipa i skupom atributa koji reprezentuju svojstva entiteta tog tipa. Tip entiteta opisuje strukturu i karakteristike entiteta koji pripadaju istom entitetskom skupu. **Entitetski skup** je skup entiteta istog tipa u bazi podataka, u nekom trenutku.

Veza je asocijacija između određenog broja entiteta. **Tip veze** definiše karakteristike i mogući skup veza između entiteta odnosnih entitetskih tipova. **Vezni skup** je skup veza istog tipa. **Uloge** entiteta u vezi su najčešće implicitno određene. U rekurzivnim vezama između entiteta istog entitetskog skupa potrebno je eksplicitno naznačiti ulogu pojedinih entiteta u vezi, putem odgovarajućeg naziva uloge. Veze mogu imati svoje vlastite, **opisne** attribute.

Stepen tipa veze (i odgovarajućeg veznog skupa) je određen brojem entitetskih tipova (i analogno entitetskih skupova) koji participiraju u tipu veze (veznom skupu). Najčešći je **binarni tip veze**, koji povezuje dva entitetska tipa (entitetska skupa). Tip veze koji povezuje više od dva entitetska tipa (skupa) naziva se **n-arni tip veze**.

Tip veze ne samo da definiše koji tipovi entiteta učestvuju u vezama, nego specifikuje i dodatna ograničenja na učešće entiteta u vezama. **Kardinalnost mapiranja** određenog tipa veze, specifikuje maksimalan broj veza u kojima neki entitet odnosnog entitetskog tipa (skupa) može da učestvuje. Kardinalnost mapiranja za binarni tip veze može biti *jedan:jedan* (1:1), *jedan:više* (1:M), *više:jedan* (M:1) i *više:više* (M:M). **Ograničenje učešća entiteta u vezi** specifikuje, za određeni tip veze, minimalni broj veza u kojima neki entitet odnosnog entitetskog tipa (skupa) mora da učestvuje. Učešće entiteta u vezi može biti *totalno* i *parcijalno*. Učešće nekog entitetskog tipa (skupa) u veznom tipu (skupu) je totalno ako svaki entitet tog entitetskog tipa (skupa) mora učestvovati barem u jednoj vezi iz veznog skupa, a parcijalno ako samo neki entiteti tog entitetskog tipa (skupa) učestvuju u vezama iz veznog skupa.

Značajna forma ograničenja u MOV modelu su ključevi. **Superključ** je skup atributa entitetskog tipa, na kojem svaki entitet entitetskog skupa (tog tipa) ima jedinstvenu vrijednost. **Kandidatski ključ** je minimalni superključ. Neki entitetski tip može sadržavati više kandidatskih ključeva. Za entitetski tip (skup) jedan od njegovih kandidatskih ključeva selektuje se kao **primarni ključ**, odnosno kao osnovni ključ za jedinstvenu identifikaciju entiteta u tom entitetskom skupu. Slično, superključ tipa veze je skup atributa koji jedinstveno identificira svaku vezu u veznom skupu odnosnog tipa. Kandidatski/primarni ključ tipa veze je minimalni superključ tipa veze.

Neki entitetski tipovi nemaju attribute koji čine primarni ključ. Takvi entitetski tipovi se nazivaju **slabi entitetski tipovi**, a entitetski skupovi tog tipa **slabi**

entitetski skupovi. Entitetski tip koji ima attribute koji čine primarni ključ naziva se **jaki entitetski tip** (i analogno tome, odnosni entitetski skupovi su **jaki entitetski skupovi**). Tip veze koji povezuje slabi i jaki entitetski tip naziva se **identificujući tip veze**.

Osnovni MOV model nije dovoljan za pogodno predstavljanje specifičnih svojstava entiteta i odnosa između entiteta, pa je MOV model proširen dodatnim konceptima (generalizacija/specijalizacija, unija, agregacija) i u literaturi se često naziva **EER** (*Enhanced Entity-Relationship*) model.

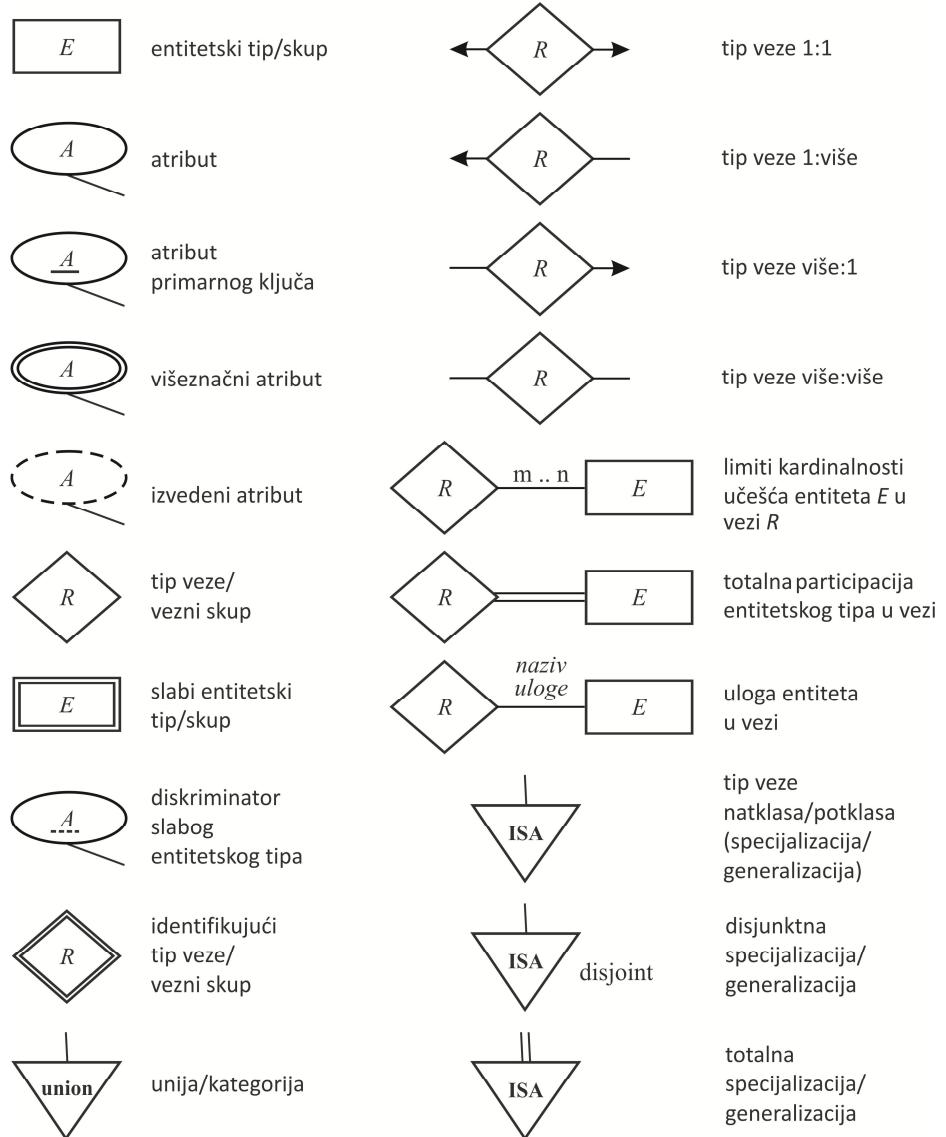
U nekim entitetskim skupovima mogu se uočiti podgrupe entiteta koji se razlikuju po nekim specifičnim svojstvima u odnosu na druge entitete iz entitetskog skupa. Proces identifikacije i reprezentacije ovih podgrupa u okviru nekog entitetskog skupa naziva se **specijalizacija**. Podgrupe entiteta sa specifičnim svojstvima unutar neke grupe modeluju se **entitetskim tipovima nižeg nivoa** koji su sa **entitetskim tipom višeg nivoa** povezani ISA vezom. ISA veza reprezentuje činjenice da su (specijalizovani) entiteti entitetskog tipa nižeg nivoa ujedno i entiteti višeg entitetskog tipa. Za entitetske tipove višeg nivoa koristi se i termin **superklasa** ili **natklasa**, a za entitetske tipove nižeg nivoa termin **potklasa**. Entitetski tipovi nižeg nivoa (potklase) nasljeđuju sve attribute entitetskog tipa višeg nivoa (superklase) sa kojim su povezani ISA vezom, i imaju, dodatno, svoje specifične attribute. Analogno nasljeđivanju atributa, nasljeđuju se i tipovi veza u kojima entitetski tip višeg nivoa (superklasa) učestvuje. Do identifikacije generalnijih entitetskih tipova višeg nivoa i entitetskih tipova nižeg nivoa može se doći i **generalizacijom**, obrnutim procesom u odnosu na specijalizaciju. Ukoliko se u toku analize sistema prvo identikuju neki entitetski tipovi (skupovi), a zatim utvrdi njihova sličnost, onda se može izvršiti generalizacija i definisanje entitetskog tipa (skupa) višeg nivoa (superklase), čiji bi atributi bili zajednički atributi polaznih entitetskih tipova.

Za određenu specijalizaciju/generalizaciju mogu se specifikovati sljedeća ograničenja: uslov članstva entiteta entitetskog skupa višeg nivoa u entitetskom skupu nižeg nivoa (*definisano uslovom* ili *korisnički definisano*); ograničenje na participaciju entiteta višeg entitetskog skupa u entitetskim skupovima nižeg nivoa (*disjunktno i preklapajuće*); ograničenje na kompletност učešća entiteta entitetskog tipa višeg nivoa u entitetskim skupovima nižeg nivoa (*totalno i parcijalno*).

