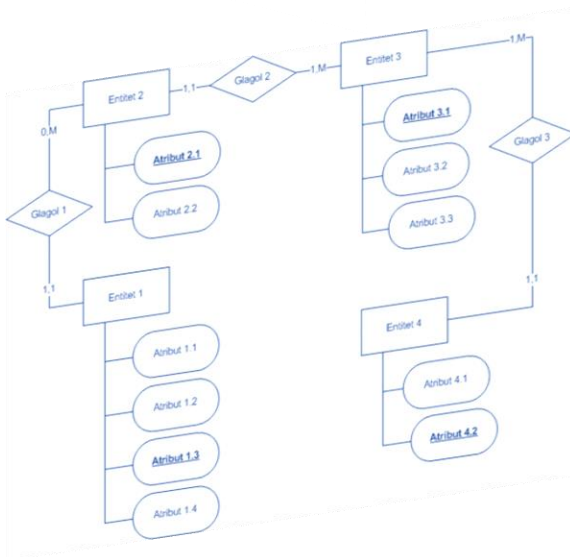
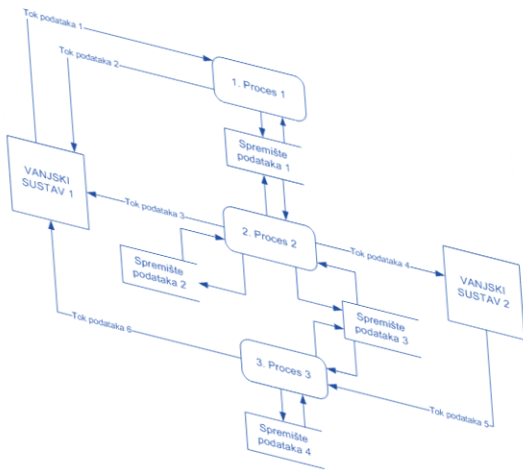


# MPP



## MODELIRANJE PODATAKA I PROCESA



# MPP

Ida Panev  
Marin Kaluža  
2022.

*Dr. sc. IDA PANEV, v. pred.*

*Dr. sc. MARIN KALUŽA, prof. v. š.*

## **MODELIRANJE PODATKA I PROCESA**

*Nakladnik:*

Veleučilište u Rijeci  
Trpimirova 2/V, Rijeka

*Za nakladnika:*

dr. sc. Saša Hirnig, prof. v. š.

*Recenzenti:*

izv. prof. dr. sc. Sanja Čandrlić, Sveučilište u Rijeci, Odjel za informatiku  
dr. sc. Bernard Vukelić, prof. v. š., Veleučilište u Rijeci

*Lektorica:* doc. dr. sc. Sanja Grakalić Plenković

*Elektroničko izdanje*

Nastavna skripta „*Modeliranje podataka i procesa*“ intelektualno je vlasništvo, neotuđivo, zakonom zaštićeno i mora se poštovati. Nijedan dio ove publikacije ne smije se preslikavati, umnožavati ili na bilo koji drugi način reproducirati, uključujući *web*-distribuciju i sustave za pretraživanje te skladištenje podataka, bez pisanoga dopuštenja izdavača.

**ISBN: 978-953-6911-99-8**

Povjerenstvo za izdavačku djelatnost Veleučilišta u Rijeci odobrilo je izdavanje ove nastavne skripte (KLASA: 003-11/20-01/10, UR. BROJ: 2170-57-01-20-24/MJG).

## **Predgovor:**

Nastavni materijal iz kolegija Modeliranje podataka i procesa izrađen je s ciljem da se studentima 2. godine stručnog studija Informatika na Veleučilištu u Rijeci pruže informacije o modeliranju podataka i o modeliranju procesa kao početnih koraka u izgradnji informacijskog sustava odabranog poslovnog sustava.

U prvom dijelu nastavnog materijala studente se upoznaje s teorijskom podlogom, odnosno pojmovima *sustav, poslovni sustav, podatak, informacija, informacijski sustav, model, modeliranje i modeli informacijskog sustava*.

Nadalje, studente se upoznaje s pojmovima *model podataka, EVA model podataka, s konceptima EVA modela podataka, notacijama EVA modela, pravilima modeliranja te koracima u izgradnji EVA modela*. U sljedećem poglavlju studente se upoznaje s pojmovima *relacijske sheme, relacijskog modela podataka, pravilima prevođenja EVA modela podataka u relacijski model podataka te relacijskom algebrom*. Dalje se u nastavnom materijalu opisuje *funkcijska zavisnost, redundancija podataka, anomalije redundancije, normalizacija podataka* te se daje primjer normalizacije relacijske sheme do treće normalne forme.

Sljedeći dio nastavnog materijala upoznaje studente s pojmovima *model procesa, dijagram dekompozicije procesa, dijagram toka podataka (DTP), s konceptima DTP-a te pravilima kreiranja DTP-a*.

U posljednjem dijelu nalaze se zadaci i rješenja na temelju kojih studenti mogu uvježbavati izradu modela podataka i modela procesa.

Ovaj nastavni materijal namijenjen je studentima kao osnovna literatura u okviru kolegija Modeliranje podataka i procesa.

Rijeka, 2022.

Autori

## Sadržaj:

<b>1. SUSTAV .....</b>	<b>1</b>
1.1. VEZA SUSTAVA I OKOLINE .....	2
1.2. KONSTRUKTIVNI ELEMENTI SUSTAVA.....	3
1.3. ZAHTJEVI ZA SUSTAV.....	3
1.4. OBILJEŽJA (KARAKTERISTIKE, OSOBINE) SUSTAVA.....	4
1.5. ORGANIZACIJSKI (POSLOVNI) SUSTAV .....	5
<b>2. PODACI I INFORMACIJE .....</b>	<b>6</b>
<b>3. INFORMACIJSKI SUSTAV (IS).....</b>	<b>8</b>
3.1. ELEMENTI RAČUNALOM PODRŽANOG INFORMACIJSKOG SUSTAVA .....	9
3.2. ZAŠTO JE POTREBAN INFORMACIJSKI SUSTAV? .....	10
3.3. CILJEVI (TEMELJNE FUNKCIJE) INFORMACIJSKOG SUSTAVA .....	11
3.4. POLOŽAJ INFORMACIJSKOG SUSTAVA U POSLOVNOJ ORGANIZACIJI .....	11
<b>4. PROJEKT INFORMACIJSKOG SUSTAVA .....</b>	<b>13</b>
4.1. MODELI INFORMACIJSKOG SUSTAVA .....	14
4.2. MODELIRANJE INFORMACIJSKOG SUSTAVA.....	14
4.3. VRSTE MODELA INFORMACIJSKOG SUSTAVA .....	15
<b>5. MODEL PODATAKA.....</b>	<b>22</b>
5.1. MODELIRANJE PODATAKA .....	23
5.1.1. POLOŽAJ MODELA PODATAKA .....	23
5.1.2. STRUKTURA, OGRANIČENJA I OPERATORI U MODELU PODATAKA.....	24
5.1.3. MODELI PODATAKA PREMA RAZINI APSTRAKCIJE .....	24
5.1.4. ZAŠTO IZRADITI MODEL PODATAKA?.....	25
5.1.5. IDENTIFIKACIJA RELEVANTNIH OBJEKATA ZA MODELIRANJE PODATAKA IZ OPISA POSLOVNOG SUSTAVA.....	25
5.2. EVA MODEL PODATAKA.....	26
5.2.1. ENTITET KAO KONCEPT EVA MODELA PODATAKA.....	27
5.2.2. VEZA KAO KONCEPT EVA MODELA PODATAKA.....	27
5.2.2.1. VRSTE VEZA PREMA REDU VEZA U EVA MODELU PODATAKA.....	25
5.2.3. ATRIBUT KAO KONCEPT EVA MODELA PODATAKA .....	28
5.2.4. IDENTIFIKATOR NA EVA MODELU PODATAKA .....	31
5.2.5. ZAVISNOST KAO KONCEPT EVA MODELA PODATAKA.....	32
5.2.6. AGREGIRANI TIP ENTITETA KAO KONCEPT EVA MODELA .....	33
5.2.7. DOKUMENTACIJA MODELA PODATAKA .....	35
5.3. RELACIJSKI MODEL PODATAKA .....	36
5.3.1. USPOREDBA KONCEPATA EVA I RELACIJSKOG MODELA PODATAKA .....	38

5.3.2.	PREVOĐENJE EVA MODELA U RELACIJSKI MODEL PODATAKA .....	39
5.4.	RELACIJSKA ALGEBRA.....	46
5.4.1.	NASLIJEĐENI RELACIJSKI OPERATORI .....	47
5.4.1.1.	RELACIJSKI OPERATOR <i>UNIJA</i> .....	47
5.4.1.2.	RELACIJSKI OPERATOR <i>PRESJEK</i> .....	48
5.4.1.3.	RELACIJSKI OPERATOR <i>RAZLIKA</i> .....	48
5.4.2.	OSNOVNI RELACIJSKI OPERATORI .....	50
5.4.2.1.	RELACIJSKI OPERATOR <i>PROJEKCIJA</i> .....	50
5.4.2.2.	RELACIJSKI OPERATOR <i>SELEKCIJA</i> .....	51
5.4.2.3.	RELACIJSKI OPERATOR <i>PRIRODNI SPOJ</i> .....	52
5.4.2.4.	RELACIJSKI OPERATOR <i>PREIMENOVANJE</i> .....	57
5.4.3.	IZVEDENI RELACIJSKI OPERATORI .....	58
5.4.3.1.	RELACIJSKI OPERATOR <i>PRODUKT</i> .....	58
5.4.3.2.	RELACIJSKI OPERATOR <i>AKTIVNI KOMPLEMENT</i> .....	58
5.4.3.3.	RELACIJSKI OPERATOR <i>KVOCIJENT</i> .....	59
5.5.	FUNKCIJSKA ZAVISNOST.....	59
5.6.	REDUNDANCIJA PODATAKA.....	60
5.7.	NORMALIZACIJA PODATAKA .....	62
5.7.1.	PRIMJER NORMALIZACIJE RELACIJSKE SCHEME DO 3. NORMALNE FORME .....	65
<b>6.</b>	<b>MODEL PROCESA.....</b>	<b>71</b>
6.1.	KORACI U IZGRADNJI MODELA PROCESA.....	72
6.2.	DIJAGRAM DEKOMPOZICIJE.....	72
6.3.	DIJAGRAM TOKOVA PODATAKA (DTP) .....	73
6.3.1.	PRAVILA KREIRANJA DIJAGRAMA TOKOVA PODATAKA.....	76
6.3.2.	RAZINE DEKOMPOZICIJE DIJAGRAMA TOKOVA PODATAKA .....	78
<b>7.</b>	<b>ZADACI ZA VJEŽBU I NJIHOVA RJEŠENJA.....</b>	<b>81</b>
7.1.	IDENTIFIKACIJA RELEVANTNIH OBJEKATA IZ OPISA POSLOVNIH SUSTAVA.....	81
7.2.	IZRADA EVA MODELA NA TEMELJU OPISA POSLOVNIH PODSUSTAVA.....	84
7.3.	TEKSTUALNI OPIS PRIKAZANOG EVA MODELA PODATAKA.....	95
7.4.	PREVOĐENJE EVA MODELA PODATAKA U RELACIJSKI MODEL PODATAKA .....	97
7.5.	MODELIRANJE PROCESA .....	101
	<b>POPIS LITERATURE .....</b>	<b>110</b>
	<b>POPIS SLIKA.....</b>	<b>112</b>
	<b>POPIS TABLICA .....</b>	<b>115</b>

# 1. Sustav

Kako bi se shvatio kontekst potreban za uspješno modeliranje podataka i modeliranje procesa, potrebno je krenuti s općenitim pojmovima kao što su sustav, okolina ili informacija.

Neke od definicija pojma *sustav* su sljedeće:

**Sustav** je skup dijelova, veza među dijelovima te osobina tih dijelova i veza, a organiziranih s nekim ciljem (npr. neka određena tvrtka je jedan sustav) (Pavlič, *et al.*, 2014).

ili

**Sustav** je organizacija međuzavisnih elemenata povezanih u cjelinu po nekom planu za postizanje određenog cilja.

Jedan od primjera sustava je *živčani sustav* u organizmu čovjeka. U živčanom sustavu se nalaze *dijelovi* (mozak, leđna moždina i živci) koji imaju svoje *osobine* (npr. mozak se sastoji od milijardi neurona koji su na specifičan način povezani sinapsama, određene je veličine ili na određeni način reagira na pojedine podražaje) i koji su na određeni način međusobno *povezani* (npr. živci šalju signale mozgu) s ciljem ispravnog funkcioniranja ljudskog organizma.

Drugi primjer sustava je *ekosustav* kao skup živih bića i nežive prirode sa svim svojim osobinama, a koji međusobno djeluju i razmjenjuju tvari na zajedničkom području.

Treći primjer sustava je *transportni sustav plinovoda* koji se sastoji od plinovoda različitih osobina (npr. različitih kapaciteta ili namjena), a kojima je svrha povezati sve dijelove određenog područja s ciljem dostave plina krajnjim korisnicima.

Primjera sustava je bezbroj i može se zaključiti da je sve što nas okružuje dio nekog većeg ili manjeg sustava bilo koje vrste.

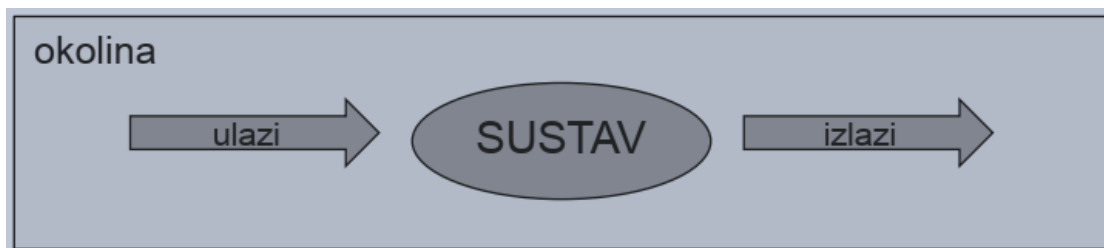
## 1.1. Veza sustava i okoline

Svaki *sustav* se nalazi u *širem sustavu* kojega je dio i s kojim je u vezi.

Dio sveukupne cjeline koji nije obuhvaćen promatranim sustavom naziva se **okolina** tog sustava.

Veza između sustava i okoline predstavlja se **ulazima** odnosno **izlazima**. Sustav koristi potrebne elemente iz okoline koji putem ulaznih kanala ulaze u sustav. Sustav takve elemente obrađuje, kombinira ih s drugim ulaznim elementima te ih prosljeđuje putem izlaznih kanala u okolinu. Drugim riječima, sustav koristi određene *ulaze*, obrađuje ih, te u okolinu isporučuje određene *izlaze* (slika 1).

Kada sustav ne bi imao ulazni ili izlazni kanal bio bi krnji sustav. Kada sustav ne bi imao ulaze pitanje je bi li postojala različitost u formiranim izlazima iz sustava ili bi izlazi bili uvijek isti. Ako je tako, upitna je svrhovitost takvog sustava. Kada sustav ne bi imao izlaze nego samo primao ulazne elemente i obrađivao ih, pitanje je potrebe za takvim sustavom, odnosno svrhovitosti i učinkovitosti takvog sustava.



Slika 1: Veza sustava i okoline

Sustav može imati više *ulaza* i *izlaza*. *Ulazi* i *izlazi* prolaze kroz granicu sustava (Pavlić, *et al.*, 2014).

**Proces** je jedan od unutarnjih dijelova sustava koji *pretvara ulaze u izlaze*.

Procesi od kojih se sustav sastoji nalaze se unutar granica sustava, a izvan granica su procesi drugih sustava koji se nazivaju *okolinom sustava*.

## 1.2. Konstruktivni elementi sustava

Da bi funkcionirao, svaki se *sustav* mora sastojati od određenih dijelova – konstruktivnih elemenata. To su (Whitten & Bentley, 2007):

- **ulazni i izlazni podaci** (dokumenti, poruke, izvješća...)
  - npr. u informacijskom sustavu za evidenciju studenata nekog fakulteta, ulazni podaci su podaci koje dostavlja student prilikom upisa (npr. ime, prezime, OIB...), a izlazni podaci su izvještaji o broju upisanih studenata u određenoj akademskoj godini koji se dostavljaju nadležnim službama;
- **proces koji se izvode u sustavu** (postupci izvođenja, upravljanja, odlučivanja, planiranja i nadzora nad podacima)
  - npr. u informacijskom sustavu za evidenciju studenata nekog fakulteta odvija se proces evidentiranja pristiglih podataka o studentima u sustav;
- **fizički dijelovi sustava** (uređaji, alati, pomagala... koji omogućuju da sustav funkcionira i izvršava svoju funkciju)
  - npr. da bi informacijski sustav za evidenciju studenata nekog fakulteta funkcionirao, potreban je server na kojem su podaci pohranjeni te računalo i aplikativno rješenje putem kojih korisnik pristupa tim podacima.

## 1.3. Zahtjevi za sustav

Da bi uspješno izvršavao svoju funkciju, svaki *sustav* mora biti:

- **Prilagodljiv okolini** - okolina je konstantno podložna promjenama. Sustav funkcionira u toj promjenljivoj okolini, zbog čega treba biti *stabilan* i *dugotrajan*.
- **Upravljiv** - da bi sustav izvršavao svoju funkciju, njime se mora moći upravljati kako bi se postigao željeni *izlaz* iz sustava (kako bi se dobile odgovarajuće informacije kao rezultat djelovanja sustava). Željeni *izlaz* iz sustava postiže se promjenama: *ulaznih informacija* i/ili *strukture samog sustava*.



#### 1.4. Obilježja (karakteristike, osobine) sustava

Svaki *sustav* ima sljedeće karakteristike (Kaluža, 2018):

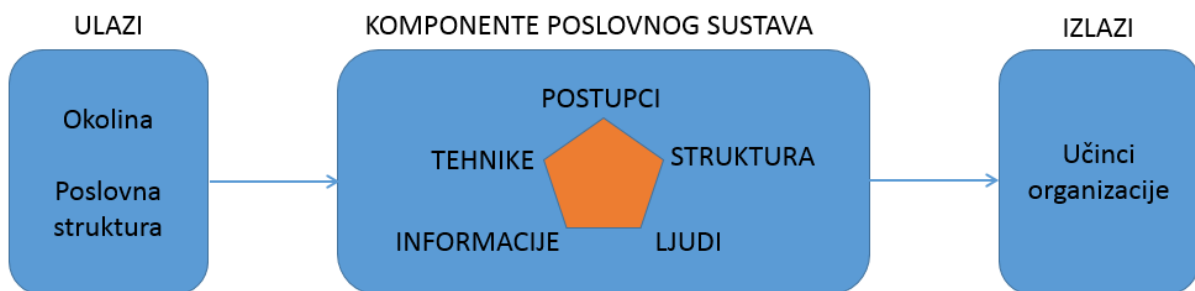
- **Postoji s razlogom** - ima svoju svrhu, razlog, cilj postojanja (npr. informacijski sustav nekog fakulteta izrađen je u svrhu evidencije svih informacija potrebnih za funkcioniranje tog fakulteta kao organizacije);
- **Sastoji se od podsustava** - može se podijeliti (dekomponirati) na manje podsustave (npr. informacijski sustav nekog fakulteta može se dekomponirati na podsustav za evidenciju studenata, podsustav za evidenciju zaposlenika, podsustav za evidenciju kolegija itd.);
- **Ima svoj opseg** - ima poznate granice, širinu svoga djelovanja (npr. informacijski sustav nekog fakulteta obuhvaća samo one instance koje se odnose na funkcioniranje tog fakulteta kao organizacije);
- **Ima svoj doseg** - ima poznatu dubinu, potrebnu detaljnost (procesе i njihove podprocesе) za izvršavanje svog cilja (npr. informacijski sustav nekog fakulteta sastoji se od podprocesа koji mogu pohraniti sve, i najjednostavnije, podatke koji su potrebni za njegovo funkcioniranje);
- **Komunicira s okolinom** - djeluje u okruženju, unutar određenog konteksta, tj. komunicira s drugim sustavima iz okoline (npr. informacijski sustav fakulteta razmjenjuje relevantne podatke s nadležnim službama);
- **Ima ulaze** - kroz koje dobiva informacije koje koristi u provođenju internih aktivnosti, tj. za izvršavanje procesа (npr. informacijski sustav nekog fakulteta dobiva potrebne informacije od samih studenata tijekom njihova upisa na fakultet);
- **Ima izlaze** - na temelju ulaznih informacija, djelovanjem procesа stvara izlazne informacije kao rezultat svog rada te ih isporučuje u okolinu (npr. informacijski sustav nekog fakulteta na temelju pohranjenih ulaznih informacija stvara izlazne informacije u obliku izvješća koja dostavlja nadležnim službama);
- **Ima sučelje** - unaprijed definirane procedure i postupke za provođenje aktivnosti, tj. izvođenje svih procesа (npr. informacijski sustav nekog fakulteta ima aplikativno rješenje uporabom čijeg sučelja odgovorni zaposlenici mogu pristupati željenim informacijama);
- **Ima ograničenja** - nametnuta pravila, tj. zakonitosti pri izvođenju aktivnosti, poznate skupove korektnih (odobrenih, dozvoljenih) vrijednosti i mogućnost prepoznavanja nekorektnih

(neodobrenih) vrijednosti (npr. u informacijskom sustavu nekog fakulteta poslovna logika nalaže da student ne može istovremeno biti upisan na više različitih godina studija).

## 1.5. Organizacijski (poslovni) sustav

**Poslovni (organizacijski) sustav** je složeni sustav koji se sastoji od više povezanih podsustava. Sadrži *postupke, tehnike, strukturu, ljude i informacije* kao komponente poslovnog sustava čija je svrha ispunjavanje ciljeva postojanja tog sustava. Djeluje u okolini iz koje prihvaća ulazne informacije. Na temelju njih, djelovanjem svojih komponenti, isporučuje izlazne informacije kao rezultate svoga djelovanja, a u svrhu dostizanja svoga cilja (slika 2). Poslovni sustav je npr. neka tvrtka, ustanova, udruga ili organizacija.

Svaki poslovni sustav ima svoj **informacijski sustav**.



Slika 2: Poslovni sustav (izvor: Autori)

## 2. Podaci i informacije

Prije objašnjenja pojma *informacijski sustav* potrebno je pojasniti pojmove koji su ključni za njegovo postojanje, a to su *podatak*, *informacija* i *znanje*.

**Podatak** je neka *sirova činjenica* ili *istina iz realnog svijeta*. Podatak samostalno ima malo ili nikakvo značenje (Pavlič, 1996).

ili

**Podatak** je skup prepoznatljivih znakova, odnosno simbola zapisanih na nekom mediju, npr. papiru, filmu, magnetskom ili kojem drugom mediju. Pomoću podataka zapisujemo činjenice, pritom ne razmatramo ni njihovu interpretaciju ni njihov kontekst (Varga, 2020).

Npr. „42” je broj koji se sastoji od dva znaka, no ne zna se što on znači, koja mu je semantika (značenje). Broj „42” je primjer jednog *podatka*.

Podaci se mogu zapisivati na različite medije, u analognom (npr. papir, knjiga) ili digitalnom obliku (npr. datoteka).

**Informacija** je podatak sa značenjem (interpretirani podatak, protumačeni podatak). Informacija je smisljeno (logički) obrađen podatak koji je pročišćen i organiziran (Jakupović, 2018).

ili

**Informacija** je protumačeni podatak koji primatelju donosi novost, odnosno, o nečemu ga obavještava (Varga, 2020).

Ako se, primjerice, kaže „visina kutije je 42 cm”, tada je to informacija (protumačeni podatak, podatak koji ima određeno značenje za čitatelja). Ta informacija je mogla biti zapisana i na sljedeće načine (slika 3):

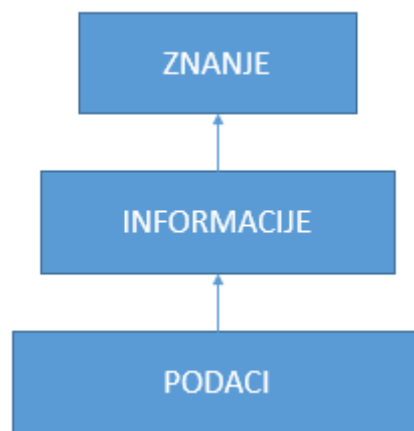
Visina kutije u cm	ili	Kutija
42		Visina u cm 42

Slika 3: Načini bilježenja informacije (izvor: Autori)

S obzirom na to da imaju značenje, informacije mijenjaju naše *znanje*.

**Znanje** je kombinacija podatka i informacije kojoj se dodaje ekspertno mišljenje, vještina i iskustvo (Jakupović, 2018). Znanje su one informacije kojima je pridružen smisao.

Na slici 4 vidljivo je strukturiranje podataka, informacija i znanja u jednu informacijsku hijerarhiju gdje svaka razina dodaje određena svojstva iznad i/ili ispod one prethodne. Podatak je najosnovnija razina, informacija joj dodaje kontekst, a znanje dodaje objašnjenje kako ga uporabiti.



Slika 4: Od podatka do znanja (izvor: Autori)

### 3. Informacijski sustav (IS)

**Informacijski sustav** je sustav koji upravlja *informacijama* koje su važne za određenu organizaciju (sredinu u kojoj se koriste).

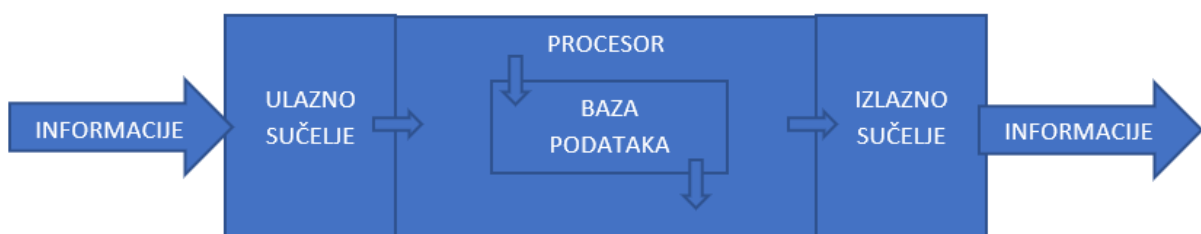
Informacijski sustav jedan je od *podstava organizacijskog (poslovnog) sustava*. Postoje dvije vrste informacijskih sustava. To su:

- **Računalom podržani informacijski sustavi** – IS-i podržani informacijskom tehnologijom (npr. Središnji informacijski sustav primarne zdravstvene zaštite temeljen na NISHI aplikaciji (engl. National Information System on Health Infrastructure) sustav je koji korištenjem virtualne privatne mreže (engl. VPN - Virtual Private Network) povezuje sve liječničke ordinacije primarne zdravstvene zaštite, Hrvatski zavod za zdravstveno osiguranje i Hrvatski zavod za javno zdravstvo);
- **Klasični informacijski sustavi** - u analognom obliku bez korištenja informacijsko-komunikacijskih tehnologija – ICT (npr. bilo koji informacijski sustav u kojem se odvija ručna obrada informacija).

**Cilj informacijskog sustava** - prikupljanje, obrada, pohranjivanje, korištenje, prikaz i distribucija informacija koje su potrebne za praćenje rada i upravljanje organizacijskim sustavom ili nekim njegovim podstavom (Pavlič, 1996).

**Računalom podržani informacijski sustav** čine (slika 5):

- računalni programi koji putem ulaznog sučelja prihvaćaju informacije o stanju organizacijskog sustava (ulazne informacije u sustav) i spremaju ih preko procesora u bazu podataka;
- računalni programi koji koriste preko procesora podatke iz baze podataka i kreiraju izvješća koja se distribuiraju putem izlaznog sučelja (izlazne informacije iz sustava).



Slika 5: Računalom podržani informacijski sustav (izvor: Autori)

### 3.1. Elementi računalom podržanog informacijskog sustava

Svaki računalom podržani informacijski sustav sastoji se od sljedećih međusobno povezanih elemenata (Srića & Spremić, 2000) vidljivih i na slici 6:

- **Hardware** (hardver), tj. sklopovlje ili dijelovi računala;
- **Software** (softver), tj. programska podrška za rad informacijskog sustava. Dijeli se na *sustavski softver*, tj. operacijski sustav, te na *aplikacijski softver*, tj. korisničke programe/aplikacije.

Hardver i softver čine *strojnu opremu* potrebnu za rad informacijskog sustava.

- **Dataware**, tj. pohrana podataka i upravljanje podacima. Pohrana podataka se izvršava u *bazu podataka*, a upravljanje podacima je moguće zahvaljujući *sustavu za upravljanje bazom podataka* (SUBP).
- **Netware**, tj. mrežna povezanost računala, odnosno komunikacijske mreže za povezivanje računala (OSI slojevi referentnog modela za mrežnu komunikaciju, protokoli (TCP/IP, UDP...) koji omogućuju komunikaciju preko raznih međusobno povezanih mreža);
- **Orgware**, tj. organizacijski postupci potrebni za funkcioniranje IS-a kao što su poslovna logika organizacije, poslovna pravila i propisi kojih se potrebno pridržavati u organizaciji te algoritmi izvođenja poslovnih procesa, projekata i programa;
- **Lifeware**, tj. ljudstvo – krajnji korisnici IS-a, administratori i developeri (projektanti, programeri) čije su uloge i prava točno definirani;



Slika 6: Elementi informacijskog sustava (izvor: Autori)

### 3.2. Zašto je potreban informacijski sustav?

Postoji niz pitanja vezanih uz postizanje ciljeva nekog *poslovnog (organizacijskog) sustava* koja se mogu postaviti promatranjem tog poslovnog sustava. Odgovori na takva pitanja mogu se dobiti kroz *izvještaje* ili *transakcije* koji se stvaraju putem funkcionalnog *informacijskog sustava*.

Slijede primjeri takvih pitanja (Jakupović, 2018):

- Kako znati je li prodavač zaradio plaću ako se ne zna koliko je prodao robe s kojom provizijom iz dana u dan?
- Kako znati je li kvaliteta izlaznog proizvoda u funkciji kvalitete ugrađenog materijala?
- Kako znati kolika je produktivnost stroja, alata, osobe ili programa, ako se ne evidentira svaki događaj?
- Kako znati koje proizvode treba imati na zalihama i koliko?
- Kako znati donosi li proizvod ili tvornica dobit?

Informacijski sustav nekog poslovnog sustava treba pomoći poslovnom sustavu u ostvarenju ciljeva poslovnog sustava. To se postiže implementacijom u korisničko sučelje informacijskog sustava svih onih poslovnih procesa kojima se pokazuje stupanj ostvarivanja ciljeva poslovnog sustava. To znači da informacijski sustav treba omogućavati *pregled i ažuriranje postojećih podataka* (npr. u informacijskom sustavu nekog fakulteta mora biti omogućeno ažuriranje podataka o studentima ako dođe do promjene adrese, prezimena i sl.) *kreiranje izvještaja* na temelju postojećih podataka (npr. u informacijskom sustavu nekog fakulteta mora biti omogućeno kreiranje izvještaja za nadležne službe o upisanim studentima u nekoj akademskoj godini), *transakcijsku obradu podataka* (npr. u informacijskom sustavu nekog fakulteta to može biti automatizirani upis studenta u višu godinu studija nakon što je zadovoljio sve kriterije) itd.

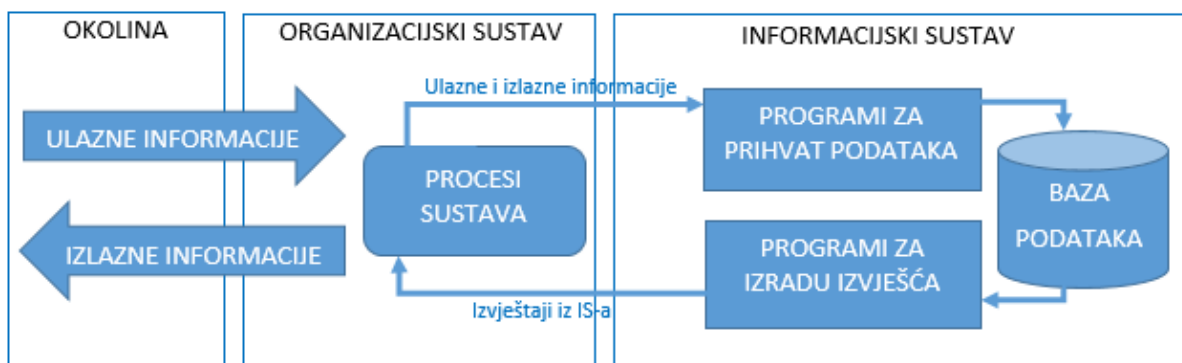
### 3.3. Ciljevi (temeljne funkcije) informacijskog sustava

Svaki informacijski sustav mora izvršavati sljedeće temeljne ciljeve (Jakupović, 2018):

- **Prikupljanje i upis podataka u bazu podataka:** Ostvaruje se programima za prihvati i upis podataka - nakon što program *prihvati* ulazne podatke, oni se *upisuju* u bazu podataka.
- **Obrada (procesiranje) podataka:** Ostvaruje se programima za automatizaciju procesa sustava (ili ljudskoga rada) - uzimaju se postojeći podaci iz baze podataka i novi podaci iz sustava te se njihovom kombinacijom generiraju podaci za bazu podataka sustava.
- **Prikaz i dostava podataka iz baze podataka:** Ostvaruje se programima za *izradu izvještaja* koji se zatim mogu prikazati na ekranu ili dostaviti u okolinu.
- **Čuvanje podataka:** Ostvaruje se sustavima za upravljanje bazama podataka (SUBP). Čuvanje podataka omogućuje stalnu dostupnost i sigurnost podataka.

