

UDŽBENICI SVEUČILIŠTA U RIJECI
MANUALIA UNIVERSITATIS STUDIORUM FLUMINENSIS



prof. dr. sc. Mile Pavlić
OBLIKOVANJE BAZA PODATKA

Izdavač:
Odjel za informatiku Sveučilišta u Rijeci

Recenzenti:
dr. sc. Vladan Jovanović, redoviti profesor, Georgia Southern University, USA
dr. sc. Neven Vrčak, redoviti profesor, FOI, Varaždin
dr. sc. Krešimir Fertalj, izvanredni profesor, FER, Zagreb

Lektorica:
Marijana Trinajstić

Grafička priprema i tisak:
Digital Point tiskara d.o.o., Rijeka

Objavlјivanje ovoga sveučilišnog udžbenika odobrilo je
Povjerenstvo za izdavačku djelatnost Sveučilišta u Rijeci Odlukom pod brojem –
Klasa: 602-09/10-01/39, Ur. broj: 2170-57-05-10-2.

CIP - Katalogizacija u publikaciji

ISBN:

Mile Pavlić

Oblikovanje baza podataka



Rijeka, veljača 2011.

djedu Franji i baki Kati

i

djedu Ivici i baki Rozzi

SADRŽAJ

PREDGOVOR	9
1 UVOD	11
2 INFORMACIJSKI SUSTAVI	15
2.1 Osnovni pojmovi	15
2.1.1 Informacijski sustav	15
2.2 Baza podataka	19
2.2.1 Projektiranje informacijskoga sustava	25
2.2.2 Proizvodnja softvera	29
2.2.3 Uvođenje u primjenu	30
2.3 Specijalizirana metodologija MIRIS	31
3 RELACIJSKE BAZE PODATAKA	35
3.1 Uvod u relacijsku metodu	35
3.2 Struktura relacijskoga modela	36
3.2.1 Koncepti strukture relacijskoga modela	36
3.2.2 Primjer relacijske baze podataka	38
3.3 Ograničenja relacijskoga modela	43
3.3.1 Svojstvo 1. - svi elementi skupa različiti su	43
3.3.1.1 Integritet ključnih atributa	44
3.3.2 Svojstvo 2. - poredak elemenata skupa nebitan je	45
3.3.3 Svojstvo 3. - relacija je podskup Kartezijsevoga produkta	45
3.3.4 Eksplicitna ograničenja	45
3.4 Operatori relacijskoga modela	46
3.4.1 Relacijska algebra	46
3.5 SQL - standardni upitni jezik za relacijske baze podataka	54
3.5.1 Osnove SQL-a	54
3.5.2 Tipovi podataka, operatori i funkcije SQL-a	55
3.5.2.1 Tipovi podataka	55
3.5.2.2 Operatori SQL-a	56
3.5.2.3 Funkcije SQL-a	57
3.5.3 Naredbe SQL - DML	58
3.5.3.1 SELECT - Naredba za pretraživanje	58
3.5.4 Naredbe SQL - DDL	67
3.5.4.1 CREATE TABLE	68

4 MODELI PODATAKA	69
4.1 Problemi, strategije i kriteriji modeliranja podataka.....	69
4.2 Osnovni pojmovi o modeliranju podataka.....	71
4.3 Apstrakcija podataka	74
4.3.1 Klasifikacija	75
4.3.2 Agregacija	75
4.3.3 Generalizacija	76
5 METODA ENTITETI-VEZE	77
5.1 Uvod u metodu ENTITETI - VEZE	77
5.2 Koncepti strukture metode entiteti - veze.....	78
5.3 Entitet.....	81
5.4 Ograničenja metode entiteti - veze	85
5.4.1 Posebna ograničenja	85
5.5 Tip vrijednosti i njegova ograničenje	86
5.6 Atribut	87
5.6.1 Ograničenje na dopuštene vrijednosti atributa	90
5.6.2 Ključ tipa entiteta	90
5.6.3 Vrste atributa.....	92
5.6.4 Brojnost preslikavanja atributa.....	95
5.7 Tip veze	98
5.7.1 Brojnost tipa veze.....	103
5.8 Jak i slab tip entiteta.....	107
5.9 Generalizacija i specijalizacija	113
5.10 Povratni tip veze.....	117
5.11 Agregacija	119
5.12 Operatori metode entiteti - veze	125
5.12.1 Chen operatori	126
5.12.1.1 Operacija 1. Dodavanje.....	126
5.12.1.2 Operacija 2. Izmjena	126
5.12.1.3 Operacija 3. Brisanje	126
5.13 Varijante metode entiteti-veze	127
5.14 Meta modeli	131
5.14.1 Meta model metode entiteti-veze	131
5.14.2 Meta model relacijske metode.....	132
5.15 Primjer EV modela podataka.....	134
6 PREVOĐENJE	139
6.1 Usporedba struktura i ograničenja relacijske i EV metode	139
6.2 Pravila prevođenja modela EV-a u RM	140
7 NORMALIZACIJA	147

7.1 Normalizacija podataka	147
7.1.1 Prva normalna forma (1. NF)	149
7.1.2 Druga normalna forma (2. NF)	153
7.1.3 Treća normalna forma (3. NF)	156
7.1.4 Boyce-Coddova normalna forma (BCNF)	159
8 PROCES KONCEPTUALNOG OBLIKOVANJA PODATAKA	163
8.1 Uočiti podatak	165
8.2 Odrediti tip vrijednosti (odakle ta vrijednost dolazi)	166
8.3 Imenovati atribut čija je to vrijednost	167
8.4 Uočiti entitet koji ima taj atribut	167
8.5 Klasificirati entitete u tip entiteta	169
8.6 Pronaći ključ tipa entiteta	169
8.7 Odrediti brojnost tipa veze	170
8.8 Odrediti brojnosti atributa	171
8.9 Odrediti svojstva atributa	171
8.10 Dodati izvedene atrubute	171
8.11 Dekomponirati složene tipove veza	171
8.12 Specijalizirati podtipove tipa entiteta	172
8.13 Definirati specijalne tipove veza i slabe tipove entiteta	173
8.14 Definirati povratne tipove veza	174
9 PRAKTIČNI PRIMJERI MODELIRANJA PODATAKA	175
9.1 Osobna iskaznica	175
9.2 Narudžbenica	178
9.3 Račun-otpremnica	181
9.4 Skladišna primka	182
9.5 Skladišna kartica	184
9.6 Recept	185
9.7 Prijava štete od požara, eksplozije i drugih opasnosti	186
9.8 Primitak novca	189
9.9 Račun za prikupljanje RTV – pristojbe	192
9.10 Indeks (2 strane)	194
9.11 Putni nalog	197
9.12 Tečajna lista	200
10 LITERATURA	201

PREDGOVOR

S osobitim zadovoljstvom preporučam ovu knjigu informatičarima zainteresiranim za stjecanje neophodnih znanja i vještina pri izradi modela podataka u kontekstu integriranih informacijskih sustava. Iako je modeliranje ključna kompetencija u oblikovanju baza podataka, na žalost, čak i u svijetu, posvećen joj je malen broj kvalitetnih knjiga. Smatram da je ovakva situacija posljedica nedostatka iskustva, tj. prilike za aktivno oblikovanje komercijalnih baza podataka. Profesor Pavlić je jedan od rijetkih koji je znanje modeliranja stjecao u praksi, a posjeduje sposobnost apstrakcije neophodnu pri koncipiranju generalnih znanja.

Pred vama je knjiga pisana jasnim stilom, koncizan priručnik bogatoga sadržaja na manje od 200 stranica. Posebno poglavlje čini čak dvanaest praktičnih primjera. Odabir izdvojenih primjera doprinosi bržem čitanju i istovremeno daje punu slobodu da se primjeri biraju u slijedu primjerenom interesu svakoga čitatelja. Jedno je sigurno, nakon korištenja knjige i ponuđenih primjera, čitatelji će biti bogatiji za značajno iskustvo i steći će samopouzdanje kojim će se uspješno posvetiti oblikovanju baze podataka.

Pitanje izbora notacije modeliranja, tj. dijagrama reprezentacije, nije jednostavno. Knjizi je cilj što lakše naučiti čitatelja kako modelirati i u tu svrhu koristiti semantički bogat model. Očito je da će nakon usvajanja semantički bogatih koncepta biti lakše naučiti modelirati. Nakon naučenoga modeliranja, sama notacija neće predstavljati problem. Alternative, kao što su UML ili IDEF1X, popularne su i podržavaju ih CASE alati, autor je iz vlastita iskustva zaključio da je predložena notacija uspješna u fazi učenja konceptualnoga modeliranja te da će pomoći usvajanju drugih notacija kada se za njima ukaže potreba. Pitanje notacije uvijek je kontroverzno jer ne postoji jedna notacija odgovarajuća svim zahtjevima. Za crtanje modela u bilo kojoj notaciji može se adaptirati Visio stencil i izbjegći kupovina skupih alata.

Slijed autorova izlaganja u skladu je s modernim pristupom predavanju i praksi oblikovanja baza podataka što se u punoj mjeri odnosi i na dva pomoćna poglavlja, treće i sedmo. Kao preduvjet validaciji modela podataka zahtijeva se umijeće oblikovanja upita nad bazom podataka korištenjem SQL-a jer se ne može očekivati korištan model, ukoliko ne postoji jasna predodžba o pitanju koje se otvara mogućoj budućoj bazi podataka. Kao drugo, kada se logički dizajn (relacijska shema) izvede iz modela podataka, potrebna je validacija kvalitete samoga dizajna. Normalizacija, obrađena u sedmome poglavljju, služi prvenstveno izbjegavanju anomalija u ažuriranju podatka, do anomalija može doći u slučaju primjene baze s redundantnim sa-

držajem. Vrijednost dobrog modeliranja očituje se u kvalitetnome dizajnu koji ne treba dalje normalizirati jer je dobrom modeliranjem normalizacija već ostvarena, svaki podatak na jednometu mjestu i jedno mjesto za svaki podatak.

Na kraju, zadovoljstvo je vidjeti kako škola modeliranja podataka napreduje u rukama sposobnog učenika. Prije 25 godina autor knjige bio je gdje ste sada, dragi čitatelji, i Vi kao studenti, na početku karijere, a sada se, jasnoćom ove knjige, i Vama otvara mogućnost uspješne karijere.

21.08.2010.

Vladan Jovanovic, Ph.D.

Professor Computer Sciences

Georgia Southern University

Statesboro GA, USA

Cilj je ove knjige pomoći oblikovati bazu podataka. Većina poslovnih aplikacija podatke čuva u relacijskim bazama podataka te je potrebno poznavati teoriju oblikovanja relacijskih baza podataka (nazovimo je relacijska metodu, skraćeno RM). Oblikovanje baza podataka (engl. Database Design) može se izvesti relacijskom metodom na logičkoj razini i definirati shemu baze podataka. Na osnovi sheme se, uz pomoć softverskih sustava za upravljanje relacijskom bazom podataka (kao Oracle, MS SQL, DB2...) organizira fizička baza podataka. Fizičko i logičko oblikovanje baza podataka dva su odvojena koraka. Oblikovanje baza podataka općenito sastoji se od triju koraka i to konceptualnog, logičkog i fizičkog oblikovanja.

Standardna metoda za konceptualno oblikovanje je EV (metoda entiteta i veza, engl. ER) koja dovodi do konceptualnoga prikaza poslovnog sustava. Model podataka dobi-ven EV metodom jednostavno je prevesti u relacijsku shemu baze podataka i s njom, uz pomoć sustava za upravljanje bazom podataka, oblikovati fizičku bazu podataka.

U ovoj knjizi bit će prikazane navedene metode, EV i relacijska metoda te proces modeliranja podataka obadvjema metodama, kao i proces i pravila prevođenja EV modela podataka u relacijsku shemu.

Kome je knjiga namijenjena?

Ova je knjiga proistekla iz želje i s nadom da će spoznaje biti korisne praktičari- ma koji projektiraju informacijske sustave (oblikuju bazu podataka i oblikuju pro- gramski proizvod) i studentima, kako za modeliranje podataka tako i za razvoj in- formacijskih sustava nakon formalnoga školovanja.

Knjiga je namijenjena projektantima informacijskih sustava, analitičarima, orga- nizatorima, programerima, rukovoditeljima informatičkih centara, studentima i svima onima koji u poslovno - proizvodnim sustavima sudjeluju u razvoju informacijskih sustava i polaznicima seminara o projektiranju informacijskih sustava, pro- jektiranju baza podataka, modeliranju podataka, sistem analizi i dizajnu. Knjiga će poslužiti studentima redovitog i postdiplomskog obrazovanja.

Zašto čitati ovu knjigu?

Ukoliko imate problema s informacijskim sustavom, bazom podataka i pro- gramskom podrškom, potrebno Vam je znanje o tome kako izgraditi informacijski sustav, oblikovati bazu podataka i oblikovati programsку podršku.

Knjiga obiluje objašnjenjima i definicijama vezanim uz metode za oblikovanje baza podataka.

Knjiga je rezultat višegodišnjeg iskustva u projektiranju i gradnji informacijskih sustava, baza podataka, različitih projekata i primjeni metoda za oblikovanje pa su u knjizi iznjeta iskustva i mišljenje o metodama.

Prikazane metode primjenjive su kako za razvoj novih informacijskih sustava tako i za reinženjeringu postojećih programske rješenja i baza podataka.

Koje teme obrađuje knjiga?

Središnji je dio knjige metoda entiteta i veza. Prikazani su svi koncepti potrebni za izradu modela podataka. Prikazana je relacijska metoda. Većina baza podataka su danas relacijske baze podataka. Modeliranje podataka je lakše izvesti metodom entiteta i veza te se ona preporuča kao metoda za oblikovanje baza podataka. Nakon oblikovanja pristupamo postupku prevođenja EV modela podataka u relacijsku shemu baze podataka. Posebno poglavje detaljno opisuje ovaj proces prevođenja. Prikazana je normalizacija podataka i njezino tumačenje dano konceptima metoda entitetima veza. Proces izrade modela podataka prikazan je kao zasebna metoda VATAK (opisana u 8. poglavljtu) koja pomaže projektantima pri brzoj i kvalitetnoj izradi modela podataka. Na kraju je dano desetak primjera dokumenata i njihovih modela podataka.

Knjiga ima devet poglavlja. Prvo je poglavje uvodno.

Poglavlje 2 Informacijski sustavi prikazuje položaj baze podataka u informacijskome sustavu. Definirani su pojmovi vezani uz baze podataka. Prikazano je projektiranje informacijskih sustava, proizvodnja softvera i uvođenje gotovih aplikacija. Iz teorije i prakse razvijena je i u knjizi prikazana vlastita specijalizirana metodologija MIRIS, metodologija za analizu i oblikovanje informacijskih sustava, kao mogući okvir svim fazama razvoja od strateškoga planiranja do održavanja. Oblikovanje baza podataka samo je jedna od faza razvoja informacijskoga sustava. Postoje druge brojne slične specijalizirane metodologije koje su standardizirali pojedinci ili tvrtke, ali gotovo svaki programer ima vlastiti nestandardizirani način razvoja informacijskih sustava.

Poglavlje 3 Relacijska baza podataka prikazuje pojmove koji čine strukturu relacijske metode i pojmove ograničenja u relacijskome modelu iz kojih slijedi pojam ključnog atributa. Ukratko su prikazani operatori relacijske algebre pri čemu je SQL - standardni upitni jezik za relacijske baze podataka, prikazan ukratko.

Poglavlje 4 Modeli podataka prikazuje osnovne pojmove i metode modeliranja podataka. Posebno je opisana apstrakcija podataka koja se nalazi u osnovi svih metoda.

Poglavlje 5 Metoda entiteta i veza prikazuje jedan po jedan koncepte metode EV detaljno i predstavlja središnji dio ove knjige. Uz svaki koncept odmah se izlažu njegova ograničenja i prikazuje njegovo prevođenje u shemu relacijske baze podataka.

Poglavlje 6 Prevođenje definira pravila za prevođenje dijagrama dobivenih metodom EV-a u relacijsku shemu baze podataka.

Poglavlje 7 Normalizacija je uobičajen prikaz pogreške pri organiziranju sheme baze podataka te procesa otklanjanja jedne po jedne pogreške sve dok se shema ne dovede u oblik koji ne stvara probleme u održavanju baze podataka. Pored uobičajenog prikaza tablice, prikazan je i model podataka metodom EV-a i to prije i nakon normalizacije. Ovim načinom grafički se predočuju pogreške koje pri modeliranju uzrokuju probleme u bazi podataka.

Poglavlje 8 Proces oblikovanja baze podataka pokazuje proceduru pri izradi modela podataka metodom EV-a. Sama procedura je zasebna metoda modeliranja podataka i kao takva može se, s manjim prilagodbama, koristiti za različite varijante metode EV-a.

Poglavlje 9 Praktični primjeri modeliranja podataka prikazuje konkretnе popunjene dokumente iz poslovnoga sustava te za njih daje prikaz modela podataka. Tako čitatelj može vidjeti konkretnu primjenu metode EV-a i prevođenja dijagrama EV-a u relacijsku shemu. Na primjerima su također postavljeni zadaci koje treba dodatno riješiti proširenjem modela podataka zbog korisničkih zahtjeva.

Prilozi knjizi

Na internetskoj adresi: <http://tinyurl.com/oblikovanje-baze-podataka> nalaze se korisni tekstovi i koncepti poput, na primjer projektne dokumentacije koju su izradili analitičari i studenti. Takvim pristupom čitatelju se omogućuje uvid u primjenu navedenih metoda i potpuni oblik informacijskoga sustava dokumentiran fazama analize i oblikovanja.

Zahvala

Zahvaljujem mnogobrojnim prijateljima i suradnicima s kojima sam radio na nizu projekata u okviru kojih smo učili i isprobavali metode.

Zahvaljujem brojnim učiteljima, prijateljima, kolegama i partnerima na savjetima i pomoći u promišljanjima o temama, mogućnostima, primjerima i praktičnim problemima informacijskih sustava.

Zahvaljujem programerima u tvrtki RIS na suradnji pri izradi mnogobrojnih primjera softvera i informacijskih sustava.

Zahvaljujem mnogobrojnim suradnicima s HRT-a, iz Croatia osiguranja, Croatia banke, Elektromaterijala, Konta i drugim poslovnim organizacijama na suradnji, na njihovoj želji za unapređenjem informacijskoga sustava i njihovoj vjeri u naše napore.

Posebno zahvaljujem Martini Ašenbrener, za golem trud u izradi knjige, prikupljanju i obradi izvorne građe, podršci i vjeri u uspjeh.

Zahvaljujem studentima Odjela za informatiku, Sveučilišta u Rijeci, mojim magistrantima i doktorandima, polaznicima jednogodišnje Škole za projektiranje informacijskih sustava na njihovim mnogobrojnim sugestijama, pitanjima, diplomskim, seminarским, magistarskim, doktorskim radovima koji su pridonijeli nastanku ove knjige.

Rijeka, siječanj 2011.
prof. dr. sc. Mile Pavlić

2 INFORMACIJSKI SUSTAVI

2.1 Osnovni pojmovi

2.1.1 Informacijski sustav

Pojam informacijski sustav (skraćeno IS) sastoji se od dvaju pojmova, termina koje je potrebno definirati: „sustav“ i „informacija“.

Sustav (engl. *system*) je skup dijelova (elemenata), veza između dijelova te osobina dijelova svrshishodno organiziranih za neki proces (funkciju).

Svaki sustav nalazi se u širemu sustavu kojemu pripada i s kojim je u vezi. Dio sveukupne cjeline koji nije obuhvaćen sustavom naziva se **okolina sustava**. Okolina je skup okolnih sustava. Vezu sustava i okoline predstavljamo ulazima kada okolina predaje sustavu materiju, energiju ili informacije, a izlazima kada to sustav predaje okolini. Na slici 2.1. prikazan je osnovni model sustava.



Slika 2.1 Osnovni model sustava

U ljudskome društvu postoje mnogobrojne formalne i neformalne organizacije. Svaka organizacija je sustav.

Poslovna organizacija (poslovni sustav, poduzeće, ustanova, tvrtka, engl. *business organization*) je organizacija koja se bavi jednom ili više poslovnih djelatnosti kao svojom misijom. Svaka poslovna organizacija ima informacijski sustav.

Poslovna djelatnost (engl. *business sector*) je sve ono što poslovna organizacija čini kako bi ostvarila prihode, bilo izravno (kao što je izrada proizvoda za prodaju) ili neizravno (kao što je pružanje usluga – na primjer konzultant koji savjetuje poslovne organizacije kako poboljšati svoj proizvod) (CEO, 2005.).

Za poslovne organizacije i informacijske sustave temeljni pojmovi su podatak i informacija. Za podatak kažemo da je nositelj informacija, a informacija je podatak koji su ljudi protumačili. Za te pojmove postoje brojne definicije (samo ISO ima u svojoj dokumentaciji preko 50 definicija pojma informacija, vidi <http://cdb.iso.org/cdb/search.action>). Ovdje ćemo navesti samo neke.

Podatak (engl. *data*) je bilo koji predmet mišljenja koji može prenijeti informaciju; formalizirani znakovni prikaz činjenica, pojmove i naredaba pogodan za priopćavanje, interpretiranje te analognu i digitalnu obradu (Kiš, 2002.).

Informacija (engl. *information*)¹ je takav podatak koji u tijeku privredne aktivnosti donosi nova znanja koja nam na osnovi postojećih znanja i iskustava govore o promjeni nekoga stanja unutar poduzeća ili u okruženju. To je podatak koji je prethodno provjerен i analiziran te kao takav predstavlja prikladnu osnovu za donošenje odluka o utjecaju na daljnji tijek privredne aktivnosti (Papp, 1979.).

Podatak je diskretna vrijednost neke fizičke veličine, na primjer 67 je broj koji se sastoji od dva znaka. Podatak je skup znakova. Znajući vrijednost broja 67, a ne značući kojoj vrsti stvari u sustavu pripada i što znači, kažemo da imamo neki podatak. Podaci su nositelji informacija. Podaci se mogu zapisivati na različitim medijima.

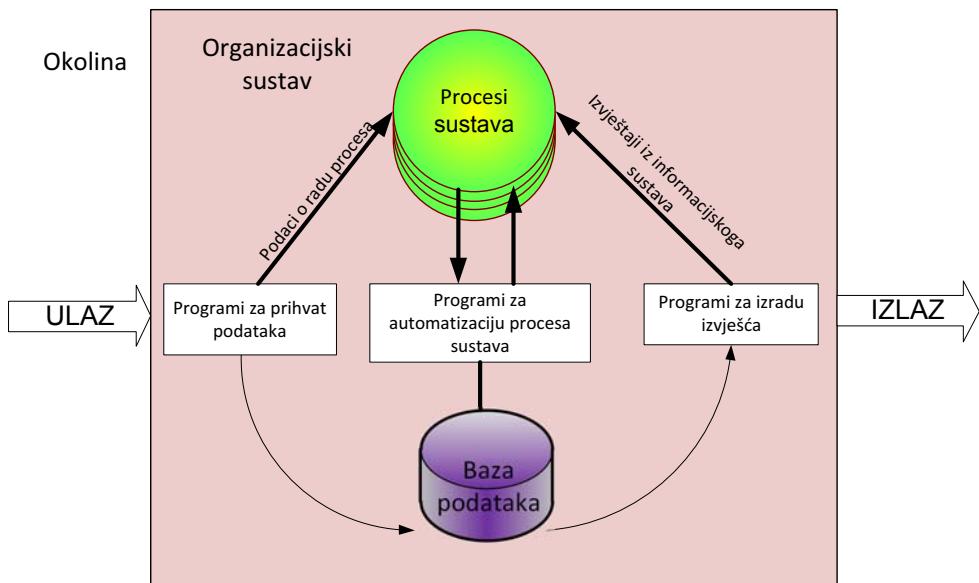
Ako kažemo (ili pomislimo, ili ako znamo, ili je na papiru zapisana činjenica), „visina stola je 67 cm“, onda imamo informaciju. Ova zapisana činjenica je informacija svakoj osobi koja će je znati protumačiti iz svoga konteksta znanja o stvarnosti. Za osobu koja ne zna pročitati rečenicu na tom jeziku, ta činjenica je podatak. Dakle, ovisno o kontekstu, informacija za nekoga može biti podatak i obrnuto.

Informacijski sustav (engl. *Information System, skraćeno IS*) je objedinjeni skup komponenti (računalnog hardvera i softvera, baza podataka, telekomunikacijskih sustava, ljudskih resursa te procesa) koje služe za prikupljanje, pohranu, obradu te prenošenje informacija (vidi <http://www.britannica.com/EBchecked/topic/287895/information-system>).

Informacijski sustav je skup povezanih dijelova (softver, hardver, ljudi, procedure, informacije, te komunikacijske mreže) kojima je cilj pribaviti, pohraniti, obraditi i prenijeti informacije za funkcioniranje, planiranje, odlučivanje i/ili upravljanje poslovnom organizacijom.

Informacijski sustav zasnovan na računalu čine ljudi (obrazovani za rad s računalima) te programska i računalna oprema (softver i hardver) koja je napravljena, oblikovana i dovedena u operativno stanje te se pomoću nje informacije skupljaju (engl. *data acquisition*), zapisuju (engl. *recording*), obrađuju (engl. *processing*), spremanju i pronalaze (engl. *information storage and retrieval*) te prikazuju u odgovarajućem obliku (Kiš, 1993.).

¹ U hrvatskome jeziku naziva se još i „obavijest“. Postoji velik broj definicija pojma informacije s različitim aspekata kao: obavijest, neizvjesnost, komunikacija, semantika, događaj, struktura, znanstvena činjenica, svojstvo materije.



Slika 2.2 Dijelovi informacijskoga sustava i njihova povezanost s okolinom

Informacijski sustav, sa stajališta osnovnih komponenti strukture, pogodno je prikazati grafički (vidi sl. 2.2.).

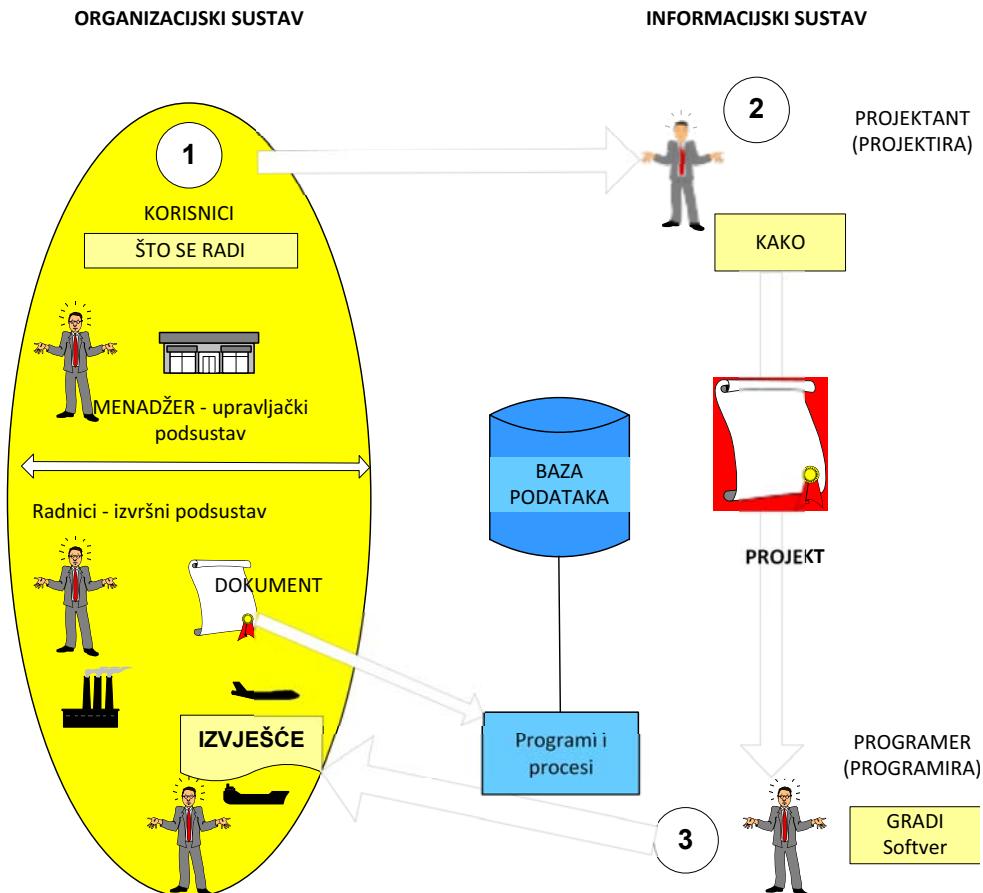
Börje Langefors je definirao osnove teorije informacijskih sustava². Većinu svojih istraživanja posvetio je generalnoj teoriji koja definira tu disciplinu, neovisno o tehnologiji koja se stalno razvija. Langeforsova definicija glasi: "Informacijski sustav temeljen na računalima je tehnološki ostvaren medij za zapisivanje, spremanje i razumijevanje te širenje lingvističkih izraza, kao i zaključivanje iz istih."

Informacijski sustav, sa stajališta funkcije u organizacijskom sustavu, ima zadatak permanentno opskrbljivati potrebnim podacima poslovne procese i sve razine upravljanja tim procesima te odlučivanja u tim procesima.

Temeljne funkcije informacijskoga sustava su:

- prikupljanje
- upis (pohranjivanje, engl. storage) podataka u bazu podataka
- obrada (procesiranje, engl. processing) podataka
- prikaz i ispostavljanje (izdavanje na korištenje, engl. dissemination) podataka iz baze podataka
- čuvanje (dokumentiranje, trajno pohranjivanje) podataka.

² Vidi http://hr.wikipedia.org/wiki/Informacijski_sustavi, 19.4.2010.



Slika 2.3 Informacijski sustav kao integralni dio organizacijskoga sustava

Informacijski sustav je dio organizacijskoga sustava (vidi sl. 2.3) koji podacima povezuje cijeli sustav i koji se kontinuirano održava.

U širemu smislu dijelovi informacijskoga sustava su: dokumenti s podacima, ljudi koji koriste i kreiraju dokumente, uređaji za manipuliranje dokumentima, računalna tehnologija, baza podataka, ljudi koji upisuju podatke u bazu podataka i komunikacijska tehnologija.

Skup elemenata koji iz okoline djeluju na sustav naziva se „**ulaz u sustav**“ (vidi sl. 2.2). Ulazi u sustav su informacije koje sustav prima iz okoline. To mogu biti: nacrti, računi, dostavnice, telefonske poruke, izvodi banke, izvješća, usmene poruke, novine i dr. Podaci s ulaza u sustav prepisuju se u bazu podataka pomoću programa za prihvatanje podataka.

Programi za prihvat ulaznih podataka važan su dio informacijskoga sustava, jednakao kao i programi za izvješćivanje. Prvi prihvaćaju podatke i smještaju ih u bazu podataka za buduće potrebe kada će iz njih programi za izvješćivanje kreirati potrebne skupove podataka.

Baza podataka je mjesto gdje se čuvaju podaci. Baza podataka može biti različitih vrsta i oblika, na raznim tehnologijama. Najvažnija je značajka baze podataka, tj. ono što je čini bazom podataka, što ona sadrži podatke o prošlosti i sadašnjosti organizacijskoga sustava.

Prihvatom i upisom podataka u bazu podataka informacijski sustav je ostvario svoju prvu temeljnu funkciju. Temeljna je funkcija organizacijskoga sustava korištenje ulaza u sustav kako bi na osnovi njih poslovni procesi sustava kreirali nove informacije koje mogu ići kao ulazi u druge procese ili predstavljati **izlaze iz sustava**. Ako se neki od procesa poslovnoga sustava može potpuno ili djelomice automatizirati i ugraditi u informacijski sustav, onda informacijski sustav svojim radom obavlja temeljne funkcije organizacijskoga sustava.

Druga temeljna funkcija informacijskoga sustava je izvođenje poslovnih procesa organizacijskoga sustava. Što je više poslovnih procesa organizacijskoga sustava automatizirano i ne izvode ih ljudi, već IT (informacijska tehnologija) uz pomoć i kontrolu ljudi, to je organizacijski sustav efikasniji i ekonomičniji. Cilj je automatizirati što više poslovnih procesa (na primjer uvesti bankomate umjesto šalterskih radnika za podizanje novca, uvesti aplikaciju za obradu plaća ili završni račun tvrtke koja će omogućiti izradbu više varijanti plaćanja poreza, uvesti robote za automatsko rezanje limova u tvornici na osnovi nacrta projektnog ureda i sl.)

Izvješćivanje je proces transformacije podataka iz baze podataka u podatke potrebne korisnicima. Podaci se prikazuju na izvješćima u analitičkom ili sintetičkom obliku, u većem ili manjem obimu. Izvješće može biti na različitim medijima, papirnatima, od bušene trake do A0, mikrofilmovima, magnetskim medijima svih vrsta, optičkim diskovima, e-mail i drugim porukama, elektromagnetskim signalima emitiranim različitim komunikacijskim kanalima u uređaje primateljima i dr. Nekada je to bio uglavnom papir, a danas je to najčešće zaslon računala.

2.2 Baza podataka

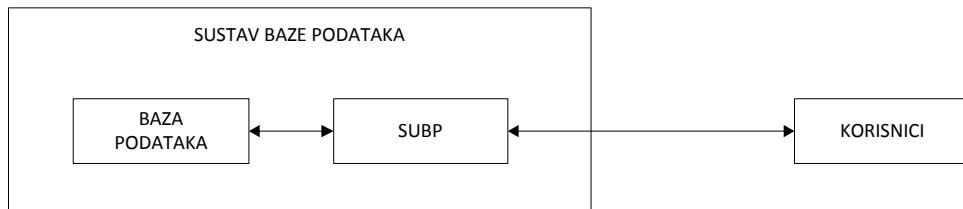
Baza podataka (skraćeno BP, engl. *Database*) predstavlja neredundantni skup podataka o stanju sustava strukturiran na način opisan u shemi baze podataka. Baza podataka je skup podataka o svim pojavljivanjima entiteta, svih veza i svih njihovih atributa opisanih u shemi baze podataka.

Baza podataka je skupina ustrojenih, logički povezanih zapisa (engl. *Record*) ili datoteka (engl. *File*); ili skupina datoteka koje sadrže zapise s podacima što su međusobno u nekoj vezi, a korisnici ih mogu rabiti u različite svrhe; može se sastojati i od pomoćnih datoteka (na primjer datoteke s indeksima) (Kiš, 1993.).

Shema baze podataka je zapis konkretnе podatkovne strukture i ograničenja među podacima poslovne organizacije odabranom metodom za modeliranje podataka.

Sustav baze podataka je baza podataka i sustav za upravljanje bazom podataka (vidi sl. 2.4.).

Sustav za upravljanje bazama podataka (skraćeno SUBP, engl. *Database Management System (DBMS)*) je softver koji upravlja korištenjem baze podataka, odnosno ima ulogu posrednika između korisnika i same baze podataka. Neki od poznatijih SUBP-ova dani su u tablici 2.1.



Slika 2.4 Struktura sustava baze podataka

Konkretni SUBP-ovi sastoje se iz dvaju dijelova:

- jezika za opis podataka (skraćeno JOP) i
- jezika za rukovanje podacima (skraćeno JRP)

Jezik za opis podataka (engl. *Data Definition Language*, skraćeno DDL) je skup sintaksnih pravila pomoću kojih je moguće definirati (opisati) shemu baze podataka.

Jezik za rukovanje podacima (engl. *Data Manipulation Language*, skraćeno DML) je računalni jezik pomoću koga programeri ili korisnici mogu dodavati, bri-sati ili mijenjati podatke u bazi podataka.

Pored operatora u DML jeziku postoje i procedure baze podataka za:

- očuvanje integriteta
- optimiziranje
- praćenje stanja baze podataka
- statističke operacije

Generalizacija tipova SUBP-ova koji postoje, razvijenih na osnovi tipova modela podataka su:

- hijerarhijski
- mrežni
- relacijski
- postrelacijski (entity – relationship)
- objektni

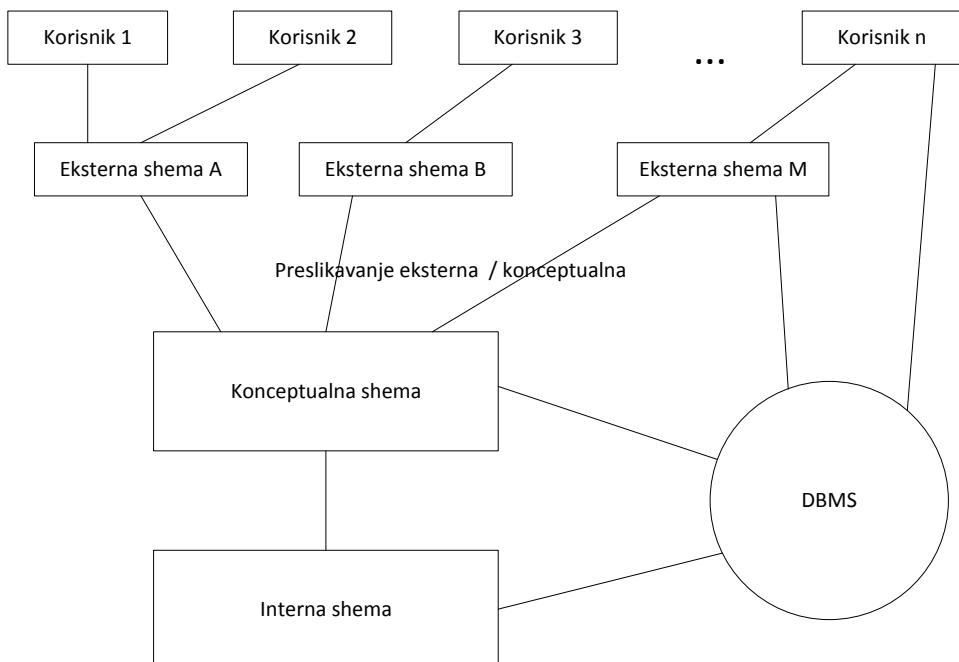
Tijekom razvoja baza podataka razvijan je i DDL te tako postoje generacije arhitekture DDL-a. U ovisnosti o stupnju razdvojenosti fizičke organizacije podataka (interne sheme) u bazi podataka od pogleda na podatke (eksterne sheme) iz aplikacijskoga programa, postoje generacije SUBP-a:

I. generacija: Interna shema → aplikacijski program

II. generacija: Interna shema → konceptualna shema → aplikacijski program

III. generacija: Interna shema → konceptualna shema → eksterna shema → aplikacijski program

Svaka promjena konceptualne sheme zahtijeva izmjenu aplikacijskih programa i potreba za održavanjem softvera i dalje postoji.



Slika 2.5 Tri razine arhitekture baze podataka

ANSI (*American National Standards Institute*) iz skupine SPARC (*Standarts Planning And Requirements Committee*), predložio je zbog prije navedenih razloga, idealnu arhitekturu DDL, organiziranu u tri odvojene razine³ (vidi sl. 2.5).

Interna shema (engl. *internal schema*) je skup definicija o fizičkoj organizaciji baze podataka i putevima pristupa specifičnim za pojedine SUBP-ove. Aplikacijski program u I. generaciji, koji pristupa takvoj shemi, mora sadržavati znanje o fizičkoj strukturi i vezama među podacima te je ovisan o fizičkoj lokaciji podataka u bazi podataka. Problem je izmjena definicije u DDL-u jer to zahtijeva izmjenu aplikacijskih programa.

Konceptualna shema (engl. *conceptual schema*) je skup definicija o logičkoj organizaciji baze podataka. Programi nisu ovisni o promjenama u internoj shemi. Postignut je viši stupanj neovisnosti, ali postoji ovisnost programa o logičkim pozicijama podataka (na primjer relacija u relacijskoj bazi podataka).

Eksterna shema (engl. *external schema*) je skup definicija o podacima kako ih vidi aplikacijski program. Pogled na podatke zaseban je za svaku aplikaciju i pri izmjeni aplikacije mijenja se samo pripadni pogled u eksternoj shemi, neovisno o fizičkoj i logičkoj organizaciji podataka. Pri izmjeni konceptualne sheme ne mijenja se niti jedan aplikacijski program koji nije povezan s danom promjenom. Ako je dodan novi atribut, a želimo ga imati u aplikaciji, mijenja se aplikacijski program. Premještanje atributa, proširenje polja, promjena tipa polja, dodavanje novih atributa i entiteta ne utječe na programe.

Eksterna shema izvodi se iz konceptualne sheme.

Konceptualna shema sadrži definicije o svim podacima, bez zalihosti (redundancije), na jedinstven način, neovisno o fizičkoj implementaciji i aplikaciji. Iz nje se mogu izvesti svi pogledi.

Osnovni je cilj projektanta baze podataka modelirati konceptualnu shemu. Administrator baze podataka kreira internu shemu na osnovi konceptualne i priprema eksternu shemu za pojedini aplikacijski program. Projektanti aplikacija zahtijevaju eksternu shemu. Ako u eksternoj shemi ima atributa koji ne postoje u konceptualnoj, oni se dodaju u konceptualnu i internu shemu.

Ova se knjiga bavi metodama za modeliranje konceptualne sheme, bez obzira na konkretnе SUBP-ove.

Administracija baze podataka različita je za različite SUBP-ove, i detaljnije se izučava pri upotrebi konkretnog SUBP-a.

Neki sustavi za upravljanje relacijskim bazama i relacijske baze podataka prikazani su u tablici 2.1.

³ Vidi <http://db.grussell.org/section002.html>, 6.5.2010.

Tablica 2.1 Sustavi za upravljanje bazom podataka

Naziv alata	Autor	Web
CA-Datacom	Computer Associates	http://www.ca.com/us/products/product.aspx?id=1237
Ingres	University of California, Berkeley	http://www.ingres.com/
DB2	IBM	http://www-01.ibm.com/software/data/db2/
Oracle	Oracle	http://www.oracle.com/index.html
Informix	IBM	http://www-01.ibm.com/software/data/informix/
MySQL	Ulf Michael Widenius	http://www.mysql.com/about/
PostgreSQL	Michael Stonebraker University of California at Berkeley Computer Science Department	http://www.postgresql.org/docs/current/static/intro-whatis.html
Model 204	Computer Corporation	http://www.cca-int.com/prodinfo/m204.html
NonStop SQL	Tandem Computers	http://h20338.www2.hp.com/NonStopComputing/cache/76708-0-0-225-121.html
SQL Server	Microsoft	http://www.microsoft.com/sqlserver/2008/en/us/
SQLBase	Centura Software Corporation	http://www.unify.com/Products/Data_Management/default.aspx
Impromptu	Cognos Inc.	http://www-01.ibm.com/software/data/cognos/products/series7/impromptu/
Teradata	Teradata	http://www.teradata.com/t/
Progress	Progress Software Corporation	http://web.progress.com/en/index.html
Access	Microsoft	http://www.microsoft.com
dBASE	Ashton-Tate Corporation	http://en.wikipedia.org/wiki/Ashton-Tate

Vrlo velike baze podataka (engl. *Very Large Database*, VLDB) dosežu po kapacitetu oko: 128 TB (terabajta), a cijeli sustav baze može imati 64000 takvih baza⁴, što iznosi 8.192.000.000.000.000 znakova ili 8,192 EB-a. Veličina datoteka (ili baze podataka) može se izraziti u jedinicama mjere prikazanima u tablici 2.2.

⁴ podaci su preuzeti s <http://www.oracle-base.com/articles/10g/ImprovedVLDBSupport10g.php> (10.4.2008.)

Tablica 2.2 Jedinice mjere količine podataka

Ime prefiksa	Uobičajen naziv	Simbol	Standard SI ⁵	Potencija broja 2
	jedan	1	100	$2^0 = 1$
deka-	deset	D	101	-
hecto-	sto	h	102	-
kilo-	tisuću	k ili K	103	$2^{10} = 1,024$
mega-	milijun	M	106	$2^{20} = 1,048,576$
giga-	milijarda	G	109	2^{30}
tera-	bilijun	T	1012	2^{40}
peta-	bilijarda	P	1015	2^{50}
exa-	trilijun	E	1018	2^{60}
zetta-	trilijarda	Z	1021	2^{70}
yotta-	kvadrilijun	Y	1024	2^{80}
...				
googol ⁶	gugol	10^7	10100	
googolplex	gugolopleks	$10^{10^{100}}$	$(10)^{10^{100}}$ $(10)^{10^{100}}$	

Prefiksi množitelji: *kilo*, *mega*, *giga* i dr. koriste se u fizici i drugim znanostima, ali i u ICT-u za mjerjenje količine bita (engl. *bit*, skraćeno „b“) i bajtova (engl. *byte*, skraćeno „B“).

Prefiks *kilo* ima kraticu *k* i predstavlja 10^3 , odnosno 1000 jedinica, međutim u informacijskim znanostima, za termine koji su prefigirani tim prefiksom, koristi se kratica *K* (dakle veliko slovo) kako bi bilo jasno da vrijednost *kilo* u tome području predstavlja drugu vrijednost: 2^{10} , odnosno 1024 jedinice. Uobičajeno je da se količi-

⁵ SI je oznaka za međunarodni standard (prema francuskome nazivu Système International d'Unités, engl. International System of Units).

⁶ Gugol je veliki broj. Može se zapisati kao 1 iza koga slijedi 100 nula (Naprimjer broj tisuću je s tri nule, tj. 1000). Procjenjuje se da u cijelom vidljivom svemiru ima najviše 10^{81} atoma. Nazivom za taj broj motivirano je ime poznate softverske kompanije *Google* (ova je tvrtka ispočetka doista i nosila naziv *googol*, međutim ubrzo se na jednomet čeku koji je trebao biti isplaćen na ime tvrtke dogodila pravopisna zabuna. Osnivači tvrtke ocijenili su u tome trenutku da im je spretnije prihvatiti novi način zapisivanja imena, nego tražiti od isplatitelja da ispravi svoju pogrešku te je tako tvrtka dobila ime koje nosi i danas).

⁷ Autor nije našao kraticu za taj i sljedeće termine te je predložio kratice.

⁸ *Googolplex* je naziv za broj (izvedeno iz *googol* i *complex*), a predstavlja broj koji se može zapisati kao 1 iza koga slijedi gugol nula. Taj je broj velik kao broj atoma u gugolu svemira malo većih od našega svemira.

na prenesenih podataka mjeri u bitima s potencijom na bazi 10. Kapacitet memorije mjeri se uobičajeno u bajtovima s potencijom na bazi 2. Tako na primjer kapacitet nekoga medija od 5 KB (kilobajta) predstavlja 5120 znakova. U većih jedinica nema razlike u velikom i malom slovu i u komunikaciji se podrazumijeva zaokružen broj prema SI standardu, dok je stvarni kapacitet obično malo različit.

Velike baze podataka brzo se šire i proizvođači softvera za upravljanje bazom podataka stalno proširuju mogućnosti svojih proizvoda. VLDB održavaju mnogo-brojne ekipe projektanata, analitičara i programera.

U najvećim poslovnim organizacija baza podataka može dnevno porasti za nekoliko gigabajta. Potrebno je, pored upisa podataka, manipulirati i obradivati velike skupove podataka. Takve obrade nad skupinama informacija u gigabajtima podataka o transakcijama usporavaju rad i ometaju funkcioniranje sustava. Stoga se podaci izdvajaju u zasebnu bazu podataka nad kojom se izvode operacije. Tri koncepta koji omogućuju kopanje po podacima, i predstavljaju proširenje informacijskoga sustava i sa stajališta baze podataka i sa stajališta softverskih i metodoloških rješenja su:

- „skladište“ podataka (engl. *Data warehouse*)
- „tržnica“ podataka (engl. *Data mart*)
- „rudarenje“ (prekopavanje) podataka (engl. *Data mining*).

Data Warehouse (skraćeno DW) je baza starih podataka o poslovnim transakcijama organizirana tako da olakša učinkovite upite za donošenje marketinških, taktičkih i strateških odluka.

Data Mart (skraćeno DM) je skupina prikupljenih podataka iz baze podataka, skladišta podataka i drugih izvora koja je oblikovana kako bi omogućila analize prošlih trendova i prošlih iskustava za daljnje strateške odluke.

Data Mining (skraćeno DM) je analiza velikoga broja podataka za veze koje nisu prethodno bile otkrivene pomoću zasebnih metoda.

Na primjer DM analiza podataka može otkriti prodaji da korisnici koji kupuju skupu hranu, kupuju i skupa pića; koji kupuju dječju hranu, kupuju i dječje igračke; koji kupuju pivo, kupuju odmah i grickalice; koji kupuju keramičke pločice, kasnije kupuju predmete za kućanstvo i sl. Data mining je otkrivanje znanja.

2.2.1 Projektiranje informacijskoga sustava

Informacijski inženjering (engl. *Information Engineering*, skraćeno IE) je disciplina koja se bavi svim aspektima razvoja informacijskoga sustava: analizom poslovnih procesa (ne i njihovim inženjeringom), oblikovanjem informacijskih sustava, softverskim inženjeringom, uvođenjem i održavanjem informacijskih sustava. IE obuhvaća sustavnu primjenu prikladnoga skupa metoda i alata u procesu razvoja informacijskoga sustava.

Projekt informacijskoga sustava (engl. *design of information systems*) u biti je skup nacrta, sličan projektima u građevini, brodogradnji i drugim strukama koji logički definira sadržaj i strukturu informacijskoga sustava uz pomoć niza modela informacijskog inženjeringu.

Prvi je korak u projektiranju informacijskoga sustava analiza. Analiza predstavlja spoznaju dijelova sustava kao funkcionalne cjeline. Nakon analize slijedi oblikovanje budućega informacijskoga sustava koji će zadovoljiti potrebe i želje korisnika ustanovljene tijekom analize. Analiza i oblikovanje logičke su faze u kojima se komunicira, promišlja i dokumentira informacijski sustav.

Nakon analize i dizajna informacijskoga sustava potrebno je definirati relacijsku shemu baze podataka i napraviti dizajn arhitekture softvera. Navedeno ulazi u projektnu dokumentaciju informacijskoga sustava. Dokumentacija će odgovoriti na pitanja: što i zašto, kako i kada, tko, gdje i s čim.

Projektiranje informacijskih sustava (engl. *information systems designing*) je nalaženje modela informacijskoga sustava. Model informacijskoga sustava sastoji se od najmanje triju modela: modela procesa, modela podataka i modela resursa. U ovoj knjizi detaljno se definira model podataka.

Model procesa (engl. *process model*) prikazuje skup procesa koji prihvaćaju ulaze u sustav, mijenjaju stanje sustava i pomoći kojih se formiraju izlazi iz sustava. Model procesa je skup poslova nad skupovima podataka. Procesi na modelu skupovi su poslovi koji stvaraju ili koriste informacije za svoje funkcioniranje.

Model resursa (engl. *resource model*) specificira tehnološku osnovicu. On prikazuje „procesore“ (kadrove, organizacijske jedinice, IT (opremu) u pogledu njihovih kapaciteta i dinamike korištenja tih kapaciteta) koji omogućuju smještanje i dinamiku podataka i procesa sustava. U modelu resursa skriveni su svi aspekti različiti od podataka i procesa.

Baza podataka (engl. *Database*) predstavlja neredundantni skup podataka o stanju sustava strukturiran na način opisan u shemi baze podataka. Baza podataka je skup podataka o svim pojavljivanjima entiteta, svih veza i svih njihovih atributa opisanih u shemi baze podataka.

Aplikacija (engl. *application*)⁹ je (Kiš, 2002.)

1. kraći naziv za namjenski program; program posebne namjene
2. područje ili vrsta problema na koje se primjenjuju tehnikе obrade podataka.

Simonični su termini za *aplikaciju*: aplikativni program, namjenski program, program s posebnom namjenom, aplikativni softver, aplikacijska programska oprema, namjenska programska oprema, poslovna aplikacija.

⁹ Pod pojmom aplikacija podrazumijeva se pojam softverska aplikacija.

Aplikacija je računalni program stvoren kako bi obavljao neku zadaću (Anić, 2002). Aplikacija je programski proizvod i baza podataka koji obavljaju određenu poslovnu funkciju ili njezin kohezivni dio.

Proces izrade softvera (softverski proces, faze softverske proizvodnje, engl. *software process*) je definiranje skupina aktivnosti i redoslijeda izvođenja aktivnosti tako da se redom izrađuju dokumentacija i softverski moduli sve do konačnoga integriranog softverskog proizvoda.

Analiza informacijskih sustava je proces u kome analitičar proučava postojeći poslovni sustav i njegov informacijski sustav kako bi omogućio kreiranje modela novoga stanja procesa sustava i novi informacijski sustav. Rezultati analize su: softverski zahtjevi, potrebni podaci, izlazne informacije, zadaci ljudima, logika procedura.

Analitičar (engl. *analyst*) je osoba koja provodi analizu, tj. traži veze, uzroke i posljedice te izvodi zaključke pomoću rastavljanja cjeline na sastavne elemente (Anić, 2002.).

Korisnici (engl. *users*) su radnici u poslovnoj organizaciji i njezinoj okolini koji imaju koristi od sustava (bilo da unose podatke u bazu podataka ili imaju mogućnost pregleda izvještaja). Pod korisnicima podrazumijevamo radnike u sustavu, odnosno izvršitelje, na bilo kojoj hijerarhijskoj razini poslovne organizacije.

Proces (engl. *process*) je skup povezanih aktivnosti i odluka, preko kojih elementi sustava ostvaruju dijelove cilja svoga postojanja, a za njihovo izvršenje potrebni su određeni resursi i određeno vrijeme.

Procesi se sastoje od potprocesa koji se na najnižoj razini raščlanjivanja poklapaju s aktivnostima.

Aktivnosti (engl. *activities*) su jednostavni poslovi od kojih se sastoje procesi i funkcije.

Funkcija (engl. *function*) je apstraktan naziv za skupinu sličnih poslova koji ostvaruju jedan parcijalni interni cilj jednoga organizacijskog dijela poslovne organizacije.

Cilj je analize sustava istražiti procese koji prihvataju ulaz, istražiti što se sve s ulazom događa dok se on ne transformira u izlaz i sustav ne postigne svoj cilj. Transformaciju ulaza u izlaz u okviru sustava obavljaju procesi.

Jedna od poznatih metoda (često u literaturi nazivana tehnika) analize sustava i izrade modela procesa je metoda pod imenom **Dijagram toka podataka** (skraćeno DTP, engl. *Dataflow Diagram*, skraćeno DFD). DTP je grafičko sredstvo za modeliranje i prezentaciju procesa sustava. Mnoge metode za modeliranje procesa koriste jednu od varijanti DTP-a za prikaz modela procesa.

Odlike DTP-a su:

- DTP je grafički prikaz procesa
- korisnik i analitičar zajedno dolaze do modela procesa
- precizno se definiraju zahtjevi korisnika
- DTP je jezik za komunikaciju korisnika i analitičara
- malen je broj različitih koncepata na DTP-u (četiri osnovna simbola).

Skup osnovnih koncepata za gradnju dijagrama toka podataka (Yourdon, 1979.) prikazan je na slici 2.6.

Dizajn informacijskoga sustava i metode za modeliranje podataka opisane su detaljnije u drugim poglavljima.

KONCEPT	SIMBOL po DeMarcu, Yourdonu	SIMBOL po Ganeu i Sarsonici
TOK PODATAKA predstavlja se vektorom ili usmjerenim lukom		
PROCES (funkcija) predstavlja se ovalom, elipsom, krugom i slično		
SPREMIŠTE (skladište) podataka predstavlja se dvjema paralelnim crtama		
VANJSKI SUSTAV (izvoriste ili odredište, granični entitet) predstavlja se pravokutnikom		

Slika 2.6 Koncepti DTP-a

2.2.2 Proizvodnja softvera

Definirajmo neke važnije pojmove vezane uz proizvodnju softvera.

Implementacija (engl. *implementation*) je rad na provođenju čega; provedba, izvršenje, primjena, ostvarenje (Anić, 2002.).

Implementacija informacijskoga sustava je skup aktivnosti koji modeli iz projekta informacijskoga sustava pretvara u dijelove strukture budućega informacijskog sustava te novonastale dijelove uvodi u postojeći sustav, mijenjajući postojeći informacijski sustav.

Softversko inženjerstvo (programsко inženjerstvo, engl. *software engineering*) obuhvaća poslove kojima se oblikuje i razvija programska oprema; sastoje se od analiziranja i pobližeg opisivanja postupka koji treba programirati, razvijanja programa, pri čemu se odabire neki od standardnih pristupa, tehnika testiranja, odnosno provjeravanja programa, pisanja dokumentacije, pokusnih izvedbenih programa, analize vremenskog vođenja itd. (Kiš, 2002.).

Softverski proces (engl. *software process*) je skup aktivnosti, metoda, postupaka i transformacija koji se koriste za oblikovanje i održavanje softvera i pratećih produkata kao što su projektni planovi, kodovi, testovi slučaja i priručnici za korisnike (Kellner, 1999.).

Osnovne aktivnosti unutar softverskoga procesa su:

- specifikacija (analiza zahtjeva korisnika; utvrđuje se što softver treba raditi)
- oblikovanje (oblikovanje građe sustava, načina rada komponenti te sučelja između komponenti; projektiranje rješenja koje određuje kako će softver raditi)
- implementacija (oblikovano rješenje realizira se korištenjem raspoloživih programskih jezika i alata)
- verifikacija (provjera radi li softver prema specifikaciji)
- validacija (provjera radi li softver ono što korisnik želi) te
- održavanje (popravljanje, mijenjanje i nadograđivanje softvera u skladu s promjenama potreba korisnika).

Testiranje programa (ispitivanje) je provjera rada programa njegovim izvođenjem, uz uporabu pripremljenih ulaznih testnih podataka te analizom izlaznih podataka dobivenih kao rezultata obrade ulaznih podataka.

Pravilo je da svaki programer provodi testiranje svoga koda.

Verifikacija programa je testiranje kojim programer utvrđuje obavlja li program ono čemu je namijenjen, odnosno ostvaruje li planiranu funkciju.

Alfa-testiranje (engl. *alpha testing*) ili tzv. verifikacijsko testiranje je probna uporaba programskoga sustava koju provode interno testeri tvrtke u kojoj se izrađuje programski proizvod kako bi provjerili odgovara li softverskom projektu. Alfa-testiranje je simulacija rada programa kao dijela stvarnoga informacijskog sustava (probni podaci su realni iz poslovne organizacije, baza podataka je puna, više testera radi istodobno).

Beta-testiranje (engl. *beta testing*) ili tzv. validacijsko testiranje je korisnička probna uporaba i provjera koja podrazumijeva provjeru rada u stvarnim uvjetima pod vršnim opterećenjem mreže i punom bazom podataka tako da se mjeri stvarna brzina pristupa i vrijeme izrade pojedine funkcije. Za generički softver beta-testiranje može izvoditi vrlo velik broj korisnika.

Funkcionalno testiranje (engl. *black-box, functional testing*) je validacija koju izvodi korisnik i provjerava što program radi. Najčešće korištena metoda funkcijskoga testiranja je testiranje domene¹⁰ (engl. *domain testing, partition testing*) (Beizer, 1990.).

CASE alat (*Computer-Aided Software Engineering*) je softverski proizvod koji omogućuje automatizaciju procesa izrade softvera. CASE alat ima u sebi podršku za: dizajn, programiranje i testiranje softverskoga proizvoda. Vrsta CASE alata su i RAD (engl. *Rapid Application Development*) alati koji nemaju podršku dizajniranja, ali imaju podršku za automatsko programiranje i testiranje.

Pod **outsourcingom** (engl. *outsourcing*) podrazumijevamo oduzimanje poslovnih operacija (funkcija, procesa) koje nisu temeljna djelatnost organizacije vlastitim organizacijskim dijelovima i njihovo delegiranje vanjskim specijaliziranim organizacijama za vođenje tih operacija. Organizacije kreću u *outsourcing*, ne samo u proizvodnji softvera, radi: smanjenja troškova, usmjeravanja cjelokupnoga npora tvrtke ostvarenju misije, korištenja kvalitetnih potencijala kojih u samoj organizaciji nedostaje (ljudi, kapitala, tehnologije, resursa).

2.2.3 Uvođenje u primjenu

Gотов programski proizvod uvodi se u sustav, odnosno primjenjuje u svakodnevnome radu.

Izrađen, novi programski proizvod, treba zauzeti mjesto u poslovnom sustavu i zamjeniti postojeću „staru“ aplikaciju. Proces zamjene stare aplikacije novom aplikacijom nazivamo **uvođenje**. Moguće je da sustav do tada nije informatiziran te da

¹⁰ **Domena** je skup svih ulaza kojima se može izvoditi neka testna jedinica.

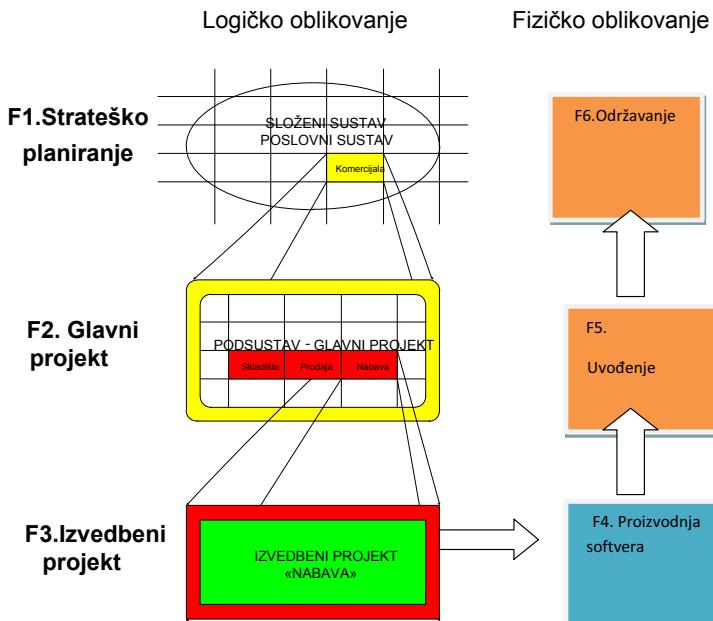
ne postoji „stara“ aplikacija, tada je uvođenje lakše. Kada se aplikacija uvede, počinje njezino dnevno operativno korištenje i tada se sustav nalazi u fazi primjene aplikacije.

Uvođenje je razdoblje od nekoliko dana do nekoliko mjeseci. Primjena programskoga proizvoda traje više godina. Uvođenje može biti i neuspješno, ako se pokažu nepremostivi problemi. Tada se planiraju korektivne aktivnosti vezane uz probleme: izmjena softvera, dorada baze podataka, dodatna obuka, i dr. Po uklanjanju uzroka odgađanja uvođenja ponovo se pristupa uvođenju. To se može ponoviti više puta. Moguće je i odustati od uvođenja.

Aktivnosti faze uvođenja su prikazane u tablici 2.7. u okviru Specijalizirane metodologije MIRIS.

2.3 Specijalizirana metodologija MIRIS

Specijalizirana metodologija MIRIS (skraćeno od hrvatskog Metodologija za Razvoj Informacijskog Sustava) je skup metoda i uputa čiji je ukupni cilj projektirati i izgraditi informacijski sustav. Poznata je i kao metodika MIRIS. Ta specijalizirana metodologija propisuje faze razvoja i aktivnosti pojedine faze do potrebne razine detalja informacijskih sustava.



Slika 2.7 Raščlana sustava i faze specijalizirane metodologije MIRIS

Faze životnoga ciklusa grupirane su u dvije skupine faza: logičko oblikovanje (projektiranje informacijskog sustava) i fizičko oblikovanje (izgradnja informacijskog sustava). Svaka skupina faza ima tri faze (vidi sl. 2.7). Faze se dalje dijele u aktivnosti.