Modelovanje unija i agregacija uglavnom nije uključeno i podržano u alatima za konceptualno modelovanje. Koncept **unije** ili **kategorije** omogućava modelovanje veza superklasa/potklasa, pri čemu veza povezuje više superklasa različitog tipa sa jednom potklasom. Potklasa (kategorija/unija) može da sadrži entitete različitih tipova, koji su elementi superklasa sa kojima je kategorija/unija povezana, odnosno kategorija (unija) je podskup unije entiteta različitih tipova, članova superklasa. **Agregacija** je apstrakcija kojom se veze tretiraju kao entiteti višeg nivoa, i koje, kao takvi apstraktни entiteti višeg nivoa, mogu učestvovati u drugim vezama.

1.2.1. MOV dijagram

U MOV modelu, konceptualna šema reprezentuje se grafičkom notacijom poznatom pod nazivom **MOV dijagram**². MOV notacija (niti osnovna niti proširena) nije standardizovana. Na sl. 1.1. dat je sumarni pregled MOV notacije koja se koristi u ovoj knjizi.



Slika 1.1. Sumarni pregled MOV notacije

² U literaturi se često koristi termin E-R (*Entity-Relationship*) dijagram.

1.2.2. Proces projektovanja konceptualnog modela

Ne postoji metodologija koja propisuje precizne korake koji bi uvijek vodili ka jedinstvenom i jednoznačnom konceptualnom modelu nekog realnog sistema. U procesu projektovanja projektanti se susreću sa različitim mogućnostima i dilemama. Specifične percepcije realnog sistema mogu rezultovati različitim zaključcima i različitim konceptualnim modelima za isti realni sistem. Proces izgradnje konceptualnog modela baze podataka je gotovo uvijek iterativan, pri čemu se prvo kreira početni model, a zatim se u iterativnom procesu u svakoj iteraciji vrši analiza, otklanjaju nedostaci, rješavaju projektne dileme i poboljšava model.

Najopštiji pristup jeste da se inicijalna identifikacija entitetskih skupova (tipova) i veza vrši na osnovu specifikacije zahtjeva, odnosno informacionih potreba sistema za koji se projektuje baza podataka, pri čemu se (posebno u alatima za automatsko projektovanje) koriste heuristička, odnosno neformalna pravila za identifikaciju karakterističnih koncepata. Tako su opšte imenice, koje reprezentuju objekte, pojave ili događaje realnog sistema, kandidati za entitetske tipove, a glagoli kandidati za vezne tipove. To ne znači da sve opšte imenice treba reprezentovati entitetskim tipovima, niti sve glagole veznim tipovima. Imenice mogu da imaju različite uloge u rečenicama (subjekti, objekti, apozicije itd.). *Subjekti* u rečenicama najčešće reprezentuju entitete koji treba da se modeluju odgovarajućim entitetskim tipom. *Predikati* (glagoli i glagolske fraze) koji u rečenicama povezuju subjekte i objekte (već reprezentovane entitetskim tipovima) reprezentuju se veznim tipovima.

U primjerima koji slijede, konceptualni model baze podataka projektovan je inkrementalnim pristupom na osnovu zadate specifikacije informacionih potreba, što podrazumijeva analizu zadate tekstualne specifikacije rečenicu po rečenicu i identifikaciju entiteta i njihovih veza sa prethodno identifikovanim entitetima. Ovakav pristup inkrementalno (korak po korak) vodi ka ciljnom konceptualnom modelu i pogodan je za ilustraciju primjene različitih (alternativnih) koncepata.

Primjer 1.1. Projektovati konceptualni model baze podataka univerziteta primjenom MOV notacije.

Rješenje.

Prepostavimo da smo u procesu analize identifikovali informacione potrebe hipotetičkog univerzitetskog sistema. S obzirom na njihov obim, razmatraćemo tekstualnu specifikaciju i projektovati konceptualni model po dijelovima koji predstavljaju logičke cjeline.

Specifikacija informacionih potreba (1. dio)

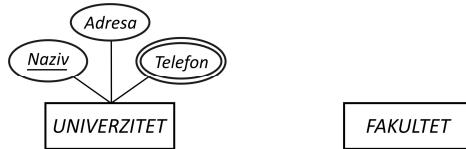
Univerzitet je obrazovna institucija koja ima naziv, adresu (sjedište) i određeni broj telefonskih brojeva za kontakt, u čiji sastav ulazi određeni broj fakulteta. Svaki fakultet ima naziv, adresu i određeni broj telefonskih brojeva za kontakt. Na fakultetima se izvodi nastava na jednom ili više studijskih programa. Studijski program karakteriše naziv studijskog programa, ciklus studija (I ciklus – dodiplomski, II ciklus – master/postdiplomski, III ciklus – doktorski studij), trajanje (broj godina/seimestara), ukupan broj ECTS kredita koji se stiče studiranjem na studijskom programu, kao i naziv zvanja koji se stiče završetkom studijskog programa. Nastavnim planom definišu se predmeti koje sadrži studijski program, semestar u kojem se po nastavnom planu izvodi nastava na predmetu i tip predmeta (obavezni/izborni). Predmete karakteriše jedinstven identifikator/kod, naziv predmeta i broj ECTS bodova. Svi predmeti su jednosemestralni. Jedan predmet može da se izvodi na više studijskih programa, jednog ili više fakulteta. Za svaki predmet je matičan određeni fakultet.

Projektovanje konceptualnog modela

Analizom prve rečenice date tekstualne specifikacije zaključujemo da je *univerzitet* entitet od interesa, tj. entitet čija su svojstva bitna u konkretnom sistemu. Dakle, prvi entitetski tip u ciljnog MOV dijagramu jeste *UNIVERZITET*. Univerzitet karakterišu *naziv, adresa, brojevi telefona*, kao i *fakulteti koji ga sačinjavaju*. Prva karakteristika predstavlja jednostavno svojstvo univerziteta, koje se modeluje prostim (atomičnim) atributom *Naziv*. Druga karakteristika predstavlja složeno svojstvo univerziteta, jer adresu čine *naziv ulice, broj, naziv mjesta i poštanski broj*. U inicijalnoj fazi razvoja konceptualnog modela i ovo svojstvo može da se reprezentuje prostim atributom *Adresa*, koji u kasnijoj fazi može da se modeluje složenim atributom. Treća karakteristika ukazuje na to da univerzitet ima veći broj telefonskih brojeva. Ovo svojstvo može da se modeluje višeznačnim atributom *Telefon*. Četvrta karakteristika univerziteta (*fakulteti koji ga sačinjavaju*) mogli bismo (slično telefonskim brojevima) da modelujemo višeznačnim atributom. Međutim, fakulteti nisu samo svojstva univerziteta, kao što bi se moglo zaključiti iz prve rečenice date tekstualne specifikacije, nego se oni višestruko pominju u narednim rečenicama kao entiteti koji imaju svoja svojstva. Dakle, fakulteti treba da se modeluju zasebnim entitetskim tipom *FAKULTET*.

Tipično u sistemu, za koji se kreira konceptualni model, postoji veći broj entiteta svakog entitetskog tipa. Identifikovani entitetski tip *UNIVERZITET* je specifičan u pogledu broja entiteta tog tipa. U konkretnom slučaju imamo samo jedan univerzitet. I bez obzira na to, treba da selektujemo primarni ključ datog entitetskog tipa. Atribut *Naziv* jedinstveno identificiše univerzitet, pa možemo da ga izaberemo za primarni ključ. Preostali atributi nisu pogodni za primarni ključ, jer je atribut *Telefon* višeznačan, a *Adresa* nema semantiku identifikatora i može više puta da se mijenja (promjena lokacije sjedišta univerziteta, promjena naziva ulice i sl.).

U ovoj fazi projektovanja, MOV dijagram je kao na sl. 1.2.



Slika 1.2. Inicijalni MOV dijagram

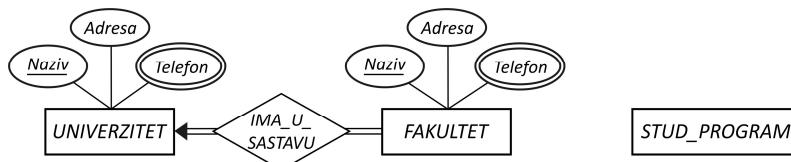
Svojstva svakog fakulteta su, slično univerzitetu, njegov *naziv*, *adresa* i *brojevi telefona*, pa po analogiji dolazimo do zaključka da entitetski tip *FAKULTET* ima jednoznačne atribute *Naziv* i *Adresa*, te višeznačni atribut *Telefon*. Karakteristika fakulteta jesu i studijski programi na kojima se izvodi nastava. Međutim, modelovanje studijskih programa korespondentnim višeznačnim atributom nije pogodno, jer se u nastavku tekstualne specifikacije studijski programi višestruko pominju kao entiteti koji imaju svojstva i veze sa drugim entitetima. Dakle studijski programi treba da se modeluju entitetskim tipom *STUD_PROGRAM*.

Svi fakulteti koji ulaze u sastav univerziteta imaju različita imena, tj. ne postoje dva istoimena fakulteta u okviru univerziteta. Dakle, vrijednost atributa *Naziv* je jedinstvena za svaki fakultet, pa se atribut *Naziv* može izabrati za primarni ključ entitetskog tipa *FAKULTET*.

Identifikujmo još veze između entiteta tipa *UNIVERZITET* i *FAKULTET*. S obzirom na to da univerzitet u svom sastavu ima određen broj fakulteta, veze između entiteta ova dva tipa možemo da modelujemo veznim tipom *IMA_U_SASTAVU*. Opciono bismo mogli, s obzirom na pripadnost fakulteta univerzitetu, ovaj vezni tip da nazovemo *PRIPADNA*.