### 3.4. Položaj informacijskog sustava u poslovnoj organizaciji

Organizacijski sustav procesima sustava prihvaća ulazne informacije iz okoline te ih uz pomoć programa za prihvati i upis podataka sprema u bazu podataka informacijskog sustava. Informacijski sustav uz pomoć svoje baze podataka i programa za kreiranje izvještaja dostavlja odgovarajuće izvještaje organizacijskom sustavu koji zatim dobivene izlazne informacije procesima sustava šalje u okolinu. To je vidljivo na slici 7:



Slika 7: Položaj informacijskog sustava u poslovnoj organizaciji (izvor: Autori)



Primjerice, prilikom upisa na neki fakultet student dostavlja svoje osnovne podatke (kao što su ime, prezime, adresa, datum rođenja, OIB itd.). Ti podaci predstavljaju *ulazne informacije*. Službenik/ica u Studentskoj referadi, kao *organizacijskom podsustavu* fakulteta, zaprima te podatke te ih korištenjem *programa za prihvatanje podataka* računalom podržanog informacijskog sustava upisuje u njegovu *bazu podataka*. Kada nadležne službe zatraže izvještaj o broju novoupisanih studenata na fakultet, službenik/ica dohvaća relevantne podatke iz *baze podataka* te se uz pomoć *programa za izradu izvješća* ona kreiraju i isporučuju nadležnima kao *izlazne informacije organizacijskog sustava*.

## 4. Projekt informacijskog sustava

**Projekt informacijskog sustava** je skup nacрта, sličan projektima, primjerice, u građevini ili brodogradnji, koji definira *sadržaj i strukturu, opseg i doseg informacijskog sustava* pomoću raznih konceptualnih i logičkih metoda modeliranja.

Prvi korak u projektiranju informacijskog sustava je **analiza**. Analiza predstavlja spoznaju *dijelova sustava* kao funkcionalne cjeline, tj. ciljeve (potrebe) koje budući informacijski sustav mora zadovoljiti. Rezultat analize su definirani elementi koje budući informacijski sustav treba sadržavati.

Nakon analize slijedi **oblikovanje (dizajn)** dijelova budućeg informacijskog sustava (npr. model podataka, model korisničkog sučelja) koji treba zadovoljiti potrebe ustanovljene tijekom analize. Rezultat dizajna su modeli na temelju kojih se može prijeći u sljedeću fazu projektiranja.

Analiza i oblikovanje su **logičke faze razvoja informacijskog sustava** u kojima se komunicira, promišlja i dokumentira informacijski sustav.

<b>Projektiranje informacijskog sustava</b> je izrada modela informacijskog sustava (Pavlič, <i>et al.</i> , 2014).
---

Projekt informacijskog sustava uključuje više modela, npr. model procesa, model podataka, model resursa.

**Model** je pojednostavljena reprezentacija (prikaz) dijela realnog sustava. Za isti sustav može postojati više modela koji iz različitih perspektiva opisuju IS-i. **Neke vrste modela:** *slikovni* (npr. u kemiji: slikovni prikaz molekule glukoze), *analogni/simulacijski* (npr. u medicini: model kostiju lubanje), *matematički* (npr. u matematici: eksponencijalni model rasta), *konceptualni/simbolički* (npr. u razvoju informacijskih sustava: EVA model podataka).

## 4.1. Modeli informacijskog sustava

**Modeli informacijskog sustava** strukturirano prikazuju te dostatno, potpuno i jasno opisuju ulaze i izlaze iz sustava, spremišta podataka te procese (postupke, procedure) za korištenje IS-a.

**Logički (konceptualni) model** je prikaz podataka i procesa koji se odvijaju u promatranom organizacijskom sustavu. Nije vezan uz način realizacije (ne odgovara na pitanje: kako?). Osnovica za definiranje logičkog modela IS-a je fizički model postojećeg IS-a (Kaluža, 2018).

**Fizički model** je opis postupaka koji se u sustavu odvijaju. Postupci se opisuju u obliku programa, procedure, servisa itd. U postupcima su implementirana poslovna (organizacijska) pravila i mogućnosti (uloge) korisnika.

## 4.2. Modeliranje informacijskog sustava

**Modeliranje** je proces kojim se izrađuje model.

**Modeliranje informacijskog sustava** je korištenje logičkih (konceptualnih) modela koji nisu ovisni o fizičkoj realizaciji, a razvijaju se pomoću koncepata definiranih u određenoj metodi. Takvi modeli sastoje se od različitih simbola koji predstavljaju konstrukte modela (elemente od kojih je izgrađen model) (Kaluža, 2018).

Modeliranjem se izrađuju različiti modeli budućeg IS-a kojima se utvrđuju svi koncepti potrebni za ispravno funkcioniranje tog IS-a. Drugim riječima, modelima se utvrđuju svi koncepti potrebni za izgradnju IS-a, a na temelju kojih će se u potpunosti zadovoljiti cilj, odnosno svrha organizacijskog sustava za koji se taj IS gradi. Navedeni koncepti su:

- *proces* promatranog sustava vezani uz pohranjivanje i rukovanje informacijama (podacima),
- *struktura podataka i informacija* poslovnog sustava koje informacijski sustav prikuplja, obrađuje, pohranjuje i distribuira,
- *granice informacijskog sustava* koje predstavljaju granice promatranog poslovnog sustava.

### 4.3. Vrste modela informacijskog sustava

**Osobine svakog modela informacijskog sustava** su sljedeće:

- nezavisan je o postojećem i budućem rješenju (samostalan je i ne ovisi o fizičkoj realizaciji informacijskog sustava),
- sveobuhvatan je (prikazuje sve elemente potrebne za funkcioniranje informacijskog sustava),
- jedinstven je (neredundantan je, tj. ne prikazuje iste elemente na više mjesta),
- fleksibilan je (ako se pokaže potreba, podložan je promjenama),
- razumljiv (izrađen je na korisnicima razumljiv, standardiziran način).

**Model procesa** prikazuje skup procesa koji prihvaćaju ulaze u sustav, izvode promjene u sustavu (promjena stanja sustava - ažuriranje podataka, kretanje podataka kroz sustav) i formiraju izlaze iz sustava.

**Model procesa** (Pavlič, *et al.*, 2014):

- prikazuje poslovna pravila unutar prikazanog IS-a;
- predstavlja skup aktivnosti (poslova) nad skupom podataka;
- prikazuje dekompoziciju poslovnih funkcija – dijagramom dekompozicije;
- opisuje obradu podataka – dijagramom toka podataka;
- prikazuje promjene unutar sustava.

**Svrha modeliranja procesa/funkcija** (modeliranje procesa - engl. *process modelling*) je:

- Prikaz kako se potrebni podaci prikupljaju, obrađuju i distribuiraju.
- Dekompozicija sustava (razlaganje složenih procesa na više jednostavnih) – najčešće je *top-down*.
- Opis obrade podataka – najčešće dijagramima tokova podataka (DTP) – engl. *data-flow diagram* (DFD).

**Model podataka** prikazuje objekte (entitete) iz promatranog poslovnog sustava o kojima je potrebno voditi evidenciju, osobine (atribute) kojima su ti objekti opisani te veze među prikazanim objektima.

**Model podataka** (Pavlič, *et al.*, 2014):

- prikazuje značenje promatranog IS-a;
- prikazuje povezanost dijelova promatranog IS-a;
- prikazuje zapise (podatke, informacije) unutar IS-a i njihove osobine;
- prikazuje stanje sustava;
- prirodni odnosi između podataka u objektivnoj stvarnosti stvaraju logičke veze i odnose između tipova podataka u modelu podataka;
- preduvjet je za stvaranje (izgradnju) baze podataka.

**Svrha modeliranja podataka** (modeliranje podataka – engl. *data modelling*) je kvalitativni opis podataka. To je opis podataka koji govori što su podaci, što opisuju i koje su njihove osobine. Takav opis ostvaruje se izradom:

- konceptualnog modela koji opisuje podatke (entitete) i veze među njima (model Entiteti-Veze-Atributi, tj. EVA model) i
- logičkog modela koji opisuje strukturu podataka i logičkih spremišta (Relacijski model, RM).

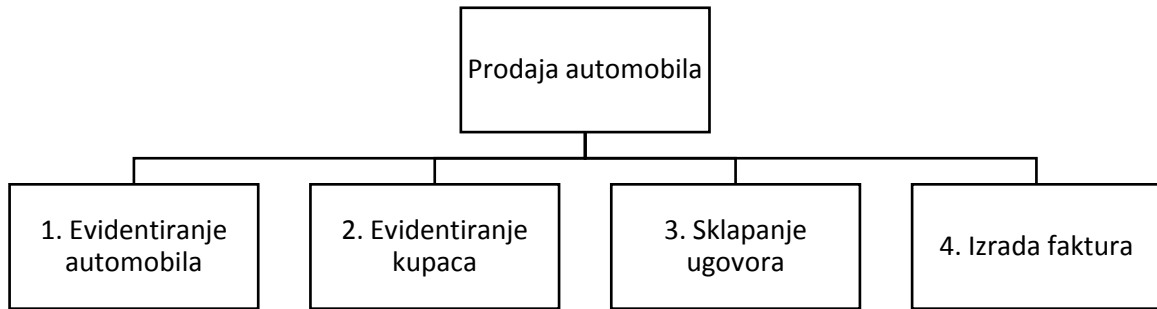
**Metoda za modeliranje podataka** je definirani postupak nalaženja i prikazivanja informacijskih objekata i njihovog međusobnog odnosa.

**Model resursa** je prikaz tehnološke, računalne osnove za izgradnju informacijskog sustava. Prikazuje: kadrove, organizacijsku strukturu za izgradnju informacijskog sustava te korištenu informacijsko-komunikacijsku tehnologiju (IKT), tj. opremu koja se koristi.

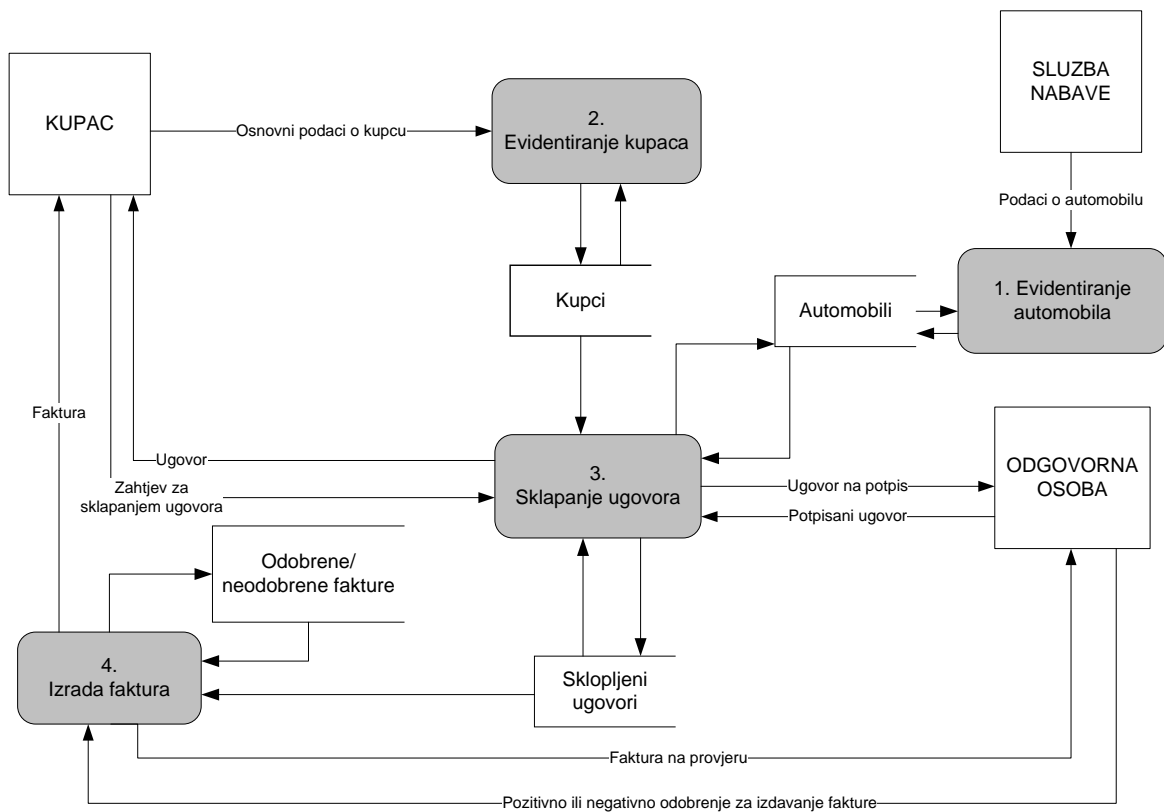
**Svrha modeliranja resursa je:** prikaz tko provodi obradu, tj. tko su korisnici IS-a (organizacijska struktura, ljudski resursi), prikaz računalne i mrežne infrastrukture IS-a.

**Redoslijed izrade modela informacijskog sustava je sljedeći<sup>1</sup>:**

**1. Model procesa, npr. (slike 8 i 9):**



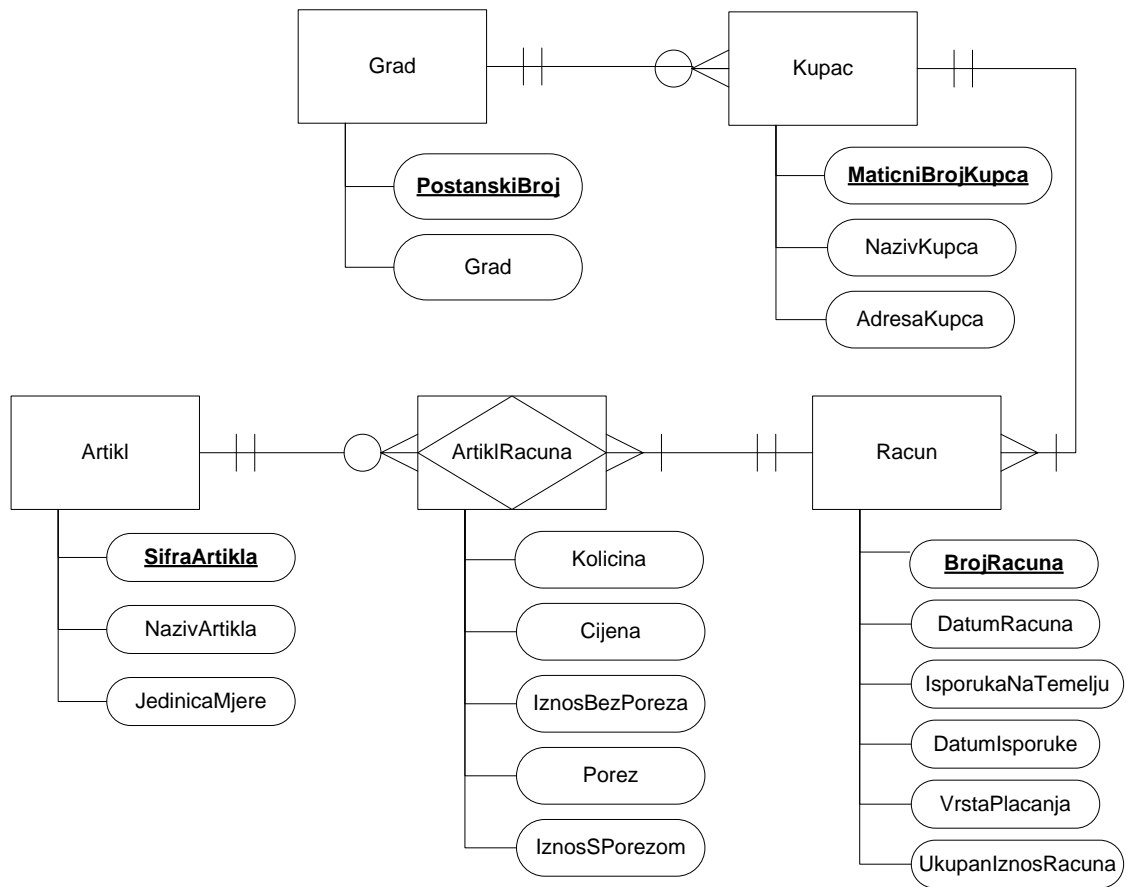
Slika 8: Dijagram dekompozicije procesa (izvor: Autori)



Slika 9: Primjer modela procesa: dijagram toka podataka prve razine dekompozicije (izvor: Autori)

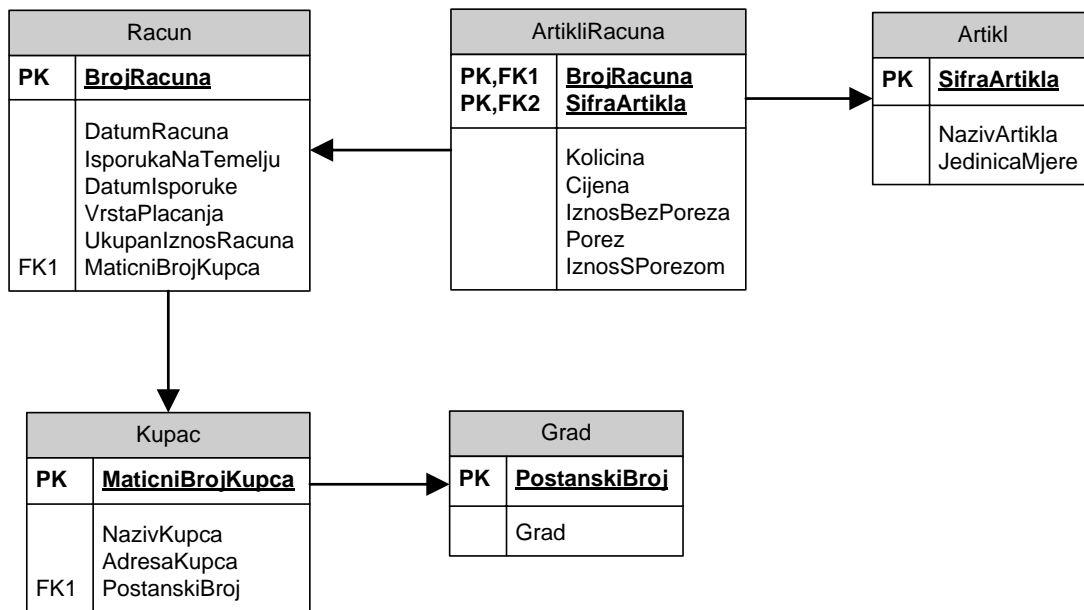
<sup>1</sup> Slike služe samo kao primjer modela, u nastavku skripte bit će detaljno objašnjeni postupci njihova modeliranja.

2. Konceptualni (EVA) model podataka, npr. (slika 10):



Slika 10: Primjer EVA modela podataka (izvor: Autori)

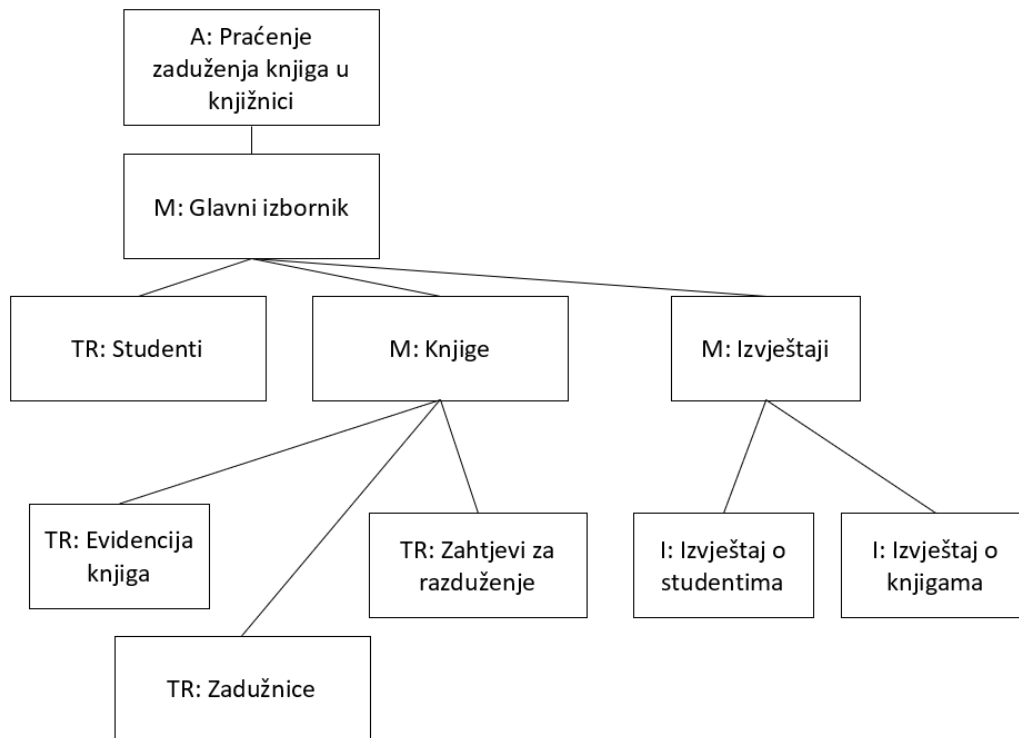
3. **Logički (relacijski) model podataka**, kao preduvjet izgradnje relacijske baze podataka, npr. (slika 11):



Slika 11: Primjer relacijskog modela podataka (izvor: Autori)

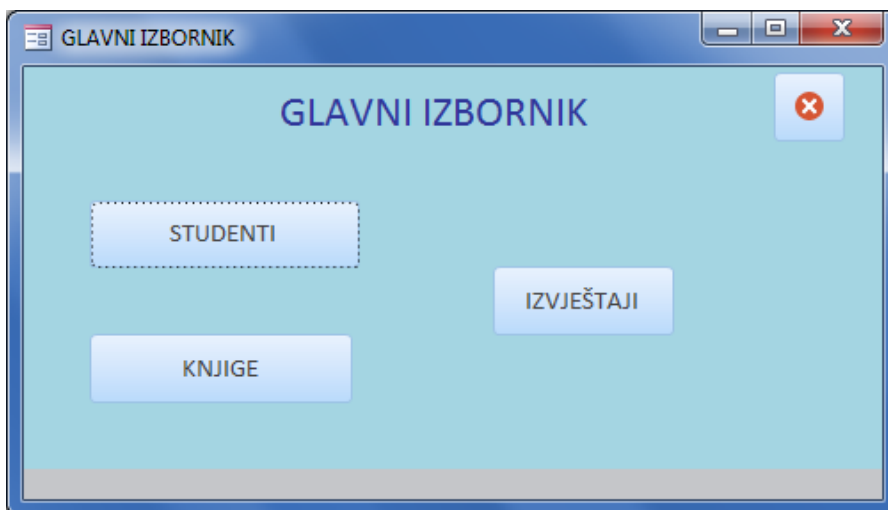


**4. Model arhitekture programskog proizvoda**, kao preduvjet izrade elemenata korisničkog sučelja u programskom rješenju u kojem će se koristiti izgrađena baza podataka, npr. (slika 12):



Slika 12: Primjer modela arhitekture programskog proizvoda (izvor: Autori)

Prototip dijela korisničkog sučelja izrađen na temelju modela arhitekture programskog proizvoda sa slike 12 prikazan je na slici 13. Vidljivo je sučelje Glavnog izbornika koji je na slici 12 prikazan modulom „M: Glavni izbornik“, a koji se grana na tri podmodula: „TR: Studenti“, „M: Knjige“ i „M: Izvještaji“.



Slika 13: Prototip dijela korisničkog sučelja (izvor: Autori)

Prototip dijela korisničkog sučelja izrađen na temelju modela arhitekture programskog proizvoda sa slike 12 prikazan je na slici 14. Vidljivo je sučelje obrasca za upis novih studenata u knjižnicu te pregled i izmjenu podataka postojećih studenata, koje je na slici 12 prikazano modulom „TR: Studenti“.

## STUDENTI

Zadužnica

Signatura knjige	Datum zaduženja	Šifra zahtjeva za knjigom
2	25.5.2015.	2
*		(Novi)

Zapis: 1 od 1 | Nema filtra | Pretraži

Slika 14: Prototip dijela korisničkog sučelja (izvor: Autori)

## 5. Model podataka

**Model podataka** je pojednostavljena reprezentacija nekog sustava koja opisuje skupove podataka bitne za funkcioniranje tog sustava, attribute koji opisuju te podatke i veze među podacima. Prikazuje stanje sustava.

Model podatka izrađuju dizajneri (projektanti informacijskih sustava), a potvrdu njegove ispravnosti daju krajnji korisnici. Rezultat je procesa modeliranja zasnovan na nekoj metodi.

<b>Model podataka</b> je preduvjet za izgradnju baze podataka informacijskog sustava.
---

**Baza podataka** je strukturirani (organizirani) skup međusobno povezanih podataka o svim pojavljivanjima entiteta, veza i atributa opisanih na modelima. To je organizirana zbirka podataka pohranjenih na određenom mjestu, npr. na računalu (Pavlič, 2011).

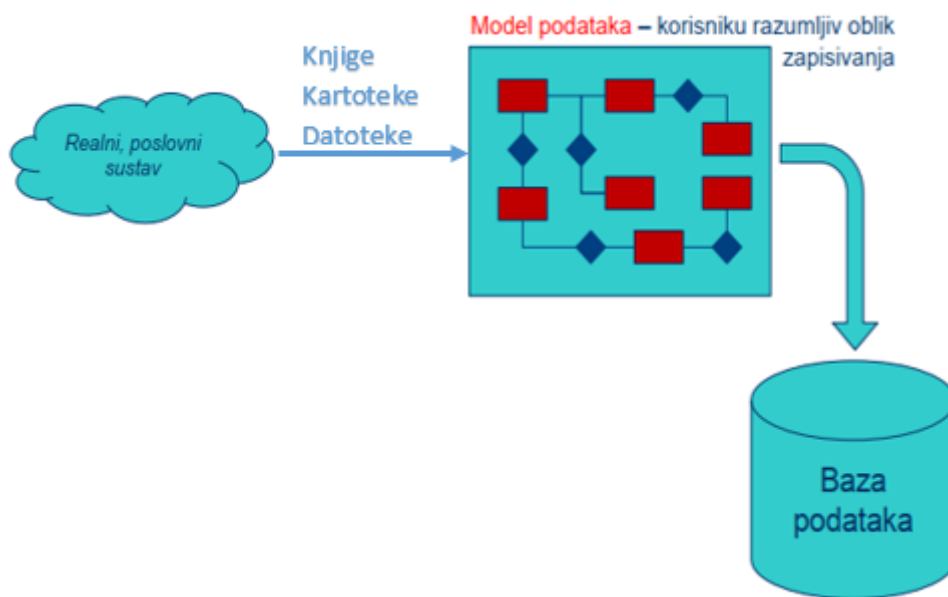
ili

**Baza podataka** je skup međusobno povezanih podataka, pohranjenih u vanjskoj memoriji računala. Podaci su istovremeno dostupni raznim korisnicima i aplikacijskim programima. Upisivanje, promjena, brisanje i čitanje podataka obavlja se posredstvom posebnog softvera, tzv. sustava za upravljanje bazom podataka – SUBP (Manger, 2014).

## 5.1. Modeliranje podataka

### 5.1.1. Položaj modela podataka

**Model podataka** izrađuje se na temelju informacija dobivenih iz realnog, poslovnog sustava, a preduvjet je za kreiranje buduće baze podataka informacijskog sustava za taj poslovni sustav (slika 15):



Slika 15: Položaj modela podataka (izvor: Autori)

Svaki poslovni (organizacijski) sustav kojem su za izvršavanje njegove funkcije potrebne informacije, ima potrebu za izradom odgovarajućeg informacijskog sustava s pripadajućom bazom podataka. Da bi se baza podataka izgradila, potrebno je iz poslovnih knjiga, kartoteka, datoteka itd. prikupiti sve relevantne informacije te identificirati njihove osobine i međusobne odnose. Na temelju prikupljenog izrađuje se model podataka na klijentu razumljiv način. Izrađeni model podataka je preduvjet izgradnje baze podataka za informacijski sustav promatranog poslovnog (organizacijskog) sustava (slika 15).

### 5.1.2. Struktura, ograničenja i operatori u modelu podataka

Svaki model podataka ima svoju strukturu, ograničenja i operatore (Pavlič, 2011):

**Struktura** modela podataka je prikazani skup svih tipova entiteta, njihovih kvalitativnih ili kvantitativnih osobina (atributa) te veza među tim tipovima entiteta<sup>2</sup>.

**Ograničenja** modela podataka omogućuju točnu interpretaciju podataka razdvajajući dopuštena od zabranjenih stanja skupa podataka preko: dopuštenih *podataka u okviru jednoga tipa entiteta*, dopuštenih *vrijednosti podataka svakog atributa tipova entiteta* i dopuštenih *povezivanja među tipovima entiteta*.

**Operatori**<sup>3</sup> omogućuju izmjenu stanja podataka u *bazi podataka*<sup>4</sup> u skladu s promjenom stanja u poslovnom sustavu.

### 5.1.3. Modeli podataka prema razini apstrakcije

Modeli podataka prema razini apstrakcije dijele se na konceptualne, logičke i fizičke modele (Šuman, et al., 2015):

**Konceptualni modeli podataka:** predstavljaju konceptualnu razinu apstrakcije. Opisuju entitete, njihove attribute i veze među entitetima (npr. EVA model, mrežni model, hijerarhijski model).

**Logički modeli podataka:** predstavljaju implementacijsku razinu apstrakcije. Opisuju slog (redak tablice, n-torku), polje (stupac tablice) i veze među redovima u različitim tablicama (npr. relacijski model, objektni model).

**Fizički modeli podataka:** predstavljaju fizičku razinu apstrakcije, tj. modeliranje fizičkog rasporeda podataka. Opisuju grupiranje slogova, fizička spremišta podataka, datoteke, bajtove itd. (npr. baza podataka, tekstualna datoteka, multimedijски sadržaj).

---

<sup>2</sup> Koncepti: *tip entiteta*, *atribut* i *veza* bit će objašnjeni u narednim poglavljima.

<sup>3</sup> Operatori će podrobnije biti objašnjeni u poglavlju 6.4 Relacijska algebra.

<sup>4</sup> Vrijednosti podataka ne mogu se bilježiti u *model podataka*, nego u repozitorij (bazu podataka) koji je strukturiran prema modelu podataka.

#### 5.1.4. Zašto izraditi model podataka?

Tijekom razvoja informacijskog sustava potrebno je izraditi adekvatan model podataka. Time će se spriječiti negativne **posljedice loše organizacije podataka** od kojih su neke npr:

- Troškovi i vrijeme razvoja informacijskog sustava premašuju planove ako ne postoji unaprijed izrađena dokumentacija s prikazanim modelom podataka na temelju kojeg će se izgraditi budući informacijski sustav.
- Bez dokumentiranog modela podataka informacijskog sustava, njegovo održavanje je mnogo složenije/zahtjevnije te postaje dominantan posao inženjera.
- Promjene u poslovnom sustavu se sporije i teže implementiraju ako ne postoji dokumentacija s modelom podataka na kojem je vidljiva organizacija njegova informacijskog sustava.
- Bez dokumentacije s modelom podataka, zbog loše organizacije podataka, može se dogoditi nekontrolirana redundancija podataka i problemi u njihovu ažuriranju.

#### 5.1.5. Identifikacija relevantnih objekata za modeliranje podataka iz opisa poslovnog sustava

**Identificirati relevantne objekte** u opisu poslovnog (organizacijskog) sustava znači odrediti koji su objekti iz opisa tog sustava bitni za njegovo funkcioniranje. Ti objekti će postati dio *modela podataka* kojim se prikazuje budući informacijski sustav. Drugim riječima, ovdje se određuje koje je informacije iz promatranog poslovnog sustava potrebno pohraniti u njegovu buduću bazu podataka. To su informacije koje su bitne za ispunjavanje funkcije (cilja) informacijskog sustava koji se razvija za određeni poslovni sustav.

Da bi se identificirali relevantni objekti iz opisa nekog poslovnog sustava potrebno je učiniti sljedeće:

1. Identificirati glavne pojmove (imenice) u opisu poslovnog sustava o kojima želimo voditi zapise (evidentirati podatke) – identificirati **entitete**.
2. Identificirati pojmove koji služe za opis glavnih pojmova (za prikaz njihovih kvalitativnih ili kvantitativnih osobina, svojstava) – identificirati **atribute entiteta**.

3. Identificirati kako su glavni pojmovi međusobno povezani (identificirati odnose (glagoli) među glavnim pojmovima) – identificirati **veze među entitetima**.

Nakon uspješne identifikacije svih relevantnih objekata promatranog poslovnog sustava kreće se s izradom modela podataka.

## 5.2. EVA model podataka

**Metodu Entiteti – Veze – Atributi (EVA metodu)** prvi put je objavio Chen 1976. godine (Chen, 1976). Dijagrami nacrtani uz pomoć ove metode sastoje se od:

- Entiteta;
- Atributa koji opisuju te entitete;
- Veza među entitetima.

**EVA model podataka** prikazuje *entitete* o kojima se želi voditi evidencija u informacijskom sustavu, *attribute* koji služe za opis osobina prikazanih entiteta te *veze* kojima se prikazuje u kakvim su odnosima promatrani entiteti.

Eva model podataka je simbolički, konceptualni opis skupa podataka. Dobro je prihvaćen u praksi jer su njegovi koncepti bliski korisniku. Osnova je za izgradnju *relacijskog modela* (prethodi fazi izgradnje relacijskog modela).

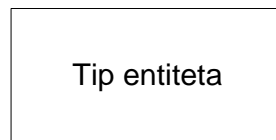
Postoje različite notacije koje se koriste za izradu EVA modela. To su, primjerice, *Chenova notacija* (Chen, 1976), *Martinova notacija* (Martin, 1993), *MIRIS notacija* (Pavlić, 2009).

U literaturi je moguće pronaći različite varijacije naziva *EVA modela podataka*, npr. *EV model* ili *ER model* (ER – engl. Entity Relationship).

### 5.2.1. Entitet kao koncept EVA modela podataka

**Entitet (Tip entiteta)** je onaj element tj. objekt u opisu poslovnog sustava koji je relevantan, bitan, koji se može jasno identificirati i za koji je potrebno čuvati (prikupljati, evidentirati) podatke. On se po svojim obilježjima razlučuje od okolice (Varga, 2020).