Faze i aktivnosti životnoga ciklusa razvoja informacijskoga sustava prema metodologiji MIRIS prikazane su u sljedećim tablicama.

Tablica 2.3 Strateško planiranje

Faza 1: STRATEŠKO PLANIRANJE INFORMACIJSKOGA SUSTAVA (SP)
1.1 Analiza: Definiranje i obuka tima, dekompozicija procesa, popis dokumentacije i kretanje kroz sustav
1.2 Podsustavi: Određivanje podsustava i veza
1.3 Prioriteti: Određivanje prioriteta
1.4 Resursi: Definiranje cjelovite infrastrukture
1.5 Plan: Planiranje glavnih projekata i aktivnosti

Tablica 2.4 Glavni projekt

Faza 2: GLAVNI PROJEKT (GP)
2.1 PZ: Izrada projektnoga zadatka
2.2 DTP: Intervjuiranje, raščlanjivanje i modeliranje procesa (DTP)
2.3 Procesi GP: Analiza procesa, problema i prijedloga poboljšanja
2.4 Podaci GP: Opisivanje podataka
2.5 Plan GP: Planiranje izvedbenih projekata
2.6 Resursi GP: Definiranje modela resursa glavnoga projekta

Tablica 2.5 Izvedbeni projekt

Faza 3: IZVEDBENI PROJEKT (IP)
3.1 DEV: Intervjuiranje, apstrakcija i modeliranje podataka (EV)
3.2 Prevođenje: Prevodenje modela podataka u shemu BP (RM)
3.3 Arhitektura IP: Definiranje arhitekture programskoga proizvoda (APP)
3.4 Operacije IP: Projektiranje operacija nad shemom BP

Tablica 2.6 Proizvodnja softvera

Faza 4: PROIZVODNJA SOFTVERA (PS)
4.1: PLANIRANJE PROIZVODNJE
<ul style="list-style-type: none"> • Planiranje aktivnosti proizvodnje SW • Određivanje izvršitelja za pojedine zadatke i određivanje rokova • Određivanje i kreiranje produkcijске, testne i razvojne okoline
4.2: OBLIKOVANJE BAZA PODATAKA
<ul style="list-style-type: none"> • Prevođenje logičkoga modela podataka u fizički model sheme baze podataka • Kreiranje razvojne okoline za svakoga pojedinog programera • Punjenje sheme baze podataka u razvojnoj okolini iz postojeće produkcijске BP • Dodavanje novih koncepata iz modela podataka u razvojnu shemu BP (tip entiteta i dr.) • Kreacija razvojne baze podataka • Inicijalno punjenje testne baze podataka
4.3: RAZVOJ PROGRAMSKOGA PROIZVODA
<ul style="list-style-type: none"> • Izrada glavnog izbornika (aplikacijskoga stabla) ili dorada već postojećega • Izrada ekrana za pregled redaka svake tablice po jednom ili više ključeva • Izrada ekrana za operacije nad jednim retkom tablice (unos, izmjena, brisanje i pregled) • Izrada programskih modula različitih vrsta i namjena: obračuna, procedura, funkcija kontrola, <i>look-upova</i> nad tablicama (s prvim testiranjem modula) • Izrada izvještaja (s prvim testiranjem modula)
4.4: TESTIRANJE U TESTNOJ OKOLINI
<ul style="list-style-type: none"> • Prijenos razvijenih programskih modula u testno okruženje • Spajanje novih modula s postojećim • <i>Back-up</i> verzija softvera • Testiranje prototipa softvera nad testnom bazom podataka • Ažuriranje planova proizvodnje softvera • Izvođenje prema potrebi aktivnosti iz ranijih skupina aktivnosti i ponovno testiranje
4.5: TESTIRANJE I ISPRAVLJANJE U RADNOJ OKOLINI
<ul style="list-style-type: none"> • Prijenos razvijenih programskih modula u radno okruženje • Spajanje novih modula s postojećima • <i>Back-up</i> verzija softvera • Punjenje baze podataka • Testiranje prototipa softvera nad produksijskom bazom podataka koje obavlja programer • Ažuriranje planova proizvodnje softvera • Izvođenje prema potrebi aktivnosti iz ranijih skupina aktivnosti i ponovno testiranje
4.6: KORISNIČKO TESTIRANJE
<ul style="list-style-type: none"> • Prezentacija softvera korisniku • Testiranje od strane korisnika • Izrada popisa primjedbi korisnika • Ažuriranje planova proizvodnje softvera • Izvođenje prema potrebi aktivnosti iz ranijih skupina aktivnosti i ponovno testiranje • Izrada zapisnika o testiranju i prihvaćanju faze uvođenja

Tablica 2.7 Uvođenje

Faza 5: UVODENJE (UVO)
5.1 Instalacija gotovoga softvera na proizvodnjičko okruženje (kod korisnika)
5.2 Izrada uputa
5.3 Prezentacija gotovoga softvera
5.4 Obuka
5.5 Završne konverzije
5.6 Završno testiranje
5.7 Početak primjene nove aplikacije
5.8 Uspostava novoga sustava i potpisivanje primopredajnoga zapisnika

Tablica 2.8 Primjena i održavanje

Faza 6: PRIMJENA I ODRŽAVANJE (ODR)
6.1 Podešavanje novoga aplikacijskoga sustava
6.2 Izvješe o procjeni novoga projekta
6.3 Raspolaganje odgovornosti korisnika i programera
6.4 Korištenje aplikacijskoga sustava
6.5 Korisničko postavljanje zahtjeva za izmjenama

Cjelokupni posao započinje izradom strateškoga plana. Potom se u fazi Glavnoga projekta analizira poslovanje i kreira model procesa.

U fazi Izvedbenoga projekta modeliraju se podaci i definira logička arhitektura programskega proizvoda.

Faza „Proizvodnje softvera“ započinje planiranjem proizvodnje. Uzimaju se u obzir postojeći programski proizvodi i baza podataka u sustavu. Cilj je reorganizirati postojeću bazu podataka (stvoriti novu ako ništa ne postoji) i kreirati potreban novi programski proizvod, odnosno izmijenjen ili zamijeniti postojeći programski proizvod.

Gotov programski proizvod uvodi se u sustav, a po uvođenju primjenjuje se u svakodnevnome radu. Dijelovi informacijskoga sustava održavaju se po potrebi kako bi on zadovoljio potrebe korisnika.

3 RELACIJSKE BAZE PODATAKA

3.1 Uvod u relacijsku metodu

Relacijski model podataka objavljen je prvi put u Coddovu¹¹ članku (Codd, 1970.). Edgar Frank Codd (vidi sliku 3.1) rođen je 1923. u Engleskoj.

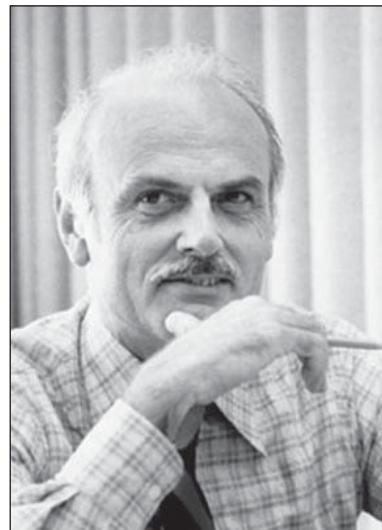
Nakon serije Coddovih radova započeo je razvoj relacijskih modela podataka i relacijskih baza podataka, a teorija modela podataka započinje bržim razvojem. Nakon razvoja teorije modela podataka uočeno je da relacijski model ima niz nedostataka. Na osnovi tih saznanja definiran je prošireni relacijski model.

U osnovi postoje dvije vrste relacijskih metoda, odnosno odgovarajućih modela: relacijski model (RM) i prošireni relacijski model (RM/T).

Cood je objavio prošireni relacijski model podataka (nazvan RM/T) nakon razvoja teorije modela podataka (Codd, 1979.) u okviru koje je razvijeno niz semantički bogatih koncepata. Ovi koncepti omogućuju na formalan način ugradnju interpretacije podataka u model.

Osnovni koncepti za gradnju strukture relacijskoga modela podataka su relacije. Relacija u relacijskome modelu podataka je isto što i relacija u matematici s tom razlikom da su relacije u relacijskome modelu podataka vremenski promjenljive.

U ovome se poglavlju predstavlja teorija relacijskoga modela, upitnih jezika i daje kratak pregled SQL-a kao standardnoga jezika upita. Radi



Slika 3.1 Edgar Frank Codd

¹¹ 1970. godine objavio je rad "A Relational Model of Data for Large Shared Data Banks" u kojem je predstavljena relacijska metoda.

kompletnosti prikazana je neophodna teorija za razumijevanje relacijskoga modela i relacijskih baza podataka i dana osnova upitnoga jezika SQL. Kao preduvjet validaciji modela podataka zahtjeva se umijeće oblikovanja upita nad bazom podataka korištenjem SQL-a, jer ne može se očekivati korisnost modela ukoliko se ne stvori jasna predodžba o pitanju budućoj bazi podataka. Ovo znači da bi čitatelj mogao, prilikom validacije strukture modela podataka, odgovoriti može li se danim modelom odgovoriti na raspoloživa razborita korisnička pitanja, poznata i kao tipični zahtjevi korisnika, tj. može li se u principu formulirati SQL upiti (nad danom strukturom buduće baze podataka) koji bi dali tražene odgovore. Na ovo je i u procesu oblikovanja baze podataka važno odgovoriti rano, tj. na osnovi modela podataka kako se ne bi gubilo vrijeme razvijajući neodgovarajuće baze podataka. SQL je predstavljen sažeto i s osnovnom sintaksom kako bi se prenijele ideje i mogućnosti upitnoga jezika što je zadovoljavajuća razina detalja u skladu s ciljem koji ovo poglavlje sadrži u okviru cjeline.

3.2 Struktura relacijskoga modela

3.2.1 Koncepti strukture relacijskoga modela

Coddova definicija relacije je: neka su dani skupovi D_1, D_2, \dots, D_n (ne obavezno različiti), R je **relacija** (engl. *Relation*) nad ovih n skupova ($n > 0$) ako je to skup n -torki takav da za svaku n -torku vrijedi da je prvi element n -torke iz D_1 , drugi iz D_2, \dots, n -ti iz D_n .

Skup D_i naziva se **domena relacije** R . Domena je skup sličnoga tipa podataka, na primjer, skup svih prezimena radnika sektora Proizvodnja.

Matematički stroža definicija relacije je: Neka je D_1, D_2, \dots, D_n niz od n skupova ne obavezno različitih. **Relacija R** je podskup Kartezijevoga produkta ovih skupova.

$$R \subseteq D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) : d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n\}$$

Relacija R je skup n -torki (d_1, d_2, \dots, d_n) . Redoslijed domena u Kartezijevome produktu važan je. Imenujući funkciju koja i -tom članu n -torke pridružuje danu vrijednost d_i imenom A_i (poznato kao imenujući domene) redoslijed postaje nebitan, n -torka (d_1, d_2, \dots, d_n) relacije R zamjenjuje se n -torkom parova. Funkcija A_i se naziva **atribut relacije**.

Broj atributa n je brojnost atributa relacije ili **stupanj relacije**.

Broj n -torki je brojnost pojavljivanja relacije ili **kardinalni broj relacije**.

Dva ili više atributa relacije mogu biti definirani nad istom domenom.

Uobičajeni zapis relacije preko atributa je:

Naziv relacije (Naziv atributa₁, N.atr₂, N.atr₃, ..., N.atr_n)

Na primjer, relacija Radnik zapisana preko atributa je

Radnik(#Radnika, Prezime, Ime, Starost),

a sastoji se od triju n -torki :

<#Radnika: 234, Prezime: Luketić, Ime: Marko, Starost: 25>

<#Radnika: 258, Prezime: Palijan, Ime: Ivan , Starost: 34>

<#Radnika: 108, Prezime: Čorak, Ime: Pero , Starost: 56>

S motrišta korisnika vrijedi definicija relacije:

- relacija je imenovana tablica,
- atribut je stupac tablice,
- n -torka je redak tablice.

Relacija Osoba zapisana u obliku tablice prikazana je u tablici 3.1.

Tablica 3.1 Relacija Osoba predstavljena tablicom

Šifra	Prezime	Ime osobe	Godina rođenja
234	Ostrun	Ivan	1860
119	Cule	Foška	1870
258	Brajdić	Daniel	1860
132	Ćiković	Ida	1890
108	Sabljak	Marija	1850
109	Sabljak	Stipan	1927
508	Sabljak	Marija	2010
178	Šajna	Josipa	1907
111	Brajdić	Kata	1891
003	Hodak	Stjepan	NULL
004	Hodak	Marija	NULL
145	Bacchia	Marino	1921
131	Nadišić	Silvana	1929
133	Nadišić	Mira	1932

Imenovani stupci tablice su atributi. Tablica predstavlja popis osoba. Reci tablice (n -torke relacije) su pojedinačna pojavljivanja osobe.

Na primjer, u tablici Radnik svaki redak predstavlja zasebnoga radnika.

Relacijski model podataka sadrži podatke i njihovu interpretaciju preko kategorija:

- strukture modela,
- ograničenja u modelu,
- skupa operatora u modelu.

Struktura relacijskoga modela definira se shemom relacijskoga modela podataka. Elementi strukture relacijske sheme su relacije.

Na primjer, relacijska shema baze podataka o nacrtima je:

Nacrt(#Nacrt, Naziv-nacrta, #Vrsta-nacrt, Komentar, #Proizvod, #Radnik)
 Vrsta-nacrt (#Vrsta-nacrt, Naziv-vrste-nacrt)
 Proizvod (#Proizvod, Naziv-proizvoda, Visina, Duljina, Širina, Površina-bruto, Težina, Nomenklturni-broj)
 Radnik(#Radnik, Ime, Prezime, Datum-zaposlenja)
 Predan-nacrt(#Radnik, #Nacrt, Datum predavanja)

Relacijska baza podataka (engl. relational database) je skup u vremenu promjenljivih relacija opisanih u shemi baze podataka.

Veza između relacija se ostvaruje isključivo preko vrijednosti atributa (bez ikavkih pokazivača). S motrišta izvodivosti relacije iz ostalih relacija u relacijskome modelu razlikuju se sljedeće vrste relacija:

- **Bazna relacija** (realna, stvarna) (engl. *fundamental relation*) je relacija koja se definira neovisno o drugim relacijama u modelu
- **Izvedena relacija** (virtualna, pogled) (engl. *derived relation*) je relacija koja se može potpuno izvesti iz drugih relacija
- **Poluizvedena relacija** (engl. *semi-derived relation*) je relacija čija je bar jedna projekcija (s barem jednim atributom) izvedena relacija.

3.2.2 Primjer relacijske baze podataka

Za prikaz relacijskoga modela podataka i relacijske baze definirat ćemo dio kadrovskog informacijskog sustava, nazovimo ga Kadrovi.

Struktura sheme relacijske baze podataka, njezinih atributa i ključeva, prikazana je sljedećom tablicom.

Tablica 3.2 Shema relacijske baze podataka

Tablica	Stupac	Opis	Ključevi, Ograničenja
Radnik	Mbr	Char(4)	PK, Unique, Not Null
	Prezime	Char(20)	
	Mje	Char(15)	
	Adr	Char(20)	
	Drod	Date	Godina, mjesec, dan

	Soj	Char(3)	SK
	Srm	Char(2)	SK
	Sss	Char(3)	SK
Orgjed	Soj	Char(3)	PK, Unique, Not Null
	Noj	Char(20)	
Radnomj	Srm	Char(2)	PK, Unique, Not Null
	Nrm	Char(15)	
	Sss	Char(3)	SK
	Bod	Number(3)	
Strspr	Sss	Char(3)	PK, Unique, Not Null
	Nss	Char(15)	
Radiojena	Mbr	Char(4)	PK, Unique, Not Null
	Srm	Char(2)	PK, Unique, Not Null
	Dod	Date	Godina, mjesec, dan
	Ddo	Date	Godina, mjesec, dan
	Bod	Number(3)	
	Sss	Char(3)	SK

Legenda:

- PK - primarni (glavni) ključ
- SK - Strani ili vanjski ključ
- Not Null - ne nul vrijednost (nul vrijednost je nepoznata vrijednost atributa)
- Unique - jedinstvenost vrijednosti

Prema tome, shema relacije baze podataka:

- Radnik (**Mbr**, Prezime, Mje, Adr, Drod, **Soj**, **Srm**, **Sss**)
- Orgjed (**Soj**, Noj)
- Radnomj (**Srm**, Nrm, Bod, **Sss**)
- Strspr (**Sss**, Nss)
- Radiojena (**Mbr**, **Srm**, Dod, Ddo, Bod, **Sss**)

U relaciji Radnik primarni (glavni) ključ je Mbr. Primarni je ključ u relacijskoj metodi podvučen punom crtom, što je oznaka PK. Ova relacija ima tri strana ključa: Soj, Srm, Sss koji redom predstavljaju primarne ključeve relacija Orgjed, Radnomj, Strspr.

Relacija Radnomj ima strani ključ Sss koji je primarni (glavni) ključ relacije Strspr.

Sama baza podataka Kadrovi definirane strukture sadrži prikazano u sljedećim četirima tablicama.

Tablica 3.3 Radnik

Mbr	Prezime	Mje	Adr	Drod	Soj	Srm	Sss
1001	Ivan Perić	Zagreb	Pulska10	1948-11-26	001	P1	101
1002	Mario Vukić	Split	Zadars.3	1952-03-19	002	P2	102
1003	Zlatko Puh	Split	Bana J.7	1959-09-13	002	P3	100
1004	Gordan Marić	Zagreb	Illica 7	1968-11-10	001	P1	101
1005	Ana Žurić	Rijeka	M. Grbac 3	1971-03-06	003	P4	102
1006	Branka Horvat	Split	Velbits.9	1962-04-17	002	P5	102
1007	Josip Ban	Rijeka	Korzo 8	1964-12-25	003	P3	100

Tablica 3.4 Organizacijska jedinica

Soj	Noj
001	Proizvodnja
002	Priprema
003	Opći poslovi

Tablica 3.5 Radno mjesto

Srm	Nrm	Sss	Bod
P1	Zavarivač	101	205
P2	Planer	102	400
P3	Operator	100	240
P4	Pravnik	102	315
P5	Referent nabave	102	355

Tablica 3.6 Stručna spremna

Sss	Nss
100	Kvalificirani radnik
101	Visoko kvalificirana
102	Visoka stručna spremna

Tablica 3.7 Radio je na

Mbr	Srm	Dod	Ddo	Bod	Sss
1005	P1	1990-1-1	2000-1-1	205	101
1005	P3	2000-1-1	2001-2-1	290	100
1005	P4	2001-2-1	2005-1-1	410	101
1007	P1	2006-4-10	2007-6-10	205	101

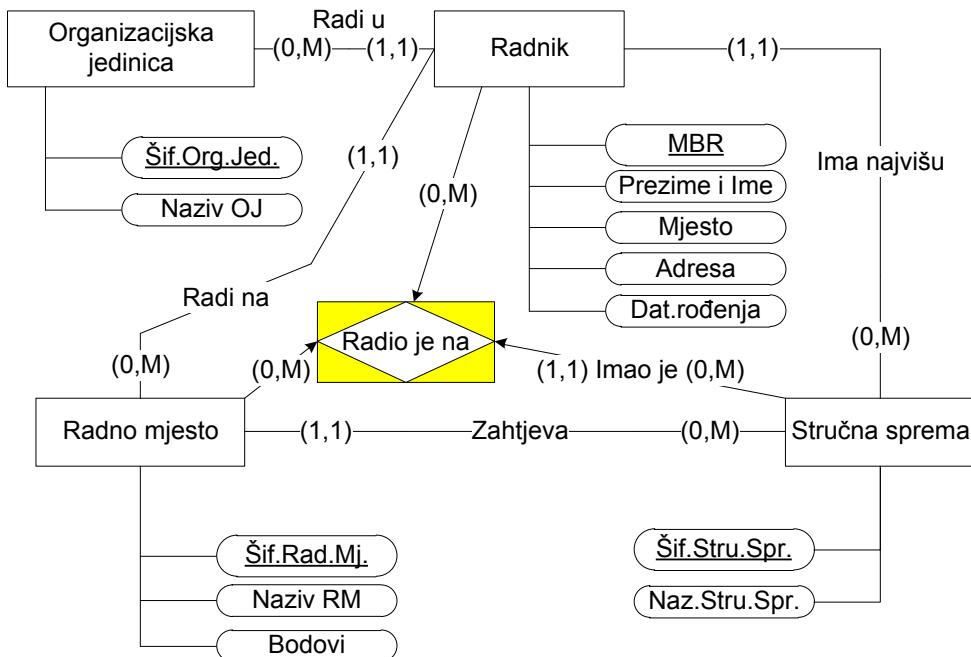
Svakoj od tablica pripadaju određeni atributi koji ga detaljno opisuju. U našemu primjeru definirali smo attribute (vidi tablicu 3.8.).

Tablica 3.8 Atributi tipova entiteta

Tip entiteta	Atribut	Kratika
Radnik	Matični broj radnika	(Mbr)
	Prezime i ime radnika	(Prezime)
	Mjesto stanovanja	(Mje)
	Adresa stanovanja	(Adr)
	Datum rođenja radnika	(Drod)
Organizacijska jedinica	Šifra org. Jedinice	(Soj)
	Naziv org. Jedinice	(Noj)
Radno mjesto	Šifra radnoga mjesta	(Srm)
	Naziv radnoga mjesta	(Nrm)
	Bodovi radnoga mjesta	(Bod)
Stručna spremna	Šifra stručne spreme	(Sss)
	Naziv stručne spreme	(Ns)
Radio je na	Datum – od	(Dod)
	Datum – do	(Ddo)
	Bodovi radnoga mjesta	(Bod)

Radi cjelovitoga prikaza ovdje će biti priložen dijagram entiteti – veze (DEV) nastao kao rezultat modeliranja metodom EV-a, a njegov potpun opis bit će dan u kasnijim poglavljima. Stoga se čitatelju predlaže da u prvome čitanju zanemari ovaj dio teksta.

DEV na slici 3.2. nastao je na temelju već prikazanoga relacijskog modela.



Slika 3.2 DEV sustava "KADROVI"

Opišemo li ovaj dijagram uobičajenim riječima, slijedi značenje:

Svaki radnik radi u jednoj i samo jednoj organizacijskoj jedinici, radi na jednom samo jednome radnom mjestu i ima jednu i samo jednu stručnu spremu.

U jednoj organizacijskoj jedinici radi nitko, jedan ili više radnika.

Na jednome radnom mjestu radi nijedan, jedan ili više radnika i to radno mjesto zahtijeva samo jednu stručnu spremu.

Jednu stručnu spremu može imati nijedan, jedan ili više radnika (pa čak i nijedan), i ta stručna sprema može biti stručna spremu nijednog, jednog ili više radnih mjeseta.

3.3 Ograničenja relacijskoga modela

Relacija u relacijskome modelu podataka je skup pa svojstva skupa vrijede i za relaciju. Ta svojstva koja se odnose na ograničenja su:

Svojstvo 1. Svi elementi skupa su različiti

Svojstvo 2. Poredak elemenata skupa je proizvoljan

Svojstvo 3. Relacija je podskup Kartezijskog produkta skupova D_1, D_2, \dots, D_n

3.3.1 Svojstvo 1. - svi elementi skupa različiti su

Ako je dan skup $S = \{a_1, a_2, \dots, a_n\}$, tada za svaki $a_i \in S$ vrijedi da ne postoji element $a_j \in S$ za koji je $a_i = a_j$, odnosno:

$$S = \{a_1, a_2, \dots, a_n\} \wedge (\forall a_i \in S) \wedge (\forall a_j \in S) \wedge i \neq j \Rightarrow a_i \neq a_j$$

pri čemu $i = 1, 2, \dots, n$ i $j = 1, 2, \dots, n$.

U relaciji to znači da bilo koje dvije n -torke nisu jednake, odnosno da bilo koja dva retka tablice nisu jednaka.

Iz ovog ograničenja proizlazi definicija ključa relacije.

Ključ (engl. key) relacije u relacijskome modelu podataka je onaj podskup atributa čije vrijednosti jedinstveno identificiraju n -torke relacije.

Moguće je da relacija ima više različitih ključeva. Svi ključevi relacije čine skup **kandidata za ključ**.

Ako neki skup atributa u promatranoj relaciji nije ključ, ali je ključ u nekoj drugoj relaciji u modelu, onda se on naziva **vanjski ključ (engl. foreign key)**.

Da bi neki skup atributa relacije bio kandidat za ključ, on mora zadovoljiti, neovisno o vremenu, dva uvjeta:

1. Uvjet jedinstvenosti

Vrijednost ključa svake n -torke relacije jedinstveno određuje n -torku, to jest, ne postoje dva takva retka u tablici da imaju sve iste vrijednosti svih atributa koji čine ključ.

2. Uvjet neredundantnosti

Ne postoji niti jedan atribut kao dio ključa koji se može izostaviti iz ključa, a da se pritom uvjet jedinstvenosti ne gubi, to jest, ključ je unija minimalnoga broja atributa.

Od niza kandidata za ključ bira se jedan koga se naziva **primarni (glavni) ključ** (*engl. primary key*). Definirajući u strukturi relacijske baze podataka primarni ključ relacije, relacija posjeduje svojstvo jedinstvenosti, samo sa stajališta statičkoga stanja relacije.

Ključ relacije može se definirati preko vrste veze između atributa relacije.

Atribut (skup atributa) relacije je ključ ako je jednoj vrijednosti atributa u relaciji pridružena samo jedna vrijednost ostalih atributa relacije, odnosno ako su svi ne-ključni atributi funkcionalni ovisni o ključu.

Proglaši li se niz atributa za ključ relacije, tada je inherentno nametnut tip ograničenja između ključnih i ostalih atributa.

3.3.1.1 Integritet ključnih atributa

Sa stajališta dinamike, zbog pravilnog izvršenja operacija nad relacijama, tijekom vremena definirana su dodatna ograničenja (Codd, 1979.) na primarni (glavni) ključ. Ta ograničenja omogućuju bespriječoran prelazak iz jednoga stanja baze podataka u drugo, a nazivaju se pravila integriteta baze podataka, a ona su:

Pravilo I1. Cjelovitost ključa (integritet entiteta, provjera ključa, postojanje entiteta)

Neka je atribut A dio primarnoga ključa relacije R . Tada atribut A ne smije poprimiti nul vrijednost. To znači da ne postoji n -torka u relaciji R takva da je vrijednost atributa A te n -torke jednaka nul vrijednosti.

Ključ relacije je cjelovit ako svi atributi koji ulaze u ključ poprimaju za svaku n -torku relacije realnu vrijednost iz domene, različitu od nul vrijednosti. Nikada ključ relacije ne smije poprimiti nul vrijednost.

Pravilo I2. Cjelovitost ovisnosti (referencijski integritet, provjera zavisnosti)

Neka postoji domena D i relacija S s prostim primarnim ključem A definiranim nad D .

Neka postoji relacija R s atributom A definiranim nad domenom D . Tada vrijednost atributa A bilo koje n -torke u relaciji R može biti ili nul vrijednost ili vrijednost " k ", pri čemu vrijednost " k " postoji u domeni D i postoji n -torka u relaciji S takva da je vrijednost ključnog atributa A te relacije jednaka vrijednosti " k ". R i S relacije ne moraju biti različite. Atribut A relacije R , koji postoji kao primarni (glavni) ključ u relaciji S , naziva se vanjski (strani) ključ.

Ako u relaciji postoji atribut koji je strani ključ i ako taj atribut ima neku vrijednost n -torke različitu od nul vrijednosti, onda mora postojati n -torka, u relaciji u kojoj je strani ključ primarni ključ, takva da je njezina vrijednost jednaka vrijednosti stranoga ključa.

Ako je jedan redak tablice A povezan s jednim retkom tablice B , onda mora postojati redak u tablici B . Ako se izbriše redak u tablici B koji je povezan s retkom u tablici A , onda se mora izbrisati vrijednost vanjskoga ključa u tablici A onog atributa koji povezuje dvije tablice. Ako je taj atribut u tablici A dio primarnoga ključa, onda se ne briše samo njegova vrijednost već i cijeli redak tablice A , jer to zahtijeva cjelovitost ključa (prvi integritet).

3.3.2 Svojstvo 2. - poredak elemenata skupa nebitan je

Poredak n-torki relacije (redaka tablice) može biti proizvoljan u bazi podataka.

Ovo svojstvo ne zahtijeva specifikaciju ograničenja.

Uočimo da poredak redaka u tablici Osoba nije sortiran po atributima šifra, prezime, ime, godina rođenja.

Relacijski model podataka sastoji se od maloga broja inherentnih ograničenja. To daje slobodu projektantu modela podataka u vrijeme modeliranja koncepcata tipova entiteta i tipova veza u relacije. Međutim to stvara poteškoće korisnicima baze podataka u vrijeme korištenja znanja o vezama između relacija.

3.3.3 Svojstvo 3. - relacija je podskup Kartezijsevoga produkta

Svojstvo 3. (relacija je podskup Kartezijsevoga produkta skupova D_1, D_2, \dots, D_n) znači u ovisnosti od skupova moguće je mnoštvo podskupova Kartezijsevoga produkta. Od interesa je postaviti ograničenja koja će izdvojiti semantički moguće podskupove.

3.3.4 Eksplicitna ograničenja

Kako bi se u model podataka unijelo što više znanja o poslovnom sustavu, uvode se dodatna eksplicitna ograničenja na relacije. Ova ograničenja postavljaju logičke granice na pojedine podatke i njihove međusobne veze preko posebno uvedenih struktura. Ta ograničenja mogu biti:

- a) kako bi dali bolju interpretaciju domena u modelu podataka
 - na vrijednost pojedinog atributa
 - na agregiranu vrijednost atributa
 - usporedbom više atributa
- b) kako bi provjeravali potrebna statička svojstva baze podataka
 - izračunavaju agregirane vrijednosti atributa
 - koriste interne funkcije pri računanju
 - uvjetuju akcije prije korištenja operatora

- c) zbog očuvanja integriteta baze podataka
- kakva mora biti nova vrijednost u odnosu na vrijednost prije operacije (prije izvršenja transakcije)
 - egzistencijalna ovisnost među relacijama (postojanje retka u jednoj tablici ovisi o postojanju retka u nekoj drugoj tablici)

3.4 Operatori relacijskoga modela

Operatori nad relacijama mogu biti navigacijski ili specifikacijski.

Navigacijski operatori (engl. *navigational operators*) zahtijevaju dodatnu definiciju pojmove: tekući redak tablice i poredak pretrage, a koriste se unutar viših programskih jezika. Korisnicima je složeno upotrebljavati ove operatore. Oni se mogu koristiti za implementaciju specifikacijskih operatora.

Relacija u relacijskome modelu podataka je skup, pa operacije u teoriji skupova vrijede i za relacije. Na taj način definirana je relacijska algebra. **Relacijska algebra** (engl. *relational algebra*) je jezik koji omogućava definiranje novih relacija iz postojećih. Nova relacija sadrži podatke koji su bili u bazi u trenutku izvršenja operacije.

Relacijama se može operirati preko pogleda. **Pogled** (engl. *View*) je nova relacija dobivena iz postojećih proširenjem sheme relacijske baze podataka za novu relaciju. U pogledu se nalaze podaci trenutno smješteni u bazi. Operacije nad pogledima imaju matematički jednake definicije kao operacije u relacijskoj algebri, iako operiraju tipovima podataka različitim od tipova podataka u relacijskoj algebri.

Pored relacijske algebre razvijeni su i drugi jezici za rukovanje relacijama, a u ovoj knjizi će biti prikazan i SQL (jezik s upitnim blokom).

3.4.1 Relacijska algebra

Relacijska algebra je proceduralni jezik (piše se procedura) u kome se pomoću operatora relacijske algebre definira skup operacija koje se izvode nad relacijama relacijskoga modela i generira željena relacija.

Osnovne operacije ažuriranja trebaju omogućiti:

Upis	n -torki u relaciju
Brisanje	n -torki iz relacije
Ažuriranje	n -torki u relaciji

Operatori relacijske algebre pogodni za ažuriranje baze podataka su:

Unija i
Razlika

a operatori pogodni za izvješćivanje iz baze podataka su:

- Projekcija
- Selekcija (ili restrikcija)
- Kontrolirano spajanje (join)
- Nekontrolirano spajanje
- Presjek
- Dijeljenje

Operandi u relacijskoj algebri su relacije iz relacijskoga modela. Ne postoji ograničenje na izbor relacija nad kojima se obavljaju operacije. Zbog ove slobode mogu uslijediti poteškoće.

Pri operatoru za ažuriranje eksplisitno se zahtijeva da su sve relacije koje ulaze u operaciju definirane nad istim domenama.

Ovo pravilo treba još čvršće postaviti i zahtijevati da su odgovarajući atributi relacija iste semantike. Na primjer, domena "skup svih mogućih datuma" u različitim relacijama može imati različit smisao.

Zajedničko svojstvo svih navedenih operacija je svojstvo zatvorenosti, tj. sve operacije djeluju na relacijama i za rezultat opet daju relaciju.

Operacije unije, presjeka i razlike su binarne operacije i mogu se izvoditi samo nad unijski kompatibilnim (usporedivim) relacijama. Dvije relacije su unijski kompatibilne ako imaju isti stupanj i ako odgovarajući atributi obje relacije imaju iste domene.

Tablice početnih relacija su: Djetalnik_1, Djetalnik_2 i Projekt (vidi tablice 3.9, 3.10 i 3.11).

Tablica 3.9 Djetalnik_1

# Djetalnika	Ime i prezime
122	Ivo Ivić
222	Ana Anić
322	Marko Zec

Tablica 3.10 Djetalnik_2

# Djetalnika	Ime i prezime
123	Marina Rak
322	Marko Zec
425	Nada Rožić

Tablica 3.11 Projekt

# Projekta	Datum početka
101	05.05.1995
201	10.06.1994.

Unija (engl. *union*) je operacija nad dvjema relacijama (binarna operacija) koja omogućava dodavanje različitih n -torki jedne relacije drugoj te omogućava upis u relaciju. Rezultat unije je relacija koja sadrži sve n -torke koje pripadaju ili prvoj ili drugoj relaciji.

Oznaka operatora je “ \cup “.

Unija je komutativna ($R_1 \cup R_2 = R_2 \cup R_1$) i asocijativna $R_1 \cup (R_2 \cup R_3) = (R_1 \cup R_2) R_3$ operacija.

Tablica 3.12 Tablica unije relacija: Djelatnik_1 Djelatnik_2

# Djelatnika	Ime i prezime
122	Ivo Ivić
222	Ana Anić
322	Marko Zec
123	Marina Rak
425	Nada Rožić

Razlika (engl. *difference*) je operacija nad dvjema relacijama koja omogućava izbacivanje istih n -torki u obje relacije. Operacija nije komutativna te je poredak operandâ bitan. Rezultat su n -torke iz prve relacije koje ne postoje u drugoj relaciji. Omogućuje brisanje n -torki iz relacije.

Oznaka operatora je “-”.

Tablica 3.13 Tablica razlike relacija: Djelatnik_1 - Djelatnik_2

# Djelatnika	Ime i prezime
122	Ivo Ivić
222	Ana Anić

Presjek (engl. *intersection*) je operacija nad dvjema relacijama koja za rezultat daje relaciju koja sadrži samo one n -torke koje se istovremeno nalaze u dvjema početnim relacijama. Presjek je komutativna ($R_1 \cap R_2 = R_2 \cap R_1$) i asocijativna ($R_1 \cap R_2 \cap R_3 = R_1 \cap R_2 \cap R_3$) operacija.

Oznaka operatora je “ \cap “.

Tablica 3.14 Tablica presjek relacija: Djelatnik_1 \cap Djelatnik_2

# Djelatnika	Ime i prezime
322	Marko Zec

Kartezijev produkt (nekontrolirano spajanje) (engl. *cartesian product*) je operacija izvješčivanja iz dviju relacija.

Osnovni operator za izvješčivanje iz više tablica je **kartezijev produkt**. On se još naziva i: descartesov produkt, prošireni kartezijev produkt, nekontrolirano spajanje, kompleksni produkt.

Rezultat je skup n -torki, koje su nastale spajanjem svake n -torke prve relacije sa svakom n -torkom druge relacije. Brojnost dobivene relacije jednak je produktu brojnosti početnih relacija.

Oznaka operatora je "x".

Tablica 3.15 Tablica kartezijev produkt relacija: Djelatnik_1 x Projekt

# Djelatnika	Ime i prezime	# Projekta	Datum početka
122	Ivo Ivić	101	05.05.1995
122	Ivo Ivić	201	10.06.1994.
222	Ana Anić	101	05.05.1995
222	Ana Anić	201	10.06.1994.
322	Marko Zec	101	05.05.1995
322	Marko Zec	201	10.06.1994.

Da bi skup operatora ažuriranja bio potpun, potreban je operator za mijenjanje sadržaja n -torki relacije. Za ovo nije uveden poseban operator, već se to izvodi uza-stopnom primjenom operatora razlike (pri čemu se uklanja n -torka čiji sadržaj želi-mo izmijeniti) pa zatim operatora unije (pri čemu se dodaje n -torka sa željenim sadržajem).

Osnovne operacije izvješčivanja trebaju omogućiti prikaz sadržaja dijela relacijskoga modela (jedne ili više relacija) uz mogućnost restrikcije samo nekih atributa ili samo nekih n -torki.

Osnovne dvije operacije za izvješčivanje iz jedne relacije su **projekcija** (engl. *projection*) i **selekcija** (engl. *selection*).

Projekcija (engl. *projection*) je operacija nad jednom relacijom koja omogućava izdvajanje skupa atributa jedne relacije u novu relaciju i pritom eliminaciju duplih n -torki ukoliko ove postoje. Rezultat projekcije ne mora biti relacija (može sadrža-vati više istih stupaca).

Oznaka operatora je “ Π “.

Rezultat na primjer projekcije relacije $R(A,B,C,D)$ po atributima B i C označava se: $\Pi_{BC}(R)$.

Selekcija (engl. *selection*) je operacija nad jednom relacijom koja omogućava izdvajanje skupa n -torki jedne relacije u novu relaciju. Izdvajaju se samo one n -torce koje zadovoljavaju zadani uvjet uz operator selekcije.

Uvjet selekcije može biti jednostavan ($=, \neq, <, \leq, >, \geq$) ili složen (jednostavnii uvjeti povezani logičkim operatorima i, ili, ne, isključivo ili).

Oznaka operatora je “ σ “.

Rezultat selekcije relacije $R(A,B,C)$, uz uvjet $C > 25$, označava se: $\sigma_{C > 25}(R)$.

Primjer projekcije i selekcije nad relacijom Djelatnik.

Tablica 3.16 Početna relacija Djelatnik

# Djelatnika	Ime i prezime	Odjel	Starost
122	Ivo Ivić	A	25
222	Ana Anić	B	34
322	Marko Zec	C	56
123	Marina Rak	A	22
425	Nada Rožić	B	28

Tablica 3.17 Projekcije relacija: $\Pi_{\#Djelatnika, Odjel}(Djelatnik)$ i $\Pi_{Odjel}(Djelatnik)$

# Djelatnika	Odjel	Odjel
122	A	A
222	B	B
322	C	C
123	A	A
425	B	B

Tablica 3.18 Selekcije relacije: $\sigma_{Starost > 30}(Djelatnik)$

# Djelatnika	Ime i prezime	Odjel	Starost
222	Ana Anić	B	34
322	Marko Zec	C	56

Tablica 3.19 Selekcije relacije: $\sigma_{Starost > 30 \ \& \ Odjel = B}(Djelatnik)$

# Djelatnika	Ime i prezime	Odjel	Starost
222	Ana Anić	B	34

Spajanje (kontrolirano spajanje) (engl. *join*) je operacija za izvješćivanje iz dviju ili više relacija. Operacija se sastoji od dva ili tri koraka. Rezultat prvoga koraka je kartezijev produkt relacija, koji može, ali ne mora biti relacija. U drugome koraku izdvaja se iz kartezijevog produkta skup n -torki koje zadovoljavaju zadani uvjet. Uvjet se upisuje u uglate zgrade, između naziva relacija koje sudjeluju u spajanju. To je slučaj selekcije iz kartezijevoga produkta. Uvjet selekcije može biti jednostavan operator ili složen. Ako je odabrani operator “=” govorimo o **spajanju s izjednačavanjem** (engl. *equi join*). Pri takvoj vrsti spajanja dobijemo tablicu koja u dva ili više stupaca sadrži vrijednost istog atributa. Takva tablica nije relacija, pa moramo ukloniti “kopije” stupaca. Spajanje s izjednačavanjem, uz isključivanje redundantnih stupaca, zovemo **prirodnim spajanjem** (engl. *natural join*). Pri prirodnom spajanju uvjet se upisuje između dvije uspravne crte. Primjeri “običnoga” spajanja i prirodnoga spajanja dani su u primjeru A i B.

A. Primjer “običnog” spajanja:

Tablica 3.20 Početne relacija: Djelatnik_1

# Djelatnika	Ime i prezime	Odjel
122	Ivo Ivić	A
222	Ana Anić	B
322	Marko Zec	A

Tablica 3.21 Početne relacija: Projekt

# Projekta	Datum početka	Mjesto
101	05.05.1995	A
201	10.06.1994.	D

Tablica 3.22 Kartezijev produkt relacija: Djelatnik_1 x Projekt

# Djelatnika	Ime i prezime	Odjel	# Projekta	Datum početka	Mjesto
122	Ivo Ivić	A	101	05.05.1995	A
122	Ivo Ivić	A	201	10.06.1994.	D
222	Ana Anić	B	101	05.05.1995	A
222	Ana Anić	B	201	10.06.1994.	D
322	Marko Zec	A	101	05.05.1995	A
322	Marko Zec	A	201	10.06.1994.	D

Tablica 3.23 Rezultat spajanja relacija: Djelatnik_1[Odjel^1Mjesto]Projekt

# Djelatnika	Ime i prezime	Odjel	# Projekta	Datum početka	Mjesto
122	Ivo Ivić	A	201	10.06.1994.	D
222	Ana Anić	B	101	05.05.1995	A
222	Ana Anić	B	201	10.06.1994.	D
322	Marko Zec	A	201	10.06.1994.	D

B. Primjer prirodnoga spajanja:

Tablica 3.24 Početna relacija: Djelatnik_94

# Djelatnika	Ime i prezime	Odjel
122	Ivo Ivić	A
222	Ana Anić	B
322	Marko Zec	C
123	Marina Rak	A

Tablica 3.25 Početna relacija: Djelatnik_95

# Djelatnika	Ime i prezime	Odjel
122	Ivo Ivić	A
222	Ana Anić	A
322	Marko Zec	C
123	Marina Rak	B
425	Nada Rožić	B

Rezultat spajanja s izjednačavanjem (engl. *equi join*):

Djelatnik_94[#Djelatnika_94=#Djelatnika_95&Odjel_94=Odjel_95]Djelatnik_95

Tablica 3.26 Rezultat spajanja s izjednačavanjem (engl. *equi join*)

# Djelatnika_94	Ime i prezime	Odjel_94	# Djelatnika_95	Ime i prezime	Odjel_95
122	Ivo Ivić	A	122	Ivo Ivić	A
322	Marko Zec	C	322	Marko Zec	C

Rezultat prirodnoga spajanja:

Djelatnik_94#Djelatnika_94=#Djelatnika_95&Odjel_94=Odjel_95|Djelatnik_95

Tablica 3.27 Rezultat prirodnoga spajanja

# Djelatnika	Ime i prezime	Odjel
122	Ivo Ivić	A
322	Marko Zec	C

Dijeljenje (engl. *division*) je operacija za izvješćivanje iz relacija.

Oznaka operatora je “/”.

Rezultat dijeljenja relacije $R_1 = \{A, B\}$ s relacijom R_2 je relacija $R_3(A)$, sa svojstvom: R_3 je najveći podskup od $\tilde{O}_A(R_1)$, za koji vrijedi $R_3|A|R_2$ je sadržan u R_1 .

Bolje i jasnije objašnjenje dijeljenja relacija vidi se iz sljedećeg primjera.

Početne relacije: Projektant_Na_Projektu i Stari_Projekt

Tablica 3.28 Projektant_Na_Projektu

# Djelatnika	Ime i prezime	# Projekta	Datum početka
222	Ana Anić	101	05.05.1995
222	Ana Anić	201	10.06.1994.
222	Ana Anić	401	20.07.1995.
322	Marko Zec	101	05.05.1995
425	Nada Rožić	101	05.05.1995
425	Nada Rožić	201	10.06.1994.

Tablica 3.29 Stari_Projekt

# Projekta	Datum početka
101	05.05.1995
201	10.06.1994.

Tablica 3.30 Rezultat dijeljenja: Projektant_Staroga_Projekta

# Djelatnika	Ime i prezime
222	Ana Anić
425	Nada Rožić

3.5 SQL - standardni upitni jezik za relacijske baze podataka

3.5.1 Osnove SQL-a

SQL (*engl. Structured Query Language, skraćeno SQL*) je jezik četvrte generacije koji služi za operiranje nad relacijskim bazama podataka. On je standardizirani skup instrukcija koje omogućuju definiranje relacijske baze podataka, pristup i manipulaciju podacima.

Četiri godine nakon objave relacijskoga modela podataka objavljen je rad pod nazivom "SEQUEL" (*A Structured English Query Language*) D. D. Chamberlina i R. F. Boycea koji su bili zaposleni u IBM-u.

Oni su definirali jezik, preteču SQL-a, koji je zadovoljavao zahtjeve Coddove relacijske algebре.

Dvije godine kasnije Chambelin i suradnici razvili su novu verziju jezika, SEQUEL/2 i ubrzo nakon toga je u IBM-u izrađen prototip nazvan Sistem R u koji je ovaj jezik implementirala Oracle kompanija.

Tvrtka „SDL“ (sada Oracle korporacija) razvija Oracle bazu podataka i od 1979. isporučuju prvu komercijalnu realizaciju SQL-a kupcima.

Oko 1980. godine uveden je novi naziv SQL.

Danas je SQL standardiziran. Standardizirala su ga organizacije ANSI (*American National Standards Institute*) i ISO (*The International Organization for Standardization*) i postoji više generacija SQL standarda uključivo SQL 2008¹².

Standard donosi sintaksu i semantiku dvaju podjezika, podjezika za definiranje baze podataka (SQL-DDL) i podjezika za manipulaciju podacima u bazi podataka (SQL-DML).

SQL je skupovno orijentiran što znači da se naredba može obaviti nad skupinom redaka podataka ili samo nad jednim retkom.

Osnovna karakteristika SQL-a je jednostavnost pri korištenju.

Tablica (relacija) se kreira jednom izvršnom naredbom. Odmah po kreiranju tablica je raspoloživa za korištenje. Svi podaci pohranjeni su u tablicama i rezultat bilo koje operacije se logički prikazuje u obliku tablice.

Koristeći SQL dobivaju se odgovori na trenutne, unaprijed nepredviđene zahtjeve ili se SQL blokovi "ugrađuju" u klasični viši programski jezik (FORTRAN, COBOL, PL/1, C) omogućavajući klasičnu obradu.

SQL je u velikoj mjeri neproceduralan jer definira što se želi napraviti, a ne kako se to želi napraviti; koji podaci se žele, koje tabele se referenciraju i koji se uvjeti trebaju ispuniti, bez specifikacije procedure za dobivanje željenih podataka.

¹² vidi [http://en.wikipedia.org/wiki/SQL\(20.10.2010.\)](http://en.wikipedia.org/wiki/SQL(20.10.2010.))

Točnije rečeno, SQL je na višoj razini apstrakcije nego klasični viši programski jezici.

Osim toga SQL ne zahtijeva određivanje metode pristupa podacima, tj. nije potrebno opisati način pristupa podacima, već je potrebno samo dati naziv tablice i stupca.

Osnovni problem pri povezivanju je razlika u operandima SQL-a i klasičnih viših programske jezika. Dok je operand klasičnoga višeg programskog jezika zapis (n -torka), odnosno redak tabele, operand SQL-a je cijela Tablica. Da bi se riješio taj problem, u SQL-u je definiran specifičan objekt, CURSOR koji predstavlja mehanizam pojedinačne obrade n -torki unutar definiranoga skupa.

SQL uključuje skupine naredbi za:

- pretraživanje baze (SELECT),
- upis, dodavanje i brisanje redaka u tabeli (UPDATE, INSERT, DELETE),
- definiranje relacijske sheme baze podataka, opis ograničenja u bazi, definiranje pogleda na bazu (CREATE TABLE, CREATE INDEX, CREATE VIEW, ALTER TABLE, DROP TABLE)
- kontrola pristupa bazi (GRANT, REVOKE),
- osiguranje konzistentnosti podataka u bazi (COMMIT WORK, ROLLBACK WORK).

Za prikaz naredbi SQL-a koristit će se baza podataka KADROVI (vidi sliku 3.2. Primjer relacijske baze podataka).

3.5.2 Tipovi podataka, operatori i funkcije SQL-a

3.5.2.1 Tipovi podataka

Svaki stupac baze podataka ima određeni tip (engl. *data type*). SQL dopušta korištenje sljedećih tipova podataka:

Tablica 3.31 Tipovi podataka u SQL-u

Oznaka tipa	Opis
CHAR	Tip podataka koji koristimo za tekst slobodnoga formata. Ovaj tip prihvaca sve ASCII znakove, a maksimalna dužina ovoga tipa je 255 znakova.
TEXT	Označava niz karaktera smještenih u navodnike.
NUMBER	Tip podataka za spremanje numeričkih vrijednosti. Numeričke stupce možemo označiti kao: NUMBER NUMBER(precision) - s ukupnim brojem znamenki NUMBER(precision, scale) - s ukupnim brojem znamenki i brojem znamenki desno od decimalne točke.
DATA	Tip podataka koji koristimo za datum i vrijeme. Standardni datum je oblika DD-MM-YY
NULL	Koristimo ga u slučaju da neki stupac nema vrijednost.

3.5.2.2 Operatori SQL-a

Operatori SQL-a dijele se na:

1. *Aritmetičke operatore (engl. arithmetic operators):*

+, -, *, /

2. *Znakovne operatore (engl. character operators):*

// kojeg koristimo za povezivanje znakovnih vrijednosti

3. *Operatori usporedbe (engl. comparation operators):*

=, >, <, <=, >=, !=(različito od)

[NOT] BETWEEN X AND Y - daje sve vrijednosti veće ili jednake X i manje ili jednake Y. BETWEEN uključuje granične vrijednosti dok NOT BETWEEN ne uključuje.

IS [NOT] NULL - ispituje vrijednost stupca na NULL (NOTNULL)

[NOT] LIKE - Omogućuje pretraživanje na osnovi 'uzorka', tj. dobivanje informacija kada ne znamo točan naziv (vrijednost) određenog atributa tipa 'char'. Naredba koristi dva specijalna karaktera "%"; "_", gdje "%" predstavlja string od 0 ili više karaktera dok "_" predstavlja poziciju jednoga karaktera. Uspoređuje vrijednost varijable (stupca) na 'sličnost' ili 'ne sličnost' sa zadanim uzorkom.

[NOT] IN - ispituje se jednakost sa jednim od članova zadanoga stupca

4. *Logičke operatore (engl. logical operators):*

NOT, AND, OR

5. *Unarne operatore:*

- +, - (predznak)
- ~ (bitovni "NE")

6. *Binarne operatore:*

- & binarno "I"
- | binarno "ILI"
- ^ binarno isključivo „ILI“

3.5.2.3 Funkcije SQL-a

1. Numeričke funkcije na jednome retku (engl. *single row number functions*):

Tablica 3.32 Numeričke funkcije SQL-a

Funkcija	Opis
ABS(<i>n</i>)	daje absolutnu vrijednost
CEIL(<i>n</i>)	daje najmanji cijeli broj veći ili jednak <i>n</i>
FLOOR(<i>n</i>)	daje najmanji cijeli broj manji ili jednak <i>n</i>
MOD(<i>m,n</i>)	daje ostatak kod dijeljenja <i>m</i> sa <i>n</i>
POWER(<i>m,n</i>)	daje <i>m</i> na <i>n</i> -tu potenciju gdje <i>n</i> mora biti cijeli broj
SIGN(<i>n</i>)	daje za <i>n</i> <0 vrijednost -1, za <i>n</i> =0 daje 0, a za <i>n</i> >0 daje 1
SQRT(<i>n</i>)	daje drugi korijen broja <i>n</i> , a ako je <i>n</i> <0 onda daje NULL
TRUNC(<i>n[m]</i>)	daje broj <i>n</i> sa <i>m</i> decimalnih mesta (ostatak decimala otpisuje)
ROUND (<i>m,n</i>)	zaokružuje broj <i>m</i> na <i>n</i> decimala

2. Znakovne funkcije na jednome retku (engl. *single row character functions*):

Pod 'char' podrazumijevamo ili naziv stupca tipa char, ili niz karaktera unutar navodnika.

Tablica 3.33 Znakovne funkcije SQL-a

Funkcija	Opis
CHAR (char)	daje znak koji ima ASCII vrijednost
LOWER (char)	u izrazu char sva slova pretvoriti u mala
SUBSTR (char,<i>m[,n]</i>)	daje dio izraza char, koji počinje od <i>n</i> -tog znaka i ima duljinu <i>n</i> (pr. SUBSTR("PROJEKTIRAJE";1,4)="PROJ")
UPPER (char)	sve znakove u izrazu char pretvoriti u velika slova
ASCII (char)	daje vrijednost prvoga znaka u izrazu char
LENGTH (char)	daje duljinu (broj znakova) izraza char
INSTR (char;ch[<i>n</i>])	u izrazu char traži podizraz ch polazeći od pozicije <i>n</i> , a ako je nađen vraća se njegova pozicija, inače 0
TO_NUM (char)	pretvara niz karaktera (numeričkih) u broj
TO_CHAR (char)	pretvara broj u niz znakova
NVL (char1,char2)	ako je char1 null vrijednost onda vraća char2, a inače vraća char1
DECODE (char1, cs1, rst1, cs2, rst2,..., dft)	rezultat je rst1 ako je char=rst1; rezultat je rst2 ako je char=rst2,...,posljednja dft vrijednost je default

3. Agregatne funkcije:

Agregatne funkcije (grupirajuće, funkcije nad grupama podataka, engl. *group functions*) daju rezultat temeljen na grupi podataka, a ne na jednom podatku, tj. retku.

Tablica 3.34 Grupne funkcije SQL-a

Funkcija	Opis
AVG (atribut)	daje srednju vrijednost atributa
MAX (atribut)	daje maksimalnu vrijednost atributa
MIN (atribut)	daje minimalnu vrijednost atributa
SUM (atribut)	daje zbroj vrijednosti atributa
COUNT (*)	daje broj selektiranih n -torki
COUNT (atribut)	daje broj notnull-vrijednosti u atributu

3.5.3 Naredbe SQL - DML

SQL - DML su naredbe za manipulaciju s podacima.

Postoje sljedeće naredbe SQL-DML-a:

- SELECT - naredba za pretraživanje, tj. naredba za prikaz sadržaja relacijske baze
- INSERT - naredba za dodavanje redaka postojećoj tablici
- DELETE - naredba za brisanje (izbacivanje) redaka unutar tabele
- UPDATE - naredba za izmjenu vrijednosti pojedinih n -torki
- COMMIT WORK - naredba za fiksiranje promjene u bazi podataka
- ROLLBACK WORK - naredba za poništavanje svih promjena u bazi podataka
- SAVEPOINT - naredba za djelomično poništavanje promjena u bazi podataka
- BEGIN WORK, COMMIT WORK – za potvrđivanje promjena u bazi podataka

Prikažimo samo naredbu SELECT.

3.5.3.1 SELECT - Naredba za pretraživanje

Naredba SELECT je najvažnija i najkompleksnija naredba SQL-a. Služi za izdvajanje i ispis podataka nekog upita. Primjerima ove naredbe vidjet ćemo da se SQL-om mogu jednostavno izvoditi sve operacije relacijske algebre.

Osnovni oblik SELECT naredbe je:

```
SELECT [ALL/DISTINCT] atribut [, atribut...] FROM relacija [, relacija...]
      [WHERE uvjet ]
      [GROUP BY atribut [, atribut] ]
      [HAVING uvjet]
      [ORDER BY specifikacija uređaja [ASC/DESC] ]
```

Dijelovi naredbe unutar] nisu obavezni.

Tablica 3.35 Značenje parametara naredbe SELECT

Parametar	Značenje
ALL	Svi reci koji zadovoljavaju SELECT naredbu bit će prikazani.
DISTINCT	Ukoliko dva ili više redaka koji zadovoljavaju uvjet u SELECT naredbi imaju iste vrijednosti u svim stupcima, korištenjem DISTINCT opcije bit će prikazan samo jedan takav redak.
ASC	Sortiranje rastućim retkom, a ako ne navedemo ništa podrazumijeva se ASC.
DESC	Sortiranje padajućim retkom.

Tablica 3.36 Značenje klauzula naredbe SELECT

Klauzula	Značenje
FROM	Služi za definiranje Tablica koje želimo pretraživati.
WHERE	WHERE klauzula nije obavezna, a može se koristiti sa SELECT, UPDATE i DELETE naredbama. Uloga ove klauzule je definiranje uvjeta. Definira ograničenja koja utječu na to koji će reci dobiveni kao rezultat SELECT naredbe. Uvjete definiramo kao logičke izraze koji kao rezultat provjere na svakome pojedinom retku daju true ili false. Korištena u SELECT bloku ona omogućuje: <ul style="list-style-type: none"> - selekciju pojedinih n-torki relacije (redak tabele) - selekciju n-torki koje zadovoljavaju višestruke uvjete - selekciju n-torki koje zadovoljavaju bar jedan od više uvjeta - selekciju n-torki koje ne zadovoljavaju određene uvjete - selekciju n-torki istovremenim korištenjem AND i OR logičkih operatora - selekcija n-torki unutar zadanog intervala - selekcija n-torki koje zadovoljavaju vrijednost u listi istih - selekcija n-torki koje sadrže određenu kombinaciju karaktera.
GROUP BY	Ova klauzula određuje kriterij grupiranja. Ona kao rezultat daje grupne informacije za svaku različitu vrijednost atributa po kome se obavlja grupiranje. Grupiranje se može obaviti po više stupaca.
HAVING	Određuje kriterije za selekciju grupe budući da su grupe već formirane sa GROUP BY klauzulom, tj. definira koje će grupe iz GROUP BY klauzule biti uzete u obzir.
ORDER BY	Ova klauzula sortira rezultat pretraživanja. Korištenjem ove klauzule moguće je sortirati rezultirajuću tabelu po jednom ili više atributa u rastućem ili padajućem redoslijedu. Za specificiranje rastućeg ili padajućega redoslijeda koristi se parametar ASC, odnosno DESC. ORDER BY je uvijek posljednja klauzula u SELECT bloku. Kada se sortiranje obavlja po stupcu s NULL vrijednostima, n -torce s NULL vrijednostima moguće su na početku rezultirajuće tabele, bez obzira je li sortiranje po rastućem ili padajućem redoslijedu.

WHERE i GROUP BY klauzula mogu se koristiti zajedno, pri čemu WHERE klauzula ide uvijek prije GROUP BY.

WHERE klauzulom najprije se obavi selekcija n -torki, a zatim se selektirane n -torke grupiraju, pa se eventualno obavi formiranje grupa HAVING klauzulom.

Primjer 1. Ispis čitavoga sadržaja jedne relacije

SELECT * FROM Radnik;

(* označava da će biti prikazani svi stupci iz svih tablica navedenih u FROM klu-
uzuli)

Rezultat:

Tablica 3.37 Rezultat naredbe SELECT * FROM Radnik

Mbr	Prezime	Mje	Adr	Drod	Soj	Srm	Sss
1001	Ivan Perić	Zagreb	Pulska10	1948-11-26	001	P1	101
1002	Mario Vukić	Split	Zadars.3	1952-03-19	002	P2	102
1003	Zlatko Puh	Split	Bana j.7	1959-09-13	002	P3	100
1004	Gordan Marić	Zagreb	Illica 7	1968-11-10	001	P1	101
1005	Ana Žurić	Rijeka	nul	1971-03-06	003	P4	102
1006	Branka Horvat	Split	Velbits.9	1962-04-17	002	P5	102
1007	Josip Ban	Rijeka	Korzo 8	1964-12-25	003	P3	100

Primjer 2. Ispis pojedinih atributa jedne relacije

a) **SELECT Mbr, Prezime FROM Radnik;**

Ova naredba ispisuje attribute Mbr, Prezime navedenim redoslijedom (u relacijskoj algebri ovo je operacija projekcije).

Rezultat:

Tablica 3.38 Rezultat SELECT Mbr, Prezime FROM Radnik

Mbr	Prezime
1001	Ivan Perić
1002	Mario Vukić
1003	Zlatko Puh
1004	Gordan Marić
1005	Ana Žurić
1006	Branka Horvat
1007	Josip ban

b) SELECT DISTINCT Mje FROM Radnik

Ova naredba omogućuje prikazivanje samo različitih poslova.

Rezultat:

Tablica 3.39 Rezultat naredbe SELECT DISTINCT Mje FROM Radnik

MJE
Zagreb
Split
Rijeka

Primjer 3. Selekcija pojedinih n -torki jedne relacije (operacija selekcije u relacijskoj algebri)

a) SELECT * FROM Radnik WHERE Mje='Zagreb'

Ova naredba ispisuje sve radnike iz Zagreba.

Rezultat:

Tablica 3.40 Rezultat naredbe SELECT DISTINCT MJE FROM RADNIK

MBR	PREZIME	MJE	ADR	DROD	SOJ	SRM	SSS
1001	Ivan Perić	Zagreb	Pulska10	1948-11-26	001	P1	101
1004	Gordan Marić	Zagreb	llica 7	1968-11-10	001	P1	101

b) SELECT Prezime, Soj FROM Radnik WHERE Soj!=002

Ova naredba ispisuje imena svih radnika koji ne rade u organizacijskoj jedinici čija je šifra 002.

Rezultat:

Tablica 3.41 Rezultat naredbe SELECT Prezime, Soj FROM Radnik WHERE Soj!=002

Prezime	Soj
Ivan Perić	001
Goran Marić	001
Ana Žurić	003
Josip Ban	003

c) SELECT Srm, Nrm, Bod FROM Radnomj WHERE Sss='102' AND Bod>350

Ispisuje šifru i naziv radnoga mjesta onih radnih mjesta čija je stručna spremna 102 i imaju bodove veće od 350.

Rezultat:

Tablica 3.42 Rezultat naredbe

SELECT Srm, Nrm, Bod FROM Radnomj WHERE Sss='102' AND Bod>350

Srm	Nrm	Bod
P2	Planer	400
P5	Referent nabave	355

d) **SELECT Nrm FROM Radnomj WHERE Bod BETWEEN 240 AND 355**

Naredba ispisuje sva radna mjesta čiji je broj bodova u intervalu 240 i 355. Uvjet BETWEEN je identičan uvjetu WHERE (Bod>=240) AND (Bod<=355).

Rezultat:

Tablica 3.43 Rezultat naredbe

SELECT Nrm FROM Radnomj WHERE Bod BETWEEN 240 AND 355

Nrm
Planer
Operator
Pravnik
Referent nabave

e) **SELECT Prezime, Mje FROM Radnik WHERE Prezime LIKE 'M%';**

Ova naredba ispisuje prezime i mjesto radnika čije ime i prezime počinje slovom M.
Rezultat:

Tablica 3.44 Rezultat naredbe

SELECT Prezime, Mje FROM Radnik WHERE Prezime LIKE 'M%'

Mbr	Prezime	Mje
1002	Mario Vukić	SPLIT

Znak % zamjenjuje niz bilo kojih znakova (0 ili više), a znak _ zamjenjuje jedan znak.

f) **SELECT Nrm, Bod FROM Radnomj WHERE Nrm NOT IN ('Zavarivač', 'Referent nabave')**

Ova naredba ispisuje radna mjesta, broj bodova tih radnih mjesta koji nisu zavarivači ili referenti nabave. Korištenjem operatora NOT IN izbjegavamo višestruko korištenje operatora OR.