Univerzitet u svom sastavu ima *više* fakulteta, a svaki fakultet pripada samo *jednom* univerzitetu. Dakle, kardinalnost mapiranja je 1:M (posmatrano sa strane univerziteta). *Svaki* fakultet mora da pripada univerzitetu, što znači da *svi* fakulteti učestvuju u vezama tipa *IMA_U_SASTAVU*. Dakle, učešće entiteta tipa *FAKULTET* u vezama tipa *IMA_U_SASTAVU* je *totalno*. S druge strane, univerzitet mora da ima fakultete u svom sastavu, što znači da je i učešće entiteta tipa *UNIVERZITET* u ovom tipu veze *totalno*.

U ovoj fazi projektovanja, MOV dijagram je kao na sl. 1.3.



Slika 1.3. MOV dijagram nakon identifikacije entitetskih tipova *UNIVERZITET* i *FAKULTET*

Opciono bismo telefonske brojeve (univerziteta i fakulteta) mogli modelovati zasebnim entitetskim tipovima, a ne višeznačnim atributima. Ovo bi posebno bilo bitno ako bi, osim broja telefona za kontakt, bili značajni i drugi podaci kao što je mjesto kontakta (npr. centrala, rektor/dekan, prorektori/prodekan, računovodstvo itd.), model telefona, serijski i/ili inventarski broj itd. S obzirom na to da se u datoj specifikaciji ne navode dodatna svojstva, brojeve telefona smo modelovali višeznačnim atributima.

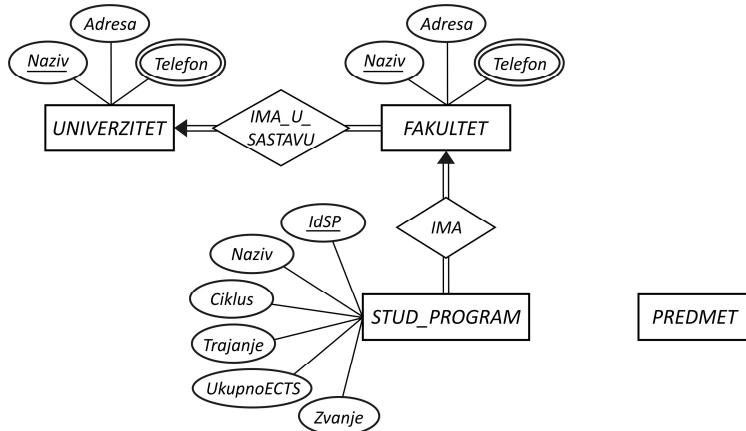
Svaki studijski program ima slijedeća svojstva: *naziv, ciklus studija, trajanje, ukupan broj ECTS kredita te naziv zvanja*, koja modelujemo odgovarajućim prostim atributima: *Naziv, Ciklus, Trajanje, UkupnoECTS* i *Zvanje*, respektivno. Dodatna karakteristika svakog studijskog programa jesu i predmeti na kojima se izvodi nastava u okviru studijskog programa. Međutim, iz nastavka tekstualne specifikacije slijedi da su predmeti takođe entiteti koji imaju svoja svojstva. Dakle, predmeti treba da se modeluju zasebnim entitetskim tipom *PREDMET*.

Opciono bismo ciklus studija takođe mogli da modelujemo zasebnim entitetskim tipom, a ne atributom. Međutim, prema datoj specifikaciji, ovo svojstvo studijskog programa reprezentuje samo činjenicu o nivou univerzitskog obrazovanja, pri čemu se o ciklusima ne vode nikakve dodatne informacije. Zato u konkretnom slučaju nema potrebe za modelovanjem ciklusa studija zasebnim entitetskim tipom.

Odredimo primarni ključ entitetskog tipa *STUD_PROGRAM*. Nijedan od atributa ne može samostalno da predstavlja primarni ključ, jer se ne može očekivati da će neki od identifikovanih atributa garantovano imati vrijednosti koje omogućavaju jedinstvenu identifikaciju svakog studijskog programa ponaosob. Može se očekivati da neće postojati dva istoimena studijska programa na različitim fakultetima, ali se može očekivati da fakultet ima više istoimenih studijskih programa na različitim ciklusima. Drugim riječima, može se očekivati da će kombinacija naziva studijskog programa i ciklusa studija biti jedinstvena na univerzitetu, pa kombinaciju atributa *Naziv* i *Ciklus* možemo da selektujemo kao primarni ključ. Međutim, često se u praksi umjesto kombinacije atributa, kao primarni ključ koristi tzv. *surogat ključ*, tj. dodatni atribut koji se uvodi u cilju jedinstvene identifikacije entiteta datog entitetskog tipa. U konkretnom primjeru možemo da uvedemo dodatni atribut *IdSP*, koji predstavlja šifru (identifikacioni kod) studijskog programa.

Između entiteta tipa *FAKULTET* i *STUD_PROGRAM* postoje veze tipa *IMA* u smislu *izvodi se nastava*. Iz specifikacije se vidi da *svaki* fakultet ima *jedan ili više* studijskih programa, a da *svaki* studijski program pripada samo *jednom* fakultetu. Dakle, kardinalnost mapiranja je 1:M posmatrano sa strane fakulteta, a učešće entiteta oba entitetska tipa *totalno* (*svaki* fakultet ima studijske programe i *svaki* studijski program pripada nekom fakultetu).

U ovoj fazi projektovanja, nakon potpune identifikacije entitetskog tipa *STUD_PROGRAM* i veznog tipa *IMA*, MOV dijagram je kao na sl. 1.4.



Slika 1.4. MOV dijagram nakon identifikacije entitetskog tipa *STUD_PROGRAM*

Svaki predmet ima sljedeća svojstva: *jedinstveni identifikator/kod*, *naziv* i *broj ECTS bodova*, koja modelujemo odgovarajućim atributima: *IdPredmeta*, *Naziv* i *ECTS*, respektivno. Činjenica da su svi predmeti jednosemestralni ne treba da se modeluje dodatnim atributom, jer bi svi entiteti tipa *PREDMET* na tom atributu imali istu vrijednost. Pošto atribut *IdPredmeta* predstavlja jedinstveni identifikator predmeta, selektujemo ga kao primarni ključ.

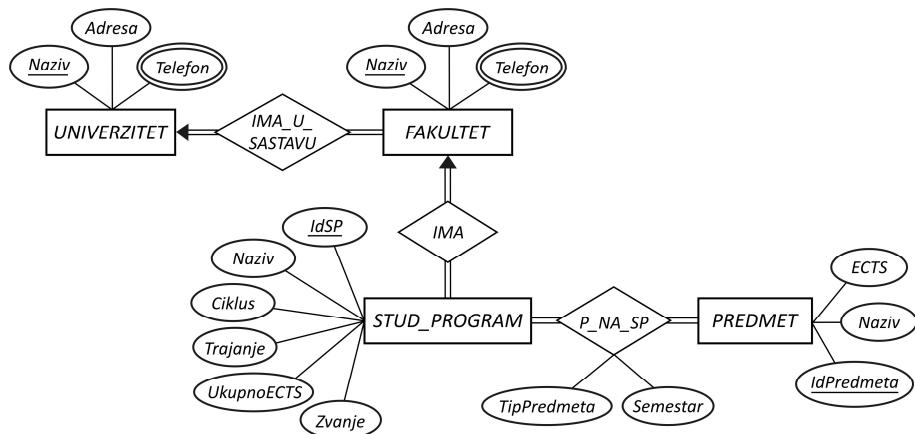
Prema zadatoj specifikaciji, *studijski program* ima *nastavni plan*. Ta činjenica može da se modeluje zasebnim entitetskim tipom *NASTAVNI_PLAN* koji je vezama tipa *IMA* povezan sa entitetskim tipom *STUD_PROGRAM*. Entitetski tip *NASTAVNI_PLAN* bio bi slabi entitetski tip (jer je nastavni plan egzistencijalno zavisao od studijskog programa), a vezni tip *IMA* bio bi identificujući sa kardinalnošću mapiranja 1:1 i totalnom participacijom entiteta oba tipa (svaki studijski program mora da ima nastavni plan, a svaki nastavni plan pripada nekom studijskom programu). S obzirom na to da se u specifikaciji ne ukazuje na mogućnost postojanja više različitih nastavnih planova (koji važe za različite generacije studenata) za isti studijski program, nema potrebe da se uvodi zaseban entitetski tip za modelovanje nastavnih planova, nego se i studijski program i njegov nastavni plan mogu smatrati jedinstvenim entitetom sa jedinstvenim atributima. U konkretnom slučaju, nastavni plan nema svoja dodatna svojstva (npr. datum usvajanja, broj odluke itd.), pa entitetskom tipu *STUD_PROGRAM* ne treba dodavati nove atribute.

Prema specifikaciji, *nastavni plan* definiše *predmete* koji pripadaju studijskom programu, *semestar* u kojem se izvodi nastava na svakom predmetu i *tip predmeta* (obavezni/izborni). Dodatno je još specifikovano da neki predmet može da se izvodi na više studijskih programa različitih fakulteta. Dakle, entiteti tipa *STUD_PROGRAM* (koji ujedno predstavlja i studijski program i njegov nastavni plan) i *PREDMET* vezani su vezama tipa *P_NA_SP* (predmet na studijskom programu). *Svaki* studijski program (u svom nastavnom planu) sadrži

više predmeta, a *svaki* predmet je sadržan u nastavnom planu *jednog ili više* studijskih programa. Dakle, kardinalnost mapiranja je M:M, a učešće entiteta oba entitetska tipa *totalno* (niti može da postoji studijski program koji u svom nastavnom planu nema predmete, niti mogu da postoje predmeti koji ne pripadaju nekom studijskom programu).

Prema specifikaciji, neki predmet može da se izvodi u okviru različitih studijskih programa, u različitim semestrima i pod različitim uslovima – negdje kao obavezni, a negdje kao izborni. Dakle, *semestar* i *tip predmeta* svojstva su veze (koja je definisana nastavnim planom) konkretnog predmeta i konkretnog studijskog programa, pa se modeluju odgovarajućim atributima *Semestar* i *TipPredmeta* veznog tipa *P_NA_SP*. Treba uočiti da vezni tip *P_NA_SP* opisuje nastavni plan odnosnog studijskog programa, tj. raspored i status predmeta po semestrima.