Kada se govori o entitetu, potrebno je razlikovati pojmove *tip entiteta* i *pojava entiteta*. **Tip entiteta** je imenovani skup entiteta istog tipa - skup entiteta iste vrste (npr. Student). **Entitet** je objekt koji pripada nekom tipu entiteta – to je jedna *pojava (instanca) tipa entiteta* (npr. Marko Babić, Lucija Horvat su entiteti (objekti, pojave) u tipu entiteta Student). Na slici modela podataka **entitet** se ne prikazuje. **Tip entiteta** prikazuje se na slici modela podataka. Simbol za tip entiteta na EVA modelu podataka vidljiv je na slici 16:



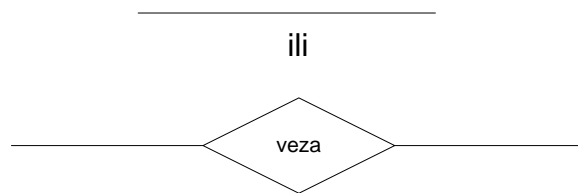
Slika 16: Simbol za tip entiteta na EVA modelu podataka (izvor: Autori)

### 5.2.2. Veza kao koncept EVA modela podataka

**Veza** na EVA modelu prikazuje u kakvom su odnosu (interakciji) promatrani entiteti (Varga, 2020).

Vezom se povezuju tipovi entiteta. Grafički prikaz veze ovisi o notaciji (sintaksi) koju koristimo prilikom izrade modela podataka, npr.: Martinova notacija, Chenova notacija, MIRIS notacija itd. Simboli za vezu između tipova entiteta na EVA modelu podataka u različitim notacijama prikazani su na slici 17:





Slika 17: Simboli za vezu na EVA modelu podataka (izvor: Autori)

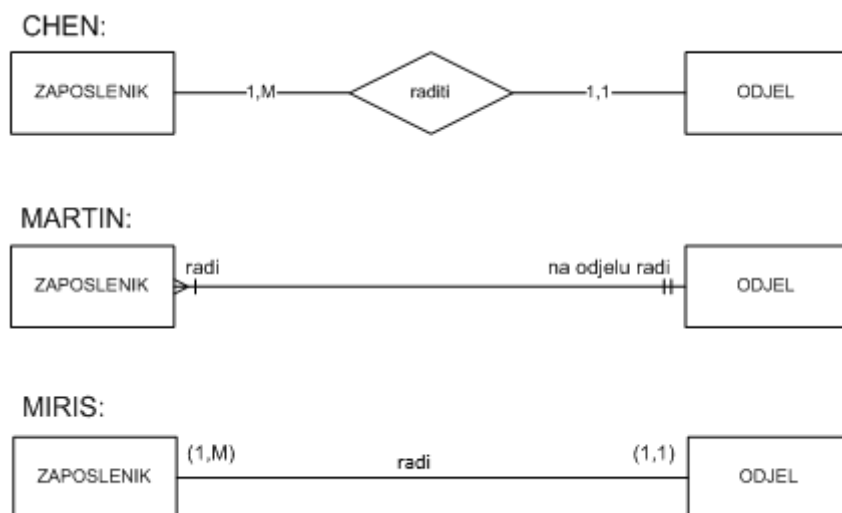
Kada se govori o vezama, potrebno je razumjeti sljedeće pojmove: *red veze*, *kardinalnost (brojnost) veze* i *opcionalnost veze* (Kaluža, 2018).

**Red veze** je broj tipova entiteta koji sudjeluje u jednoj vezi, tj. odnosu (unarna veza, binarna veza, ternarna veza, ..., n-arna veza<sup>5</sup>).

**Kardinalnost (brojnost) veze** – broj koji pokazuje koliko je pojava jednog tipa entiteta povezano s jednom pojavom drugog tipa entiteta.

**Opcionalnost veze** – način sudjelovanja u vezi. Postoje dvije vrste opcionalnosti veze. Prva je sudjelovanje *svih* pojava entiteta u vezi s *nekim* pojavama drugog entiteta -> *obavezno sudjelovanje*. Druga je sudjelovanje *samo nekih* pojava entiteta u vezi s *nekim* pojavama drugog entiteta -> *neobavezno (opcionalno) sudjelovanje*.

Primjeri grafičkih prikaza iste veze među tipovima entiteta Zaposlenik i Odjel, prema Chenu, Martin i MIRIS notaciji, prikazani su na slici 18.



Slika 18: Primjeri grafičkih prikaza veze prema različitim notacijama (izvor: Autori)

<sup>5</sup> Pojmovi će biti objašnjeni u poglavlju 6.2.2.1

Bez obzira koja notacija se koristi, veza je uvijek opisana *glagolom* koji opisuje odnos između dva povezana entiteta te *kardinalnošću* kojom se obostrano prikazuje koliko je pojava jednog tipa entiteta povezano s jednom pojavom drugog tipa entiteta.

Kardinalnost (brojnost) veze označava se različitim simbolima s obzirom na korištenu notaciju:

Martinova notacija:

○	niti jedan (nula)
	jedan
< ili >	više

Chenova i MIRIS notacija:

0	nula
1	jedan
M ili N	više

Udaljeniji od simbola tipa entiteta:

○ – opcionalno

| – obvezno

Bliži simbolu tipa entiteta:

| – jedan

< ili > - više

Lijeva vrijednost – minimalan broj povezanih entiteta:

0 – opcionalno

1 – obvezno

Desna vrijednost – maksimalan broj povezanih entiteta:

1 – jedan

M ili N - više

Kao primjer možemo pogledati sliku 19. Na slici su prikazana dva tipa entiteta - Zaposlenik i Odjel te veza među njima koja je tekstualno opisana u nastavku slike:



Slika 19: Primjer veze između dva entiteta na EVA modelu (izvor: Autori)

Opis veze prikazane na slici 19:

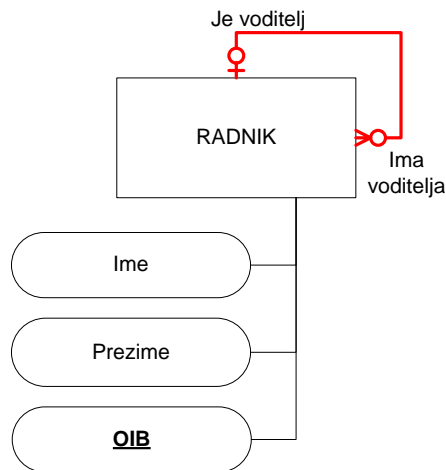
Zaposlenik<sup>(A1)</sup> **radi**<sup>(A2)</sup> na jednom i samo jednom<sup>(A3)</sup> Odjelu<sup>(A4)</sup> (Zaposlenik radi na obvezno jednom i na ne više od jednom Odjelu).

Na Odjelu<sup>(B1)</sup> **radi**<sup>(B2)</sup> jedan ili više<sup>(B3)</sup> Zaposlenika<sup>(B4)</sup> (Na odjelu radi obvezno jedan, a može i više Zaposlenika).

### 5.2.2.1. Vrste veza prema redu veza u EVA modelu podataka

Prema tome koliko tipova entiteta jedna veza međusobno povezuje (red veze), postoje: *unarna, binarna, ternarna, ..., višestruka (n-arna) veza*.

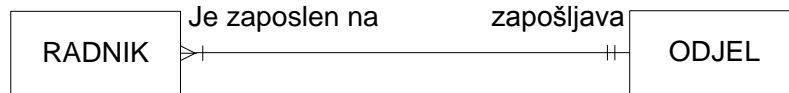
**Unarna (povratna) veza** je veza entiteta samog sa sobom. Npr. (slika 20):



Slika 20: Primjer unarne veze na EVA modelu (izvor: Autori)

Na slici 20 vidljiv je tip entiteta radnik kao dio EVA modela podataka neke male tvrtke koja ima jednog voditelja i njemu podređene radnike. Korištenjem unarne veze vidljivo je da radnik u promatranoj tvrtki može imati *niti jednog* (ako je sam voditelj) ili *jednog* voditelja. Isto tako, radnik može biti u funkciji voditelja *niti za jednog* (ako nije voditelj) ili *više* radnika.

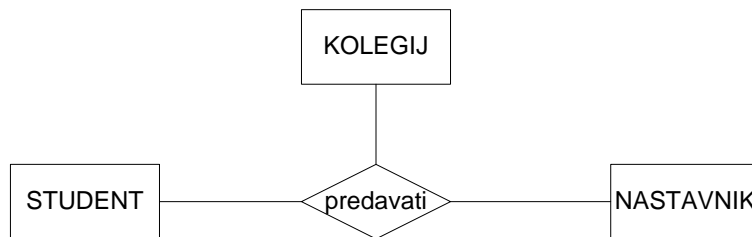
**Binarna veza** je jedna veza koja povezuje dva entiteta. Npr. (slika 21):



Slika 21: Primjer binarne veze na EVA modelu (izvor: Autori)

Na slici 21 vidljivi su tipovi entiteta radnik i odjel kao dijelovi EVA modela podataka neke tvrtke. Korištenjem binarne veze vidljivo je da je radnik zaposlen na *jednom i samo jednom* odjelu. S druge strane, na jednom odjelu može biti zaposlen *jedan ili više* radnika.

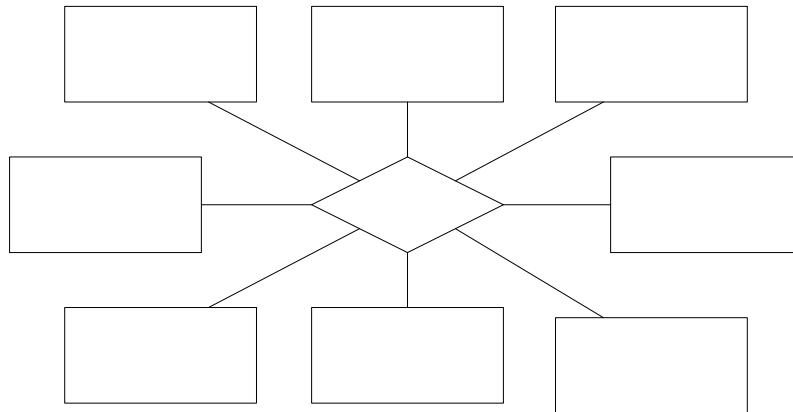
**Ternarna veza** je jedna veza koja povezuje tri entiteta. Npr. (slika 22):



Slika 22: Primjer ternarne veze na EVA modelu (izvor: Autori)

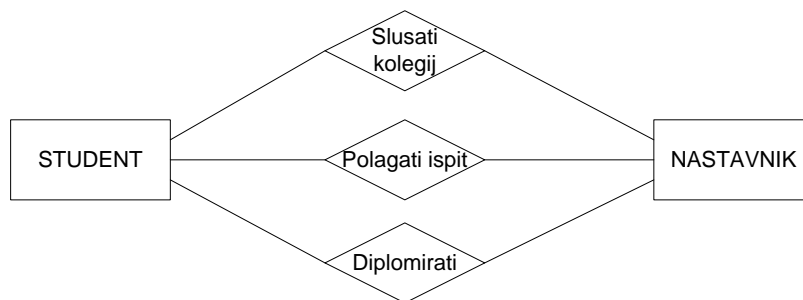
Na slici 22 vidljivi su tipovi entiteta student, nastavnik i kolegij kao dijelovi EVA modela nekog fakulteta. Korištenjem ternarne veze vidljivo je da tipovi entiteta student, nastavnik i kolegij mogu biti u jednom (istom) odnosu. Taj odnos je ovdje označen glagolom *predavati*. To znači da: studentu *predaje* nastavnik; nastavnik *predaje* kolegij; studentu *se predaje* kolegij.

**N-arna veza** je jedna veza koja povezuje više (od tri) entiteta. Općeniti prikaz n-arne veze vidljiv je na slici 23:



Slika 23: Primjer n-arne veze na EVA modelu (izvor: Autori)

**Višestruka binarna veza** moguća je između dvaju tipova entiteta kada se odnosi između tih entiteta ostvaruju zbog različitih razloga (pa se u opisu sustava koriste različiti glagoli). Npr. (slika 24):



Slika 24: Primjer višestruke veze na EVA modelu (izvor: Autori)

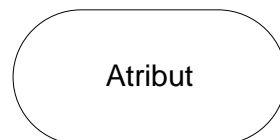
Na slici 24 vidljivi su tipovi entiteta student i nastavnik kao dijelovi EVA modela nekog fakulteta. Vidljivo je da ta dva entiteta mogu biti u više različitih odnosa. Prvo student *sluša kolegij* kod nastavnika. Zatim student *polaže ispit* kod nastavnika. Na kraju, ako se odluči, student može i *diplomirati* kod istog nastavnika.

### 5.2.3. Atribut kao koncept EVA modela podataka

**Atribut** je netko neko kvalitativno ili kvantitativno svojstvo entiteta tj. osobina koja opisuje entitet.

Svaki atribut može poprimiti jednu vrijednost iz domene vrijednosti atributa. Vrijednost atributa može se mijenjati tijekom vremena (Varga, 2020).

Simbol za atribut na EVA modelu podataka prikazan je na slici 25:



Slika 25: Simbol za atribut entiteta na EVA modelu podataka (izvor: Autori)

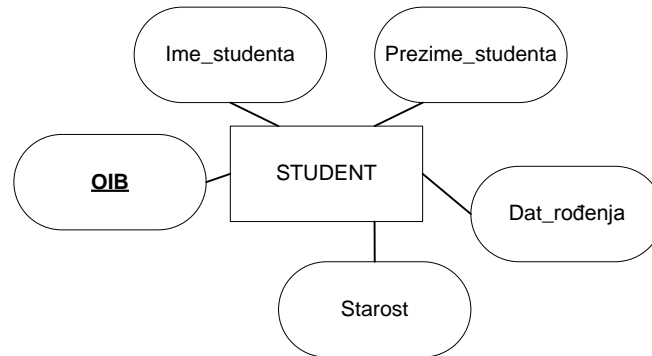
Prilikom određivanja atributa za promatrane entitete nekog modela podataka potrebno je držati se određenih *pravila*:

- Svaki entitet može imati proizvoljan broj atributa;
- Pojedini atribut može pripadati samo jednom entitetu;
- Atribut može imati samo jednu vrijednost za određenu pojavu entiteta u jednom trenutku;
- Atribut ima samo jedno značenje;
- Različite pojave entiteta mogu imati iste ili različite vrijednosti za isti atribut.

Svaki atribut promatranog entiteta ima **skup dozvoljenih vrijednosti** (domenu, tip) podataka koji smiju biti upisani u svrhu opisa tog entiteta u bazi podataka. Ako je atribut koji opisuje neku osobu, npr. „OIB”, onda je kao njegove vrijednosti dozvoljeno zapisivati samo numeričke podatke dužine 11 znakova.

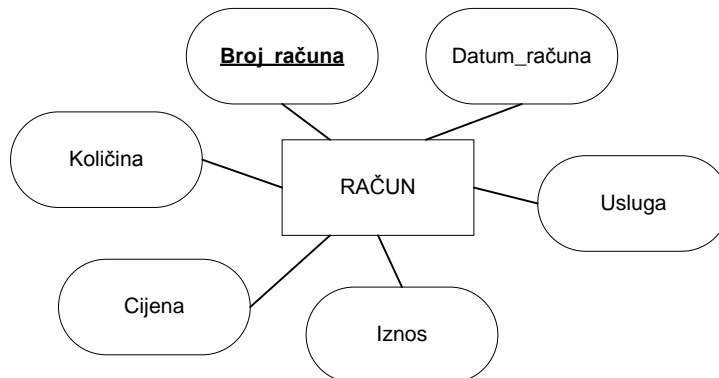
Vrijednosti atributa mogu biti i izvedene iz već postojećih vrijednosti. Takvi atributi se nazivaju izvedeni atributi. Postoje dvije vrste takvih atributa: interno izvedeni atributi i eksterno izvedeni atributi.

**Interno izvedeni atributi** su atributi čije se vrijednosti izvode iz već postojećih vrijednosti nekog drugog/drugih atributa istog tipa entiteta. Primjerice, u tipu entiteta „Student” podatak „Starost studenta” može se izvesti iz njegova već upisanog podatka „Datum rođenja”, tako da se od trenutnog datuma oduzme datum rođenja (slika 26).



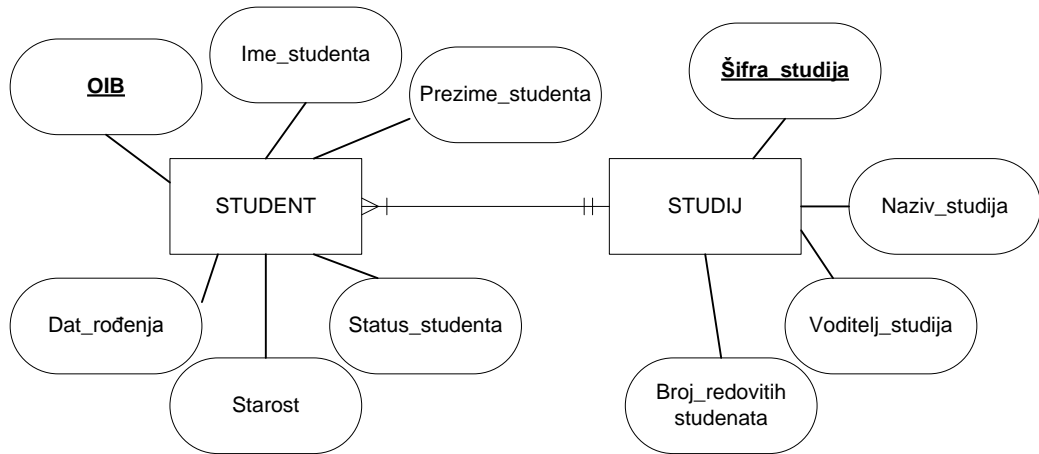
Slika 26: Tip entiteta Student (izvor: Autori)

Ili, primjerice, u tipu entiteta „Račun” podatak „Iznos” se može dobiti množenjem vrijednosti atributa „Količina” i „Cijena” (slika 27).



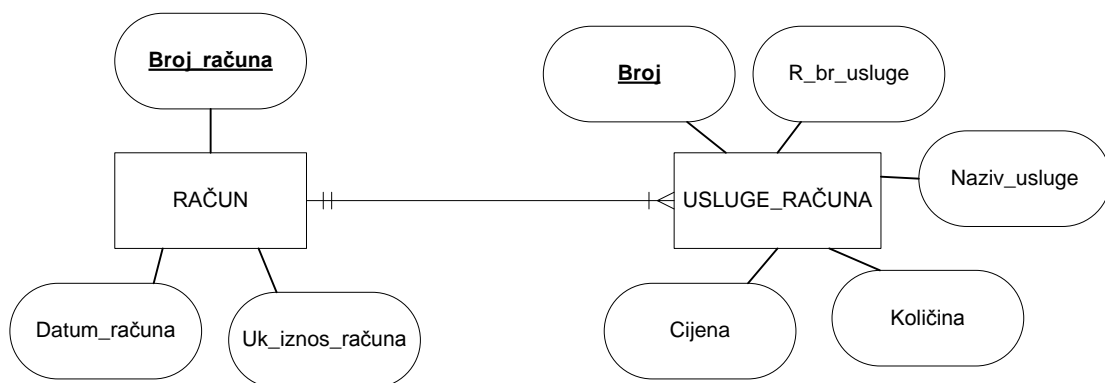
Slika 27: Tip entiteta Račun (izvor: Autori)

**Eksterno izvedeni atributi** su atributi čije se vrijednosti izvode iz već postojećih vrijednosti nekog drugog/drugih atributa nekog drugog tipa entiteta. Npr. atribut „Broj redovitih studenata” u tipu entiteta „Studij” može se izvesti na temelju prebrojavanja vrijednosti podataka atributa „Status studenta” iz tipa entiteta „Student” (slika 28).



Slika 28: Odnos tipova entiteta "Student" i "Studij" (izvor: Autori)

Ili npr. atribut „Ukupni iznos” u tipu entiteta „Račun” može se izvesti na temelju zbrajanja vrijednosti dobivenih množenjem vrijednosti podataka atributa „Količina” i „Cijena” iz tipa entiteta „Usluge računa” (slika 29).

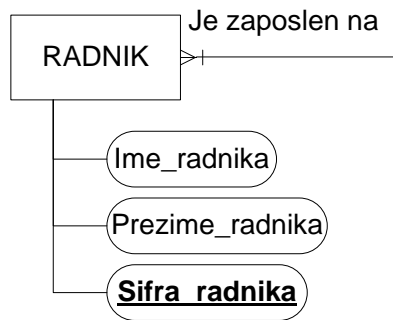


Slika 29: Odnos tipova entiteta "Račun" i "Usluge računa" (izvor: Autori)



#### 5.2.4. Identifikator na EVA modelu podataka

Da bi se svaki entitet mogao jedinstveno identificirati, tip entiteta mora sadržavati *identifikator*. **Identifikator** je atribut ili skup atributa koji *jedinstveno* opisuje određeni entitet, odnosno služi za jedinstvenu identifikaciju entiteta. U nekom opisu ili na modelu identifikator se dodatno naglašava, primjerice, *podcrtavanjem* (slika 30):



Slika 30: Primjer podcrtanog identifikatora na dijelu EVA modela (izvor: Autori)

Atribut koji je u svojstvu **identifikatora** zadovoljava sljedeća pravila:

- Jedinstven je, tj. jedinstveno identificira entitet. U tipu entiteta ne postoje dva različita entiteta koji imaju isti identifikator.
- Minimalan je – sastoji se od najmanjeg mogućeg broja atributa kojima se zadovoljava jedinstvenost;
- Obvezan je – prilikom pohranjivanja vrijednosti atributa nekog entiteta vrijednost identifikatora mora biti uvijek dostupna.
- Nepromjenjiv je – vrijednost identifikatora ne mijenja se tijekom vremena. Npr. atribut „Starost osobe“ ne može biti identifikator tipa entiteta „Osoba“ jer se s promjenom vremena mijenja i njegova vrijednost.

### 5.2.5. Zavisnost kao koncept EVA modela podataka

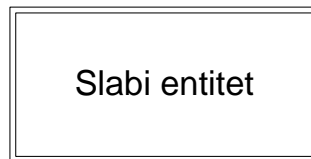
Osim *jakih (nezavisnih) entiteta* koji egzistiraju samostalno, postoje i *slabi entiteti*. Jaki tip entiteta egzistira samostalno, što znači da se entiteti mogu memorirati bez potrebe za postojanjem drugih entiteta.

**Slabi tip entiteta** je entitet koji na neki način ovisi o drugom entitetu. To znači da je u postupku memoriranja vrijednosti entiteta potrebno ostvarivanje veze s drugim entitetima (Pavlič, 2011).

Slabi entitet o drugom (jakom) entitetu može ovisiti:

- **Egzistencijalno** - entitet koji ne postoji samostalno, njegovo postojanje (egzistencija) ovisi o postojanju drugog entiteta;
- **Identifikacijski** - entitet koji se ne identificira samostalno, već uz pomoć nekog drugog entiteta.

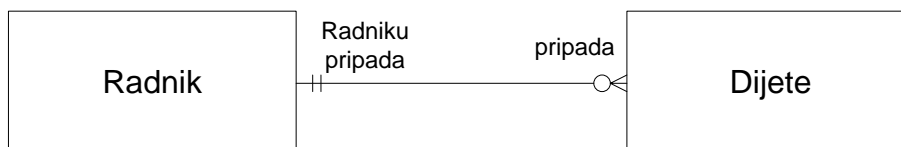
Simbol za identifikacijski zavisni tip entiteta na EVA modelu podataka vidljiv je na slici 31:



Slika 31: Simbol za identifikacijski zavisni tip entiteta na EVA modelu podataka (izvor: Autori)

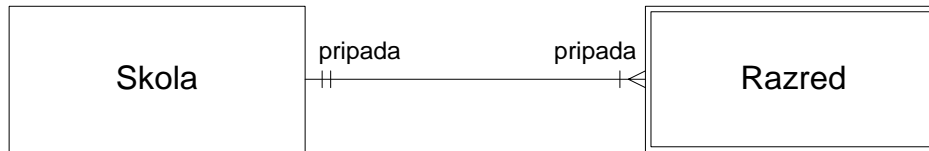
Primjeri slabih entiteta:

**Egzistencijalno zavisni entitet** (slika 32): na slici su vidljivi tipovi entiteta Radnik i Dijete kao dio EVA modela podataka neke tvrtke. Vrijednosti koje opisuju dijete ne mogu biti upisane u bazu podataka, ako vrijednosti koje opisuju njegova roditelja/staratelja, kao radnika tvrtke, nisu prije toga upisane u bazu podataka.



Slika 32: Primjer egzistencijalno slabog entiteta na EVA modelu (izvor: Autori)

**Identifikacijsko zavisni entitet** (slika 33): na slici su vidljivi tipovi entiteta Škola i Razred kao dio EVA modela nekog sustava koji npr. vodi evidenciju o svim osnovnim školama u nekom gradu. Tip entiteta Razred je egzistencijalno i identifikacijski slab i ovisan o tipu entiteta Škola. Vrijednosti koje opisuju razred ne mogu biti upisane u bazu podataka, niti se kasnije mogu identificirati, ako prije toga u bazu podataka nisu upisane vrijednosti koje opisuju školu.



Slika 33: Primjer egzistencijalno i identifikacijski slabog entiteta na EVA modelu (izvor: Autori)

*Identifikacijski zavisan entitet ujedno je i egzistencijalno zavisan. Egzistencijalno zavisan entitet ne mora biti i identifikacijski zavisan.*

Slabi entitet ima **složeni identifikator** koji se sastoji od njegova identifikatora i identifikatora entiteta o kojem ovisi. Na primjeru sa slike 32 može se pretpostaviti da je identifikator tipa entiteta Radnik npr. ID radnika, a da je identifikator tipa entiteta Dijete npr. ID djeteta. Budući da je Dijete slabi tip entiteta, on prema pravilu ima složeni identifikator koji se sastoji od atributa ID radnika i ID djeteta.

### 5.2.6. Agregirani tip entiteta kao koncept EVA modela

**Agregirani entitet (agregacija)** je specijalna vrsta slabog tipa entiteta oslabljena s dvije strane.

Nastaje u slučaju (Pavlič, 2011):

- kada neka veza ima atribute,
- kada su dva entiteta povezana vezom koja ima brojnost (kardinalnost) M:N (više : više) po gornjim granicama - takva se veza razdvaja na dvije 1:M veze po gornjim granicama, između kojih se postavlja agregirani entitet.

Agregacija *nema svoj identifikator*, već ga dobiva od susjednih entiteta s kojima je povezana M:1 vezom po gornjim granicama pa time agregacija ima **složeni identifikator**. Na EVA modelu se

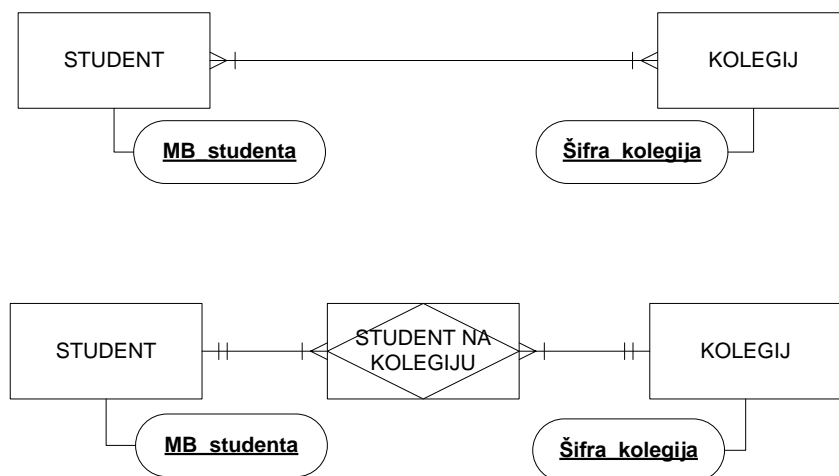
identifikatori agregiranog entiteta *ne prikazuju*, već se podrazumijevaju (poznajući pravila, mogu se pročitati iz modela, iako nisu prikazani).

Simbol za agregirani tip entiteta na EVA modelu podataka vidljiv je na slici 34:



Slika 34: Simbol za agregirani tip entiteta na EVA modelu podataka (izvor: Autori)

Slijedi primjer uvođenja agregiranog tipa entiteta između tipova entiteta čija je kardinalnost po gornjim granicama s obje strane M. Na slici 35 vidljiv je primjer tipova entiteta Student i Kolegij. Opis veze među njima bio bi sljedeći: *Jedan Student pohađa jedan ili više Kolegija. Jedan Kolegij pohađa jedan ili više Studenata.* Takva veza s brojnostima M:M razdvaja se u dvije 1:M veze uz pomoć agregacije *Student na kolegiju*. Tip entiteta *Student na kolegiju* kao identifikator preuzima identifikatore *Studenta* i *Kolegija*, dakle identifikatore onih tipova entiteta između kojih se nalazi, te dobiva složeni identifikator *MB studenta, Šifra kolegija*.



Slika 35: Primjer uvođenja agregacije na vezu M:M (izvor: Autori)

### 5.2.7. Dokumentacija modela podataka

Prilikom izgradnje modela podataka potrebno je provesti analizu, tj. opis promatranog poslovnog sustava (Kaluža, 2018) na način da se izradi:

1. **Kvalifikacija tipova entiteta** – potrebno je izraditi liste tipova entiteta (odrediti njihov naziv, izraditi njihov opis te sastaviti popis njihovih osnovnih atributa).
2. **Identifikacija veza** - nakon izrade popisa entiteta potrebno je identificirati međusobne veze u kojima se oni nalaze, kako bi se nakon toga mogao izraditi EVA model.
3. **Definiranje tipova entiteta** – osim popisa entiteta, potrebno je izraditi i detaljan opis svih entiteta.
4. **Identifikacija i opis atributa** – potrebno je izraditi listu (popis) atributa i provesti definiranje (opisivanje) atributa.
5. **Optimizacija modela** – na kraju je potrebno provesti prevođenje “neobičajenih” veza u „prevodljive” kako bi se EVA model mogao prevesti u Relacijski model podataka.

Osim samog modela podataka, kao dodatno pojašnjenje, uz njega se nalazi i dokumentacija. Svaki model podataka ima sljedeću dokumentaciju (Kaluža, 2018):

1. **Listu entiteta:** sastoji se od identifikacijskog (rednog) broja svakog entiteta; naziva entiteta (imenica u jednini, naziv mora biti jedinstven unutar promatranog modela); kratkog opisa svakog entiteta (objašnjenja značenja i uloge entiteta u promatranom poslovnom sustavu).
2. **Dijagram entiteta i veza (EVA dijagram):** grafički prikaz entiteta, njihovih atributa, veza među entitetima i brojnosti veza pomoću EVA metode.
3. **Listu i definiranje entiteta:** detaljan opis svakog entiteta; životni ciklus entiteta (uvjeti nastanka i uništavanja entiteta); iznimke (ograničenja) entiteta.
4. **Listu i definiranje atributa:** popis atributa za svaki entitet posebno - u naslovu liste atributa piše naziv entiteta ispod kojeg slijedi popis svih njegovih atributa (nazivi atributa), označen identifikator, opis svakog atributa, pravila za izračunavanje vrijednosti i primjeri vrijednosti.

### 5.3. Relacijski model podataka

**Relacijski model podataka** je *logički* model podataka. Izgrađuje se nakon *konceptualnog* EVA modela. Izgrađuje se prije *fizičkog* modela (baze podataka). Objavio ga je Codd 1970.

Osnovni pojmovi vezani uz relacijski model podataka:

**Relacijska metoda** je metoda za modeliranje podataka u skladu s pravilima organizacije relacijskih baza podataka (Pavlič, 2009).

**Relacijska baza podataka** je baza podataka koja se sastoji od međusobno povezanih tablica. Svaka tablica u bazi podataka predstavlja jednu relaciju iz relacijskog modela podataka.

**Relacijska shema** (shema relacije) je osnovni koncept strukture relacijskog modela podataka.

**Relacijska shema** se sastoji od *imena relacijske sheme* (imena relacije, tablice) i *popisa atributa* (prikazuju se u zagradama). Relacijska shema je konačan, neprazan skup atributa kojim se opisuje odgovarajuća relacija.

Relacijska shema (algebarski):

Ime\_relacije (atribut\_1, atribut\_2, ..., atribut\_n)

Relacijska shema (primjer):

STUDENT (Ime\_studenta, Prezime\_studenta, MB\_studenta)

Relacijska shema iz primjera služi za opis relacije (tablice u relacijskoj bazi podataka) STUDENT koja se sastoji od atributa (stupaca tablice u relacijskoj bazi podataka) Ime\_studenta, Prezime\_studenta i MB\_studenta.

Podaci unutar tablice (sadržaj baze podataka, redci relacije) prikazani su *slogovima*. **Slog** je jedna definirana vrijednost po svakom atributu relacijske sheme. To je redak stvarnih vrijednosti podataka u tablici baze podataka. Npr. u tablici s atributima: *Ime*, *Prezime*, *MB*, jedan slog (jedan entitet) je: *Marko, Horvat, 12345*. Slog služi za opis promatranog entiteta (objekta u poslovnom sustavu).

Skup istovrsnih slogova (s istim atributima) čini jednu **relaciju**. Svaka relacija (tablica) ima svoje *ime* po kojem je razlikujemo od ostalih u istoj bazi. Slijedi primjer relacije STUDENT s tri sloga (retka) u kojima su sadržane vrijednosti za attribute Ime\_stud, Prezime\_stud i MB\_stud (tablica 1):

Tablica 1: Primjer relacije s tri sloga (izvor: Autori)

STUDENT		
Ime_stud	Prezime_stud	MB_stud
Pero	Perić	12345
Marko	Markić	23456
Ana	Anić	34567

**Relacija** je skup slogova koji pripadaju istoj *relacijskoj shemi*. Uvijek je *konačan broj slogova* u relaciji. *Prazna relacija* je ona relacija koja nema niti jedan slog (tablica u kojoj nije zapisan niti jedan redak s vrijednostima atributa).

Da bi se svaki slog u relaciji mogao jedinstveno identificirati, minimalno jedan od atributa mora predstavljati *primarni ključ*.