Rezultat:

Tablica 3.45 Rezultat naredbe

SELECT Nrm, Bod FROM Radnomj WHERE Nrm NOT IN ('Zavarivač', 'Referent nabave')

Nrm	Bod
Planer	400
Operater	240
Pravnik	315

g) **SELECT Prezime FROM Radnik WHERE Prezime LIKE '%N'**

Naredba ispisuje prezimena radnika čije ime završava sa N.

Rezultat:

Tablica 3.46 SELECT Prezime FROM Radnik WHERE Prezime LIKE '%N'

PREZIME
Josip Ban

Mogući su i drugi primjeri korištenja LIKE:

... gdje je treći znak imena A: ... WHERE Prezime LIKE '_ _ A'

... gdje je radno mjesto dugačko 5 znakova: ... WHERE Mje LIKE '_ _____'

... gdje je u imenu mjesta slovo E iza slova A: ... WHERE Mje LIKE '%A%E'

h) **SELECT Mbr, Prezime FROM Radnik WHERE Adr IS NULL ;**

Ispisuje matični broj i prezime radnika čije su adrese nepoznate.

Rezultat:

Tablica 3.47 SELECT Mbr, Prezime FROM Radnik WHERE Adr IS NULL

Mbr	Prezime
1005	Ana Žurić

Primjer 4. Selekcija n -torki jedne relacije sortiranim ispisom.

a) **SELECT Nrm, Sss, Bod FROM Radnomj ORDER BY Bod ASC ;**

Ispisuje naziv radnoga mjesta, šifru stručne spreme i bodove radnih mjesta tako da budu uređeni po rastućemu redoslijedu bodova.

Rezultat:

Tablica 3.48 Rezultat naredbe SELECT Nrm, Sss, Bod FROM Radnomj ORDER BY Bod ASC

Nrm	Sss	Bod
Zavarivač	101	205
Operater	100	240
Pravnik	102	315
Referent nabave	102	355
Planer	102	400

- b) **SELECT Sss, Nrm, Bod FROM Radnomj WHERE Bod>205 ORDER BY Sss ASC, Nrm DESC**

Ova naredba prikazuje naziv i bodove radnoga mjesta i šifru stručne spreme radnoga mjesta čiji su bodovi veći od 205, u rastućem redoslijedu šifre stručne spreme i padajućem redoslijedu naziva radnoga mjesta.

Rezultat:

Tablica 3.49 SELECT Sss, Nrm, Bod FROM Radnomj WHERE Bod>205 ORDER BY Sss ASC, Nrm DESC

Sss	Nrm	Bod
100	Operater	240
102	Referent nabave	355
102	Pravnik	315
102	Planer	400

Primjer 5. Selekcija n -torki neke relacije korištenjem funkcija.

- a) **SELECT MIN(Bod), AVG(Bod), MAX(Bod), COUNT(*) FROM Radnomj**
Izračunava minimalan, srednji i maksimalni broj bodova te broj n -torki radnih mjestra koja su sudjelovala u izračunu.

Rezultat:

Tablica 3.50 Rezultat naredbe SELECT MIN(Bod), AVG(Bod), MAX(Bod), COUNT(*) FROM Radnomj

MIN(Bod)	AVG(Bod)	MAX(Bod)	COUNT(*)
205	303	400	5

- b) **SELECT MIN(Bod), AVG(Bod), COUNT(*), Sss FROM Radnomj
GROUP BY Sss ;
HAVING COUNT(*) > 2**

Izračunati minimalni broj bodova, srednji broj bodova, broj n -torki koje su sudjelovale u izračunu, izračun načiniti po grupama Sss koje imaju više od dva elementa.

Rezultat:

Tablica 3.51 Rezultat naredbe

SELECT MIN(Bod), AVG(Bod), COUNT(*), Sss FROM Radnomj GROUP BY Sss; HAVING COUNT(*) > 2

MIN(Bod)	AVG(Bod)	COUNT(*)	Sss
315	356.66	3	102

c) **SELECT Nrm, ROUND(Bod*0.05,1)"Stimulacija"
FROM Radnomj WHERE Bod BETWEEN 205 AND 390**

Ova naredba ispisuje radna mjesta i izračunava stimulaciju za to radno mjesto tako da broj bodova pomnoži sa 0.05, tj. izračuna 5% od bodova i to zaokruži na 1 decimalu, gdje je broj bodova iz intervala [205,390].

Rezultat:

Tablica 3.52 Rezultat naredbe

SELECT Nrm, ROUND(Bod*0.05,1)"Stimulacija" FROM Radnomj WHERE Bod BETWEEN 205 AND 390

Nrm	Stimulacija
Zavarivač	10.3
Operater	12
Pravnik	15.8
Referent nabave	17.8

Vidi se da je zaglavlje rezultirajućega stupca ROUND((Bod*0.05),1) privremeno zamijenjeno izrazom pod dvostrukim navodnicima "Stimulacija".

d) **SELECT Prezime II,"II Mje "Radnik" FROM Radnik
WHERE Soj=002 ORDER BY Prezime**

Ispisuje prezimena radnika iza kojih trebaju biti ispisana mjesta stanovanja razdvojena zarezom, koji rade u organizacijskoj jedinici šifre 002 i poredani su po račućem redoslijedu prezimena.

Rezultat:

Tablica 3.53 Rezultat naredbe

SELECT Prezime II,"II Mje "Radnik" FROM Radnik WHERE Soj=002 ORDER BY Prezime

Radnik
Branka Horvat, Split
Mario Vukić, Split
Zlatko Puh, Split

e) **SELECT Nrm, LENGTH(Nrm) FROM Radnomj**

Naredba ispisuje nazine radnih mjesta i dužinu naziva radnih mjesta.

Rezultat:

Tablica 3.54 Rezultat naredbe SELECT Nrm, LENGTH(Nrm) FROM Radnomj

Nrm	Length
Zavarivač	9
Planer	6
Operater	6
Pravnik	7
Referent nabave	14

f) **SELECT Nrm, Srm, DECODE(Nrm,'Operater', 1,'Zavarivač',2,'Pravnik',3,4)**

'Klasa' FROM Radnomj

Koristeći stupac Nrm formiraj stupac Klasa tako da je za posao operatera vrijednost klase 1, za posao zavarivača 2, za posao pravnika 3, a inače 4.

Rezultat:

Tablica 3.55 Rezultat naredbe

SELECT Nrm, Srm, DECODE(Nrm,'Operater',1,'Zavarivač',2,'Pravnik',3,4) 'Klasa' FROM Radnomj

Nrm	Srm	Klasa
Zavarivač	P1	2
Planer	P2	4
Operater	P3	1
Pravnik	P4	3
Referent nabave	P5	4

SELEKCIJA U SELEKCIJI

Prvi je način povezivanja Tablica relacijske baze upotrebom selekcije u selekciji, tj. dinamičkom zamjenom rezultata jednog upita u WHERE klauzuli drugog upita. Prvi upit, odnosno upit u zagradama naziva se unutarnji upit i on se uvijek izvršava prvi. Drugi upit se naziva vanjski uvjet. Prije nego što počne njegovo izvršavanje, unutarneći upit je završen i u zagradi se nalazi konkretna vrijednost rezultata tog upita.

Primjer 6. Selekcija n -torki jedne relacije korištenjem poluupita (selekcija u selekciji)

a) **SELECT Mbr, Prezime FROM Radnik
WHERE Sss=(SELECT Sss FROM Radnik
WHERE Prezime=' Mario Vukić ');**

Ova naredba ispisuje matični broj i imena onih radnika koji imaju istu stručnu spremu kao i Mario Vukić.

Rezultat:

Tablica 3.56 Rezultat naredbe

SELECT Mbr, Prezime FROM Radnik WHERE Sss=(SELECT Sss FROM Radnik WHERE Prezime=' Mario Vukić ')

Mbr	Prezime
1002	Mario Vukić
1005	Ana Žurić
1006	Branka Horvat

b) **SELECT Nrm, Bod FROM Radnomj
WHERE Srm IN (SELECT Srm FROM Radnik
WHERE Drod > ' 1964-11-10 ');**

Ispisuje naziv radnih mjesto onih radnika koji su rođeni prije 11. 10. 1964. god.

Rezultat:

Tablica 3.57 Rezultat naredbe

SELECT Nrm, Bod FROM Radnomj WHERE Srm IN (SELECT Srm FROM Radnik WHERE Drod > ' 1964-11-10 ');

Nrm	Bod
Zavarivač	205
Pravnik	315

3.5.4 Naredbe SQL - DDL

Naredbe za definiranje baze podataka omogućuju izmjenu dijelova sheme baze podataka.

U ovu skupinu naredbi spadaju:

- CREATE DATABASE, CREATE TABLE, CREATE VIEW, CREATE INDEX
 - naredbe za kreiranje baze podataka
- ALTER TABLE, ALTER INDEX - naredbe za izmjenu baze podataka
- RENAME - naredba za preimenovanje objekta u bazi podataka
- GRANT, REVOKE - naredbe za kontrolu pristupa bazi.

3.5.4.1 CREATE TABLE

Naredbom CREATE TABLE definira se nova relacija, tj. opisuje njezina relacijska shema. Naredba ima oblik:

CREATE TABLE relacija (atribut [,atribut...]);

Naredbom se definira naziv relacije te nazivi i formati podataka atributa relacije. Uz svaki atribut opisuju se ograničenja vrijednosti atributa. Svi atributi definirani kao NOT NULL moraju dobiti vrijednost prije ubacivanja n -torke.

Primjer 1. Definirati relacijsku shemu relacije Radnik

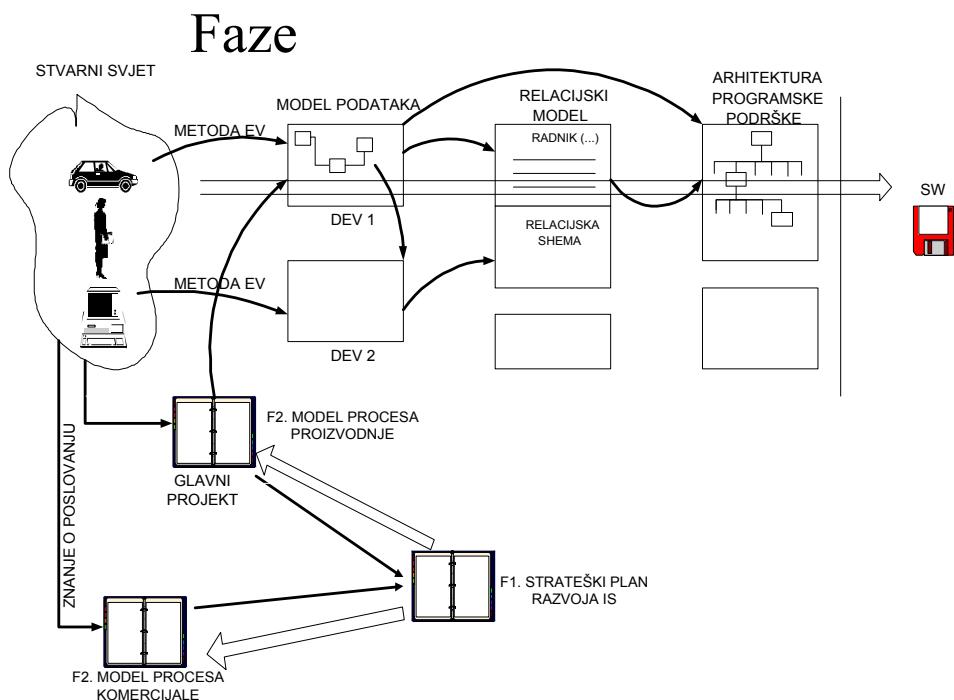
Tablica 3.58 Naredba CREATE TABLE

CREATE TABLE Radnik		
(Mbr	Char(4)	NOT NULL UNIQUE
Prezime	Char(20)	NOT NULL
Mje	Char(15)	
Adr	Char(20)	
Drod	Date	
Soj	Char(3)	NOT NULL
Srm	Char(2)	NOT NULL
Sss	Char(3)) ;	

4 MODELI PODATAKA

4.1 Problemi, strategije i kriteriji modeliranja podataka

Ovome je poglavlju za cilj izložiti osnovne pojmove o modelu podataka. U poglavlju 2.3 MIRIS prikazane su osnovne aktivnosti razvoja informacijskoga sustava čiji dio je modeliranje podatka, a na slici 4.1 prikazan je položaj metoda za modeliranje podataka u cjelokupnometu procesu razvoja informacijskoga sustava.



Slika 4.1 Položaj modela podataka u odnosu na ostale metode

Model podataka je shvaćan i analiziran kao skup entiteta i operacija nad njima.

Jedan od većih problema razvoja informacijskoga sustava je nepostojanje jedinstvene baze podataka i jedinstvenog modela podataka, odnosno postavlja se pitanje kako razriješiti probleme razvoja i postići integralnost informacijskoga sustava? To je moguće uz pomoć triju strategija:

- Uspostavlja se jedinstveni šifrarski sustav
- Kontinuirano se unapređuju i koriste CASE alati
- Koriste se semantički bogate metode za modeliranje podataka. Razvoj informacijskoga sustava oslanja se svim programskim proizvodom na jedinstveni model podataka.

Semantički¹³ bogate metode (metode mrežne prezentacije znanja) dovode do znanjem bogatih modela podataka, točnije do modela informacija.

Prve od tih metoda bile su semantičke mreže i danas postoje u većemu broju različitih oblika. **Semantička mreža** predstavlja znanje sadržano u pojmovima skupova objekta (čvorova, engl. *node*) i binarnih veza među čvorovima (engl. *direct labeled edges*). Po definiciji **baza znanja** je skup objekata i skup odnosa među objektima. Izmjena baze znanja predstavlja umetanje ili brisanje objekata i veza među njima.

Semantičke mreže se razvijaju od 1968. godine (Brodie, 1984.). Najvažniji intelektualni alati semantičkih mreža su organizacijske osi, odnosno apstrakcije prema kojima se znanje može rasporediti u model. Uočeni su osnovni temeljni aksiomi (pristupi, misli, metode) za organizaciju znanja i to: klasifikacija, agregacija, generalizacija i različite vrste veza (asocijacija) te operacije nad definiranim konceptima.

Danas postoje različite metode za modeliranje podataka. Ideje baštine od semantičkih mreža te su i same manje ili više semantički bogate. S obzirom na količinu znanja koje se može ugraditi u model, odnosno s obzirom na koncepte koji omogućavaju semantički bogatu prezentaciju podataka, modeli podataka mogu se dijeliti u generacije.

Modeli podataka (tipovi podataka) I. generacije su:

Svaki programski jezik je zaseban model podataka, a podaci se modeliraju preko koncepata kojima dani jezik raspolaze, kao što su na primjer: pointer, integer, real, matrica, stack i dr.

¹³ Semantika – 1. *lingv.* Grana lingvistike koja proučava pojedine riječi, njihove oblike i grupacije kao nositelje određena značenja te kao sredstva za označavanje predmeta, pojava i odnosa u materijalnom i duhovnom svijetu; utvrđuje glavne procese koji vode do promjena u značenju riječi 2. *log. a.* opća označka za teoriju istinitosti logičkih rečenica i nizove **b. mat.** dio formalnoga sustava koji svakom iskazanom elementu logičkoga računa pridružuje jedan od znakova koji pripadaju svjetu označenih (signifikanta) (Anić, 2003.)

Modeli podataka II. generacije su:

- Funkcionalni model podataka
- Hijerarhijski model podataka
- Mrežni model podataka (Bachman, 1969.)
- Klasični relacijski model podataka (RM) (Codd, 1970.)
- Warnierovi dijagrami
- Ovi modeli sadrže koncepte za prezentaciju podataka kao: stablo, set, relacija, i dr. Oni nisu dovoljno semantički bogati i imaju samo “atomsku” semantiku.

Modeli podataka III. generacije su:

- Model entiteti-veze (autor: Chen)
- Prošireni relacijski model – RM/T (autor: Codd)
- SDM-IBM (autor: IBM)
- Model podataka semantičkih mreža (autor: Roussopoulos i Mylopoulos)
- Semantički model podataka - SDM (autor: Hammer i McLeod)
- Petrijeve mreže (autor: Reti di Petri)
- Semantic Association Model - SAM (autor: Stanlay)
- D-Graph Model (autor: Weber)
- Palmer (autor: Palmer)
- Diam II (autor: Senko)

Ovi modeli imaju koncepte generalizacije i agregacije te, pored “atomske”, omogućuju ugradnju “molekularne” semantike u reprezentaciju. Dobro modeliraju poslovni sustav, korisnički su razumljivi, ali danas uglavnom bez razvijene softverske podrške.

Premda su ovi modeli različiti u detaljima, skoro svi posjeduju koncepte pomoću kojih je moguće obuhvatiti više informacijskih zahtjeva i bolje opisati entitete poslovnoga sustava.

4.2 Osnovni pojmovi o modeliranju podataka

Uvest ćemo neke osnovne pojmove vezane uz modeliranje podataka, uz minimalno ponavljanje ranije definiranih pojmove u knjizi.

Metoda za modeliranje podataka (engl. *Data modelling method*) je definirani postupak nalaženja i prikazivanja informacijskih objekata i njihovog međusobnog odnosa.

Primjenom procesa modeliranja podataka zasnovanog na nekoj metodi dobiva se kao rezultat model podataka. Modeliranje je proces razvoja modela. Model nasta-

je procesom apstrakcije u kome se prvo biraju relevantni koncepti koje reprezentacija treba sadržati, a zatim se svakome konceptu pridružuju relevantne osobine (atributi) koje se žele prikazati u okviru modela.

Model podataka sustava je pojednostavljena reprezentacija o relevantnim karakteristikama sustava preko skupa entiteta (objekata), veza među entitetima i atributa entiteta i agregacija entiteta. Model podataka treba istovremeno sadržavati i same podatke i njihovu interpretaciju, odnosno treba predstavljati strukturirani skup informacija o prošlosti i sadašnjosti sustava.

Osnovni zadatak istraživanja u području modeliranja podataka je naći koncepte u okviru metode pomoću kojih će se izgraditi model koji predstavlja naše znanje o sustavu.

Koncepti su ideje (simboli, znakovi, intelektualno sredstvo, aksiomi, zamisli, uzorci) koji predstavljaju klasu odabranih pojmova iz poslovnoga sustava. Neki koncepti imaju grafički prikaz u obliku simbola. Dobiveni je model podataka, sa stajališta statičke strukture, skup povezanih koncepata (slika) koji predstavlja cijeli sustav.

Općenito se na području modela podataka došlo do saznanja da se svaki model podataka sastoji iz triju dijelova i to:

- **Strukture** (engl. *structure*)
- **Ograničenja** (engl. *constraints*)
- **Operatora** (engl. *operators*)

Ponekad se struktura i ograničenja nazivaju strukturnom komponentom modela podataka jer zajedno omogućuju opis statičkih svojstava poslovnoga sustava u modelu. Dinamika u modelu podataka predstavlja se pomoću operatora.

Pomoću **strukture** podataka, podaci o pojmovima poslovnoga sustava spajaju se u međusobno povezane grupe i tako se znanje o grupiranju i interakciji pojmovima iz poslovnoga sustava prenosi u model podataka. Na primjer, grupa podataka će biti recepti, druga grupa podataka će biti pacijenti, a treća lijekovi. Lijekove ćemo povezati s receptima, ali ne i s pacijentima.

Pomoću **ograničenja** na podatke u strukturi podataka unosi se dalje znanje o interakciji koncepata iz poslovnoga sustava. Na primjer, što je zabranjeno, ili u kojoj količini je dopušteno, pa možemo ograničiti da jedan recept može propisati samo jedan lijek, ali da pacijent može dobiti više recepta.

Pomoću **operatora** nad podacima, uvodi se skup operacija nad strukturu podataka, koje omogućuju dinamiku podataka o konceptima analogno dinamici stvari u poslovnom sustavu. Na primjer, operacija bi bila upis novoga recepta ili složena operacija pretrage pacijenata koji su primili ove godine bilo koji antibiotski lijek tvrtke „Pliva“.

Model podataka (engl. *Data model*) je skup međusobno povezanih podataka koji opisuju entitete, veze i atribute poslovnog sustava. On je reprezentacija skupa podataka koji se modelom interpretiraju preko aspekta: strukture, ograničenja i operatora.

Struktura modela podataka je skup entiteta i veza koji interpretira podatke klasificirajući ih u tipove entiteta i tipove veza među tipovima entiteta, zajedno s relevantnim svojstvima tipova entiteta.

Ograničenja modela podataka su koncepti strukture modela, a omogućuju daljnju interpretaciju podataka razdvajajući dopuštena od zabranjenih stanja skupa podataka preko: dopuštenih podataka u okviru jednoga tipa entiteta, dopuštenih vrijednosti podataka pojedinoga svojstva tipa entiteta, dopuštenih povezivanja među tipovima entiteta.

Operatori modela podataka čine skup koncepata koji omogućuju interpretaciju dinamičkih karakteristika skupa podataka. Koncept strukture i ograničenja reprezentira statička svojstva poslovnog sustava, a operatori omogućuju izmjenu stanja podataka u bazi podataka u skladu s promjenom stanja u poslovnome sustavu.

Struktura modela podataka gradi se od temeljnih koncepata: entitet, veza i atribut.

Entitet (engl. *Entity*) sustava je neki njegov realni ili konceptualni element, to je neka posebnost što u poslovnome sustavu postoji i jasno se razlikuje od drugih entiteta. Na primjer: muškarci, narudžbe, organizacijske jedinice i dr.

Veza (engl. *Relationship*) je koncept koji predstavlja neku interakciju među entitetima u sustavu, odnosno predstavlja znanje o njihovoj povezanosti. Na primjer: brak je veza između muškaraca i žena.

Atribut entiteta (engl. *Attribute*) je neko svojstvo entiteta. Na primjer: ime radnika, datum prispjeća narudžbe, faktor uvećanja, brzina vozila, šifra općine i dr.

U modelu podataka, kao i u nizu drugih modela (matematičkome sustavu jednadžbi putanje rakete, geografskoj karti teritorija, nacrta broda, shemi električnih strujnih krugova pojačala) koji predstavljaju neki sustav, ne opisuje se potpuni skup znanja o sustavu, već se obavlja odabir relevantnih karakteristika sustava te se u model uključuju samo relevantni podaci.

Relevantni podatak (do koga nam je stalo, od interesa) je onaj podatak za koga smo zainteresirani te ulazi u model sustava, odnosno onaj koji pri analizi podataka smatramo važnim s aspekta namjene informacijskoga sustava.

4.3 Apstrakcija podataka

Kako savladati (opisati) složene sustave? Sustavi se opisuju i predstavljaju modelom sustava. Model se gradi pomoću pravila koje propisuje metoda. Metoda je proizvod sastavljen od intelektualnih alata (koncepata) pomoću kojih se gradi model sustava. Tako na primjer metodu EV možemo zamisliti kao stolarsku radionicu u kojoj ima niza alata (pila, blanja, čekić,...). Stolar odabire pojedine fizičke alate primjenjuje ih gradeći proizvod, na primjer stolicu. Slično će projektant IS birati koncepte u metodi (intelektualne alate) i graditi model.

Intelektualni alati u metodama za modeliranje podataka zasnovani su na principu apstrakcije.

Apstrakcija (engl. *abstraction*) je kontrolirano uključivanje detalja u cjelinu, pri čemu u mislima istovremeno poima ideja detalja cjeline i veze među njima.

Apstrakcija se sastoji od dvaju kretanja različitih smjerova i to: apstrakcija k općem (skraćeno apstrakcija) i apstrakcija k pojedinačnom (skraćeno detaljizacija). Ova dva smjera skraćeno možemo zapisati kao: apstrakcija (detaljizacija).

Apstrakcija se još naziva i „skrivanje“ detalja, ili „izvlačenje“ općih karakteristika. **Apstrakcija k općem** je spoznajni proces u kome se od uočenoga pojedinačnog elementa zaključi o postojanju skupa koji sadržava više sličnih pojedinačnih elemenata. Na primjer: Na dokumentu Osobna iskaznica je u polju Ime napisan podatak „Marko“. Apstrakcija je sljedeći niz zaključaka: Marko je ime pojedinoga studenta. Ima još takvih podataka i ima još pojedinih studenata. Svi oni predstavljaju natponjam Student. Dakle skup Student je apstraktan naziv za niz pojedinačnih studenata. Ovdje je koncept Student apstraktan koncept u odnosu na osobu Marka koja je konkretna. Pored toga je korištena metoda „apstrakcije“ u ovome zaključivanju koja povezuje konkretno i pojedinačno s apstraktnim i u svjesnom umu izgrađenome pojmu Student. Valja naglasiti da je ljudskom umu to prirodan način funkcioniranja.

Inverzan postupak od apstrakcije je detaljiziranje. **Detaljiziranje** je spoznajni proces u kome se za uočeni skup pita od kojih se elemenata skup sastoji i potom uzrokuje jedan element skupa i on dovede u svjesni um te analiziraju svojstva odabranog elementa. Na primjer: Ako u modelu imamo apstraktan koncept Student koji je projektant IS ranije oblikovao, onda se možemo pitati, od kojih se detalja taj skup sastoji? Kada „uđemo“ u taj skup naći ćemo osobu imena Marko. Tada promatramo samo tu osobu i njezina ostala svojstva (adresu stanovanja, datum rođenja, ...). Korištena je metoda detaljiziranja pomoću koje smo od općega došli do pojedinačnog. Naglasimo da je naziv „detaljiziranje“ samo naziv smjera kretanja metode apstrakcije. Također se u jednome zaključivanju i poopćava i detaljizira i to po nekoliko puta krećući se prema gore i nazad.

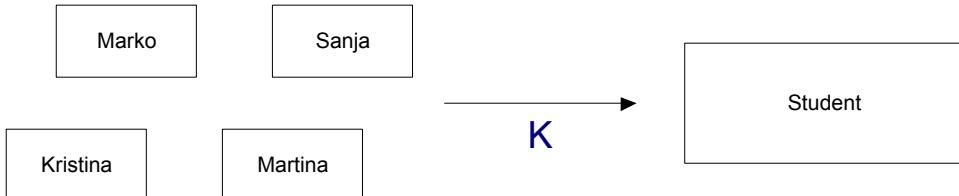
Postoji više vrsta apstrakcije. Nama su za gradnju metoda za modeliranje podataka od interesa neke osnovne vrste apstrakcije (s detaljizacijom¹⁴⁾ i to:

- Klasifikacija (uzorkovanje)
- Generalizacija (specijalizacija)
- Agregacija (dekompozicija)
- Veza (veza)

Pored apstrakcija postoje i druge ideje, principi i definicije od kojih se grade metode.

4.3.1 Klasifikacija

Klasifikacija (ili tipizacija) (engl. *classification*) je apstrakcija u kojoj se skup sličnih objekata predstavlja jednom klasom objekata.

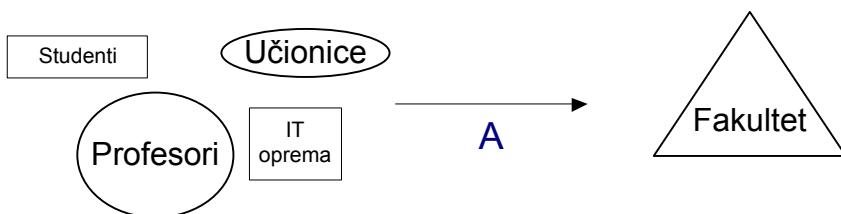


Slika 4.2 Klasifikacijska apstrakcija

Slični objekti su oni objekti koji imaju iste atribute, koji mogu stupiti u iste veze s drugim objektima i na koje se mogu primjeniti iste operacije.

4.3.2 Agregacija

Agregacija (engl. *aggregation*) je apstrakcija gdje se skup tipova entiteta i njihovih veza (te svojstava entiteta i veza) predstavlja novim izvedenim tipom entiteta.



Slika 4.3 Agregacijska apstrakcija

¹⁴ podrazumijeva se da svaka vrsta apstrakcije ima oba smjera

4.3.3 Generalizacija

Generalizacija (engl. *generalisation*) je apstrakcija gdje se skup djelomično sličnih tipova entiteta tretira (predstavlja) kao novi izvedeni tip entiteta na višoj razini, općenitiji, generički tip, **nadtip**.



Slika 4.4 Generalizacijska apstrakcija

Djelomično slični tipovi entiteta su oni koji imaju jedan broj istih (zajedničkih) atributa, tipova veza s drugim entitetima i operacija.

Agregirani i generalizirani tip entiteta mogu se dalje agregirati i generalizirati u nove tipove entiteta.

5 METODA ENTITETI-VEZE

5.1 Uvod u metodu ENTITETI - VEZE

Metoda entiteti – veze poznata pod imenom model entiteti – veze (engl. *Entity-relationship model*, skraćeno ER) je prvi put objavljena u Chenovu članku (Chen 1976.). Metoda ER je jedan od prvih semantički bogatijih metoda za modeliranje podataka. ER model je opisan u većini knjiga koje se bave razvojem informacijskih sustava i bazama podataka i na gotovo svim svjetskim sveučilištima se uči ER modeliranje, a većina praktičara oblikovanja baze podataka u informacijskim središti ma koristi metodu ER.

Dr. Peter Pin-Shan Chen je američki znanstvenik koji se bavi informatikom i profesor je informatike na fakultetu Louisiana State University (wikipedia, 2010.). Poznat je po tome što je razvio metodu entiteti – veze.

Peter Chen je rođen na Tajvanu gdje je završio studij elektrotehnike. Dobiva stipendiju za Harvard i odlazi na postdiplomski studij računarstva. U Las Vegasu je 1975. godine na “1st International Conference on Very Large Databases” predstavio prvi rad na temu ER metode. Ovaj rad smatra se jednim od najutjecajnijih radova u informatičkoj povijesti.

Reagirajući na Chenov rad Codd (utemeljitelj relacijske metode) šalje dugo pismo uredniku ACM Transaction on Database Systems-a i kritizira velikim zamjerkama na ER metodu. Codd je 1979. godine predložio novu verziju relacijskoga modela u koju je uveo neke koncepte preuzete iz ER modela. Tijekom devedesetih godina su Codd i “Data Consulting Group” nekoliko puta pozvali Chena da, uz Codda, bude glavni govornik na njihovim simpozijima u Londonu što dokazuje opću prihvaćenost ER modela. Peter Chen je redovito glavni govornik na raznim međunarodnim konferencijama.



Slika 5.1 Peter Pin-Shan Chen

Petera Chena su mnogi pitali kako je došao na ideju za ER model. Promišljajući o tome, došao je do zaključka da je njegovo kinesko podrijetlo imalo utjecaja. I kineski znakovi i ER model pokušavaju modelirati svijet, koristeći grafičke simbole za predstavljanje entiteta iz stvarnoga svijeta.

Metoda entiteti - veze (skraćeno EV) je grafički prikaz međusobno povezanih grupa podataka promatranoga sustava. EV je semantički bogata metoda za modeliranje podataka jer raspolaže ljudski bliskim konceptima. EV se odlikuje prirodnošću opisa a njezini koncepti su bliski korisniku, pa je shema modela podataka laka za razumijevanje i komunikaciju korisnika i projektanta.

Kao praktično sredstvo EV model podataka je uskoro dobio niz komercijalnih softverskih produkata DBMS (*Database Management System*) (Chaing i Bargeron, 1980.; Iossifidis, 1980.; Munz, 1980.; Esculier i Glorieux, 1980.; Zhang i Mendelzon, 1983.; Nakata i Yamazaki, 1983.; Ford, 1985.; Goldberg, 1986.; Junet, 1987., i drugi).

Razvojem EV-a nastalo je niz varijanti polazne metode i ona je postala jedna od najčešće korištenih metoda. Tako se EV koristi u specijaliziranim metodologijama: CASE*Method, MIRIS, SSADM, IEM, i dr.

Za modeliranje podataka postoje i često korištene standardizirane notacije kao što su ISO UML (class diagrams) i IEEE IDEF1x.

Ovdje je opisana jedna varijanta metode EV-a korištena u specijaliziranoj metodologiji MIRIS.

Kako bi se metoda EV prepoznala u različitim specijaliziranim metodologijama, dana je tablica usporedbi grafičkih simbola u raznim metodama (vidi tablicu 5.53 i 5.54).

Metoda EV i pomoću nje izgrađen model DEV služi za izgradnju najmanje dvaju dijelova informacijskoga sustava i to: shemu baze podataka i arhitekturu programskoga proizvoda.

Procesom modeliranja podataka metodom EV dolazi se do modela podataka u kome su podaci u 5. normalnoj formi bez mukotrpнoga procesa normalizacije te je ova metoda značajna za projektiranje baze podataka.

5.2 Koncepti strukture metode entiteti - veze

Osnovni koncepti metode entiteti - veze od kojih se gradi struktura modela entiteti - veze su:

- Entitet i tip entiteta
- Veza i tip veze
- Atribut tipa entiteta
- Slab tip entiteta i specijalni tipovi veza

- Agregirani tip entiteta
- Povratni tip veze
- Generalizacijski tip veze

Model podataka metodom EV gradi se upotrebom grafičkih simbola. Osnovni simboli za gradnju DEV-a dani su na slici 5.2.

KONCEPT	SIMBOL	PRIMJER
TIP ENTITETA		
SLAB TIP ENTITETA		
TIP VEZE	 ili 	
ATRIBUT		
AGREGACIJA		
POVRATNA VEZA		
GENERALIZACIJA		

Slika 5.2 Grafički oblik koncepta strukture EV metode

Dijagram strukture modela naziva se **dijagram entiteta i veza**, skraćeno: DEV, EV dijagram, shema EV (engl. *Entity - Relationship Diagram, skraćeno ERD*).

DEV je grafički prikaz modela podataka sustava, metodom EV.

Podmodel je dio cjelokupnoga modela podataka prikazan na jednoj od niza slika DEV-a, a sadrži odabrane tipove entiteta, tipove veza i atributе jedne kohezivne cjeline.

Na primjer: podmodel "Nabava" je dio modela podataka "Komercijalno poslovanje", a sadrži tipove entiteta kao što su: Zahtjev za nabavu, Narudžbenica dobavljaču, Roba, Dobavljači i dr. Uočimo da se tip entiteta Roba nalazi i u podmodelu Prodaja.

Kako bi se pravilno gradio DEV, definiran je skup zabrana i pravila kombiniranja koncepata. Ovi skupovi pravila detaljnije određuju semantiku koncepata strukture DEV-a.

Postoji više verzija simbola za izgradnju DEV-a, ali semantika koncepata je osnova koja naoko različite modele svrstava u DEV. Pored osnovnih koncepata koji imaju grafičke simbole, EV metoda ima i niz koncepata koji se definiraju u okviru metode, ali se ne crtaju u modelu. Koncepte DEV-a razvrstavamo i definiramo u okviru niza sljedećih grupa pojmoveva.

ENTITET

- Entitet (engl. *entity*)
- Tip entiteta (engl. *entity type*)
- Uzorkovanje (engl. *sampling*)
- Klasifikacija (engl. *classification*)

VEZA

- Veza (asocijacija, engl. *relationship*)
- Tip veze (engl. *relationship type*)
- Uloga (engl. *role*)

ATRIBUT

- Vrijednost (engl. *value*)
- Nul vrijednost (engl. *null value*)
- Tip vrijednosti (engl. *value type*)
- Atribut tipa entiteta (osobina, engl. *entity type attribute*)
- Složeni atribut (engl. *compound attribute*)
- Ključ tipa entiteta (engl. *entity type key*)
- Primarni (glavni) ključ (engl. *primary key*)

SLAB ENTITET I SPECIJALNE VEZE

- Tip slabog entiteta - tip jakog entiteta (engl. *Weak entity type – strong entity type*) (W)
- Egzistencijalni tip veze (E) (engl. *Existential relationship type*)
- Identifikacijski tip veze (I)
- Egzistencijalni i identifikacijski tip veze (E&I)
- Tip veze prethodenja (P)
- Ključ slaboga tipa entiteta (engl. *Weak entity type key*).

GENERALIZACIJA

- Tip entiteta podtip i tip entiteta nadtip (engl. *Child & Parent entity type*)
- Generalizacijski tip veze (ovisnost podtipa, S, IS, JE) (engl. *Generalized realtionship type*)
- Hjerarhijski Tip Veze (H) (engl. *Hierarchical realtionship type*)
- Operacije i veze podtipa i nadtipa (engl. *Operations and relationships between parent and child*)

AGREGACIJA

- Agregirani tip entiteta (mješoviti tip entiteta) (engl. *Aggregate Entity Type*)
- Ključ tipa agregacije (engl. *Aggregate Entity Type Key*)

5.3 Entitet

Osnovni pojam u metodi EV je entitet. Uz pomoć niza definicija razjasnimo značenje tога pojma.

Entitet (engl. *entity*) je pojam u poslovanju o kome se zahtijeva čuvanje nekih podataka.

Entitet je stvar koja ima stvarno ili pojedinačno postojanje u stvarnosti ili mislima (Neufeldt, 1991.).

Entitet je nešto što ima realnost i svojstvenost (osobenost) u fizičkome postojanju ili mislima (Senko M.E.).

Entitet je nešto (bilo što) pojedinačno što se može jednoznačno imenovati. Imenovanjem stvari definiran je i izdvojen entitet iz ostalog skupa različitih pojmoveva.

Entitet može biti stvaran predmet, događaj, transakcija, osoba, apstraktni pojam, dokument i nešto drugo. Na primjer: Entitet je naše Sunce, entitet je čovjek s imenom Mato Hodak rođen 26.03.1930. u Rakovici sa stanom u Rijeci. Razni entiteti

su: mjesec, pojedini auto, ubrani cvijet, današnji dan, račun, narudžbenica, pojedini kupac, konkretni PC, seminar o modeliranju podataka, recept, prvi semestar studija, ispitni rok, porinuće broda i dr. Neki entiteti su prikazani na slici 5.3.



Slika 5.3 Razni entiteti

Slični entiteti se sa stajališta zajedničkih svojstava klasificiraju (kategoriziraju) u tipove entiteta. **Klasifikacija** je apstrakcija u kojoj se slični entiteti proglašavaju zajedničkim skupom nazvanim Tip entiteta. Klasifikacija je naziv za kategorizaciju¹⁵ na polju modeliranja podataka. Klasifikaciju kao znanstvenu metodu koriste brojne druge znanosti (biologija, bibliografija, ...).

Klasifikacija, razvrstavanje na temelju nekoga sustava; podjela predmeta ili pojmoveva na razrede, skupine, rodove, vrste, tipove, prema određenim načelima koja ovise o vrsti građe i svrsi klasifikacije (CARNet, 2010.).

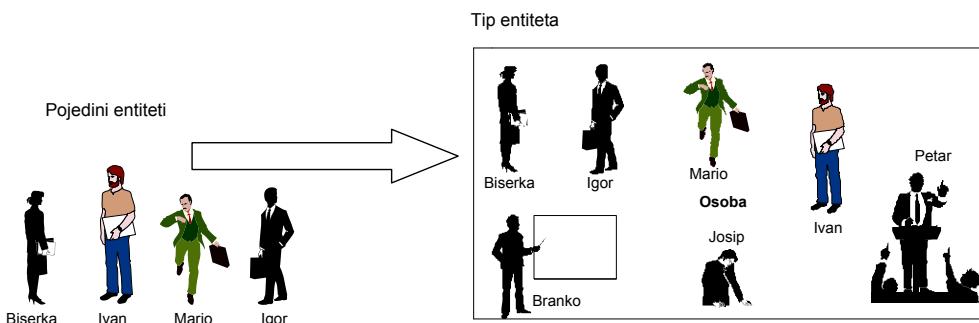
¹⁵ Kategorizacija je proces u kome se objekti i ideje prepoznaju, razlikuju i razumijevaju. Kategorizacija implicira objekte grupirane u kategorije, obično za neke specifične svrhe. U idealnome slučaju, kategoriju osvjetljava odnos između subjekata i objekata znanja. Kategorizacija je osnova u jeziku, predviđanjima, zaključivanjima, odlučivanjima i to u svim mogućim interakcijama s okolinom (Wikipedia, 2010.).

Definirajmo **sličnost** entiteta. Dva entiteta su slična ako imaju iste osobine (svojstva, attribute), ako mogu učestvovati u istim vezama (biti u odnosu s drugim entitetima) i ako se nad njima mogu izvesti iste operacije (aktivnosti koje izvode entiteti ili se provode nad entitetima).

Na primjer osobe: Biserka, Igor, Perino, Veljko i dr. su slični entiteti jer svi imaju osobine i to: ime osobe, datum rođenja, visinu, mjesto rođenja, brzinu trčanja na 100 metara, i dr., stoga možemo te osobe klasificirati u tip entiteta Osoba (vidi sl. 5.4.).

Svi entiteti istoga tipa imaju iste osobine.

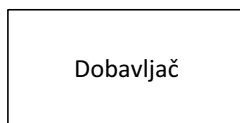
Tip entiteta (engl. *entity type*) je skup sličnih pojedinačnih entiteta dobiven procesom klasifikacijske apstrakcije. To je skup entiteta istoga tipa.



Slika 5.4. Klasifikacija entiteta u Tip entiteta Osoba

Na primjer: Radnik, Organizacijska jedinica, Dobavljač, Dio, Prekršaj su tipovi entiteta.

Tip entiteta grafički se predstavlja pravokutnikom proizvoljnih dimenzija (vidi sliku 5.5.).



Slika 5.5 Naziv tipa entiteta upisuje se u pravokutnik

Jedan tip entiteta sastoji se od entiteta. Entitete preciznije nazivamo pojedinačno pojavljivanja (pojedinačnim pojavama) tipa entiteta. Pojedinačna pojavljivanja entiteta ne crtaju se u modelu i ne prikazuju, ona se podrazumijevaju. Svaki tip entiteta ima jedno ili više pojedinačnih pojavljivanja.

Postoji razlika između pojma klasa entiteta i tip entiteta.

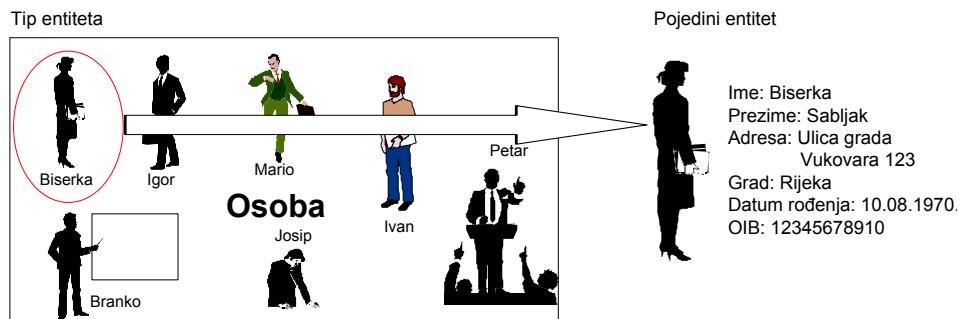
Klasa entiteta je skup pojedinačnih pojavljivanja entiteta.

Tip entiteta je imenovana klasa entiteta.

Klasa sadrži pojavljivanja. Tip je predstavnik klase na modelu podataka.

Praktičari, uobičajeno, ne prave razliku između klase i tipa, odnosno često i između entiteta i tipa entiteta. Tako se govori entitet, radi jednostavnije komunikacije a misli na tip entiteta. Poželjno je precizno definirati pojmove i komunicirati ispravnim terminima.

Uzorkovanje je apstrakcija u kojoj iz odabranoga tipa entiteta izdvajamo jedno konkretno pojedinačno pojavljivanje entiteta i promatramo ga detaljnije. Tako je na sl. 5.6 prikazano uzorkovanje entiteta „Biserka“ iz tipa entiteta Osoba.



Slika 5.6. Uzorkovanje entiteta „Biserka“ iz tipa entiteta Osoba

Uzorkovanje je suprotna apstrakcija od klasifikacije.

Tipove entiteta možemo klasificirati u stvarne i apstraktne. Vidi tablicu 5.1.

Tablica 5.1 Primjeri stvarnih i apstraktnej tipova entiteta

Stvarni tip entiteta	Apstraktni tip entiteta
Kupac	Aktivnost
Vozilo	Seminar
Roba	Projekt

Jedan entitet može biti član više različitih tipova entiteta. Na primjer, entitet Fakultet organizacije i informatike je član tipa entiteta Škola ako ga u bazi podataka želimo promatrati kao obrazovnu ustanovu, a isto tako je i član tipa entiteta Partner ako sudjeluje u poslovnim odnosima. Isto tako on ne mora biti član niti jednog tipa entiteta ako tvrtka za njegovo praćenje nije zainteresirana ili iz navedenih dvaju tipova entiteta može biti samo u jednom, ako se ne želi imati redundancu podataka.

Zadatak: Pokušajte definirati nekoliko stvarnih i nekoliko apstraktnih tipova entiteta. Za svaki tip entiteta uočite nekoliko njegovih pojavljivanja.

5.4 Ograničenja metode entiteti - veze

Koncepti ograničenja su znanja koja detaljnije opisuju odnose među podacima. Ona se uvode radi očuvanja integriteta podataka u modelu podataka.

Metoda entiteti-veze propisuje niz ograničenja.

Ograničenja se posebno upisuju na DEV i detaljnije tumače model podataka.

5.4.1 Posebna ograničenja

Postoje sljedeće vrste posebnih ograničenja:

- ograničenje na dopuštene vrijednosti domene
- ograničenje na dopuštene vrijednosti atributa (atribut ne smije primiti sve moguće vrijednosti iz domene)
- ograničenje na agregiranu vrijednost jednog ili više atributa po nekoj definiciji (izvedeno ograničenje)
- ograničenje na vrstu preslikavanja (brojnost) između tipa entiteta (agregacija) i vrijednosti atributa
- ograničenje na vrstu preslikavanja (brojnost) između tipova entiteta.

Pored ovih posebno definiranih ograničenja postoje i ograničenja definirana konceptima strukture EV modela, kao:

- identifikacija entiteta
- slab tip entiteta
- specijalni tipovi veze

U posebnim ograničenjima istaknimo ograničenja u pogledu tipova veza na primjer ekskluzivnost (vjeridba ili brak) ili prethođenja (planiran, ponuđen, ugovoren, izvršen, primljen, isplaćen) i sl.

Ograničenja su detaljnije opisana uz pojedine koncepte strukture metode EV-a.

5.5 Tip vrijednosti i njegova ograničenje

Vrijednost (engl. *value*) je podatak, element, elementarni podatak. Podatak je detaljno opisan na početku knjige.

Primjer vrijednosti: plav, 5, 10.08.1986., Mara Palijan, Ulica grada Vukovara 123, žena, Croatia banka, i dr.

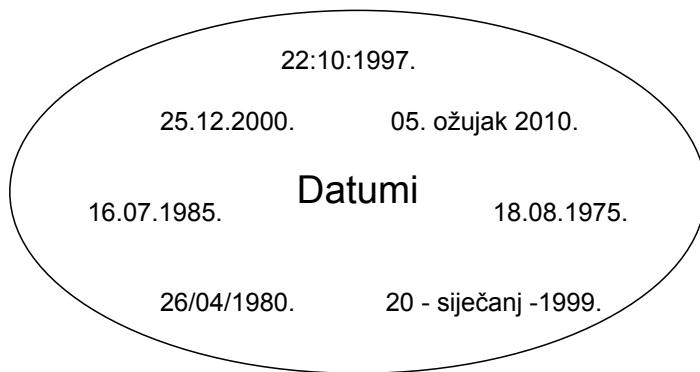
Vrijednost je osnovni pojam, tj. atomski iznos karakteristike određenog entiteta.

Jedno od ograničenja koje se može postaviti na podatke je **dužina zapisa podataka** u broju znakova. Tako na primjer Ime osobe može biti dugo najviše do 13 znakova.

Tip vrijednosti (u matematici domena, engl. *value set*, engl. *domain of values*) je skup pojedinačnih vrijednosti.

Na primjer: Boje, Ocjene, Datumi rođenja, Imena, Prezimena, Nazivi država, Nazivi organizacijskih jedinica, Školske ustanove.

Tip vrijednosti grafički se predstavlja krugom (ili ovalom) proizvoljnih dimenzija (vidi sliku 5.7).



Slika 5.7 Naziv tipa vrijednosti upisuje se u krug ili oval

Tip podataka (engl. *Data Type*) je oznaka vrste podataka u tipu vrijednosti. Mogući su tipovi podataka kao: datum, slova (engl. *alphabetic, latin letters*), znakovi, brojevi (engl. *Numeric, Arabic digits*), Alfanumerik (engl. *Alphanumeric*, kombinacija slova i brojeva). Svaki tip vrijednosti sadrži podatke jednoga tipa. Tako je tip podatka „Slova“ za vrijednosti u tipu vrijednosti Ime, a „Datum“ za tip vrijednosti Datumi.

Ako za tip vrijednosti odredimo kojeg su tipa podaci u njemu, nametnuto je ograničenje na tip vrijednosti.

Ograničenja na dopuštene vrijednosti u tipu vrijednosti mogu se definirati kombinirajući sljedeće vrste ograničenja:

- eksplisitnim navođenjem svih dopuštenih vrijednosti
- definiranjem intervala (jednog ili više) iz kojih se vrijednosti mogu, ali se ne moraju pojaviti
- određivanjem tipa podataka koji su dopušteni u tipu vrijednosti
- određivanjem minimalne i maksimalne dužine zapisa podataka
- određivanjem oblika zapisivanja podataka

Na primjer: domena Datumi događaja u životopisu osobe je skup datuma (određen je tip podatka) koji je jednak ili veći od datuma rođenja, a manji ili jednak od današnjega datuma (određen je interval $30.07.1959. \leq \text{datum} \leq \text{današnji datum}$) pri čemu datum mora biti u obliku DD.MM.GGGG. (DD. je kratica za dan, MM je kratica za mjesec, a GGGG je kratica za godinu).

5.6 Atribut

Atribut tipa entiteta (f) je funkcija koja preslikava tip entiteta (E) u tip vrijednosti (Vi) ili u kartezijev produkt tipova vrijednosti

$$(Vi_1 \times Vi_2 \times \dots \times Vi_n) \text{ za } 1 \leq i \leq n$$

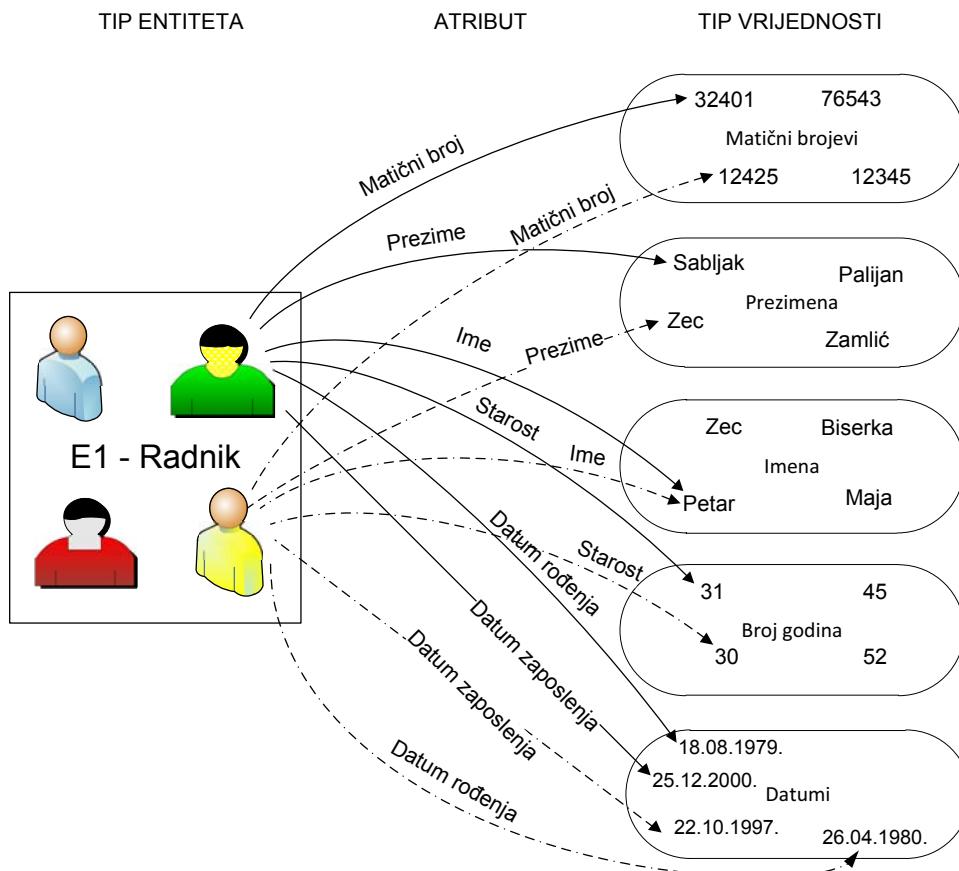
$$f: E \rightarrow Vi \text{ ili } (Vi_1 \times Vi_2 \times \dots \times Vi_n).$$

Ovo možemo opisati: Tip entiteta sastoji se od entiteta, a svaki entitet ima niz svojstava (karakteristika, odlika). **Atribut** je imenovana karakteristika (svojstvo) nekog entiteta. Ovo svojstvo je imenovana funkcija koja pridružuje tom entitetu neku vrijednost iz tipa vrijednosti. Tip vrijednosti povezan je s tipom entiteta preko atributa (vidi sl. 5.8).

Na primjer, atributi su: boja, datum rođenja, naziv organizacijske jedinice, ime radnika, naziv djela, šifra dobavljača i slično.

Atribut daje tipu vrijednosti interpretaciju preko pojavljivanja entiteta u tipu entiteta, odnosno prevodi podatak u informaciju. Precizno tumačenje razlike podatka i informacije moguće je EV metodom. Podatak je vrijednost u tipu vrijednosti. Informacija je model koji sadrži tip entiteta i tip vrijednosti povezani atributom. Tako je 31 podatak. A atribut Starost radniku pridružuje 31 godinu iz tipa vrijednosti Broj godina. Isto tako iz slike vidimo da je to starost Petra Sabljaka.

Atribut Prezime pridružuje jedno pojavljivanje tipa entiteta Radnik jednome podatu iz tipa vrijednosti Prezimena (vidi sl. 5.8). Atribut izvodi i suprotan pridruživanja te pridružuje jedan podatak iz tipa vrijednosti, na primjer prezime Sabljak, jednome pojavljivanju u tipu entiteta Radnik.



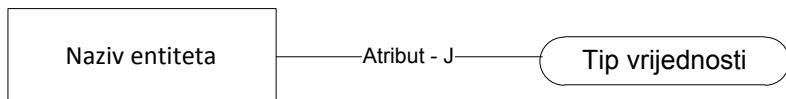
Slika 5.8 Grafički prikaz semantike koncepata tip entiteta - atribut - tip vrijednosti

Jedna vrijednost iz tipa vrijednosti može biti povezana s više različitim entitetima iz istog tipa entiteta. To znači da na primjer isto prezime može imati više radnika.

Može postojati više različitih tipova vrijednosti s istim vrijednostima, a da im se ipak da originalno i posebno ime. U tome slučaju mora postojati opravdanost kreiranja dviju skupina podataka. Moguće je da se u kasnijim fazama pojave novi podaci i dođe do razlika u tipovima vrijednosti.

Moguće je da novi atribut pridružuje entitete nad postojećim tipom vrijednosti, na primjer tip vrijednosti Datum je odgovarajući za niz atributa kao: Datum rođenja, Datum raskida radnog odnosa, odnosno sve moguće datume. Moguće je neki od datuma izdvojiti u zaseban tip vrijednosti radi naglašavanja njegove specifičnosti i ograničenja na datume, na primjer Datum učestvovanja u II. svjetskom ratu, pri čemu taj datum može biti u granicama od 1941. do 1945. za našu zemlju, dok ostali datumski atributi ne moraju imati ograničenja.

Atribut tipa entiteta grafički se predstavlja imenovanom linijom između pravokutnika i kruga (vidi sliku 5.9).



Slika 5.9 Atribut pridružuje entitete vrijednostima i obrnuto

Dva različita tipa entiteta ne mogu imati isti atribut (pojava homonima).

Homonimi su riječi ista oblika, ali različita značenja spram neke druge riječi; istoznačnica (Anić, 2003.). Na primjer riječ „pas“ ima više značenja i to: 1. Pojas i 2. Životinja. Slično je i s riječima: struja, brzina, put, projekt, aplikacija.

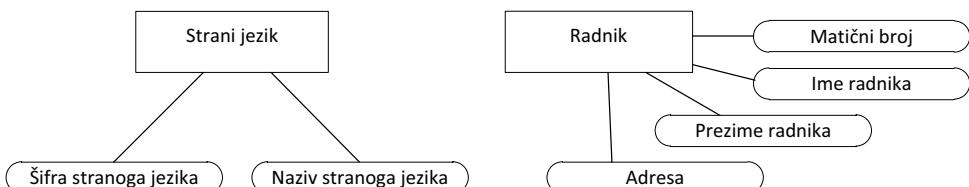
Primjeri homonima i njihova različita značenja:

- „bor“ – 1. vrsta četinjače, 2. kemijski element
- „bar“ – 1. jedinica za mjerenje tlaka, 2. vrsta lokala
- „list“ – 1. list papira, 2. vegetativni organ bilja, 3. vrsta ribe, 4. dio noge, 5. novine
- „korijen“ – korijen biljke, korijen zuba, korijen živca, korijen riječi, korijen problema, korijen zla, korijen jednadžbe, kvadratni korijen, n -ti korijen (u matematičari).

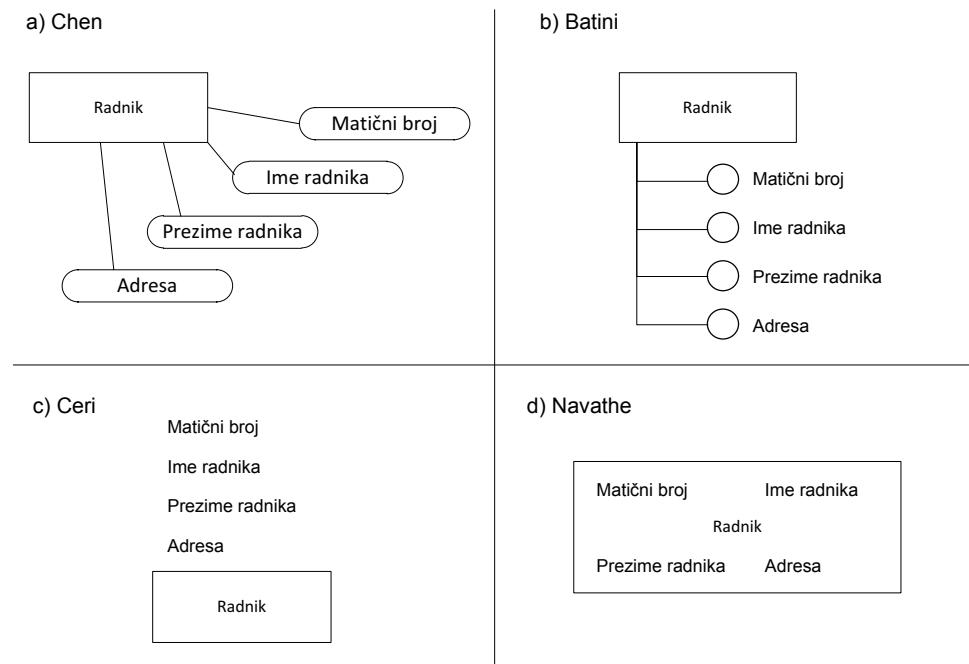
Potrebno je jedinstveno imenovati attribute u modelu podataka.

Više različitih atributa može pridružiti različitim entitetima istu vrijednost (bilo iz istog ili različitog tipa vrijednosti).

Naziv atributa može biti i uglavnom je isti kao naziv pridruženoga tipa vrijednosti. Ovo koristimo za **pojednostavljenje crtanja atributa** na DEV. Tada se na crtu koja spašava tip vrijednosti s tipom entiteta ne mora pisati naziv atributa, već se naziv atributa piše u oval za tip vrijednosti, a sam naziv tipa vrijednosti podrazumijeva se. Ovo pojednostavljenje se koristi jer se ne gubi previše informacija na DEV-u, a slike su jednostavnije.



Slika 5.10 Uobičajeni način crtanja atributa tipa entiteta na DEV-u



Slika 5.11 Tip entiteta i atributi prema: Chen, Batini, Ceri, Navathe

Na slici 5.11. prikazano je nekoliko načina crtanja atributa u DEV.

Zadatak: Nacrtajte nekoliko stvarnih i nekoliko apstraktnih tipova entiteta i njihove attribute.

5.6.1 Ograničenje na dopuštene vrijednosti atributa

Atributi entiteta uzimaju vrijednosti iz tipa vrijednosti. Može se postaviti ograničenje na ove vrijednosti, tako da atribut može poprimiti samo uži skup vrijednosti iz tipa vrijednosti. Na primjer: atribut Datum rođenja radnika je definiran nad tipom vrijednosti Datum ali prima vrijednosti u granicama od 01.01.1910. do 01.01.1972. Pored toga mogu se postaviti ograničenja na tip i dužinu podatka.

Ograničenje se može postaviti na vrijednost atributa preko vrijednosti drugog atributa. Na primjer: Datum rođenja radnika u bazi podataka mora biti uvijek manji od atributa Datum zaposlenja radnika.

5.6.2 Ključ tipa entiteta

Ključ tipa entiteta (identifikator entiteta, engl. *entity type key*) je takav skup atributa koji neovisno o vremenu zadovoljava uvjet jedinstvenosti i uvjet neredundantnosti.

U1: Uvjet jedinstvenosti

Ne postoje dva pojedinačna pojavljivanja entiteta u tipu entiteta takva da imaju istu vrijednost atributa koji čine ključ i ne postoje dva tipa entiteta koji imaju isti skup atributa za ključ.

Atribut Matični broj radnika je ključ tipa entiteta Radnik (vidi sl. 5.8). Jednomu radniku pripada jedan i samo jedan broj, a jedan broj pripada uvijek jednom i samo jednom radniku. Time je uvjet jedinstvenosti zadovoljen. Ostali atributi, na primjer Ime, ne zadovoljavaju taj uvjet.

U2: Uvjet neredundantnosti

Ako se iz ključa izostavi bilo koji od atributa, gornja osobina se gubi.

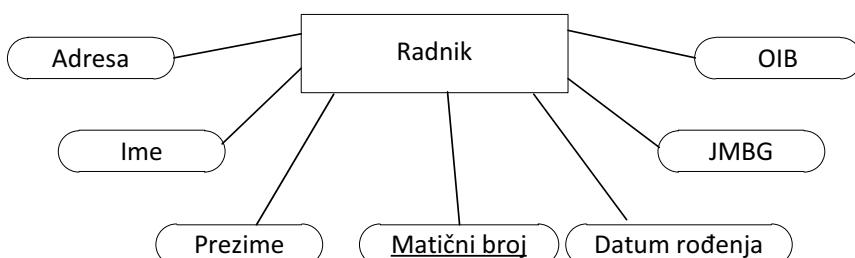
Ne postoji nijedan atribut kao dio ključa koji se može izostaviti iz ključa, a da se pritom uvjet jedinstvenosti ne gubi.