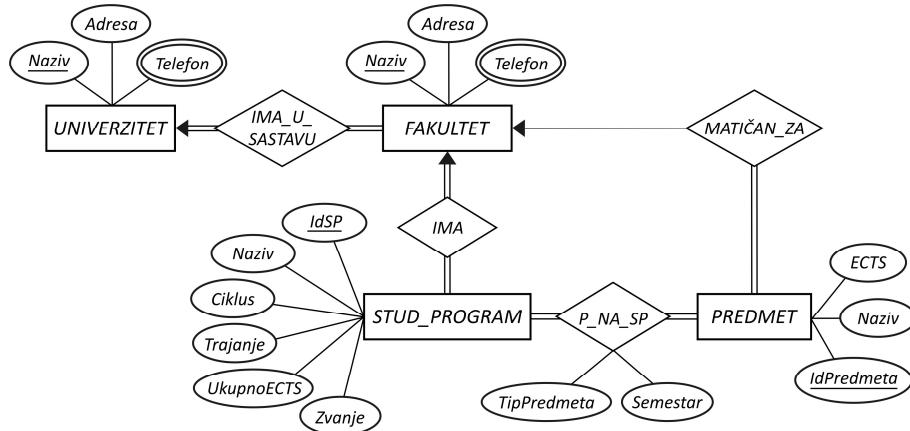
U ovoj fazi projektovanja, nakon identifikacije entitetskog tipa *PREDMET* i veznog tipa *P_NA_SP*, MOV dijagram je kao na sl. 1.5.



Slika 1.5. MOV dijagram nakon identifikacije entitetskog tipa *PREDMET*

Još treba da se identifikuju veze fakulteta i predmeta po osnovu matičnosti. Prema zadatoj specifikaciji, za *svaki* predmet matičan je određeni (*jedan*) fakultet. S druge strane, fakultet može biti matičan za *više* predmeta. Dakle, kardinalnost mapiranja veza tipa *MATIČAN_ZA* je 1:M posmatrano sa strane fakulteta. Ako pretpostavimo da mogu da postoje i fakulteti za multidisciplinare studije, koji nisu matični ni za jedan predmet, zaključujemo da je učešće entiteta tipa *PREDMET totalno* (za *svaki* predmet matičan je neki fakultet), a da je učešće entiteta tipa *FAKULTET parcijalno* (mogu da postoje fakulteti koji nisu matični ni za jedan predmet).

Dodavanjem veznog tipa *MATIČAN_ZA* dobijamo MOV dijagram kao na sl. 1.6, koji predstavlja konceptualni model baze podataka kreiran na osnovu zadate tekstualne specifikacije.



Slika 1.6. MOV dijagram kreiran na osnovu tekstualne specifikacije informacionih potreba hipotetičkog univerzitetskog sistema (1. dio)

Kreirani konceptualni model rezultat je detaljne analize zadate tekstualne specifikacije, koja je sprovedena korak po korak. Do istog rezultata bi se došlo i primjenom ranije navedenih neformalnih pravila. Sve opšte imenice (koje predstavljaju subjekte u rečenicama, a reprezentuju objekte, pojave i događaje realnog sistema) bile bi reprezentovane odnosnim entitetskim tipovima. Tako bi se modelovali *univerzitet*, *fakultet*, *studijski program* i *predmet*, a *nastavni plan* bi se nakon dodatne analize (nema sopstvene atribute, zavisi od studijskog programa i postoji samo jedan nastavni plan za svaki studijski program) objedinio sa *studijskim programom*. S druge strane, glagolske fraze *ima u sastavu*, *matičan za* itd. bile bi reprezentovane odgovarajućim veznim tipovima između odnosnih entitetskih tipova.

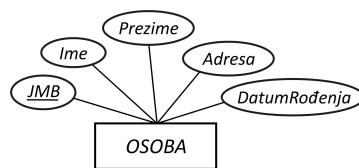
Specifikacija informacionih potreba (2. dio)

Na univerzitetu je neophodno voditi evidenciju o različitim osobama. Svaka osoba ima jedinstven identifikator (matični broj), prezime, ime, datum rođenja i adresu. Za odvijanje osnovnih procesa na univerzitetu – nastavnih i naučno-istraživačkih, ključne grupe osoba su nastavnici, asistenti/saradnici i studenti. Ostale osobe su u funkciji podrške za odvijanje osnovnih procesa na univerzitetu. Nastavu na fakultetu izvode zaduženi nastavnici i saradnici (asistenti). Na izvođenju nastave moraju biti angažovani nastavnici, dok na nekim (teorijskim) predmetima vježbe nisu obavezne. Nastavnici i asistenti su osobe koje mogu biti zaposlene na univerzitetu, ili mogu biti angažovani kao spoljni saradnici. Svaki zaposleni ima odgovarajući platu. Spoljne saradnike takođe karakteriše institucija u kojoj su stalno zaposleni. Za angažovanje na univerzitetu, spoljni saradnici zaključuju ugovor. Nastavnici i asistenti imaju nastavnička, odnosno asistentska zvanja. Nastavnici i asistenti mogu izvoditi nastavu na više predmeta i na više studijskih programa/fakulteta univerziteta. Dekan koji rukovodi fakultetom i rektor koji rukovodi univerzitetom, biraju se iz redova zaposlenih nastavnika univerziteta.

Projektovanje konceptualnog modela

Na osnovu analize prve dvije rečenice zaključujemo da se na univerzitetu vodi evidencija o različitim *osobama* te da su svojstva svake osobe: *jedinstveni identifikator* (*matični broj*), *prezime*, *ime*, *datum rođenja* i *adresa*. Dakle, *osoba* je entitet od interesa i treba da se reprezentuje odgovarajućim entitetskim tipom *OSOBA*, čiji su atributi: *JMB*, *Prezime*, *Ime*, *DatumRođenja* i *Adresa*. Primarni ključ je *JMB*, jer predstavlja jedinstveni identifikator svake osobe.

To što se na univerzitetu vodi evidencija o različitim osobama, još uvijek ne ukazuje na potrebu uvodenja nekog novog entitetskog tipa niti tipa veze između osoba i nekih drugih entiteta. Inicijalni MOV dijagram za drugi dio specifikacije informacionih potreba, nakon identifikacije entitetskog tipa *OSOBA*, dat je na sl. 1.7. Radi preglednosti, dio sa sl. 1.6. nije prikidan.



Slika 1.7. Inicijalni MOV dijagram za drugi dio tekstuale specifikacije

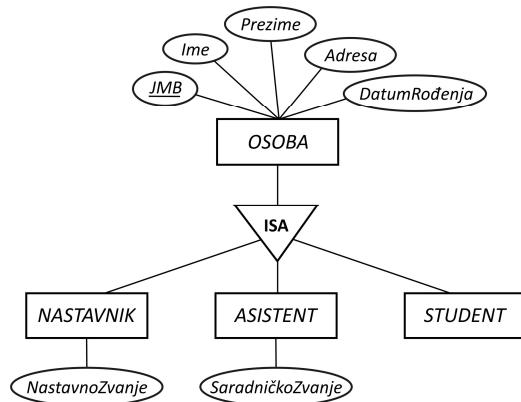
Za odvijanje osnovnih procesa na univerzitetu ključne grupe osoba su *nastavnici*, *asistenti* i *studenti*. Svaka od tih grupa ima karakteristične *uloge* (izvode ili slušaju nastavu, rukovode fakultetom itd.) i *svojstva* (plata, zvanje itd.), pa treba da se modeluje korespondentnim entitetskim tipom *NASTAVNIK*, *ASISTENT* i *STUDENT*, respektivno. Ovi entitetski tipovi reprezentuju svojevrsne podvrste (*potklase*) osoba, a entitetski tip *OSOBA* uopštenu reprezentaciju (*natklasa*) svih osoba. Veze natklase i potklasa modelujemo specijalizacijom (ISA veza).

Prema tekstualonej specifikaciji mogu da postoje i druge osobe, koje su u funkciji podrške osnovnim procesima. Dakle, postoje i osobe koje nisu ni nastavnici ni asistenti ni studenti, što znači da je specijalizacija entitetskog tipa *OSOBA*, po datom kriterijumu, *parcijalna*. S obzirom na to da su asistenti najčešće i studenti postdiplomskih ili doktorskih studija, data specijalizacija *nije disjunktna*, već *preklapajuća*.

Svojstva koja su zajednička za entitete potklasa već su identifikovana i reprezentovana atributima natklase. Zato te attribute (u konkretnom slučaju: *JMB*, *Prezime*, *Ime*, *DatumRođenja* i *Adresa*) ne treba dodavati potklasama. Treba identifikovati i dodati samo one attribute koji reprezentuju svojstva karakteristična za konkretnu potklasu.