**Primarni ključ** (engl. Primary key) je svojstvo ili skup svojstava koji *jedinstveno* opisuje određeni slog, odnosno služi za jedinstvenu identifikaciju sloga. U algebarskom ili grafičkom obliku, primarni ključ se dodatno naglašava npr. oznakom primarnog ključa - PK, podcrtavanjem ili ispisano naglašeno (bold) atributa koji čine primarni ključ.

**Primarni ključ** zadovoljava sljedeća pravila (Kaluža, 2018):

- Funkcijski određuje sve neključne attribute – za svaku pojavu vrijednosti ključnog atributa u svim slogovima u relaciji postoji uvijek ista vrijednost (jedna vrijednost) u svim neključnim atributima relacije;
- Jedinstven je – ne postoje dva pojavljivanja istih vrijednosti ključnih atributa u slogovima relacije;
- Minimalan je – sastoji se od najmanjeg mogućeg broja atributa kojima se ostvaruje jedinstvenost u relaciji;

**Vanjski ključ** relacije je atribut koji je u nekoj drugoj relaciji primarni ključ. *Vrijednost vanjskog ključa* u jednoj relaciji mora postojati kao *vrijednost primarnog ključa* u nekom slogu druge relacije, inače može biti NULL. Drugim riječima, ako neki skup atributa u promatranoj relaciji nije primarni ključ, ali je primarni ključ u nekoj drugoj relaciji u modelu, onda se on naziva *vanjski ključ* (Pavlič, 2011)

Povezivanje relacija u relacijskom modelu ostvaruje se dodavanjem dodatnog atributa u tablici koju se želi logički povezati s drugom. Pritom se dodanom atributu postavlja ograničenje dopuštenih vrijednosti na skup vrijednosti primarnog ključa u povezanoj tablici. Drugim riječima, u slogovima relacije na dodanim atributima koji čine vanjski ključ dopuštaju se samo vrijednosti koje postoje u nekom slogu u povezanoj relaciji u njezinom primarnom ključu.

### 5.3.1. Usporedba koncepata EVA i Relacijskog modela podataka

U tablici 2 prikazana je usporedba koncepata (pojmovi) koji se koriste za opis EVA modela podataka i Relacijskog modela podataka:

Tablica 2: Usporedba koncepata EVA i Relacijskog modela (izvor: Autori)

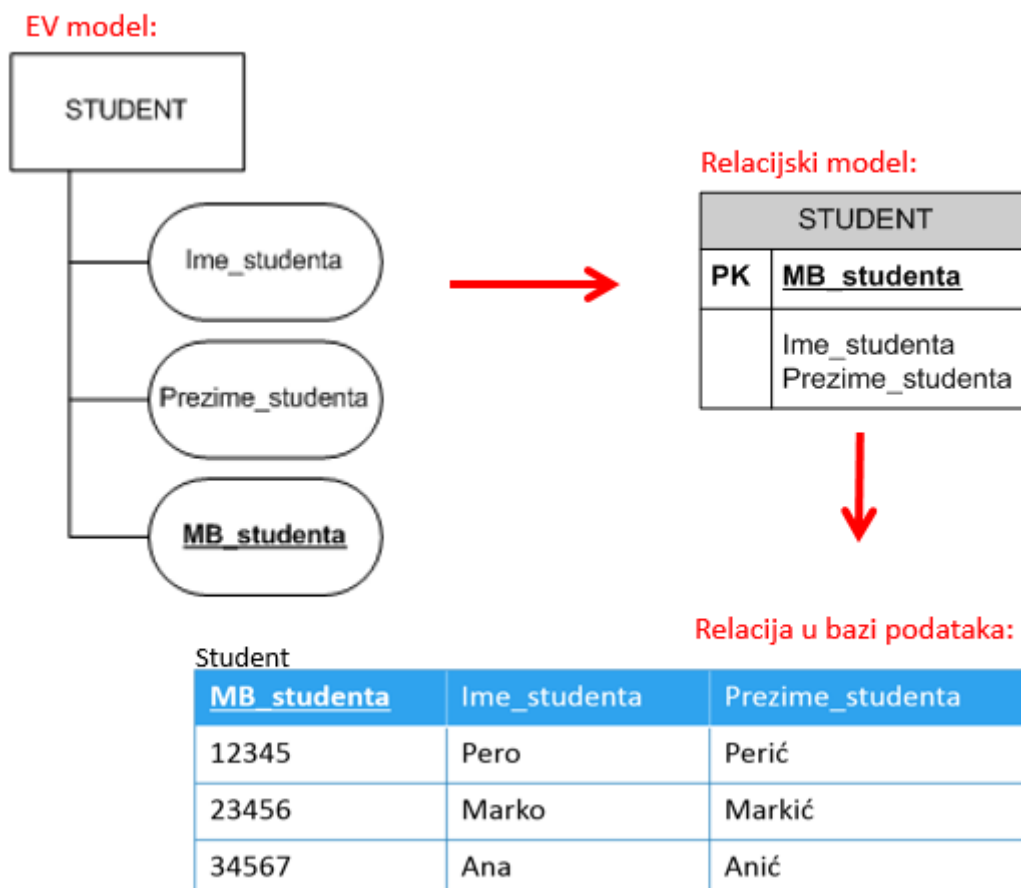
EVA model podataka	Relacijski model podataka
Tip entiteta	Relacija
Atributi tipa entiteta	Atributi relacije
Identifikator tipa entiteta	Primarni ključ relacije
Entitet	N-torka ili slog
Tip vrijednosti	Domena
Broj atributa u tipu entiteta	Stupanj relacije
Broj pojava u tipu entiteta	Brojnost relacije
Tip veze (M:1)	Vanjski ključ
Null vrijednost	Null vrijednost
Slabi tip entiteta	Relacija
Agregirani tip entiteta	Relacija



Iz tablice 2 vidljivo je da za sve koncepte EVA modela postoji odgovarajući koncept u Relacijskom modelu. Pritom veći broj koncepata EVA modela odgovara jednom konceptu u Relacijskom modelu. Za svaki koncept EVA modela postoji jedan koncept u Relacijskom modelu, pa se model EVA može transformirati u Relacijski model. S druge strane, za koncept relacija u Relacijskom modelu postoji veći broj koncepata u EVA modelu, pa se jednoznačna transformacija Relacijskog modela u EVA model ne može provesti.

### 5.3.2. Prevođenje EVA modela u Relacijski model podataka

Na slici 36 vidljiv je primjer rezultata prevođenja tipa entiteta *Student* s EVA modela u relaciju *Student* na Relacijskom modelu te prikaz prijedloga izgleda relacije *Student* u relacijskoj bazi podataka:

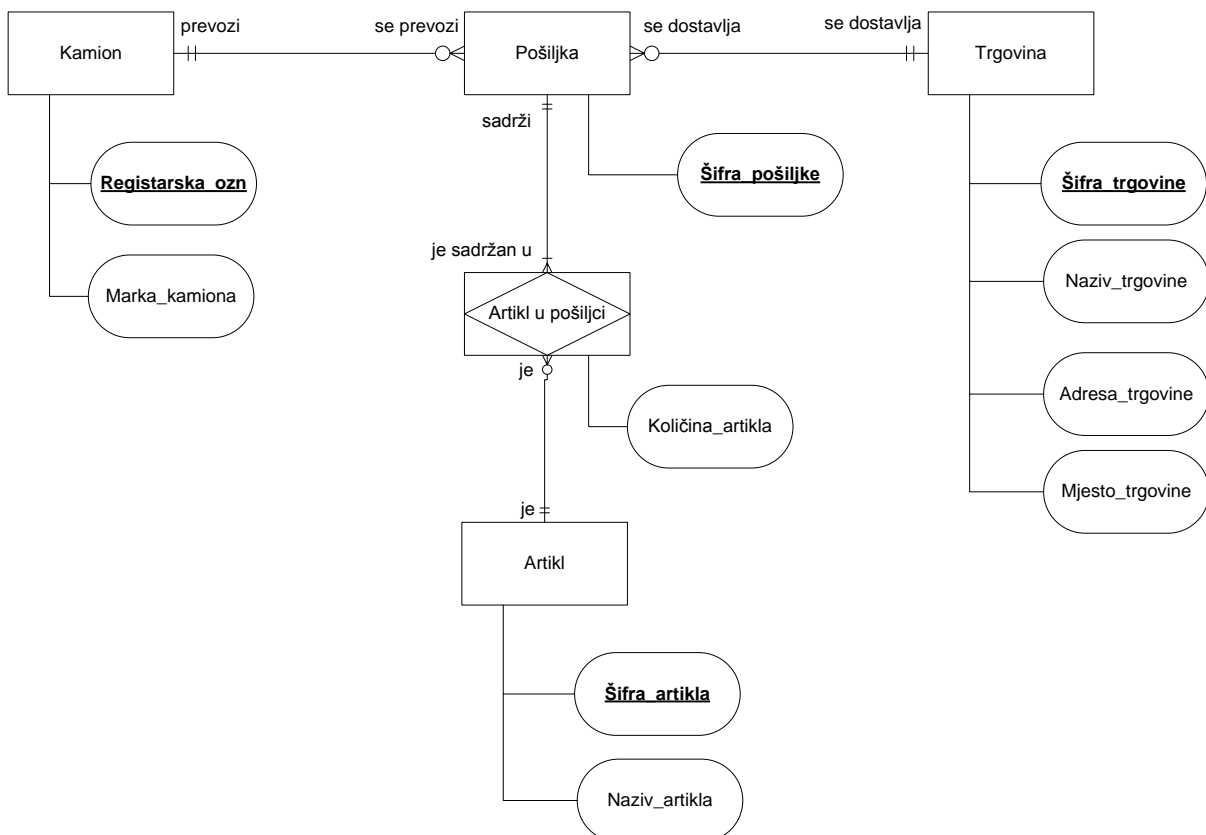


Slika 36: Od EVA modela do Relacijske baze podataka (izvor: Autori)

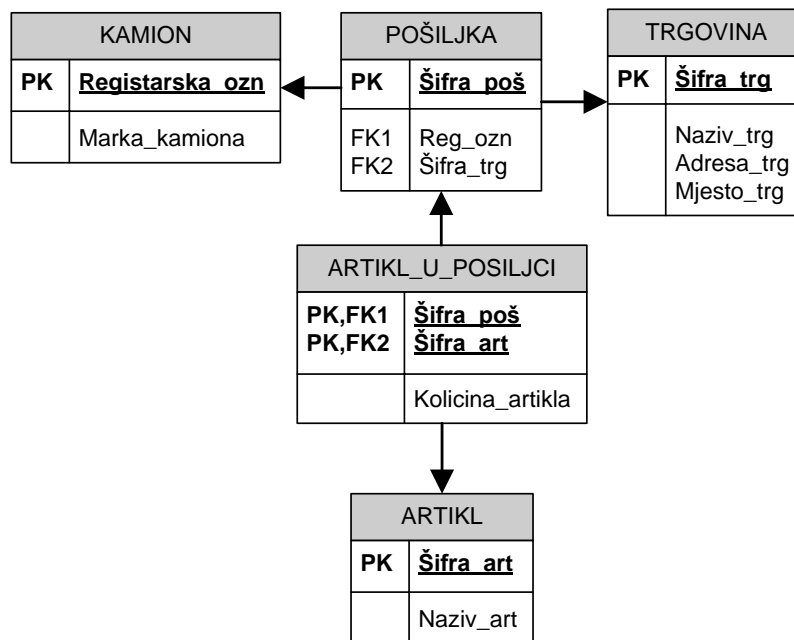
Tip entiteta *Student* na EVA modelu postaje relacija *Student* na relacijskom modelu. Atributi tipa entiteta *Student* na EVA modelu (*Ime\_studenta*, *Prezime\_studenta*, *MB\_studenta*) postaju atributi relacije *Student* na relacijskom modelu. Atribut u svojstvu identifikatora (*MB\_studenta*) na EVA modelu postaje primarni ključ na relacijskom modelu.

Temeljem relacije *Student* s relacijskog modela, gradi se relacija *Student* u relacijskoj bazi podataka. Atributi relacije *Student* s relacijskog modela postaju atributi (stupci) relacije *Student* u relacijskoj bazi podataka. Stupac *MB\_studenta* je dodatno označen kao primarni ključ relacije *Student* u relacijskoj bazi podataka. Redci (n-torke, slogovi) u relaciji *Student* (npr. *12345*, *Pero*, *Perić*) su stvarne pojave (vrijednosti) tipa entiteta *Student*.

Na slikama 37 i 38 prikazan je primjer EVA modela podataka prevedenog u Relacijski model podataka. Svi *tipovi entiteta* s EVA modela podataka postali su *relacije* na relacijskom modelu podataka. Svi *atributi tipova entiteta* s EVA modela podataka postali su *atributi relacija* na relacijskom modelu podataka. Atributi tipova entiteta koji su u svojstvu *identifikatora* na EVA modelu podataka postali su *primarni ključevi* relacija na relacijskom modelu podataka:



Slika 37: EVA model podataka podsustava distribucije pošiljki (izvor: Autori)



Slika 38: Relacijski model podataka podsustava distribucije pošiljki (izvor: Autori)

Osim gore navedenog, tijekom prevođenja EVA modela podataka u Relacijski model podataka, potrebno se držati određenih pravila (Pavlič, 2011):

**Pravilo 1:** Neki *tip entiteta* ne mora postati relacija ako je povezan samo s jednim drugim *tipom entiteta* i to vezom čije su brojnosti s obje strane 1:1 po gornjim granicama. Dakle, od dva tipa entiteta koja povezuje veza s brojnostima npr. (1,1):(1,1) u relaciju se prevodi samo jedan proizvoljno odabran tip entiteta. Toj se relaciji onda dodaju i atributi drugog tipa entiteta. Za ključ relacije može se proizvoljno odabrati bilo koji od identifikatora dva početna tipa entiteta.

Primjer pravila 1 (slika 39):

EVA model:



Slika 39: Entiteti povezani vezom 1:1 po gornjim granicama (izvor: Autori)

Relacijska shema dobivena prevođenjem EVA modela:

*STUDENT (Ime\_stud, Prezime\_stud, OIB\_stud, **MB\_indeksa** (PK))*

U prikazanom primjeru tipovi entiteta *Student* i *Indeks* prevode se u relaciju *Student* prikazanu relacijskom shemom *Student*. Svi atributi tipova entiteta *Student* i *Indeks* postaju atributi relacije *Student*. Kao primarni ključ relacije *Student* određen je identifikator *MB\_indeksa*.

**Pravilo 2:** Kod prevođenja agregiranog tipa entiteta u relaciju kao vanjski ključevi relacije upisuju se identifikatori tipova entiteta s kojima je agregirani tip entiteta povezan vezom čije su brojnosti M (više) po gornjoj granici. Ti vanjski ključevi relacije ujedno čine i njezin složeni primarni ključ.

Primjer pravila 2 (slika 40):

EVA model:



Slika 40: EVA model s agregiranim tipom entiteta

Relacijska shema agregiranog tipa entiteta dobivena prevođenjem s EVA modela:

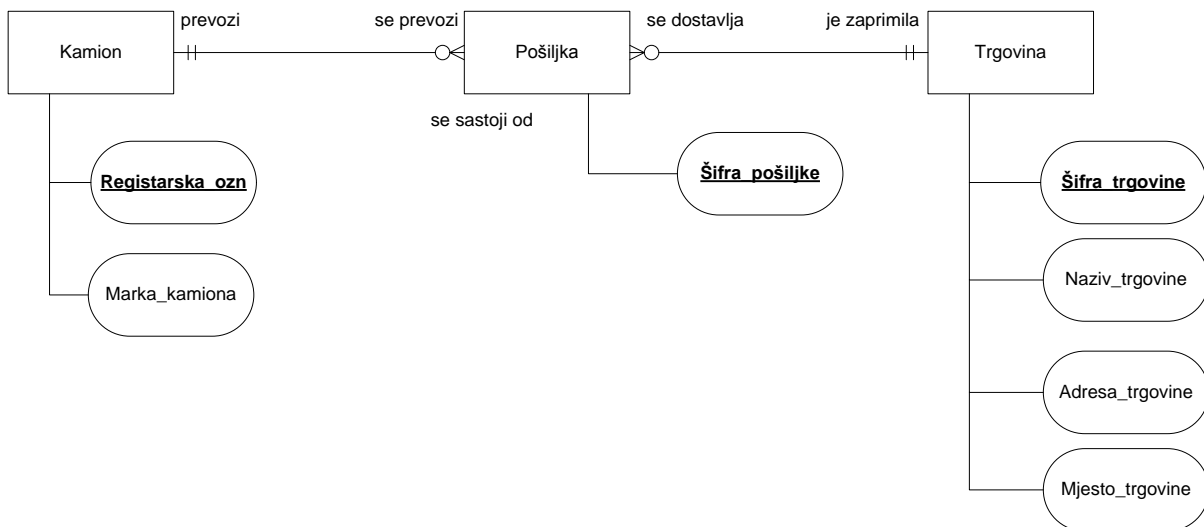
*Artikl\_u\_pošiljci* (Šifra pošiljke(PK,FK1), Šifra artikla (PK,FK2), Količina\_artikla)

U prikazanom primjeru agregirani tip entiteta *Artikl\_u\_pošiljci* se prevodi u relaciju / relacijsku shemu na način da, osim svog atributa *Količina\_artikla*, dobiva vanjski ključ *Šifra\_artikla* koji je identifikator u tipu entiteta *Artikal* te vanjski ključ *Šifra\_pošiljke* koji je identifikator u tipu entiteta *Pošiljka*. Ujedno, ta dva atributa zajedno čine složeni primarni ključ relacije *Artikl\_u\_pošiljci*.

**Pravilo 3:** Kada su dva tipa entiteta povezana vezom s brojnostima npr. (1,1):(0,M), tada se identifikator tipa entiteta koji se nalazi na strani (1,1) upisuje kao vanjski ključ u relaciju dobivenu prevođenjem tipa entiteta koji se nalazi na strani gdje je brojnost (0,M). Isto vrijedi i za sve ostale tipove entiteta koji imaju veze s brojnostima 1:M po gornjim granicama.

Primjer pravila 3 (slika 41):

EVA model:



Slika 41: EVA model s vezama 1:M po gornjim granicama (izvor: Autori)

Relacijska shema tipa entiteta Pošiljka:

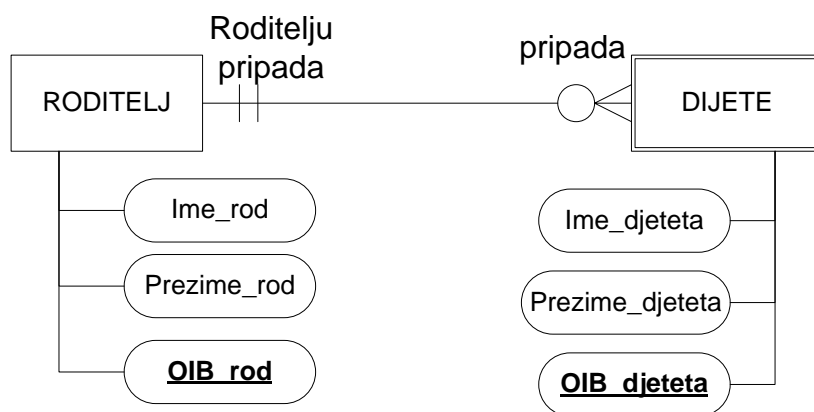
*Posiljka (Šifra\_posiljke(PK), Registarska\_ozn(FK1), Sifra\_trgovine(FK2))*

U prikazanom primjeru tip entiteta *Pošiljka* prevodi se u relaciju /relacijsku shemu *Pošiljka* koja osim primarnog ključa *Šifra\_posiljke*, prema pravilu 3 dobiva i dva vanjska ključa: *Registarska\_ozn* iz tipa entiteta *Kamion* i *Šifra\_trgovine* iz tipa entiteta *Trgovina*.

**Pravilo 4:** Slabi tip entiteta prevodi se u relaciju, no osim svojih atributa i identifikatora, sadrži i atribut koji je identifikator u tipu entiteta o kojem zavisi. Takav atribut ima dva ograničenja: postaje prvi član složenog primarnog ključa u relaciji, i postavlja se kao vanjski ključ prema relaciji o kojoj zavisi. Dakle, primarni ključ ovakve relacije sastoji se od identifikatora jakog entiteta i identifikatora slabog entiteta.

Primjer pravila 4 (slika 42):

EVA model:



Slika 42: EVA model sa slabim entitetom (izvor: Autori)

Relacijska shema slabog tipa entiteta dobivena prevođenjem EVA modela:

*Dijete* (*Ime\_djeteta*, *Prezime\_djeteta*, *OIB\_djeteta* (PK), *OIB\_rod* (PK, FK1))

U prikazanom primjeru, slabi tip entiteta *Dijete* prevodi se u relaciju / relacijsku shemu *Dijete* koja, osim svojeg primarnog ključa *OIB\_djeteta*, dobiva i primarni ključ *OIB\_rod* iz tipa entiteta *Roditelj*. Ta dva atributa zajedno čine složeni primarni ključ relacije *Dijete*. *OIB\_rod* je ujedno i vanjski ključ u relaciji *Dijete*.

**Pravilo 5:** Unarna veza prevodi se tako da se prepozna o kakvoj brojnosti veze entiteta samog sa sobom se radi (1:1, 1:M ili M:N) nakon čega se prevođenje provodi kao kod binarnih veza.

**Pravilo 6:** Ternarna veza prevodi se na način da se svaki entitet prikazuje zasebnom relacijom. Za povezivanje se uvodi nova relacija koja sadrži identifikatore sva tri entiteta u obliku vanjskih ključeva. Primarni ključ nove relacije može biti složen od sva tri vanjska ključa ili se može uvesti generički primarni ključ (npr. ID), pri čemu sva tri vanjska ključa zajedno čine složeni sekundarni ključ.

#### 5.4. Relacijska algebra

**Relacijska algebra** je skup formalnih operacija na relacijama relacijske baze podataka. Izvodi se pomoću poznatih operatora na način da operatori djeluju nad jednom ili više relacijskih shema i pripadnim relacijama (Kaluža, 2008).

Relacijska algebra (engl. relational algebra) je jezik koji omogućava definiranje novih relacija iz već postojećih. Nova relacija sadrži podatke koji su bili u bazi podataka u trenutku izvršenja operacije (Pavlič, 2011).

Korištenjem operatora stvara se nova relacijska shema s novom pripadnom relacijom (skupom slogova). Rezultat relacijskih algebarskih operacija je *relacija* (Codd, 1970).

Relacijski operatori koriste se kada želimo dobiti neku novu tablicu (relaciju) na temelju postojećih tablica i njihovih slogova.

Prema izvoru nastanka relacijski operatori se mogu podijeliti u tri kategorije. To su *naslijeđeni*, *osnovni* i *izvedeni operatori* (Kaluža, 2008):

- **Naslijeđeni relacijski operatori** - naslijeđeni su iz matematičke algebre: *unija* ( $\cup$ ), *presjek* ( $\cap$ ), *razlika* ( $-$ ).
- **Osnovni relacijski operatori** - definirani su ovdje u relacijskoj teoriji: *projekcija* ( $\Pi$ ), *selekcija* ( $\sigma_F$ ), *prirodni spoj* ( $\infty$ ), *preimenovanje* ( $\delta$ ).
- **Izvedeni relacijski operatori** - mogu se izvesti kombinacijom naslijeđenih i osnovnih operatora: *produkt* ( $\otimes$ ), *aktivni komplement* (AC), *kvocijent* ( $\div$ ).



## 5.4.1. Naslijeđeni relacijski operatori

### 5.4.1.1. Relacijski operator *unija*

Relacijski operator *unija* je *binaran operator*. To znači da djeluje između najmanje *dva* operanda (u našem slučaju tablice). Koristi se između dvije (ili više) zadanih relacija koje imaju SVE iste atribute - imaju jednake relacijske sheme. Operator je komutativan, pa redoslijed relacija nije bitan.

**Unija ( $r \cup s$ )** je skup svih slogova koji su u relaciji **r** ili u relaciji **s**.

Primjer unije dvaju relacija (Slika 43):

KUPAC				DOBAVLJAČ			
OIB	Naziv	Adresa	PostBr	OIB	Naziv	Adresa	PostBr
12345678912	Tvrtka1	Ivana Lučića 4	10000	23456789123	Tvrtka2	Vukovarska 55	51000
23456789123	Tvrtka2	Vukovarska 55	51000	56789456321	Tvrtka4	Forum 5	52100
34567891234	Tvrtka3	Cvečićev uspon	52100	65410236987	Tvrtka5	Sv. Nedjelje	23000

#### Kupac $\cup$ Dobavljač

PARTNER			
OIB	Naziv	Adresa	PostBr
12345678912	Tvrtka1	Ivana Lučića 4	10000
23456789123	Tvrtka2	Vukovarska 55	51000
34567891234	Tvrtka3	Cvečićev uspon	52100
56789456321	Tvrtka4	Forum 5	52100
65410236987	Tvrtka5	Sv. Nedjelje	23000

Slika 43: Primjer unije (izvor: Autori)

U prikazanom primjeru relacije *Kupac* i *Dobavljač* imaju jednake relacijske sheme s atributima *OIB*, *Naziv*, *Adresa*, *PostBr*. Njihovu uniju čine *svi slogovi* (redci s vrijednostima) *obaju relacija* (*Tvrtka 1*, *Tvrtka 2*, *Tvrtka 3*, *Tvrtka 4*, *Tvrtka 5* pri čemu se *Tvrtka 1* i *Tvrtka 3* nalaze samo u relaciji *Kupac*, *Tvrtka 4* i *Tvrtka 5* samo u relaciji *Dobavljač*, a *Tvrtka 2* se nalazi i u relaciji *Kupac* i u relaciji *Dobavljač*).

### 5.4.1.2. Relacijski operator *presjek*

Relacijski operator *presjek* je *binaran operator*. To znači da djeluje između najmanje *dva* operanda (u našem slučaju tablice). Koristi se između dvije (ili više) zadanih relacija koje imaju SVE iste attribute - imaju jednake relacijske sheme. Operator je komutativan pa redoslijed relacija nije bitan.

**Presjek ( $r \cap s$ )** je skup svih slogova koji su u relaciji  $r$  i u relaciji  $s$ .

Primjer presjeka dvaju tablica (Slika 44):

KUPAC				DOBAVLJAČ			
OIB	Naziv	Adresa	PostBr	OIB	Naziv	Adresa	PostBr
12345678912	Tvrtka1	Ivana Lučića 4	10000	23456789123	Tvrtka2	Vukovarska 55	51000
23456789123	Tvrtka2	Vukovarska 55	51000	56789456321	Tvrtka4	Forum 5	52100
34567891234	Tvrtka3	Cvečićev uspon	52100	65410236987	Tvrtka5	Sv. Nedjelje	23000

#### Kupac $\cap$ Dobavljač

PARTNER – samo oni koji su Kupac i Dobavljač			
OIB	Naziv	Adresa	PostBr
23456789123	Tvrtka2	Vukovarska 55	51000

Slika 44: Primjer presjeka (izvor: Autori)

U prikazanom primjeru relacije *Kupac* i *Dobavljač* imaju jednake relacijske sheme s atributima *OIB*, *Naziv*, *Adresa*, *PostBr*. Njihov presjek čine slogovi (redci s vrijednostima) čije se vrijednosti nalaze u *obje* relacije (*Tvrtka 2*).

### 5.4.1.3. Relacijski operator *razlika*

Relacijski operator *razlika* je *binaran operator*. To znači da djeluje između *dva* operanda (u našem slučaju tablice). Koristi se između dvije zadanih relacija koje imaju SVE iste attribute - imaju jednake relacijske sheme. Operator nije komutativan, pa je redoslijed relacija bitan.

**Razlika ( $r - s$ )** je skup svih slogova koji su u relaciji  $r$  nisu u relaciji  $s$  – bitan je redoslijed relacija jer prikazuje redove koji jesu u prvoj, a nisu u drugoj relaciji.

Primjer razlike dvaju tablica (Slika 45):

KUPAC				DOBAVLJAČ			
OIB	Naziv	Adresa	PostBr	OIB	Naziv	Adresa	PostBr
12345678912	Tvrtka1	Ivana Lučića 4	10000	23456789123	Tvrtka2	Vukovarska 55	51000
23456789123	Tvrtka2	Vukovarska 55	51000	56789456321	Tvrtka4	Forum 5	52100
34567891234	Tvrtka3	Cvečićev uspon	52100	65410236987	Tvrtka5	Sv. Nedjelje	23000

Kupac — Dobavljač

PARTNERI			
OIB	Naziv	Adresa	PostBr
12345678912	Tvrtka1	Ivana Lučića 4	10000
34567891234	Tvrtka3	Cvečićev uspon	52100

Dobavljač -- Kupac

PARTNERI			
OIB	Naziv	Adresa	PostBr
56789456321	Tvrtka4	Forum 5	52100
65410236987	Tvrtka5	Sv. Nedjelje	23000

Slika 45: Primjer razlike (izvor: Autori)

U prikazanom primjeru relacije *Kupac* i *Dobavljač* imaju jednake relacijske sheme s atributima *OIB*, *Naziv*, *Adresa*, *PostBr*. Ako se provodi razlika relacija *Kupac – Dobavljač*, rezultat su samo oni slogovi (redci s vrijednostima) koji se nalaze u relaciji *Kupac*, a ne nalaze se u relaciji *Dobavljač* (*Tvrtka 1* i *Tvrtka 3*). Ako se provodi razlika relacija *Dobavljač – Kupac*, rezultat su samo oni slogovi (redci s vrijednostima) koji se nalaze u relaciji *Dobavljač*, a ne nalaze se u relaciji *Kupac* (*Tvrtka 4* i *Tvrtka 5*).

## 5.4.2. Osnovni relacijski operatori

### 5.4.2.1. Relacijski operator *projekcija*

Relacijski operator **projekcija** je *unaran operator*. To znači da se izvodi nad jednom relacijskom shemom i pripadnom relacijom (tablicom).

**Projekcija ( $\Pi$ )** prikazuje **željeni skup atributa** (stupaca relacije) izdvojen iz relacijske sheme. Pri tome prikazuje sve slogove (retke) iz te relacije.

Primjer projekcije iz jedne tablice (Slika 46):

KUPAC			
OIB	Naziv	Adresa	PostBr
12345678912	Tvrtka1	Ivana Lučića 4	10000
23456789123	Tvrtka2	Vukovarska 55	51000
34567891234	Tvrtka3	Cvečićev uspon	52100

Iz relacije KUPAC prikazati OIB i Naziv:

KUPAC	
OIB	Naziv
12345678912	Tvrtka1
23456789123	Tvrtka2
34567891234	Tvrtka3

Slika 46: primjer projekcije (izvor: Autori)

U prikazanom primjeru potrebno je prikazati attribute *OIB* i *Naziv* iz relacije *Kupac*. Projekcija prikazuje željeni skup atributa (*OIB* i *Naziv*) sa svim slogovima (redcima s vrijednostima za *Tvrtku 1*, *Tvrtku 2* i *Tvrtku 3*).

#### 5.4.2.2. Relacijski operator *selekcija*

Relacijski operator **selekcija** je *unaran operator*. To znači da se izvodi nad jednom relacijskom shemom i pripadnom relacijom (tablicom).

**Selekcija ( $\sigma_r$ )** prikazuje sve slogove relacije koji zadovoljavaju zadani **uvjet** (obično na temelju određenih vrijednosti pojedinih atributa). Drugim riječima, selekcija izdvaja željene retke iz tablice (one retke koji zadovoljavaju određeni uvjet).

Primjer selekcije iz jedne tablice (Slika 47):

KUPAC			
OIB	Naziv	Adresa	PostBr
12345678912	Tvrtka1	Ivana Lučića 4	10000
23456789123	Tvrtka2	Vukovarska 55	51000
34567891234	Tvrtka3	Cvečićev uspon	52100

Iz relacije KUPAC prikazati samo one koji su iz Rijeke:

KUPAC			
OIB	Naziv	Adresa	PostBr
23456789123	Tvrtka2	Vukovarska 55	51000

Slika 47: Primjer selekcije (izvor: Autori)

U prikazanom primjeru potrebno je prikazati samo one tvrtke čije je sjedište u gradu s poštanskim brojem čija je vrijednost *51000*. Selekcija prikazuje sve atribute relacije *Kupac* (OIB, Naziv, Adresa, PostBr), ali samo one slogove (retke s vrijednostima) koji zadovoljavaju uvjet da im je vrijednost atributa *PostBr* jednaka *51000* (*Tvrtka 2*).

### 5.4.2.3. Relacijski operator *prirodni spoj*

Relacijski operator *prirodni spoj* je *binaran operator*. To znači da djeluje između dvije relacije, tj. ima najmanje *dva* operanda (relacije - tablice). Koristi se između dvije zadane relacije koje imaju različite atribute - imaju različite relacijske sheme. Operator je komutativan, odnosno redosljed relacija nije bitan.

**Prirodni spoj ( $r \bowtie s$ )** je skup svih slogova nad relacijskom shemom koja je spoj obiju shema ( $r$  i  $s$ ). Rezultat prirodnog spoja je spoj istih vrijednosti atributa koji su zajednički jednoj i drugoj tablici.

Proširenje prirodnog spoja je *vanjski spoj*. Vrste vanjskih spojeva:

- lijevi vanjski spoj (engl. *left outer join*),
- desni vanjski spoj (engl. *right outer join*),
- potpuni vanjski spoj (engl. *full join*).

*Lijevi vanjski spoj* - postupak:

1. Izvršiti prirodni spoj.
2. Za svaki redak iz prve tablice, koji u prvom koraku nije spojiv ni s jednim retkom druge tablice dodati jedan redak u kojem se rezultat oblikuje iz neuparenog retka prve tablice, a ostatak retka poprima NULL vrijednosti.

*Desni vanjski spoj* – postupak:

1. Izvršiti prirodni spoj.
2. Za svaki redak iz druge tablice, koji u prvom koraku nije spojiv ni s jednim retkom prve tablice, dodati jedan redak u kojem se rezultat oblikuje iz neuparenog retka druge tablice, a ostatak retka poprima NULL vrijednosti.

*Potpuni vanjski spoj* – postupak:

1. Izvršiti prirodni spoj.
2. Za svaki redak iz prve tablice, koji u prvom koraku nije spojiv ni s jednim retkom druge tablice, dodati jedan redak u kojem se rezultat oblikuje iz neuparenog retka prve tablice, a ostatak retka poprima NULL vrijednosti.