Na primjer: ako bi ključ Stavke robe na računu bio sastavljen od Šifre računa i Šifre robe koja se plaća tom stavkom na tome računu. Bez obiju spomenutih šifri ne može se jedinstveno identificirati o kojoj je robi na kome računu riječ te su potrebne obje u isto vrijeme.

Uvjet jedinstvenosti i neredundantnosti detaljnije su opisani u relacijskom modelu podataka.

Jedan tip entiteta može imati više atributa koji su kandidati za ključ jer zadovoljavaju gornja dva uvjeta. Koji će od njih biti odabran za ključ, određuje projektant koji kreira model podataka. Obično se bira jednostavniji i dostupniji atribut.

Na primjer: atributi Matični broj radnika, Jedinstveni matični broj građana i OIB su ključevi tipa entiteta Radnik. Za primarni ključ u modelu, a potom i bazi podataka o kadrovima, bira se na primjer, Matični broj radnika, tj. broj koji poslovna organizacija sam dodjeljuje svakome radniku (vidi sl. 5.12).



Slika 5.12 Primarni ključ Matični broj je podvučen

Primarni ključ (glavni ključ, engl. primary key) je jedan od ključeva tipa entiteta koji je odabran zbog nekoga značajnijeg razloga za zamjenika entiteta toga tipa (za ključ). Jedan od razloga je njegovo često korištenje. U EV metodi uvodimo samo jedan ključ za svaki tip entiteta i to primarni (glavni) ključ. Ovaj će ključni atribut postati osnova za fizičko zapisivanje podataka u bazu podataka.

Primarni ključ entiteta grafički se predstavlja jednim od triju načina:

- podylačenjem imena ključnog atributa linijom (kao u relacijskome modelu), i taj će standard biti korišten u ovoj knjizi
- dvostrukom linijom između tipa entiteta i tipa vrijednosti
- stavljanjem oznake “*” iza imena atributa, odnosno na neki način ga istaknemo od ostalih atributa

Ključ se može definirati i preko ograničenja preslikavanja što je i prikazano kod definicije ograničenja atributa.

Zadatak 1: Nadite tipove entiteta koji imaju nekoliko kandidata za ključ. Provjerite zadovoljavaju li uvjete za ključ.

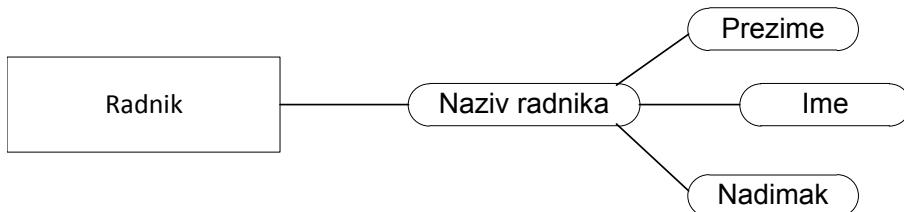
Zadatak 2: Nadite tipove entiteta koji imaju nekoliko atributa koji čine ključ.

5.6.3 Vrste atributa

Atribut može biti: atomski atribut, složeni atribut, viševrijednosni i izvedeni.

Složeni atribut (engl. *composite attribute*), ponekad nazvan grupni atribut je atribut spojen (agregiran) od više atributa.

Složeni atribut tipa entiteta je pogled na dva ili više atributa kao na jedan novi grupni atribut. Ovaj novi atribut dobije svoj naziv i vezu s njemu pripadajućim atributima. On se može grafički predstaviti kao na slici 5.13. povezivanjem atomskih atributa sa složenim atributom.



Slika 5.13 Grafički prikaz složenog atributa Naziv radnika

Obični atributi se u odnosu na složene attribute još nazivaju **jednostavniji** atributi (engl. *simple attribute*) ili **atomski** atributi (engl. *atomic attribute*).

U EV modelu može se izbjegći crtanje složenih atributa i prikaz samo atomskih atributa, a složeni natpojam može se definirati zasebnim iskazima ili potpuno izbjegći ako korišteni SUBP i korišteni softverski alat ne podržava definiranje složenih atributa.

Pojam složenog atributa razlikuje se od pojma složenoga ključa po tome što složeni atribut ne mora biti ključ tipa entiteta, a složeni ključ od dvaju i više atributa ne mora imati naziv grupe.

Zadatak: Nadite složeni atribut.

Nul vrijednost (engl. *null value*) je neodređena vrijednost atributa nekog entiteta.

Zanimljivo nam je uočiti dva tipa nul vrijednosti

1. trenutno nepoznata nul vrijednost
2. nul vrijednost zbog neprimjenjivog svojstva

Trenutno je nepoznata nul vrijednost dopuštena u EV modelu. Nul vrijednost zbog neprimjenjivoga svojstva treba izbjegavati, ali ako to nije moguće može ostati u modelu, a kasnije i u bazi, ali trebamo analizirati što to znači i priхватiti posljedice. Koji put posljedice nisu značajne.

Na primjer: atribut Adresa tipa entiteta Radnik u trenutku zapošljavanja radnika koji još nije trenutno našao smještaj, je nepoznata vrijednost. Već sutradan kada radnik nađe stan taj podatak će biti poznat, a to znači moguće ga je upisati s jednim danom zakašnjenja. Ovo nema značajniji utjecaj na integritet baze podataka.

Nul vrijednost atributa entiteta zbog neprimjenjivoga svojstva u EV-u se izbjegava (nije dopuštena) prilikom projektiranja modela podataka.

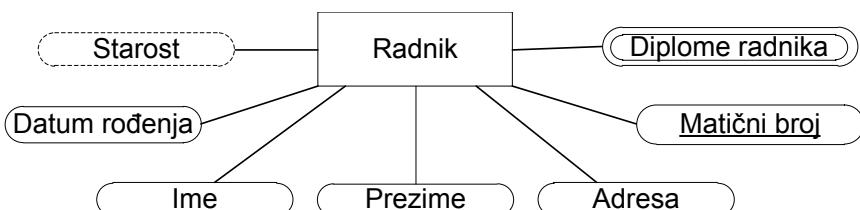
Na primjer, atribut Dubina gaza tipa entiteta Prijevozno sredstvo je primjenjiv za vodena vozila, ali nije svojstvo za: autobuse, avione i dr. nevodena vozila. Ako naša baza podataka sadrži pretežno (recimo 75%) brodove, onda će većina redaka entiteta imati popunjenu vrijednost toga atributa. Ali, ako naša baza podataka sadrži uglavnom automobile, a tu i tamo koji brod, onda će gotovo svi retci za taj atribut u bazi biti nepotpuni. Ako to ne smeta, onda to može tako ostati. Uklanjanje je takvih atributa moguće pomoći slabih tipova entiteta (objašnjeno kasnije).

Nul vrijednost je prikazana u tablici 5.2, i označava se simbolom „δ“ (moguće su i druge oznake kao: „Null“, „?“).

Atribut je viševrijednosan (engl. *multi-valued*) ako jednom tipu entiteta odgovara više vrijednosti iz tipa vrijednosti.

Tako neki Radnik može imati više diploma različitih škola, a ne samo jednu diplomu, kao što ima po jednu vrijednost svih drugih atributa (vidi sl. 5.14). Grafički se viševrijednosni atribut crta dvostrukim ovalom.

Viševrijednosni atribut se preciznije definira pomoći brojnosti preslikavanja kod ograničenja EV metode i predlaže se njegovo pretvaranje u tip entiteta.



Slika 5.14 Grafički prikaz više vrijednosnog atributa Diplome radnika

Atribut je izведен (agregirani atribut, engl. derived) ako se do njegovih vrijednosti može doći operacijama iz drugih atributa u bazi podataka.

Na slici 5.14 prikazan je izvedeni atribut Starost. Do podatka o starosti može se doći iz atributa Datum rođenja i današnjega datuma. Grafički se na DEV-u izvedeni atribut crta isprekidanom linijom.

Ograničenje se može postaviti na moguće vrijednosti izvedenog atributa. Takvo ograničenje se postavlja korištenjem matematičko – logičkih operacija nad osnovnim atributima. Na primjer: Atribut „Ukupne obustave“ na plaće radnika u nekom mjesecu ne smije biti veći od jedne trećine mjesecne plaće, ili atribut „Radni staž“ može biti manji ili jednak atributu „Starost“ umanjenom za 18.

Svaki složeni atribut je izведен spajanjem osnovnih atributa.

Relacija entiteta. Tip entiteta se može prezentirati pomoću tablice, a ovakav način prikaza tipa entiteta se naziva relacija entiteta ili tablica.

Relacija entiteta je slična relaciji u relacijskome modelu podataka. Dakle, možemo zaključiti da je tip entiteta sličan pojmu relacija ili tablica. Proširenjem ove definicije, opisujući što sve to može postati u trenutku implementacije na računalu, možemo definirati tip entiteta kao: skup podataka, datoteka, master datoteka, varijabilna datoteka, relacija i sl. To sve je približno točno, ali se ipak i međusobno razlikuje.

Relacija (tablica) koja nastaje od DEV-a prikazanoga na slici 5.14 je prikazana u tablici 5.2.

Svaki atribut postaje stupac tablice. Složeni atribut ne prikazuje se kao stupac, već se njegovim atomskim atributima u nazivu dodaje naziv složenog atributa. Izvedeni atribut ne postaje stupac, već se on u programu računa i prikazuje korisnicima.

Viševrijednosni atribut je prikazan u posljednjem stupcu. *D1, D2, ...* su različite diplome raznih škola. Neki radnik nema ni jednu diplomu a neki ima više diploma. Sve su diplome prikazane u jednom stupcu odvojene zarezom. Zbog toga je viševrijednosni atribut u modelu podataka nepoželjan. Tablica nije dobra (nije u poželjnim normalnim formama, opisanim kasnije) ako svaki redak ne sadrži samo jednu vrijednost nekog atributa.

Tablica 5.2 Tablica nastala prevođenjem tipa entiteta Radnik

MBR	Naziv.Prezime	Naziv.Ime	Adresa	Datum rođenja	Starost	Diploma
1001	Perić	Ivan	Pulska10	1948-11-26	45	D1
1002	Vukić	Mario	Zadars.3	1952-03-19	...	-
1003	Puh	Zlatko	Bana J.7	1959-09-13		D6,D8
1004	Marić	Gordan	Illica 7	1968-11-10		D2,D5,D6
1005	Žurić	Ana	“null”	1971-03-06		-
1006	Horvat	Branka	Velbits.9	1962-04-17		D5
1007	Ban	Josip	Korzo 8	1964-12-25		D6,D7

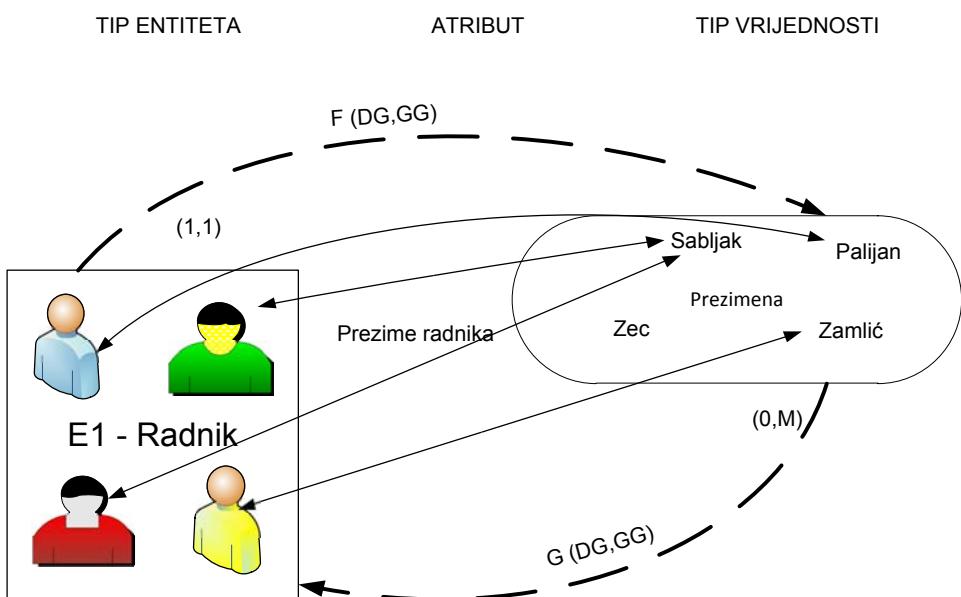
Zadatak: Nađite Tip entiteta i u njemu pronađite po jedan: atomski atribut, složeni atribut, izvedeni atribut i viševrijednosni atribut, nacrtajte ga kao DEV i prevedite u tablicu.

5.6.4 Brojnost preslikavanja atributa

Atribut je definiran kao povezivanje između entiteta tipa entiteta i vrijednosti tipa vrijednosti. Ovo povezivanje sastoji se od dvaju preslikavanja F i G, pri čemu je G inverzno preslikavanje od F. Na to preslikavanje možemo postaviti ograničenja (brojnost, kardinalnost, tip povezanosti). Ovo ograničenje možemo nazvati i ograničenje na vrstu preslikavanja između tipa entiteta i vrijednosti atributa, ili brojnost preslikavanja atributa (vidi sl. 5.15).

Atribut Prezime pridružuje jednom radniku iz Tipa entiteta Radnik jednu konkretnu vrijednost iz Tipa vrijednosti Prezimena. Obrnuto preslikavanje, jedna konkretna vrijednost iz Tipa vrijednosti Prezimena (dakle jedan podatak na primjer „Sabljak“) može biti pridružena jednom ili više radnika iz Tipa entiteta Radnik.

Atribut je povezivanje pojavljuvanja dvaju skupova koje se sastoјi iz dvaju preslikavanja različitog smjera od Tipa entiteta k Tipu vrijednosti (preslikavanje F) i obrnuto (preslikavanje G). Uvijek se možemo pitati kako su elementi tih skupova povezani.



5.15 Brojnost atributa Prezime radnika

Za svaki tip entiteta atribut toga tipa entiteta može pridružiti jednom pojavljujuću entitetu vrijednosti iz tipa vrijednosti u sljedećoj količini:

- 0 - nijedan
- 1 - jedan
- Br - cijeli prirodan broj
- M - mnogo (engl. *Many*), neodređeni cijeli prirodan broj, (0, 1, Br ili više)

Inverzno preslikavanje od pojedinačne vrijednosti iz tipa vrijednosti u tip entiteta može se preslikati na 0, 1, Br ili M pojedinačnih pojavljivanja entiteta.

Brojnost preslikavanja atributa su dva (za preslikavanje F i inverzno preslikavanje G) uređena para F(DG, GG) i G(DG, GG) pri čemu je DG donja granica, a GG gornja granica brojnosti. Uređen par predstavlja zapis dvaju elemenata u uglatim zagradama odvojena zarezom.

Za preslikavanje F možemo se pitati koliko će se najmanje (DG) i najviše (GG) pojedinačnih vrijednosti tipa vrijednosti preslikavati u jedno bilo koje proizvoljno odabранo pojavljivanje entiteta iz tipa entiteta? Odgovarajući na to pitanje dobivamo brojnost preslikavanja F.

Za preslikavanje G možemo se pitati koliko će se najmanje (DG) i najviše (GG) pojedinačnih entiteta tipa entiteta preslikavati u jedno bilo koje proizvoljno odabranu pojavljivanje vrijednosti iz tipa vrijednosti? Odgovarajući na to pitanje dobivamo brojnost preslikavanja G.

Opišimo proces određivanja i zapisivanja brojnosti preslikavanja atributa A za tip entiteta E i tip vrijednosti D.

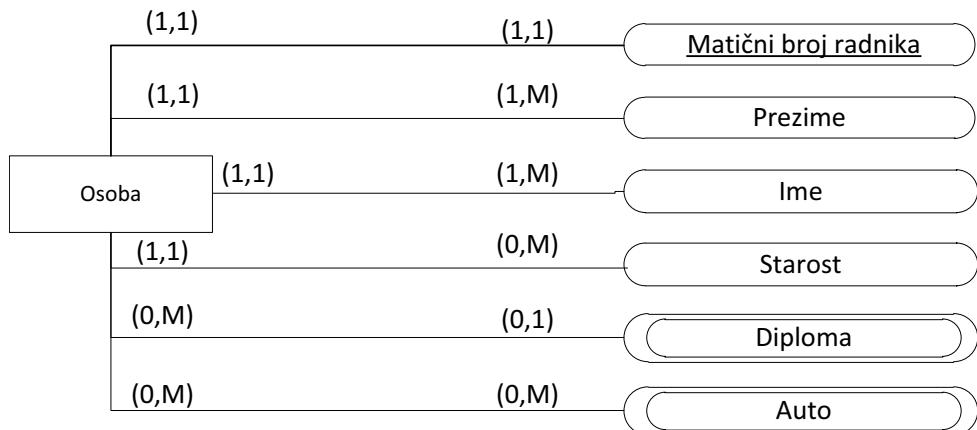
Do brojnosti atributa A dolazimo tako da uzmemo jedno (bilo koje konkretno e) pojavljivanje entiteta iz tipa entiteta E. Pitamo se; koliko je najmanje (donja granica) i najviše (gornja granica) vrijednosti iz D povezano s e iz E.? Odgovor je brojnost atributa A. Brojnost se upisuje na DEV kao na primjer (1,M) na liniji spoja između tipa entiteta i naziva atributa uz tip entiteta.

Do inverzne brojnosti atributa A dolazimo tako što uzmemo jednu vrijednost (bilo koju konkretnu x) iz tipa vrijednosti D. Pitamo se; koliko je najmanje (donja granica) i najviše (gornja granica) entiteta iz E povezano s jednom vrijednošću x iz tipa vrijednosti D.? Odgovor je inverzna brojnost atributa A. Brojnost se upisuje na DEV kao uređen par na liniji spoja između tipa entiteta i tipa vrijednosti uz tip vrijednosti.

Brojnost atributa je definirana s dva uređena para.

Na primjer: atribut za Starost tipa entiteta Radnik definiran nad domenom Godine (vidi sl. 5.8). Jedan bilo koji radnik (entitet) iz skupa svih Kadrova ima samo jednu vrijednost atributa Starost i upisuje se 1 kod simbola domene. Jedna bilo koja vrijednost iz domene je vrijednost atributa Starost, koja može istovremeno biti starost mnogo (M) radnika, i piše se M kod simbola za tip entiteta.

Pojednostavljenjem crtanja DEV-a, pri čemu se naziv atributa i naziv tipa vrijednosti izjednačavaju, brojnosti se crtaju na DEV kao na slici 5.16.



Slika 5.16 Atribut je preslikavanje između tipa entiteta i tipa vrijednost

Donje granice preslikavanja mogu biti 0 ili 1.

Gornje granice preslikavanja mogu biti 1 ili M.

Općenito preslikavanje između tipa vrijednosti i tipa entiteta može biti, prema razlikovanju gornje granice (u ovome razmatranju donja granica DG nije bitna):

- (DG,1):(DG,1) - jedan entitet ima najviše samo jednu vrijednost atributa, a jedna vrijednost pripada najviše samo jednom entitetu,
- (DG,1):(DG,M) - jedan entitet ima najviše samo jednu vrijednost atributa, a jedna vrijednost može biti vrijednost više od jednog entiteta,
- (DG,M):(DG,1) - jedan entitet ima za GG više vrijednost za jedan atribut (tzv. viševrijednosni atribut), a jedna vrijednost atributa je vrijednost samo jednog entiteta,
- (DG,M):(DG,M) - jedan entitet ima više vrijednosti za jedan atribut, a jedna vrijednost može biti vrijednost više entiteta.

Za stabilne je poslovne sustave uobičajeno da se brojnosti atributa ne mijenjaju tijekom vremena. Ukoliko promjena poslovnoga sustava uzrokuje promjenu modela te ipak mora doći do promijene brojnosti atributa, onda će ta promjena uzrokovati promijene u organizaciji baze podataka i promjene izvornoga koda aplikativne programske podrške.

Definirajmo ključ tipa entiteta pomoću brojnosti atributa.

Ključ tipa entiteta je atribut (ili skup atributa) tipa entiteta za koga vrijedi da je brojnost preslikavanja $F = (1,1)$ i $G = (1,1)$.

Odnosno, ključ tipa entiteta može biti samo onaj skup atributa koji ima svojstvo da je vrsta preslikavanja između tipa entiteta i tipa vrijednosti i obrnuto po DG i GG jednaka 1.

Na DEV-u se zahtijeva da svaki tip entiteta ima ključ. Svi tipovi entiteta u modelu podataka moraju imati određen ključ tipa entiteta. Svrha je toga doći do sheme relacijske baze podataka za koju se traži postojanje ključnih atributa za svaku tablicu.

Ako tip entiteta nema relevantnih ključnih atributa, ili je iz nekih razloga do njihovih vrijednosti otežano doći, uvodi se **Eksterni ključ**. **Eksterni ključ** (dodani ključ, vanjski ID, engl. surrogate key) je umjetno dodani identifikacijski atribut koji ima svojstva primarnoga ključnog atributa i koji će jedinstveno identificirati sve entitete. Eksterni se ključ često sakrije od korisnika.

5.7 Tip veze

Veza (asocijacija, povezivanje, engl. *relationship*) je pridruživanje između entiteta. Neka su e_1, e_2, \dots, e_n entiteti, tada je n -torka (e_1, e_2, \dots, e_n) veza entiteta e_1, e_2, \dots, e_n .

Uzmimo dva entiteta i to entitet e_1 iz tipa entiteta E_1 i e_2 iz tipa entiteta E_2 . Uređeni par (e_1, e_2) je pojedina veza pojedinih entiteta. Dva tipa entiteta mogu imati više pojedinačnih veza.

Veza predstavlja odnos koji postoji među entitetima bilo u stvarnosti, bilo u mislima. Entiteti se nalaze u različitim odnosima, a te odnose nazivamo veze.

Na primjer: "Otac Jure ima sina Franja"; veza "ima" povezuje entitete "Jure" i "Franjo". Veze su i "Radar isporučuje SPERRY" i "10.07.98. je porinuće broda "Panda".

Veza brak između entiteta Biserka i entiteta Igor je prikazana na sl.5.17. Vezu можемо zamisliti kao „uže“ koje povezuje dva entiteta i to s jedne strane Biserka drži uže u ruci, a s druge strane uže je zavezano Igoru oko vrata.

Dva tipa entiteta mogu imati više pojedinačnih veza. Tako dolazimo do pojma Tip veze.

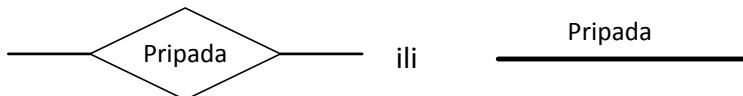
Tip veze (engl. *relationship type*) je skup veza između istih tipova entiteta.



Slika 5.17 Jedna veza "Brak Igora i Biserke" između muškarca "Igora" i žene "Biserke"

Neka su E_1, E_2 tipovi entiteta i e_1, e_2 neki entiteti tih tipova entiteta. Tada je tip veze skup uređenih parova (e_1, e_2) takvih da je $e_1 \in E_1$ i $e_2 \in E_2$, pri čemu E_1 i E_2 ne moraju biti različiti.

Tip se veze grafički predstavlja rombom ili linijom proizvoljnih dimenzija. Naziv tipa veze se upisuje u romb ili na liniju.

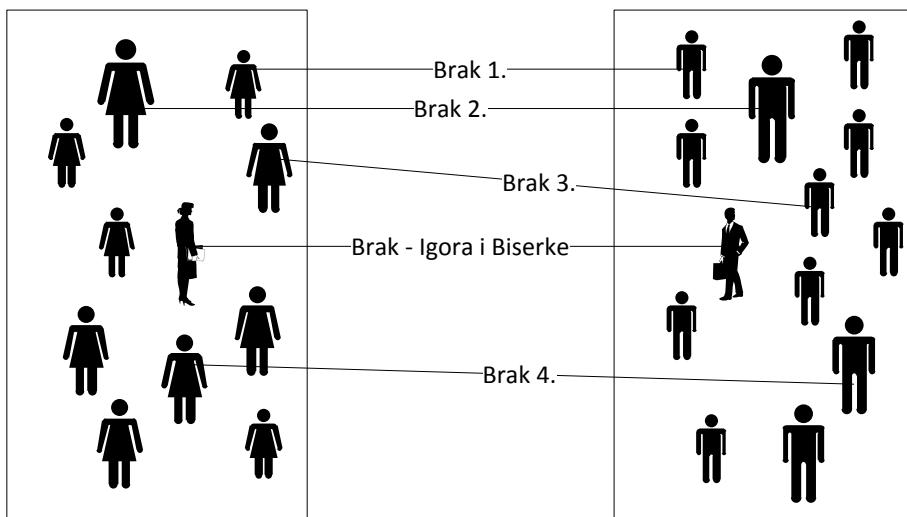


Slika 5.18 Simboli za tipa veze su romb s linijama ili samo linija

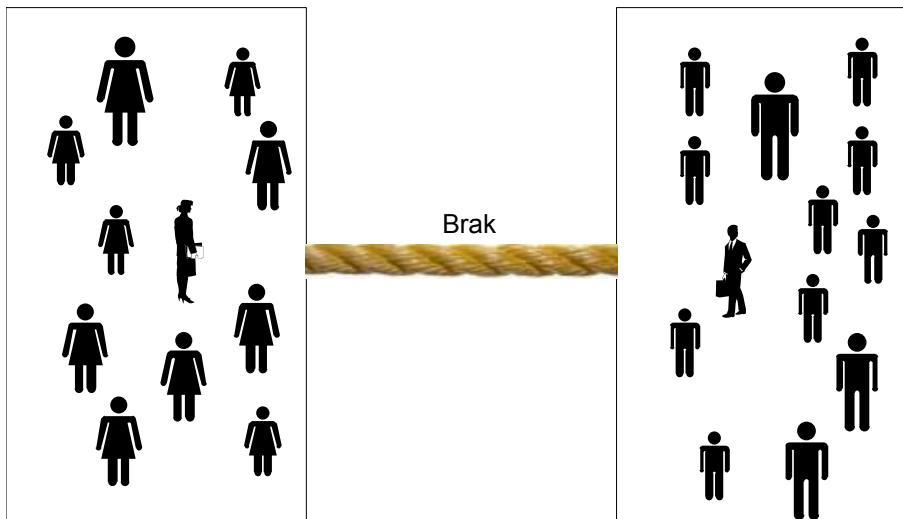
Na primjer, tipovi veza su: ima, isporučuje, porinuće, studira, posjeduje.

Tip veze je skup pojedinačnih veza. Na slici 5.19 je prikazan tip veze "Brak" između muškaraca i žena kao skup pojedinačnih brakova muškaraca i žena. Na modelu podataka ne crtaju se pojedinačne veze, već se sve pojedinačne veze klasificiraju u jednu klasu veza, odnosno u tip veze.

Na slici 5.20. prikazan je uobičajeni dijagram. Uočimo da se tip veze crta kao linija koja spaja tipove entiteta, a ne i pojavljuvanja tipa entiteta. Linija Brak je suma linija pojedinih brakova.



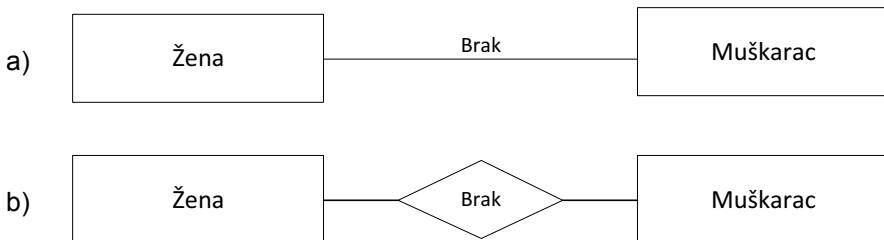
Slika 5.19 Pojedinačna veza brak postoji između pojedinačnih entiteta



Slika 5.20 Prikaz veze između dva tipa entiteta

Veza se još naziva i jedno pojedinačno pojavljivanje tipa veze. Time se ističe postojanje jednog u odnosu na cjelinu.

Tip veze je opća struktura povezivanja tipova entiteta (uže s puno niti). Veza je pojedinačna struktura povezivanja pojedinačnih entiteta (jedna nit užeta).

Slika 5.21 Uobičajeni prikazi tipova entiteta i tipa veze
a) kao linije i b) kao romba.

Veza i tip veze je koncept koji spaja ostale koncepte entitete i tipove entiteta. Veza nema atributte. Ona je spoj stvari bez osobina.

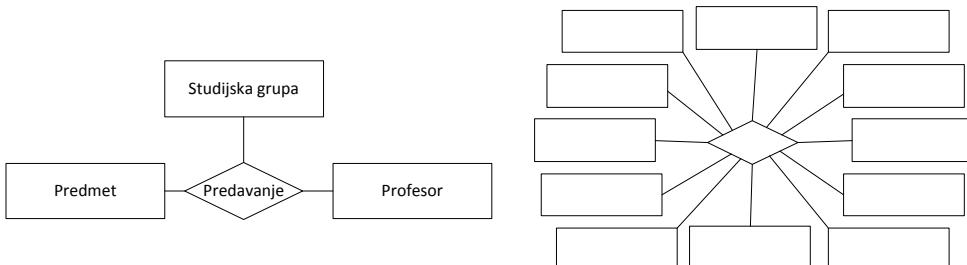
Red tipa veze je broj tipova entiteta povezanih tim tipom veze.

Tip veze čiji je red binaran, povezuje pojavljivanja dvaju tipova entiteta. Tip veze ne smije biti povezan samo s jedne strane. Takva veza nije moguća i nema smisla. Veza uvijek ima dva kraja. U našemu razmatranju prikazivali smo samo binarne veze. Tip veze je **binaran** ako spaja dva tipa entiteta.

Moguće su i višestruke veze. U teoriji modeliranja podataka i u različitim variantama metode EV-a dopuštene su višestruke veze. *One su u ovoj verziji modela podataka, koja se prezentira u ovoj knjizi, nepoželjne.* Razlog tome je otežano prevođenje u relacijsku shemu baze podataka.

Trostruki tip veze (reda 3) povezuje istodobno tri tipa entiteta. Prikazan je tip trostrukih veza Predavanje koji povezuje tipove entiteta: Predmet, Profesor i Studijska grupa (vidi sl. 5.22).

N-arni tip veze povezuje n tipova entiteta, pri čemu je n bilo koji prirodan broj veći od jedan.



Slika 5.22 Tip trostrukih veza Predavanje i n-arni tip veze

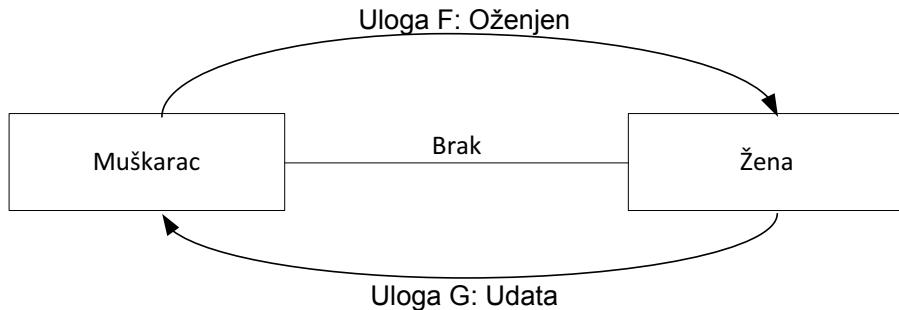
Naziv tipa veze je riječ ili grupa riječi koji predstavljaju radnju (glagoli, glagolske imenice), operaciju ili odnos među entitetima (imenice), a koji nam jasno prikazuje u kom odnosu su tipovi entiteta.

Naziv tipa veze ne mora biti jedinstven na DEV-u, ali ne smije postojati još jedan tip veze između istih tipova entiteta s istim imenom.

Kako čitamo DEV? Tip veze uvijek čitamo formirajući rečenicu uz obavezno čitanje naziva tipova entiteta i to tako da krenemo od jednoga tipa entiteta pa preko naziva tipa veze i na kraju čitamo naziv drugoga tipa entiteta, a pritom dodajemo potrebne veznike. Tako na primjer iz „Žena Brak Muškarac“ na sl. 5.21 čitamo DEV kao „Žena je u Braku s Muškarcem“. Ali vrijedi i obrnuto čitanje „Muškarac je u Braku sa Ženom“.

Koji put je ime veze takvo da se logično čita samo s jedne strane, od jednoga tipa entiteta, a za čitanje s druge strane potreban je novi naziv jer je odnos takav da taj suprotan smjer ima poseban termin u jeziku. Na primjer suprotna čitanja tipova veza su: Račun Sadrži Stavke – Stavka Se nalazi na Računu, Radnik Vodi Automobil, Automobil Pripada Radniku, Dijete ima Roditelja, Roditelj Ima Dijete.

Svaki tip veze ima samo jedan naziv koji ima dvije uloge. Pored jednoga naziva tipa veze pogodno je definirati ulogu tipa entiteta u tipu veze. Tako je uloga muškarca u braku „Oženjen“, a uloga žene u braku „Udana“ (vidi sliku 5.23.).



Slika 5.23 Primjer uloge tipa veze Brak

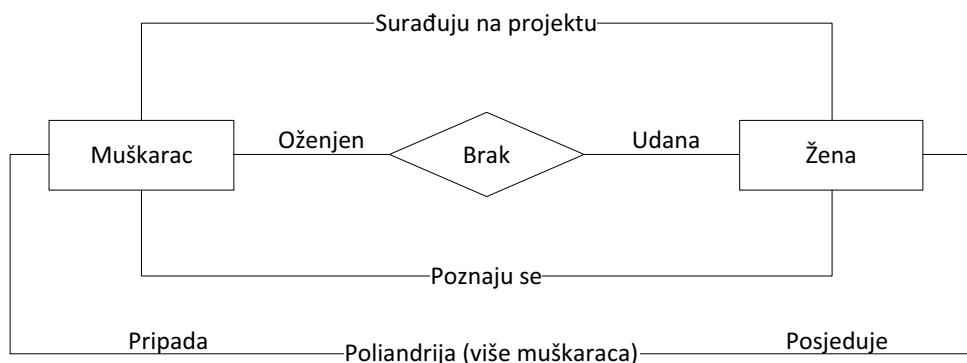
Uloga (rola, engl. role) entiteta u vezi je imenovana funkcija (r) koja pridružuje taj entitet nekome drugom entitetu. U uređenome paru neke veze (e_1, e_2) svaki entitet ima neku ulogu r_1 i r_2 , respektivno, pa se veza označava sa $(r_1/e_1, r_2/e_2)$.

Uloga je imenovano opravdanje jednog tipa entiteta u tipu veze. Ulogu još možemo nazvati i značenje veze s jedne strane, odnosno naziv funkcije koja jedan tip entiteta preslikava u drugi tip entiteta.

Uloga je naziv za razlog participacija tipa entiteta u tipu veze.

Ako se veza prikazuje rombom, onda se uloga grafički predstavlja crtom koja spaja tip entiteta i tip veze (vidi sl. 5.24). Uloga se imenuje i upisuje na DEV. Ona se obavezno upisuje kod rekurzivnih tipova veza, odnosno tipova veza koje povezuju entitete istog tipa entiteta, opisanih kasnije.

Uloga tipa entiteta u tipu veze obilježava se pisanjem imena na liniju između pravokutnika i romba, odnosno na liniju veze uz tip entiteta.



Slika 5.24 Primjer imenovanja uloga "oženjen" i "udana" tipa veze Brak.

Na slici 5.24 su prikazani različiti tipovi veza između dvaju tipova entiteta. Između dvaju tipova entiteta je moguće postojanje nijednog, jednog, dva ili više tipova veze.

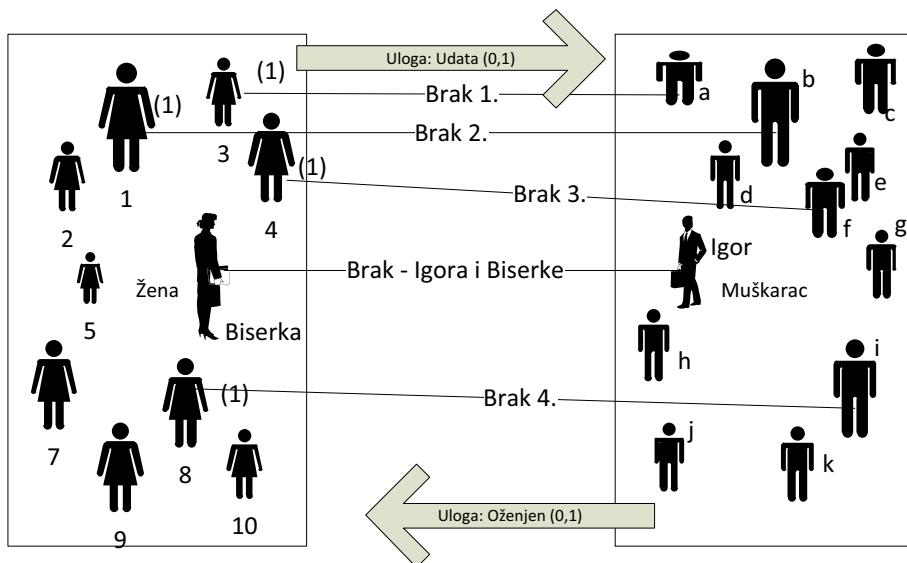
Zadatak: Ustanovite postojanje i drugih tipova veza (poligamije, ...) i imenujte uloge tih tipova veza.

Ključ tipa veze ne postoji. Tip veze koji izlažemo u ovoj varijanti EV metode, po konvenciji, nema atributa pa prema tome nema ni ključ tipa veze.

Tip veze koji treba imati attribute nije tip veze već je po definiciji agregacija, a to se obrađuje kasnije u knjizi.

5.7.1 Brojnost tipa veze

Tip veze sastoji se od dvaju preslikavanja (koja nazivamo uloge) između tipova entiteta. Ta preslikavanja mogu se ograničiti što nas dovodi do pojma brojnosti preslikavanja.



Slika 5.25 Brojnosti pojedine veze

Detaljnije opišimo ograničenje na preslikavanja između dvaju tipova entiteta. Svaki tip veze ima dvije uloge, odnosno svaki tip veze se sastoji od dvaju preslikavanja prvog skupa (tipa entiteta) na drugi i obrnuto. Svako od tih preslikavanja ograničeno je dopuštenim brojem povezanih entiteta. Brojnosti tipa veze su slične brojnostima atributa. Razmotrimo brojnosti na sl. 5.25. Analizom preslikavanja pojedinih žena prema mušarcima dolazimo do broja (broj naveden u zagradi pored poje-

dine žene) koji kaže koliko muškaraca je povezana vezom brak s odabranom ženom. Tako vidimo da je žena s rednim brojem 4 u braku s jednim (1) muškarcem vezom brak broj 3. i to s muškarcem označenim simbolom f . Žena pod rednim brojem 5 povezana je s (0) muškaraca. Za sve žene možemo općenito reći da ili nisu u braku i brojnost veze im je (0) ili jesu, pa je brojnost veze (1). Nula je najmanji mogući broj, a jedan je najveći mogući broj muškaraca s kojima jedna žena može biti povezana. Brojnost uloge Udana je uređen par (min, max) pri čemu je min najmanji mogući broj, a max najveći mogući broj muškaraca koji su u braku s nekom ženom. Brojnost uloge Udana je (0,1). Slično možemo zaključiti da uloga Oženjen ima brojnosti (0,1). Brojnost tipa veze je skup brojnosti njegovih uloga, odnosno brojnost tipa veze Brak je (0,1) : (0,1).

Svaka pojedina veza iz nekoga tipa veze povezuje entitete dvaju tipova entiteta. Možemo ići od jednog do drugog entiteta jednog tipa entiteta, na primjer redom od jedne do druge žene i pitati se kolika je brojnost te konkretnе veze.

Brojnost tipa veze je broj (upisuje se eksplisitno na DEV) koji kaže koliko entiteta pojedinoga tipa entiteta E_1 sudjeluje (pojavljuje se) u tipu veze V s entitetom iz tipa entiteta E .

Neka u vezi V sudjeluju tipovi entiteta E, E_1 . Brojnost je tipa entiteta E nula (0), jedan (1), dva (2), ..., mnogo (M) ako se nula (0), jedan (1), dva (2), ..., mnogo (M) pojavljivanja tipa entiteta E povezuje s tipom veze V .

Do brojnosti tipa entiteta E_1 dolazimo tako da uzmemmo jedno (bilo koje konkretno e_1) pojavljivanje entiteta iz tipa entiteta E_1 . Pitamo se; koliko je najmanje (donna granica) i najviše (gornja granica) entiteta iz E povezano s e_1 iz E_1 i nalaze li se u tipu veze V ? Odgovor je brojnost tipa entiteta E_1 u tipu veze V . Brojnost se upisuje na DEV kao: (1,1), (0,M), (1,M), (M,M) i sl. na liniji spoja između E i V .

Slično se određuje inverzna brojnost tipa entiteta E , preslikavanje od E ka E_1 i upisuje uz tip entiteta E .

Moguće je odrediti donju (DG) i gornju granicu (GG) brojnosti za obije uloge, odnosno odgovoriti na pitanje:

- U koliko će se najmanje i najviše pojedinačnih pojavljivanja tipa entiteta E preslikavati pojavljivanje tipa entiteta E_1 ?
- U koliko će se najmanje i najviše pojedinačnih pojavljivanja tipa entiteta E_1 preslikavati pojavljivanje tipa entiteta E ?

Oba broja, najmanji i najveći broj preslikavanja, upisujemo na DEV, odvojene zarezom i postavljene u zagrade kao uređeni par (DG,GG). Uredeni par (DG,GG) nazivamo brojnost preslikavanja uloge. Brojnost preslikavanja tipa veze dva su uređena para oblika (DG,GG) : (DG,GG), pri čemu prvi par pripada jednoj ulozi, a drugi inverznoj ulozi.

Između dva tipa entiteta moguće su različite kombinacije preslikavanja. Razmotrimo neke karakteristične tipove brojnosti i prikažimo kako se navedene brojnosti čitaju.

- (1,1):(1,1) - jedan entitet prvoga tipa pridružuje se najmanje nijednom ili najviše jednom entitetu drugoga tipa i obrnuto.

Naprimjer, jednoj Osobi pripada jedna Domovnica, a jednu se Domovnicu izdaje samo jednoj Osobi (vidi sl. 5.26). Ukoliko zanemarimo Europsku uniju i neke države bez svoje valute možemo reći: jedna valuta Pripada najmanje jednoj i najviše jednoj zemlji, a jednoj zemlji Pripada (izdaje) najmanje jedna i najviše jedna valuta.

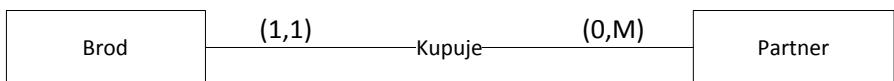


Slika 5.26 Tip veze (1,1):(1,1)

- (1,1):(0,M) - jedan entitet prvoga tipa pridružuje se najmanje nijednom i najviše jednom entitetu drugog tipa, a jedan entitet drugog tipa može se pridružiti najmanje nijednom i najviše većem broju entiteta prvoga tipa entiteta.

Na primjer, jedan brod iz tipa entiteta Brod kupuje najmanje i najviše jedan partner, a jedan partner može kupiti nijedan ili više brodova (vidi sl 5.27).

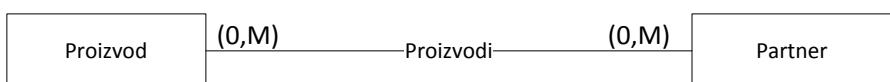
TIP VEZE prikazan kao linija



Slika 5.27 Tip veze (1,1):(0,M)

- (0,M):(0,M) - jedan entitet prvoga tipa može se pridruživati najmanje nula i najviše većem broju entiteta drugog tipa i obrnuto.

Na primjer jedan partner Proizvodi od niti jednog do mnogo proizvoda i jedan proizvod Proizvodi najmanje jedan do mnogo partnera. Partneri mogu kupovati ili proizvoditi proizvode, a nalaze se u istome tipu entiteta dok proizvod mora proizvoditi bar jedan partner, a može i više (vidi sl 5.28).



Slika 5.28 Tip veze M:M po gornjoj granici

Ograničenje na količinu entiteta koji sudjeluju u vezi naziva se još i sveobuhvatnost entiteta u vezi. Postoje dva tipa sveobuhvatnosti i to: Parcijalnost i Totalnost.

Tip entiteta povezan je s nekim tipom veze preko pojedinačnih pojavljivanja entiteta. Moguće je da, ili svi entiteti tipa entiteta sudjeluju u vezi, ili samo neki entiteti tipa entiteta sudjeluju u vezi.

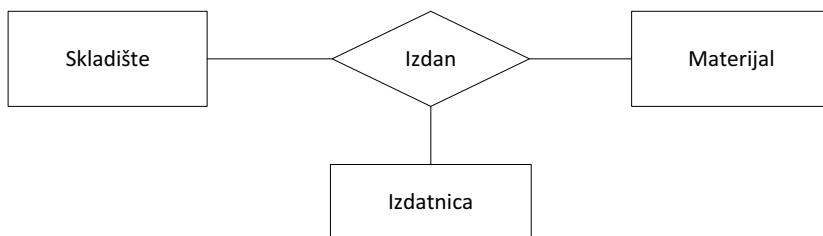
Tip entiteta sudjeluje **potpuno** (totalno, sveobuhvatno, obavezno) u vezi ako svi entiteti tega tipa entiteta sudjeluju bar u jednome pojavljivanju veze. Tada za tip veze kažemo da je totalna sa stajališta tega tipa entiteta, odnosno da je $DG>0$.

Tip entiteta sudjeluje **parcijalno** (djelomično, opcionalno) u vezi ako svi entiteti tega tipa entiteta NE sudjeluju u pojavljivanju veze, odnosno da je $DG=0$. Tada za tip veze kažemo da je parcijalna sa stajališta tega tipa entiteta.

Na DEV-u se totalna veza označava točkom na mjestu spoja linije od tipa entiteta s tipom veze, ili oznakom donje granice 1 u uređenome paru. Parcijalna veza je označena $DG=0$. Neki autori označavaju parcijalnu vezu kružićem na liniji spoja konceptata, a totalnost ne označavaju.

Zadatak: Razmotrimo vezu između Nacrta i Analitičkoga lista. Za svaki pojedini nacrt iz popisa nacrta postoji analitički list (popis operacija, odnosno radova koje treba izvesti po tome nacrtu) pa je veza prema Nacrtu totalna, ali ima analitičkih listova koji nisu nastali iz nacrtu (već na primjer iz Škartnoga lista, Internoga priopćenja ili Zahtjevnice), pa je veza prema Analitičkom listu veza parcijalna. Nacrtajte taj dijagram.

Uočimo da pisanjem donje i gornje granice brojnosti imamo informaciju o totalnosti, jer donja granica 1 označava totalnost te se totalnost i ne mora eksplicitno navoditi.



Slika 5.29 Trostruka veza - nije dozvoljena

Ako u tipu veze sudjeluju tri tipa entiteta, onda se pojedinačno promatraju sve kombinacije preslikavanja dvaju tipova entiteta prema trećem, a brojnost tog preslikavanja upisuje na liniju između tipa veze i trećega tipa entiteta. Tada postoje tri različita preslikavanja, od po dva različita entiteta u preostali, treći tip entiteta. Isto vrijedi ako u tip veze ulazi više od tri tipa entiteta. U tome je slučaju otežano prevođenje modela u relacijsku shemu baze podataka i dobivena shema ne mora biti u

višim normalnim formama. U našem slučaju modela podataka, prema konvenciji koja se u ovoj knjizi iznosi, zabranjene su trostruka i višestruka veza. Svaka višestruka veza se dekomponira na više dvostrukih veza te tako nastaju agregacije. Moguće je trostruku vezu proglašiti agregacijom triju tipova entiteta. Ključ te agregacije je složen od ključeva komponenti agregacije. Ovo je u praksi rijedak, ali moguć slučaj. Agregacija je kasnije detaljno opisana.

Na primjer trostruka veza je prikazana na slici 5.29. Tip veze Izdan povezuje tipove veza Materijal koji je izdan sa Skladišta pomoću Izdatnika.

Zadatak: Razmotrite odnos Slušao predavanja između tipova entiteta: Student, Profesor, Predmet.

5.8 Jak i slab tip entiteta

Jak tip entiteta (ili samo Tip entiteta¹⁶, engl. *strong¹⁷ entity*) je tip entiteta koji ima vlastiti primarni ključ i nije ovisan o drugim tipovima entiteta u modelu podataka.

Svaki tip entiteta mora imati primarni ključ. Ako se u modelu ustanovi tip entiteta koji nema attribute sa svojstvom ključa, u model se dodaje atribut tome tipu entiteta sa svojstvom ključa. Ovime se u postupku gradnje DEV-a za svaki tip entiteta utvrđuje ključ tipa entiteta.

Na primjer jaki tipovi entiteta i njihovi ključevi su: Radnik (Mbr), Škola (Šifra škole), Dobavljač (OIB), Proizvod (Šifra proizvoda), Partner (Redni broj partnera), Plan (Šifra plana).

Jaki su tipovi entiteta nezavisni od drugih tipova entiteta i mogu predstavljati na primjer šifarnike koji imaju uglavnom dva atributa (šifru i naziv) kao: Jedinica mjeere (Šifra jedinice mjere, Naziv jedinice mjere), Valute (Šifra valute, Naziv valute), Države (...), Tip skladišta (...), Vrsta dokumenta (...), Stručna spremna (...) i dr.

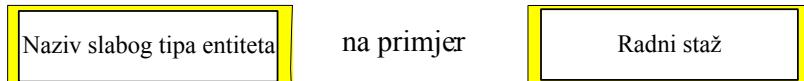
Slab tip entiteta (engl. *weak entity*) je tip entiteta koji je na neki način ovisan o nekom drugome tipu entiteta, a ta se ovisnost prikazuje specijalnim tipom veze među jakim i slabim tipom entiteta.

Na primjer: Djeca (ovise o Roditeljima), Pošiljka (ovisi o Narudžbi), Grad (ovisi o Državi), Staž (ovisi o ...), Prekršaj (ovisi o ...), Nagrada (ovisi o ...).

Tip slabog entiteta grafički se predstavlja dvostrukim pravokutnikom, jedan u drugom (vidi sliku 5.30).

¹⁶ Često se u razgovoru i literaturi kaže samo Entitet, a podrazumijeva se Tip entiteta.

¹⁷ Chen je u prvoj varijanti modela ER 1976. godine predložio termin *Regular entity*.



Slika 5.30 Grafički prikaz slaboga tipa entiteta

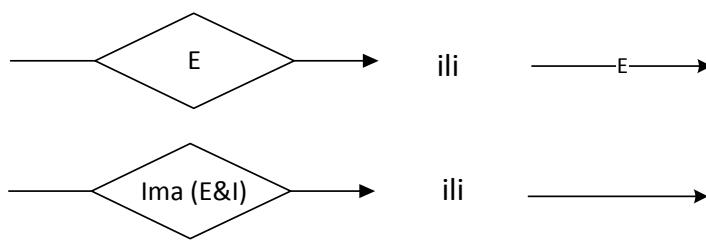
Pošto je slab tip entiteta uvijek vezan s jakim tipom entiteta tipom veze i samo se kroz ovu vezu ostvaruje odnos slab/jak, prikažimo specijalni tip veze koji povezuje jake i slabe tipove entiteta.

Specijalni tip veze je tip veze koji povezuje dva tipa entiteta među čijim entitetima postoji ovisnost. Moguće su različite vrste ovisnosti, a prema njoj definira se vrsta specijalnog tipa veze.

Različite vrste specijalnih tipova veza su:

- egzistencijalna (E)
- identifikacijska (I)
- egzistencijalna i identifikacijska (E&I)
- prethođenja (P)
- generalizacijska (G) i
- hijerarhijska (H)
- agregacijska (sastavljena od dvije E&I između triju tipova entiteta)

Grafički prikaz specijalnog tipa veze sličan je prikazu uobičajenih tipova veze (romb ili linija) s dodatkom strelice na liniji od jakoga k slabome tipu entiteta. Specijalni tip veze imenuje se kao i tip veze s upisanom oznakom vrste na primjer "E" (vidi sliku 5.31).



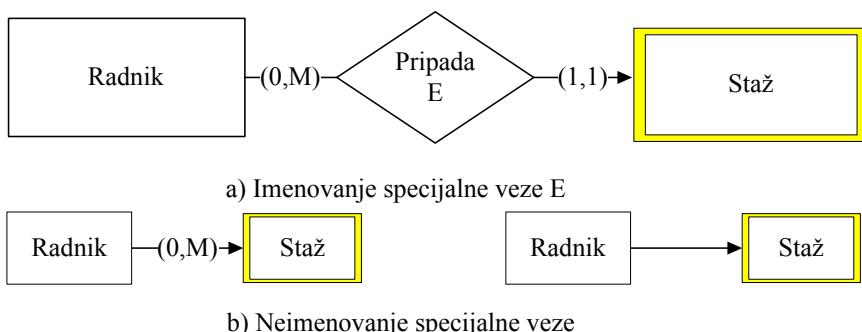
Slika 5.31 Prikaz specijalnog tipa veze

Naziv i oznaka specijalnog tipa veze može se i ne mora pisati na liniji veze. Specijalni tip veze bez naziva i oznake znači da je veza specijalna (prepoznajemo po liniji sa strelicom), ali nije bitan tip i naziv za dani model.

Prikažimo pojedine vrste specijalnih tipova veza.

Egzistencijalni tip veze (E) je tip veze koji povezuje dva tipa entiteta od kojih jedan egzistencijalno¹⁸ ovisi o drugom. Egzistencijalno ovisan tip entiteta je slab tip entiteta u odnosu na tip entiteta o kome ovisi, koji zovemo jak tip entiteta.

Na primjer, tip veze Pripada veže Radnik i Staž. Staž je slab tip entiteta, a tip veze Pripada je egzistencijalni tip veze (vidi sliku 5.32.). DEV sa slike čita se na sljedeći način: ima Radnika kojima Pripada najmanje nula Staževa (svi novozaposleni radnici i radnici koji nisu mijenjali radnu organizaciju ili nisu isli na radno mjesto s drugom vrstom staža, na primjer beneficirani staž), odnosno ima Radnika kojima Pripada najviše mnogo Staževa (svi radnici koji su radili u više firmi...). Obrnuto, jedan Staž Pripada jednom i samo jednom Radniku.



Slika 5.32 Egzistencijalna ovisnost Staža o Radniku

Egzistencijalna ovisnost između tipova entiteta eksplisitno se deklarira u DEV preko tipa specijalne veze (E), i ona predstavlja ograničenje na postojanje entiteta u modelu podataka u ovisnosti o postojanju drugih tipova entiteta. Tip entiteta koji je egzistencijalno ovisan u modelu naziva se slab tip entiteta i prikazuje dvostrukim pravokutnikom, a u tip veze prema entitetu od koga je ovisan upisuje se oznaka "E" i predstavlja specijalni tip veze (vidi sl. 5.32). Tip entiteta koji nije ovisan o drugim tipovima entiteta u modelu je jaki tip entiteta. On se na DEV-u predstavlja jednim pravokutnikom.

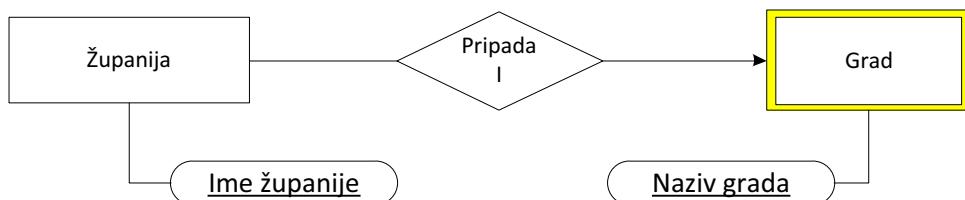
Preslikavanje je od slabog k jakome tipu entiteta uvijek (1,1), nazivamo ga **trivialno preslikavanje**, i ne moramo ga pisati na DEV-u. Obrnuto preslikavanje može biti bilo kakvo po brojnostima, a ako je (0,M) ne moramo ga eksplisitno upisati na DEV-u. To pojednostavljuje crtanje DEV-a. Neki autori (Bagui, 2003.) umjesto strelice na liniju prikazuju specijalni tip veze pomoću dvostrukoga romba.

¹⁸ Postojanje jednog ovisi o postojanju drugog

Identifikacijski tip veze (I) je tip veze koji označava da se jedan tip entiteta u modelu ne može jedinstveno identificirati vlastitim ključnim atributima, već se mora koristiti identifikator nekoga drugog tipa entiteta da bi se postigla jedinstvena identifikacija.

Pri izradi DEV-a potrebno je za svaki tip entiteta ustanoviti njegov ključ koji će jedinstveno identificirati sve entitete. Ukoliko to ne možemo, onda je to možda moguće pomoći ključa nekoga drugog tipa entiteta. U tome je slučaju taj tip veze koji ih povezuje identifikacijski.

Na primjer: tip je veze Pripada između tipa entiteta Grad s identifikatorom Naziv grada i tipa entiteta Županija s identifikatorom Ime županije identifikacijska veza, jer se svi gradovi u bazi podataka ne mogu jedinstveno identificirati samo s ključem Naziv grada. Postoje gradovi kao na primjer Novigrad u više županija i iz samoga naziva ne znamo točno o kome gradu se radi. Naziv je grada dobar identifikator unutar jedne županije. Tada kažemo da je ključ svakoga pojedinog grada složen od dvaju ključeva i to: Ime županije i Naziv grada.



Slika 5.33 Identifikacijski slab tip entiteta

Tip identifikacijski tip veze grafički se predstavlja rombom imenovanim kao svaki tip veze, s oznakom vrste "I" ili samo vektorom bez naziva i oznake vrste.

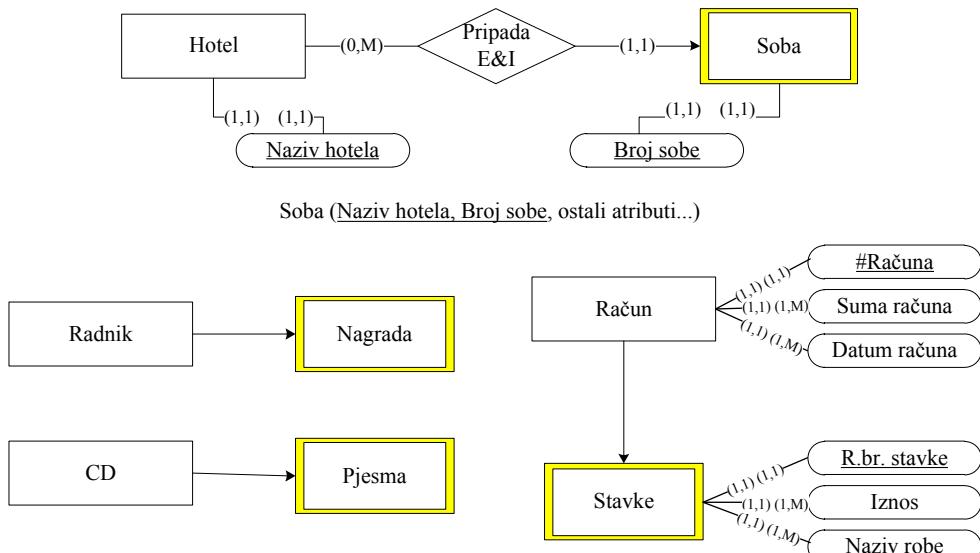
Identifikacijski ovisan entitet nije i egzistencijalno ovisan, na primjer, Grad se jedinstveno identificira Državom u kojoj je smješten, ali postojanje grada u stvarnosti ne ovisi o postojanju države. Propadanjem države ne moraju nužno propasti i gradovi.

Općeniti vrijedi da je identifikacijsko ograničenje istovremeno i egzistencijalno ograničenje (stoga što ne postoje identifikatori), a egzistencijalno ograničenje nije i identifikacijsko. Stoga se u relacijskim bazama za razliku od modela, ova dva ograničenja prevode kao jedno.

Egzistencijalni i identifikacijski tip veze (E&I). Dva su tipa entiteta povezana specijalnim tipom veze E&I ako je jedan tip entiteta istovremeno i egzistencijalno i identifikacijski ovisan o drugome tipu entiteta.

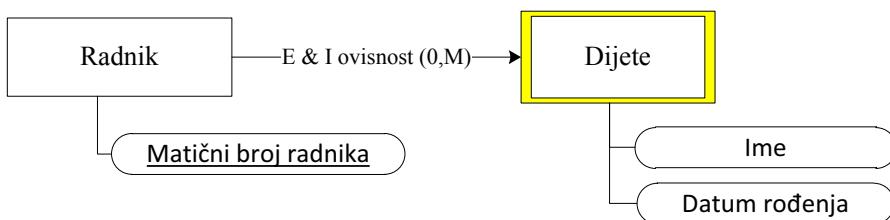
Na primjer: takva je veza između jakoga tipa entiteta Narudžbe i slaboga tipa entiteta Pošiljke te između Hotela i Sobe i sl.

Tip egzistencijalne i identifikacijske veze grafički se predstavlja rombom imenovanim kao svaki tip veze, s oznakom vrste "E&I" ili samo vektorom bez naziva i označke vrste (vidi sliku 5.34.).



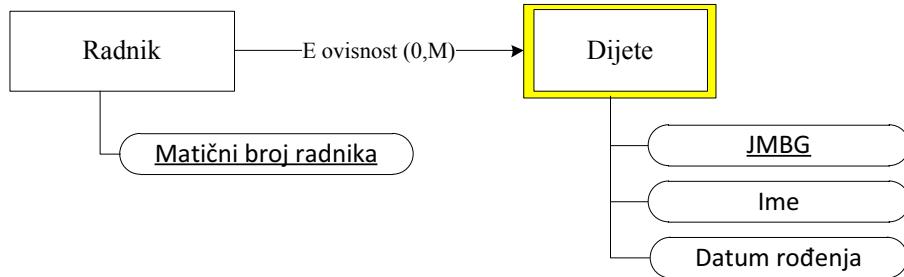
Slika 5.34 Primjeri slaboga tipa entiteta

Ako se želi istaći da je neki tip entiteta i egzistencijalno i identifikacijski ovisan, uvodi se ograničenje s oznakom "E&I" preko tipa specijalne veze. Na primjer, tip entiteta Dijete je egzistencijalno i identifikacijski ovisan o tipu entiteta Radnik.



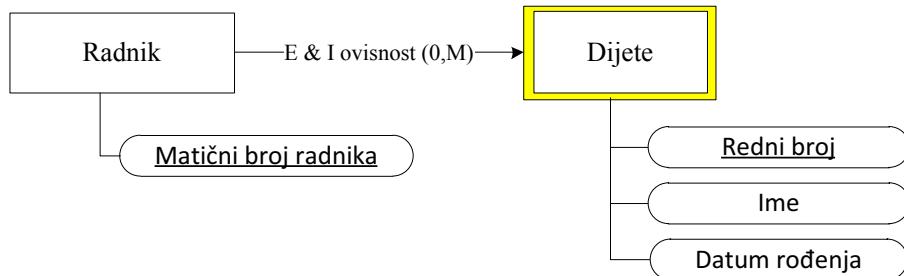
Slika 5.35 Slabi tip entiteta može biti istovremeno i ovisan i identificiran drugim tipom entiteta.

Uvede li se atribut Jedinstveni matični broj građana (JMBG) za ključ tipa entiteta Dijete, tada se E&I ograničenje može zamjeniti s E ograničenjem.



Slika 5.36 Iz modela se može ukloniti identifikacijska ovisnost uvodeći ključni atribut slabom tipu entiteta.

Slabi se tip entiteta može proširiti tako da pored svojih atributa sadrži i ključ jakega entiteta s kojim je u vezi.