Analizom specifikacije uočavamo da nastavnici i asistenti imaju nastavnika, odnosno asistentska (saradnička) zvanja. Pošto zvanja nisu detaljno specifikovana niti se očekuje da se za nastavnike i asistente vode evidencije o izborima (datum

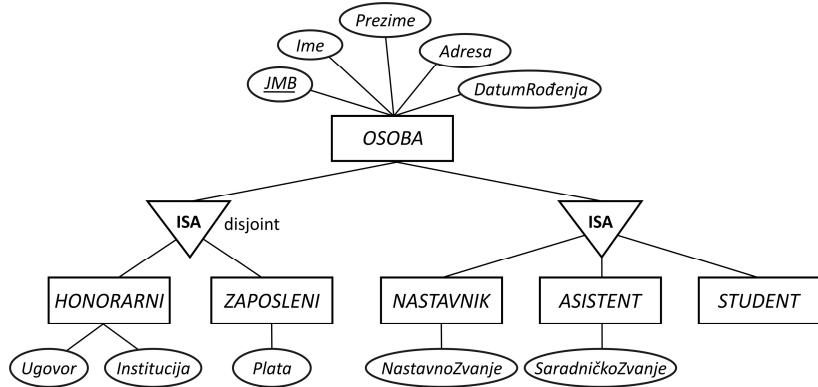
izbora, broj odluke, naziv zvanja, period itd.), zaključujemo da se zvanja mogu modelovati odgovarajućim atributima *NastavnoZvanje* i *SaradničkoZvanje* odnosnih entitetskih tipova *NASTAVNIK* i *ASISTENT*, respektivno. U suprotnom, bilo bi neophodno uvesti odgovarajuće entitetske tipove *NASTAVNO_ZVANJE* i *SARADNIČKO_ZVANJE*, a zatim identifikovati i tipove veza između entiteta tipa *NASTAVNIK* i *NASTAVNO_ZVANJE*, te *ASISTENT* i *SARADNIČKO_ZVANJE*. U ovoj fazi projektovanja, MOV dijagram je kao na sl. 1.8 (dio sa sl. 1.6. nije prikazan).



Slika 1.8. MOV dijagram nakon identifikacije potklasa *NASTAVNIK*, *ASISTENT* i *STUDENT*

Specifikacija definiše dvije vrste angažmana na univerzitetu. Osim *stalno zaposlenih* radnika, univerzitet može da angažuje i *spoljne saradnike*. Ovo su dvije specifične podvrste osoba. Zaposlene karakteriše *plata*, a spoljne saradnike *ugovor* i *institucija* u kojoj su zaposleni. Zbog toga ove dvije grupe treba da se modeluju odgovarajućim potklasama. Potklasa *ZAPOSLENI*, sa atributom *Plata*, reprezentuje stalno zaposlene, a potklasa *HONORARNI*, sa atributima *Ugovor* i *Institucija*, reprezentuje spoljne saradnike. Ne ostvaruju sve osobe na univerzitetu primanja (npr. studenti). Zato je specijalizacija po ovom osnovu *parcijalna*. Angažovani ne mogu istovremeno da budu i spoljni saradnici i stalno zaposleni na univerzitetu, pa je specijalizacija *disjunktna*. Nakon identifikacije potklasa *ZAPOSLENI* i *HONORARNI*, MOV dijagram je kao na sl. 1.9 (dio sa sl. 1.6. nije prikazan).

Prema specifikaciji, s obzirom na angažman, razlikujemo dvije vrste nastavnika, i to: stalno zaposlene nastavnike i honorarne nastavnike. To znači da entitetski tip *NASTAVNIK* treba da se specijalizuje potklasama *NASTAVNIK_ZAP* i *NASTAVNIK_HON* koje reprezentuju te dvije vrste. Pošto nastavnik ne može istovremeno da bude stalno zaposlen na univerzitetu i honorarni saradnik, ova specijalizacija je *disjunktna*, a ujedno i *totalna*, jer ne može da postoji nastavnik koji ima drugačiji radnopravni status.



Slika 1.9. MOV dijagram nakon identifikacije potklasa ZAPOSLENI i HONORARNI

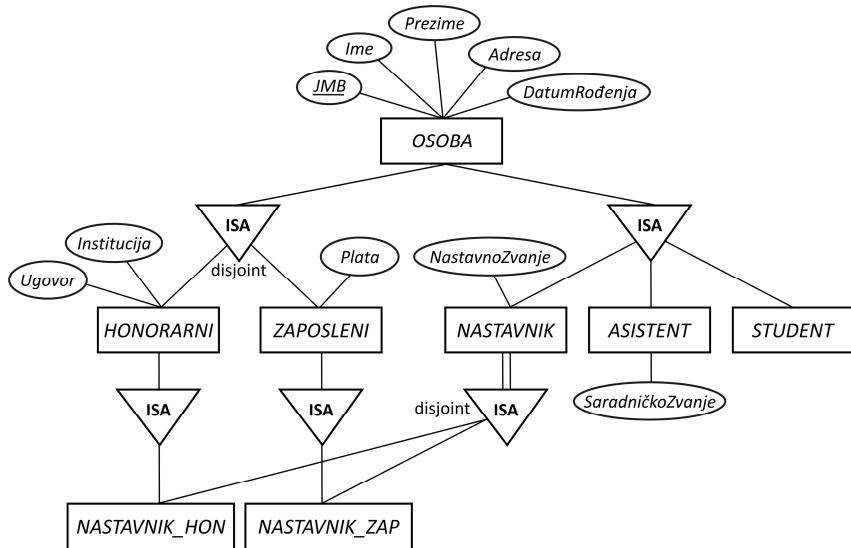
Stalno zaposleni nastavnik nije samo vrsta nastavnika, nego je ujedno i vrsta zaposlene osobe. Zbog toga potklasa NASTAVNIK_ZAP treba da specijalizuje, ne samo natklasu NASTAVNIK, nego i natklasu ZAPOSLENI, a entiteti tipa NASTAVNIK_ZAP imaju svojstva karakteristična za entitete obje natklase. Tako će zaposleni nastavnici imati i nastavničko zvanje (kao nastavnici) i platu (kao stalno zaposleni). Budući da se potklasom NASTAVNIK_ZAP modeluje samo jedna grupa zaposlenih (stalno zaposleni nastavnici), ova specijalizacija natklase ZAPOSLENI je *parcijalna*.

Slično, honorarni nastavnik ujedno je i vrsta honorarnog saradnika, pa potklasa NASTAVNIK_HON treba da specijalizuje, ne samo natklasu NASTAVNIK, nego i natklasu HONORARNI. Na taj način će honorarni nastavnici imati nastavničko zvanje (kao nastavnici) te ugovor i matičnu instituciju (kao honorarni saradnici). Ova specijalizacija natklase HONORARNI takođe je *parcijalna*, jer potklasa NASTAVNIK_HON reprezentuje samo honorarne nastavnike. U ovoj fazi projektovanja, MOV dijagram je kao na sl. 1.10 (dio sa sl. 1.6. nije prikazan).

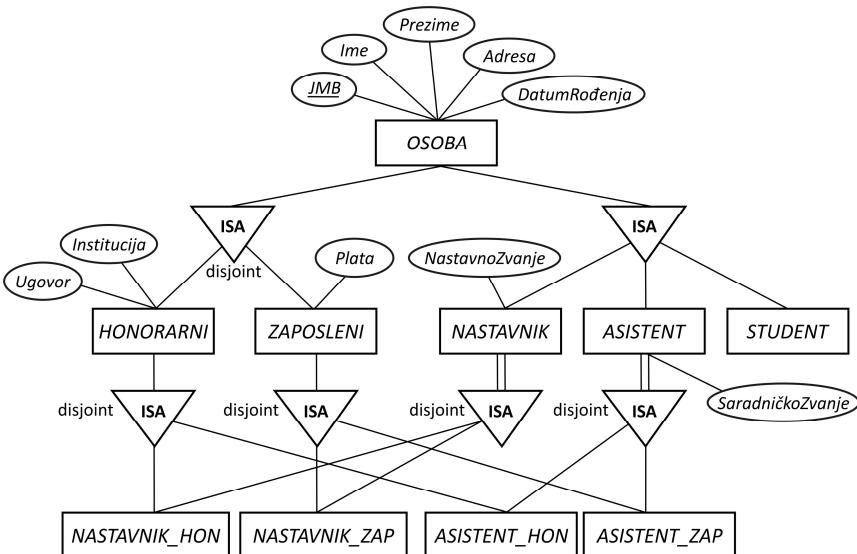
Analogno prethodnom, razlikujemo i dvije vrste asistenata: stalno zaposlene i honorarne, što možemo da modelujemo *totalnom* i *disjunktnom* specijalizacijom natklase ASISTENT potklasama ASISTENT_ZAP i ASISTENT_HON. Stalno zaposleni asistent ujedno je i vrsta zaposlene osobe, pa potklasa ASISTENT_ZAP treba da specijalizuje, ne samo natklasu ASISTENT, nego i natklasu ZAPOSLENI. Tako će zaposleni asistenti imati i asistentsko zvanje (kao asistenti) i platu (kao stalno zaposleni). Slično, honorarni asistent ujedno je i vrsta honorarnog saradnika, pa potklasa ASISTENT_HON treba da specijalizuje, ne samo natklasu ASISTENT, nego i natklasu HONORARNI.

Budući da natklasu ZAPOSLENI specijalizuju dvije potklase (odranije NASTAVNIK_ZAP i sada ASISTENT_ZAP), ova specijalizacija treba da se modeluje kao *disjunktna*, jer ne može stalno zaposlena osoba istovremeno da bude i nastavnik i asistent. Analogno, i specijalizacija natklase HONORARNI treba da se modeluje kao *disjunktna*. Obje specijalizacije (i natklase ZAPOSLENI

i natklase *HONORARNI*) su *parcijalne*, jer osim nastavnika i asistenata postoje i druge osobe koje na univerzitetu imaju angažman (npr. administrativno osoblje). U ovoj fazi projektovanja, MOV dijagram je kao na sl. 1.11 (radi preglednosti, dio sa sl. 1.6. nije prikazan).