3. Za svaki redak iz druge tablice, koji u prvom koraku nije spojiv ni s jednim retkom prve tablice, dodati jedan redak u kojem se rezultat oblikuje iz neuparenog retka druge tablice, a ostatak retka poprima NULL vrijednosti.

Primjer prirodnog spoja dvaju tablica (Slika 48):

Artikli skladišta				
Šifra artikla	Naziv artikla	Jedinica mjere	Dobavljačeva šifra	Planska cijena
001	Coca Cola	LIT	CC1L-005	15,00
005	Mineralna	LIT	007	10,00
002	Pršut	KG	Pr-Slav-15	190,15
006	Kruh	KG	11	7,00

Artikli maloprodaje				
Šifra artikla	Naziv artikla	Barcode	Planska cijena	Trenutno stanje
002	Pršut	38635687	190,15	15
001	Coca Cola	38566833	15,00	200
005	Mineralna	876432	10,00	195
007	Kava	38523456	20,00	65

Spojeni artikli						
Šifra artikla	Naziv artikla	Jedinica mjere	Dobavljačeva šifra	Planska cijena	Barcode	Trenutno stanje
001	Coca Cola	LIT	CC1L-005	15,00	38566833	200
005	Mineralna	LIT	007	10,00	876432	195
002	Pršut	KG	Pr-Slav-15	190,15	38635687	15

Slika 48: Primjer prirodnog spoja dvaju tablica (izvor: Autori)

U prikazanom primjeru izvršava se prirodni spoj relacija *Artikli skladišta* (s atributima: *Šifra artikla*, *Naziv artikla*, *Jedinica mjere*, *Dobavljačeva šifra*, *Planska cijena*) i *Artikli maloprodaje* (s atributima: *Šifra artikla*, *Naziv artikla*, *Barcode*, *Planska cijena*, *Trenutno stanje*). Navedene relacije imaju različite relacijske sheme. U prirodnom spoju tih dvaju relacija nalaze se svi atributi iz obje relacije. Rezultat prirodnog spoja je spoj vrijednosti tih atributa koje su zajedničke jednoj i drugoj tablici (u ovom primjeru to su artikli s nazivima *Coca Cola*, *Mineralna* i *Pršut*).

Primjer lijevog vanjskog spoja (Slika 49):

Artikli skladišta				
Šifra artikla	Naziv artikla	Jedinica mjere	Dobavljačeva šifra	Planska cijena
001	Coca Cola	LIT	CC1L-005	15,00
005	Mineralna	LIT	007	10,00
002	Pršut	KG	Pr-Slav-15	190,15
006	Kruh	KG	11	7,00

Artikli maloprodaje				
Šifra artikla	Naziv artikla	Barcode	Planska cijena	Trenutno stanje
002	Pršut	38635687	190,15	15
001	Coca Cola	38566833	15,00	200
005	Mineralna	876432	10,00	195
007	Kava	38523456	20,00	65

Spojeni artikli						
Šifra artikla	Naziv artikla	Jedinica mjere	Dobavljačeva šifra	Planska cijena	Barcode	Trenutno stanje
001	Coca Cola	LIT	CC1L-005	15,00	38566833	200
005	Mineralna	LIT	007	10,00	876432	195
002	Pršut	KG	Pr-Slav-15	190,15	38635687	15
006	Kruh	KG	11	7,00	NULL	NULL

Slika 49: Primjer lijevog vanjskog spoja dvaju tablica (izvor: Autori)

U prikazanom primjeru izvršava se lijevi vanjski spoj relacija *Artikli skladišta* (s atributima: *Šifra artikla*, *Naziv artikla*, *Jedinica mjere*, *Dobavljačeva šifra*, *Planska cijena*) i *Artikli maloprodaje* (s atributima: *Šifra artikla*, *Naziv artikla*, *Barcode*, *Planska cijena*, *Trenutno stanje*). Navedene relacije imaju različite relacijske sheme. Bitan je redosljed relacija. Iz tablice koja je smještena lijevo (*Artikli skladišta*) prikazuju se svi željeni atributi i svi redci s vrijednostima. Dodaju im se ostali željeni atributi iz desne tablice (*Artikli maloprodaje*) koji se ne nalaze u lijevoj tablici (*Barcode*, *Trenutno stanje*). Ako za vrijednosti iz lijeve tablice postoje odgovarajuće vrijednosti iz desne tablice, one se dodaju, u protivnom se zapisuju NULL vrijednosti.



Primjer desnog vanjskog spoja (Slika 50):

Artikli skladišta				
Šifra artikla	Naziv artikla	Jedinica mjere	Dobavljačeva šifra	Planska cijena
001	Coca Cola	LIT	CC1L-005	15,00
005	Mineralna	LIT	007	10,00
002	Pršut	KG	Pr-Slav-15	190,15
006	Kruh	KG	11	7,00

Artikli maloprodaje				
Šifra artikla	Naziv artikla	Barcode	Planska cijena	Trenutno stanje
002	Pršut	38635687	190,15	15
001	Coca Cola	38566833	15,00	200
005	Mineralna	876432	10,00	195
007	Kava	38523456	20,00	65

Spojeni artikli						
Šifra artikla	Naziv artikla	Jedinica mjere	Dobavljačeva šifra	Planska cijena	Barcode	Trenutno stanje
001	Coca Cola	LIT	CC1L-005	15,00	38566833	200
005	Mineralna	LIT	007	10,00	876432	195
002	Pršut	KG	Pr-Slav-15	190,15	38635687	15
007	Kava	NULL	NULL	20,00	38523456	65

Slika 50: Primjer desnog vanjskog spoja dvaju tablica (izvor: Autori)

U prikazanom primjeru izvršava se desni vanjski spoj relacija *Artikli skladišta* (s atributima: *Šifra artikla*, *Naziv artikla*, *Jedinica mjere*, *Dobavljačeva šifra*, *Planska cijena*) i *Artikli maloprodaje* (s atributima: *Šifra artikla*, *Naziv artikla*, *Barcode*, *Planska cijena*, *Trenutno stanje*). Navedene relacije imaju različite relacijske sheme. Bitan je redosljed relacija. Iz tablice koja je smještena desno (*Artikli maloprodaje*) prikazuju se svi željeni atributi i svi redci s vrijednostima. Dodaju im se ostali željeni atributi iz lijeve tablice (*Artikli skladišta*) koji se ne nalaze u desnoj tablici (*Jedinica mjere*, *Dobavljačeva šifra*). Ako za vrijednosti iz desne tablice postoje odgovarajuće vrijednosti iz lijeve tablice, one se dodaju, u protivnom se zapisuju NULL vrijednosti.

Primjer potpunog vanjskog spoja dvaju tablica (Slika 51):

Artikli skladišta					Artikli maloprodaje				
Šifra artikla	Naziv artikla	Jedinica mjere	Dobavljačeva šifra	Planska cijena	Šifra artikla	Naziv artikla	Barcode	Planska cijena	Trenutno stanje
001	Coca Cola	LIT	CC1L-005	15,00	002	Pršut	38635687	190,15	15
005	Mineralna	LIT	007	10,00	001	Coca Cola	38566833	15,00	200
002	Pršut	KG	Pr-Slav-15	190,15	005	Mineralna	876432	10,00	195
006	Kruh	KG	11	7,00	007	Kava	38523456	20,00	65

Spojeni artikli						
Šifra artikla	Naziv artikla	Jedinica mjere	Dobavljačeva šifra	Planska cijena	Barcode	Trenutno stanje
001	Coca Cola	LIT	CC1L-005	15,00	38566833	200
005	Mineralna	LIT	007	10,00	876432	195
002	Pršut	KG	Pr-Slav-15	190,15	38635687	15
006	Kruh	KG	11	7,00	NULL	NULL
007	Kava	NULL	NULL	20,00	38523456	65

Slika 51: Primjer potpunog vanjskog spoja (izvor: Autori)

U prikazanom primjeru izvršava se potpuni vanjski spoj relacija *Artikli skladišta* (s atributima: *Šifra artikla*, *Naziv artikla*, *Jedinica mjere*, *Dobavljačeva šifra*, *Planska cijena*) i *Artikli maloprodaje* (s atributima: *Šifra artikla*, *Naziv artikla*, *Barcode*, *Planska cijena*, *Trenutno stanje*). Navedene relacije imaju različite relacijske sheme. Za svaki redak iz relacije *Artikli skladišta* koji nakon provedenog prirodnog spoja nije spojiv ni s jednim retkom relacije *Artikli maloprodaje*, dodaje se redak u kojem se rezultat oblikuje iz neuparenog retka relacije *Artikli skladišta*, a ostatak retka poprima NULL vrijednosti (dodan je redak s artiklom *Kruh*). Za svaki redak iz relacije *Artikli maloprodaje*, koji nakon provedenog prirodnog spoja nije spojiv ni s jednim retkom relacije *Artikli skladišta*, dodaje se redak u kojem se rezultat oblikuje iz neuparenog retka relacije *Artikli maloprodaje*, a ostatak retka poprima NULL vrijednosti (dodan je redak s artiklom *Kava*).

#### 5.4.2.4. Relacijski operator *preimenovanje*

Relacijski operator **preimenovanje** je *unaran operator*. To znači da se izvodi nad jednom relacijskom shemom i pripadnom relacijom (tablicom).

Relacijski operator **preimenovanje** ( $\delta$ ) omogućuje preimenovanje određenih atributa postojeće relacijske sheme. Vrijednosti atributa po svim slogovima postojeće relacije ostaju nepromijenjene, isključivo se mijenja naziv atributa (stupca tablice).

Primjer preimenovanja tablice (Slika 52):

KUPAC			
OIB	Naziv	Adresa	PostBr
12345678912	Tvrtka1	Ivana Lučića 4	10000
23456789123	Tvrtka2	Vukovarska 55	51000
34567891234	Tvrtka3	Cvečićev uspon	52100

Preimenovati attribute *Naziv* i *Adresa* u *Naziv\_tvrtke* i *Sjedište*

KUPAC			
OIB	Naziv_tvrtke	Sjedište	PostBr
12345678912	Tvrtka1	Ivana Lučića 4	10000
23456789123	Tvrtka2	Vukovarska 55	51000
34567891234	Tvrtka3	Cvečićev uspon	52100

Slika 52: Primjer preimenovanja (izvor: Autori)

U prikazanom primjeru izvršava se preimenovanje atributa *Naziv* i *Adresa* u *Naziv\_tvrtke* i *Sjedište*. Vrijednosti atributa se ne mijenjaju, mijenjaju se samo željeni nazivi atributa.

### 5.4.3. Izvedeni relacijski operatori

#### 5.4.3.1. Relacijski operator *produkt*

Relacijski operator **produkt** koristi se za zadane dvije različite relacijske sheme i različite relacije.

**Produkt**  $r \otimes s$ : Ako relacijske sheme nisu disjunktne, tada se na jednoj relacijskoj shemi primjenjuje operator preimenovanje ( $\delta$ ) da se postigne disjunktnost, a zatim se primjenjuje operator prirodni spoj ( $\infty$ ).

Prvi korak je provjera jesu li relacijske sheme disjunktne (to znači da relacijske sheme nemaju zajedničke atribute), pa ako nisu, potrebno ih je učiniti disjunktima. To će se postići preimenovanjem zajedničkih atributa u jednoj relacijskoj shemi. Drugi korak je prirodno spajanje dobivenih relacija iz prvog koraka, a s obzirom na to da relacijske sheme zbog 1. koraka nemaju zajedničke atribute, pa ne postoji mogućnost spajanja nekih redova iz prve relacije s nekim redovima iz druge relacije po jednakim vrijednostima u zajedničkim atributima jer relacijske sheme nemaju zajedničke atribute, spojiti će se svaki red iz prve relacije sa svakim redom iz druge relacije.

#### 5.4.3.2. Relacijski operator *aktivni komplement*

Relacijski operator **aktivni komplement (AC)** je *unaran operator*. To znači da se izvodi nad jednom relacijskom shemom i pripadnom relacijom (tablicom).

Rezultat **aktivnog komplementa** je jednaka relacijska shema. Slogovi komplement relacije su *projekcije* koje su nastale projiciranjem početne relacijske sheme po svakom pojedinom atributu. Izvodi se *prirodno spajanje* svih navedenih projekcija. Dobiveni slogovi oduzimaju se (*razlika*) od početne relacije.

### 5.4.3.3. Relacijski operator *kvocijent*

Relacijski operator **kvocijent** ( $\div$ ) je *binaran operator*. To znači da djeluje između dvije relacije. Koristi se između dvije zadane relacije koje imaju različite relacije i različite relacijske sheme. Relacijska shema S je pravi podskup relacijske sheme R, dakle svi atributi iz S su sigurno sadržani u R. Nastaje nova relacijska shema T gdje je  $T=R-S$ . Dakle atributi nove relacijske sheme su oni koji su u relacijskoj shemi R, a nisu u relacijskoj shemi S.

## 5.5. Funkcijska zavisnost

**Funkcijska zavisnost** je svojstvo jednog ili više atributa da jedinstveno određuju vrijednost jednog ili više preostalih atributa. Funkcijska zavisnost je temelj funkcioniranja relacijskih baza podataka.

Kako u relaciji ne smiju postojati dvije jednake n-torke (dva retka tablice s jednakim vrijednostima), mora se definirati **primarni ključ n-torke** koji jedinstveno identificira svaku pojedinu n-torku (Hercigonja, 2017).

**Primarni ključ** je atribut koji *funkcijski određuje* (jedinstveno identificira) sve ostale neključne attribute relacijske sheme.

**Vrijednost primarnog ključa** (Kaluža, 2008):

- mora biti jedinstvena,
- ne smije ni u jednoj n-torki ostati neupisana (imati null vrijednost),
- mora biti minimalna – sastojati se od najmanjeg mogućeg broja atributa,
- mora **funkcijski određivati** sve ostale neključne attribute relacijske sheme.

Funkcijska zavisnost temelji se na dva podatka: *tko* i *što* -> jedan podatak koji *određuje* i drugi koji je od njega *određen*.

**A → B** - atribut (A) određuje atribut (B)

Atribut A određuje atribut B samo onda kada za sve slogove (n-torke, retke) relacije vrijedi da određena vrijednost atributa A uvijek ima jednu (istu) vrijednost atributa B. Atribut A određuje atribut B čita se i kao: atribut B funkcijski zavisi (ovisi) o atributu A.

## 5.6. Redundancija podataka

Relacije dobivene u prvoj fazi logičkog oblikovanja (izrada Relacijskog modela) mogu imati nepoželjna svojstva. Osnovno nepoželjno svojstvo je pamćenje (zapisivanje) *istih informacija* na više mjesta nego što je potrebno (**redundancija podataka**). *Redundancija podataka* utječe na performanse i integritet podataka te stvara *anomalije* prilikom *upisivanja* podataka u relaciju, *izmjene* podataka u relaciji ili prilikom njihovog *brisanja*. Kako bi se izbjegli problemi u ažuriranju baze podataka provodi se postupak **normalizacije**.

**Redundancija podataka** je nepotrebno ponavljanje podataka u relaciji tj. skladištenje jedne informacije na više različitih mjesta u modelu podataka. Podaci u relaciji koji se mogu izvesti iz već postojećih podataka su redundantni (Kaluža, 2008).

Na slici 53 vidljiv je primjer prikaza redundantnih podataka. Npr. svaki korisnik bi znao odrediti da je u stupcu *Sjedište* naziv grada *Rijeka* na temelju vrijednosti u stupcu *Broj pošte* koji je *51000*. Dakle, podaci zapisani u stupcu *Sjedište* su redundantni podaci jer se mogu izvesti iz već postojećih podataka o *Broju pošte*.

KUPAC		
Naziv kupca (PK)	Sjedište	Broj pošte
Veleučilište	Rijeka	51000
3. Maj	Rijeka	51000
Filozofski fakultet	Zagreb	10000
Telekom	Zagreb	10001
Zidine d.o.o.	Dubrovnik	20000

Slika 53: Primjer redundantnih podataka (izvor: Autori)

Kako je već navedeno, zbog redundantnih podataka javljaju se **anomalije**. One se očituju onemogućavanjem jednostavnog ažuriranja ili pak gubitkom informacije (Data Anomalies, 2018).

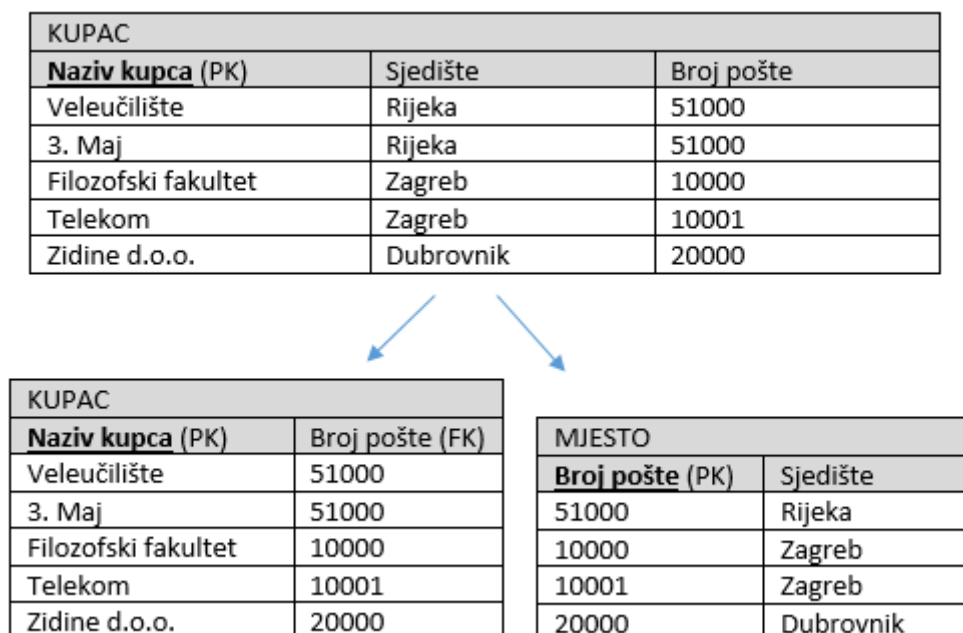
Na gornjoj slici (slika 53) mogu se uočiti tri vrste anomalija: anomalija brisanja, anomalija upisivanja i anomalija modifikacije (izmjene, ažuriranja).

**Anomalija brisanja** – brisanjem 5. reda iz relacije *KUPAC* na slici 53 (kupac *Zidine d.o.o.*) gubi se informacija o gradu *Dubrovniku* i njegovom poštanskom broju.

**Anomalija upisivanja** – zbog toga što primarni ključ relacije *KUPAC* čini atribut *Naziv kupca*, čime je u slogu obvezno postojanje vrijednosti tog atributa, nije moguće spremati podatke o novom gradu dok se ne upiše podatak za atribut *Naziv kupca*.

**Anomalija modifikacije (izmjene)** – kada bi se trebao izvršiti zahtjev promjene imena grada koji ima poštanski broj *51000*, kako bi baza podataka bila konzistentna, promjena imena grada na slici 53 trebala bi se izvršiti na dva mjesta (u dvije ćelije). Ako se promjena imena ne izvrši u obje ćelije nego samo u jednoj, baza podataka više neće biti konzistentna, zbog toga što će postupak čitanja informacije o gradu koji ima poštanski broj *51000*, iz prvog retka dati jedan rezultat, a iz drugog retka dati drugi rezultat (postavlja se pitanje koja je vrijednost točna – baza je nekonzistentna).

Navedene anomalije rješavaju se **dekomponiranjem relacije** na dvije nove relacije poštujući funkcijsku zavisnost: *Broj pošte* -> *Sjedište* (slika 54):



Slika 54: Primjer dekompozicije relacije (izvor: Autori)

U prikazanom primjeru se redundantne vrijednosti atributa *Sjedište* izdvajaju u novu relaciju kojoj je dodijeljeno ime *Mjesto*. Atribut *Sjedište* funkcijski zavisi o atributu *Broj pošte* koji postaje primarni ključ relacije *Mjesto*. U relaciji *Kupac* atribut *Broj pošte* ostaje u obliku vanjskog ključa kako bi se poštovalo pravilo da dekomponirane relacije moraju imati barem jedan zajednički atribut.

Anomalije unosa, modifikacije i brisanja više ne postoje, a redundancija je nestala. Sve funkcijske zavisnosti i dalje su žive što znači da je dekompozicija ispravno izvedena bez gubitka zavisnosti i informacija.

**Svaka ispravna dekompozicija mora poštivati dva pravila:**

- dekompozicije moraju imati *barem jedan zajednički atribut*;
- ti zajednički atributi moraju biti *ključni* u barem jednoj od dekomponiranih tablica.

## 5.7. Normalizacija podataka

Postoje razne vrste normalizacija (npr. horizontalna normalizacija, vertikalna normalizacija sintezom itd). Ovdje će se detaljno objasniti *vertikalna normalizacija dekompozicijom*.

Ako se relacijska shema temelji na konceptualnom oblikovanju (izrađenom EVA modelu podataka) i logičkom oblikovanju (izrađenom Relacijskom modelu podataka), njene relacije vjerojatno već jesu u normalnim formama (objašnjeno ispod). No, često je potrebno normalizirati drugačije dobivene sheme (npr. "shema" excel tablica koje se žele zamijeniti relacijskom bazom podataka).

**Normalizacija** je proces dekomponiranja relacijske sheme na dvije ili više novih relacijskih shema (Hercigonja, 2017).

**Cilj normalizacije** je maksimalno ukloniti redundanciju podataka te ukloniti sve uočene anomalije (anomalije upisivanja, brisanja ili izmjene).



Normalizacija se temelji na postupku koji rezultira relacijskom shemom u odgovarajućoj *normalnoj formi* (Maleković & Schatten, 2017).

**Normalne forme:**

- 0. normalna forma (nenormalizirana relacija)
- 1. normalna forma
- 2. normalna forma
- 3. normalna forma
- Boyce-Coddova normalna forma
- 4. normalna forma
- 5. normalna forma

**Normalizacija** je proces transformiranja (dekompozicije) relacija u oblik koji zadovoljava normalne forme. To se postiže *analizirajući funkcijske zavisnosti* između atributa tablice te, po potrebi, razdvajanja jedne veće tablice na više manjih (Hercigonja, 2017). Dekomponiranje se vrši dok se ne postigne maksimalna željena normalna forma – minimalno 3NF (treća normalna forma). Potrebno je imati što manje redundantnih podataka. *Normalna forma* je osnovno pravilo pri strukturiranju baza podataka.

**Normalizacija** je metoda za raščlanjivanje relacije na dvije ili više relacija koje se nalaze u višoj normalnoj formi od početne relacije i koje nemaju anomalije početne relacije. Ako je u relaciji prisutna redundancija, relacija se treba normalizirati (transformirati u višu normalnu formu). Normalna forma ovisi o vrsti redundancije odnosno, ako relacija nema ključ nije niti u 1NF pa se treba provesti postupak svođenja u 1NF. Provođenje transformacije u više normalne forme ovisi o vrsti tranzitivne zavisnosti koja je prisutna u relaciji koja je u 1NF. Najviša normalna forma koja se može postići funkcijskim zavisnostima je BCNF.

Ako se već na početku uoče svi potrebni entiteti, atributi i veze, tada se oni prirodno pretvaraju u relacije koje zadovoljavaju pravila normalnih formi. Ako je polazna shema loše dizajnirana, tada će postupak normalizacije ispraviti greške. Za većinu praktičnih primjera dovoljno je relacije normalizirati do 3NF.

Normalizirane relacije uvijek moraju imati jedan *zajednički atribut* koji je u jednoj od njih u svojstvu primarnog, a u drugoj u svojstvu vanjskog ključa.

Tijekom normalizacije ne smije doći do gubitka informacije. Smatra se da je dekompozicija sheme relacije izvedena bez gubitka informacije ako vrijedi: bilo koja relacija stvorena nad shemom R rezultat je prirodnog spajanja svih njezinih projekcija koje su nastale dekompozicijom.

Reverzibilnost sheme relacije dokazuje se reverzibilnošću dekompozicije relacija zadanih na toj shemi. Operatori projekcija i prirodni spoj osnova su vertikalne normalizacije. Projekcija relacije odgovara dekompoziciji sheme relacije, a unija dekomponiranih shema odgovara prirodnom pridruživanju dekomponiranjem dobivenih relacija. Dekompozicija sheme relacije je reverzibilna ako: dekomponirane sheme imaju barem jedan zajednički atribut; zajednički atributi čine ključ u barem jednoj od dekomponiranih shema.

Normalizacijom bez gubitka informacije može se dogoditi gubitak funkcijskih zavisnosti. Dekomponirane sheme respektiraju određene funkcijske zavisnosti iz osnovne sheme relacije. Dekompozicija bez gubitka funkcijskih zavisnosti dogodit će se ako unija projekcija funkcijskih zavisnosti na dekomponiranim shemama predstavlja osnovni skup funkcijskih zavisnosti definiran na početnoj shemi relacije.

*Rissanenovo pravilo (Kaluža, 2008):*

- Svaka funkcijska zavisnost zadana nad početnom shemom R sadržana je ili može biti izvedena iz unije funkcijskih zavisnosti zadanih na projekcijama.
- Zajednički atributi u dvije projekcije sadrže ključ u barem jednoj od njih.

### 5.7.1. Primjer normalizacije relacijske sheme do 3. normalne forme

Dana je sljedeća relacijska shema koju je potrebno dekomponirati na manje relacijske sheme poštujući pravila normalizacije, a s ciljem uklanjanja redundantnih podataka i s njima povezanih anomalija:

Računi (BrojRačuna, NazivKupca, AdresaKupca, PoštanskiBroj, Grad, MatičniBrojKupca, DatumRačuna, IsporukaNaTemelju, DatumIsporuke, VrstaPlaćanja, NazivArtikla, ŠifraArtikla, Količina, JedinicaMjere, Cijena, IznosBezPoreza, Porez, IznosSPorezom, UkupniIznosRačuna)

#### **1. normalna forma:**

Relacija je u **1 NF** ako je vrijednost svakog atributa **jednostruka i nedjeljiva** (vrijednosti u slogovima na svim atributima moraju biti jednostavne. To znači da se u jednu ćeliju ne mogu upisati dvije vrijednosti).

Relacija je u **1 NF** ako su svi njezini neključni atributi **funkcijski zavisni o ključu** relacije.

Relacija je u **1 NF** ako se može prikazati u obliku potpune dvodimenzionalne tablice.

**Funkcijska zavisnost:** neki atribut A2 je funkcijски zavisan o atributu A1 ako je svakoj vrijednosti atributa A1 moguće pridružiti samo jednu vrijednost atributa A2.

#### **Postupak svođenja relacije u 1NF:**

Sve attribute koji nisu funkcijски zavisni o ključu relacije potrebno je izdvojiti u nove relacije. Nove relacije imaju ključ sastavljen od ključa početne relacije i ključa grupe atributa koji su izdvojeni.

Primjer:

Računi (BrojRačuna, NazivKupca, AdresaKupca, PoštanskiBroj, Grad, MatičniBrojKupca, DatumRačuna, IsporukaNaTemelju, DatumIsporuke, VrstaPlaćanja, NazivArtikla, ŠifraArtikla, Količina, JedinicaMjere, Cijena, IznosBezPoreza, Porez, IznosSPorezom, UkupniIznosRačuna)

Je li relacija u 1NF?

Postoji li takav skup atributa koji čini zasebnu tablicu? Da. To znači da relacija nije u 1NF.

Uočen skup atributa se izdvaja u novu relaciju kojoj se dodjeljuje novi naziv – npr. ArtikliRačuna. Ključ nove relacije sastoji se od ključa početne relacije i ključa grupe atributa koji su izdvojeni. Sada umjesto jedne, postoje dvije relacije:

Računi (BrojRačuna, NazivKupca, AdresaKupca, PoštanskiBroj, Grad, MatičniBrojKupca, DatumRačuna, IsporukaNaTemelju, DatumIsporuke, VrstaPlaćanja, UkupniIznosRačuna)

ArtikliRačuna (BrojRačuna, ŠifraArtikla, NazivArtikla, Količina, JedinicaMjere, Cijena, IznosBezPoreza, Porez, IznosSPorezom)

## **2. normalna forma:**

Relacijska shema je u **2NF** ako je u prvoj normalnoj formi i ako su svi njezini neključni atributi potpuno **funkcijski zavisni o ključu** (o svim dijelovima ključa).

**2NF** odnosi se samo na relacijske sheme **sa složenim ključevima**.

**Potpuna funkcijska zavisnost:** atribut A je potpuno funkcijski zavisan o složenom ključu ako funkcijski zavisi o cijelom ključu, a ne o nekom od njegovih sastavnih dijelova.

### **Postupak svođenja relacije u 2NF:**

Izdvajanje iz relacije (koja je već u 1NF) onih atributa koji zavise samo o dijelu primarnog ključa i njihovo smještanje u novu relaciju.

Ključ nove relacije je dio ključa polazne relacije o kojem funkcijski zavise atributi nove relacije.

Primjer:

Računi (BrojRačuna, NazivKupca, AdresaKupca, PoštanskiBroj, Grad, MatičniBrojKupca, DatumRačuna, IsporukaNaTemelju, DatumIsporuke, VrstaPlaćanja, UkupniIznosRačuna)

ArtikliRačuna (BrojRačuna, ŠifraArtikla, NazivArtikla, Količina, JedinicaMjere, Cijena, IznosBezPoreza, Porez, IznosSPorezom)



Za svođenje u 2. normalnu formu interesantne su samo relacije sa *složenim ključevima*

Jesu li relacije u 2NF?

Postoji li funkcijska zavisnost neključnih atributa o dijelu ključa? Da (vidljivo uokvireno iznad).

To znači da relacija nije u 2NF.

Uočen skup atributa se izdvaja u novu relaciju kojoj se dodjeljuje novi naziv – npr. Artikli. Ključ nove relacije je dio ključa polazne relacije o kojem funkcijski zavise atributi nove relacije. Sada postoje tri relacijske sheme:

Računi (BrojRačuna, NazivKupca, AdresaKupca, PoštanskiBroj, Grad, MatičniBrojKupca, DatumRačuna, IsporuKaNaTemelju, DatumIsporuke, VrstaPlaćanja, UkupniIznosRačuna)

ArtikliRačuna (BrojRačuna, ŠifraArtikla, Količina, Cijena, IznosBezPoreza, Porez, IznosSPorezom)

Artikli (ŠifraArtikla, NazivArtikla, JedinicaMjere)

### **3. normalna forma:**

Relacijska shema je u **3NF** ako je u prvoj normalnoj formi i ako neključni atributi nisu međusobno funkcijski zavisni - svi atributi koji nisu dio ključa ne smiju biti međusobno ovisni.

Drugim riječima: svaki neključni atribut mora zavisiti o ključu i to o cijelom ključu i ni o čemu drugom nego o ključu.

#### **Postupak svođenja relacije u 3NF:**

Sve neključne attribute koji su međusobno funkcijski zavisni potrebno je izdvojiti u novu relacijsku shemu.

Primjer:

Računi (BrojRačuna, NazivKupca, AdresaKupca, PoštanskiBroj, Grad, MatičniBrojKupca, DatumRačuna, IsporuKaNaTemelju, DatumIsporuke, VrstaPlaćanja, UkupniIznosRačuna)

ArtikliRačuna (BrojRačuna, ŠifraArtikla, Količina, Cijena, IznosBezPoreza, Porez, IznosSPorezom)

Artikli (ŠifraArtikla, NazivArtikla, JedinicaMjere)

Jesu li relacije u 3NF?

Postoje li neključni atributi koji su međusobno funkcijski zavisni? Da (vidljivo uokvireno iznad).  
To znači da relacija nije u 3NF.

Uočen skup atributa se izdvaja u novu relaciju kojoj se dodjeljuje novi naziv – npr. Kupci. Ključ nove relacije je dio ključa polazne relacije o kojem funkcijski zavise atributi nove relacije. Sada postoje četiri relacijske sheme:

ArtikliRačuna (BrojRačuna, ŠifraArtikla, Količina, Cijena, IznosBezPoreza, Porez, IznosSPorezom)

Artikli (ŠifraArtikla, NazivArtikla, JedinicaMjere)

Računi (BrojRačuna, MatičniBrojKupca, DatumRačuna, IsporukaNaTemelju, DatumIsporuke, VrstaPlaćanja, UkupniIznosRačuna)

Kupci (MatičniBrojKupca, NazivKupca, AdresaKupca, PoštanskiBroj, Grad)

Jesu li relacije u 3NF?

Postoje li i dalje neključni atributi koji su međusobno funkcijski zavisni? Da (vidljivo uokvireno iznad). To znači da relacija nije u 3NF.