Slika 5.37 Ako radnik ima više djece, tada je ključ slabog tipa entiteta Dijete sastavljen od Matičnoga broja i Rednoga broja djeteta.

Prevođenje slabog tipa entiteta u relacijsku shemu baze podataka prevodi se jednako za vrste specijalne veze I, E i E&I.

Pri prevođenju slabog tipa entiteta u relacijsku shemu baze podataka, slab tip entiteta postaje relacija, ključ je relacije složen od ključa jakoga tipa entiteta i ključa slaboga tipa entiteta. Ključ je slaboga tipa entiteta jedan od atributa slabog tipa entiteta koji osigurava jedinstvenu identifikaciju svih slabih entiteta za jedno pojavljivanje jakoga tipa.

Na primjer ako u bazi podataka postoje brojne stavke računa i ako je Redni broj stavke računa za svaki račun počinje od jedan na više, onda broj jedan nije ključ Stavke, ali za jedan odabran račun na primjer s brojem 231, redni broj stavke 1 točno označava jednu i samo jednu stavku.

Za operacija unosa, izmjene i brisanja u slabome tipu entiteta vrijede sljedeća pravila:

- Unos slabog entiteta nije moguć dok se ne unese jak entitet komu slab pripada
- Brisanjem jakoga tipa entiteta brišu se i svi slabi koji su s njim povezani
- Izmjena neključnih atributa je proizvoljna

Tip veze prethodenja (P). Tip specijalne veze je tip veze prethodenja među tipovima entiteta ako postojanju jednoga tipa entiteta (ne obavezno slabog) prethodi postojanje drugoga tipa entiteta.

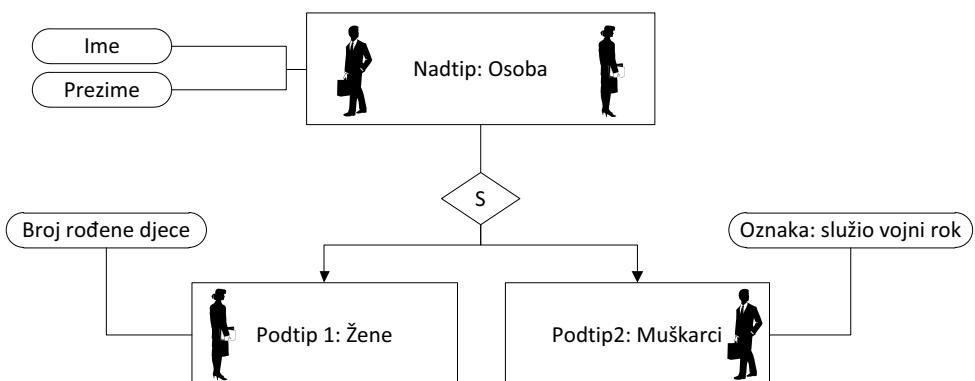
Na primjer: tip entiteta Porinuće prethodi tipu entiteta Primopredaja.

U praksi obično nije važan uzrok slabosti jednoga tipa entiteta od drugog te se na modelu ne mora pisati ime specijalne veze niti njezin tip te se, umjesto rombom, veza crta kao vektor od jakog ka slabom tipu entiteta. U modelu prethodenje označava redoslijed nastajanja entiteta i ovaj tip možemo zamijeniti E vrstom.

5.9 Generalizacija i specijalizacija

Neki tipovi entiteta E_1, E_2, \dots su **podtipovi** nekoga tipa entiteta E , koga zovemo **nadtip**, ako su sva pojavljivanja entiteta iz tipova entiteta E_1, E_2, \dots i pojavljivanja entiteta iz tipa entiteta E . Nadtip i podtipovi povezani su **generalizacijskim tipom veze** (ovisnost podtipa, G, S, engl IS, JE).

Na sl. 5.38. vidimo dvije osobe u nadtipu Osoba i te se iste osobe pojavljuju u svim podtipovima. Podtipovi i nadtipovi razlikuju se u atributima. U nadtip se dodaju atributi zajednički svim entitetima, a u podtip samo oni atributi koji pripadaju jednoj grupi entiteta, na primjer ženama.



Slika 5.38 Generalizacijsko stablo

Tip specijalne veze je generalizacijski tip veze ako povezuje jedan ili više tipova entiteta (tipovi entiteta podtip) s jednim tipom entiteta (tip entiteta nadtip).

Podtip i nadtip povezani su specijalnim tipom veze, koju zovemo generalizacijski tip veze, a skup tipova entiteta (podtipova i nadtipova), zajedno s generalizacijskom vezom, nazivamo **generalizacijsko stablo**.

Svi atributi nadtipa su i atributi podtipa (vrijedi pravilo da podtip nasljeđuje atribute nadtipa), ne i obrnuto.

Sve veze nadtipa su i veze podtipa, ne i obrnuto.

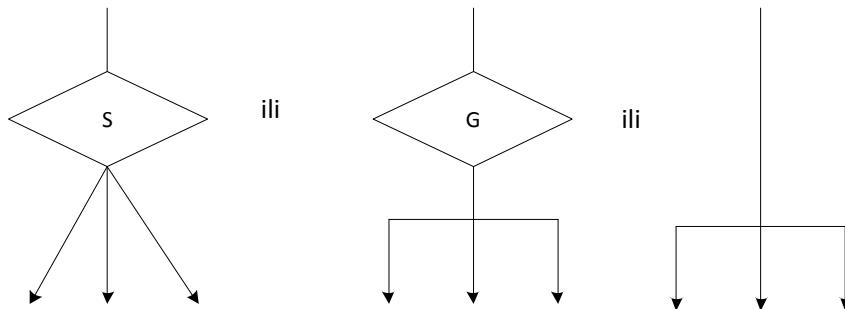
Sve operacije nadtipa su i operacije podtipa, ne i obrnuto.

Unija tipova entiteta E_1, E_2, \dots tip je entiteta E .

Presjek skupova E_1, E_2, \dots je prazan ako se svako pojavljivanje iz tipa entiteta E nalazi u jednom i samo u jednom od podtipova, ali to nije obavezno.

Nadtip tip entiteta E je s podtipovima, tipovima entiteta E_1, E_2, \dots , povezan specijalnim tipom veze, nazvanom generalizacija. Taj tip veze ima dvije uloge i to Specijalizaciju koja preslikava entiteta nadtipa u entitete podtipova i Generalizaciju koja preslikava entiteta podtipova u entitete nadtipa.

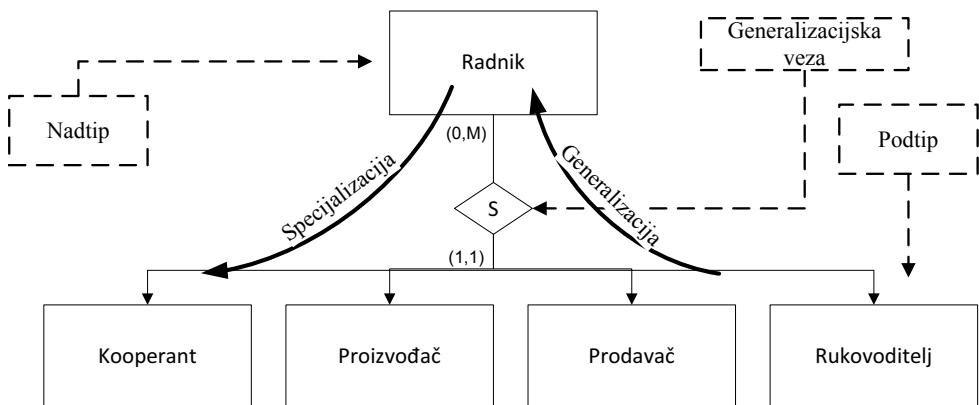
Tip generalizacijske veze grafički se predstavlja rombom imenovanim nazivom "S" od specijalizacija ili "JE" ili engleski "IS" ili "G" od generalizacija. Uobičajeno se nadtip postavlja iznad oznake generalizacijske veze (romba), a podtipovi ispod, s time da se od romba prema podtipovima povlače vektori (usmjerene strelice).



Slika 5.39 Generalizacijski tip veze

Na sljedećoj je slici dan primjer upotrebe tipa generalizacijske veze. Nadtip Radnik, podtipovi Kooperant, Proizvođač, Prodavač i Rukovoditelj i specijalno povezivanje nadtipa s podtipovima. Kada se u mislima krećemo od nadtipa k podtipu, kažemo da specijaliziramo entitete u podtipove, a obrnuto kažemo da generaliziramo razne pojmove u natpojam. Jedan radnik iz nadtipa može biti ni jednog podtipa ili može biti specijaliziran u M podtipova. Obrnuto jedan radnik iz bilo kojega podtipa mora biti najmanje jednom u nadtipu i najviše samo jednom u nadtipu radnik. Dakle preslikavanje od podtipa k nadtipu je trivijalno, odnosno uvijek (1,1). Pri analizi

preslikavanja svi podtipovi se promatraju kao jedan skup. Moguće je promatrati preslikavanje nadtipa k svakome podtipu i obrnuto i to posebno označiti na modelu, što nema posebnu korisnost jer su brojnosti preslikavanja od podtipa k nadtipu uvijek (1,1) a obrnuto se uvijek promatraju svi entiteti nadtipa.

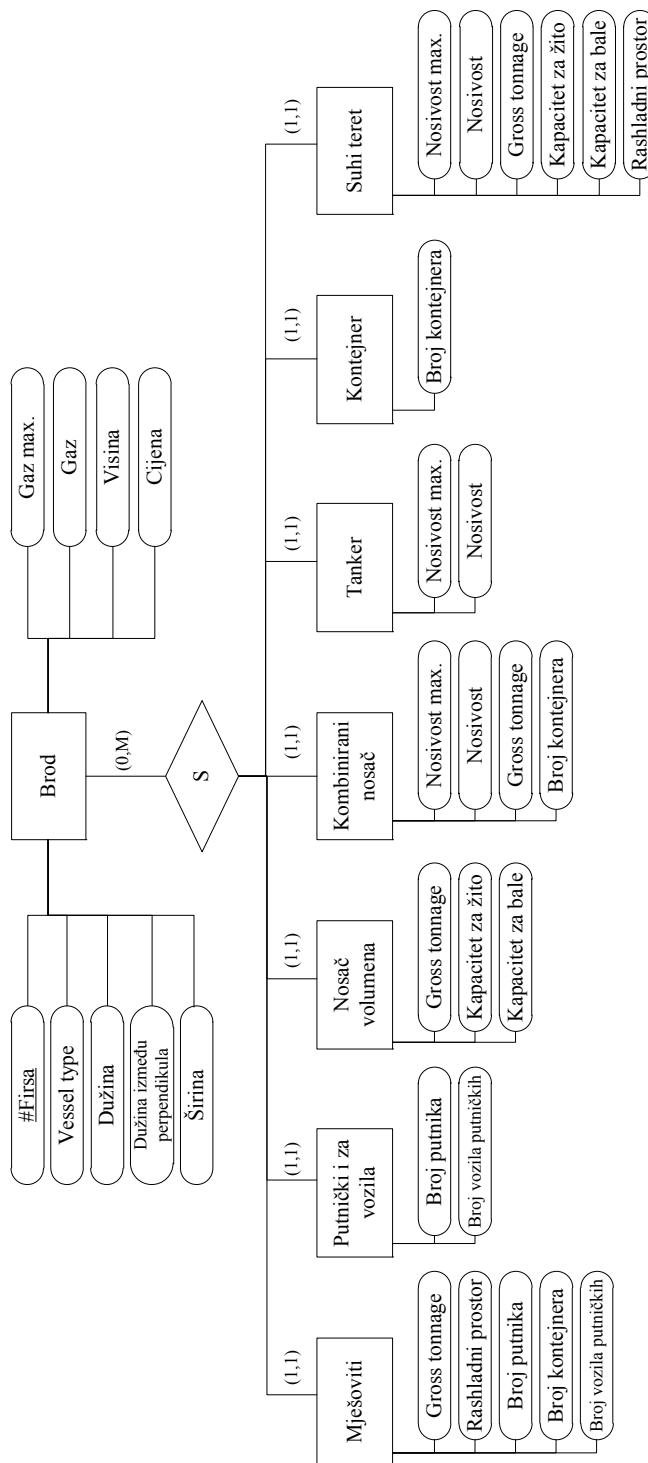


Slika 5.40 Generalizacijsko stablo s ulogama

Skup podtipova predstavlja raščlambu entiteta nadtipa u skupine koje predstavljaju jednu klasu sličnih entiteta. Podtipovi predstavljaju definiciju nadtipa grupiranjem pojavljivanja u skupove prema nekom aspektu gledanja na primjer sve vrste živih bića, zvijezda, vozila, proizvoda. Isti tip entiteta može se raščlaniti (generalizirati) s više aspekata u više generalizacijskih hijerarhija. Na primjer radnike se može generalizirati po spolu na muškarce i žene, po vrsti radnog odnosa na stalno i privremeno zaposlena, a po zrelosti na pripravnike i redovne radnike.

Atributi se između podtipova i nadtipa raspoređuju tako da se nadtipu pripisuju oni atributi koji imaju svi podtipovi, a podtipovima pripisuju preostali atributi (vidi sliku 5.41). Naravno, pojedinačno pojavljivanje entiteta u stvarnosti ima sve atribute.

Nadređeni tip entiteta govori općenito o kadru, a podređeni govori detaljnije o određenoj vrsti radnika.

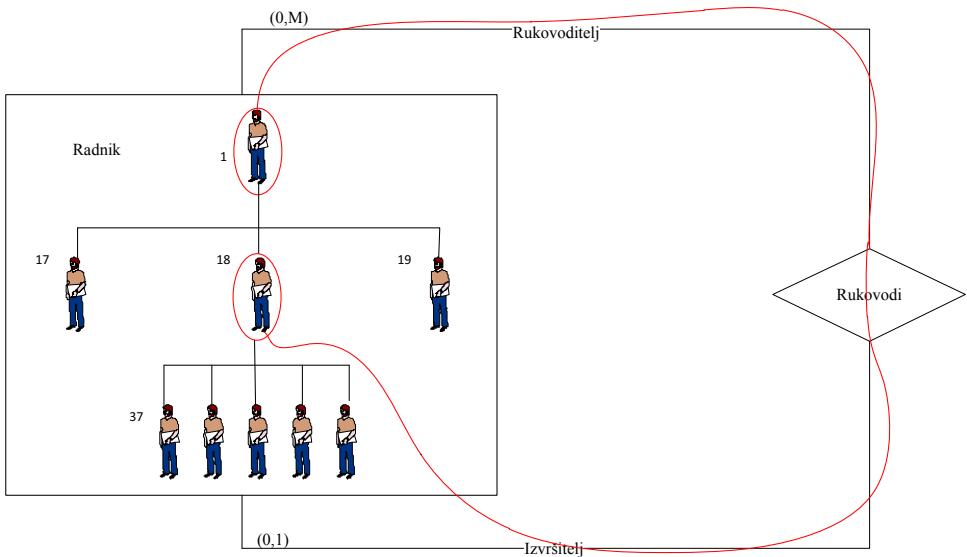


Slika 5.41 Primjer generalizacije tipa entitet Brod

5.10 Povratni tip veze

Povratni tip veze (rekurzivni tip veze, involucija, unarna, upletena, hijerarhijska).

Tip veze je povratni tip veze ako povezuje jedan tip entiteta sam sa sobom.

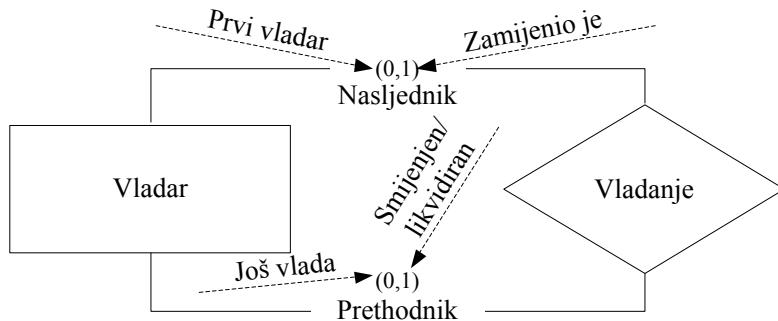


Slika 5.42 Tip entiteta Radnik s povratnom vezom Rukovodi

Jedno pojavljivanje radnika (vidi sliku 5.42) na višoj razini hijerarhije (dakle šefa) je u vezi s više pojavljivanja entiteta na nižoj razini (dakle izvršiteljima), dok je pojavljivanje entiteta na nižoj razini (izvršitelj) u vezi samo s jednim pojavljivanjem na višoj razini hijerarhije (ima najviše jednoga šefa). Pojavljivanja na najvišoj razini nisu povezana vezom prema gore (glavni direktor sa šifrom 1. na slici) pa je brojnost nula po donjoj granici uz ulogu Rukovoditelj. Na najnižoj razini (na primjer radnik sa šifrom 37) radnici nisu povezani povratnom vezom Radi prema dolje (ne-maju svojih podređenih) pa je brojnost nula po donjoj granici uz ulogu Izvršitelj.

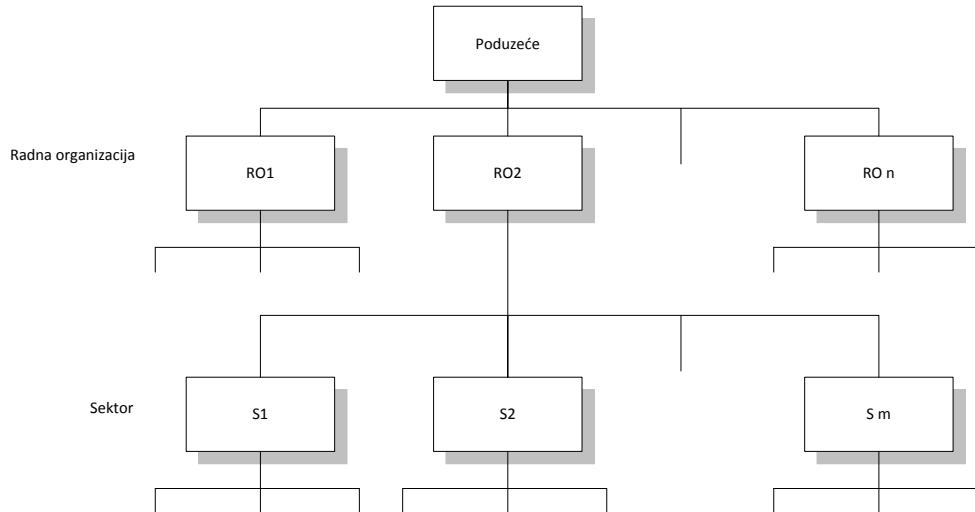
Povratni se tip veze grafički predstavlja rombom i imenovanjem uloga povratnoga tipa veze. U nekim se varijantama ER metode povratna veza prikazuje kao linija bez imenovanih uloga.

Na sl. 5.43 prikazana je povratna veza za praćenje naslijđivanja vladara u nekoj državi. Tip entiteta Vladar ima povratni tip veze Vladanje. Trenutnom vladaru ne zna se naslijednik - nepoznata vrijednost - a također je nepoznato koga je naslijedio prvi vladar (on nije nikoga naslijedio jer je prvi).



Slika 5.43 Povratni tip veze Vladanje (nasljedivanje, smjenjivanje)

Zadatak: Na slici 5.44 prikazana je uobičajena hijerarhijska organizacijska shema poduzeća. Za nju nacrtajte DEV, napišite imena uloga tipova entiteta u vezi i odredite brojnosti.



Slika 5.44 Hijerarhijske razine organizacije

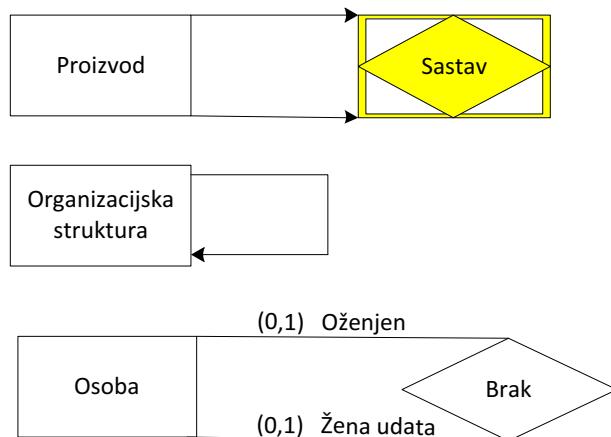
Konačno rješenje sastoje se od tipa entiteta OJ (Organizacijska jedinica), tipa veze Struktura i dviju uloga i to: "Sadržan" i "Sadrži". OJ je tip entiteta koji predstavlja sve razine hijerarhije organizacije. Jedno pojavljivanje entiteta, na primjer Radna organizacija Brodogradilište (ROB) "Sadrži U Sastavu" više Sektora; a jedan Sektor, na primjer Sektor Predmontaža sadržan je u sastavu samo jedne Radne organizacije. Organizacijske jedinice na višoj razini hijerarhije sadrže više jedinica na nižoj razini. Imena uloga: Sadržan i Sadrži. Ime tipa veze je U sastavu.

Dekompozicija hijerarhijskih razina poduzeća može se načiniti i pomoću slaboga tipa entiteta tako da na najvišoj razini imamo cijelo poduzeće, a svaka je niža ra-

zina slab tip entiteta s pojavljivanjem naziva organizacijskih dijelova te razine i tako do najniže razine. Na kraju bi se analizom atributa i ključeva, po metodi VATEK, opisanoj u poglavlju 8., slabi tipovi entiteta spojili u jedan tip entiteta, a specijalne veze spojile u povratni tip veze.

Na sl. 5.45. su dani primjeri povratne veze. Možete uočiti razliku u crtanjima povratne veze i to:

- ako su brojnosti M:M po gornjoj granici povratna veza crta se kao agregacija, uloge i brojnosti se mogu i ne moraju pisati
- ako je veza 1:M po gornjoj granici povratna veza se crta i kao specijalna veza, uloge i brojnosti se mogu i ne moraju pisati
- ako je veza 1:1 po gornjoj granici povratna veza se crta kao obična veza (kao linija ili kao romb), uloge i brojnosti se mogu i ne moraju pisati

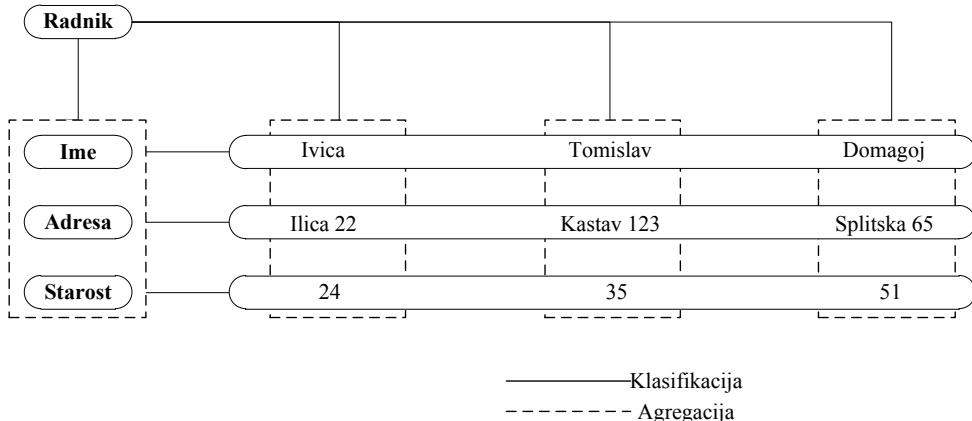


Slika 5.45 Različiti načini crtanja povratne veze, jednako kao i načini crtanja tipa veze

5.11 Agregacija

Agregacija (mješoviti tip entiteta, agregirani tip entiteta) je apstrakcija u kojoj se tip veze između dvaju ili više tipova entiteta tretira kao novi tip entiteta.

Razlika između dvaju tipova apstrakcije: klasifikacije i agregacije prikazana je na podacima o radnicima na slici 5.46..



Slika 5.46 Prikaz semantike klasifikacijske i agregacijske apstrakcije na tablici Radnik

Dok se pri klasifikaciji slični elementi povezuju u natpojmove, pri agregaciji se različiti elementi povezuju u natpojmove. Na slici se agregirani pojmovi prikazuju iscrtkanom linijom, a klasificirani punom linijom. Tako se pojedina imena klasificirana u natpojam atribut Ime, a pojedine osobe klasificirane u tip entiteta Radnik. Nasuprot tome podaci <Ivica, Ilica 22,24> su agregirani, a predstavljaju jedan pojedinačni entitet, a različiti atributi: Ime, Adresa i Starost agregiraju se u natpojam atributi tipa entiteta.

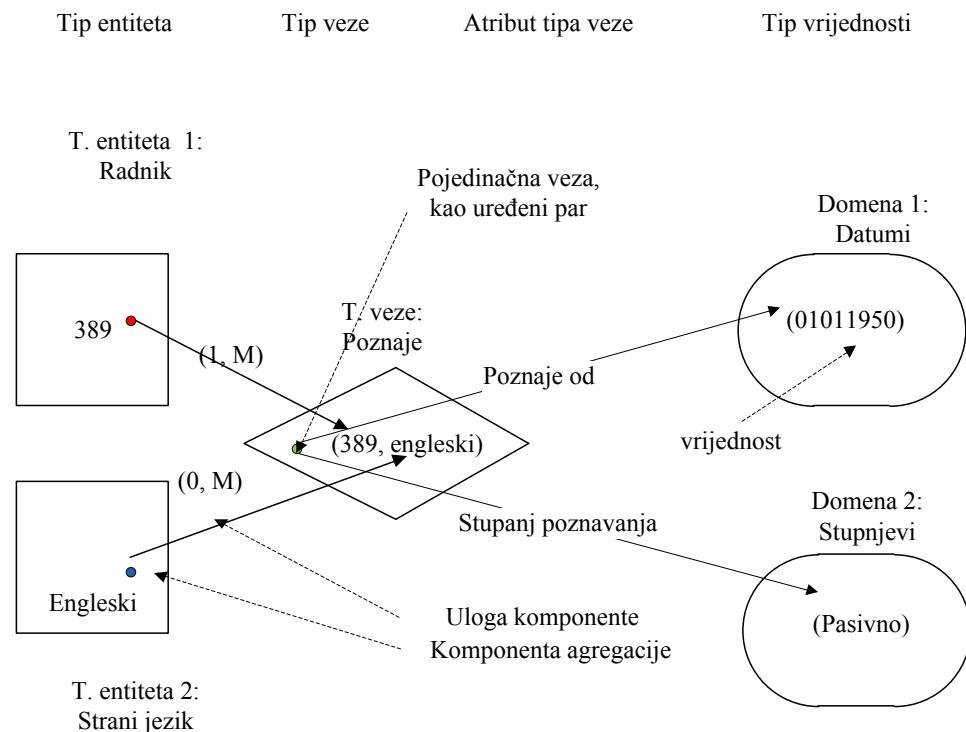
Svaka se veza u modelu podataka može proglašiti agregacijom. Tako je na primjer veza Studira između Predmeta i Studenta agregacija.

U ovoj su varijanti modela podataka agregacije samo oni tipovi veza s kojima bismo imali problema pri prevođenju u relacijsku shemu baze podataka. Tako se samo dio tipova veza pretvara u agregacije.

- Tip veze postaje i tip entiteta, odnosno postaje **agregirani tip entiteta** u slučaju
- 1. pojavljivanja atributa tipa veze (atribut ne može pripadati ni jednom pojedinačnom tipu entiteta)*
 - 2. potrebe za povezivanjem dvaju tipova veze međusobno (na primjer veza Student Polaze ispiti iz Kolegija treba povezati s vezom Ispituje Profesor)*
 - 3. rastavljanja višestrukih tipova veza u binarne tipove (na primjer veza Isporuka između entiteta: Roba, Dobavljač, Kupac, Proizvođač, Dostavljač i Skladište)*
 - 4. utvrđivanja brojnosti tipa veze u iznosu M po gornjoj granici s obiju strana (na primjer jedan student polaze ispiti iz puno kolegija (M), a iz jednog kolegija ispitu pristupa puno studenata (M), pa kažemo da su brojnosti po gornjoj granici M:M).*

Radnik i Strani jezik su tipovi entiteta povezani tipom veze Poznaje (slika 5.47.). Ako želimo imati podatke o stupnju poznavanja stranoga jezika, to ne možemo do-

dati kao atribut stranoga jezika (jer bi to značilo da svi radnici znaju strani jezik na jednakome stupnju) niti kao atribut Radnika (jer bi to značilo da radnik ima isti stupanj poznavanja svih stranih jezika). Prirodno mjesto atributa Stupnja poznavanja stranoga jezika je u tipu veze Poznaje. Time je moguće precizno reći koji radnik poznaje koji strani jezik u kome stupnju.



Slika 5.47 Grafički prikaz semantike koncepta tip veze, tip entiteta, atribut tipa veze i tip vrijednosti.

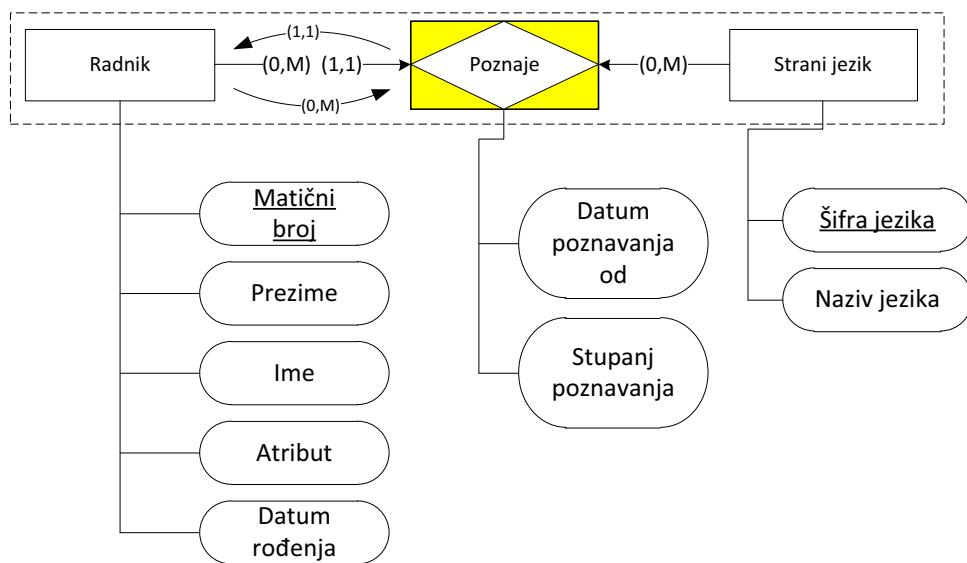
Agregirani tip entiteta nastaje od tipa veze u trenutku kada se pojavio atribut toga tipa veze. Svaki tip veze koji ima bar jedan atribut također postaje agregacija, odnosno aggregirani tip entiteta. Aggregirani tip entiteta je tip veze koji je postao tip entiteta. Tip veze koji ima atribut je tip entiteta, a ne tip veze zato što agregacija pri prevođenju u relacijsku shemu postaje relacija koja sadrži podatke potrebne za praćenje poslovanja. Tip veze ne postaje relacija.

Agregacija (aggregirani tip entiteta) se grafički predstavlja pravokutnikom u koji je ucrtan romb (vidi sl 5.48). Takav simbol ukazuje da je taj koncept prvo bio tip veze pa onda postao tip entiteta, odnosno da je riječ o mješovitom tipu i vezi i enti-

tetu istovremeno. Uloge tipa veze crtaju se kao vektori sa strelicama prema agregaciji, bez naziva uloga, što označava od kojih je tipova entiteta nastala agregacija. Sama uloga sada postaje specijalni tip veze između agregacije i jakoga tipa entiteta.



Slika 5.48 Tip veze i tip entiteta istodobno označavaju aggregirane tipove entiteta



Slika 5.49 Agregacija Poznaje je promatranje tipova entiteta i tipa veze kao jednoga podmodela

Agregacija Poznaje prikazana je na slici 5.49. Tip entiteta Strani jezik ima ključ Šifra stranoga jezika i atribut Naziv jezika. Tip entiteta Radnik ima ključ Matični broj i attribute Ime, Prezime, Adresa i Datum rođenja. Tip veze Poznaje povezuje Radnik i Strani jezik, a ima attribute Datum poznavanja od i Stupanj poznavanja.

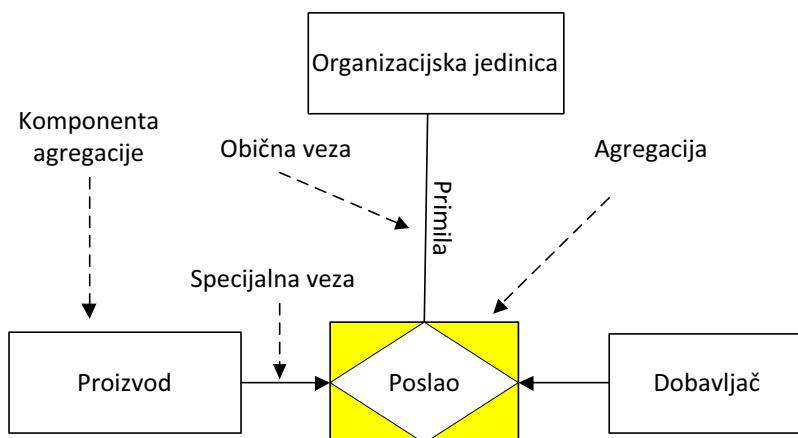
Analizom brojnosti preslikavanja agregacije zaključujemo da je svako preslikavanje od agregacije prema komponenti (1,1), zbog toga što je to specijalni tip veze E&I. Obrnuto preslikavanja može biti M po gornjoj granici.

Na sl. 5.49 prikazane su obje uloge preslikavanja (kao F i G preslikavanje) između Radnika i Poznaje. Po gornjoj granici preslikavanja od tipova entiteta prema

agregaciji s obje strane brojnost je M pa prema tome tip veze Poznaje postaje agregacija. Tako smo od tipa veze (na slici romb) nacrtali pravokutnik preko romba i dobili agregaciju. Ovo je najčešći razlog zašto tip veze postaje agregacija.

Agregacija može nastati u trenutku kada se ukazala potreba povezivanja tipa veze s tipom veze. Pri takvom vezivanju, tipa veze s tipom veze, jedan od tipova veze pretvara se u agregaciju. Izbor je koji puta proizvoljan.

Na primjer: ako bismo htjeli znati da je neki proizvod poslao neki dobavljač i da je taj proizvod primila neka organizacijska jedinica, imamo potrebu spojiti tipove veze Poslao i Primila. Tip veze Poslao između tipa entiteta Proizvod i Dobavljač proglašavamo agregacijom kada tip veze Poslao želimo povezati tipom veze Primila s tipom entiteta Organizacijska jedinica (vidi sl. 5.50).



Slika 5.50 Agregirani tip entiteta Poslao

Tipove entiteta koje povezuje agregacija nazivamo **komponente agregacije**.

Od komponenti agregacije k agregaciji crtaju se linije s vektorima. Na taj se način jasno prepoznaje koje tipove entiteta aggregira agregacija. Komponenta agregacije je povezana s agregacijom specijalnom vezom tipa E & I. To znači da je agregacija slab tip entiteta komponenti i to s dviju strana. Slab tip entiteta je jednostruko slab, a agregacija je dvostruko slaba.

Slaba agregacija je apstrakcija u kojoj se agregacija i slab tip entiteta od te agregacije prikazuju kao jedan tip slabe agregacije. Slaba agregacija grafički je skraćeni zapis sastavljen od simbola agregacije i od simbola slaboga tipa entiteta istovremeno (vidi sl. 5.2).

Slaba agregacija se eksplicitno može prikazati kao obična agregacija (koja se ne prevodi u relaciju i crta se iscrtkano na DEV-u), a koja ima slab tip entiteta. Pojednostavljenje slike je korištenje simbola slabe agregacije.

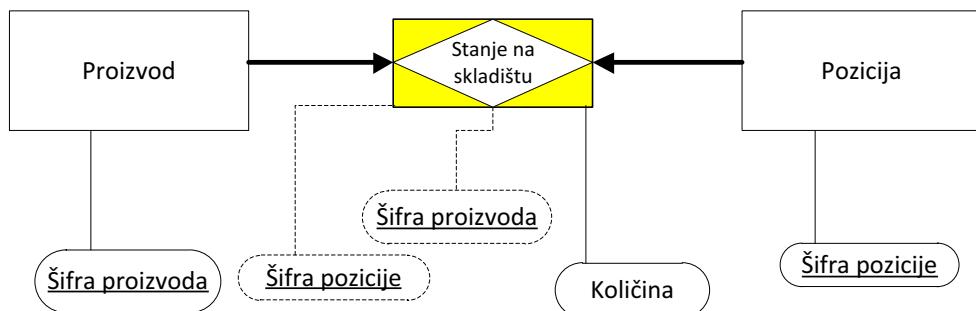
Svaka komponenta agregacije ima svoj identifikator.

Ključ aggregacije je skup ključnih atributa komponenti agregacije. Agregacija dobiva ključ od komponenti agregacije. Brojnosti ključa tipa aggregacije uvijek je (1,1):(1,1) inače to ne bi bio ključ.

Ključ slabe aggregacije je skup ključnih atributa komponenti aggregacije plus ključ slaboga tipa entiteta u slaboj aggregaciji. Slaba aggregacija dobiva ključ od komponenti aggregacije i obično redni broj stavke.

Ključ aggregacije ne crta se na DEV-u, on se podrazumijeva, a pri prevođenju u relacijski model zapisuje se ključ aggregacije u relacijskoj shemi baze podataka. Ostali se atributi aggregacije crtaju. Na sl. 5.51. gore nacrtan je ključ aggregacije isprekidanim linijom, samo kako bi se pokazalo kako izgleda model, ako se nacrtava ključ aggregacije. Ovaj model predstavlja trenutno stanje proizvoda na skladišnoj poziciji. Ali, želimo li pamtitи sav promet proizvoda i sve prošle raspoložive količine nije nam dovoljna aggregacija, već nam je potrebna slaba aggregacija.

Ključ slabe aggregacije djelomično se crta na DEV-u (vidi sl. 5.51 dolje), i to tako da se ne crtaju ključevi komponenti aggregacije, ali se crta ključ slabe komponente (npr. Redni broj). Pri prevođenju u relacijski model ključ relacije nastale od slabe aggregacije konkatenirali su ključevi komponenti i rednoga broja.



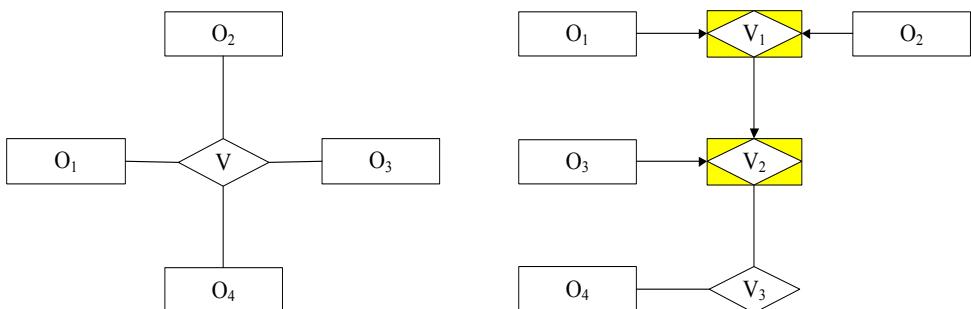
Slika 5.51 Ključ aggregacije i slabe aggregacije

Primjeri slabe aggregacije su: Hotel-Kategorija (praćenje svih promjena kategorije hotela na gore ili na dole), Dnevni TP program emitiran na jednom TV kanalu (moguće je da se ista emisija ponovi istoga dana na istome TV kanalu te je potreban Redni broj stavke emitiranja, a Šifra emisije je vanjski ključ).

Ključ tipa aggregacije Poznaje (vidi sl 5.49) je sastavljen od ključeva tipova entiteta koji ulaze u aggregaciju. Tako aggregacija Poznaje između tipa entiteta Radnik i Strani jezik ima ključ sastavljen od Matičnog broja radnika i Šifre stranoga jezika.

Ključ ove agregacije nije nacrtan eksplisitno na DEV-u, već se podrazumijeva implicitno, u skladu s prethodnom definicijom i pravilima prevođenja u relacijsku shemu baze podataka.

Agregacija može nastati i u trenutku kada se ukaže potreba za rastavljanjem višestrukoga tipa veze u binarne tipove veze. Postupak ide tako da odaberemo dva tipa entiteta O_1 i O_2 i spojimo ih agregacijom V_1 . Potom tu agregaciju spajamo s trećim tipom entiteta O_3 novom agregacijom V_2 . I na kraju aggregaciju V_2 spajamo tipom veze V_3 s preostalim tipom entiteta O_4 . Tako nastaju dvije agregacije i jedan tip veze. Prvo povezujemo tipove entiteta koji su iz nekoga razloga semantički bliži. Ponekad je izbor povezivanja proizvoljan.



Slika 5.52 Rastavljanje četverostrukih veza u tri binarne veze

5.12 Operatori metode entiteti - veze

Jezik za manipulaciju podataka (DML) za EV nije definiran u vremenu nastanka modela entiteti-veze. Prvi se radovi uglavnom koncentriraju na prikaz upotrebljavanja sredstava za opisivanje sustava.

Kasnije je predloženo niz različitih DML-a (Shoshani, 1978., Lusk, 1980., Lien, 1980., Chan, 1980., Santos, 1980., Ursprung, 1983., Zhang i Mendelzon, 1983., Markovitz, 1983., Markovitz, 1983., Parent, 1985.).

Mada postoji niz radova, nije općenito uskladen standard osnova za manipulaciju, odnosno koji tipovi koncepcata mogu biti operandi. Većina radova ne daje potpun skup operatora nad struktukom EV modela.

Izložit ćemo Chenov doprinos razvoju DML-a za EV i prijedlog jednostavne algebre.

5.12.1 *Chen operatori*

U prvoj verziji EV-a (Chen, 1976.) definirana je semantika operacija:

1. *Dodavanje*
2. *Izmjena*
3. *Brisanje*

Pored ovih operacija definirane su i operacije za pretraživanje baze podataka i to:

4. *Selektiranje podskupova vrijednosti iz tipa vrijednosti*
5. *Selektiranje podskupova entiteta iz tipa entiteta*
6. *Selektiranje podskupova veza iz tipa veze*
7. *Selektiranje podskupova atributa iz tipa entiteta ili tipa veze*

5.12.1.1 Operacija 1. Dodavanje

Moguća su sljedeća dodavanja podataka:

- kreirati entitet u tipu entiteta, ali provjeriti je li vrijednost ključa entiteta moguća ili već postoji entitet s tom vrijednošću
- kreirati vezu u tipu veze, ali provjeriti postoje li entiteti u tipu entiteta s danim vrijednostima ključeva
- dodati vrijednost atributa entiteta, ali provjeriti je li ta vrijednost dopuštena.

5.12.1.2 Operacija 2. Izmjena

Moguće su sljedeće izmjene:

- izmijeniti vrijednost neključnog atributa entiteta
- izmijeniti vrijednost atributa koji je dio ključa znači izmijeniti tu vrijednost u svim entitetima tipa entiteta i u svim vezama tipova veze koji su povezani s entitetom
- izmijeniti vrijednost neključnog atributa u vezi. Ne može se mijenjati vrijednost ključnog atributa veze ako nova vrijednost ne postoji kao vrijednost ključnog atributa entiteta
- izmijeniti vrijednost atributa entiteta koji je dio ključa znači izmijeniti ovu vrijednost u svim slabim tipovima entiteta u kojima je ta vrijednost spuštena kao dio ključa.

5.12.1.3 Operacija 3. Brisanje

Moguća su sljedeća brisanja:

- brisati entitet u tipu entiteta i sve veze u svim tipovima veza u kojima se vrijednost ključa entiteta pojavljuje
- brisati veze u tipu veze
- brisati entitet u tipu entiteta i sve entitete u slabim tipovima entiteta čije postojanje ovisi o danom entitetu.

U svom sljedećem radu Chen je predložio operatore za rad sa shemom baze podataka (Chen, 1977.) i to: ADD, DELETE, SPLIT, MERGE, SHIFT.

Opći oblik formata operatora ADD, DELETE, SPLIT i MERGE nad operandima konceptima EV modela je:

Tablica 5.3 Operatori nad konceptima EV modela

Operator	Operand
ADD	Tip entiteta
DELETE	Tip veze
SPLIT	Tip vrijednosti
MERGE	Atribut

Pri čemu operator:

- ADD omogućuje dodavanje novoga tipa entiteta, tipa veze, tipa vrijednosti, atributa
- DELETE omogućuje brisanje postojećeg tipa entiteta, tipa veze, tipa vrijednosti, atributa
- SPLIT omogućuje dijeljenje postojećega tipa entiteta, tipa veze, tipa vrijednosti, atributa u više tipova entiteta, tipova veza, tipova vrijednosti, atributa, respektivno
- MERGE omogućuje spajanje više postojećih tipova entiteta, tipova veza, tipova vrijednosti, atributa u jedan tip entiteta, tip veze, tip vrijednosti, atribut, respektivno

Ovako definirani operatori ne predstavljaju jezik za rad s podacima (DML). Jezik za rad s podacima nazvan CABLE¹⁹ definirao je Shoshani (Shoshani, 1978.).

5.13 Varijante metode entiteti-veze

Neka od varijanti metode EV-a, iako se drugačije naziva ili crta, koristi se u svim specijaliziranim metodologijama koje uključuju modeliranje baza podataka.

Na slikama 5.53 i 5.54 prikazana je usporedba koncepcata strukture metoda za modeliranje podataka raznih specijaliziranih metodologija i to: MIRIS, CASE*Method, SSADM, Chen, i IEM. Metode u pojedinim specijaliziranim metodologijama mijenjaju se i prilagođuju potrebama informatičkog inženjeringu i mogućnostima CASE alata.

¹⁹ Chain - Based Language

Formirani model entiteti - veze treba zadovoljiti postojeće i buduće informacijske zahtjeve. Kvaliteta postignutoga modela može se provjeriti u suradnji s poznavateljima poslovnoga sustava koji postavljaju upite i koje se pokušava logički izvesti iz DEV-a. Tako je moguće proširivati i metodu entiteti - veze novim konceptima, vođeći računa o utjecaju na proces modeliranja. Koncepte strukture i ograničenja metode je moguće dodavati ili sužavati s ciljem da se utječe na proces modeliranja podataka. To može biti opravданo jer utječe na efikasnost razvoja informacijskoga sustava. Valja istaći da se EV treba mijenjati kako bi najprije pridonio dizanju razine kvaliteta organizacije informacija i informacijskom sustavu, a tek onda nenarušavanjem kvaliteta olakšao posao projektanta.

Postoje koncepti (jedan ili više) strukture metode koji su apstrakcija pojedinačnih pojavljivanja.

Svaki se model sastoji od koncepata strukture metode koji su osnova za prepoznavanje pojmoveva iz poslovnoga sustava.

Do modela se dolazi intervjuiranjem, analizom dokumentacije ili opservacijama nad poslovnim sustavom. Intervju korisnika - eksperta sastoji se od pitanja koja vode do niza uzastopnih otkrića modela. Model poslovnoga sustava pohranjen u ljudskome pamćenju prevodi se u formalan model. Analiza dokumentacije predstavlja proces prevođenja jednoga slobodnog i nestandardnog modela opisa sustava u formalan model. Opervacija nad sustavom je proces pronalaženja i prevođenja činjenica sustava u formalan model opisa sustava.

Koncept	E-V MIRIS	CASE*Method	SSADM
Entitet			
Veza	ili		
Atribut		nema simbol	nema simbol
Rekursivna veza			
Slab tip entiteta		nema poseban simbol	nema poseban simbol
Specijalna veza spuštanja ključa		nema	nema
Kardinalnost preslikavanja veze			
Generalizacija			nema
Agregacija		nema poseban simbol	nema poseban simbol
Alternativna zavisnost	zabranjena		

Slika 5.53 Usporedba koncepata strukture modela podataka specijalizirane metodologije: MIRIS (autor M. Pavlić), CASE*Method (autor R. Barker) i SSADM (autor G. Longworth)

Koncept	Chen	IEM	UML
Entitet			
Veza		—	—
Atribut		nema simbol	
Rekurzivna veza			
Slab tip entiteta	nema poseban koncept	nema poseban koncept	nema poseban koncept
Specijalna veza spuštanja ključa	nema poseban koncept	nema poseban koncept	nema poseban koncept
Kardinalnost preslikavanja veze			
Generalizacija			
Agregacija			
Alternativna zavisnost			

Slika 5.54 Usporedba koncepcata strukture modela podataka specijalizirane metodologije Chen (autor P.Chen), IEM (autor J. Marin) i UML.

5.14 Meta modeli

5.14.1 Meta model metode entiteti-veze

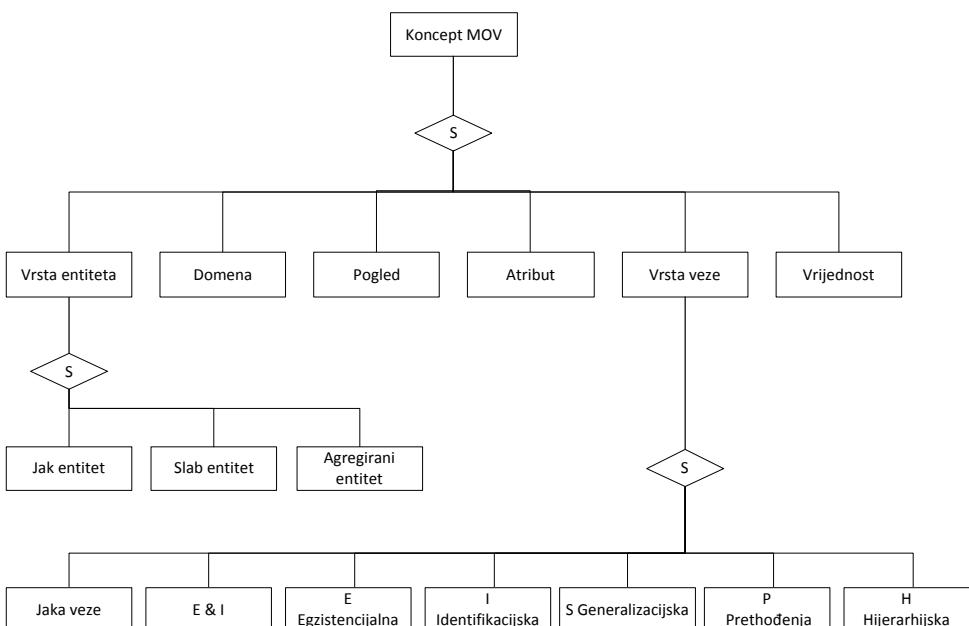
Meta model metode EV-a dan je u radovima: (Atzeni, 1983., Marti, 1983., Schuldt, 1987. i dr.).

Sveukupni model EV metode prikazan je na slici 5.56, a generalizacijski odnos koncepata strukture prikazan je na slici 5.55.

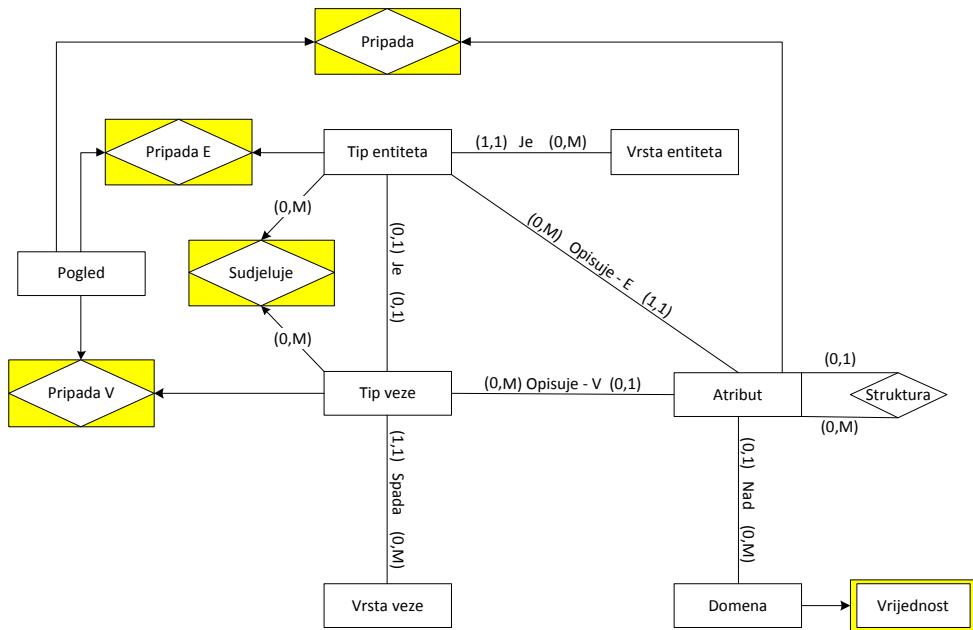
Osnovni je koncept za gradnju modela podataka Tip entiteta. Tipovi entiteta su međusobno povezani konceptom Tip veze. Jedan tip veze može povezivati više tipova entiteta, a ta se veza naziva „Sudjeluje“.

Model je potpun tek ako je tip entiteta definiran pomoću atributa. Ta pripadnost atributa entitetu označava se kao “Opisuje O”. Također postoje atributi agregacija (tipa veze), a ta pripadnost naziva se kao “Opisuje V”. Svaki atribut je definiran nad jednom domenom. “Nad” je veza između koncepata Atribut i Domena.

Neki tip veze može biti agregacija. To se znanje ugrađuje u model uvodeći novu vezu “Je aggregacija” između Tipa entiteta i Tipa veze. Tip entiteta i Tip veze mogu se klasificirati u razne vrste. Domena može imati neke diskretne vrijednosti. Pogled je koncept koji predstavlja agregaciju tipova entiteta, tipova veza i atributa iz sveukupnoga modela preko koje korisnik promatra model.



Slika 5.55 Metamodel generalizacija koncepata strukture



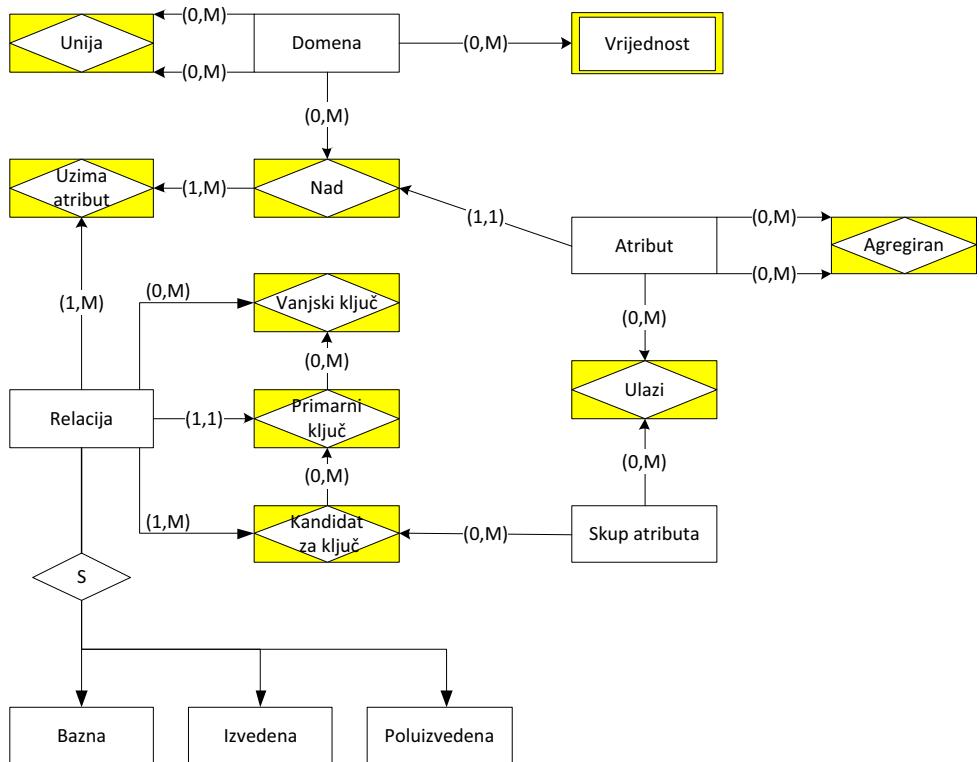
Slika 5.56 Meta model strukture EV modela

5.14.2 Meta model relacijske metode

Relacijski model sastoji se od niza koncepata. Osnovni koncepti su dani u Tablici 5.4.

Tablica 5.4 Koncepti relacijskog modela

Struktura	Ograničenja integriteta	Operatori
Domena	Brojnost	Unija
Atribut	Kandidat za ključ	Presjek
Relacija	Primarni (glavni) ključ	Diferencija
Bazna	Vanjski ključ	Descartov produkt
Izvedena	Alternativni ključ	Theta-selekcija
Poluizvedena	Integritet entiteta	Theta-spajanje
N-Torka	Referencijalni integritet	Projekcija
Stupanj		Equi-spajanje
		Natural-spajanje
		Dijeljenje



Slika 5.57 Meta model relacijskoga modela

Relevantni tipovi entiteta meta modela su: Relacija, Atribut, Domena, Vrijednost, Skup atributa, Bazna relacija, Poluizvedena relacija i Izvedena relacija. Postupkom apstrakcije koncepta i odnosa između koncepata relacijskoga modela dobiven je meta model. Meta model RM je prikazan na slici 5.57.

Osnovni tipovi entiteta su: Relacija, Atribut i Domena koji su međusobno povezani tipom veze: Nad. U jednoj relaciji nad jednom domenom može biti definirano više atributa. Atributi upravo i služe zato da se mogu razlikovati stupci relacije, što imena domena ne omogućava. Domene u relaciji Radnik su na primjer: Prezimena, Matični brojevi, a atributi: Prezime poslovodio, Matični broj radnika.

Jedan atribut koji je definiran nad jednom domenom može se pojaviti u više relacija. Ovo znači da nije traženo jedinstveno imenovanje atributa u čitavoj shemi.

U modelu ne pamtimos domene složene od drugih domena pa su domene sve na istoj razini hijerarhije, te je jedan atribut u jednoj relaciji definiran nad jednom domenom. Prirodno je da u modelu imamo aggregirane attribute, odnosno da više atributa jedne relacije proglašavamo zajedničkim imenom. Što je domena aggregiranog atributa? To je unija domena atomskih atributa pa je neophodan tip veze Unija. Uočimo da je i dalje jedan atribut relacije definiran nad jednom domenom.

Atribut relacije ne mora poprimiti sve vrijednosti iz domene i vrijednosti atributa mogu biti NUL (kao nepoznate, ali i kao neprimjenjivo svojstvo) te se ova znanja specificiraju atributima tipa veze Nad.

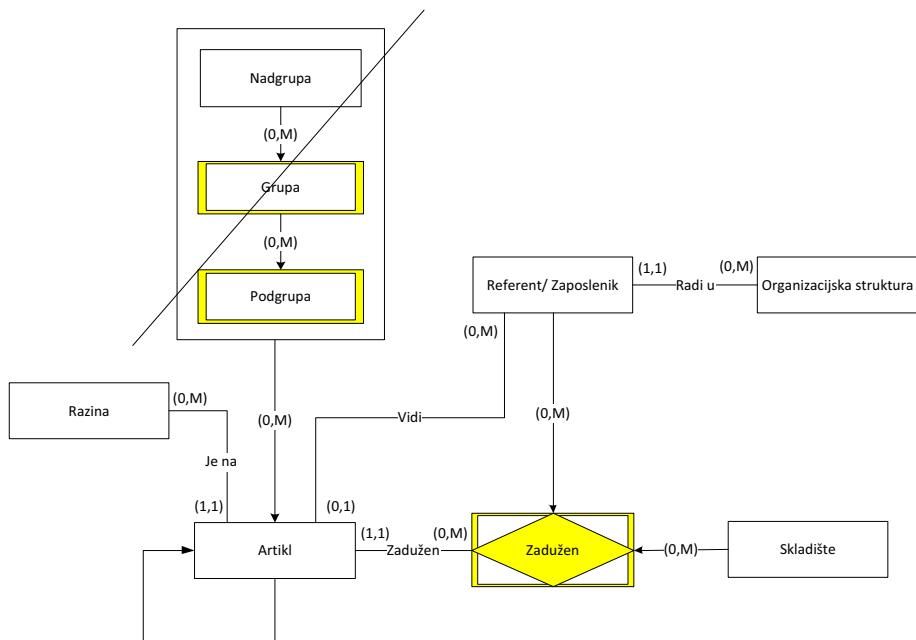
Relacijski model ima dobru matematičku osnovu u teoriji skupova pa je izučavan i jasno definiran. Temeljen je na skupu „tablica“. Korisnik može te tablice pretraživati, upisivati n -torke u njih, brisati i izmijeniti ih. Postoje različiti jezici za manipulaciju s podacima u relacijskome modelu podataka. Relacijski model je pogodan za implementaciju, a u praksi je nadograđen alatima za generiranje aplikacije, danas je vrlo popularan i praktično je sredstvo za gradnju baza podataka. Postoje brojni paketi programa za rad s relacijskim bazama podataka.

5.15 Primjer EV modela podataka

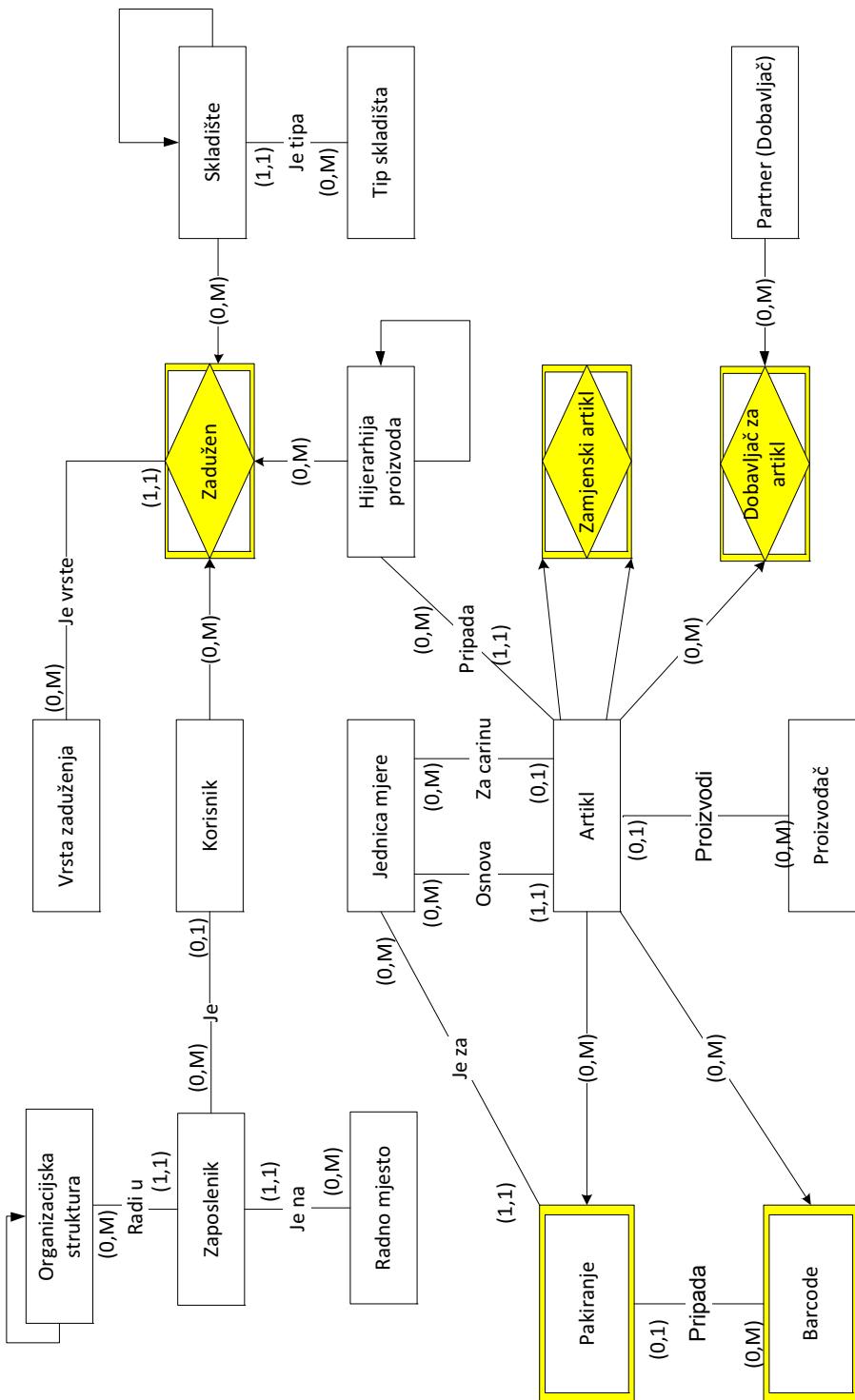
Prikažimo nekoliko modela podataka nacrtanih metodom EV-a.