Slika 1.10. MOV dijagram nakon identifikacije potklasa
NASTAVNIK_ZAP i *NASTAVNIK_HON*



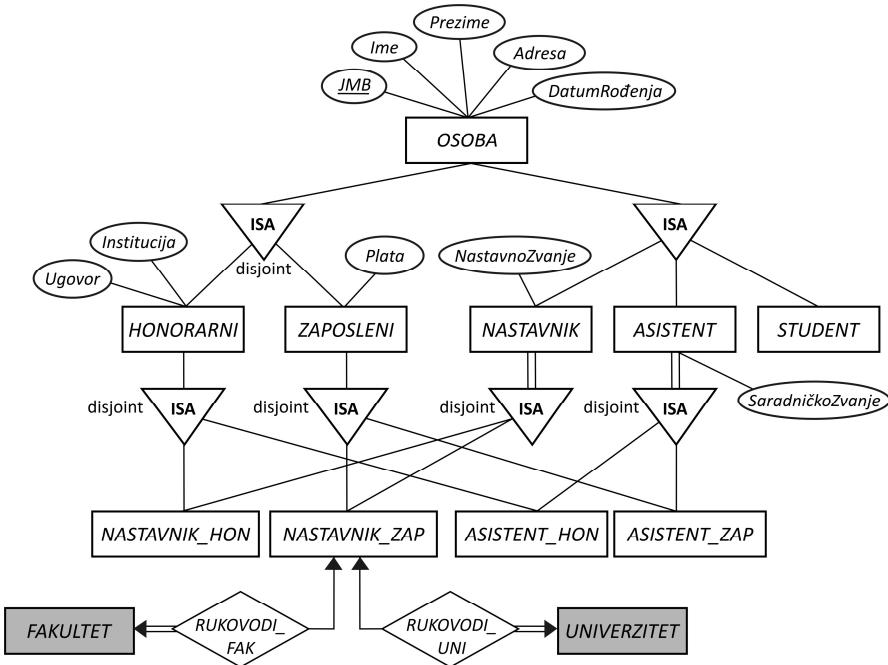
Slika 1.11. MOV dijagram nakon identifikacije potklasa
ASISTENT_ZAP i *ASISTENT_HON*

U odnosu na entitete natklasa, entiteti potklasa imaju neka specifična svojstva (koja se reprezentuju atributima u potklasama) i/ili specifične uloge (koje se reprezentuju tipovima veza u kojima participiraju entiteti potklasa). Sva specifična svojstva entiteta identifikovanih potklasa, koja su specifikovana u datom tekstuualnom opisu, reprezentovana su odgovarajućim atributima na MOV dijagramu sa sl. 1.11. Potrebno je još da se identifikuju tipovi veza između entiteta potklasa i drugih entitetskih tipova (ranije identifikovanih na osnovu prvog dijela specifikacije i prikazanih MOV dijagramom na sl. 1.6).

Prema specifikaciji, samo stalno zaposleni nastavnici mogu da rukovode fakultetom. Dakle, postoje veze tipa *RUKOVODI_FAK* između entiteta tipa *NASTAVNIK_ZAP* i *FAKULTET*. Svaka veza ovog tipa reprezentuje činjenicu da je jedan stalno zaposleni nastavnik dekan nekog fakulteta. *Svaki fakultet ima dekanu* i u posmatranom trenutku samo jedan nastavnik može da bude dekan na jednom fakultetu (tj. postoje nastavnici koji nisu dekani), pa je učešće entiteta tipa *FAKULTET* u vezi *totalno*, a entiteta tipa *NASTAVNIK_ZAP parcijalno*. Još treba da se odredi kardinalnost mapiranja. Budući da se specifikacijom ne zahtijeva vodenje evidencija o svim dekanima u istoriji fakulteta, već samo informacija o trenutnom dekanu, kardinalnost mapiranja je 1:1 (fakultet ima samo jednog dekana, a neki nastavnik može u nekom trenutku da bude dekan samo jednog fakulteta). Ako bi od značaja bili podaci o prethodnim dekanima, tada bi kardinalnost mapiranja bila M:M, jer bi neki nastavnik tokom svoje karijere mogao da bude više puta dekan na istom ili na različitim fakultetima, a svaki fakultet u svojoj istoriji (može da) ima više različitih dekana. Tada bi i odgovarajući vezni tip imao opisne atribute (npr. period rukovođenja).

Analogno prethodnom, samo stalno zaposleni nastavnici mogu da rukovode univerzitetom, što modelujemo vezama tipa *RUKOVODI_UNI* između entiteta tipa *NASTAVNIK_ZAP* i *UNIVERZITET*, pri čemu je učešće entiteta tipa *UNIVERZITET totalno*, a učešće entiteta tipa *NASTAVNIK_ZAP parcijalno*. Sličnim rezonovanjem usvajamo i da je kardinalnost mapiranja 1:1. U ovoj fazi projektovanja, MOV dijagram je kao na sl. 1.12. Radi preglednosti, dio sa sl. 1.6. nije prikazan u cijelosti, već samo tipovi entiteta (osjenčeno) s kojima su povezani entiteti tipa *NASTAVNIK_ZAP*.

Preostali dio specifikacije opisuje angažman nastavnika i asistenata u nastavnom procesu. Nastavnici mogu da izvode nastavu na *više* predmeta i na *više* studijskih programa. Iako nije eksplicitno navedeno u specifikaciji, možemo da prepostavimo da na nekim predmetima može da bude angažovano *više* nastavnika. Dakle, činjenice o angažmanu nastavnika na različitim predmetima različitih studijskih programa treba da modelujemo *ternarnim* tipom veze *PREDAJE* između odnosnih entitetskih tipova *NASTAVNIK*, *PREDMET* i *STUD_PROGRAM*, pri čemu je kardinalnost mapiranja M:M:M. Ako prepostavimo da mogu da postoje nastavnici koji (još) nemaju zaduženje na nekom predmetu, kao i studijski programi, odnosno predmeti za koje (još) nisu definisana zaduženja, participacija entiteta odnosnih tipova u vezama tipa *PREDAJE* je *parcijalna*.

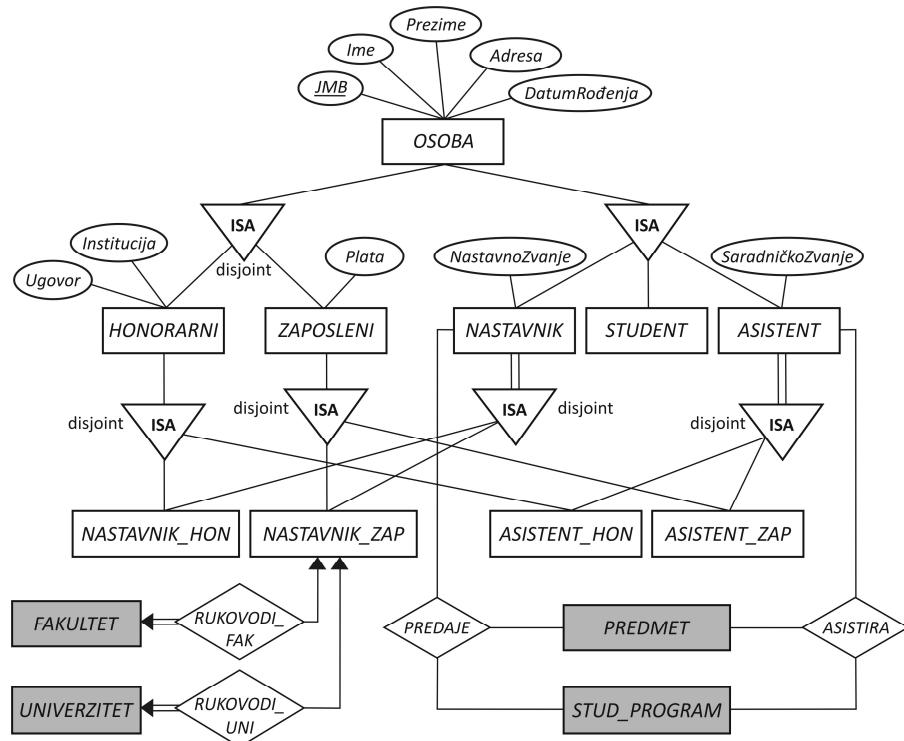


Slika 1.12. MOV dijagram nakon identifikacije veznih tipova
RUKOVODI_FAK i RUKOVODI_UNI

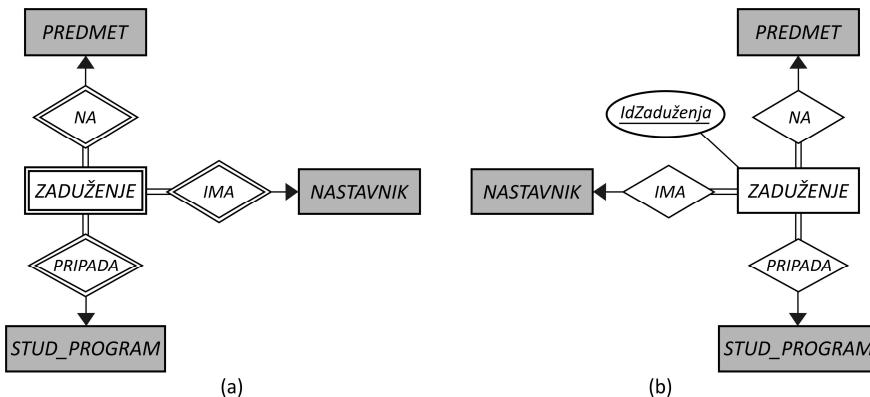
Analognim rezonovanjem dolazimo i do *ternarnog* tipa veze *ASISTIRA* između entitetskih tipova *ASISTENT*, *PREDMET* i *STUD_PROGRAM*, kojim se reprezentuju činjenice o angažmanu asistenata na različitim predmetima različitih studijskih programa. Kardinalnost mapiranja takođe je M:M:M.

Dodavanjem veznih tipova *PREDAJE* i *ASISTIRA* dobijamo MOV kao na sl. 1.13., koji predstavlja konceptualni model podataka kreiran na osnovu zadate tekstualne specifikacije. Radi preglednosti, dio MOV dijagrama sa sl. 1.6. nije prikazan u cijelosti, već samo tipovi entiteta (osjenčeno) s kojima su povezani entitetski tipovi identifikovani na osnovu drugog dijela tekstualne specifikacije.