Uočen skup atributa se izdvaja u novu relaciju kojoj se dodjeljuje novi naziv – npr. Gradovi. Ključ nove relacije je dio ključa polazne relacije o kojem funkcijski zavise atributi nove relacije. Sada postoji pet relacijskih shema:

Računi (BrojRačuna (PK), MatičniBrojKupca (FK1) DatumRačuna, IsporukaNaTemelju, DatumIsporuke, VrstaPlaćanja, UkupniIznosRačuna)

ArtikliRačuna (BrojRačuna (PK, FK1), ŠifraArtikla (PK, FK2), Količina, Cijena, IznosBezPoreza, Porez, IznosSPorezom)

Artikli (ŠifraArtikla (PK), NazivArtikla, JedinicaMjere)

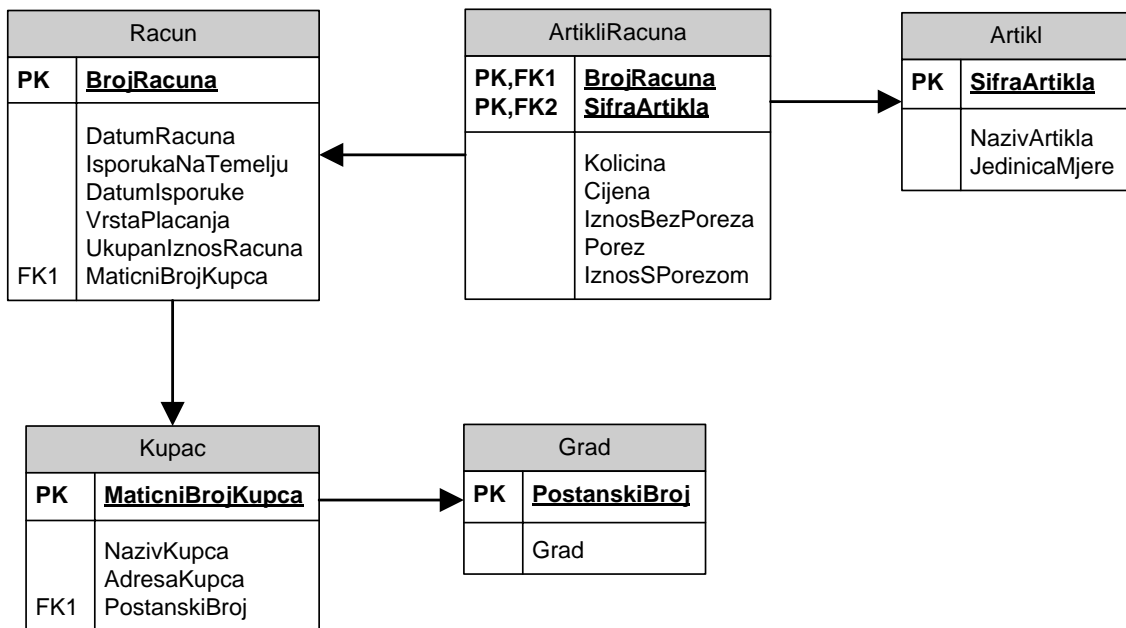
Kupci (MatičniBrojKupca (PK), NazivKupca, AdresaKupca, PoštanskiBroj (FK1))

Gradovi (PoštanskiBroj (PK), Grad)

Postoje li zahtjevi za normaliziranjem u 1NF, 2NF, 3NF?

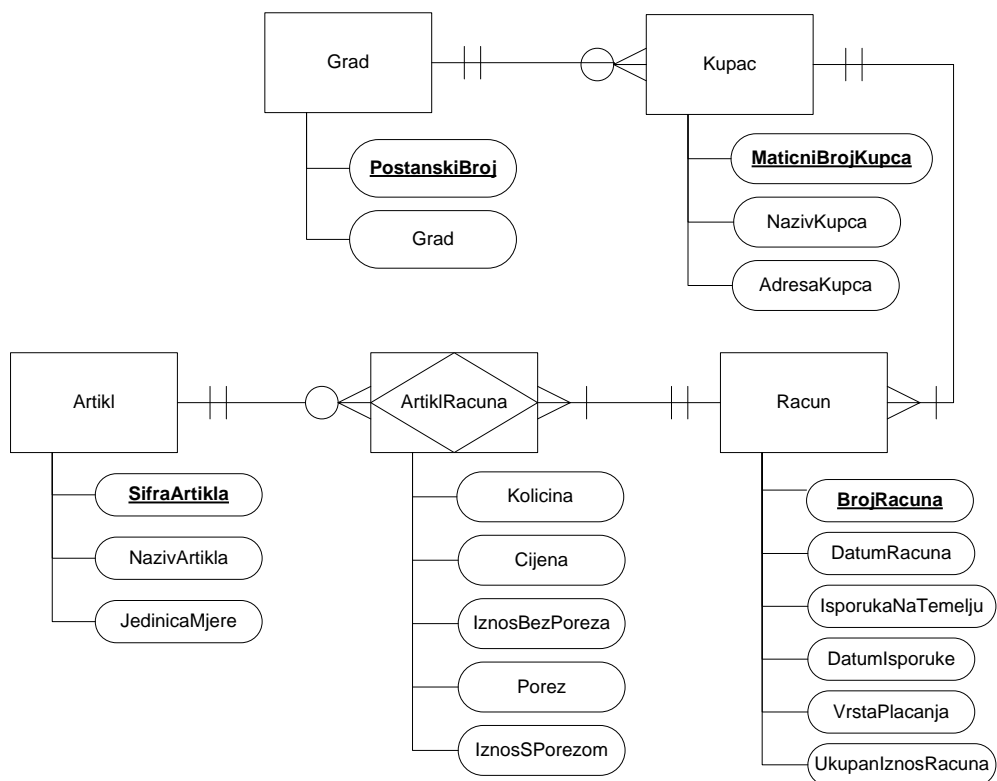
Ne, relacijske sheme sada su normalizirane.

Relacijski model izrađen na temelju dobivenih relacijskih shema (slika 55):



Slika 55: Primjer relacijskog modela podataka (izvor: Autori)

Može se pretpostaviti da bi EVA model (na temelju kojeg je nastao Relacijski model sa slike 55) izgledao kao na slici 56:



Slika 56: Primjer EVA modela podataka (izvor: Autori)

Prijedlog prikaza baze podataka izrađene na temelju dobivenih relacijskih shema (slika 57):

GRAD	
PošanskiBroj	Grad
51000	Rijeka
10000	Zagreb

KUPAC			
MatičniBrojKupca	NazivKupca	AdresaKupca	PošanskiBroj
123	Tvrtka1	Jadranski trg 5	51000
546	Tvrtka2	Vukovarska 11	10000
321	Tvrtka3	Pulska 5	51000

RACUN						
BrojRacuna	MatičniBrojKupca	DatumRacuna	IsporukaNaTemelju	DatumIsporuke	VrstaPlaćanja	UkupanIznosRačuna
1	123	12.12.2014	Narudžbe	15.12.2014	Virmanom	75000
2	546	15.10.2014	Upita	20.10.2014	Gotovinom	

ARTIKL		
ŠifraArtikla	NazivArtikla	JedinicaMjere
01	Laptop	Kom
02	Televizija	Kom
03	Tipkovnica	Kom

ARTIKL RAČUNA						
BrojRačuna	ŠifraArtikla	Količina	Cijena	IznosBezPoreza	Porez	IznosSPorezom
1	02	15	3000	45000	25	56250
1	01	3	5000	15000	25	18750

Slika 57: Primjer relacijske baze podataka (izvor: Autori)



## 6. Model procesa

**Model procesa** je pojednostavljena reprezentacija znanja o nekom sustavu s aspekta poslovnih funkcija (procesa) koje taj sustav obavlja. Opisuje processe koji se odvijaju nad podacima, a čijim se djelovanjem ostvaruju ciljevi promatranog sustava. Prikazuje skup procesa koji izvode promjene u sustavu i formiraju izlaze iz sustava. Prikazuje skup procesa nad podacima - skup aktivnosti (poslova) nad skupom podataka (Pavlič, *et al.*, 2014).

*Model procesa* sastoji se od:

1. **Dijagrama dekompozicije procesa (poslovnih funkcija)** – dijagram koji neki složeni sustav prikazuje u obliku *hijerarhijske strukture podsustava*: sustav se dekomponira na više podustava, pa svaki podsustav opet na više svojih podsustava, itd – dijeljenje procesa na podprocese dok se ne dođe do najjedostavnijih procesa koji se više ne mogu dekomponirati.
2. **Dijagrama tokova podataka (DTP)** – dijagram koji prikazuje ulaze podataka u sustav, izlaze podataka iz sustava, tokove podataka unutar sustava, spremišta podataka unutar sustava i processe obrade tih podataka unutar sustava, kao i vanjske sustave s kojima promatrani sustav razmjenjuje podatke. Dakle, DTP prikazuje ono što neki promatrani (pod)sustav radi.

**Proces** je djelovanje koje se izvodi s ciljem da se ulazne informacije transformiraju u izlazne informacije. Proces je skup povezanih aktivnosti i odluka, preko kojih elementi sustava ostvaruju cilj svoga postojanja, a za njihovo izvršenje potrebni su određeni resursi i određeno vrijeme.

**Procesi** se sastoje od podprocesa koji se na najnižoj razini raščlanjivanja poklapaju s aktivnostima. **Aktivnosti** su jednostavni poslovi od kojih se sastoje procesi i funkcije. **Funkcija** je apstraktan naziv za skupinu sličnih poslova koji ostvaruju jedan parcijalni interni cilj jednoga organizacijskog dijela poslovne organizacije. Funkcije su npr. nabava, prodaja, putovanja, proizvodnja. Funkcije se sastoje od *podfunkcija*.

## 6.1. Koraci u izgradnji modela procesa

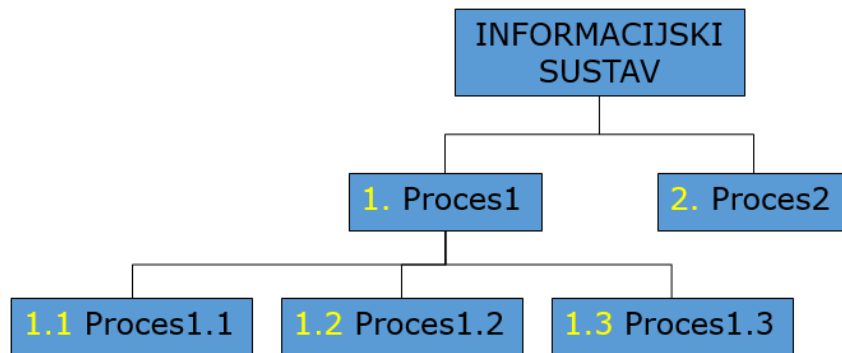
Kako bi se izradio model procesa, potrebno je proći kroz određene korake:

- **Analizirati sustav** s ciljem pripreme za modeliranje – potrebno je proučiti organizacijski (poslovni) sustav za koji se razvija informacijski sustav (IS) kako bi se identificirao kontekst u kojem sustav djeluje, kako bi se identificirale potrebe sustava i njegovi ciljevi.
- **Identificirati procese** koji se provode unutar poslovnog sustava kako bi se izvršavao cilj djelovanja tog sustava.
- Kao rezultat prethodnih koraka, metodom DTP dijagrama **grafički prikazati strukturu procesa** koji se provode unutar poslovnog sustava.

## 6.2. Dijagram dekompozicije

Kako bi se prikazali i raščlanili svi procesi koji se izvode unutar poslovnog sustava, koristi se dijagram dekompozicije. *Dekompozicijom* se složeni sustav rastavlja na više jednostavnijih podsustava. Podsustavi se dalje mogu dekomponirati na svoje manje podsustave itd. Na taj se način dolazi do hijerarhijskog opisa sustava. Rezultat dekompozicijskog postupka je *dijagram dekompozicije*.

Dijagram dekompozicije se prikazuje u obliku hijerarhijskog stabla -> top down tehnika. Proces na dijagramu se numeriraju. Svaka niža razina ima internu numeraciju. Proces se može dekomponirati na minimalno dva podprocesa. Slijedi primjer na slici 58:



Slika 58: Općeniti primjer dijagrama dekompozicije (izvor: Autori)

Na prikazanom općem primjeru (slika 58) neki *Informacijski sustav* dekomponira se na dva procesa: *1. Proces 1* i *2. Proces 2*. *Proces 1* je složen te se on dodatno dekomponira na podprocese: *1.1 Proces 1.1*, *1.2 Proces 1.2* i *1.3 Proces 1.3*.

### 6.3. Dijagram tokova podataka (DTP)

Nakon dijagrama dekompozicije izrađuje se dijagram tokova podataka (DTP) kojim se prikazuje što promatrani sustav radi. *Osobine dijagrama tokova podataka* su sljedeće:

- Služi za grafički prikaz procesa sustava.
- Korisnik i analitičar zajedničkom suradnjom dolaze do modela DTP-a.
- DTP-om su precizno definirani zahtjevi korisnika.
- DTP služi za lakšu komunikaciju korisnika i analitičara.
- Broj osnovnih koncepata (elemenata, dijelova, pojmova) DTP-a je mali (za izgradnju DTP-a se koriste 4 osnovna simbola – simboli za proces, tok podataka, spremište podataka i vanjski sustav).

DTP prikazuje što sustav radi, odnosno što će sustav raditi na logičkoj razini, koje će funkcije izvršavati. Osnovni cilj analize sustava, i svrha DTP-a, je pokazati tokove podataka kroz sustav. DTP pokazuje tokove koji se mogu paralelno izvoditi (kao u stvarnom sustavu) i ne ovise jedan o drugome.

**Koncepti dijagrama toka podataka su** (Pavlić, *et al.*, 2014):

- proces,
- tok podataka,
- spremište podataka,
- vanjski sustav.

**PROCES** (kao koncept DTP-a) predstavlja niz operacija koji na temelju jednog ili više ulaznih tokova podataka generira jedan ili više izlaznih tokova podataka.

*Naziv procesa* je glagol, glagolska imenica ili skup riječi koji ukazuju na svrhu procesa. Simbol kojim se prikazuje proces na dijagramu tokova podataka prikazan je na slici 59:

1. Proces

*Slika 59: Simbol za proces na DTP-u (izvor: Autori)*

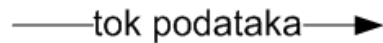
**TOK PODATAKA** kao koncept DTP-a pokazuje kretanje podataka u sustavu. To je skup podataka koji se kreću od jednog dijela sustava k drugom te služe za prijenos informacija i komunikaciju unutar i izvan sustava.

Tok podataka može se nalaziti:

- Između dva procesa
- Od spremišta podataka do procesa
- Od procesa do spremišta podataka
- Od vanjskog sustava do procesa
- Od procesa do vanjskog sustava

Tok podataka označava se linijom sa strelicom u smjeru toka. Na liniji se nalazi naziv toka. *Naziv toka podataka* je imenica u jednini ili kombinacija imenice i pridjeva. Na jednom modelu ne smiju postojati tokovi podataka s istim imenom. Tokovi podataka mogu se numerirati zbog lakše

identifikacije. Simbol kojim se prikazuje tok podataka na dijagramu tokova podataka prikazan je na slici 60:



Slika 60: Simbol za tok podataka na DTP-u (izvor: Autori)

**SPREMIŠTE (SKLADIŠTE) PODATAKA** kao koncept DTP-a je mjesto gdje su podaci pohranjeni za kasniju primjenu u poslovnim procesima. To je element sustava u kojem se čuvaju podaci.

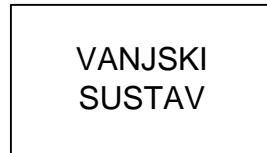
Upis podataka u spremište i čitanje podataka iz spremišta podataka vrše procesi koristeći tokove podataka. U spremištu podataka se čuvaju podaci potrebni za izvršenje nekog procesa ili podaci dobiveni kao rezultat rada nekog procesa. Drugim riječima, spremište podataka je jedan ili više tokova podataka u mirovanju – odloženi sadržaj toka podataka koji je za buduću uporabu pohranio neki proces. *Naziv spremišta podataka* je imenica. Simbol kojim se prikazuje spremište podataka na dijagramu tokova podataka prikazan je na slici 61:



Slika 61: Simboli za spremište podataka na DTP-u (izvor: Autori)

**VANJSKI SUSTAV** kao koncept DTP-a je objekt (drugi sustav) iz okruženja s kojim promatrani sustav komunicira (razmjenjuje podatke). Vanjski sustav može biti izvor tokova podataka ili odredište (ponor) tokova podataka - od njih potječu ulazni tokovi podataka ili u njih utječu izlazni tokovi podataka iz promatranog sustava.

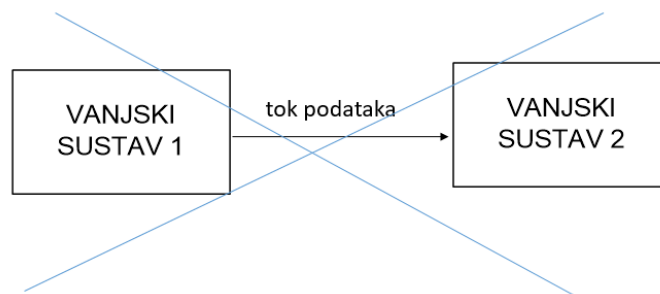
Naziv vanjskog sustava je imenica. Simbol kojim se prikazuje vanjski sustav na dijagramu tokova podataka prikazan je na slici 62:



Slika 62: Simbol za vanjski sustav na DTP-u (izvor: Autori)

### 6.3.1. Pravila kreiranja Dijagrama tokova podataka

1. Dijagram tokova podataka ne sadrži tokove podataka između vanjskih sustava - sustav ne kontrolira događaje izvan svojih granica (slika 63):



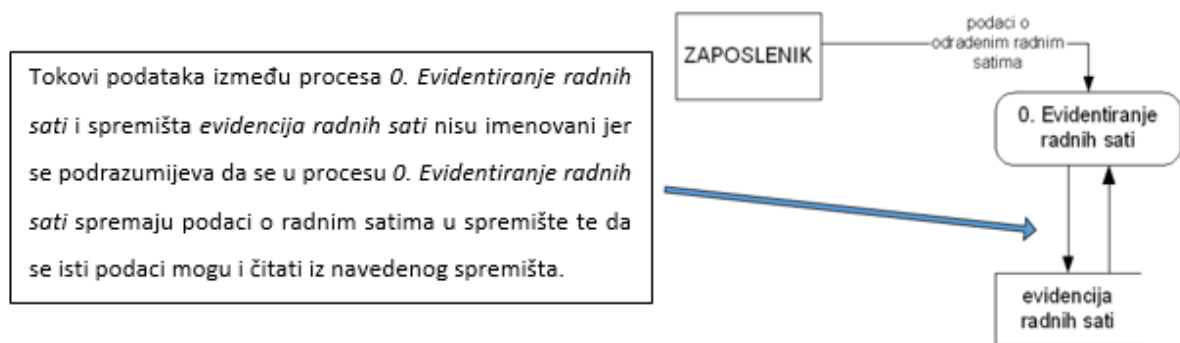
Slika 63: Pravilo kreiranja DTP-a 1 (izvor: Autori)

2. Dijagram tokova podataka ne sadrži tokove podataka od spremišta podataka prema vanjskim sustavima i obrnuto (slika 64). Između uvijek mora biti neki proces koji prijenos podataka inicira i izvodi:



Slika 64: Pravilo kreiranja DTP-a 2 (izvor: Autori)

3. Svaki tok podataka mora imati *ime* koje odražava značenje podataka koje nosi. Izuzetak su *tokovi od i prema skladištima podataka* - podrazumijeva se da tok nosi sadržaj skladišta podataka (slika 65). Ako nije tako tok treba imenovati (ako tok podataka iz skladišta podataka ne prenosi cjelokupni skup podataka, već samo dio - tada se navodi ime toka podataka, npr. "Prosječna ocjena" iz skladišta podataka "evidencija položenih ispita"):

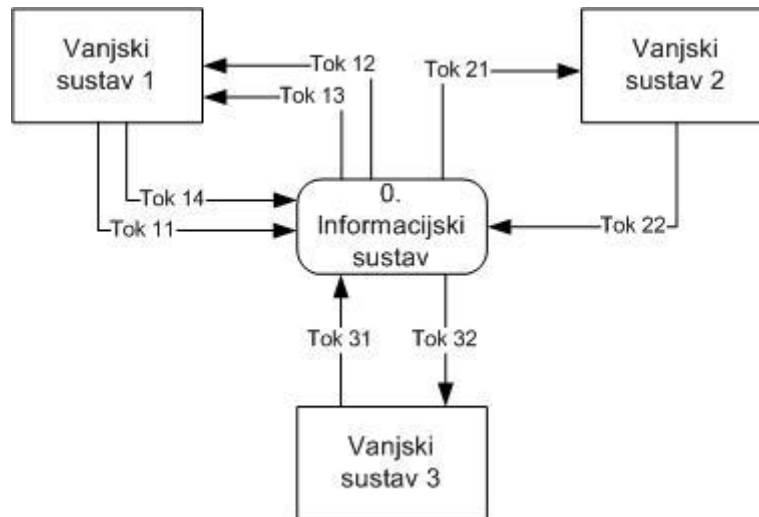


Slika 65: Pravilo kreiranja DTP-a 3 (izvor: Autori)

4. Dijagram tokova podataka cijelog sustava ne sadrži procese koji imaju samo ulaze ili samo izlaze.
5. Dijagram tokova podataka cijelog sustava ne sadrži spremišta podataka koja imaju samo ulaze ili samo izlaze.
6. Svi dijelovi dijagrama tokova podataka (vanjski sustavi, procesi, skladišta podataka, tokovi podataka) koji opisuju sustav na razini  $n$ , moraju se naći i u detaljnijim opisima sustava na razinama  $n+1$
7. Svi tokovi koji ulaze, odnosno izlaze iz jednog procesa na dijagramu više razine, moraju se pojaviti kao ulazni, odnosno izlazni tokovi na dijagramu niže razine na kojoj je taj proces dekomponiran.
8. Skladišta podataka mogu se pojaviti na nižim razinama dekompozicije dtp-a iako se nisu prethodno pojavila na višim razinama. Kada se jednom pojave, moraju se ubuduće pojavljivati i na svim nižim razinama dekompozicije.

### 6.3.2. Razine dekompozicije dijagrama tokova podataka

**Dijagram konteksta** (dijagram nulte razine dekompozicije) je dijagram najviše razine - prikazuje sustav u cijelosti (čime se određeni sustav bavi) i okolinu u kojoj se taj proces odvija. Proces ispred imena može (iako nije obvezno) imati oznaku 0.

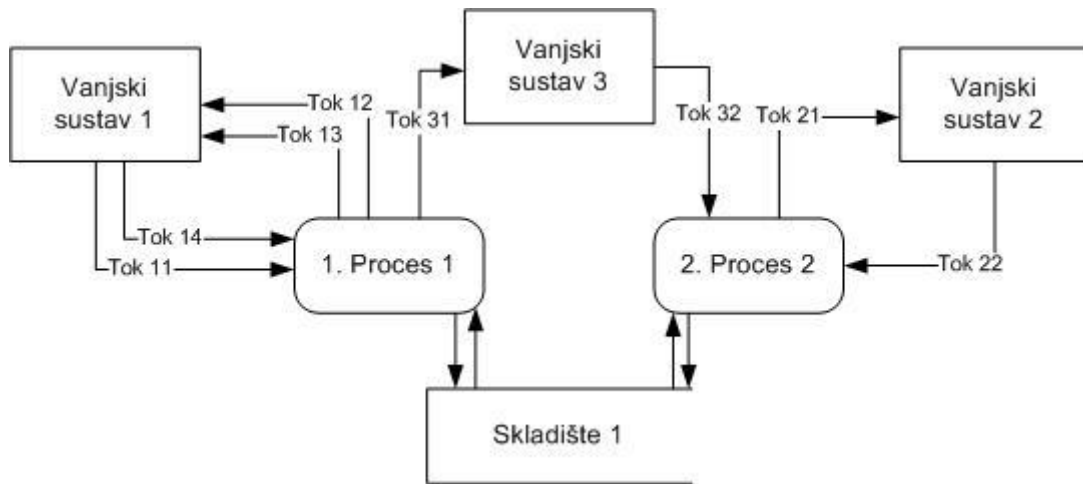


Slika 66: Općeniti primjer dijagrama konteksta (izvor: Autori)

Na prikazanom primjeru (slika 66) vidljiv je općeniti *Informacijski sustav* i okolina u kojoj djeluje. Vidljivo je da razmjenjuje podatke s *Vanjskim sustavom 1*, *Vanjskim sustavom 2* i *Vanjskim sustavom 3*. *Vanjskom sustavu 1* dostavlja podatke pomoću *tokova podataka 12* i *13*, a prihvaća od njega podatke putem *tokova podataka 11* i *14*. *Vanjskom sustavu 2* dostavlja podatke putem *toka 21*, a prihvaća od njega podatke putem *toka 22*. *Vanjskom sustavu 3* dostavlja podatke putem *toka 32*, a prihvaća od njega podatke putem *toka 31*.

**Dijagram prve razine dekompozicije** prikazuje dekompoziciju dijagrama konteksta. Na ovoj razini se prikazuju glavni procesi koji se u promatranom sustavu odvijaju. Proces na ovoj razini nose oznake 1, 2, 3, ...

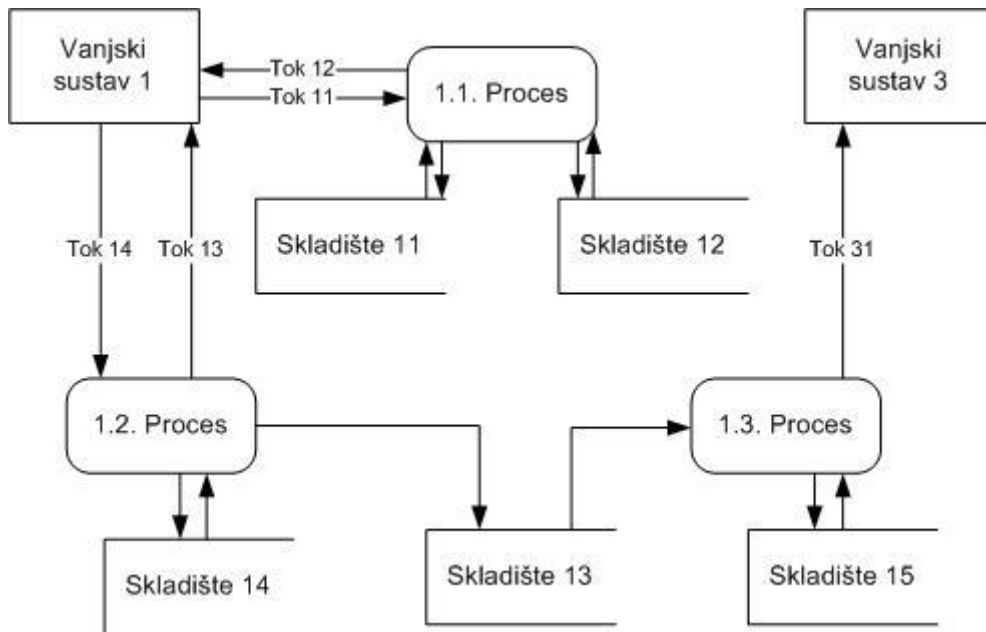




Slika 67: Općeniti primjer dijagrama tokova podataka 1. razine dekompozicije (izvor: Autori)

Na prikazanom primjeru (slika 67) vidljivo je da se *Informacijski sustav* dekomponira na procese *1. Proces 1* i *2. Proces 2*. Dijagram 1. razine dekompozicije prikazuje iste vanjske sustave i iste tokove podataka koji su opisani na 0. razini dekompozicije (na dijagramu konteksta). Razlika je u tome što se na 1. razini vidi u koje, odnosno iz kojih procesa tokovi podataka ulaze, odnosno izlaze. Na dijagramu 1. razine prikazuju se i spremišta podataka u koja procesi sustava upisuju podatke, odnosno iz kojih čitaju željene podatke. Na prikazanom primjeru vidljivo je kako *Proces 1* i *Proces 2* upisuju i čitaju podatke iz *Skladišta 1*.

**Dijagrami nižih razina dekompozicije** prikazuju dekompoziciju određenog složenog procesa. Procesi na ovoj razini nose oznaku nadređenog procesa. Npr. podproces *Procesa 1* nose oznake: *1.1*, *1.2*, *1.3...*



Slika 68: Općeniti primjer dijagrama tokova podataka 2. razine dekompozicije (izvor: Autori)

Na prikazanom primjeru (slika 68) vidljivo je da je složeni proces *Proces 1* dekomponiran na svoje *podproces* 1.1, 1.2 i 1.3. Na ovoj razini dijagrama vidljivi su samo oni tokovi podataka i oni vanjski sustavi koji su potrebni za odvijanje navedenih podprocesa. Također, prikazuju se i spremišta podataka u koje podprocesi pohranjuju, odnosno iz kojih čitaju podatke.

Dekompozicija je došla do najniže razine ako je svaki proces fundamentalan. *Fundamentalan proces* je onaj koji se sastoji od jednog ulaznog toka iz vanjskog sustava i izlaznog toka prema spremištu, ili jednog ulaznog toka iz spremišta i izlaznog toka prema vanjskom sustavu. Može se dogoditi da se kod izrade najnižih razina DTP-a dođe do spoznaje da neki proces nije fundamentalan i može ga se dekomponirati pa je zbog toga potrebno prilagoditi (proširiti, produbiti) dijagram dekompozicije procesa.

## 7. Zadaci za vježbu i njihova rješenja

- Modeliranje podataka:
  - Identifikacija relevantnih objekata iz opisa sustava
  - Izrada EVA modela
  - Opis EVA modela
  - Prevođenje EVA modela u Relacijski model
  - Normalizacija relacijske sheme
- Modeliranje procesa
  - Izrada dijagrama dekompozicije
  - Izrada dijagrama tokova podataka (DTP)

### 7.1. Identifikacija relevantnih objekata iz opisa poslovnih sustava

#### Zadatak 1:

Identificirati relevantne objekte za izradu EVA modela podataka.

- Informacijski sustav "Tvrtke d.o.o". Podsustav: Praćenje radnih sati.
- Promatrani podsustav se bavi praćenjem radnih sati koje je zaposlenik neke tvrtke odradio:
- Zaposlenik odrađuje radne sate. Svaki radni sat je odradio je netko od zaposlenika. Zaposlenik se opisuje imenom, prezimenom, adresom i mjestom stanovanja, dok se radni sati opisuju datumom kada su odrađeni, brojem odrađenih sati i opisom odrađenog posla.

### Rješenje 1:

**ENTITETI:** Zaposlenik, Radni\_Sat

#### **ATRIBUTI ENTITETA:**

- **ZAPOSLENIK:** Ime, Prezime, Adresa, Mjesto\_stanovanja, **Sifra\_zaposlenika** (PK)
- **RADNI\_SAT:** Dat\_odrade, Br\_odradjenih\_rs, Opis\_odradjenog\_posla, **Sifra\_rs** (PK), **Sifra\_zaposlenika**(FK1)

#### **VEZE MEĐU ENTITETIMA:**

- Zaposlenik odraduje *najmanje* nula, a *najviše* više radnih sati.
- Radni\_sat je odradio je *najmanje* jedan i *najviše* jedan zaposlenik (Radni\_sat je odradio *jedan i samo jedan* zaposlenik).

### Zadatak 2:

Identificirati relevantne objekte za izradu EVA modela podataka.

- Poslovni podsustav praćenja zaduženja knjiga u biblioteci.
- Promatrani podsustav se bavi evidentiranjem knjiga koje su članovi biblioteke zadužili:
- Student dolazi u biblioteku i odabire knjigu koju želi posuditi (svaka knjiga ima svoj inventarni broj). Zaposlenik u biblioteci otvara zadužnicu u koju se evidentira student koji posuđuje knjigu, knjiga, datum zaduženja i datum isteka zaduženja. Također, u biblioteci se vodi evidencija o svim upisanim studentima u biblioteku (ime, prezime, adresa, mjesto, datum upisa), te o knjigama u biblioteci (autor, naslov, izdavač, godina izdanja).

### Rješenje 2:

**ENTITETI:** Student, Knjiga, Zaduznica

#### **ATRIBUTI ENTITETA:**

- **student:** Ime, Prezime, Adresa, Mjesto, Dat\_upisa, **Mb\_studenta** (PK)
- **Knjiga:** Autor, Naslov, Izdavač, Godina\_izdanja, **Inventarni\_broj** (PK)

- **Zadužnica:** Mb\_studenta (FK1), Inventarni\_broj (FK2), Dat\_zaduzenja, Dat\_isteka\_zaduzenja, Sifra\_zaduznice (PK)

#### **VEZE MEĐU ENTITETIMA:**

- Za Studenta se otvara najmanje nula, a najviše više Zaduznica. Zaduznica se otvara za jednog i samo jednog Studenta.
- Zaduznica zaduzuje jednu i samo jednu Knjigu. Knjiga je zaduzena na najmanje nula, a najviše više Zaduznica.

#### **Zadatak 3:**

Identificirati relevantne objekte za izradu EVA modela podataka

- Poslovni podsustav zaprimanja narudžbi u kantini.
- Promatrani podsustav se bavi evidentiranjem otvorenih narudžbi koje su konobari u kantini zaprimili:
- Konobari zaprimaju narudžbe u kojima se vodi evidencija koji artikli su naručeni, njihova količina, datum narudžbe, konobar koji ih je zaprimio, te stol koji ih je naručio. Također se vodi evidencija o artiklima koji su u ponudi (naziv artikla, jedinična cijena), stolovima (broj stola), te konobarima (ime, prezime).

#### **Rješenje 3:**

**ENTITETI:** Konobar, Artikl, Stol, Narudzba

#### **ATRIBUTI ENTITETA:**

- **KONOBAR:** Ime, Prezime, Sifra\_konobara (PK)
- **ARTIKL:** Naziv\_artikla, Jed\_cijena, Sifra\_artikla (PK)
- **STOL:** Broj\_stola (PK)
- **NARUDŽBA:** Sifra\_artikla (FK1), Kolicina\_artikla, Dat\_narudzbe, Sifra\_konobara (FK2), Broj\_stola (FK3), Sifra\_narudzbe (PK)

### **VEZE MEĐU ENTITETIMA:**

- Konobar zaprima nula ili više Narudžbi. Narudžbu je zaprimio jedan i samo jedan Konobar.
- Narudžba je zaprimljena s jednog i samo jednog Stola. Sa stola je zaprimljeno nula ili više Narudžbi.
- Artikl je naručen na nula ili više Narudžbi. Narudžba naručuje jedan ili više Artikala.

## **7.2. Izrada EVA modela na temelju opisa poslovnih podsustava**

### **Zadatak 4:**

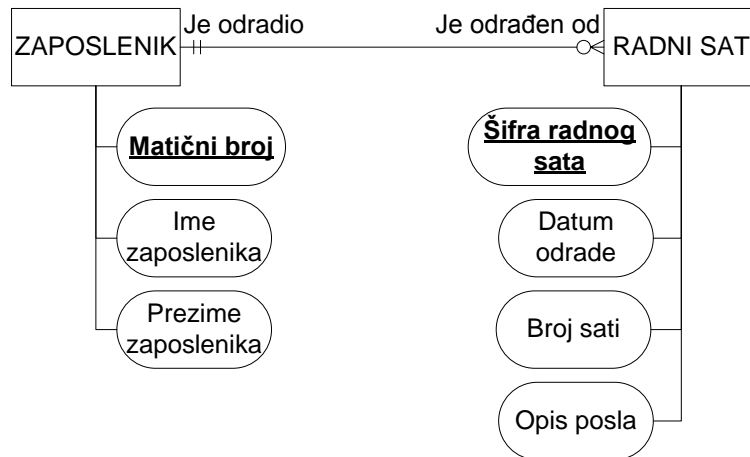
Nacrtati EVA model poslovnog podsustava praćenja radnih sati uz pomoć Martinove i Chenove notacije.