Analizom podataka na svim dokumentima projektant informacijskog sustava dolazi do modela podataka. Model podataka za poslovni sustav komercijalno poslovanje sastoji se od desetak podmodела podataka, odnosno od niza DEV-a kao: *Artikl*, *Skladište* i dr. Svaki od tih modela može imati više verzija. Nove verzije u potpunosti zamjenjuju stare.

Sve te modele ne možemo prikazati u ovoj knjizi. Prikažimo nekoliko DEV-a i razliku u verzijama.



Slika 5.58 DEV Artikl – prva verzija



Slika 5.59 DEV Article – zadnja verzija

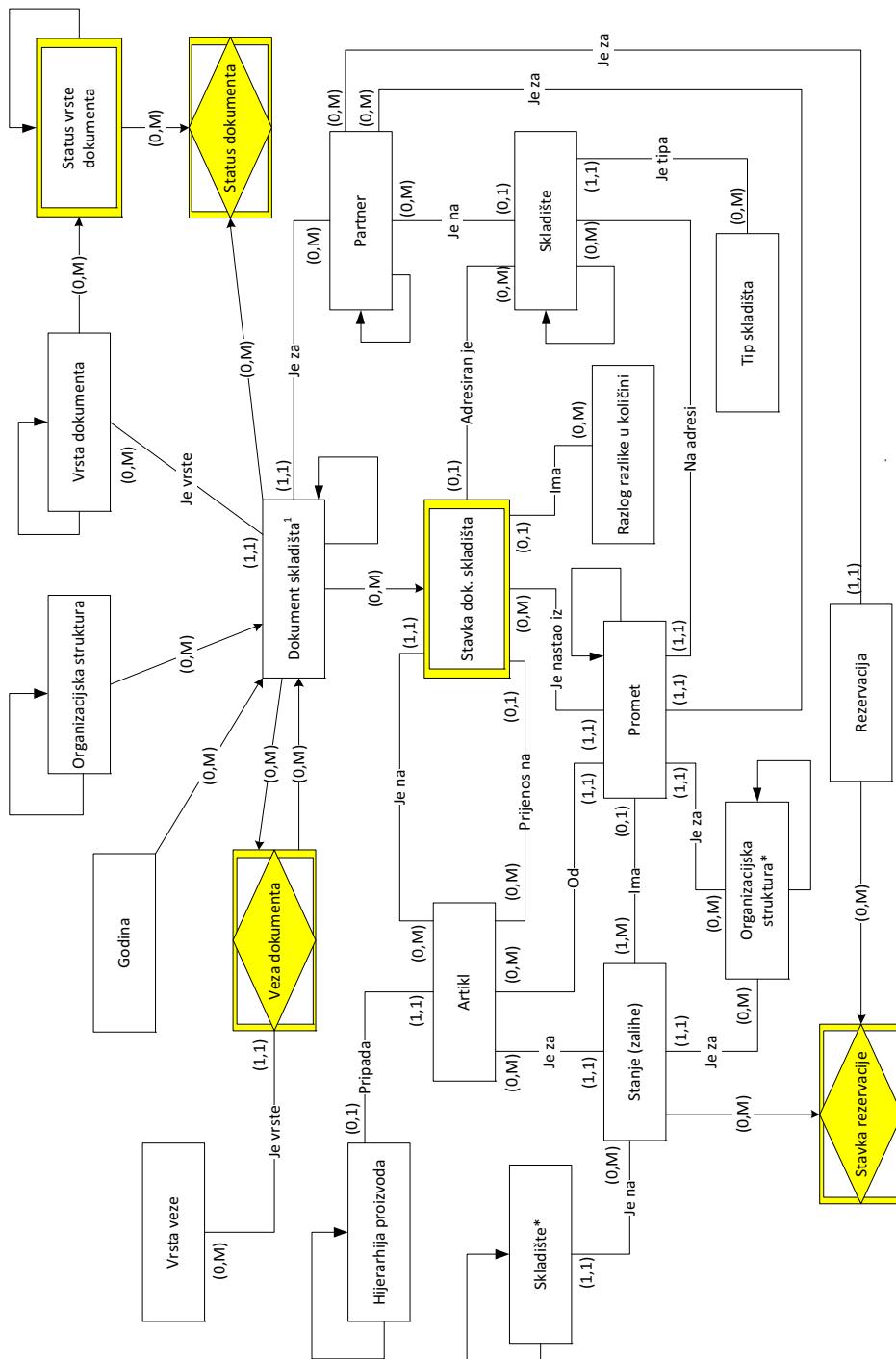
Na slici 5.58 prikazan je model podataka *Artikla* u prvoj verziji modeliranja podataka. Tijekom analize dokumenata nastalo je devet verzija. Zadnja verzija prikazana je na slici 5.59. Vidljivo je kako dizajner nije u stanju odmah napraviti najbolju verziju modela podataka, već je to put koji traje i tijekom vremena dobiva se model koji će ići u daljne faze razvoja IS.

Model podataka o artiklima (proizvodima, materijalu, robi) treba sadržavati sve podatke vezane uz articlje, a potrebne u cijelokupnome životnom ciklusu poslovne organizacije. Neki od zahtjeva za podacima su: pratiti articlje po proizvođačima, znati koji je articl nabavljen od kojeg dobavljača, imati mogućnost zamjene nekih artikala ako na tržištu nismo u stanju nabaviti potreban articl, imati mogućnost različitih pakiranja artikala s oznakom barkoda na pakiranju, za svaki articl imati barcode oznaku koja se može tijekom vremena za jedan articl mijenjati, osigurati mogućnost ugradnje artikala u složeniji proizvod, pratiti zaduženja osoba za skladištenja gotovih proizvoda i slično.

Na slici 5.60 prikazan je model podataka *Skladišta*. Atributi nisu prikazani na DEV-u, radi pojednostavljivanja slike DEV-a. Pitanje je s kojih su sve dokumenta podaci ušli u model podataka Skladišta. Model podataka Skladišta u sebe uključuje podatke s 14 poslovnih dokumenata:

1. Primka
2. Preskladištenje
3. Pripis-otpis
4. Promjena cijene
5. Inventura
6. Otpremnica
7. Otpremnica ispravak
8. Otpremnica povratnica
9. Evidencijska otpremnica
10. Rezervacija – samo podatak u modelu
11. Korekcija
12. Prijenos sa šifre robe na drugu šifru
13. Prijava neispravne robe
14. Komisijski zapisnik

Namjera ovoga prikaza nije objasniti proces integracije različitih DEV-a u jedinstven model podataka te ćemo na slici prikazati gotov DEV kako bi čitatelj video kako izgleda gotov proizvod dizajnera baze podataka.

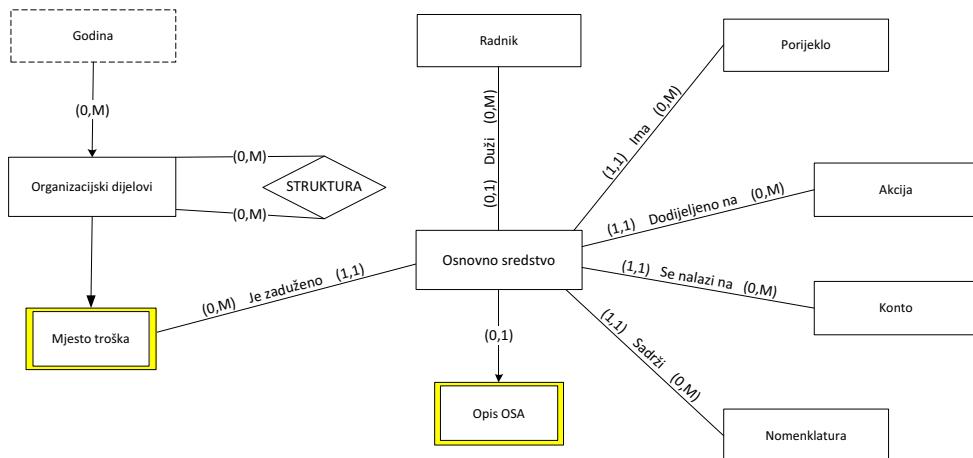


Slika 5.60 DEV dokument Skladište

Dakle, tijekom faze analize prikupljeni su dokumenti koji se dalje analiziraju i na osnovi podataka na njima kreira se model podataka. Model podataka ujedinjuje niz dokumenata u jedinstven model.

Neki modeli mogu biti jednostavniji kao model podataka osnovnih sredstava.

Primjer modela podataka za osnovna sredstva prikazan je na slici 5.62.



Slika 5.61 DEV Osnovna sredstva

6.1 Usporedba struktura i ograničenja relacijske i EV metode

Odnos EV-a i RM modela češće analiziraju razni strani autori, a analiziranje je započelo samim nastankom EV-a (Chen, 1976.).

Preko svojih koncepata strukture EV-a rješava niz semantičkih nejasnoća koje postoje u relacijskoj shemi.

Relacija je i tip entiteta i tip veze i agregacije ovih koncepata.

Upravo zbog poteškoća pri kreaciji relacijskoga modela podataka mnogi su autori analizirali provođenje EV-a u relacijski model s ciljem nalaženja praktičnoga procesa modeliranja podataka. Općenito je analizirano prevodenje DEV-a u modele podataka drugih vrsta (Chen, 1976.; Chen, 1977.; Melkanoff, 1980.; Tardieu, 1980.; Iossifidis, 1980.; Sakai, 1983.; Navathe, 1983.; Jajodia, 1983.; Yao, 1985.; Tsichritzis, 1982.).

Ograničenja se u relacijskome modelu ne iskazuju na formalan način izravno u shemi baze podataka već se, ukoliko se to znanje želi unijeti u dokumentaciju o bazi podataka (nije moguće u shemu baze podataka), iskazuje slobodnim tekstrom. Na primjer, na jednome projektu radi više radnika, a jedan radnik može raditi na više projekata.

Treba istaknuti da u relacijskoj teoriji, implementiranim relacijskim bazama podataka i CASE alatima koji podržavaju relacijske baze podataka, postoje mesta u kojima se ograničenja mogu deklarirati kroz naredbe CHECK, a neka ostvariti kao akcije baze podataka putem triggera. Ograničenja se mogu formalno deklarirati, iako većina ne, i u shemi!

Jedna klasa ograničenja je referencijalni integritet koji je potpuno deklarativan (relacijski model danas podržava opcije na primjer on delete cascade (u deklaraciji Nagrade foreing key radnik Broj references Radnik on delte cascade). CASE alati posjeduju „roundtrip engineering“ opciju da iz SQL DDL koda reverznim inženjeriranjem generiraju model podataka (uz neophodnu intervenciju projektanata).

Osnovna semantička razlika DEV-a i relacijskoga modela je da u relacijskome modelu ne postoje koncepti: slabi tip entiteta, brojnost, parcijalnost, generalizacija,

agregacija i specijalni tipovi veze. Zbog toga se kod prevodenja više tipova koncepata prevodi u jedan koncept relacijskoga modela pa obrnuta transformacija, iz relacijskoga modela u EV, nije moguća.

6.2 Pravila prevodenja modela EV-a u RM

Prikažimo kako se pojedini koncept u metodi EV-a prevodi u odgovarajući koncept relacijske sheme relacijskoga modela (RM).

PRAVILA 1.

Vrijednost (na primjer "132") u DEV-u odgovara **vrijednosti** u RM-u. Nul vrijednost u DEV-u odgovara nul vrijednosti u RM-u.

PRAVILA 2.

Tip vrijednosti (na primjer Prezime koji sadrži sva prezimena u radnoj organizaciji) u DEV-u odgovara konceptu **Domena** u RM-u.

PRAVILA 3.

Atribut u DEV-u (na primjer Ime radnika) odgovara konceptu **Atribut** u RM-u.

PRAVILA 4.

Entitet (pojedinačno pojavljivanje tipa entiteta, na primjer jedna polica) u DEV-u odgovara konceptu **n-torce (redak tablice)** u RM-u. Entitet je agregacija atributa, a *n*-torka je također agregacija atributa.

PRAVILA 5.

Kandidat za ključ tipa entiteta u DEV-u (na primjer Matični broj pribavljača, Jedinstveni matični broj građana) postaje (prevodi se, odgovara) **kandidatu za ključ tablice** u RM-u.

PRAVILA 6.

Svaki *tip entiteta* iz DEV-a postaje **relacija (tablica)** u RM-u i to tako da *atributi tipa entiteta* postaju **atributi relacije**, *primarni (glavni) ključ tipa entiteta* postaje **primarni (glavni) ključ relacije**.

Na primjer Tip entiteta Radnik s atributima: Matični broj, Prezime, Ime, Starost i ključem entiteta Matični broj u relacijskoj shemi postaje relacija:

RADNIK (**Matični broj**, Prezime, Ime, Starost)

Tip entiteta ne mora postati relacija ako je povezan samo s jednim tipom entiteta i to tipom veze čije su brojnosti (1,1):(1,1) s obje strane.

U dokumentaciji se za svaki tip entiteta popunjava tablica za detaljnije definiranje atributa toga tipa.

Tablica 6.1 Atributi stavke primke

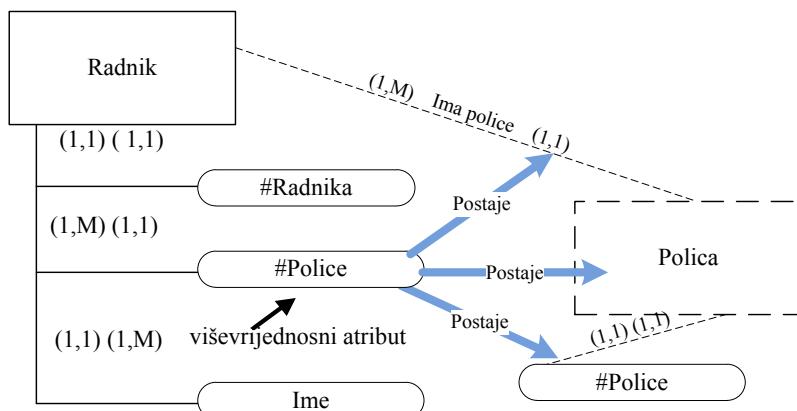
Redni br. tablice: 6 Naziv tablice: Sta_prim (stavka primke)			Kratica tablice: sprim Kategorija tablice: slaba			
Rb	Naziv stupca	Opis stupca	R. br. klj.	Bazna tablica	Tip i dulj	Prikaz
1.	Skl_sif	Šifra skladišta	1	Primka	I4	N5
2.	Prim_god	Godina primke	2	Primka	I1	N2
3.	Prim_br	Broj primke	3	Primka	I2	N4
4.	Sprim_br	Redni broj stavke primke	4		I1	N2
5.	Mat_sif	Šifra materijala		Materijal	Char10	N10
6.	Sprim_kol	Količina na stavci			F4	N9.3
7.	Sprim_cij	Stvarna cijena			F8	N16.2
8.	Sprim_knjkol	Knjižena količina			F4	N9.3
9.	Sprim_stat	Status tiskanja stavke N - nije tiskano T - tiskano G - gotovo			Char1	Ne

PRAVILA 7.

*Izvedeni atribut u DEV-u (na primjer Prosječna cijena proizvoda u organizacijskoj jedinici) odgovara konceptu **Izvedeni atribut** u RM-u.*

PRAVILA 8.

U DEV-u **nije dopušten viševrijednosni atribut**. U modelu podataka ne postoji niti jedan atribut koji pridružuje više vrijednosti iz domene jednog pojavnjivanju entiteta. Ako takav atribut ipak postoji, tada se taj *viševrijednosni atribut* pretvara u **tip entiteta**, a brojnosti atributa postaju brojnosti novoga tipa veze.



Slika 6.1 Viševrijednosni atribut #Police

Tako, na primjer, tip entiteta Radnik ima atribut # Police, kao što je prikazano na slici. Brojnost tog atributa je (1,M) od tipa entiteta k tipu vrijednosti, a obrnuto je brojnost (1,1). Ako jedan radnik može izraditi više polica, tada treba kreirati tip entiteta Polica i povezati ga s tipom entiteta Radnik.

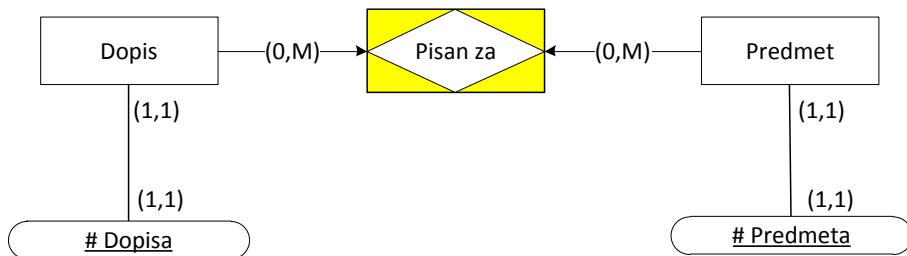
S mogućim prevodenjem viševrijednosnog atributa iz EV-a u RM dobivena relacija ne bi bila u prvoj normalnoj formi te bi se zahtijevao proces normalizacije podataka.

Zadatak: Je li atribut Staž tipa entiteta Radnik viševrijednosni atribut? Nacrtajte DEV s tipom entiteta Vrsta staža i slabim tipom entiteta Staž.

PRAVILO 9.

Svaka *agregacija* postaje **relacija** u RM-u, **ključ relacije** je složen od **ključeva tipova entiteta koji sudjeluju u vezi**. Atributi agregacije postaju atributi relacije.

1. Shema DEV - a



2. Shema RM - a

Dopis (#Dopisa)

Predmet (#Predmeta)

Pisan za (#Dopisa, #Predmeta)

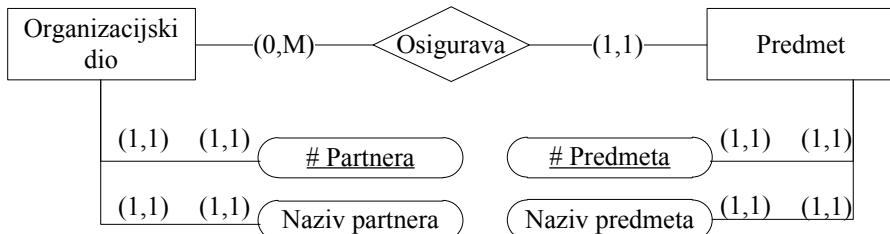
Slika 6.2 DEV preveden u relacijski model

Na primjer Tip veze Pisan za (vidi sliku 6.2) u relacijskoj shemi postaje relacija s ključem “#Dopisa,#Predmeta” složenim od ključeva tipova objekata.

PRAVILO 10.

Svaki *tip veze s brojnostima (1,1):(0,M)* (tip veze gdje je strana s brojnošću po gornjoj granici 1 totalna) **ne prevodi se u relaciju**, već se **ključ tipa entiteta koji ulazi u vezu sa strane s brojnošću M umeće kao atribut** (vanjski ključ) u **relaciju dobivenu prevodenjem tipa entiteta s brojnošću (1,1)**. Tip veze ne smije imati attribute jer, ako ih ima, to nije tip veze već agregirani tip entiteta.

1. Shema DEV - a



2. Shema RM - a

Organizacijski dio (#Partnера, Naziv partnera)

Predmet (#Predmeta, Naziv predmeta, #Partnera)

Slika 6.3 DEV preveden u relacijski model

PRAVILA 11.

Svaki $(1,1):(1,1)$ tip veze koja je s obje strane totalna (donje granice su jednake 1) **ne prevodi se u relaciju**. Od dva tipa entiteta koja povezuje taj tip veze u relaciju prevodi se samo jedan proizvoljno odabran tip entiteta, a toj relaciji dodaju se i atributi drugoga tipa entiteta. To znači da za ključ relacije možemo odabrati bilo koji od ključeva tih dvaju tipova entiteta, jer svaki od njih identificira jedan i samo jedan entitet u oba tipa entiteta.

Nacrtajte primjer DEV-a s tipovima entiteta Zemlja i Valuta s atributima šifrom i nazivom, te DEV prevedite u shemu relacijske baze podataka.

PRAVILA 12.

Svaki $(0,1):(1,1)$ tip veze i svaki $(0,1):(0,1)$ tip veze prevodi se po pravilu 11.

PRAVILA 13.

Ako tip veze povezuje više od dva tipa entiteta (uzmimo "n"), tada se taj tip veze **razlaže u binarne tipove veza**, tj. u više povezanih agregacija i prevodi ranije danim skupom pravila. Od niza dobivenih agregacija obično samo jedna postaje relacija.

PRAVILA 14.

Specijalni tipovi veze: "identifikacijska (ID)", "egzistencijalna (E)" i "ID & E" **ne prevode se**. Oni utječu na prevodenje slabih tipova entiteta.

PRAVILA 15.

Tip slabog entiteta prevodi se u relaciju. Atributi tipa slabog entiteta postaju atributi relacije. Ključ relacije je složen i od ključa jakoga tipa entiteta i ključa slaboga tipa entiteta (vidi sliku 6.4) tako da jedinstveno identificira n -torke relacije u okviru relacijskoga modela.

1. Shema DEV - a



2. Shema RM - a

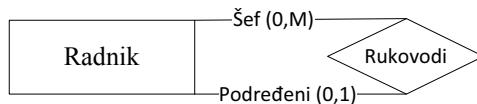
Županija (Šifra županije)
Naselje (Šifra županije, Naziv naselja)

Slika 6.4 Prevođenje slaboga tipa entiteta

PRAVILA 16.

Tip veze definiran nad istim tipom entiteta (povratni tip veze) prevodi se u relacijski model **kao i drugi tipovi veze**, u ovisnosti o brojnosti tipa veze.

1. Shema DEV - a



2. Shema RM - a (0,1) : (0,M)

Radnik (JMBG, ostali atributi, JMBG-Šef)

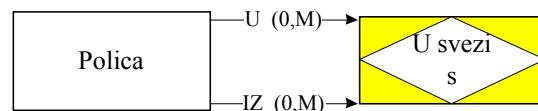
Slika 6.5 Prevođenje povratnoga tipa veze (0,1):(0,M)

1. Shema RM - a (0,M) : (0,M)

Polica (#Polica, ostali atributi, ...)

U vezi sa (#Polica nadređena, #Polica podređena)

2. Shema DEV - a



Slika 6.6 Prevođenje povratnoga tipa veze s brojnostima M:M po gornjoj granici

Nacrtajte DEV za tip entiteta Dio i tip veze Sastav te ulogama sastoje se od i nalazi se u sastavu. Prevedite DEV u relacijsku shemu.

Nacrtajte DEV za tip entiteta Organizacijska jedinica i tip veze Sastoje se te ulogama Nadređena i Podređena. Prevedite DEV u relacijsku shemu.

Nacrtajte DEV za čuvanje znanja o rukovodicima i njihovim nasljednicima ali i njihovim podređenima. Prevedite DEV u relacijsku shemu.

PRAVILO 17.

Generalizacija se prevodi u relacijski model tako da svaki *tip entiteta* postaje **relacija**, ključ svih relacija je jednak ključu tipa entiteta na vrhu generalizacije, atributi tipova entiteta postaju atributi relacije. Svaka relacija nastala od tipova entiteta na višoj razini dobiva atribut nazvan "PODTIP" koji može poprimiti vrijednost imena jedne relacije s niže razine generalizacije i označava da se pored podataka u danoj relaciji ostali podaci za odabrani entitet mogu naći u relaciji s imenom u stupcu "PODTIP" (Alagić, 1984.).

Na primjer, generalizacija tipa entiteta Brod s atributima zajedničkim svim brodovima i podtipovima Tanker, Kontejner i Putnički brod, s vlastitim atributima (nacrtajte DEV), daje relacije:

Brod (#Firsa, Dužina, Širina, Gaz, Podtip)
Tanker (#Firsa, Nosivost)
Kontejner (#Firsa, Broj Kontejnera)
Putnički (#Firsa, Broj Putnika, Broj Vozila)

PRAVILO 18.

Svaka *slaba agregacija* postaje **relacija** u RM-u. **Atributi slabe agregacije postaju atributi relacije. Ključ nastale relacije je složen od ključeva tipova entiteta koji sudjeluju u agregacijskoj vezi (komponenti agregacije) i od ključa slaboga tipa entiteta u slaboj agregaciji.** Ključ slaboga tipa entiteta u slaboj agregaciji je najčešće Redni broj, ali može biti i bilo koji drugi atribut koji zajedno s ključevima komponenti agregacije osigurava jedinstvenost i neredundantnost dobivene relacije.

Primjer modeliranja podataka niza dokumenata je dan u poglavlju 9. Za DEV Osobna iskaznica DEV-a je preведен u relacijski model podataka primjenom navedenih pravila. Za ostale modele čitatelj može sam vježbati prevođenje počevši od gotovog DEV-a.

7 NORMALIZACIJA

7.1 Normalizacija podataka

Normalizacija je metoda za raščlanjivanje sheme relacije, na dvije ili više shema koje se nalaze u višoj normalnoj formi od početne sheme i koje nemaju anomalija početne sheme relacije. U ovoj je knjizi dana uobičajena definicija normalnih formi (skraćeno NF). Poseban dodatak je prikaz normalnih formi i grešaka u organiziranju baze podataka pomoću metode EV-a. Na taj način pokazano je kako je i kako bi trebalo biti pomoću DEV-a. Tako se može vidjeti da:

- ako slab tip entiteta uklopimo u jak tip entiteta, tablica neće biti u 1. normalnoj formi (ne prepoznajemo postojanje slaboga tipa entiteta)
- ako i atributi, zajedno s ključem, jedne komponente agregacije kroz specijalni tip veze spustimo u agregaciju, tablica neće biti u 2. normalnoj formi (ne prepoznajemo postojanje agregacije)
- ako atributi jednoga tipa entiteta povezanog s drugim tipom veze 1:M spustimo zajedno s vanjskim ključem u drugi tip entiteta, tablica neće biti u 3. normalnoj formi (ne prepoznajemo postojanje tipa veze)
- ako je neki tip veze 1:M po gornjoj granici povezan s komponentom agregacije, krivo povezan s agregacijom, te zbog toga njegov ključ postane vanjski ključ agregacije, tablica neće biti u Boyce-Coddovoj normalnoj formi (krivo povežemo tipove entiteta s agregacijom)
- ako ne raščlanimo trostrukе i višestruke agregacije, tablica neće biti u 4., 5., i 6. normalnoj formi (višestruke agregacije su zabranjene u našoj verziji metode EV-a te su ovakve greške nemoguće)
- ako se definiraju ograničenja na tablicu koja nisu u okviru ograničenja na domenu i/ili ključ onda tablica nije u DCNF-u

Pored navedenog, a iz konstrukcija i koncepcata metode EV-a, možemo uočiti postojanje još nekih grešaka organiziranja podataka. Na primjer ako je krivo konstruiran model, što znači da su krivo postavljene funkcionalne zavisnosti, onda se postupkom normalizacije ne mogu dobiti tablice koje će preživjeti postupak testiranja i uvođenja. Daljnji primjer je prevođenje hijerarhijske povratne veze za organizacijsku jedinicu (1:M po gornjoj granici), postupkom normalizacije zaključujemo

da nije u 3. normalnoj formi, jer je Naziv organizacijske jedinice tranzitivno ovisan. Time bi trebalo proširiti i korigirati postupak normalizacije kako bi obuhvatio povratne veze. Ovu je temu moguće dalje istraživati.

Normalne forme su prikazane u tablici 7.11. U ovoj knjizi će biti opisan samo dio normalnih formi i uvest će se pojам nulte normalne forme.

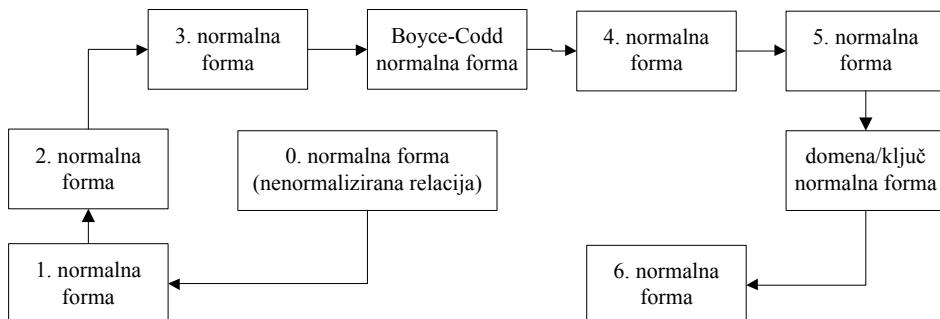
Opišimo postupak normalizacije.

Normalna forma je stanje u oblikovanju neke relacije.

Postoje sljedeće normalne forme: Nulta (0. NF), Prva (1. NF), Druga (2. NF), Treća (3. NF), Boyce-Coddova (BC NF), Četvrta (4. NF) i Peta (5. NF), domena/ključ (DKNF), Šesta (6NF).

Objasnit ćemo prve tri i Boyce-Coddovu normalnu formu. U praksi je dovoljno neku shemu dovesti do treće normalne forme. Intuitivno dobar DEV daje model u DK NF-u. 6NF je od interesa za vremenske baze podataka (engl. Temporal Database) i „skladište“ podataka (engl. Data warehouse).

Metoda normalizacije je proces prikazan na slici 7.1, koji najprije otkriva u kojoj normalnoj formi se relacija nalazi te je prevodi u više normalne forme. Pri tomu nije obvezno prevođenje u prvu sljedeću višu normalnu formu, već je moguće, na primjer, prijeći iz prve normalne forme u treću.



Slika 7.1 Slijed transformacija relacije iz najniže u najvišu normalnu formu

Normalizacija koju opisujemo je tzv. **vertikalna normalizacija**, tj. raščlanjivanje početne sheme relacije na "vertikalne podskupove", primjenom operacije projekcije.

Prilikom raščlanjivanja početne sheme relacije ne dolazi do gubitaka podataka. Normalizacija je proces kojim se to ne može dogoditi. Obnovljivost početne sheme relacije dokazuje se primjenom operacije spajanja novonastalih shema relacija.

Normalizacija započinje od sheme relacije za koju se pretpostavlja da nije normalizirana, ili je u nekoj normalnoj formi koja nije dovoljno dobra, te je treba prevesti u višu normalnu formu.

Tijekom procesa normalizacije promatrazu se odnosi atributa sheme relacije međusobno te odnos prema ključu sheme relacije. Pretpostavka za početak procesa normalizacije je postojanje primarnoga ključa sheme relacije.

Prikazat ćemo uobičajen proces normalizacije i dati njegovo tumačenje pomoću EV metode (metode entiteta i veza). Za svaku prepostavljenu relacijsku shemu prikazan je DEV (dijagram entiteta i veza) početnoga stanja i stanja nakon normalizacije te je time olakšano razumijevanje pogrešne organizacije podataka i poteškoća koje ona nosi.

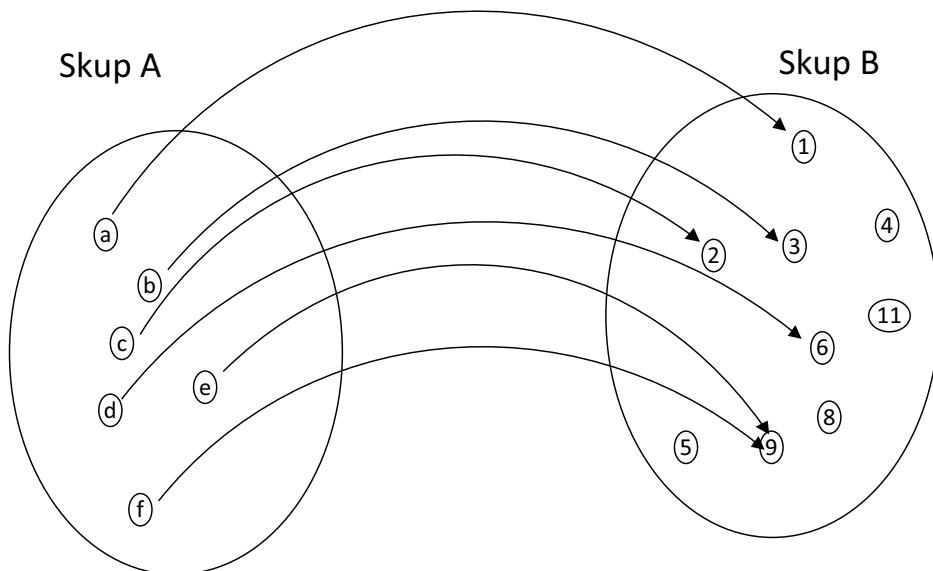
7.1.1 Prva normalna forma (1. NF)

Pri normalizaciji sheme relacije u 1. NF-u promatramo funkciju ovisnosti neključnih atributa o ključu sheme relacije.

Neključni atribut relacije je onaj atribut relacije koji nije komponenta složenoga ključa ili sam nije ključ relacije.

Funkcijska ovisnost skupa B o skupu A je preslikavanje pri kojem se svaki element skupa A preslikava u točno jedan element skupa B .

Prikazana grafički, na slici 7.2, pomoću simbola teorije skupova, funkcija ovisnosti skupa B o skupu A je pridruživanje članova skupa A (članova) članovima skupa B . Pridruživanje mora biti takvo da se jedan element skupa A preslikava (pridruži, poveže) u jedan i samo jedan element skupa B . Jedan element skupa B može biti pridružen različitim elementima iz skupa A (jednom ili više njih).



Slika 7.2 Slijed transformacija relacije iz najniže u najvišu normalnu formu

Atribut je funkcionalno ovisan o ključu, ako za svaku vrijednost ključa postoji samo jedna vrijednost tog atributa.

Uzmimo na primjer, neka je popis svih jedinstvenih matičnih brojeva građana (JMBG) skupa *A*, odnosno ključni atribut je JMBG, a imena građana (atribut Ime) je skup *B*. Svakome JMBG-u pripada jedno Ime. Isto ime može pripadati raznim građanima.

Definirajmo kada je neka shema u 1. NF-u:

Relacijska shema je u 1. NF-u, ako su svi njezini neključni atributi funkcionalno ovisni o ključu sheme relacije.

Pogledajmo primjer jedne nenormalizirane relacije i njezine sheme. Moguće je prikazati prve tri normalne forme na sadržaju narudžbe, ponude, računa, predračuna, primke i sličnih dokumenata koji imaju niz stavki robe koju netko kupuje, prodaje ili dobavlja. Potrebno je namjestiti početnu relaciju takvu da se u nju zapišu svi podaci s dokumenta, ne modelirajući podatke i ne organizirajući ih u više relacija. Uzmimo uobičajeni prikaz preko dokumenta Narudžbenice i odgovarajuće relacije Narudžba.

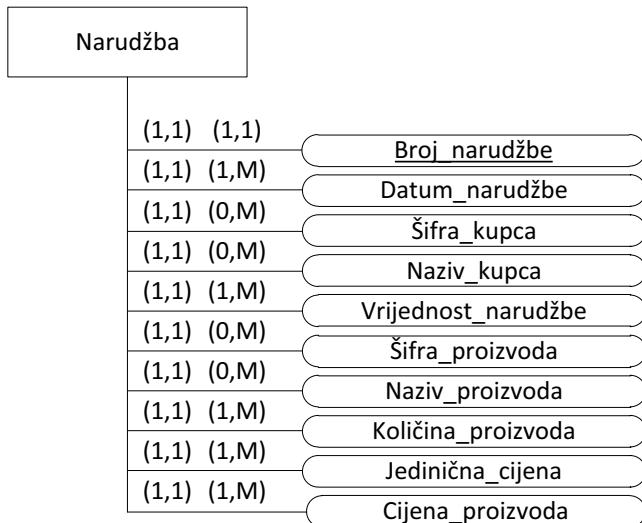
Relacija: Narudžba - Nenormalizirana relacija i njezina shema

Narudžba (Broj_narudžbe, Datum_narudžbe, Šifra_kupca, Naziv_kupca, Vrijednost_narudžbe, Šifra_proizvoda, Naziv_proizvoda, Količina_proizvoda, Jedinična_cijena, Cijena_proizvoda)

Tablica 7.1. Narudžba (0. NF)

Broj_naru-džbe	Datum_narudžbe	Šifra_kupca	Naziv_kupca	Vrijednost_narudžbe	Šifra_proizvoda	Naziv_proizvoda	Količina_proizvoda	Jedinična_cijena	Cijena_proizvoda
101	10-6-95	1527	Jadran	1000	2322	Kuh. stol	1	1000	1000
102	10-6-95	2023	Metal	20500	2434	Kabel φ20	20	25	500
102	10-6-95	2023	Metal	20500	3456	Brava S4	10	1000	10000
102	10-6-95	2023	Metal	20500	3457	Matica	100	100	10000
103	11-6-95	3532	Crota	8030	2434	Kabel φ20	400	20	8000
103	11-6-95	3532	Crota	8030	2323	Prekidač	1	30	30

Prikažemo li model podataka EV metodom, imamo početni model prikazan na slici 7.3.



Slika 7.3 DEV: Narudžba

Model podataka na slici 7.3 pokazao bi višestruke atribute pri određivanju brojnosti atributa i iz toga bismo zaključili da sam DEV nije dobar. Iz njega bi slijedila relacija Narudžba samo uz uvjet da ne određujemo brojnosti atributa.

Vidimo da postoji skupina atributa koji nisu funkcijски ovisni o ključu sheme relacije.

Atribut Šifra proizvoda je takav da jednom broju narudžbe odgovara više šifri proizvoda zato jer se jednom narudžbom može naručiti više proizvoda.

Takvi atributi uzrokuju probleme u "rukovanju" s relacijom i nazivaju se **anomalije**.

Anomalije koje se javljaju pri nenormaliziranoj relaciji nastaju u trenutku primjene operacija nad relacijama.

Osnovne operacije koje se mogu izvesti nad relacijama su: dodavanje, brisanje i izmjena.

Operacija dodavanja upisuje redak u tablicu. Operacija brisanja uklanja redak iz tablice. Operacija izmjene mijenja vrijednosti atributa jednoga retka.

Istaknimo anomalije relacije Narudžba:

- **Dodavanje** - ne možemo dodati informacije o kupcu, ako ne dodamo barem jednu narudžbu toga kupca. Ne možemo upisati u relaciju podatke o proizvodu ako ne postoji narudžba koja ga naručuje
- **Brisanje** - podatke o kupcu gubimo kada obrišemo njegovu posljednju narudžbu (isto vrijedi i za proizvod)
- **Izmjena** - ako proizvod ili kupac promijene vrijednost nekog od atributa, promjene moramo izvoditi u svim narudžbama, tj. u svim recima u kojima se pojavljuje taj proizvod ili kupac

Relacija Narudžba je u 0. NF-u i treba je prevesti u 1. NF-u kako bi se izbjegle navedene anomalije.

Normalizacija sheme relacije iz 0. NG-a u 1. NF je:

- nalaženje atributa koji nisu funkcijски ovisni o ključu
- izdvajanje nađenih atributa u novu shemu relacije

Nad novom shemom relacije ponavlja se postupak provjere normalne forme i postojanja anomalija.

Rezultat normalizacije u 1. NF-u je nova relacija Stavka narudžbe uz postojeću i izmijenjenu Narudžbu, prikazan je u tablicama 7.2 Narudžba i 7.3 Stavka narudžbe. Obje su u 1. NF-u.

Ključ nove sheme relacije je sastavljen od ključa početne sheme i ključa skupine koja je izdvojena.

A) **Narudžba** (**Broj_narudžbe**, Datum_narudžbe, Šifra_kupca, Naziv_kupca, Vrijednost_narudžbe)

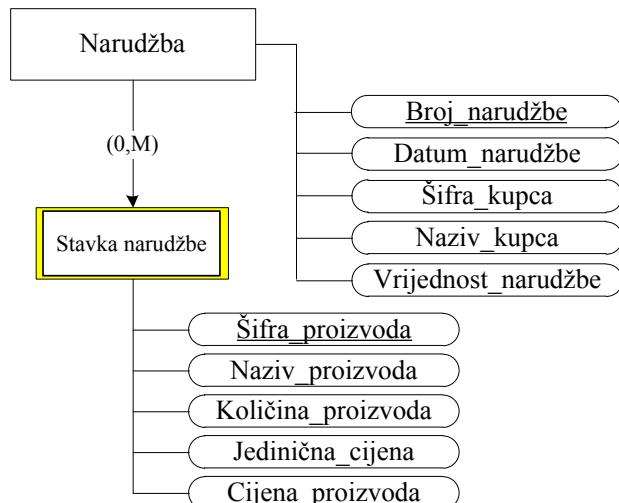
Tablica 7.2 Narudžba (1. NF)

Broj_narudžbe	Datum_narudžbe	Šifra_kupca	Naziv_kupca	Vrijednost_narudžbe
101	10-6-95	1527	Jadran	1000
102	10-6-95	2023	Metal	20500
103	11-6-95	3532	Crota	8030

B) **Stavka narudžbe** (**Broj_narudžbe**, **Šifra_proizvoda**, Naziv_proizvoda, Količina_proizvoda, Jedinična_cijena, Cijena_proizvoda)

Tablica 7.3 Stavka narudžbe (1. NF)

Broj_narudžbe	Šifra_proizvoda	Naziv_proizvoda	Količina_proizvoda	Jedinična_cijena	Cijena_proizvoda
101	2322	Kuh. stol	1	1000	1000
102	2434	Kabel φ20	20	25	500
102	3456	Brava S4	10	1000	10000
102	3457	Matica	100	100	10000
103	2434	Kabel φ20	400	20	8000
103	2323	Prekidač	1	30	30



Slika 7.4 DEV koji odgovara relacijama u 1. NF-u

Prikažemo li dobiveni model relacijski EV metodom, imamo model podataka na slici 7.4.

Prema DEV-u možemo zaključiti da je svođenje relacije u 1. NF-u zapravo traženje slabih tipova entiteta i definiranje njihove povezanosti s jakim tipom entiteta.

Prevodenjem modela podataka sa slike 7.4 dolazimo do relacijske sheme prikazane u tablicama 7.2 i 7.3.

Može se pokazati kako je proces kreacije DEV-a takav da se svaki složeni ključ, pokuša modelirati kao dva tipa entiteta od kojih je jedan slab tip. Tada ne dolazi do anomalija i osiguran je 1. NF bez provođenja posebnoga postupka normalizacije; odnosno semantičko modeliranje ne zahtijeva kasniju normalizaciju.

7.1.2 Druga normalna forma (2. NF)

Relacija u 1. NF nije bez mana. Prevodenje relacije iz 1. NF-a u 2. NF je potrebno, jer u 1. NF-u nisu riješene sve anomalije kod operacija: dodavanja, brisanja i izmjene.

Promotrimo relaciju Stavka narudžbe definirane kao tablica 7.3 i prikažimo anomalije nekih relacija koje su u 1. NF-u.

Očite anomalije su:

- **dodavanje** - ne možemo dodati novi proizvod dok nemamo stavku narudžbe u kojoj se on pojavljuje
- **brisanje** - podatke o proizvodu gubimo ako izbrišemo jedinu stavku narudžbe u kojoj se proizvod pojavljuje
- **izmjena** - ako se promijeni vrijednost atributa Jedinična cijena nekoga proizvoda moramo je mijenjati na svim narudžbama

Druga normalna forma definira se za sheme relacija koje imaju složeni ključ i koje se nalaze u prvoj normalnoj formi.

Promatra se funkcionalna ovisnost neključnih atributa o cijelome ključu.

Definicija 2. NF-a je:

Shema relacije, koja je u 1. NF-u i ima složeni ključ, je u 2. NF-u, ako su svi njezini neključni atributi potpuno funkcijски ovisni o ključu, odnosno funkcijski ovise o svim dijelovima ključa.

Drugim riječima, ako postoji neki neključni atribut koji je funkcijski ovisan o dijelu ključa, schema relacije nije u 2. NF-u.

Iz navedenoga proizlazi da relacija koja je u 1. NF-u i ima jedan atribut za ključ, je i u 2. NF-u.

Shema relacije Narudžba sa slike 7.4 je, prema tome, u 2. NF-u (ima jednostavan ključ), a za shemu relacije Stavka narudžbe s iste slike to treba provjeriti.

Ispitivanjem funkcijске ovisnosti neključnih atributa o cijelome ključu vidimo da su atributi Naziv_proizvoda i Jedinična_cijena funkcijski ovisni o dijelu ključa, tj. o ključnom atributu Šifra_proizvoda pa prema definiciji ova schema relacije nije u 2. NF-u. Jasno je da Šifra_proizvoda jednoznačno određuje Naziv_proizvoda. Slično vrijedi i za Jediničnu_cijenu. Prema tomu Naziv_proizvoda i Jedinična_cijena ne ovise o Broju_narudžbe i Šifri_proizvoda zajedno, dakle ne ovise o cijelome ključu već o njegovom dijelu.

Da bi shemu relacije doveli u 2. NF moramo izdvojiti neključne attribute, koji funkcijski ovise samo o dijelu primarnoga ključa, u novu relaciju.

Dio primarnoga ključa, o kome izdvojeni atributi funkcijski ovise, postaje ključem nove relacije.

Tablice 7.4 i 7.5 prikazuju "prevedenu" shemu relacije Stavka narudžbe u dvije sheme relacije koje su u 2. NF-u.

C) **Stavka narudžbe** (Broj_narudžbe, Šifra_proizvoda, Količina_proizvoda, Cijena_proizvoda)

Tablica 7.4 Popis stavki narudžbe

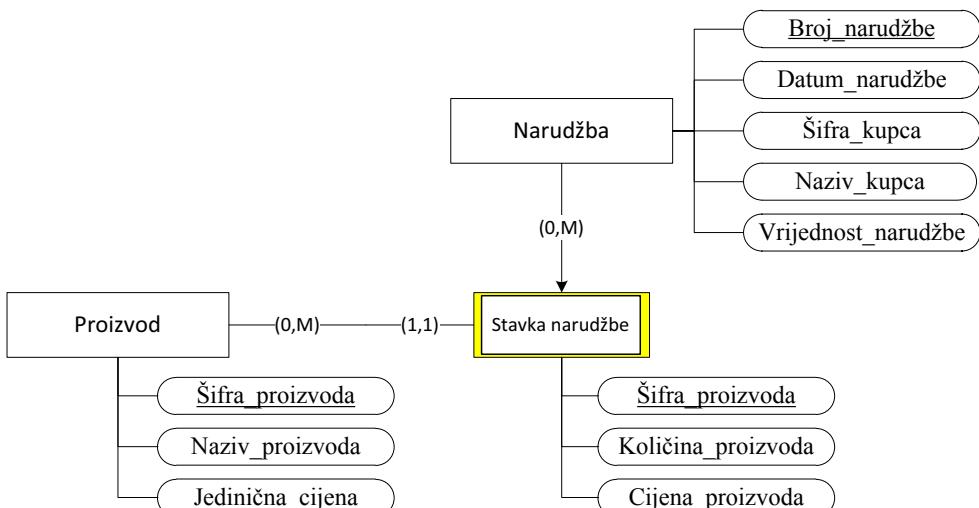
Broj_narudžbe	Šifra_proizvoda	Količina_proizvoda	Cijena_proizvoda
101	2322	1	1000
102	2434	20	500
102	3456	10	10000
102	3457	100	10000
103	2434	400	8000
103	2323	1	30

D) Proizvod (Šifra_proizvoda, Naziv_proizvoda, Jedinična_cijena)

Tablica 7.5 Popis proizvoda

Šifra_proizvoda	Naziv_proizvoda	Jedinična_cijena
2322	Kuh. stol	1000
2323	Prekidač	30
2434	Kabel φ20	20
3456	Brava S4	1000
3457	Matica	100

Prikažimo dobiveni model pomoću DEV-a. Očito je došlo do izdvajanja entiteta zasebnoga tipa iz slaboga tipa entiteta (slab tip entiteta ima složeni ključ).



Slika 7.5 DEV u 2. NF

Može se pokazati da je prevodenje u 2. NF postupak istraživanja postojanja entiteta zasebnoga tipa entiteta iz slabih tipova entiteta i iz agregiranih tipova entiteta.

Svaki jak tip entiteta koji je u 1. NF-u je i u 2. NF-u.

Može se pokazati kako je proces kreacije DEV-a takav da u njemu ne dolazi do postavljanja pojedinačnih pojavljivanja u agregacije i slabe tipove. Time je osigurana visoka normalna forma bez normalizacije.

Prevodenje relacije u drugu normalnu formu je proces izdvajanja 1:M tipova veza iz slabih tipova entiteta (moguće i agregacija), zajedno s entitetima druge strane tipa veze.

Analizom DEV-u uočavamo da je ključ Stavka narudžbe sastavljen od atributa: Broj_narudžbe i Šifra_proizvoda, odnosno od ključnih atributa dvaju raznih tipova entiteta. Ovo znači da je Stavka narudžbe zapravo agregacija, a ne slab tip entiteta. To nismo mogli zaključiti pri prevođenju u 1. NF.

Dakle proces normalizacije ne otkriva potrebu uvođenja slaboga tipa entiteta, odnosno agregacije. Ipak dobro rješenje ne bi bilo uvesti agregaciju i spustiti ključ Šifra_proizvoda u agregaciju. Zašto Šifra_proizvoda nije dobar ključ Stavke narudžbe? Ako na narudžbi postoje dva razna retka koja naručuju isti proizvod, na primjer od različitih dobavljača u različitoj količini, onda model nije dobar opis narudžbenice, jer se ta dva retka ne mogu upisati na istu narudžbenicu. Tada treba dodati ključni atribut Redni_broj_stavke umjesto ključnog atributa Šifra_proizvoda u relaciju Stavka narudžbe.

Pored toga i projektiranja ključa novih relacija uvođenjem novog atributa nije zadatak uobičajenoga procesa normalizacije. Mogli bismo proširiti proces normalizacije iz 0. NF-a u 1. NF i reći:

Ključ nove sheme relacije je sastavljen od ključa početne sheme i jedinstvenog identifikatora grupe koja je izdvojena, kao na primjer redni broj skupine i sl.

Prepostavimo da narudžbenica nema raznih redaka s istim proizvodom.

7.1.3 Treća normalna forma (3. NF)

Neke relacije koje su u 2. NF-u imaju i dalje anomalije. Njih treba prevesti u višu, 3. normalnu formu.

Prikažimo postojanje anomalija na relaciji Narudžba prikazanoj u tablici 7.2. Ta relacija je u 2. NF-u, zato jer je u 1. NF-u i nema složeni ključ.

Anomalije su:

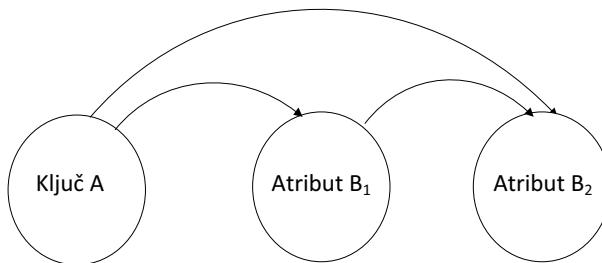
- **dodavanje** - ne možemo dodati novoga kupca bez nove narudžbe
- **brisanje** - brisanjem zadnje narudžbe nekoga kupca gube se podaci o kupcu
- **izmjena** - ako se promijeni naziv kupca moramo ga mijenjati na svim narudžbama

Uzrok problema tumači se tranzitivnim ovisnostima. Definirajmo pojam tranzitivne ovisnosti.

Ako je A primarni (glavni) ključ sheme relacije $R(A, B_1, B_2, \dots, B_n)$ s n atributa, tada za $1 < i < n$ vrijedi $A \rightarrow B_i$ (" B_i je funkcionalno ovisan o A ", odnosno svaki atribut je funkcionalno ovisan o ključu).

Ako postoji $1 < j < n$ takav da vrijedi $B_i \rightarrow B_j$ tada je B_j tranzitivno ovisan o A .

Tranzitivnost je grafički prikazana skupovima na slici 7.6.

Slika 7.6 Tranzitivno funkcionalno ovisan atribut B₂

Atributi B_1 i B_2 su funkcijски оvisni o ključu A . Pošto je atribut B_2 funkcijски оvisan o atributu B_1 onda kažemo da je atribut B_2 tranzitivno оvisan o ključu A preko atributa B_1 .

Definicija 3. NF:

Shema relacije je u 3. NF-u (ponegdje TNF), ako je u 2. NF-u i ako niti jedan neključni atribut nije tranzitivno оvisan o ključu sheme relacije.

Provjerimo sada jesu li naše sheme relacije u 3. NF-u. Za sheme relacija Stavka narudžbe i Proizvod definirane u tablicama 7.4 i 7.5, lako se vidi da ne postoji nikakva tranzitivna оvisnost neključnih atributa o ključu relacije.

Provjeravanjem sheme relacije Narudžba definirane u tablici 7.2, zaključujemo da je atribut Naziv_kupca funkcijски оvisan o atributu Šifra_kupca, pa je prema tome Naziv_kupca tranzitivno оvisan o ključu Broj_narudžbe.

Po pravilima normalizacije takav tranzitivno оvisan atribut moramo izdvojiti u novu shemu relacije. Ključ nove sheme je atribut o kome izdvojeni atributi tranzitivno ovise.

U slučaju relacije Narudžba ključ nove relacije je Šifra_kupca jer o njemu funkcijски оvisi Naziv_kupca.

Tablice 7.6 i 7.7 prikazuju dobivene sheme relacija koje su u 3. NF-u.

Za sheme relacija koje su u 3. NF-u anomalije su riješene.

E) **Narudžba** (**Broj_narudžbe**, **Datum_narudžbe**, **Šifra_kupca**, **Vrijednost_narudžbe**)

Tablica 7.6 Narudžba (3. NF)

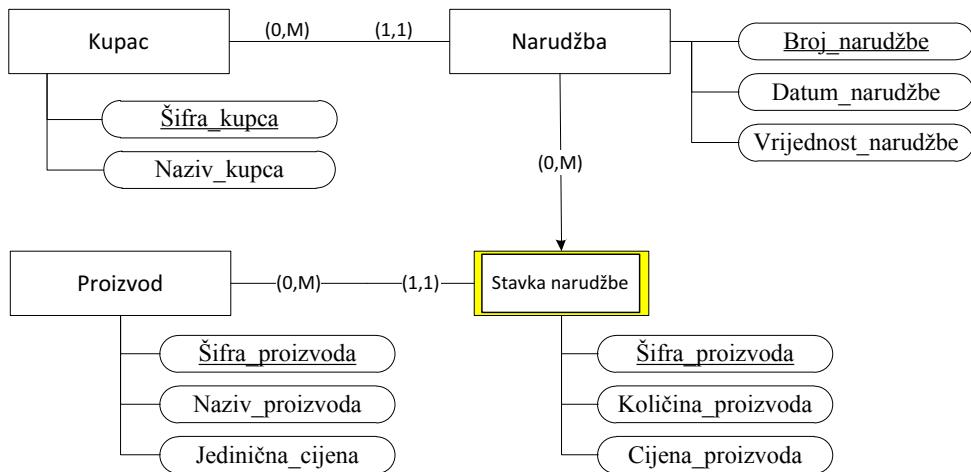
Broj_narudžbe	Datum_narudžbe	Šifra_kupca	Vrijednost_narudžbe
101	10-6-95	1527	1000
102	10-6-95	2023	20500
103	11-6-95	3532	8030

F) Kupac (Šifra_kupca, Naziv_kupca)

Tablica 7.7 Kupac (3. NF)

Šifra_kupca	Naziv_kupca
1527	Jadran
2023	Metal
3532	Crota

Cjelokupni model, svih relacija, njihovih atributa i njihovih međusobnih veza prikazan je na slici 7.7.



Slika 7.7 DEV Naručivanje

Može se pokazati da je prevođenje iz 2. NF-a u 3. NF proces istraživanja postoja-nja entiteta zasebnoga tipa entiteta iz jakih tipova entiteta. Ukoliko različite entitete iz stvarnosti čuvamo u istome tipu entiteta, onda taj tip entiteta pri prevođenju u relacijsku shemu dovodi do relacije koja nije u 3. NF-u.

Zbog ovoga se, pri modeliranju EV metodom, različiti entiteti razdvajaju u različite klase entiteta.

Prevođenje relacije u treću normalnu formu je proces izdvajanja 1:M tipova veza iz jakih tipova entiteta, zajedno s entitetima druge strane tipa veze.

7.1.4 Boyce-Coddova normalna forma (BCNF)

Sve ranije odabране relacije su i u BCNF-u i u 4. NF-u i u 5. NF-u te nam ne mogu poslužiti za daljnji postupak normalizacije, postavit ćemo drugi primjer.

BCNF je stroži oblik 3. NF-a, pa prema tome, svaka shema relacije koja je u BCNF-u je i u 3. NF-u.

Boyce-Coddova normalna forma je prva forma u kojoj se promatruju i uklanaju anomalije ključeva. Prve tri normalne forme uklanjaju anomalije neključnih atributa.

Normalizacija u BCNF-u je važna kada shema relacije ima složeni ključ ili više složenih ključeva koji se djelomično prekrivaju, tj. neki dijelovi (atributi) su im zajednički.

Definicija normalizacije u BCNF-u je:

Ako je neki atribut sheme relacije dio složenoga ključa funkcionalno ovisan o nekom atributu A sheme relacije, tada taj atribut A mora biti ključ.

Pogledajmo to na sljedećem primjeru u tablici 7.8. Relacija Radi_na_projektu je u 3. NF-u.

G) Radi_na_projektu (#Projekta, #Analitičara, #Voditelja)

Tablica 7.8 Radi na projektu

#Projekta	#Analitičara	#Voditelja
123	Mirko Matić	Ivo Matijević
234	Mirko Matić	Marija Slamnig
123	Helga Ban	Ivo Matijević
234	Hrvoje Žunić	Marija Slamnig

Iz sheme relacije Radi_na_projektu vidimo da postoji funkcionalna ovisnost #Voditelja o ključu (#Projekta i #Analitičara). Inače relacija ne bi bila u 1.NF-u. Isto tako postoji i funkcionalna ovisnost #Projekta o #Voditelja, tj. projekt vodi uvijek jedan i samo jedan voditelj. Obrnuto ne mora biti, dakle jedan voditelj vodi jedan ili više projekata.

Iz navedenih funkcionalnih ovisnosti proizlazi tranzitivna ovisnost #Projekta o #Projekta, #Analitičara.

Atribut Voditelj je onaj preko kojeg je ostvarena tranzitivna ovisnost.

Takva tranzitivna ovisnost ključnog atributa (Projekt) preko podskupa neključnih atributa može postojati u 3. NF-u, ali ne i u BCNF-u.

Normalizacija u BCNF-u je izdvajanje neključnog atributa, preko kojeg je ostvarena tranzitivna funkcionalna ovisnost dijela ključa, u novu shemu relacije, koja sadrži i atribut o kome je izdvojeni atribut funkcionalno ovisan.

Rezultat normalizacije početne sheme prikazan je u tablicama 7.9 i 7.10. Obje relacije su u Boyce-Coddovoj normalnoj formi.

H) Projekt_analitičar (#Projekta,#Analitičara)

Tablica 7.9 Projekt_Analitičar

#Projekta	#Analitičara
123	Mirko Matić
234	Mirko Matić
123	Helga Ban
234	Hrvoje Žunić

I) Voditelj (#Voditelja,#Projekta)

Tablica 7.10 Voditelj

#Voditelja	#Projekta
Ivo Matijević	123
Marija Slamnig	234
Ivo Matijević	123
Marija Slamnig	234

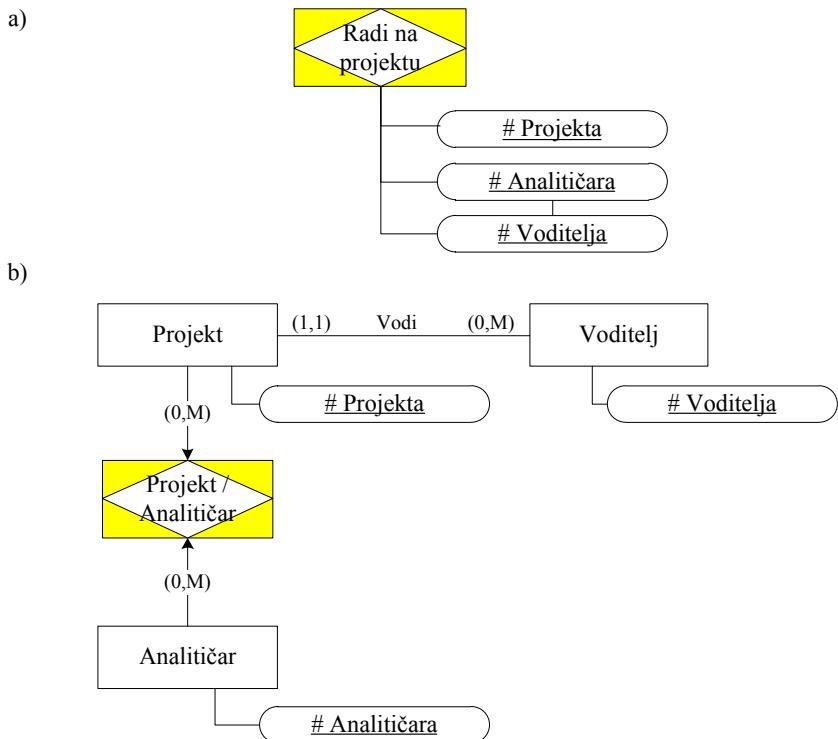
Objasnimo početnu i ciljnu shemu pomoću DEV-a. Početni i ciljni DEV je prikazan na slici 7.8. Slika 7.8a prikazuje početni model, a 7.8b konačni model uz dopunu cjelovitosti, odnosno postojanje svih jakih tipova entiteta koji tvore agregacije i veze.

Možemo zaključiti da početni model promatran metodom EV-a nema smisla jer predstavlja agregaciju bez komponenti agregacije. Sigurno se radi o višestrukoj agregaciji. Takve su agregacije zabranjene. Relacijski model ne vidi semantiku stvarnog svijeta te su u njemu dopuštene sve relacije.

Prema DEV-u na slici 7.8 b zaključujemo da ciljni relacijski model sadrži agregaciju Projekt/analitičar ponovo bez komponenti agregacije. Kada bi se u Projekt/analitičar pojavio bilo koji atribut komponenti agregacije, relacija ne bi bila u 3. NF-u (na primjer Datum početka rada, Planirano dana rada i sl.).

Relacija Voditelj prema pravilu normalizacije u BCNF-u je ispravna. Ona je sa stajališta znanja o stvarnome svijetu pogrešna. Uobičajeno je da jedan projekt vodi samo jedan voditelj, ali ne mora biti i obrnuto. Ne mora jedan voditelj voditi samo jedan projekt. Možemo se pitati kamo ćemo s podacima onoga voditelja koji vodi dva i više projekata.