Ternarni vezni tip *PREDAJE* (budući da predstavlja *zaduženje* nastavnika na nekom predmetu nekog studijskog programa), alternativno može da se reprezentuje odgovarajućim entitetskim tipom *ZADUŽENJE*, koji je sa entitetskim tipovima *NASTAVNIK*, *STUD_PROGRAM* i *PREDMET* vezan odgovarajućim binarnim veznim tipovima. Svako *zaduženje* je egzistencijalno zavisno i od konkretnog nastavnika i od konkretnog predmeta i od konkretnog studijskog programa, pa je entitetski tip *ZADUŽENJE* slabi entitetski tip, a odnosni binarni vezni tipovi identificirajući (sl. 1.14.a)). Dodatno se (ako bi entiteti tipa *ZADUŽENJE* bili u vezama sa entitetima drugih tipova), uvođenjem *surogat ključa* (*IdZaduženja*), slabi entitetski tip *ZADUŽENJE* može prevesti u jaki (jer bi imao primarni ključ), a odnosni binarni vezni tipovi tada bi bili neidentificirajući, kao na sl. 1.14.b).



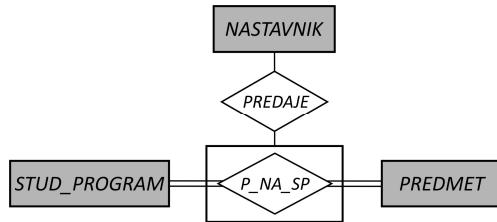
Slika 1.13. MOV dijagram kreiran na osnovu tekstualne specifikacije informacionih potreba hipotetičkog univerzitetskog sistema (2. dio)



Slika 1.14. Alternativne varijante modelovanja zaduženja nastavnika:
a) slabim entitetskim tipom, b) jakim entitetskim tipom

Koncept agregacije omogućava da se veze tretiraju kao entiteti višeg nivoa. U konkretnom slučaju, veze tipa P_NA_SP (ranije identifikovane između entitetskih tipova $STUD_PROGRAM$ i $PREDMET$) mogu da se tretiraju kao entiteti višeg nivoa. Svaki objekat tipa P_NA_SP reprezentuje činjenicu da je neki predmet

sadržan u nekom studijskom programu. Budući da nastavnici predaju predmete koji pripadaju nekom studijskom programu, vezni tip *PREDAJE* upravo treba da povezuje odnosne entitetske tipove *NASTAVNIK* i *P_NA_SP*, kao što je prikazano na sl. 1.15. Kardinalnost mapiranja veznog tipa *PREDAJE* je M:M (jer svaki nastavnik može da predaje više predmeta (na istom ili različitim studijskim programima), a isti predmet može da predaje više nastavnika), pri čemu je participacija objekata oba tipa *parcijalna* (mogu da postoje nastavnici koji nemaju zaduženje te predmeti za koje još niko nije zadužen).



Slika 1.15. Reprezentacija zaduženja nastavnika agregacijom

Ternarni vezni tip *ASISTIRA*, kojim se reprezentuju činjenice o zaduženjima asistenata, takođe može da se tretira na prethodno opisane načine.

Specifikacija informacionih potreba (3. dio)

Student je osoba koja je upisana na neki studijski program nekog fakulteta. Prilikom upisa na fakultet studenti dobijaju indeks sa jedinstvenim brojem indeksa za studijski program koji upisuju. Studenti nekog studijskog programa, upisuju odgovarajući semestar i predmete (i tokom semestra prisustvuju nastavi) i polažu ispite koji se organizuju u terminima ispitnih rokova i na određenim lokacijama. Za svaki ispit treba da postoji mogućnost registrovanja broja studenata koji su izašli na ispit i broja studenata koji su položili ispit. Nakon izlaska na ispit, student dobija odgovarajuću ocjenu.

Projektovanje konceptualnog modela

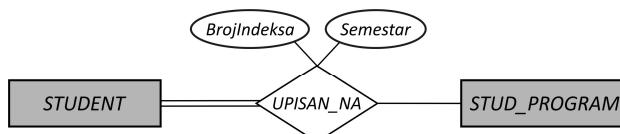
Specifikacijom se definiše da je *student* (kao vrsta *osobe*) *upisan* na neki *studijski program*. Odnosni entitetski tipovi *STUDENT* i *STUD_PROGRAM* već su identifikovani i reprezentovani u MOV dijagramu. Još treba da se identificuje odgovarajući vezni tip *UPISAN_NA*.

Iako nije eksplisitno specifikovano, treba uzeti u obzir činjenicu da ista osoba (*student*) može da upiše više studijskih programa (najčešće na različitim ciklusima na istom fakultetu). S druge strane, na nekom studijskom programu tipično studira više studenata, pa je kardinalnost mapiranja M:M. Budući da ne može da postoji student koji nije upisan ni na jedan studijski program, učešće entiteta tipa *STUDENT* je *totalno*. Učešće entiteta tipa *STUD_PROGRAM* je *parcijalno*, jer može da postoji (npr. neki tek osnovan) studijski program na koji nije upisan još nijedan student.

Za svaki upisani studijski program student dobija (novi) indeks, što znači da je indeks (tj. broj indeksa) svojstvo svakog upisa (odnosno veze između studenta i studijskog programa), pa treba da se modeluje odgovarajućim atributom *BrojIndeksa* korespondentnog veznog tipa *UPISAN_NA*.

Student upisuje semestar na upisanom studijskom programu. Upisani semestar takođe je svojstvo veze između studenta i studijskog programa. Ako prepostavimo (radi pojednostavljenja) da je dovoljna informacija o trenutnom semestru u koji je student upisan, tada ovo svojstvo možemo da reprezentujemo atributom *Semestar* veznog tipa *UPISAN_NA*. U suprotnom, upis semestra je entitet sa svojim svojstvima (npr. datum upisa/obnove itd.) i treba da se modeluje odgovarajućim entitetskim tipom.

Na sl. 1.16. prikazan je relevantni dio MOV dijagrama. Radi preglednosti, prikazani su (osjenčeno) samo entitetski tipovi koji su povezani vezama tipa *UPISAN_NA*.

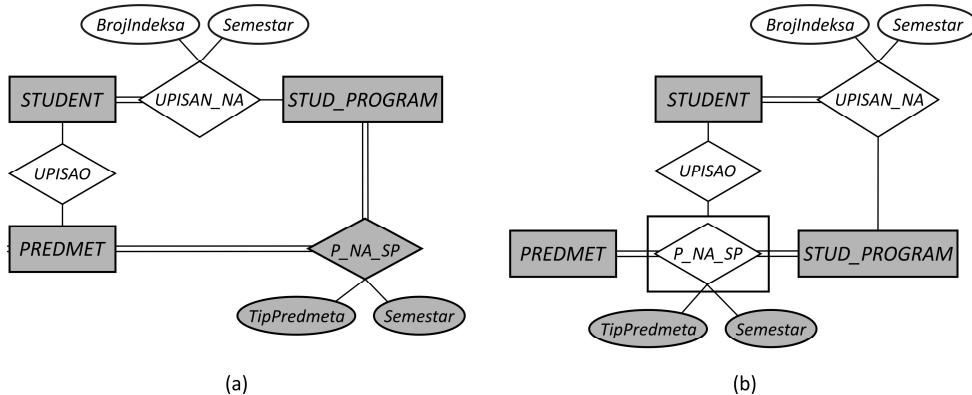


Slika 1.16. Dio MOV dijagrama nakon identifikacije veznog tipa *UPISAN_NA*

Prilikom upisa semestra, student bira i upisuje predmete koje će slušati u toku tog semestra. Najjednostavnija reprezentacija činjenice da je neki *student upisao* neki *predmet* jeste veza tipa *UPISAO* između odnosnih entitetskih tipova *STUDENT* i *PREDMET*. Tipično, student u svakom semestru upisuje *više* predmeta, a neki predmet istovremeno sluša *više* studenata, pa je kardinalnost mapiranja M:M. Ako prepostavimo da student prilikom upisa na neki studijski program, prvo upisuje studijski program, a naknadno upisuje semestar i bira predmete, možemo zaključiti da u nekom periodu mogu da postoje studenti koji (još) nisu upisali nijedan predmet, pa je participacija entiteta tipa *STUDENT* u vezama tipa *UPISAO parcijalna*. S druge strane, mogu da postoje izborni predmeti koje nije izabrao nijedan student, pa je participacija entiteta tipa *PREDMET* u vezama tipa *UPISAO* takođe *parcijalna*.

Nakon identifikacije veznog tipa *UPISAO*, relevantni dio MOV dijagrama je kao na sl. 1.17.a). Osjenčeno su prikazani koncepti koji su identifikovani u ranijim fazama projektovanja modela.

Usljed veoma pojednostavljenog tretiranja veza studenata i predmeta, kreirani model ima veoma bitan nedostatak, jer ne specifikuje ograničenje da student može da upiše samo predmet sa studijskog programa koji je upisao. Ovaj nedostatak može (djelimično) da se otkloni primjenom koncepta agregacije (slično modelovanju angažmana nastavnika), kao što je prikazano na sl. 1.17.b).