#### *Opis sustava:*

Podsustav praćenje radnih sati evidentira radne sate koje je zaposlenik odradio. Zaposlenik odrađuje radne sate. Svaki radni sat je odrađen od nekog zaposlenika. Zaposlenik se opisuje s matičnim brojem, imenom i prezimenom, dok se radni sati opisuju šifrom radnog sata, datumom, brojem sati i opisom posla.

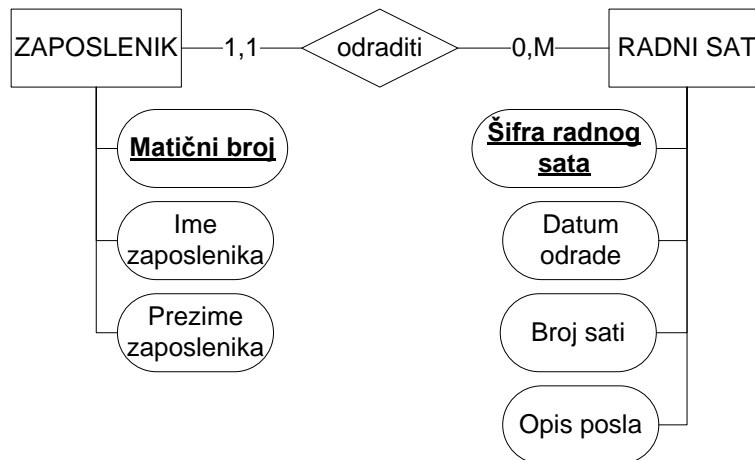
#### Rješenje 4:

Martinova notacija:



Slika 69: Rješenje zadatka 4 u Martinovoj notaciji (izvor: Autori)

Chenova notacija:



Slika 70: Rješenje zadatka 4 u Chenovoj notaciji (izvor: Autori)

### Zadatak 5:

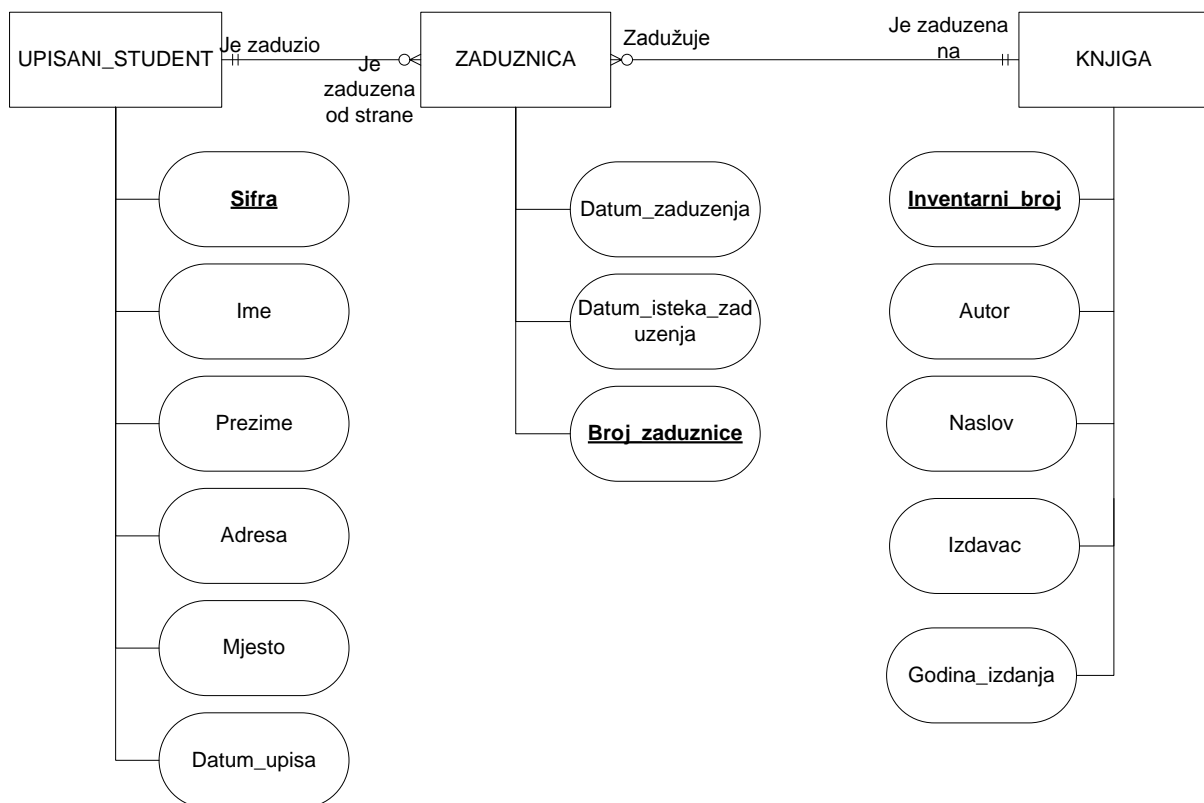
Nacrtati EVA model poslovnog podsustava praćenja zaduženja knjiga u knjižnici uz pomoć Martinove i Chenove notacije.

*Opis sustava:*

Student dolazi u knjižnicu i odabire knjigu koju želi posuditi (svaka knjiga ima svoj inventarni broj). Zaposlenik knjižnice otvara zadužnicu u koju evidentira studenta, knjigu, datum zaduženja i datum isteka zaduženja. Da bi to bilo moguće, u knjižnici se vodi evidencija o upisanim studentima u knjižnicu (ime, prezime, adresa, grad, datum upisa), te o knjigama koje knjižnica posjeduje (autor, naslov, izdavač, godina izdanja).

### Rješenje 5:

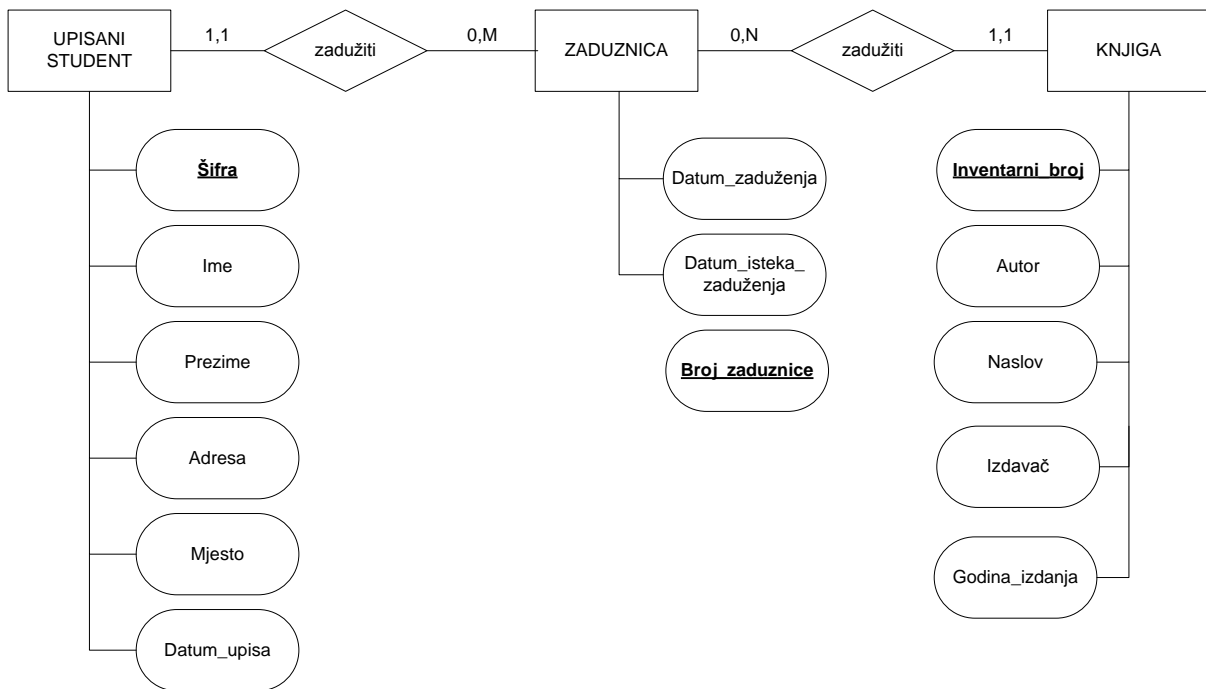
Martinova notacija:



Slika 71: Rješenje zadatka 5 u Martinovoj notaciji (izvor: Autori)



Chenova notacija:



Slika 72: Rješenje zadatka 5 u Chenovoj notaciji (izvor:Autori)

### Zadatak 6:

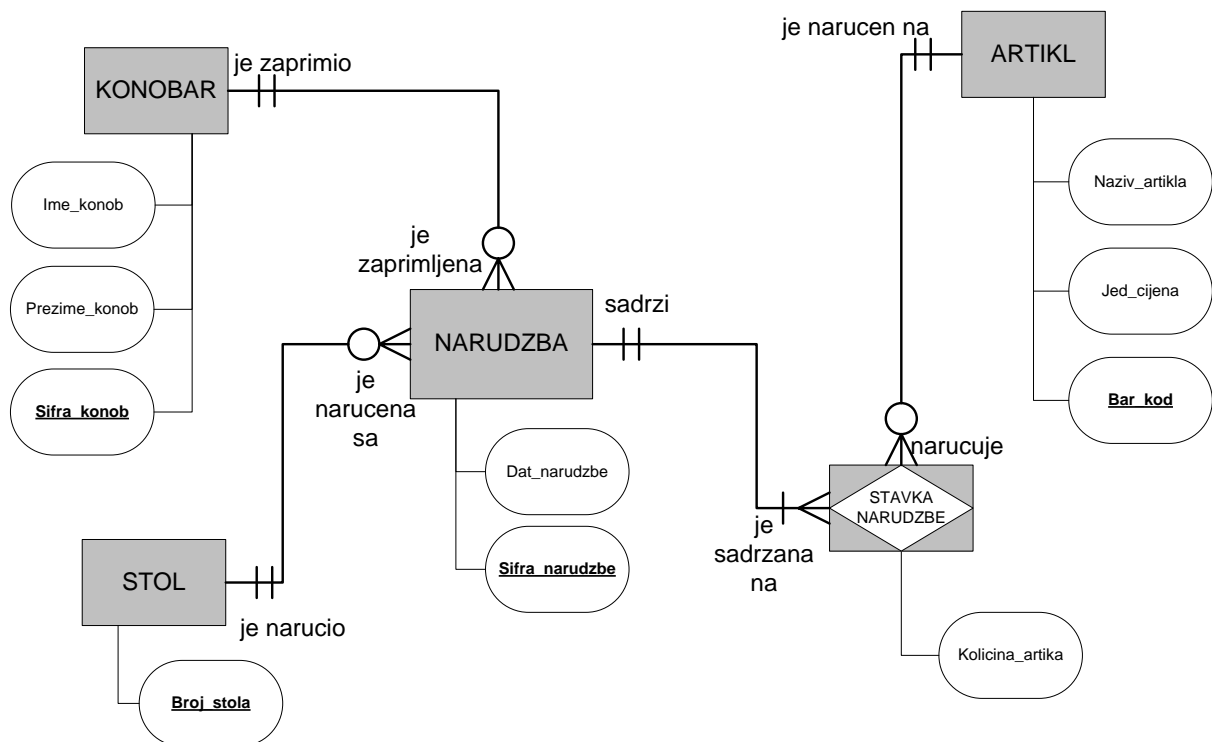
Nacrtati EVA model poslovnog podsustava praćenja zaduženja knjiga u knjižnici uz pomoć Martinove notacije.

*Opis sustava:*

Konobar zaprima narudžbu te se vodi evidencija koji artikli su naručeni, njihova količina, datum narudžbe, konobar te stol koji ih je naručio. Vodi se evidencija o artiklima (naziv artikla, jedinična cijena), stolovima (broj stola), te konobarima (ime, prezime).

## Rješenje 6:

Martinova notacija:



Slika 73: Rješenje zadatka 6 u Martinovoj notaciji (izvor: Autori)

Objašnjenje rješenja: Veza između tipova entiteta NARUDZBA i ARTIKL je sljedeća: ARTIKL je naručen na *nula ili više* (0,M) NARUDŽBI. NARUDŽBA naručuje *jedan ili više* (1,M) ARTIKALA. Veze s brojnostima M:M po gornjim granicama na EVA modelu su zabranjene (razlog tome u navedenom primjeru je što ne možemo znati koji od *više* naručenih ARTIKALA se nalazi na kojoj od *više* otvorenih NARUDŽBI). Kako bi se takva veza izbjegla, uvodi se novi *agregirani entitet* koji ju dijeli na dvije veze i time rješava problem M:M brojnosti po gornjim granicama. Taj agregirani entitet simbolizira relaciju u kojoj će biti moguće točno pročitati koji od više naručenih artikala se nalazi na kojoj od više otvorenih narudžbi).

### Zadatak 7:

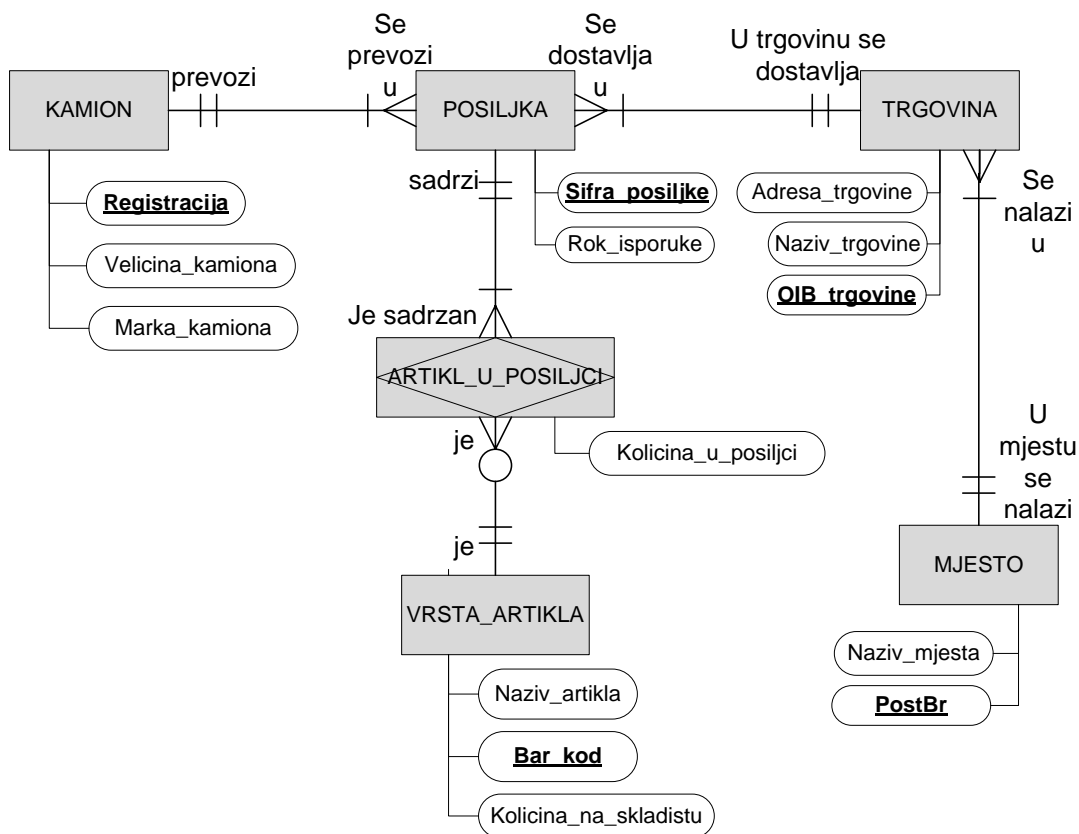
Nacrtati EVA model podataka podsustava distribucije pošiljki uz pomoć Martinove notacije. U zadatku nisu određeni atributi tipova entiteta. Attribute odrediti sami.

#### Opis sustava:

Trgovačko poduzeće posjeduje svoje kamione s kojima distribuira pošiljke iz centralnog skladišta u svoje trgovine. Jedna pošiljka se može sastojati od više različitih artikala. U cilju racionalizacije troškova transporta, jednim kamionom prevoze se pošiljke za više trgovina.

### Rješenje 7:

Martinova notacija:



Slika 74: Rješenje zadatka 7 u Martinovoj notaciji (izvor: Autori)

Objašnjenje rješenja: Prilikom određivanja atributa tipa entiteta TRGOVINA, može se zaključiti, uz pretpostavku da se trgovine nalaze u gradovima Hrvatske, da nije dovoljno samo navesti ADRESU kao atribut koji opisuje gdje se trgovina nalazi, već je potrebno i MJESTO. Mjesto kao takvo se ne može identificirati preko svog naziva (u Hr. postoji više mjesta istog naziva) pa je MJESTO izdvojeno kao

zaseban tip entiteta koji se onda vrlo jednostavno može identificirati Poštanskim brojem. Na taj način je ujedno izbjegnuta situacija pojave redundantnih podataka (Naziv mjesta) koji bi se pojavljivali Ako imamo više trgovina koje se nalaze u istom Mjestu.

### **Zadatak 8:**

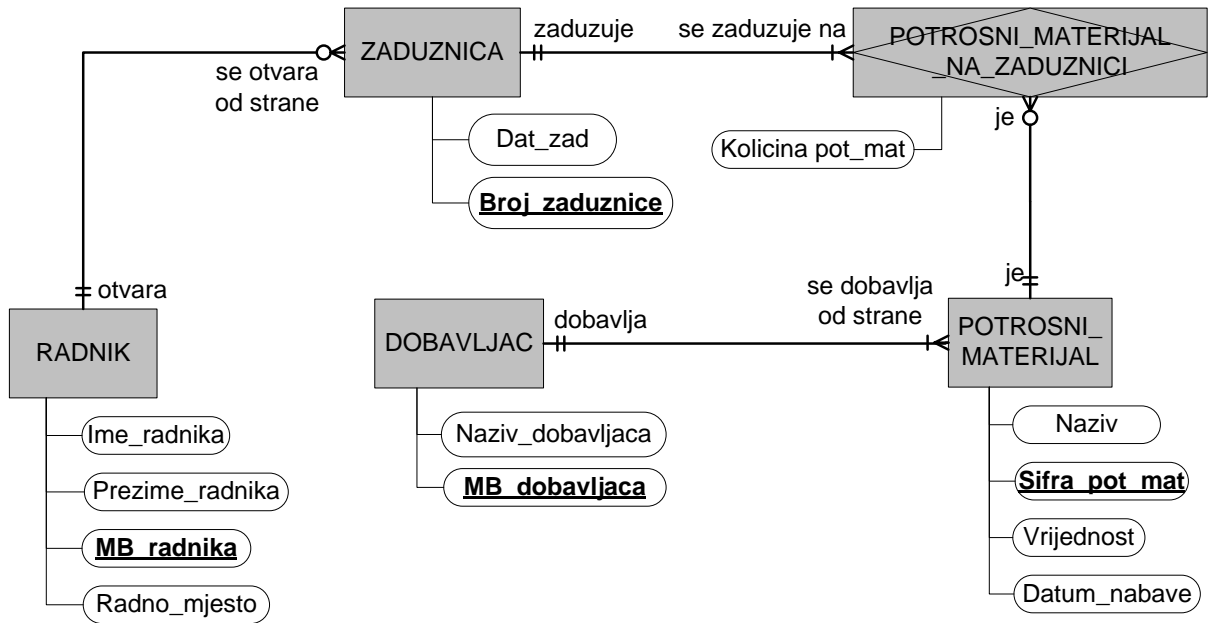
Nacrtati EVA model podataka podsustava praćenja zaduženja potrošnog materijala u nekoj tvrtki uz pomoć Martinove i Chenove notacije.

#### *Opis sustava:*

U opisu sustava stoji da želimo voditi podatke o radniku i potrošnom materijalu s kojim se radnik služi tijekom svog rada. Za svakog radnika se vode sljedeći podaci: ime, prezime, matični broj, naziv radnog mjesta. Za potrošni materijal kojim se radnik služi vodimo sljedeće podatke: naziv materijala, šifru potrošnog materijala, vrijednost i datum nabave. Potrošni materijal nabavlja se od dobavljača koji ima svoj matični broj i naziv. Jedna vrsta potrošnog materijala se nabavlja od jednog dobavljača. Svaki radnik sam zadužuje materijal koji mu je potreban. Zadužnica se sastoji od sljedećih podataka: datum zaduženja, broj zadužnice. Jednom zadužnicom radnik može zadužiti više vrsti potrošnog materijala.

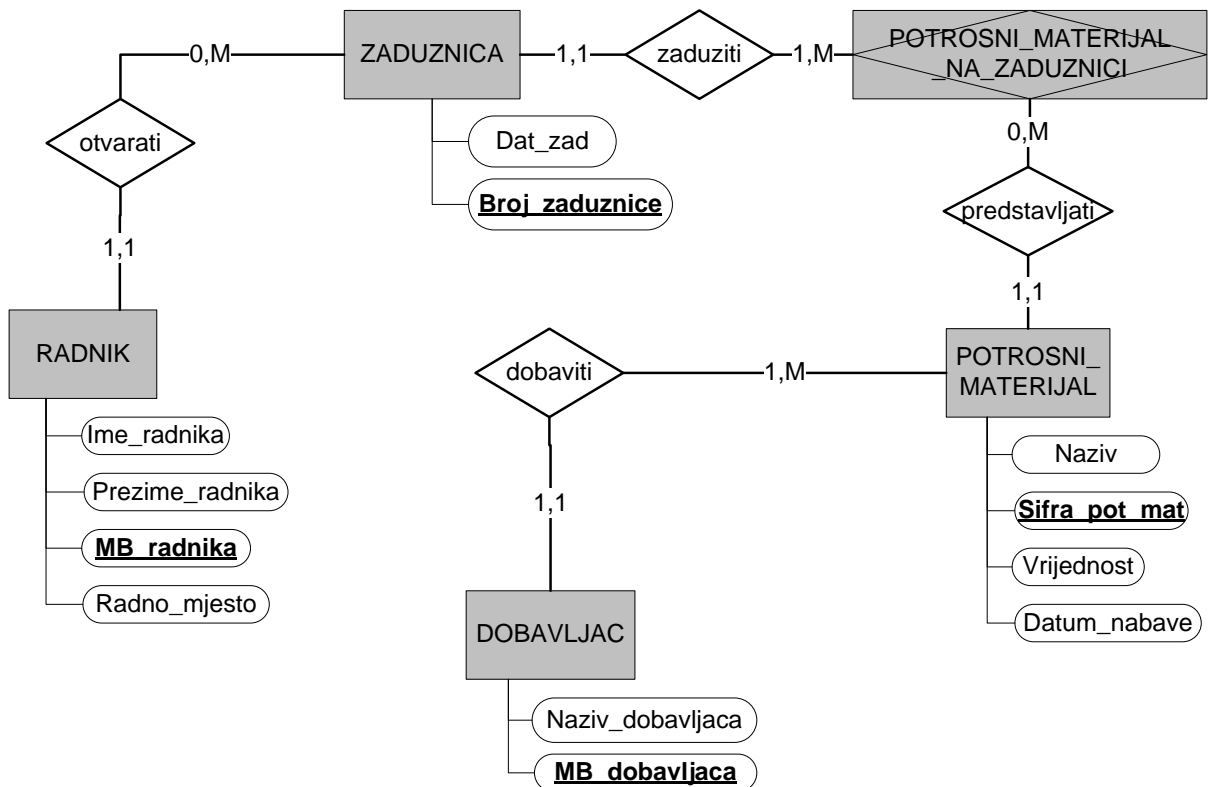
### Rješenje 8:

Martinova notacija:



Slika 75: Rješenje zadatka 8 u Martinovoj notaciji (izvor: Autori)

Chenova notacija:



Slika 76: Rješenje zadatka 8 u Chenovoj notaciji (izvor: Autori)

### Zadatak 9:

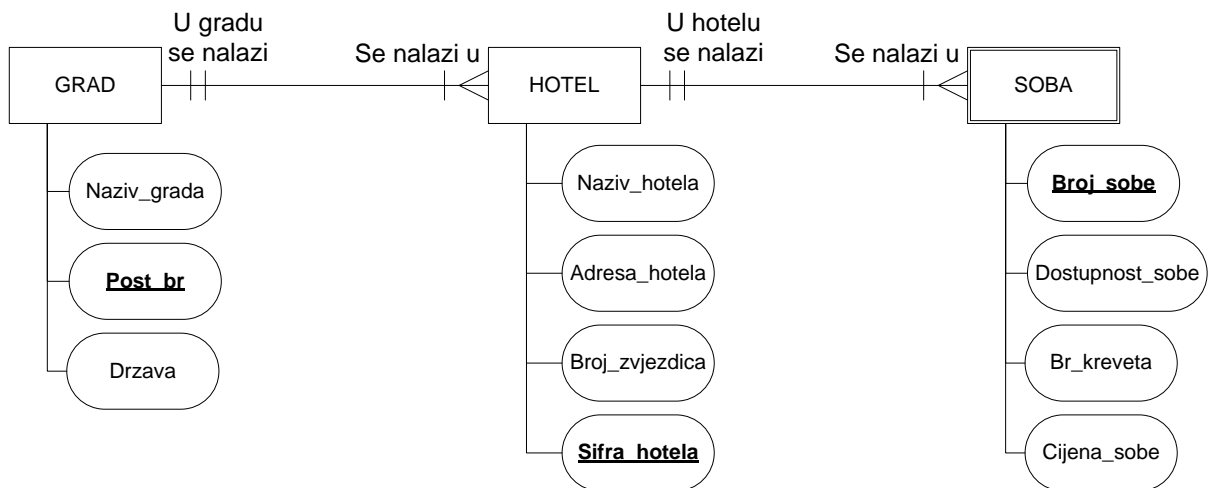
Nacrtati EVA model podataka podsustava za evidentiranje hotela uz pomoć Martinove i Chenove notacije.

*Opis sustava:*

U određenom gradu nalazi se određen broj hotela. Svaki hotel sastoji se od određenog broja soba koje imaju svoju cijenu. Svaka soba ima svoj broj. Osim toga, vode se podaci o dostupnosti sobe i o broju kreveta unutar nje. Za svaki hotel se vode podaci o njegovom nazivu, broju zvjezdica koji ima i o adresi. Podaci koji opisuju grad u kojem se hotel nalazi su: poštanski broj, naziv grada i država.

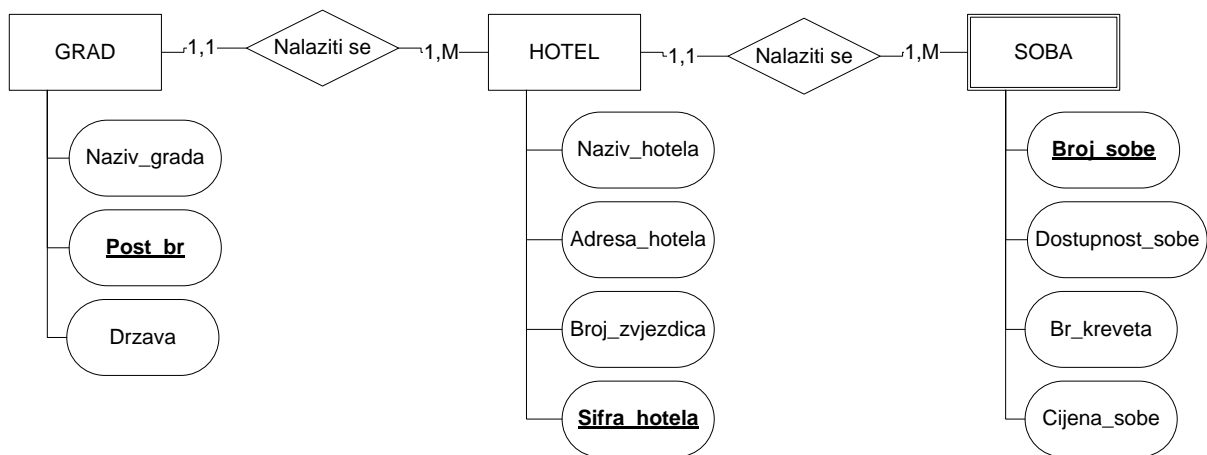
### Rješenje 9:

Martinova notacija:



Slika 77: Rješenje zadatka 9 u Martinovoj notaciji (izvor: Autori)

Chenova notacija:



Slika 78: Rješenje zadatka 9 u Chenovoj notaciji (izvor: Autori)

#### Zadatak 10:

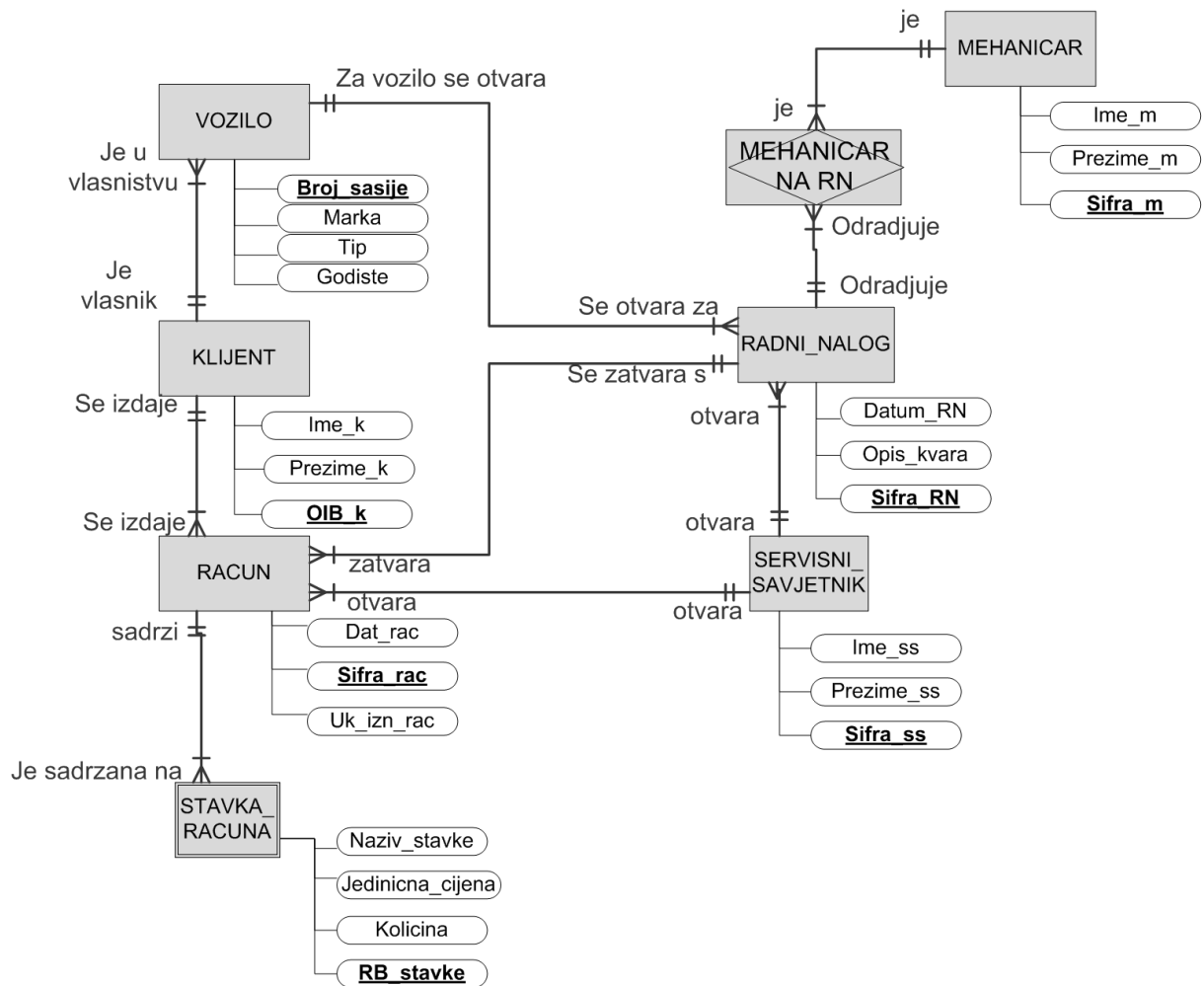
Nacrtati EVA model podataka podsustava autoservisa uz pomoć Martinove notacije.

Opis sustava:

U nekom autoservisu vodi se evidencija o izvršenim servisima vozila. Proces servisiranja vozila počinje radnim nalogom kojeg otvara servisni savjetnik po dolasku klijenta koji traži servis. Radni nalog sadrži datum otvaranja, opis kvara vozila i šifru radnog naloga. Svaki radni nalog se dodjeljuje određenim mehaničarima. Mehaničari obavljaju poslove koji su im dodijeljeni radnim nalogima. Za svakog mehaničara se vode podaci o njegovu imenu, prezimenu i šifri mehaničara. Radni nalog se zatvara računom. Račun izdaje servisni savjetnik. Na računu se bilježe datum računa, šifra računa i ukupni iznos računa. Svaka stavka računa opisuje se nazivom, jediničnom cijenom, količinom i rednim brojem stavke. Račun se izdaje klijentu. O klijentu se vode sljedeći podaci: ime, prezime, OIB. Za njegovo vozilo vode se podaci: broj šasije vozila, marka, tip, godište. O servisnom savjetniku se vode sljedeći podaci: ime, prezime, šifra servisnog savjetnika.