Do ove je poteškoće došlo zbog početne hipoteze da je ključ relacije Radi_na_projektu složen od atributa #Projekta i #Analitičara te da je atribut #Voditelja funkcionalno ovisan o ključu.



Slika 7.8 DEV početne i krajnje relacijske sheme, s uklonjenom anomalijom krajnje sheme

Popis normalnih formi, njihovi autori i vrijeme nastanka prikazani su u tablici 7.11.²⁰

Tablica 7.111. Pregled normalnih formi

Normalna forma	Autor
Prva (1NF)	Two versions: E.F. Codd (1970), C.J. Date (2003)
Druga (2NF)	E.F. Codd (1971)
Treća (3NF)	E.F. Codd (1971)
Boyce-Codd-ova (BCNF)	Raymond F. Boyce and E.F. Codd (1974)
Četvrta (4NF)	Ronald Fagin (1977)
Peta (5NF)	Ronald Fagin (1979)
Domain/key (DKNF)	Ronald Fagin (1981)
Šesta (6NF)	C.J. Date, Hugh Darwen, and Nikos Lorentzos (2002)

²⁰ http://en.wikipedia.org/wiki/Database_normalization

Sama je normalizacija kao procedura uzaludna jer su njezine hipoteze pogrešne. Svako postavljanje hipoteze o ključu na osnovi slučajnog odabira ključa i njegova provjera normalizacijom nije prihvatljiva praktičaru. Općenito je normalizacija naporanja, dugotrajna, neplodna, nepraktična i nesigurna. Ne koristi se u praksi.

Najvažnije je ipak zaključiti: uz semantičko modeliranje normalizacija podataka je nepotrebna.

Međutim, neophodno je poznavati problem normalizacije da bismo bili svjesni ispravnosti modela i ispravnosti pristupa modeliranja podataka. Zbog toga smo i razjasnili što je normalizacija, koje forme postoje, kako ih praktično možemo ostvariti bez posljedica anomalija. Najbolji način prikaza svih mogućih vrsta grešaka pri modeliranju podataka daju nam normalne forme. Na nekim intervjima za posao mogu se očekivati pitanja pa i primjeri za normalizaciju. To su važni argumenti zašto projektanti baze podataka trebaju znati normalizaciju. Drugo je pitanje koje će se metode koristiti za oblikovanje baze podataka.

Četvrta i peta normalna forma predlažu popravljanje pogrešnih početnih trostrukih i višestrukih agregacija. Prema EV metodi dopuštene su samo binarne veze i binarne agregacije koje se dalje mogu kombinirati. To nas štiti od pogrešnih modeliranja.

Općenito, možemo zaključiti: svaki DEV kreiran uz navedena ograničenja EV metode i primjenu pravila prevođenja dovodi se u relacijsku shemu baze podataka koja se nalazi u najvišoj normalnoj formi.

Učešće korisnika u validaciji model podataka u praksi je sasvim razumno, ali ne i očekivanje da se korisnici mogu uključiti u normalizaciju.

Dodatni argument protiv normalizacije, a u prilog EV oblikovanju je i da se relativno jasno uočavaju standardizirani dijelovi (uzorak, standardni dio model, engl. patterns) modela podataka, a što je gotovo nemoguće uvidjeti iz relacija.

8 PROCES KONCEPTUALNOG OBLIKOVANJA PODATAKA

Model podataka je dio modela koji prikazuju cjelokupni informacijski sustav. Jedan od načina stvaranja sheme baze podataka je primjena postupka normalizacije pretvarajući tablice iz nižih u više normalne forme. Puno jednostavniji, brži i ljudski bliži postupak je izrada DEV-a metodom EV-a i prevođenje DEV-a u relacijsku shemu baze podataka. Postupak prevođenja je opisan u ranijim poglavljima. Proces kreacije DEV-a započinje analizom izvora znanja o IS-u iz ranijih modela i dokumentata sustava i transformacijom istih u koncepte metode EV-a.

Analizom procesa modeliranja podataka možemo ustanoviti da postoje razne vrste strategija (ideja, tehnika) za modeliranje podataka. Proces razvoja modela je određen: formom modela (specifikacijom), primjenjenom metodom za modeliranje podataka, primjenjenom specijaliziranoj metodologijom, postojanjem baze podataka, ograničenjima, pogledima na podatke, životnim ciklusom objekata, raspoloživim dokumentima, razinom znanja korisnika i sl. Na osnovi toga mogu se definirati različite strategije modeliranja podataka.

Strategije modeliranja DEV-a su (Pavlić, 1992.):

1. modeliranje apstrakcijom koncepata
2. modeliranje na osnovi strukture
3. modeliranje širenjem tipova entiteta i veza
4. modeliranje integracijom pogleda
5. modeliranje na osnovi općih tipova entiteta
6. modeliranje nominalnom grupom
7. modeliranje na osnovi jezičnoga pojma (analizom rečenica)
8. modeliranje na osnovi ograničenja DEV-a
9. modeliranje na osnovi postojeće baze podataka
10. modeliranje na osnovi sadržaja EV specifikacije
11. modeliranje na osnovi faza životnoga ciklusa entiteta
12. modeliranje na osnovi dijagrama toka podataka
13. modeliranje na osnovi kardinalnosti entiteta.

U literaturi se navode i drugi brojni različiti pristupi (tehnike, strategije) modeliranju podataka:

1. modeliranje na osnovi izvještaja
2. modeliranje na osnovi programskog koda s tradicionalnim fajlovima i tabelama
3. modeliranje na osnovi postojećeg, ali neodgovarajućega modela (u različitim notacijama na primjer na osnovi objektnih modela, petri Merza, ili automata)
4. modeliranje na osnovi patterna (Blaha, 2010.)
5. modeliranje na osnovi standardiziranih XML Sema
6. modeliranje na osnovi Standardnih (generaliziranih) modela podataka (nafntna industrija, trgovina, razne industrijske asocijacije)
7. modeliranje adaptacijom Univerzalnih Modela (engl. Universal Model)
8. modeliranje na osnovi meta modela kod model-driven razvoja tehničkoga softvera ili znanstvenih modela prirodnih znanosti (kao DNK)
9. modeliranje na osnovi „fazi“, logičko-deduktivnih i temporalno/spatialnih modela
10. modeliranje na bazi empirijske Ontologije (za vrlo složene integrirane sustave)
11. modeliranje na osnovi hipoteza iz rudarenja podataka

Pojedini projektanti koriste jednu ili više tehnika prilagođenu vlastitim potrebama i mogućnostima. U ovoj knjizi nećemo izlagati svaku strategiju/tehniku zasebno. Neke strategije ćemo uklopiti u metodu VATEK koja predstavlja proceduru modeliranja podataka za klasu poslovnih aplikacija kada smo u mogućnosti s korisnicima načiniti model procesa i prikupiti dokumentaciju o poslovanju.

VATEK je skraćeno od važnih koncepata pridruženih pojedinim aktivnostima modeliranja podataka i to: Vrijednost, Atribut, Tip Entiteta i Ključ.

Prije modeliranja podataka po specijaliziranoj metodologiji MIRIS i drugim metodologijama modeliraju se procesi i crta DTP. U okviru modela procesa prikupljeni su popunjeni dokumenti (tokovi podataka) poslovnoga sustava s kojih se mogu uočiti podaci.

Predložimo aktivnosti procesa modeliranja podataka metodom VATEK vezanoga za svaki podatak uočen u sustavu. Aktivnosti modeliranja izvode se za svaki pojedinačni podatak u sustavu. Ako smo ustanovili odgovarajuće koncepte za prvi elementarni podatak, odabiremo sljedeći elementarni podatak (vrijednost) s dokumenta, odnosno iz sloga datoteke. Za njega ustanovljavamo, na isti način, odgovarajuće koncepte prolazeći prethodnim koracima.

Taj proces se nastavlja do posljednjeg elementarnoga podatka na toku podataka iz DTP-a. Poslije završetka analize jednog toka podataka, odabiremo sljedeći. Taj se proces nastavlja do potpunog iscrpljenja svih tokova podataka iz DTP-a.

Pokažimo prirodnost takvoga pristupa naročito u prvim fazama modeliranja i detaljnije rastumačimo svaku pojedinu aktivnost.

Aktivnosti (koraci) modeliranja podataka po VATEK-u su:

1. uočiti podatak
2. odrediti tip vrijednosti (odakle ta vrijednost dolazi)
3. imenovati atribut čija je to vrijednost
4. uočiti entitet koji ima taj atribut
5. klasificirati entitete u tip entiteta
6. pronaći ključ tipa entiteta
7. odrediti brojnosti tipa veze
8. odrediti brojnosti atributa
9. odrediti svojstva atributa
10. dodati izvedene atrbute
11. dekomponirati složene tipove veza
12. specijalizirati podtipove tipa entiteta
13. definirati specijalne tipove veza i slabe tipove entiteta
14. definirati povratne tipove veza

Opišimo pojedine aktivnosti (korake) detaljnije.

8.1 Uočiti podatak

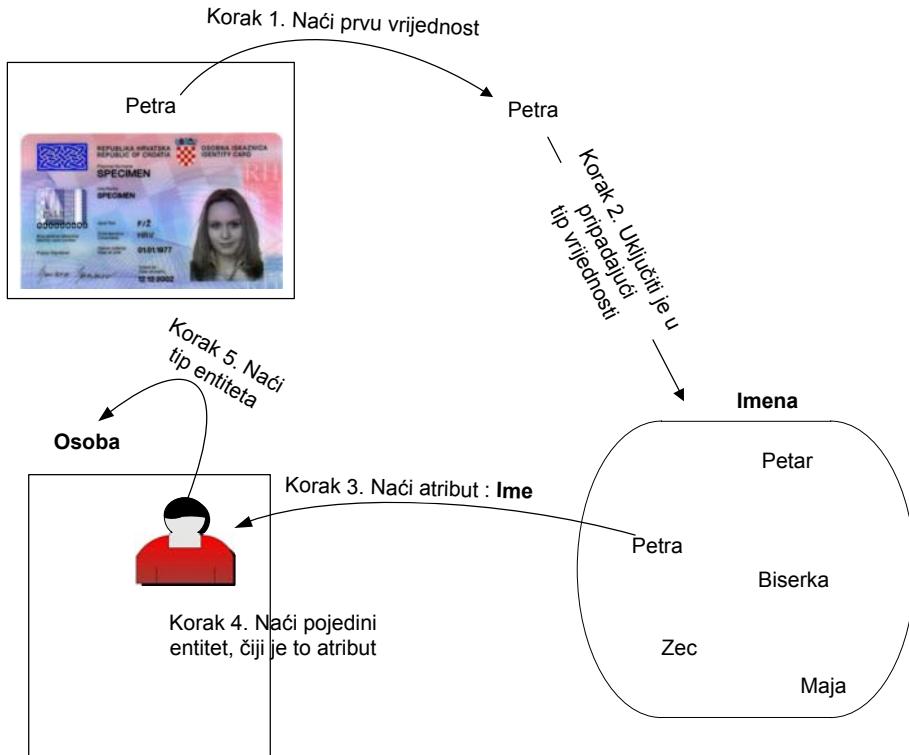
Pretpostavimo da su prikupljeni tokovi podataka i popunjeni dokumenti s podacima. Naš zadatak je uočiti neki podatak na dokumentu (vidi sliku 8.1.).

Tokovi podataka mogu biti dokumenti, datoteke, baze podataka i sl. Bez obzira na nosioce podataka, odabire se jedan dokument koji sadrži niz elementarnih i konkretnih podataka iz sustava. Na primjer, obrazac dokumenta za upis članova dobrovoljnoga vatrogasnog društva na WWW ima u niz polja upisane elementarne (varijabilne) podatke kao: Upiši ime: Marko, Upiši Prezime: Golem, Datum upisa: 01.01.2056. i dr.

Sličan postupak analize provodi se i za datoteku. Datoteka ima jedan ili više formaliranih slogova. U datoteci se slog sastoji od imenovanih polja. Za svaki slog odbiremo bar po jedno konkretno pojavljivanje.

Započinjemo analizu podataka od vrha dokumenta na desno i odozgo prema dolje. Dokument ima niz formalnih i fiksnih znakova, crta, simbola koji daju oblik dokumentu i uvode red među podacima. Oni nas ne zanimaju. To su linije, simboli organizacije, oznaka proizvođača dokumenta, tekstualni opis koji daje interpretaciju podataka, a nalazi se ispred ili iza vrijednosti. Zainteresirani smo za promjenjive podatke na dokumentu (za vrijednost).

Prvi podatak na koji naiđemo je neka vrijednost i ona zahtijeva da odgovorimo na pitanje: kojem konceptu pripada.



Slika 8.1 Faze izvođenja aktivnosti procesa modeliranja podataka

Kada na njega odgovorimo (kroz navedene aktivnosti), tražit ćemo sljedeći podatak na dokumentu idući s lijeva na desno i odozgo prema dolje. Tako sve dok ne analiziramo sve podatke na dokumentu.

8.2 Odrediti tip vrijednosti (odakle ta vrijednost dolazi)

Svaki podatak na dokumentu je konkretno pojavljivanje vrijednosti u nekom tipu vrijednosti. To je u osnovi definicije koncepcata metode EV-a.

Već je iz samoga podatka jasno je li to: broj, tekst, specijalni znaci, mješavina znakova; približno jasna je veličina koju zauzima ta vrijednost. Kako odrediti tip vrijednosti?

Do tipa vrijednosti dolazi se klasifikacijskom apstrakcijom pojedinačnih vrijednosti na dokumentu (u datoteci). Skup svih dokumenata iste vrste (na primjer osobne iskaznice) ima na sebi istovrsne podatke (na primjer imena Marko, Petra, Veljko, i dr.) na istome mjestu na dokumentu, a sva pojavljivanja tih vrijednosti čini tip vrijednosti. U datoteci tip vrijednosti je sadržaj jednoga polja datoteke bez ponavljanja vrijednosti.

Moguće je da za neku vrijednost ne možemo ustanoviti kome tipu vrijednosti pripada; tada je ispuštamo iz razmatranja i odlažemo za analizu u drugoj iteraciji.

8.3 Imenovati atribut čija je to vrijednost

Kako ustanoviti ime atributa za vrijednost iz tipa vrijednosti uočenu na dokumentu?

Ako je promatrana vrijednost u datoteci, tada je ime stupca datoteke ime atributa.

Na dokumentu tekst, koji je neposredno ispred ili iza polja s vrijednošću, daje tumačenje te vrijednosti pa se ime atributa definira prema tome tekstu. Na primjer "Marko" je vrijednost atributa „Ime“ s dokumenta na kome stoji:

"Upiši ime: Marko"

a vrijednost imena "Marko" je upisana, na primjer rukom. Riječ "Upiši" ukazuje na operaciju nad podacima, što nije predmet modeliranja strukture podataka. Riječ "Ime" govori o sadržaju polja, o tome koji su podaci upisani, što se upisuje. Tako je odabran naziv atributa Ime semantički povoljan.

Naziv polja datoteke je u većini slučajeva dobro definiran te odgovara i kao ime atributa. Mijenja se ako je polje semantički široko. Na primjer naziv polja "Datum" mijenjamo u naziv atributa "Datum računa". Mijenja se i ako je šifriran kao "DAT2" u "Datum rođenja". Mijenja se i radi jednoznačnog imenovanja.

Moguće je da na DEV-u postoji nekorisni atribut (atribut za koji svi entiteti imaju koji ima nul vrijednost, tj. koji se ne koristi). To su sva ona rezervirana polja na obrascu ili u datoteci koja se ne popunjavaju. Njega isključujemo iz modela. Ako se očekuje da će on ipak u budućnosti biti potreban, uključujemo ga s time da je njemu dopuštena nul vrijednost. Projektant dokumenta (datoteke) je nekada prepostavio da će podatak biti potreban, ali to se tijekom vremena može promijeniti i sustav funkcioniра i bez tih podataka. Postoje slučajevi kada se podatak rijetko pojavljuje, ali se ipak ponekada pojavi. Ovo se doznaće analizom svih pojedinačnih pojavljivanja (uzorkovanjem, sampliranjem) i intervjuiranjem o mogućem pojavljivanju vrijednosti u tome polju. Takav podatak je vrijednost (iz poznate domene) atributa određenoga tipa, i uključuje se u model podataka.

8.4 Uočiti entitet koji ima taj atribut

Do sada imamo ustanovljeno sljedeće: neku vrijednost, ime tipa vrijednosti u koju ta vrijednost pripada i imenovan atribut. Postavlja se pitanje čiji je to atribut koji je otkriven u prethodnome koraku? Odgovor je: entitet ili agregacija. Do tog odgovora dolazimo na osnovi pitanja čiji (ili od čega) je to atribut, kao što je prikazano u tablici 8.1.

Kojeg je entiteta (agregacije) to atribut? Mora postojati entitet i tip entiteta (agregacija i tip agregacije) s tim atributom, i to tako da se prethodna vrijednost pridružuje tim atributom nekom entitetu (agregaciji).

Tablica 8.1 Prepoznavanje entiteta

Vrijednost	Atribut	Pitanje (Čiji?, Čega?)	Odgovor	
Marko	Ime	Čije Ime?	Jednoga radnika	Kadar
Sroki 60	Adresa	Čija adresa?	Jednoga stana	Stan
MaxyT	Naziv	Naziv čega?	Jednoga projekta	Projekt
32	Količina	Čija količina?	Stavke narudžbe	Roba/Narudžba

Rezultat ovoga procesa je pojedinačno pojavljivanje entiteta ili agregacije. Još nije određeno je li u pitanju atribut entiteta ili atribut agregacije. Za sada ga razmatramo kao tip entiteta, dok ne odredimo ključ. Ako je ključ složen od ključeva drugih tipova entiteta, onda se radi o agregaciji (kao četvrti redak u tablici).

Pored te dvojbe mogu se ustanoviti i druge:

- atribut, čini se, pripada raznim tipovima entiteta
- atribut je ranije pridružen drugome tipu
- atribut ne pripada nijednome tipu
- atribut je pridružen jednome tipu, ali pojavljivanja tog tipa imaju istovremeno više vrijednosti tog atributa
- atribut je ključ toga tipa entiteta ili je ključ nekoga drugog tipa entiteta

Ako atribut pripada raznim tipovima, onda ne pripada nijednom od njih, već medu njima postoji agregacija kojoj pripada. Moguće je da se uočava vanjski ključ. Ako se čini da je ključni atribut nekoga tipa entiteta također i atribut drugoga tipa, onda to ukazuje da postoji tip veze medu tipovima entiteta. Atribut uvijek pripada jednom i samo jednome tipu.

Ako je atribut već pridružen nekome tipu, provjerava se ispravnost prethodne odluke.

Ako od postojećih tipova atribut ne pripada ni jednome tipu, tada mora postojati neki novi tip kome atribut pripada.

Višezačan atribut nije atribut toga tipa entiteta, već se uvodi slab tip entiteta ili tip veze s novim tipom entiteta čiji je to atribut.

Za svaki atribut upitamo se je li kandidat za ključ ili je možda ključ drugoga tipa. Ako je ključ drugog, tada nije atribut nego postoji tip veze s drugim tipom.

Te dileme je neophodno razriješiti.

8.5 Klasificirati entitete u tip entiteta

Kada smo ustanovili jedno konkretno pojavljivanje entiteta, onda možemo klasifikacijskom apstrakcijom naći još sličnih entiteta i klasificirati ih u tip entiteta.

Rezultat je utvrđivanja pojedinačnih pojavljivanja entiteta i veza nalaženje tipova entiteta i tipova veza njihovom klasifikacijom. Rijetko se dogodi da postoji samo jedno pojavljivanje nekoga tipa, iako je i to moguće. Skup svih pojavljivanja čini tip entiteta.

Dobiveni rezultati se crtaju na DEV. Rezultat je skup tipova entiteta i tipova veza. Atributi se također crtaju uz tipove. Dodajući nove atribute testira se prethodna teza o obliku modela.

Za ustanovljavanje tipa mogu poslužiti dva pravila:

Pravilo 1. Svi fizički objekti u sustavu su tipovi entiteta na DEV-u (ne i obrnuto).

Pravilo 2. Svi šifarnici šifriraju jedan tip entiteta, a sama šifra je ključni atribut tipa entiteta.

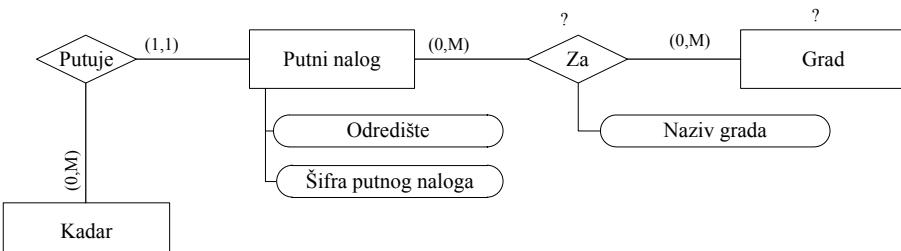
Ako nismo odlučili pripada li atribut nekome tipu entiteta ili agregaciji, taj atribut ćemo pokušati uključiti u model u drugoj iteraciji. Na taj način nastaje lista nepridruženih atributa.

8.6 Pronaći ključ tipa entiteta

Za svaki tip entiteta, postavlja se pitanje: Ima li taj tip entiteta ključ?

Ako među atributima ne postoji nijedan atribut koji je ključ tog tipa entiteta, postavlja se pitanje i otvara problem kako jedinstveno identificirati entitete tog tipa. Taj tip entiteta je potrebno ili šifrirati ili ukloniti iz modela podataka.

Na primjer Za DEV “Putni nalog” tip entiteta “Grad” nema ključ. Atribut “Naziv grada” može se preimenovati kao atribut, na primjer “Odredište” tipa entiteta “Putni nalog” ako je tip veze s kardinalnostima (0,1):(0,M) (slika 8.2). Ovaj proces pomicanja atributa može se nastaviti i ispitati ispravnost novo nastale situacije, odnosno je li Odredište viševrijednosan atribut. Ako jest, tj. ako ima više odredišta za jedan putni nalog, onda se Naziv grada može staviti kao atribut tipa veze Za, a sam tip veze proglašiti slabim tipom entiteta od Putnoga naloga. Tada nam ne treba Tip entiteta Grad u modelu podataka. Ako uvedemo ključ Grada, onda će on postati relacija, a tip veze Za će postati Agregacija.



Slika 8.2: Model podataka Putni nalog

Razmotrite što bi bilo ako bi pridružili atribut "Odredište" vezi "Putuje".

Ako različiti tipovi entiteta imaju isti ključ, tada se možda tu radi o jednom jedinom tipu. Ovo je čest slučaj u praksi koji nastaje kada je netko podijelio entitete u dvije klase po nekome svojstvu (kao na primjer Muškarci i Žene).

8.7 Odrediti brojnost tipa veze

Utvrđuje se prema definicijama danim pri opisu ograničenja. Neka u vezi V su-djeluju tipovi entiteta X i Y . Brojnost tipa entiteta Y je nula (0), jedan (1), dva (2), ..., mnogo (M) ako se nula (0), jedan (1), dva (2), ..., mnogo (M) pojavljivanja tipa entiteta Y pojavljuje (povezuje) u tipu veze V .

Možemo definirati pitanja koja se postavljaju i na osnovi kojih se jednoznačno određuju kardinalnosti i to za donju i gornju granicu zasebno.

Određivanje donje granice

Uzmimo jedno, bilo koje pojedinačno pojavljivanje x iz tipa entiteta X . S koliko se najmanje (donja granica) pojavljivanja entiteta y iz tipa entiteta Y povezuje x ? Postavimo isto pitanje za svako x iz X ; najmanja dobivena vrijednost kao odgovor je donja granica (minimum).

Određivanje gornje granice

Uzmimo jedno bilo koje pojavljivanje x iz tipa entiteta X . S koliko se najviše (gornja granica) pojavljivanja entiteta y iz tipa entiteta Y povezuje x ? Postavimo isto pitanje za svako x iz X ; najviša dobivena vrijednost kao odgovor je gornja granica (maksimum).

Ako je donja granica 0, kažemo da je tip entiteta **parcijalno povezan**, jer svi entiteti ne moraju participirati u vezi.

Ako je donja granica 1, tip entiteta je **totalno povezan** jer svi entiteti iz tipa X participiraju u vezi.

Ako je gornja granica 0, tada ne postoji veza između tih objekata i tip veze se ne crta na model DEV.

Jedan tip veze V povezuje dva tipa entiteta: X i Y . Prema tome postoje dva moguća preslikavanja (koja zovemo uloge), preslikavanje

$F: X \rightarrow Y$ i $G: Y \rightarrow X$.

Oba preslikavanja imaju gornju i donju granicu koja se određuje pomoću predloženih pitanja i prikazuje pomoću uređenog para (MIN,MAX).

Utvrđene granice preslikavanja potvrđuju model. Pri određivanju granica može se ustanoviti da veza nije semantički opravdana te korigirati model.

8.8 Odrediti brojnosti atributa

Brojnosti atributa su definirane s dva para preslikavanja (slično kao i tip veze), preslikavanjem od tipa entiteta k tipu vrijednosti, i obrnuto. Svakom preslikavanju se određuje gornja i donja granica i prikazuje u obliku uređenoga para (MIN, MAX).

Pitanja koja se postavljaju i na osnovi kojih se jednoznačno određuju brojnosti, i to za donju i gornju granicu zasebno, identična su pitanjima za određivanje brojnosti tipa veze te ih ovdje ne iznosimo.

Dopuštene brojnosti preslikavanja od tipa entiteta k tipu vrijednosti su tipa (1,1). Ako su brojnosti tipa (1,M), tada se takav atribut naziva viševrijednosni atribut i on se pretvara u tip entiteta.

Obrnuto, brojnosti preslikavanja od tipa vrijednosti k tipu entiteta mogu biti bilo kakve, na primjer (0,M). Ali, ako je brojnost toga preslikavanja (1,1), tada je atribut kandidat za ključ tipa entiteta.

8.9 Odrediti svojstva atributa

Pored brojnosti preslikavanja za svaki se atribut određuju i svojstva (osobine, atributi) atributa koja zahtijevaju programski jezici i SUBP. Tako treba odrediti:

- tip atributa (slova, brojevi, datumi i sl.)
- dužinu vrijednosti atributa
- broj decimalnih mjesta za brojeve (ovisno o tipu atributa)
- dozvoljenost nul vrijednosti
- indeksiranje po atributu
- dopuštene vrijednosti atributa

Dobivene vrijednosti se upisuju u obrazac prikazan u tablici 6.1.

8.10 Dodati izvedene atribute

Utvrđuju se atributi koji nisu bazni atributi, već se mogu izvesti iz postojećih atributa pomoću matematičkih izraza.

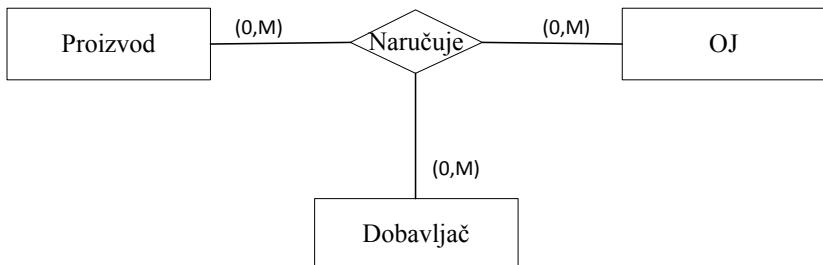
8.11 Dekomponirati složene tipove veza

Moguće je da u modelu bude ustanovljen **višestruki tip veze** (trostruka, četverostruka, ...). Takvi tipovi veza raščlanjuju se po pravilima:

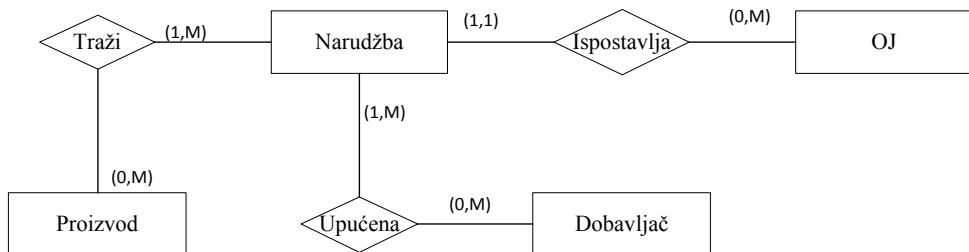
1. ili kao višestruka agregacija (opisano kod agregacije)
2. ili se pronalazi tip entiteta s ključem u središtu višestrukog tipa veze i on se povezuje s više binarnih tipova veza

Uzmimo na primjer trostruki tip veze koja se raščlanjuje (Slika 8.4) tako da:

- tip veze Naručuje postaje tip entiteta Narudžba,
- linija spoja (uloga) tipa entiteta i tipa veze postaje tip veze pri čemu novi tipovi veze mogu biti raznih brojnosti.



Slika 8.3 Tip veze Naručuje



Slika 8.4: Zamjena trostrukih veza

8.12 Specijalizirati podtipove tipa entiteta

Za svaki tip entiteta valja odgovoriti na pitanje, ima li dani tip entiteta podtipove? Moguće je da su definirani neki, ali ne i svi podtipovi, ili da nije definiran nijedan podtip.

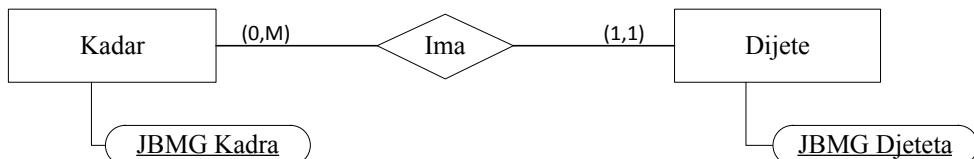
Analiza raspodjele semantike među podtipovima i nadtipovima, te semantička razlika među podtipovima istog nadtipa, odnosno postupkom specijalizacije i generalizacije potvrđuje se i razvija teza oblika generalizacijskoga stabla. Za tip entiteta koji nema podtipove postavlja se pitanje daju li se atributi toga tipa razmjestiti u podtipove i ima li dio atributa nul vrijednosti.

8.13 Definirati specijalne tipove veza i slabe tipove entiteta

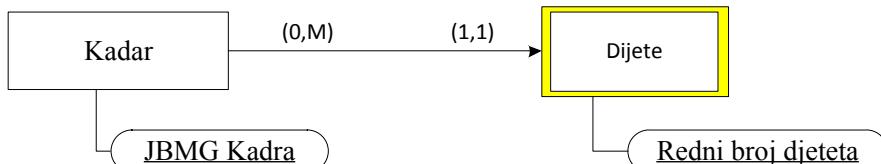
Neke su od veza tipa (1,1):(0,M) možda specijalne. Vrsta specijalnosti (I, E, I&E) nije bitna jer se jednako manifestira i ne iskazuje različito na DEV-u. Za svaki tip veze 1:M po gornjoj granici trebamo upitati: je li to specijalni tip veze (I, E, I&E) koji povezuje slab i jak tip entiteta. To se radi u slučaju kad nije moguće identificirati ključ nekoga tipa entiteta bez ključa drugoga tipa ili kada postojanje jednoga tipa ovisi o postojanju drugog, ili oboje.

Kako to prepoznati?

Oba ova slučaja svode se na činjenicu da želimo u modelu stvoriti specijalan odnos (vezu) između dvaju tipova entiteta i to takav da jedan ovisi o drugom, bez obzira na uzrok - tip specijalnosti. To je vanjska činjenica koju utvrđuje model. Do te činjenice može se doći iz uobičajenih veza gdje nije uočena specijalnost i zavisan odnos, a to se želi. Razlog neuočavanja može biti imenovanje veze semantički jasnim imenom i postojanje ključa zavisnoga tipa entiteta. Na primjer, imamo model kao na slici 8.5, a zapravo želi se model kao na slici 8.6.



Slika 8.5: Tip veze Ima



Slika 8.6: Specijalni tip veze ne imenuje se

Ta su dva modela semantički identična, ali će implementacija IS-a biti različita. Pitanje je može li svaka veza (0,M):(1,1) biti prikazana kao specijalna. Projektant odlučuje o relevantnosti informacija i bira ispravan model.

Postavlja se i pitanje što bi bilo da su kardinalnosti tipa (0,M):(0,1). Prevođenjem takve veze u specijalnu, mijenjao bi se model jer sva pojavljivanja slabih tipova

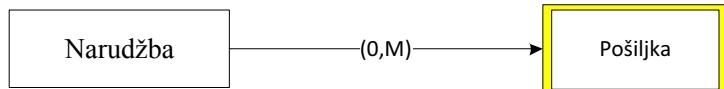
objekta participiraju u jednoj i samo u jednoj vezi. Izgubili bismo informacije o povlačenjima koje ne participiraju u vezi, odnosno promijenili bismo kardinalnosti preslikavanja.

Specijalni tip veza Prethodi ima brojnost (1,1) i znači da Pošiljka zavisi o prethodnom postojanju Narudžbe (Slika 8.7).



Slika 8.7: Tip veze Prethodi

Ako je ključ Pošiljke poželjan tada se DEV ne mijenja, a pošiljke želimo identificirati preko ključa Narudžbe, tada se to prikazuje na DEV-u kao na slici 8.8.).



Slika 8.8: Specijalna zavisnost

8.14 Definirati povratne tipove veza

Za niz tipova entiteta poželjno je imati povratnu vezu kako bi se omogućila hijerarhija odnosa. Na primjer skladišta, proizvoda ili organizacijske strukture. Za svaki tip entiteta postavljamo pitanje: treba li taj tip povratnu vezu.

9

PRAKTIČNI PRIMJERI MODELIRANJA PODATAKA

U ovome je poglavlju dan primjer modela podataka za niz različitih dokumenta. Neki dokumenti služe za prikupljanje podataka (prijava štete) i unos istih u bazu podataka, neki mogu biti generirani iz baze podataka kao izvještaji (primka), neki se izdaju kao uvjerenje (OI, receipt), neki se mogu i ukinuti (indeks), a neki ne moraju imati odgovarajući papirnati oblik već samo e-oblik (narudžbenica). Neovisno o tome za sve podatke je potrebno načiniti model podataka, a on će biti istog oblika bez obzira dolazi li atribut s izvještaja ili s ulaznoga dokumenta.

U većini primjera prikazan je dokument iz nekoga poslovnog sustava popunjeno podacima, dan je kratki opis dokumenta prikazom njegove funkcije i osnovnih podataka na dokumentu. Za dokument je nacrtan DEV i ponekad taj model podataka preveden u relacijsku shemu baze podataka.

9.1 Osobna iskaznica

Osobna iskaznica (skraćeno OI) je dokument koji izdaje policija svakome građaninu, a služi za jedinstveno identificiranje osobe, kako bi ista mogla ostvariti svoja prava u društvu (podići novac u banci sa svog računa, prodati svoje vlasništvo, ...). Na osobnoj iskaznici nalazi se slika koja je važna za prepoznavanje te osnovni podaci o osobi, izdavanju dokumenta, datumu važenja, ali ne i ključni identifikatori osobe kao OIB (osobni identifikacijski broj u RH) ili JMBG (jedinstveni matični broj građana) koji se iz sigurnosnih razloga ne nalaze na OI.

Na osobnoj iskaznici (OI) nalaze se sljedeći podaci: naziv države, ime i prezime osobe, spol, državljanstvo, datum rođenja, datum vrijedi do, potpis, broj OI, adresa stanovanja (prebivalište), policijska uprava koja izdaje OI, datum izdavanja, županija u kojoj je smještena policijska uprava, status važenja OI (V-važeća, N-poništена).



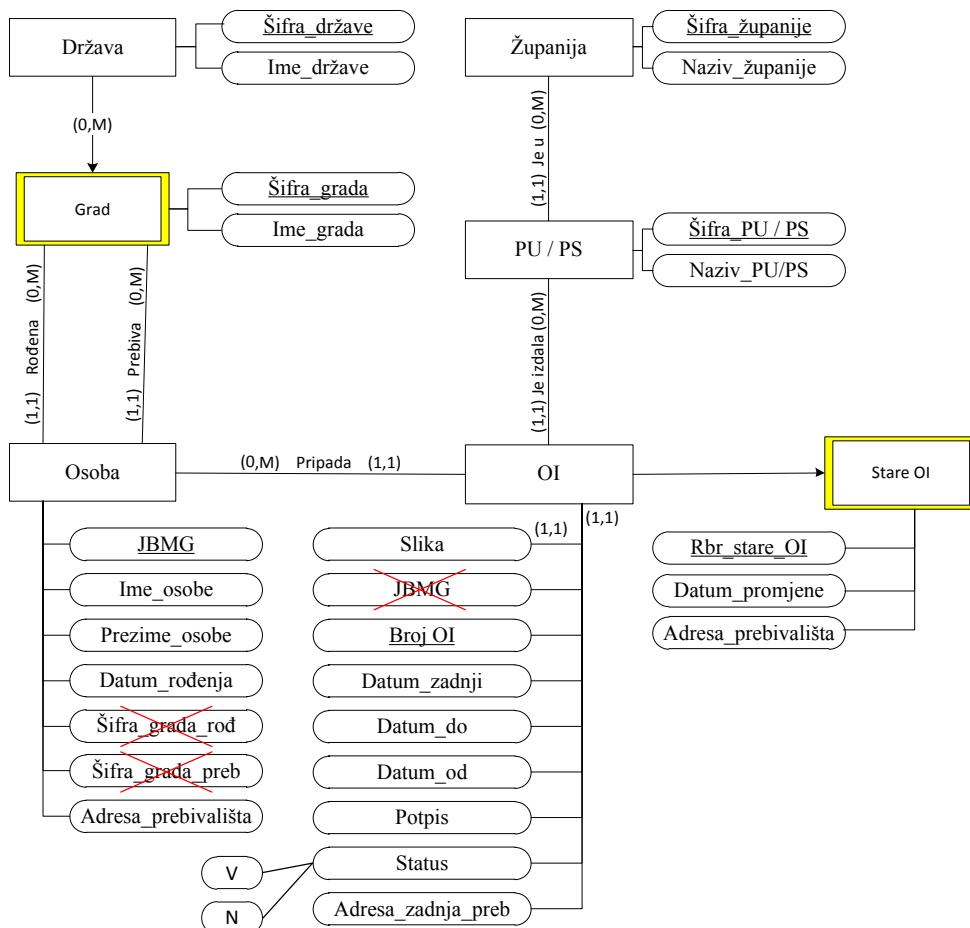
Slika 9.1 Osobna iskaznica (izvor: www.hak.hr, 12.04.2010.)

Zadatak je nacrtati DEV sa svim podacima na osobnoj iskaznici (OI) i dodati atribute koji su poželjni u bazi podataka, a ne nalaze se na OI (u slučaju da je korisnik policija). Potom prevesti DEV u relacijski model podataka.

Kako promijeniti DEV ako su dodatni zahtjevi korisnika sljedeći:

1. Potrebno je čuvati i podatke o adresi rođenja, mjestu rođenja, OIB i JMBG
2. Potrebno je čuvati i podatke sa starih OI i stare OI
3. Kako promijeniti DEV da možemo čuvati podatke teritorijalnog ustroja u kome su gradovi razvrstani po županijama, a policijske uprave pripadaju gradovima
4. Što bi bilo ako bi sustav bio namijenjen za više država koje nemaju županije i koje imaju potpuno različit teritorijalni ustroj (više različitih razina hijerarhije)
5. Kako bi se model promijenio ako bismo trebali pamtititi sve adrese bivših prebivališta na kojima je osoba bila prijavljena tijekom vremena, a taj podatak se nije mijenjao na osobnoj iskaznici, već je pri novoj adresi prebivanja izdana odgovarajuća potvrda s novom adresom

6. Što bi se dogodilo s modelom ako želimo pamtiti sve trenutne i bivše adrese (na primjer za vojne osobe i stan i vikendicu; a za političare, sve njihove kuće i stanove i adresu roditelja i važnih prijatelja) na kojima može osoba boraviti kako bi do iste po potrebi brzo došli
7. Što bi bilo kada bismo htjeli znati tko sve stanuje na odabranoj adresi? Što sve bi trebalo napraviti za takvu promjenu?
8. Kako bi podatke o putovnici povezali s ovim modelom?
9. Promislite: Je li bolje stare OI čuvati kao slab tip entiteta ili kao povratnu vezu?



Slika 9.2 DEV Osobna iskaznice (OI)

Relacijski model podataka Osobne iskaznice:

Država (**Šifra_države**, Naziv_države)

Županija (**Šifra_države**, **Šifra_županije**, Naziv_županije)

Grad (**Šifra_države**, **Šifra_grada**, Naziv_grada)

PU/PS (**Šifra_PU/PS**, Naziv_PU/PS, **Šifra_države**, **Šifra_županije**)

Osoba (**JBMG**, Ime_osobe, Prezime_osobe, Datum_rođenja, Adresa_rođenja, OIB, Spol(M/Ž), **Šifra_države**, **Šifra_grada_rod**)

OI (**Broj_OI**, Datum_vrijedi_do, Datum_vrijedi_od, Potpis, Status (V/N), **JMBG**, **Šifra_PU/PS**, **Šifra_države_prebivališta**, **Šifra_grada_prebivališta**, **Broj_OI_stare**, **Šifra_države_izdala**, **Šifra_države_državljanstva**)

9.2 Narudžbenica

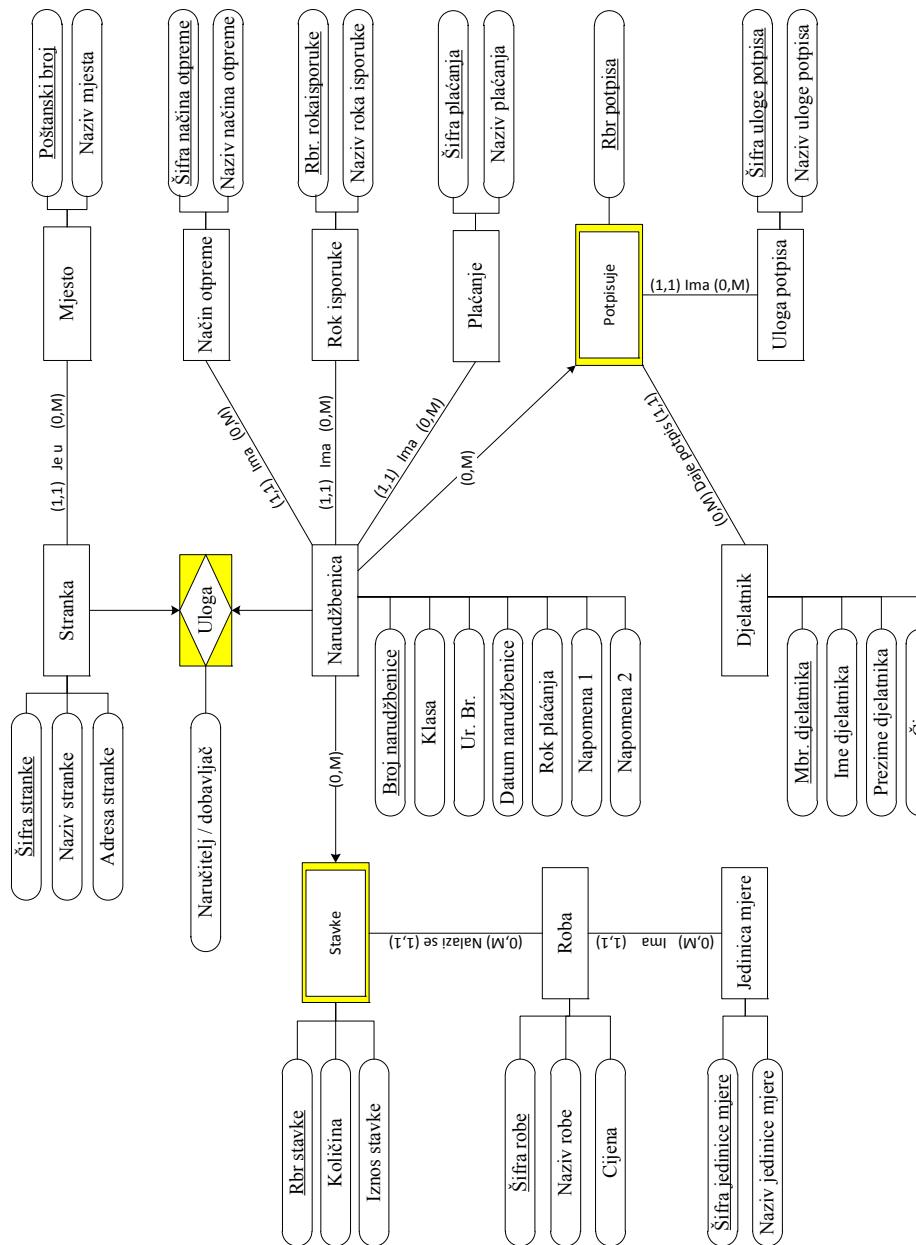
Narudžbenica je dokument (ili e-narudžba) u komercijalnome poslovanju kojim vojna ustanova naručuje kupnju robe i/ili usluga od dobavljača (vidi sl. 9.3.). Narudžba kupca može nastati na osnovi ponude dobavljača i pozvati se na istu s time da može promijeniti neke podatke, kao naručena količina ili stavke naručene robe. Najvažniji podaci su što, koliko, tko i od koga naručuje. Kod ponude dobavljača važno je i po kojoj konačnoj cijeni dobavljač nudi svoj proizvod/uslugu. Između dviju tvrtki (kupca i dobavljača) mogu nastati sljedeći dokumenti: upit kupca, ponuda dobavljača, narudžba kupca, potvrda narudžbe dobavljača (ova 4 dokumenta imaju pravnu snagu ugovora), ugovor za pojedinu isporuku, dugogodišnji ugovor o višestrukim isporukama, dostavnica dobavljača, primka kupca, račun dobavljača.

Na narudžbenici se pojavljuju ovi podaci: podaci o naručitelju (naziv naručitelja, mjesto, adresa), podaci o dobavljaču (naziv dobavljača, adresa dobavljača, mjesto), šifra klase dokumenta, urudžbeni broj pod kojim se vodi dokument, mjesto i datum sastavljanja narudžbenice, matični broj dobavljača, broj narudžbenice, opisna lokacija kamo isporučiti naručenu robu, rok isporuke, način otpreme, način plaćanja, redni broj stavke, naziv robe, jedinica mjere, količina, cijena, iznos, dvije napomene, ime i prezime i čin odgovorne osobe, ime i prezime zapovjednika.

Na slici 9.4 dan je model podataka narudžbenice. Na taj model dodajte podatke s ponude dobavljača koja je po obliku i strukturi slična narudžbenici, ali se može razlikovati u količini i vrsti robe. Prevedite DEV u relacijski model.

<p>REPUBLICA HRVATSKA MINISTARSTVO OBRANE ZAGREB VP 2112 Delnice</p> <p>Klasa: 413-02/02-01/00 Ur.br: 2112/2-S4-23-22 Rijeka, 01.05. 2005.g.</p>	<p style="text-align: center;">2. DOBAVLJAČU</p> <p>“auto fitness” Svetonedeljska cesta Kerestinec 10431 Sveta nedelja (tvrтka i adresa dobavlјаča)</p>																		
<p>Matični broj: 22422 NARUDŽBENICA br. 301</p>																			
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; padding: 5px;">NARUČENU ROBU-USLUGE ISPORUČITI NA NASLOV VP.2112 Delnice</td> <td style="width: 25%; padding: 5px;">ROK ISPORUKE ODMAH</td> <td style="width: 25%; padding: 5px;">NAČIN OTPreme poštom</td> </tr> </table>		NARUČENU ROBU-USLUGE ISPORUČITI NA NASLOV VP.2112 Delnice	ROK ISPORUKE ODMAH	NAČIN OTPreme poštom															
NARUČENU ROBU-USLUGE ISPORUČITI NA NASLOV VP.2112 Delnice	ROK ISPORUKE ODMAH	NAČIN OTPreme poštom																	
<p>plaćanja: <u>virman 15 dana</u></p>																			
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Red br.</th> <th style="width: 40%;">Naziv robe - Usluge - materijala</th> <th style="width: 10%;">Jed.mjere</th> <th style="width: 10%;">Količina</th> <th style="width: 10%;">Cijena</th> <th style="width: 10%;">Iznos</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>Klinasti remen 12,5x1200</td> <td>Kom.</td> <td>1</td> <td>150,00</td> <td>150,00</td> </tr> <tr> <td></td> <td>Zaključno sa rednim brojem jedan.</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>		Red br.	Naziv robe - Usluge - materijala	Jed.mjere	Količina	Cijena	Iznos	1.	Klinasti remen 12,5x1200	Kom.	1	150,00	150,00		Zaključno sa rednim brojem jedan.				
Red br.	Naziv robe - Usluge - materijala	Jed.mjere	Količina	Cijena	Iznos														
1.	Klinasti remen 12,5x1200	Kom.	1	150,00	150,00														
	Zaključno sa rednim brojem jedan.																		
<p>PRILIKOM DOSTAVE DOSTAVNICE I RAČUNA OBVEZATNO SE POZVATI NA GORNJI BROJ NARUDŽBENICE</p>																			
<p>ODGOVORNA OSOBA ntp. Ivan Marković</p> <hr/> <p>Ime i prezime, čin</p>	<p>ZAPOVJEDNIK pukovnik Marko Bukša</p> <hr/> <p>potpis</p>																		

Slika 9.3 Narudžbenica



Slika 9.4 DEV Narudžbenica

9.3 Račun-otpremnica

Zadatak: Nacrtajte DEV za dokument računa-otpremnice sa slike 9.5. proširujući model na sl. 9.4.

Na računu – otpremnici pojavljuju se sljedeći atributi: broj računa – otpremnice, podaci o prodavaču (naziv tvrtke prodavača, adresa, poštanski broj i mjesto, broj ţiro računa, broj telefona, broj faxa, porezni br.), podaci o kupcu (naziv kupca, adresa, poštanski broj i mjesto, ţifra kupca, sjedište kupca, porezni broj kupca), mjesto troška (ţifra i naziv), datum i mjesto izdavanja računa – otpremnice, nadnevak (datum) isporuke i nadnevak plaćanja, ţifre i vrste vezanih dokumenata (vezani dokument može biti narudžbenica), redni broj stavke plaćanja, ţifra artikla, naziv artikla, jedinična mjera, količina, cijena, % rabata, iznos bez PDV-a, PDV, iznos PDV-a, iznos s PDV-om, vrijednost robe, ukupan iznos bez PDV-a, ukupan iznos PDV-a, iznos računa, iznos računa slovima, napomena, ime i prezime osobe koja je izdala robu, ime i prezime osobe koja je primila robu.

BRODOKOMERC NOVA D.O.O.											
JELAČIĆEV TRG 4											
51000 RIJEKA											
Žiro račun:	2501182-1100009147										
Telefon:	+385(51)465-111										
Fax:	+385(51)211657										
Porezni br.:	1657528										
RAČUN – OTPREMNICA br. 41-21004-3											
Mjesto troška:	21004										
DISKONT 4	SIND.BRODOGRADILIŠTE KRALJEVICA/BON										
KRALJEVICA	12.01.2007.										
Nadnevak isporuke:	12.01.2007.										
Nadnevak plaćanja:	22.01.2007.										
OBALA KRALJA TOMISLAVA 6											
51262 KRALJEVICA											
Kupac:	001397										
Sjedište kupca:	1										
Porezni br.kupca	1224891										
VEZNI DOKUMENT MT: 21004 VD: 730 BR.DOK.: 13 INTERNI DOKUMENT: 13											
Rb.	Šifra	Naziv artikla		JMZ	Količina	Cijena	Rabat %	Iznos bez PDV-a	PDV	Iznos PDV-a	Iznos s PDV-om
1	002603	NAPOLITANKE CH. CREAM 500GR 10/1 KRAS		KOM	2,00	17,09	0,00%	34,18	22,00%	7,52	41,70
2	004207	KAVA MLJEV.JUB. 100GR 60/1 FRANCK		KOM	6,00	7,34	0,00%	44,04	22,00%	9,69	53,73
3	034594	BANANA		KG	0,82	5,33	0,00%	4,37	22,00%	0,96	5,33
Vrijednost robe: 82,59											
Iznos bez PDV-a: 82,59											
Iznos PDV-a: 18,17											
Iznos računa : 100,76											
Slovima: sto Kn i sedamdesetšest Ip											
Račun je izrađen računalom i punovažan je bez potpisa i pečata											
Izdao:		Primio:									

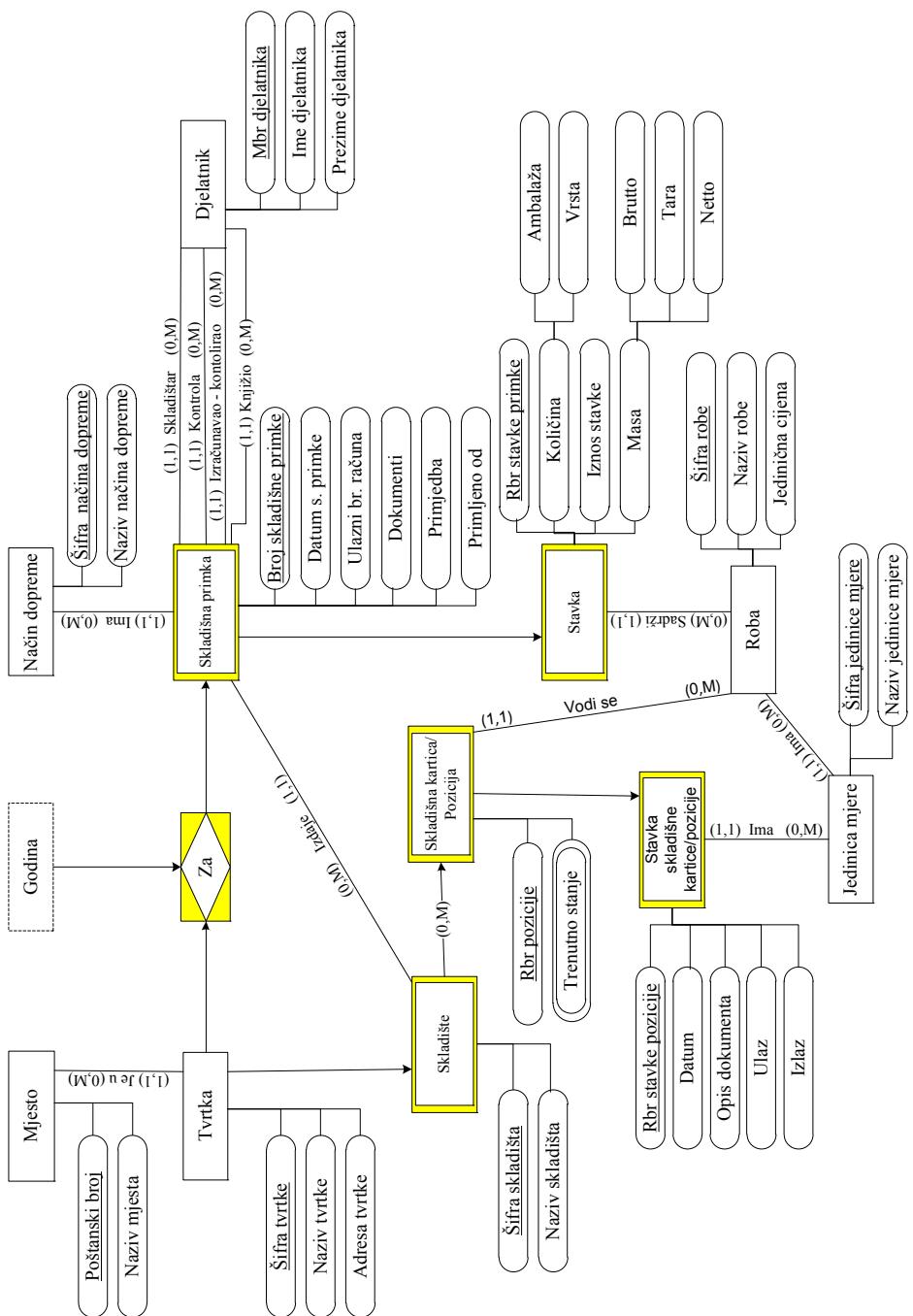
Slika 9.5 Račun - otpremnica

9.4 Skladišna primka

Skladišna primka je dokument kojim kupac zaprima robu od dostavljača. Dostavljač (tvrtka koja za dobavljača ili proizvođača prevozi robu kupcu) uz robu donosi dokument, dostavnicu na kojoj piše što isporučuje kupcu i količinski koliko je to jedinica mjere. Kupac može i ne mora potpisati taj dokument, ako se ne slaže s vrstom dostavljene robe ili količinom ili kvalitetom ili je došlo do loma proizvoda i sl. Kupac stvara novi dokument Primka, kojim popisuje svu robu koju je zaprimio od dostavljača. Dostavljač može biti posrednik između dobavljača i kupca koji se bavi transportom robe i ne mora imati sa sobom narudžbu kupca. Zaprimljena roba i naručena roba ne moraju se potpuno podudarati. Na slici 9.6. dan je prikaz dokumenta, a na slici 9.7. nacrtan je odgovarajući DEV.

Na skladišnoj primci pojavljuju se sljedeći atributi: Broj skladišne primke, podaci o poduzeću, datum primke, ulazni broj računa, naziv skladišta, primljeno od, način do-preme, broj i vrsta dokumenta, redni broj stavke, količina ambalaže, vrsta ambalaže, skladišni broj materijala - robe, naziv materijala – robe, jedinica mjere, bruto masa, tara masa, netto masa, cijena, iznos, primjedba, ime i prezime osoba i to: skladistara, kontrolora kvalitete, kontrolora finansijske finansijske vrijednosti, osobe koja knjiži.

Slika 9.6 Skladišna primka



Slika 9.7 DEV Skladišna primka i skladišna kartica

9.5 Skladišna kartica

Skladišna kartica je „tvrdi“ papirnati dokument koji se nalazi i vodi u skladištu (ukoliko nema računalom podržanog IS-a) i predstavlja papirnatu bazu podataka trenutnoga stanja i prometa robe na skladištu. Za svaku vrstu robe nastaje jedna skladišna kartica. Tvrta može imati više skladišta. Jedno skladište robe je podijeljeno u više pozicija. Na jednoj poziciji nalazi se jedna roba. Na jednoj kartici vode se podaci o stanju jedne vrste robe na jednoj poziciji u jednom skladištu. Na kartici je vidljivo trenutno stanje robe u skladištu, početno stanje, ali i sav promet, dakle i ulaz i izlaz robe sa skladišta. Na slici 9.8 prikazan je dokument skladišne kartice. Na sl. 9.7 dodani su tipovi entiteta skladišne kartice zajedno s primkom.

Na skladišnoj kartici nalaze se ovi atributi: skladišni broj, pozicija u skladištu, broj strane kartice ako je promet robe velik i potrebno je nastaviti pisati promet na drugoj kartici, naziv materijala, nomenklatura (šifra materijala), datum prometa, broj dokumenta koji uzrokuje promet, ulazna količina, izlazna količina i trenutno stanje.

Slika 9.8 Skladišna kartica

9.6 Receipt

Recept je dokument kojim se nalaže izdavanje odgovarajućega lijeka bolesniku. Na receptu je označen način i količina dnevnoga konzumiranja lijeka. U ovisnosti o pravima pacijenta, razini socijalne pomoći, vrstama osiguranja i drugom, mogući su i drugi podaci.

Na receptu se nalaze atributi: broj područnog ureda, broj osigurane osobe, broj obveze, ime i prezime, godina rođenja i MB osigurane osobe, grad/naselje, ulica i broj, šifra zdravstvene ustanove, šifra ugovornoga doktora, šifra doktora specijaliste, primarna / sekundarna prevencija, kategorija osiguranja, spol, sudjelovanje, šifra osl. od sudj., evidencijski broj i godina priznate OR/PB, država osiguranja, broj bolesničkoga lista INO, broj putovnice, europska karta ZO, šifra dijagnoze prema MKB, šifra lijeka, cijena originalnoga pakiranja, količina, cijena usluge, na teret osnovnoga zdravstvenog osiguranja, šifra ljekarne, datum izdavanja lijeka, šifra ljevkarnika, mjesto izdavanja recepta, datum, opis recepta (napomena).

HRVATSKI ZAVOD ZA ZDRAVSTVENO OSIGURANJE

Područni ured
072

Broj osigurane osobe
07203030515

* L 0 7 7 0 6 3 9 9 *

07200007249

Broj obavare

RENE GAŠPAR

Ime i prezime

Godine rođenja i

16.12.1963

MB osigurane osobe

1612963360040

RIJEKA

JOŽE VLAHOVIĆ 1

Grad/mjesto

Ulica i broj

Kat. osig.

Spol

Stanovanje*

Slišta odj.

od kuće

Br. lekarnice dosp. zdr. osig. 072/0

Br. lekarnice odj. kuće

Evidencijski broj i godina primjete

(PB)

Distr. osig.

Broj božja, lata INO,

broj potvrde,

europaksa karta ZD

ČL. 58.1113,
Zak. o zdr. osig.*

Broj evidencije
prijevoz ozljedobosilj

(PB)

Slišta dž. prema MBK

PB (OR PN TJO)

C 3 8 *

Rp.

Necesse est!

Slišta lijeka

Cij. orig. pakir. II. s L. II.

Durogesic TTS

Količina

Cijena usluge

flasteri 5x2,5 mg (25 µg/h)

Na teret osn. zdr. osig.

D. scat. orig. N° II (duo)

Na teret dosp. zdr. osig. uklj. usluga

D. S. svaka 72 sata naličjepiti

Slišta lijekarne

Datum izdavanja lijeka

Slišta lijekarska

RIJEKA

M.P.

— 200 — g.

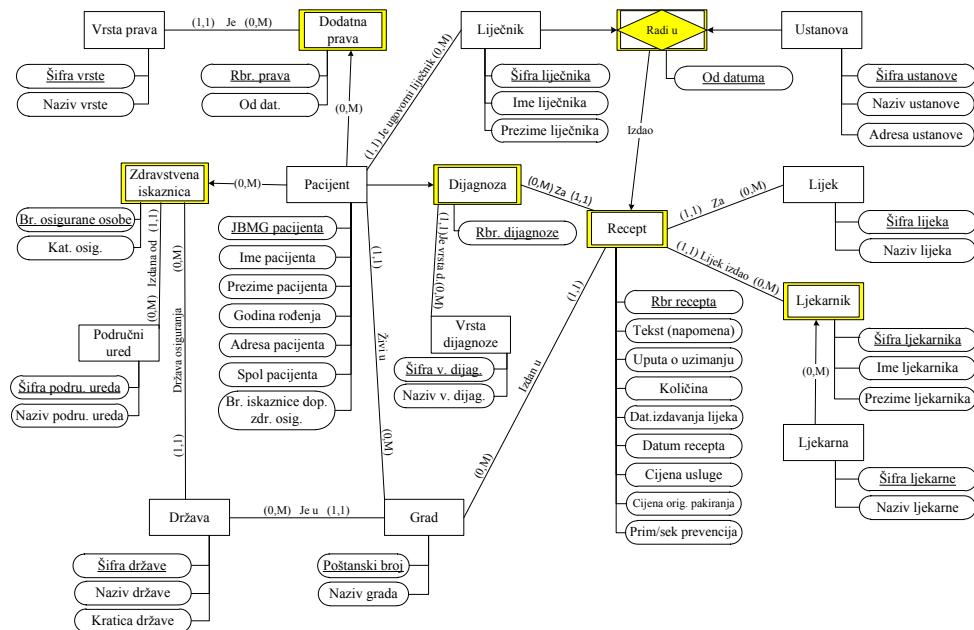
Pečat lijekarske i potpis ljekarska

HZZO-Direktor, o. Zagreb
Receptip, V11, bl. 12. 94

Prelišak zabranjen
Tiskarska zaštitna

* - U odgovarajuću kućištu staviti čvorak - X -

Slika 9.9 Recept



Slika 9.10 DEV Recept

9.7 Prijava štete od požara, eksplozije i drugih opasnosti

Osiguravajuća društava zahtijevaju od osiguranika da u slučaju nastupanja osiguratelnog slučaja (kvara, poplave, požara, krađe, loma, nezgode, smrti, ...) popune dokumentaciju o nastaloj šteti. U tome dokumentu navode se podaci o polici osiguranja te podaci o štetnom događaju. Šteta se može dogoditi na raznim objektima osiguranja. Šteta može biti izazvana raznim uzrocima, odnosno postoji različite vrste štetnih opasnosti kao: požar, eksplozija, poplava, grom, oluja, tuča i drugih vrsta opasnosti. Na slici 9.11 prikazan je dokument za prijavu jedne vrste šteta.