Slika 1.17. Dio MOV dijagrama nakon identifikacije veznog tipa *UPISAO* (a) i poboljšani model primjenom agregacije (b)

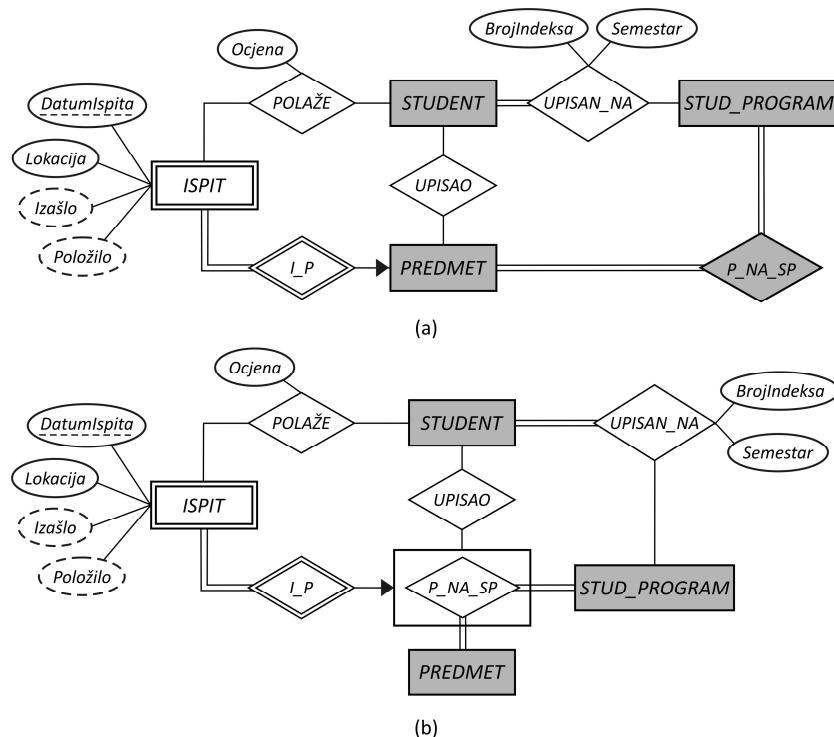
Studenti polažu *ispite* koji se organizuju u terminima ispitnih rokova i na određenim lokacijama. Dakle, *ispiti* su entiteti od posebnog značaja i treba da se reprezentuju odgovarajućim entitetskim tipom *ISPIT*, čiji atributi *DatumIspita* i *Lokacija* reprezentuju karakteristična svojstva svakog ispita. Budući da se u istom terminu i na istoj lokaciji može održati više ispita, nijedan od atributa pojedinačno niti oba atributa u kombinaciji ne mogu predstavljati primarni ključ. Pošto nema primarni ključ, entitetski tip *ISPIT* je *slabi entitetski tip*.

Ispiti su egzistencijalno zavisni od *predmeta* (ne može se organizovati ispit iz predmeta koji ne postoji), pa su veze između ovih entiteta *identifikujuće*. Za svaki predmet (može da) se organizuje više ispita, a svaki ispit se odnosi na konkretni predmet, pa je kardinalnost mapiranja 1:M posmatrano sa strane entitetskog tipa *PREDMET*, pri čemu je participacija entiteta tipa *PREDMET* *parcijalna*, a participacija entiteta tipa *ISPIT* *totalna*. Svi ispiti koji se odnose na neki predmet, razlikuju se po terminu održavanja, pa je atribut *DatumIspita* diskriminator entitetskog tipa *ISPIT*.

Student (može da) polaže *više* ispita, a svakom ispitu (može da) pristupa *više* studenata, pa je kardinalnost mapiranja veznog tipa *POLAŽE*, kojim se reprezentuju veze između odnosnih entitetskih tipova *STUDENT* i *ISPIT*, *više:više*, pri čemu su učešća entiteta oba entitetska tipa *parcijalna*. Svojstvo svakog polaganja ispita je *ocjena* (kojom je student ocijenjen) i modeluje se atributom *Ocjena* odnosnog veznog tipa *POLAŽE*.

Prema datojoj specifikaciji, za svaki ispit treba da postoji mogućnost registrovanja broja studenata koji su izašli na ispit i broja studenata koji su položili ispit. Ova dva svojstva ispita reprezentujemo atributima *Izašlo* i *Položilo* odnosnog entitetskog tipa *ISPIT*, respektivno. S obzirom na to da se podatak o broju studenata koji su izašli na ispit može izračunati na osnovu broja veza studenata sa ispitom, te da se podatak o broju studenata koji su položili ispit može izračunati analizom ocjena, odnosni atributi *Izašlo* i *Položilo* treba da se modeluju kao *izvedeni* atributi.

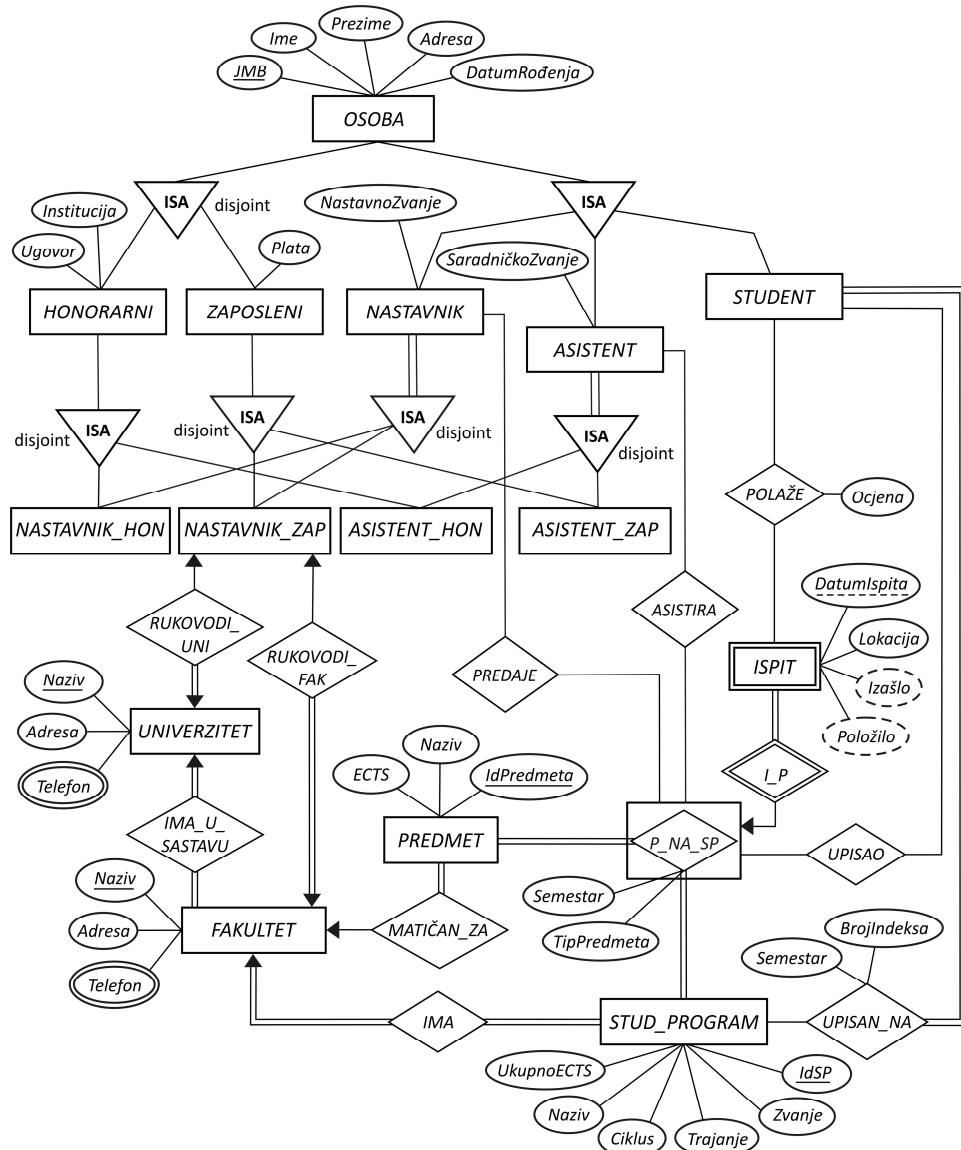
U ovoj fazi projektovanja, relevantan dio MOV dijagrama je kao na sl. 1.18.a). Osjenčeno su prikazani ranije identifikovani koncepti. Kreirani model ima nedostatak, jer omogućava da student polaže ispit iz bilo kojeg predmeta na univerzitetu, a ne samo iz predmeta koji pripadaju odgovarajućem studijskom programu ili samo iz predmeta koje je student upisao na odgovarajućem studijskom programu. Ovaj nedostatak djelimično se eliminiše korišćenjem agregacije (sl. 1.18.b)). Iako poboljšan, model sa sl. 1.18.b) i dalje omogućava da student polaže ispite iz predmeta koje nije upisao i iz predmeta sa drugih studijskih programa. Međutim, poboljšani model omogućava lakšu manipulaciju relevantnim podacima, jer se lakše dolazi do određenog predmeta na konkretnom studijskom programu.



Slika 1.18. Dio MOV dijagrama nakon identifikacije entitetskog tipa *ISPIT* (a) i poboljšani model primjenom agregacije (b)

Bez obzira na opisane nedostatke, može se smatrati da kreirani model dovoljno dobro reprezentuje specifikovane informacione potrebe. Kreirani model omogućava određenu fleksibilnost koja je često neophodna u toku eksploracije baze podataka. U procesu projektovanja treba biti svjestan činjenice da ne mogu sva ograničenja da se uvaže i reprezentuju u modelu, odnosno da modelovanje nekih ograničenja može značajno da poveća složenost modela, što može da oteža manipulaciju podacima ili degradira performanse sistema u toku eksploracije. Na kraju, neka ograničenja se mnogo lakše implementiraju na aplikativnom nivou.

Na sl. 1.19. prikazan je cijelokupni MOV dijagram koji reprezentuje konceptualni model baze podataka univerziteta u skladu sa specifikovanim informacionim potrebama. Stvarne informacione potrebe su u određenoj mjeri obimnije, a odgovarajući model kompleksniji (dodatni tipovi entiteta, značajno veći broj atributa itd.) od modela koji je projektovan u ovom primjeru. I bez obzira na to, prezentovani primjer dovoljno ilustrativno i sistematizovano prikazuje proces projektovanja, ukazujući na najčešće projektne dileme i alternativna rješenja.



Slika 1.19. Konceptualni model baze podataka univerziteta