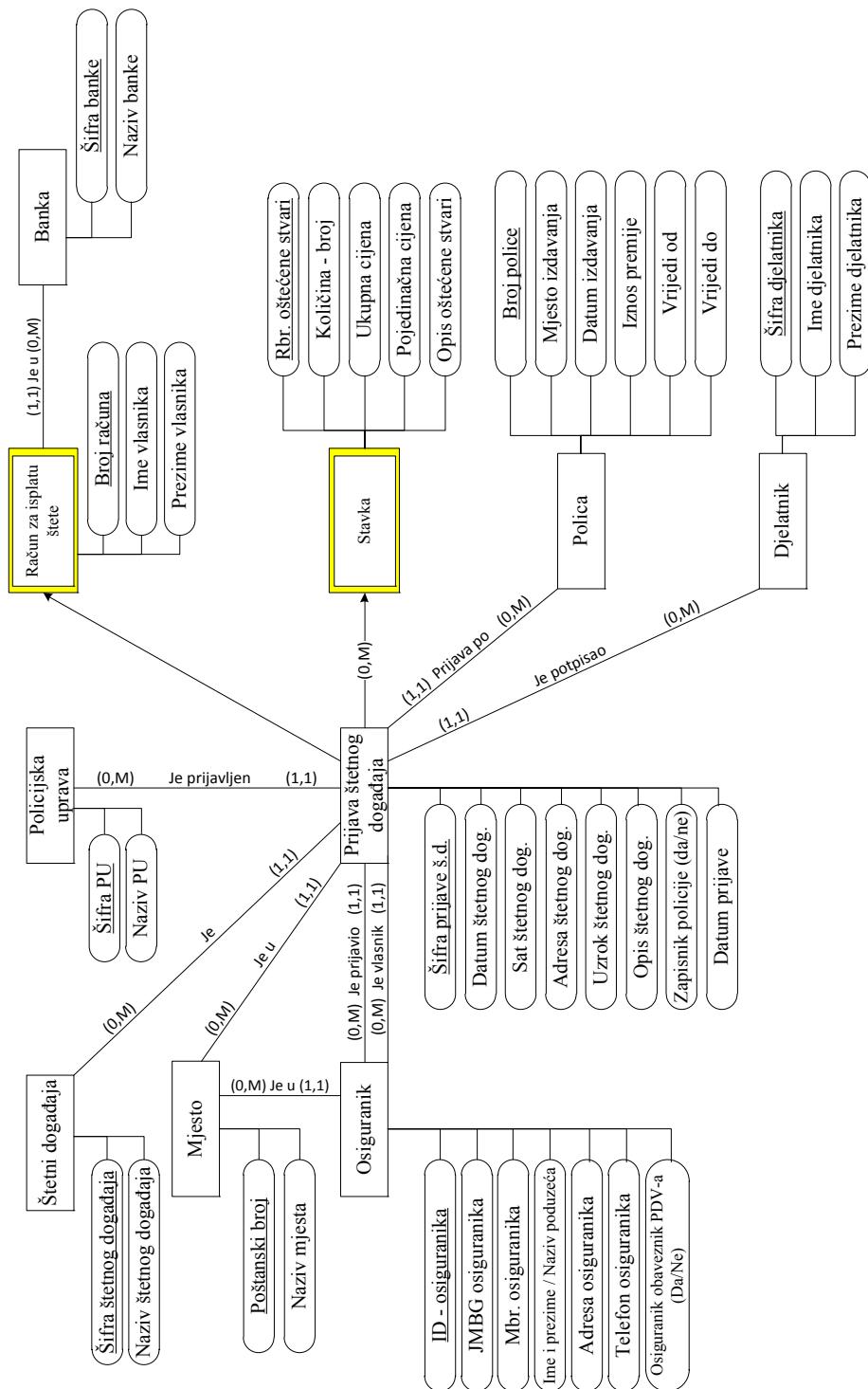
Na tome dokumentu nalaze se ovi atributi: ime i prezime/naziv poduzeća osiguranika, JMBG/MB, adresa, broj telefona, je li osiguranik obveznik poreza na dodanu vrijednost, broj police, datum i sat štetnoga događaja, mjesto štetnoga događaja, opis štetnoga događaja, zapisnik policije, naziv policijske uprave, vlasnik oštećene stvari, visina štete, naziv oštećenih stvari, marka, tvornički broj oštećene stvari, godina proizvodnje/nabavke, broj oštećene stvari, pojedinačna nabavna cijena, ukupna nabavna cijena, vlasnik računa za isplatu štete, naziv banke, žiro-račun banke, broj računa na koji se isplaćuju novci, datum prijave štete, ime i prezime osobe koja je podnijela prijavu, ime i prezime djelatnika koji je prijavu zaprimio i potpisao.

Ime i prezime osobe za tvrtke je atribut koji ima nul vrijednost za pravne osobe. Naziv osiguranika je izvedeni atribut u kome se nalaze podaci i ime i prezime za fizičku osobu ili naziv tvrtke za pravnu osobu.

Za dokument nacrtajte DEV i usporedite ga s predloženim rješenjem. DEV prevedite u relacijsku shemu.

 SUNCE OSIGURANJE BRAĆE DOMANIĆ 6, 10 000 ZAGREB tel.: 01/3844 222, fax: 01/3844 155	PRIJAVA ŠTETE OD POŽARA, EKSPLOZIJE, POPLAVE, GROMA, OLUJE, TUČE I DRUGIH OPASNOSTI																															
OSIGURANIK <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">IME I PREZIME / NAZIV PODUZEĆA</td> <td style="padding: 5px;"><u>JELENA HORVAT</u></td> </tr> <tr> <td style="padding: 5px;">JMBG / MB</td> <td style="padding: 5px;"><u>0509965308974</u></td> </tr> <tr> <td style="padding: 5px;">ADRESA</td> <td style="padding: 5px;"><u>CVIJETNA 5B, 51000 RIJEKA</u></td> </tr> <tr> <td style="padding: 5px;">TELEFON</td> <td style="padding: 5px;"><u>051/323-821</u></td> </tr> <tr> <td colspan="2" style="padding: 5px;">DA LI JE OSIGURANIK OBVEZNIK POREZA NA DODANU VRIJEDNOST (ako da, navesti broj poreznog obveznika)</td> </tr> <tr> <td colspan="2" style="padding: 5px; text-align: center;"><u>DA</u></td> </tr> <tr> <td colspan="2" style="padding: 5px;">BROJ POLICE</td> </tr> <tr> <td colspan="2" style="padding: 5px; text-align: center;"><u>13001255</u></td> </tr> </table>		IME I PREZIME / NAZIV PODUZEĆA	<u>JELENA HORVAT</u>	JMBG / MB	<u>0509965308974</u>	ADRESA	<u>CVIJETNA 5B, 51000 RIJEKA</u>	TELEFON	<u>051/323-821</u>	DA LI JE OSIGURANIK OBVEZNIK POREZA NA DODANU VRIJEDNOST (ako da, navesti broj poreznog obveznika)		<u>DA</u>		BROJ POLICE		<u>13001255</u>																
IME I PREZIME / NAZIV PODUZEĆA	<u>JELENA HORVAT</u>																															
JMBG / MB	<u>0509965308974</u>																															
ADRESA	<u>CVIJETNA 5B, 51000 RIJEKA</u>																															
TELEFON	<u>051/323-821</u>																															
DA LI JE OSIGURANIK OBVEZNIK POREZA NA DODANU VRIJEDNOST (ako da, navesti broj poreznog obveznika)																																
<u>DA</u>																																
BROJ POLICE																																
<u>13001255</u>																																
PODACI O ŠTETNOM DOGAĐAJU <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">DATUM I SAT</td> <td style="padding: 5px;"><u>10.05.2010. 17:00</u></td> </tr> <tr> <td style="padding: 5px;">MJESTO</td> <td style="padding: 5px;"><u>RIJEKA, CVIJETNA 5B</u></td> </tr> <tr> <td style="padding: 5px;">UZROK</td> <td style="padding: 5px;"><u>ŠTETA UZROKOVANA OLUJOM</u></td> </tr> <tr> <td colspan="2" style="padding: 5px;">OPIS</td> </tr> <tr> <td colspan="2" style="padding: 5px; text-align: left;"><u>OŠTEĆENJE TRI PROZORA PRIVATNE KUĆE</u></td> </tr> <tr> <td colspan="2" style="padding: 5px;">ZAPISNIK POLICIJE (da, ne), NAVESTI NAZIV PU</td> </tr> <tr> <td colspan="2" style="padding: 5px; text-align: center;"><u>DA, POSTAJA RIJEKA</u></td> </tr> </table>		DATUM I SAT	<u>10.05.2010. 17:00</u>	MJESTO	<u>RIJEKA, CVIJETNA 5B</u>	UZROK	<u>ŠTETA UZROKOVANA OLUJOM</u>	OPIS		<u>OŠTEĆENJE TRI PROZORA PRIVATNE KUĆE</u>		ZAPISNIK POLICIJE (da, ne), NAVESTI NAZIV PU		<u>DA, POSTAJA RIJEKA</u>																		
DATUM I SAT	<u>10.05.2010. 17:00</u>																															
MJESTO	<u>RIJEKA, CVIJETNA 5B</u>																															
UZROK	<u>ŠTETA UZROKOVANA OLUJOM</u>																															
OPIS																																
<u>OŠTEĆENJE TRI PROZORA PRIVATNE KUĆE</u>																																
ZAPISNIK POLICIJE (da, ne), NAVESTI NAZIV PU																																
<u>DA, POSTAJA RIJEKA</u>																																
OŠTEĆENA STVAR <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">VLASNIK OŠTEĆENE STVARI</td> <td style="padding: 5px;"><u>JELENA HORVAT</u></td> </tr> <tr> <td style="padding: 5px;">VISINA ŠTETE</td> <td style="padding: 5px;"><u>1500,00 kn</u></td> </tr> <tr> <td colspan="2" style="padding: 5px; text-align: center;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">OŠTEĆENA STVAR (naziv, marka, tvornički broj i sl.)</th> <th style="text-align: left;">God. proizvodnje</th> <th style="text-align: left;">Broj</th> <th style="text-align: left;">Nabavna cijena pojedinačna</th> <th style="text-align: left;">Nabavna cijena ukupna</th> </tr> </thead> <tbody> <tr> <td style="text-align: left;"><u>PROZOR</u></td> <td style="text-align: left;"><u>2000.</u></td> <td style="text-align: left;"><u>3</u></td> <td style="text-align: left;"><u>500,00 kn</u></td> <td style="text-align: left;"><u>1500,00 kn</u></td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table> </td> </tr> </table>		VLASNIK OŠTEĆENE STVARI	<u>JELENA HORVAT</u>	VISINA ŠTETE	<u>1500,00 kn</u>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">OŠTEĆENA STVAR (naziv, marka, tvornički broj i sl.)</th> <th style="text-align: left;">God. proizvodnje</th> <th style="text-align: left;">Broj</th> <th style="text-align: left;">Nabavna cijena pojedinačna</th> <th style="text-align: left;">Nabavna cijena ukupna</th> </tr> </thead> <tbody> <tr> <td style="text-align: left;"><u>PROZOR</u></td> <td style="text-align: left;"><u>2000.</u></td> <td style="text-align: left;"><u>3</u></td> <td style="text-align: left;"><u>500,00 kn</u></td> <td style="text-align: left;"><u>1500,00 kn</u></td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>		OŠTEĆENA STVAR (naziv, marka, tvornički broj i sl.)	God. proizvodnje	Broj	Nabavna cijena pojedinačna	Nabavna cijena ukupna	<u>PROZOR</u>	<u>2000.</u>	<u>3</u>	<u>500,00 kn</u>	<u>1500,00 kn</u>															
VLASNIK OŠTEĆENE STVARI	<u>JELENA HORVAT</u>																															
VISINA ŠTETE	<u>1500,00 kn</u>																															
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">OŠTEĆENA STVAR (naziv, marka, tvornički broj i sl.)</th> <th style="text-align: left;">God. proizvodnje</th> <th style="text-align: left;">Broj</th> <th style="text-align: left;">Nabavna cijena pojedinačna</th> <th style="text-align: left;">Nabavna cijena ukupna</th> </tr> </thead> <tbody> <tr> <td style="text-align: left;"><u>PROZOR</u></td> <td style="text-align: left;"><u>2000.</u></td> <td style="text-align: left;"><u>3</u></td> <td style="text-align: left;"><u>500,00 kn</u></td> <td style="text-align: left;"><u>1500,00 kn</u></td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>		OŠTEĆENA STVAR (naziv, marka, tvornički broj i sl.)	God. proizvodnje	Broj	Nabavna cijena pojedinačna	Nabavna cijena ukupna	<u>PROZOR</u>	<u>2000.</u>	<u>3</u>	<u>500,00 kn</u>	<u>1500,00 kn</u>																					
OŠTEĆENA STVAR (naziv, marka, tvornički broj i sl.)	God. proizvodnje	Broj	Nabavna cijena pojedinačna	Nabavna cijena ukupna																												
<u>PROZOR</u>	<u>2000.</u>	<u>3</u>	<u>500,00 kn</u>	<u>1500,00 kn</u>																												
Ako prostor nije dovoljan, sastaviti poseban spisak prema navedenim podacima.																																
RAČUN ZA ISPLATU ŠTETE <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">VLASNIK RAČUNA</td> <td style="padding: 5px;"><u>JELENA HORVAT</u></td> </tr> <tr> <td style="padding: 5px;">BANKA (naziv i ţiro - račun banke)</td> <td style="padding: 5px;"><u>PRIVREDNA BANKA ZAGREB</u></td> </tr> <tr> <td style="padding: 5px;">BROJ RAČUNA</td> <td style="padding: 5px;"><u>1524332541</u></td> </tr> </table>		VLASNIK RAČUNA	<u>JELENA HORVAT</u>	BANKA (naziv i ţiro - račun banke)	<u>PRIVREDNA BANKA ZAGREB</u>	BROJ RAČUNA	<u>1524332541</u>																									
VLASNIK RAČUNA	<u>JELENA HORVAT</u>																															
BANKA (naziv i ţiro - račun banke)	<u>PRIVREDNA BANKA ZAGREB</u>																															
BROJ RAČUNA	<u>1524332541</u>																															
Za točnost naprijed navedenih podataka odgovaram materijalno i krivično.																																
Potpis podnositelja prijave <u>Jeleva Horvat</u>	Datum prijave <u>10.05.2010.</u>																															
Za Sunce osiguranje d.d. <u>PETRA PERIĆ</u>																																
Kod osiguranja pravnih osoba obvezno priložiti zapisnik komisije o šteti. Uništene ili oštećene stvari i dijelove obvezno sačuvati do kraja likvidacije štete. 30-80101-01																																

Slika 9.11 Prijava štete od požara, eksplozije, poplave, groma, oluje, tuče i drugih opasnosti



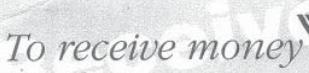
Slika 9.12 DEV Prijava štete od požara, eksplozije, poplave, gromu, oluje, tuče i drugih opasnosti

9.8 Primitak novca

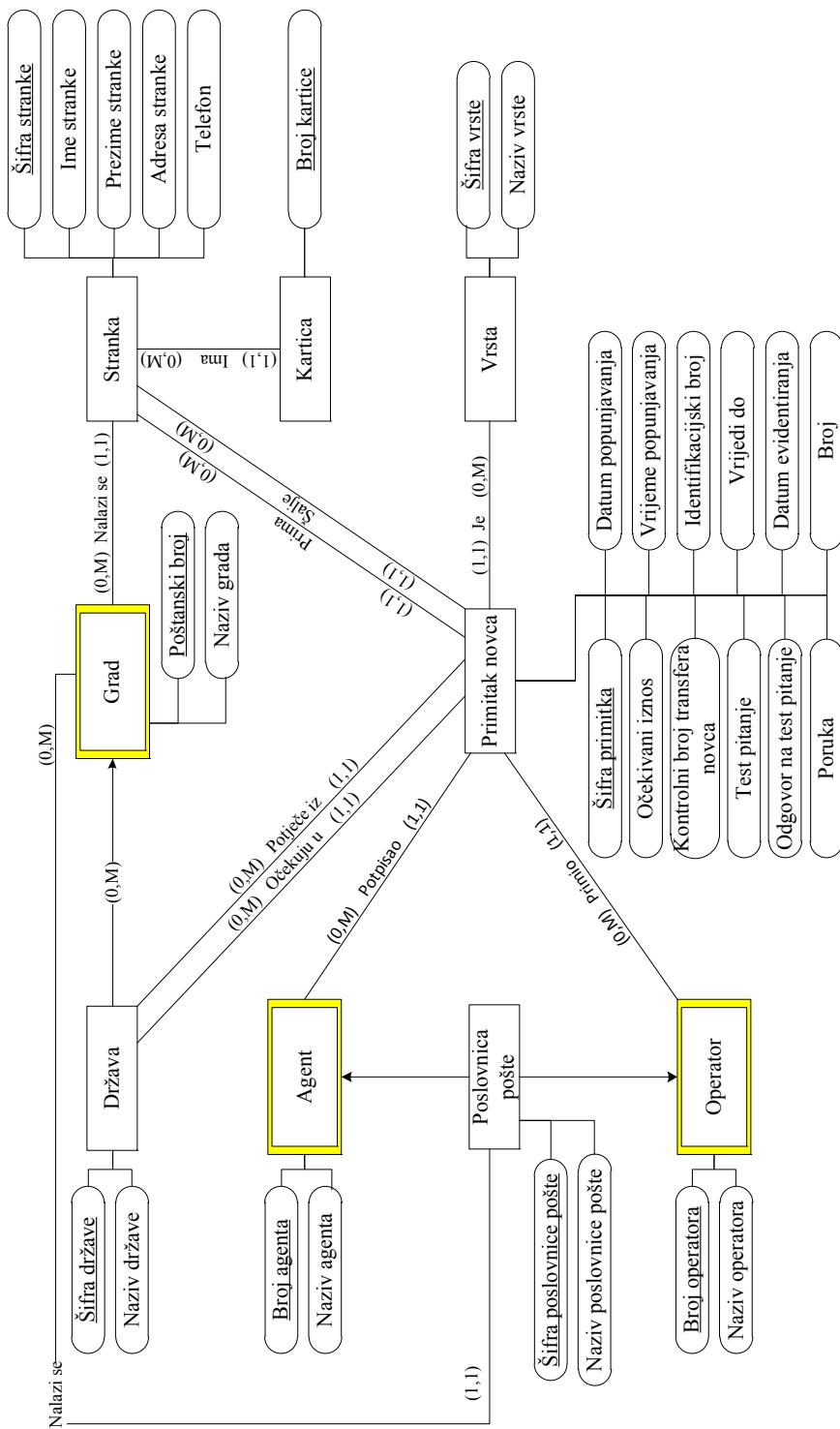
Tvrтka Western Union omogućuje transfer novca između osoba iz raznih država. Osobe se međusobno dogovore o iznosu, test pitanju i kontrolnome broju (koji izdaje Western Union pošiljatelja). Na slici je prikazan dokument koji se popunjava prilikom preuzimanja novca. Primatelj ispunjava samo lijevu stranu obrasca i predaje ga operateru Western Uniona. Operater provjerava podatke u bazi podataka i ispunjava desnú stranu, prepisujući podatke iz baze i uspoređujući ih s iskazom stranke te, ako se podaci slažu, isplaćuje novac primatelju.

Na dokumentu „Primitak novca“ nalaze se ovi atributi: broj kartice (za stalne primatelje), podaci o primatelju (ime i prezime, adresa, poštanski broj, mjesto, država, broj telefona), podaci o pošiljatelju (ime i prezime, adresa, poštanski broj, mjesto, država, broj telefona), očekivani iznos, mjesto država odakle je novac poslan, kontrolni broj transfera novca, test pitanje, odgovor na test pitanje, poruka primatelju, agencija Western Uniona, broj operatora, datum, vrijeme, broj dokumenta, datum evidentiranja, mjesto/država porijekla transfera, očekivano mjesto/država isplate, ime i prezime agenta.

Prevedite DEV u relacijsku shemu.

 To receive money <i>Primitak novca</i>		HP - Hrvatska pošta d.d.  WESTERN UNION
Ispunjava primatelj <p>Complete the form and present valid I.D. / Ispunite obrazac i predočite valjni identifikacijski dokument.</p> <p>For Western Union CardSM holders, please fill in your card number. / Za vlasnike Western Union Card, molimo unesite broj svoje kartice.</p> <p>Card No. Broj kartice 351682941</p> <p>Receiver / Primatelj</p> <p>First name(s) Ime MARKO</p> <p>Last name Prezime MARCOVIC</p> <p>Address Adresa FILIP ČKA 52 Street / Ulica City / Grad UVAGA HRVATSKA Province/Country / Država 311526 Postal Code / Poštanski broj</p> <p>Telephone no. Telefonski broj 10821123321</p>		Ispunjava pošiljatelj <p>DO NOT WRITE BELOW / NE ISPUNJAVAĆITE OVE RUBRIKE</p> <p>Agency POST Agencija VRED Operator / Broj number / operatora 34</p> <p>Date 310106 Time Vrijeme 12:34</p> <p>Vrijeme i datum pristizanja</p>
<p>Money Transfer Control Number Kontrolni broj</p> <p>3541267891</p> <p>Check / Kontrola number / broja 2000,00 kn</p> <p>Date of payout Datum isplate 310106</p> <p>Originating / Mjesto, država city/country / ponjeklja transfera VELIKA BRITANIJA</p> <p>Expected payout / Očekivano mjesto city/country / država isplate HRVATSKA</p>		<p>Amount / Očekivani expected / iznos 2000,00 kn</p> <p>Test Question / Test pitanje DATUM ROĐENJA 17.2.1975.</p> <p>Answer / Odgovor</p> <p>Operator's / Potpis signature / operatora Leporak</p>
<p>CERTAIN TERMS AND CONDITIONS GOVERNING THE MONEY TRANSFER SERVICE YOU HAVE SELECTED ARE SET FORTH ON THE BACK OF THIS FORM. BY SIGNING THIS FORM, YOU ARE AGREEING TO THOSE TERMS AND CONDITIONS. / NEKI TERMINI I UVJETI KOJI REGULIRAJU USLUGU TRANSFERA NOVCA KOJI STE ODABRALI NAVEDENI SU NA POLEDINI OVOG OBRASCA. IN ADDITION TO THE TRANSFER FEE, WESTERN UNION AND ITS AGENTS ALSO MAKE MONEY FROM THE EXCHANGE OF CURRENCIES. / POTRIVANjem OBRASCA PRIMHAĆATE NAVEDENE TERMINE I UVJETE. *PLEASE SEE IMPORTANT INFORMATION REGARDING CURRENCY EXCHANGE SET FORTH ON THE BACK OF THIS FORM. / MOLIMO POGLEDAJTE VAŽNE OBAVJESTI NA POLEDINI OVOG OBRASCA KOJE SE ODNOSE NA TEČAJ RAZMJENE VALUTA.</p>		
Customer's signature / Potpis primatelja <hr/>		Date / Datum primitka <hr/>
AGENT'S COPY / AGENTOV PRIMJERAK <hr/>		www.westernunion.com <small>WU Template 2 Eng Copyright ©2002-2003 Western Union Holdings Inc. All Rights Reserved</small>

Slika 9.13 Primitak novca



Slika 9.14 DEV Primitak novca

9.9 Račun za prikupljanje RTV – pristojbe

Javna radiotelevizija prema zakonu uglavnom predstavlja servis društva u proizvodnji emisija i emitiranju TV programa od širega društvenog interesa kao: kultura, obrazovanja, manjinska prava, znanost, šport, i druge teme koje uglavnom ne moraju imati komercijalni interes. Taj servis se naplaćuje (u nekim državama) računom (prikazanim na slici 9.15.) svih fizičkih imateljima radiotelevizijskih uređaja (misli se na građane, ali ne i na pravne osobe). Model DEV računa prikazan je na slici 9.16. Mjesečno se tiska preko milijun računa fizičkim i pravnim osobama.

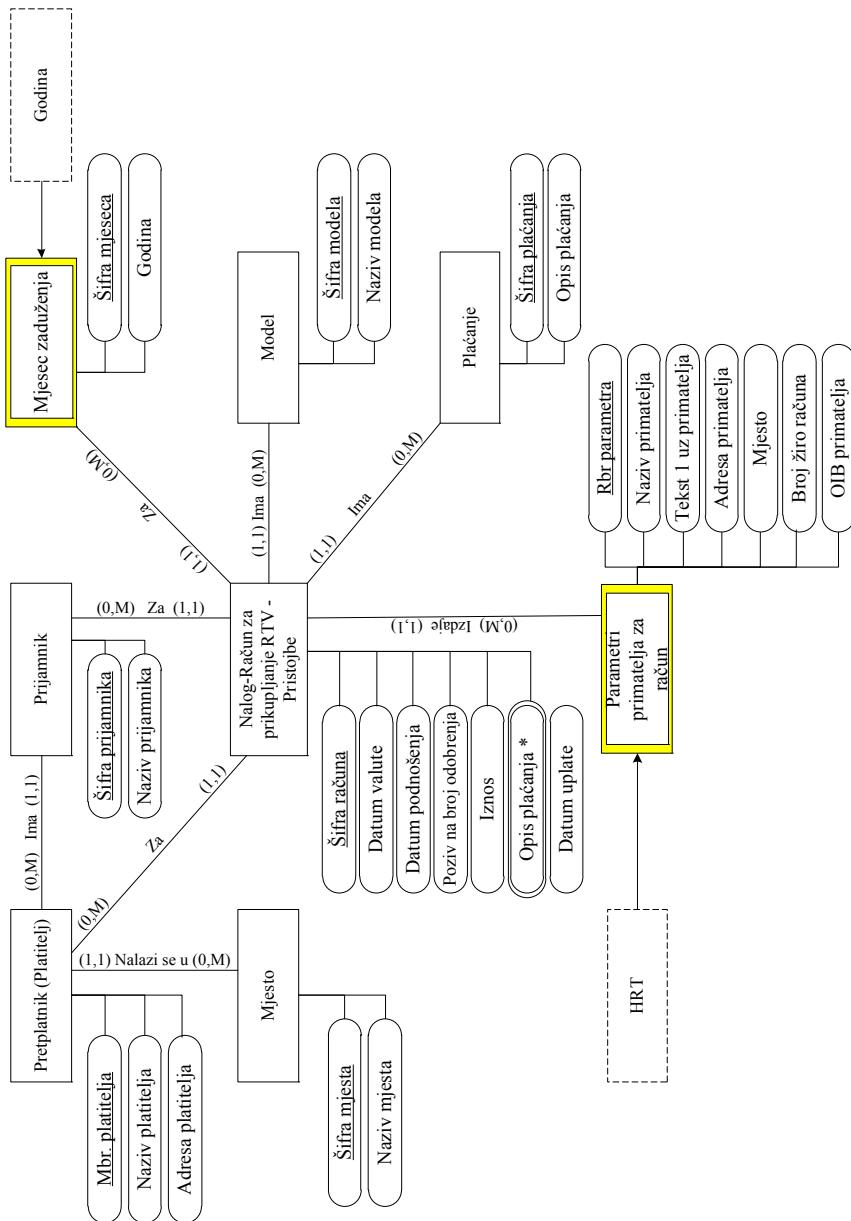
Na tome se dokumentu nalaze ovi atributi: naziv (ime i prezime) i adresa platitelja (preplatnika), naziv i adresa primatelja, šifra modela, broj računa primatelja, poziv na broj odobrenja (koji se sastoji od vrste računa (opća uplatnica), šifre preplatnika, godina i mjesec te kontrolni broj), statističko obilježje, šifra opisa plaćanja, opis plaćanja, datum valute/uplate/isplate, datum podnošenja, iznos.

Pretpostavimo da postoji jedan i samo jedan primatelj. To znači da ako je HRT izdala račun, onda ona ima svoje podatke na svim računima iste kao: naziv, adresa, broj ţiro računa, banku na kojoj se nalazi njezin račun. Ti se podaci mogu mijenjati, ali za jedan mjesec oni su isti na svim računima. Te podatke modelirajmo kao parametre koji će biti sačuvani u bazi kako bismo znali kakvi su bili kada je generiran pojedini račun.

Što bi bilo ako bi u modelu podataka trebalo udovoljiti sljedećim zahtjevima:

- Uključiti i račune pravnim osobama koji na računu mogu imati više stavki. Svaka stavka može biti za više prijamnika s jedne lokacije (recimo jednoga hotela) pravne osobe. Stavka može biti za naplatu tekućega mjeseca, ali i za naplatu ranijih mjeseci ako platitelj nije platio ranije primljeni račun.
- Ako se mjesечно vrati i ne plati 10% računa, onda je poželjno povratne račune skenirati u bazu podataka, a barcod na računu koji predstavlja šifru računa služi za povezivanje tiskanih i neplaćenih računa. Što treba napraviti na DEV-u da se čuvaju neplaćeni računi? Kako se DEV širi ako za svaki neplaćeni račun tiskamo opomenu, a treća opomena je zadnja pred tužbu?

Slika 9.15 Račun za prikupljanje RTV – pristojbe

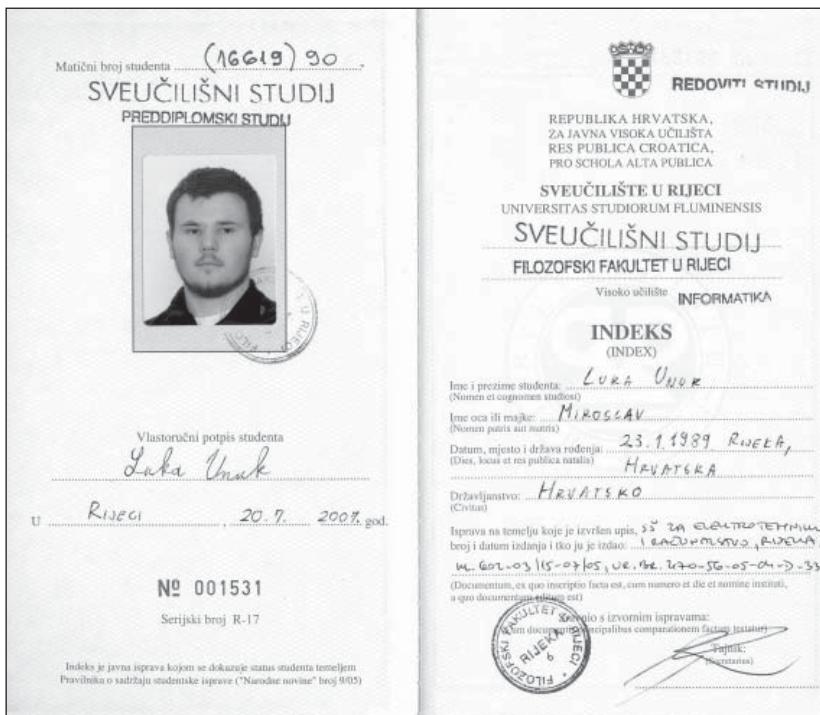


* Opis plaćanja = „PRB.“ & Mbr. platitelja & „PRISTOJBA ZA“ & Mjesec zaduženja

Slika 9.16 DEV Račun za prikupljanje RTV - pristojbe

9.10 Indeks (2 strane)

Indeks je „knjižica“ koju ima svaki student upisan na neko visoko učilište. U nje- ga se upisuju predmeti, potvrđuje pohađanje predavanja i uspjeh u polaganju ispita. Na prvoj stranici nalaze se osnovni podaci o studentu (ime i prezime studenta, datum i mjesto rođenja) te podaci o visokom učilištu. Na sljedećim stranicama nalazi se popis upisanih kolegija u pojedinoce semestru, evidencija izvršavanja obaveza, uspjeh na ispitu i sl.



Slika 9.17 Indeks

Na indeksu se nalaze sljedeći atributi: matični broj studenta, država, naziv sveučilišta, naziv visokog učilišta, naziv smjera, oznaka redoviti/izvanredni studij, ime i prezime studenta, ime oca ili majke, datum rođenja, mjesto rođenja, država rođenja, državljanstvo, naziv isprave na temelju koje je izvršen upis, broj izdane isprave, datum izdanja isprave, mjesto izdanja isprave, datum ispunjavanja indeksa, mjesto ispunjavanja indeksa, serijski broj indeksa, ime i prezime tajnika visokog učilišta koji odobrava upis studija.

Za svaki semestar koristi se više unutrašnjih strana. Unutar indeksa nalaze se sljedeći podaci: redni broj semestra, godina studija (nije predviđeno obrascem, ali se upisuje), akademska godina studija, ime i prezime nastavnika, naziv predmeta, ukupno sati

predavanja, vježbi, seminara, izvršene obveze, uspjeh na ispitu (ocjena riječima, brojčana), ocjena po bolonji (slovom, postotak), datum polaganja ispita, broj ETCS bodova, datum upisa u semestar, ime i prezime dekana koji je ovjerio stranicu indeksa.

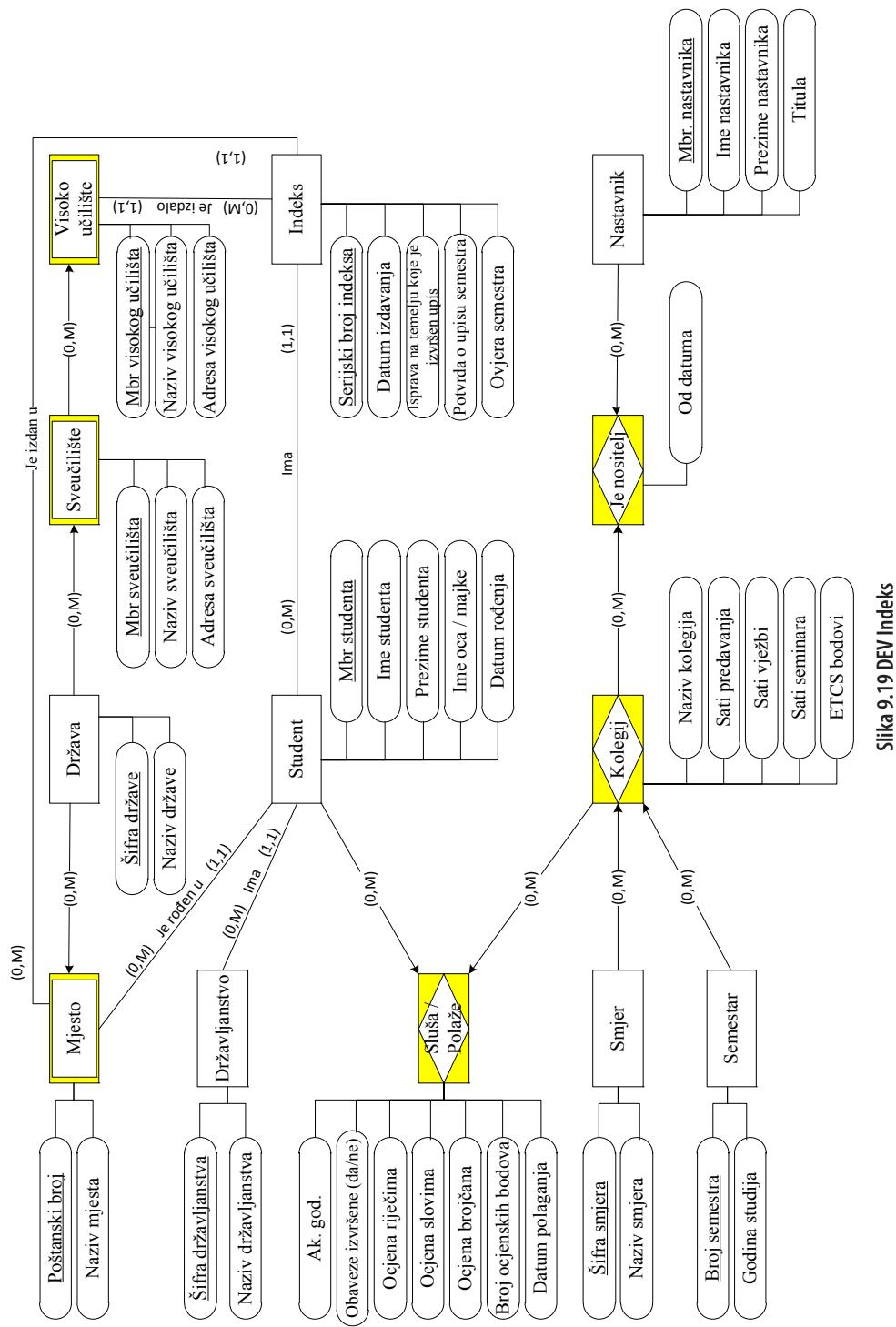
3. godina							
Broj: R-17	Š.:	zimski (izvješće)	ljetni (zastitnik)	semestar (semestar)			
Num: № 001531							
Ime i prezime nastavnika (Nomen et cognomen magistri)	Naziv predmeta (Index professionum)	Ukupno sati (Quot huiusmodi)					INFORMATIKA
		predavanje	laboratorijske	vježbe	seminari	izvješće	
DR. sc. MARIJA PEROVIĆ	DIFERENCIJALNA MATEMATIKA I	30	30	0			Nastavnik potvrđuje uredno izvršenje obveza (Magister testator effectorum processorum executionis)
DR. sc. Vesimir TOPALEVCIĆ	OBIKTNO ORIENTIRANO PROGRAMIRANJE	30	30	0			Uspjeh na pojedinim ispitima (singularem exanimis evenio)
DR. sc. Nevenka MILANOVIC	NUMERIKA MATEMATIKA	30	30	0			ECTS bodovi (ECTS puncta)
DR. sc. Mire PAVLOVIĆ	MODELLIRANJE PROCESA	30	30	0			Potvrda o upisu i ovjeri semestra Cetvrtotvorisna inscriptio et semestris probatio
DR. sc. SANJA MATEJANČIĆ	FORMALNI JEZICI I JEZIČNI PROSESORI I	30	30	0			
DR. sc. Miroslav PAVEŠIĆ 1	TEORIJA INFORMATIČKOG PRAVILA	30	30	0			
							semestar-godinu ovjeriva: (Semestre-semim probac.)
							Dekan (Decans)

Slika 9.18 Indeks (unutrašnja strana)

Nacrtajte dva DEV-a, za svaku sliku po jedan DEV odvojeno, a onda spojite ta dva DEV-a u jedan integralni model podataka. Dobiveni model prevedite u relacijsku shemu baze podataka.

Na DEV dodajte podatke u slučaju sljedećih zahtjeva:

- „Prijavnica“ za ispit je dokument (ili dio poslovnog softvera) kojom student potvrđuje svoju namjeru da će pristupiti ispit u planiranome roku. Dodajte na model podatke s prijavnice i to: ime i prezime studenta, naziv kolegija, datum prijave, datum polaganja ispita. Dodajte i podatke o samom ispitu: koji profesor je izveo ispit, koja je ocjena (pamte se i negativne ocjene). Negativne ocjene se na upisuju u indeks.
- U modelu se ne pamte svi prošli Planovi kolegija. Kako se i kojim SQL upitom ti podaci mogu dobiti iz postojeće BP?
- Što bi bilo ako bismo ukinuli fizički dokument Indeks? Kako se mijenja DEV ako umjesto Indeksa uvedemo na razini države jedinstvenu studentsku iskaznicu?



Slika 9.19 DEV Indeks

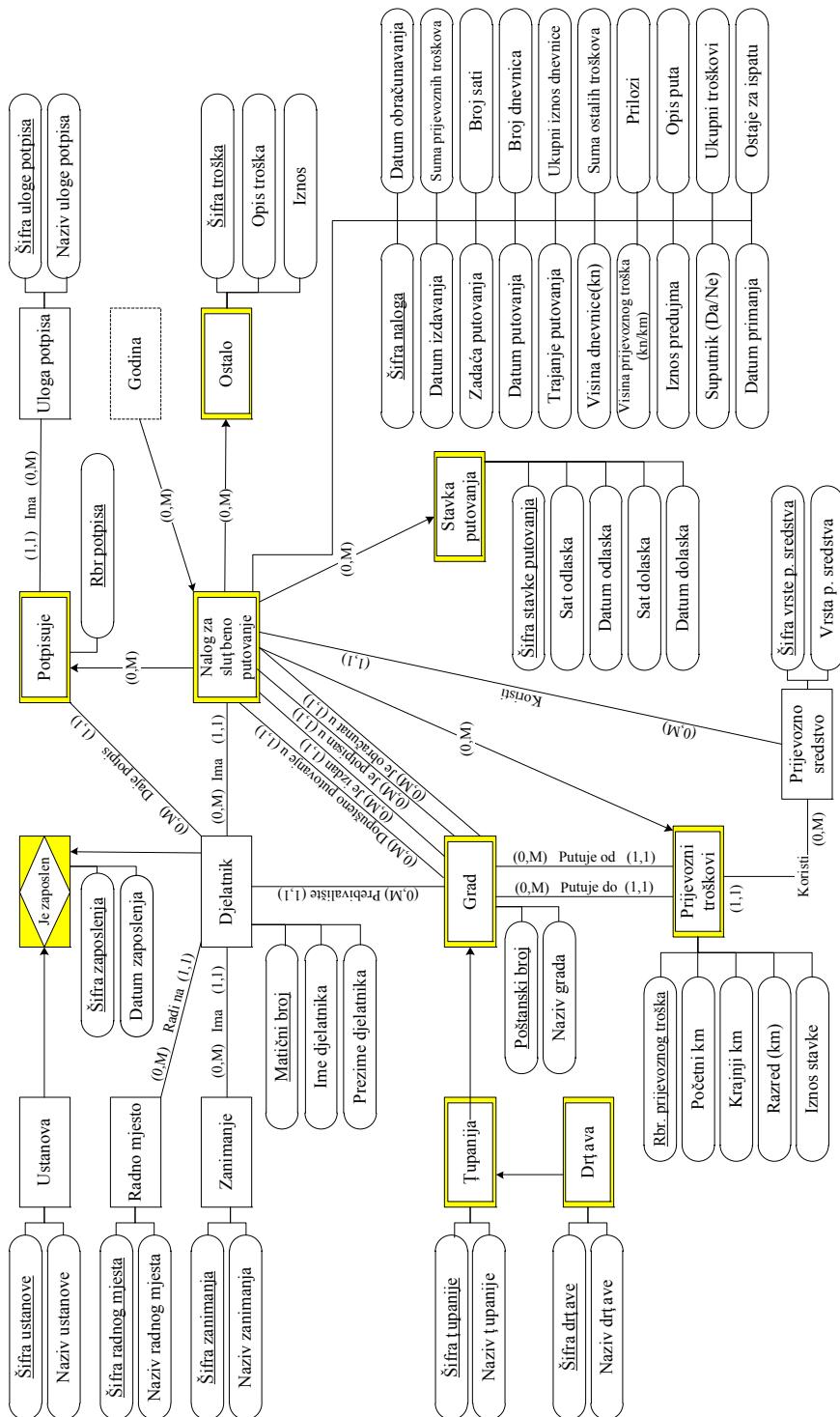
9.11 Putni nalog

Putni nalog je dokument kojim tvrtka nalaže i odobrava jednome radniku da ide na putovanje nekom vrstom prijevoza (vlak, avion, auto, brod) kako bi mogao obaviti potreban zadatak za ostvarenje poslovanja. Putovanje ima trajanje i odgovarajuću dnevnicu za svaki dan putovanja i druge pripadne troškove.

Na putnometu nalaze se sljedeći atributi: naziv trgovackog društva – ustanove, mjesto i datum izdavanja putnoga naloga, šifra putnoga naloga, ime i prezime osobe koja putuje, zanimanje osobe, prebivalište osobe, radno mjesto osobe, planirani datum početka službenoga putovanja, mjesto putovanja, zadaća putovanja, trajanje putovanja u danima, vrsta prijevoznoga sredstva, visina dnevnice, visina prijevoznoga troška, predujam, oznaka osoba je suputnik (da/ne), datum izdavanja putnoga naloga, ime i prezime direktora koji je potpisao putni nalog, datum odlaska na put, sat odlaska na put, datum povratka, sat dolaska, broj sati putovanja, broj dnevica, ukupni iznos dnevnice, relacija putovanja od, relacija putovanja do, stanje brojčanika prije, stanje brojčanika poslije(u km), razred putovanja, vrsta prijevoznoga sredstva, iznos pojedinoga troška prijevoznoga sredstva, suma prijevoznih troškova, redni broj ostalog troška, opis ostalog troška, iznos ostalog troška, suma ostalih troškova, prilozi, ukupni troškovi putnoga naloga, iznos isplaćenoga predujma, iznos ostaje za isplatu, opis puta, datum ispunjavanja putnoga naloga, mjesto ispunjavanja putnoga naloga, ime i prezime likvidatora, ime i prezime podnositelja računa, ime i prezime blagajnika, ime i prezime primatelja.

trgovačko društvo - ustanova FAROS D.O.O.		mjesto i datum izdavanja putnog naloga RIJEKA, 07.07.2010 Šifra 201007071					
Određujem da: ime i prezime MARINELLA HODAK prebivalište RIJEKA		zanimanje dipl.oec na radnom mjestu Korisnička podrška					
službeno otputuje dana 08.07.2010.	u mjesto GRAD ZAGREB	sa zadaćom Obuka korisnika					
trajanje putovanja 1 dana	odobravam upotrebu (vrsta prijevoznog sredstva) SLUŽBENI AUTOMOBIL RI 395 NI PASSAT						
visina dnevnice (kn) 170.00	visina prijevoznog troška (kn / km) 0.00	predujam (kn) 0,00	Osoba je suputnik (da / ne) NE				
08.07.2010. datum i potpis primaoca		M.P. _____ Direktor					
OBRAČUN DNEVNICA							
ODLAZAK		POVRATAK		broj sati	broj dnevnic		
datum	sat	datum		visina dnevnice	UKUPNI IZNOSI		
08.07.2010	07:40	08.07.2010	16:15	8,583	0,50		
OBRAČUN PRIJEVOZNIH TROŠKOVA							
RBR.	RELACIJA PUTOVANJA (od mjesta do mjesta)		Stanje brojčanika		razred (u km)	vrsta prijevoznog sredstva	0,00
			Prije	Poslije			
1	RIJEKA - GRAD ZAGREB		56 589	56 759	170	SLUŽBENI AUTOMOBIL	0,00
2	GRAD ZAGREB - RIJEKA		56 759	56 929	170	SLUŽBENI AUTOMOBIL	0,00
SUMA PRIJEVOZNIH TROŠKOVA						0,00	
OBRAČUN OSTALIH TROŠKOVA						SUMA OSTALIH TROŠKOVA	0,00
Prilozi:						UKUPNI TROŠKOVI	85,00
						Minus PREDUJAM	0,00
						OSTAJE ZA ISPLATU	85,00
Opis puta: Putovanje u Zagreb u tvrtku Trgopromet d.o.o. od dana 08.07.2010 do dana 08.07.2010 sa zadaćom obukom korisnika (aplikacija Faros).							
u RIJEKA dana 09.07.2010		M. P. _____		Direktor			
Likvidator		Podnositelj računa		Isplatio blagajnik			
Primalac							

Slika 9.20 Nalog za službeno putovanje



Slika 9.21 DEV Nalog za službeno putovanje

9.12 Tečajna lista

Tečajna lista je dokument koji izdaje svaka banka svaki dan kada ima promjenu tečajeva valuta. Na tečajnoj listi je dan broj (tečaj) koji kaže koliko domicilne valute te zemlje treba dati za jednu (ili 100) jedinicu valute neke druge zemlje. Kupovni tečaj je onaj po kome banka otkupljuje od stranaca njihovu valutu, a prodajni tečaj (nešto viši broj) je onaj po kome banka prodaje građanima ili tvrtkama stranu valutu.

Načinite model podataka tečajne liste koja se uobičajeno objavljuje u dnevnom tisku. Oblik dokumenta je prikazan u sljedećoj tablici.

Tablica 9.1 Tečajna lista Zagrebačke banke

Zagrebačka banka Rijeka Utvrđena na dan: 25.12.2012.			Tečajna lista br. 29		
Šifra valute	Oznaka valute	Jedinica mjere	Kupovni	Srednji	Prodajni
036	AUD	1	4.0831	4.1036	4.1242
040	ATS	100	52.2094	52.4718	52.7342
280	DEM	100	367.1550	369.0000	370.8450
...					
...					
...					
...					
...					
840	USD	1	5.4060	5.4332	5.4603
954	EU	1	6.7347	6.7686	6.8024

Proširite dobiveni DEV sljedećim zahtjevima:

- U bazi podataka treba pratiti tečajne liste više različitih banaka i institucija
- Moguća je promjena tečaja i više puta dnevno, odnosno ista banka može tijekom dana izdati više tečajnih listi
- Moguće je da neke tečajne liste nekih banaka imaju i više od tri tečaja
- Neke zemlje mogu imati više valuta i neke valute mogu važiti u više zemalja (Euro)
- Banke mogu raditi na području više zemalja, ali za svaku banku se treba znati u kojoj zemlji je izdala tečajnu listu.

Dobiveni DEV prevedite u relacijsku shemu.

10 LITERATURA

- (Alagić, 1984.) S. Alagić, "Relacijske baze podataka", Svjetlost, Sarajevo, 1984.
- (Anić, 2003.) V. Anić, „Veliki rječnik hrvatskoga jezika“, Novi liber, Zagreb, 2003.
- (Anić, Goldstein, 1999.) V. Anić, I. Goldstein, I. "Rječnik stranih riječi", Novi Liber, Zagreb, 1999.
- (Anić, Rončević, Cikota, Goldstein, Jojić, Matasović, Pranjković, 2002.) V. Anić, D. B. Rončević, LJ. Cikota, I. Goldstein, LJ. Jojić, R. Matasović, I. Pranjković, "Hrvatski enciklopedijski rječnik", Novi Liber, Zagreb, 2002.
- (Atzeni, 1983.) P. Atzeni i E. Carboni, "INCOD (A System for Interactive Conceptual Design) Revisited After the Implementation of a Prototype", vidi (Davis, 1983.).
- (Avison, 1995.) D.E. Avison and G. Fitzgerald, "Information System Development: Methodologies, Techniques and Tools", McGraw-Hill, London, 1995.
- (Bachman, 1969.) C. W. Bachman, "Data Structure Diagrams", Database 1, 2, pp. 4-10, 1969.
- (Barker, 1990.) Richard Barker, "CASE*Method Entity Relationship Modelling - ORACLE", Addison-Wesley P.C., Wokingham, England, 1990.
- (Beizer, 1990.) B. Beizer, "Software Testing Techniques"; Boston, International Thompson Computer Press, 1990.
- (Beizer, 1995.) B. Beizer, "Black Box Testing", New York, John Wiley & Sons, Inc., 1995.
- (Blaha, 2010.) M. Blaha, "Patterns of Data Modeling", CRC Press, NY, 2010.
- (Boggs, 2003.) W. Boggs, M. Boggs, "UML sa Rational Rose-om 2002", Kompjuter Biblioteka, Čačak, 2003., p.736
- (Brodie, 1984.) M. L. Brodie, J. Mylopoulos, J. W. Schmidt, "On Conceptual Modelling", Springer-Verlag, NY, 1984.
- (CARNet, 2010.) Hrvatska akademska i istraživačka mreža CARNet, tvrtka Pro Leksis d.o.o., (2010.), *Proleksis enciklopedija - prva hrvatska opća i nacionalna online enciklopedija: <http://enciklopedija.carnet.hr/login.aspx>*
- (Chaing, 1980.) T. C. Chaing, R. F. Barger, "A Database Management System With an E-R Conceptual Model", 1980, vidi (Chen, 1980).

- (Chan, 1980.) E.P.F. Chan i F.H. Lochovsky, "A Graphical Database Design Aid Using the Entity-Relationship Diagram", (vidi Chen, 1979.), str. 303-318.
- (Chen, 1976.) P. P. Chen, "The Entity-Relationship Model-Towards a Unified View of Data", ACM TODS, Vol.1, No.1, 1976.
- (Chen, 1980.) P. P. Chen, "Entity-Relationship Approach to System Analysis and Design", North-Holland, Amsterdam, 1980.
- (Chen, 1983.) P.P. Chen (Editor), "Entity-Relationship Approach to Software Engineering, Anaheim, North-Holland, Amsterdam, the Netherlands, 1983.
- (Chen, 2001.) I. J. Chen, "Planning for ERP systems: analysis and future trend", Business Process Management Journal, 7, pp.374-386, 2001.
- (Coad, 1991.) Coad P., Yourdon E., "Object-Oriented Analysis", Yourdon Press, New Jersey, 1991.
- (Database eLearning, 2010.) <http://db.grussell.org/section002.html>
- (Davis, 1983.) C.G. Davis, S. Jajodia, P. A-B. Ng. i R.T. Yeh (Editors), "Entity-Relationship Approach to Software Engineering", North-Holland, Amsterdam, the Netherlands, 1983.
- (ER, 1997.) Conceptual Modeling – ER '97, Proceedings of 16th International Conference on Conceptual Modeling, Ed. Embley, D. W., Goldstein R. C., Los Angeles, 1997., p. 1-477
- (Esculier, 1980.) C. Esculier, A. M. Glorieux, "The Sirius-Delta Distributed DBMS", vidi (Chen, 1980.), 1980.
- (Fertalj, 2006.) K. Fertalj, V. Mornar, i dr. "Komparativna analiza programske potpore informacijskim sustavima u Hrvatskoj", 5. svibnja 2006., <<http://www.zprfer.hr/projekti/erp/Dokumenti/ERP-HR.pdf>>, 28. siječnja 2007.
- (Ford, 1985.) R.M. Ford, "A Data Manager Using Entity- Relationship", PC Tech Journal for IBM Personal Computer Users, Vol.3, N0.10, Octobar 1985.
- (Giorgio, 1987.) B. Giorgio i E. Antonio, "Extending the Entity - Relationship Approach for Dynamic Modeling Purposes", vidi (Spaccapietra, 1987.).
- (Goldberg, 1986.) E. Goldberg, "Excelerator aids Straids adoption", Comuter World, 21.april 1986.
- (Hak, 2010.) http://www.hak.hr/media/35177/osobna_iskaznica.jpg (12.04.2010.)
- (Hammer, 1980.) Hammer, McLeod, "Semantis Data Model", TODS ACM, 1980.
- (Hernandez, 1997.) M. J. Hernandez, "Database Design for Mere Mortals", Addison-Wesley Developers Press, New York, 1997., p. 440
- (IBM, 1985.) IBM, "System Development Method (SDM) - System Design, IBM, 1985.

- (Iossiphidis, 1980.) J. Iossiphidis, "A Translator to Convert the DDL of ERM to the DDL of System 2000", vidi (Chen, 1980.).
- (Jajodia, 1983.) S. Jajodia i dr, "On Universal and Representative Instances for Inconsistent Databases", vidi (Davis, 1983.).
- (Junet, 1987.) M. Junet, "Design and Implementation of an Extended Entity - Relationship Data Base Management System (ECRINS/86 ", vidi (Spaccapietra, 1987.).
- (Kangassalo, 1991.) Hannu Kangassalo (Editor), "Entity - Relationship Approach: The Core of Conceptual Modelling, Proceedings of the Ninth International Conference on the Entity - Relationship Approach", Lausanne, Switzerland, 8.-10. October, 1990., North - Holland, Amsterdam, the Netherlands, 1991.
- (Kellner, 1999.) M. I. Kellner, R. J. Madachy, D. M. Raffo, "Software processes simulation modelling: Why? What? How? ", J.Syst.Software, 46, 91-105, 1999.
- (Kiš, 2002.) M. Kiš, "Englesko-hrvatski i hrvatsko-engleski INFORMATIČKI RJEĆNIK", Naklada Ljevak, Zagreb, 2002.
- (Kiš, 1993.) M. Kiš, J. Buljan, S. Vuković, O. Anić, "Englesko-hrvatski informatički rječnik s računalnim nazivljem", Školska Knjiga, Zagreb, 1993.
- (Lazarević, 1985.) B. Lazarević, V. Jovanović, P. Dizdarević i M. Vučković, "Projektiranje informacijskih sistema", I i II dio, Naučna knjiga, Beograd, 1985.
- (Lazarević, 2008.) B. Lazarević, Z. Marjanović, N. Aničić, S. Babarogić, "Baze podataka", FON, Beograd, 2008.
- (Lien, 1980.) E. Y. Lien, "Semantic of Entity-Relationship Model", vidi (Chen, 1980.).
- (Linthicum, 2000.) D. S. Linthicum, "Enterprise Application Integration", Addison-Wesley Information Technology Series, Boston, 2000., p. 377
- (Longworth, 1989.) G. Longworth, D. Nicholls, "SSADM MANUEL Techniques and documentation", National Computing Centre, England, 1989.
- (Lusk, 1980.) E. L. Lusk i R. A. Overbeek, "A DML for Entity- Relationship Models", 1980 vidi (Chen, 1980.).
- (Markowitz, 1983. a) V. M. Markowitz i Y. Raz, "ERROL: An Entity- Relationship, Role Oriented, Query Language", vidi (Davis, 1983.).
- (Markowitz, 1983.) V. M. Markowitz i Y. Raz, "A Modified Relational Algebra and its Use in an Entity-Relationship Environment", vidi (Davis, 1983.).
- (Marti, 1983.) R. W. Marti, "Integrating Database and Program Descriptions Using on ER-Data Dictionary", vidi (Davis, 1983.).
- (Martin, 1977.) J. Martin, "Computer Data - Base Organization", Prentice - Hall, New Jersey, 1977.
- (Martin, 1985.) J. Martin, "Banka podataka za krajnjeg korisnika" Informatika i društvo, Zagreb, 1985.

- (Martin, 1985.) J. Martin, & C. McClure, "Diagramming Techniques for Analysts and Programmers". Englewood Cliffs, NJ: Prentice Hall, Inc., 1985.
- (Martin, 1985.) J. Martin, C. McClure, "Action Diagrams: Clearly Structured Program Design", Prentice Hall, Englewood Cliffs, N.J. 1985.
- (Martin, 1986.) J. Martin, "Fourth Generation Languages", Prentice Hall, Vol 1 i Vol 2, 1985.-1986.
- (Melkanoff, 1980.) M. A. Melkanoff, C. Zaniolo, "Decomposition of Relations and Synthesis of Entity Relationship Diagrams", vidi (Chen, 1980.), 1980.
- (Mogin, 2000.) P. Mogin, V. Luković, M. Govedarica, "Principi projektovanja baza podataka", Univerzitet u Novom sadu, Novi sad, 2000., p. 699
- (Munz, 1980.) R. Munz, "Design of WELLSystem", 1980., vidi (Chen, 1980.).
- (Nakata, 1983.) S. Nakata, G. Yamazaki, "A System Based on the E-R Model and its Application to Database-Oriented Tool Generator", 1983., vidi (Davis, 1983.).
- (Navathe, 1983.) S. B. Navathe, A. Cheng, "A Methodology for Database Schema Mapping from Extended Entity-Relationship Models into the Hierarchical Model", 1983., vidi (Davis, 1983.).
- (Neufeldt, 1991.) V. Neufeldt, D. B. Guralnik, "Webster's New World Dictionary", Prentice Hall, New York, 1991.
- (Ng, 1980.) P.A.Ng i J.F. Paul, "A Formal Definition of Entity- Relationship Models", (vidi Chen, 1980.).
- (Papp, 1979.) L. Papp, "A konyvviteli informacions rendszer elmeleti kerdesei", Tan-konyvkiado, Budapest, 1979.
- (Parent, 1985.) C. Parent i S. Spaccapietra, "An Algebra for General Entity-Relationship Model", IEEE Transaction on Software Engineering, Volume SE-11, Number 7, pages 634-643, July 1985.
- (Pavlić, 1986.) M. Pavlić, V. Jovanović, P. Dizdarević, Z. Jugo, "Projektiranje informacijskog sustava kalkulacija cijene broda primjenom metode SSA i E-V", III savjetovanje za informatičku djelatnost, Privredna komora Rijeka, Opatija, 1986.
- (Pavlić, 1992.) M. Pavlić, "Apstrakcija metodologije modeliranja na području projektiranja informacijskih sustava - doktorski rad", FOI, Varaždin, 1992.
- (Pavlić, 1996.) M. Pavlić, "Razvoj informacijskih sustava - projektiranje, praktična iskustva", metodologija, Znak, Zagreb, 1996.
- (Sakai, 1980.) H. Sakai, "A Unified Approach to the Logical design of a Hierarchical Data Model", 1980., vidi (Chen, 1980.).
- (Sakai, 1983.) H. Sakai, "Entity-Relationship Approach to Logical Database Design", 1983., vidi (Davis, 1983.).

- (Santos, 1980.) C.S. dos Santos, E.J. Neuhold i A.L. Furtado, "A Date Type Approach to the Entity-Relationship Model", (vidi Chan, 1980.).
- (Schuldt, 1987.) G. Schuldt, " ER - Based Access Modeling", vidi (Spaccapietra, 1987.).
- (Tardieu, 1980.) H. Tardieu, H. Heckenroth, D. Pascot i D. Nanci, "A Method, A Formalism and Tools for Database Design (Three Years of Experimental Practice)", 1980., vidi (Chen, 1980.).
- (Tsichritzis, 1982.) D.C. Tsichritzis , F.H. Lochovsky , "Data Models", Prentice Hall, 1982.
- (Ursprung, 1983.) P. Ursprung i C.A. Zemnder, "HIQUEL: An Interactive Query Language to Define and Use Hierarchies", (vidi Davis, 1983.).
- (Varga, 1994.) Mladen Varga, "Baze podataka - konceptualno, logičko i fizičko modeliranje podataka", DRIP, Zagreb, 1994.
- (Vetter, 1987.) M. Vetter, "Strategy for Data Modelling, Application and Enterprise-wide", John Wiley and sons, Chichester, 1987., p. 344
- (Vujnović, 1995.) R. Vujnović, "SQL i relacijski model podataka" , Znak, Zagreb, 1995.
- (Wikipedia, 2010.) <http://en.wikipedia.org/wiki/Categorization> (09.05.2010.)
- (Wikipedia, 2010.) Wikipedija: http://en.wikipedia.org/wiki/Peter_Chen, 22.04.2010.
- (Wikipedija, 2010.) http://hr.wikipedia.org/wiki/Informacijski_sustavi (19.04.2010.)
- (Yao, 1985.) S. Bing Yao, "Principles of Database Design", Prentice - Hall, New Jersey, 1985.
- (Yourdon, 1979.) E. Yourdon, L. L. Constantine, "Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design", facsimile edn, Prentice Hall, 1979.
- (Zhang, 1983.) Z.Q. Zhang i A.O. Mendelzon, "A Graphical Query Language for Entity-Relationship Databases", 1983., vidi (Davis, 1983.).