## Rješenje 10:

Martinova notacija:



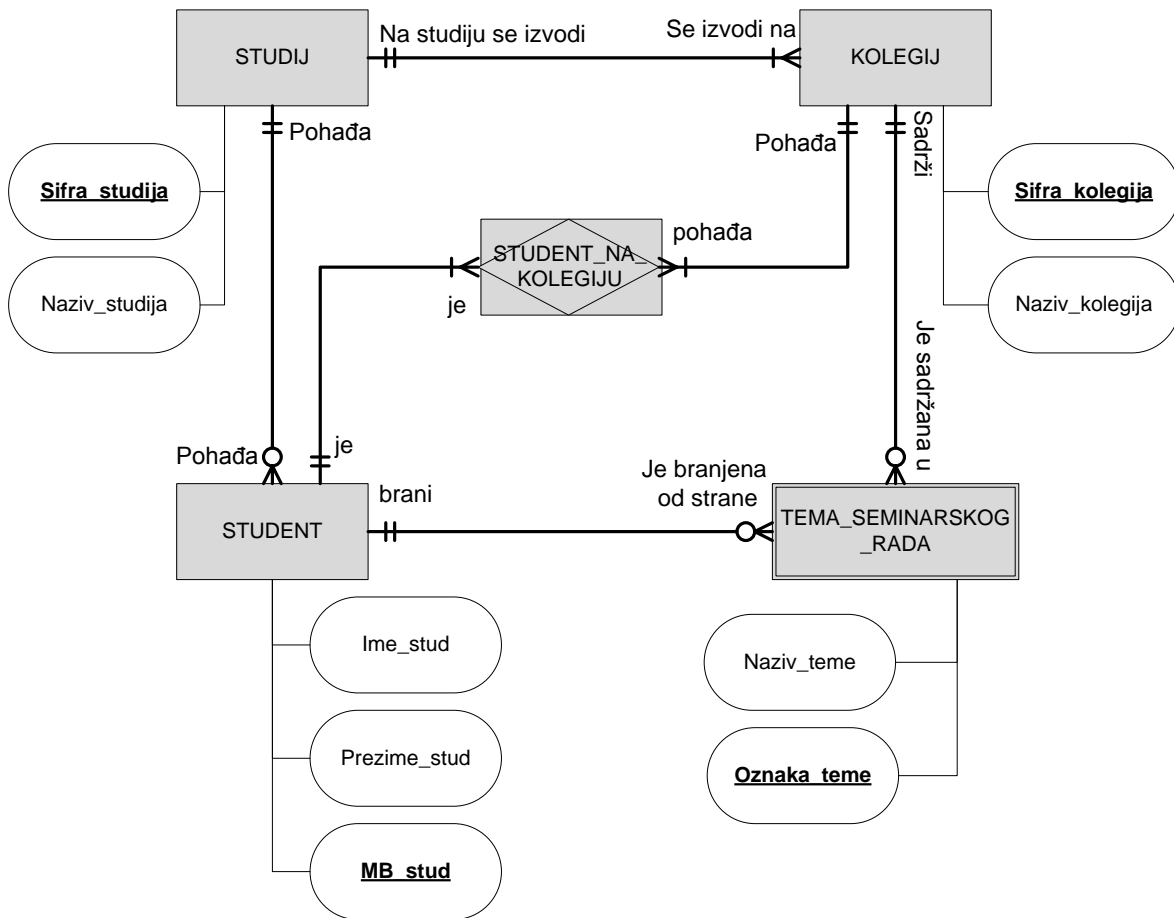
Slika 79: Rješenje zadatka 10 u Martinovoj notaciji (izvor: Autori)



### 7.3. Tekstualni opis prikazanog EVA modela podataka

#### Zadatak 11:

Detaljno opisati riječima prikazani podsustav fakulteta (entitet *Tema* je slab u odnosu na entitet *Kolegij* kako bi se logički odijelile teme s različitih kolegija).



Slika 80: EVA model podataka podsustava fakulteta – zadatak 11 (izvor: Autori)

#### Rješenje 11:

*Prikazani model predstavlja:* Podsustav za vođenje evidencije o temama seminarских radova koje studenti odabiru iz pojedinih kolegija.

**ENTITETI:** Studij, Kolegij, Student, Student\_na\_kolegiju (*agregacija*), Tema\_seminarskog\_rada (*slabi entitet*)

*ATRIBUTI ENTITETA I NJIHOVI KLJUČEVI:*

- Studij (**Sifra studija**(PK), Naziv\_studija)
- Kolegij (**Sifra kolegija**(PK), Naziv\_kolegija, Sifra\_studija(FK1))
- Student (Ime\_stud, Prezime\_stud, **MB stud**(PK), Sifra\_studija(FK1))
- Student\_na\_kolegiju (**MB stud**(PK,FK1), **Sifra kolegija**(PK,FK2))
- Tema\_seminarskog\_rada (Naziv\_teme, **Oznaka teme**(PK), **Sifra kolegija**(PK,FK1), **MB stud**(FK2))

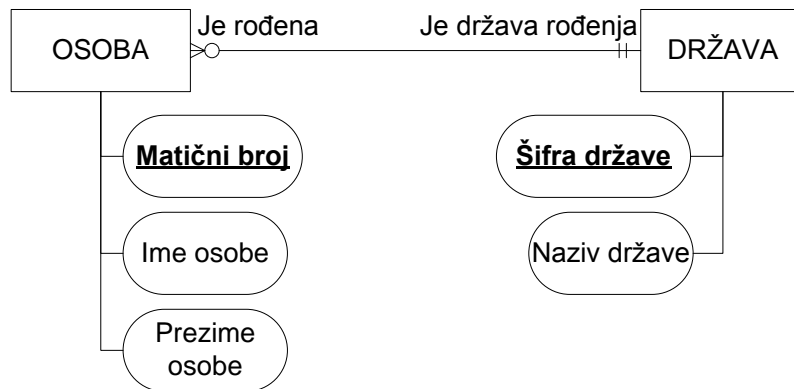
*VEZE MEĐU ENTITETIMA:*

- STUDIJ se sastoji od 1 ili više KOLEGIJA. KOLEGIJ se izvodi na jednom i samo jednom STUDIJU.
- STUDIJ pohađa nula ili više STUDENATA. STUDENT pohađa jedan i samo jedan STUDIJ.
- STUDENT je student na jednom ili više KOLEGIJA. STUDENT NA KOLEGIJU je jedan i samo jedan STUDENT.
- STUDENT na kolegiju je sa jednog i samo jednog KOLEGIJA. KOLEGIJ pohađa jedan ili više STUDENATA NA KOLEGIJU.
- STUDENT brani nula ili VIŠE TEMA SEMINARSKOG RADA. TEMA SEMINARSKOG RADA pripada jednom i samo jednom STUDENTU.
- TEMA SEMINARSKOG RADA je iz jednog i samo jednog KOLEGIJA. KOLEGIJ ima nula ili više TEMA SEMINARSKOG RADA.

## 7.4. Prevođenje EVA modela podataka u relacijski model podataka

### Zadatak 12:

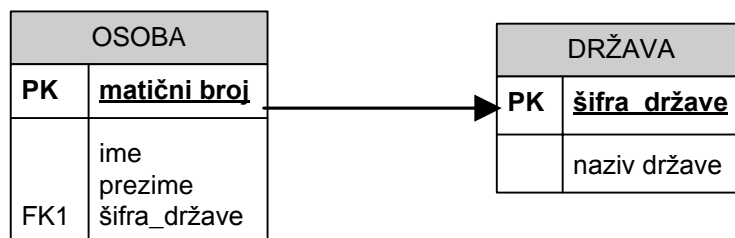
Prevesti prikazani EVA model podataka u relacijski model podataka i napisati relacijske sheme.



Slika 81: EVA model podataka nekog podsustava – zadatak 12 (izvor: Autori)

### Rješenje 12:

Relacijski model podataka:



Slika 82: Rješenje zadatka 12 - relacijski model (izvor: Autori)

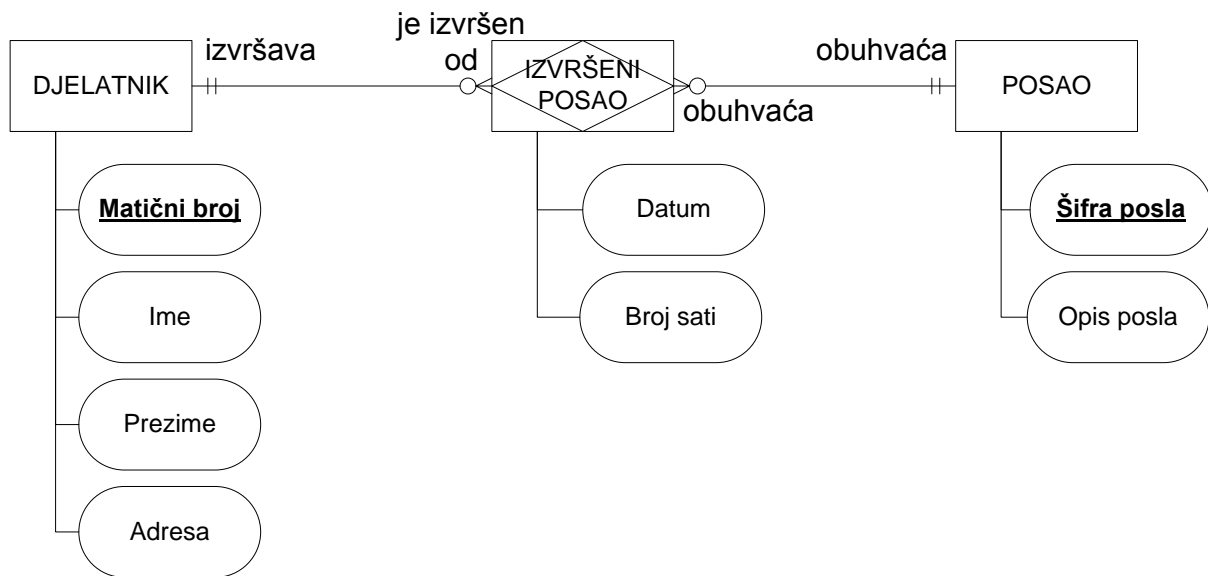
Relacijske sheme:

OSOBA (matični broj (PK), ime, prezime, šifra\_države (FK1))

DRŽAVA (šifra države(PK), naziv države)

### Zadatak 13:

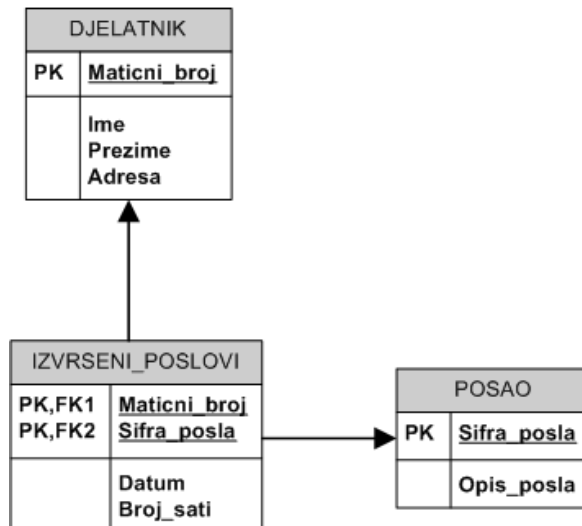
Prevesti prikazani EVA model podataka u relacijski model podataka i napisati relacijske sheme.



Slika 83: EVA model podataka nekog podsustava – zadatak 13 (izvor: Autori)

### Rješenje 13:

Relacijski model podataka:



Slika 84: Rješenje zadatka 13 - relacijski model (izvor: Autori)

Relacijske sheme:

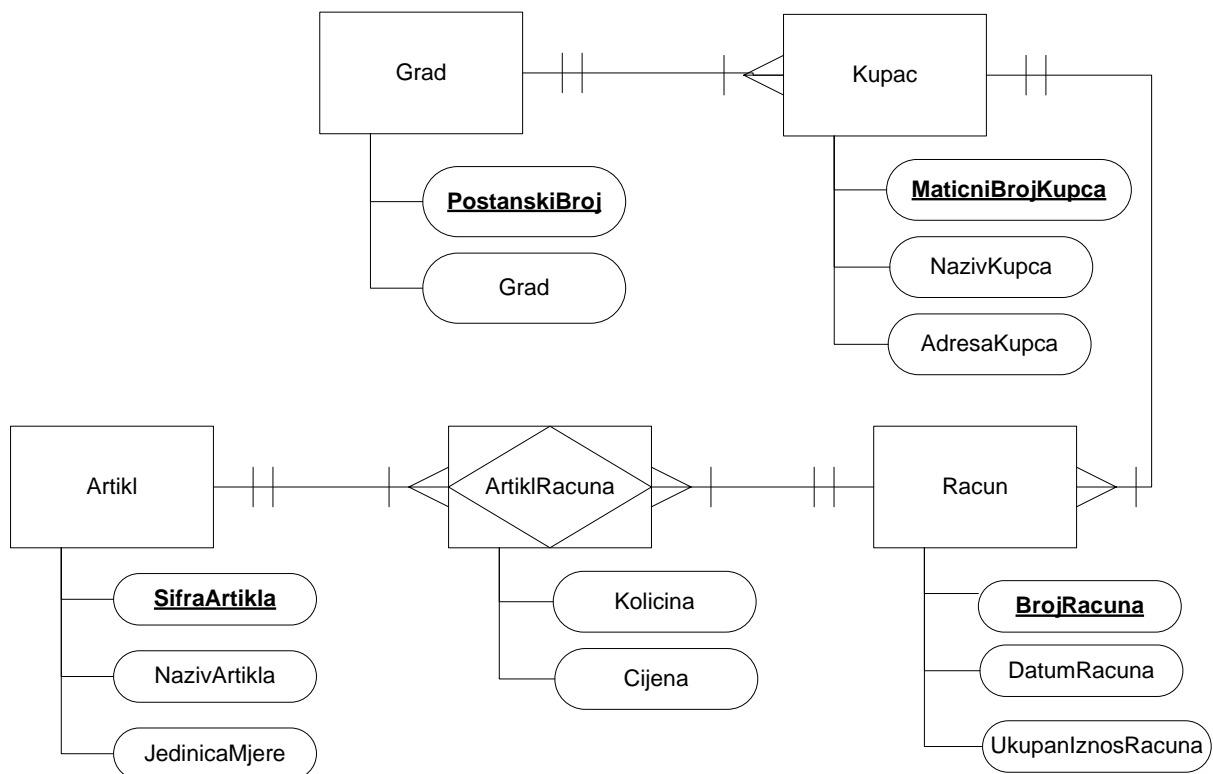
DJELATNIK (**Maticni\_broj** (PK), Ime, Prezime, Adresa)

IZVRSENI\_POSLOVI (**Maticni\_broj**(PK,FK1), **Sifra\_posla** (PK, FK2), Datum, Broj\_sati)

POSALO (**Sifra\_posla** (PK), Opis\_posla)

#### Zadatak 14:

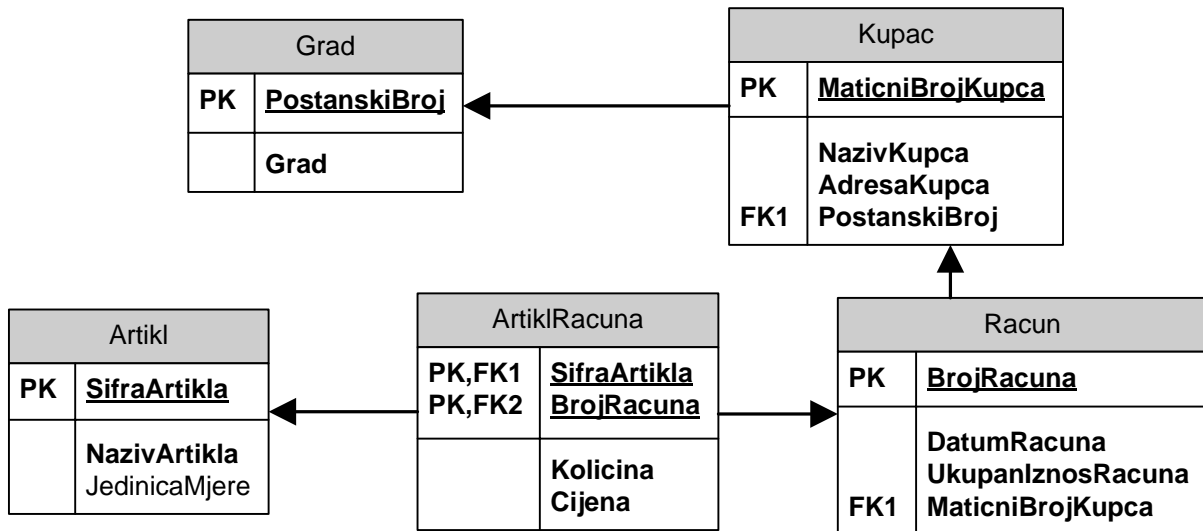
Prevesti prikazani EVA model podataka u relacijski model podataka i napisati relacijske sheme.



Slika 85: EVA model podataka nekog podsustava – zadatak 14 (izvor: Autori)

## Rješenje 14:

Relacijski model podataka:



Slika 86: Rješenje zadatka 14 - relacijski model (izvor: Autori)

Relacijske sheme:

Grad (PostanskiBroj(PK), Grad)

Kupac (MaticniBrojKupca(PK), NazivKupca, AdresaKupca, PostanskiBroj(FK1))

Racun (BrojRacuna(PK), DatumRacuna, UkupanIznosRacuna, MaticniBrojKupca(FK1))

ArtiklRacuna (SifraArtikla(PK,FK1), BrojRacuna(PK,FK2), Kolicina, Cijena)

Artikl (SifraArtikla(PK), NazivArtikla, JedinicaMjere)

## 7.5. Modeliranje procesa

### Zadatak 15:

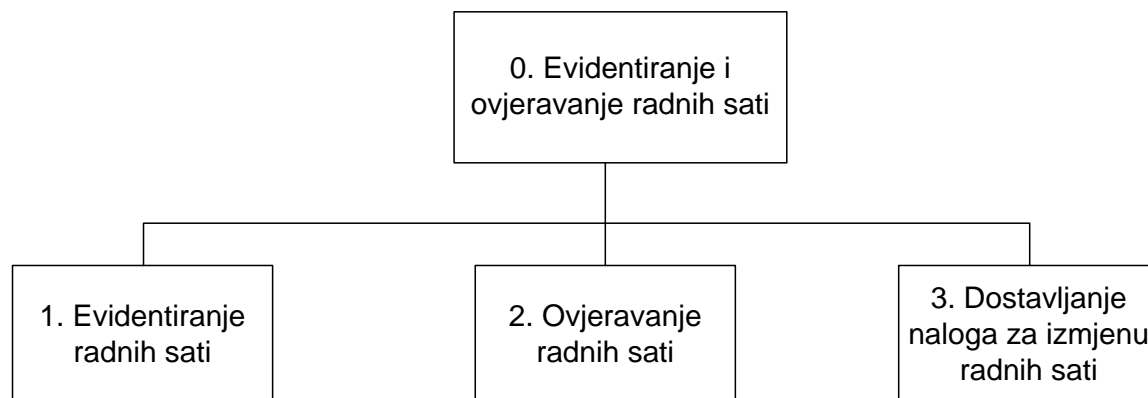
Nacrtati dijagram dekompozicije, dijagram konteksta i dijagram 1. razine koji opisuju sljedeći sustav:

*Poslovni podsustav za evidentiranje i ovjeravanje radnih sati*

U poslovnom podsustavu se vrši evidentiranje i ovjeravanje radnih sati. Obavljaju se sljedeći poslovi: evidentiranje radnih sati, ovjeravanje radnih sati i dostavljanje naloga za izmjenu. Evidentiranje radnih sati se vrši temeljem podataka o odrađenim radnim satima koje dostavlja zaposlenik. Ovjera radnih sati obavlja se temeljem podataka o evidentiranim radnim satima koji se dostavljaju voditelju odjela, te naloga za ovjeru radnih sati kojeg upućuje voditelj odjela. Dostavljanje naloga za izmjenu, temeljem naloga za izmjenu kojeg upućuje voditelj odjela, šalje zaposleniku nalog za izmjenu evidentiranih radnih sati. U tom slučaju zaposlenik mora izvršiti korekciju evidentiranih sati.

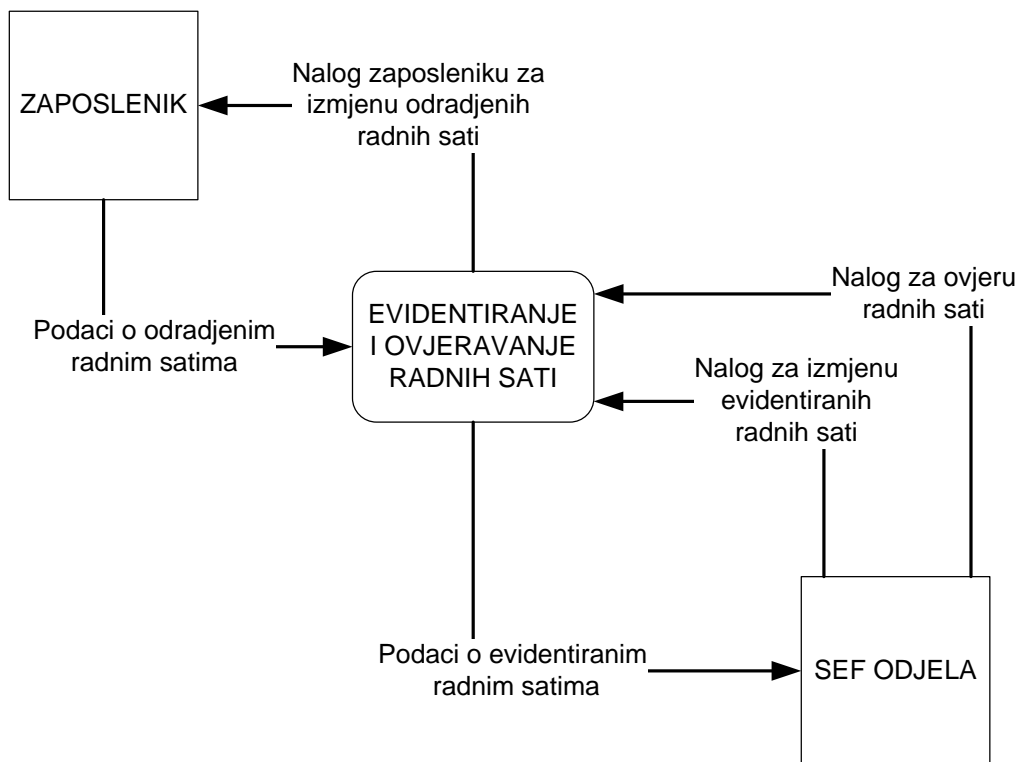
### Rješenje 15:

Dijagram dekompozicije procesa:



*Slika 87: Rješenje zadatka 15 – dijagram dekompozicije procesa (izvor: Autori)*

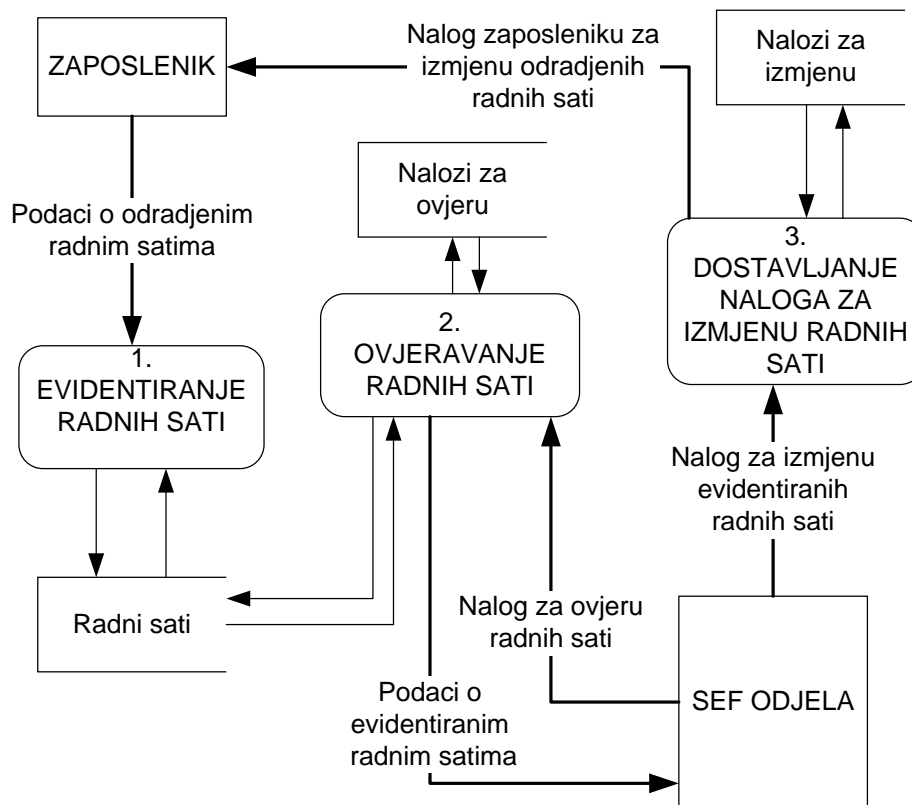
Dijagram konteksta:



Slika 88: Rješenje zadatka 15 – dijagram konteksta (izvor: Autori)



Dijagram 1. razine dekompozicije procesa:



Slika 89: Rješenje zadatka 15 – dijagram 1. razine (izvor: Autori)

### Zadatak 16:

Nacrtati dijagram dekompozicije, dijagram konteksta i dijagram 1. razine koji opisuju sljedeći sustav:

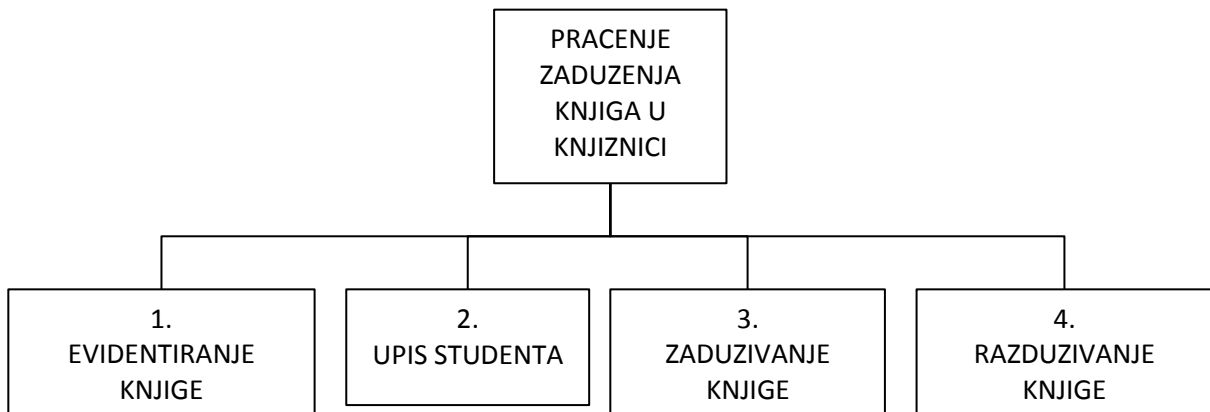
*Poslovni podsustav praćenja zaduženja knjiga u knjižnici*

U poslovnom podsustavu se vrši praćenje zaduženja knjiga. Obavljaju se sljedeći poslovi: evidentiranje knjiga, upis studenata, zaduženje knjiga i razduživanje. Evidentiranje knjige se temelji na podacima o knjizi koji su pristigli iz odjela nabave. Upis studenata se temelji na osnovnim podacima koje dostavlja student. On se u sustav upisuje. Pritom mu se dodjeljuje članski broj te mu se dostavlja članska iskaznica. Zaduženje se temelji na zahtjevu studenta za nekom knjigom. Pritom se i provjerava se da li se student nalazi u evidenciji upisanih studenata. U evidenciji knjiga se provjerava je li knjiga dostupna. Ako knjiga nije dostupna, studentu se šalje informacija o nedostupnosti knjige i o tome kada bi trebala biti dostupna. Ako je knjiga dostupna, ona se predaje studentu, te se u sustavu bilježi da je knjiga zadužena. Razduživanje knjige se temelji na zahtjevu studenta o razduživanju knjige. Nakon

pregleda knjige studentu se dostavlja pozitivno ili negativno razduživanje. Pozitivno razduživanje se evidentira u evidenciji.

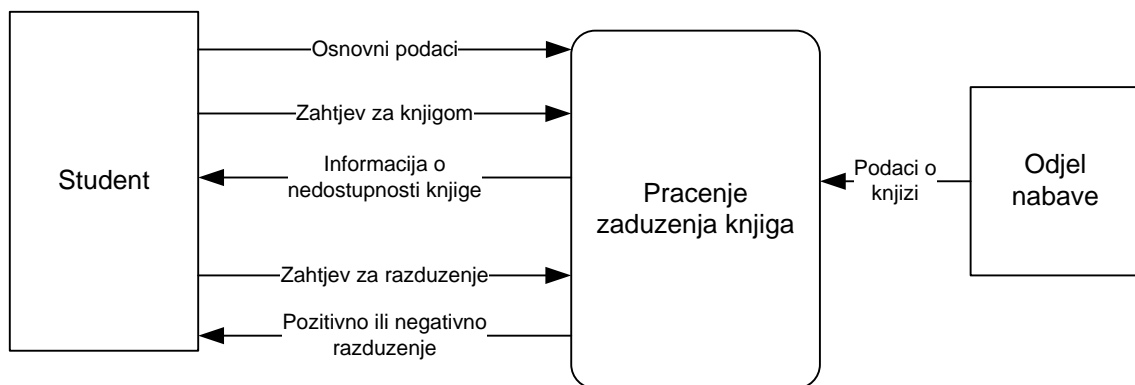
**Rješenje 16:**

Dijagram dekompozicije procesa:



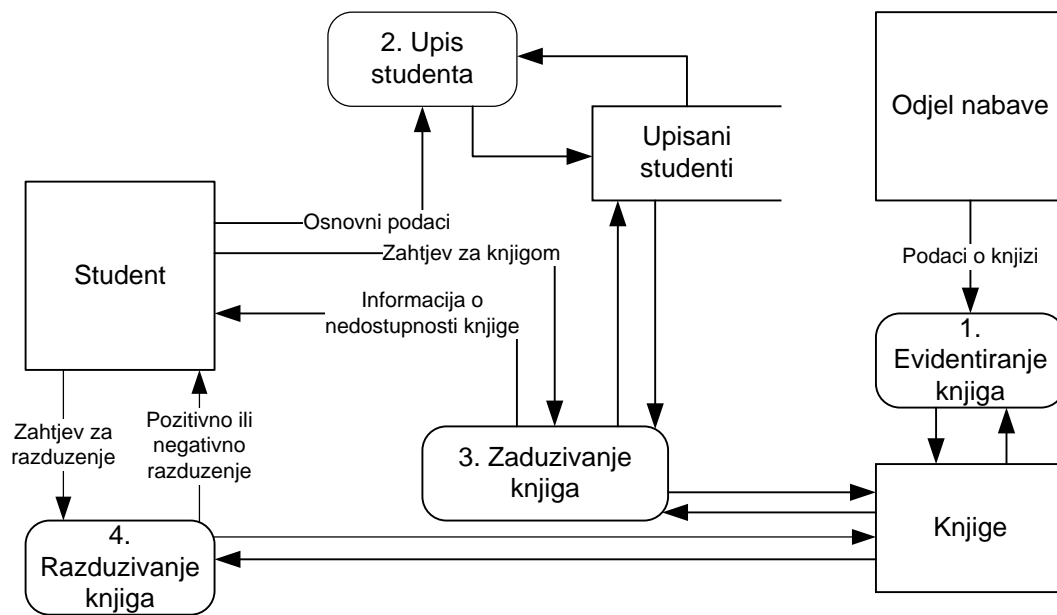
Slika 90: Rješenje zadatka 16 – dijagram dekompozicije procesa (izvor: Autori)

Dijagram konteksta:



Slika 91: Rješenje zadatka 16 – dijagram konteksta (izvor: Autori)

Dijagram 1. razine dekompozicije procesa:



Slika 92: Rješenje zadatka 16 – dijagram 1. razine (izvor: Autori)

### Zadatak 17:

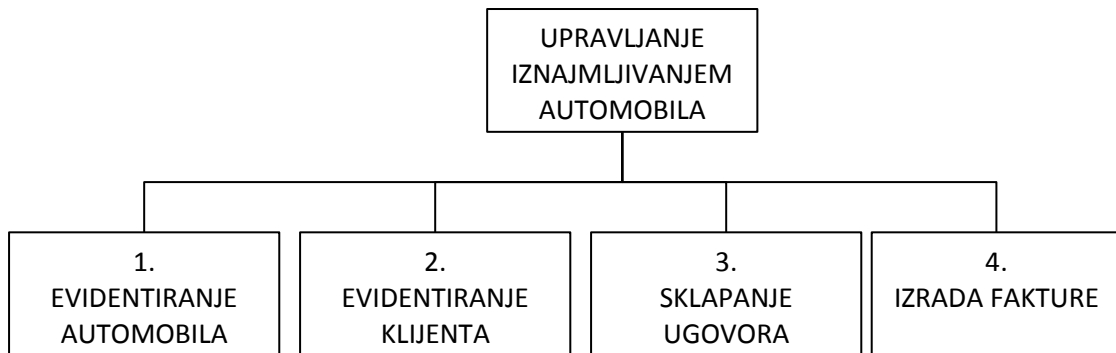
Nacrtati dijagram dekompozicije, dijagram konteksta i dijagram 1. razine koji opisuju sljedeći sustav:

*Poslovni podsustav upravljanja iznajmljivanjem automobila*

U poslovnom podsustavu obavljaju se sljedeći poslovi: evidentiranje automobila, evidentiranje klijenata, sklapanje ugovora, izrada faktura. Evidentiranje automobila obavlja se temeljem podataka o automobilu koje dostavlja služba nabave. Evidentiranje klijenata obavlja se temeljem osnovnih podataka o klijentu koje je dostavio klijent. Sklapanje ugovora započinje zahtjevom za sklapanjem kojeg dostavlja klijent. Potom se odgovornoj osobi šalje ugovor na potpis. Na ugovoru se nalaze podaci o klijentu i automobilu. Potpisani ugovor se vraća, te se dostavlja klijentu. Izrada faktura se temelji na sklopljenim ugovorima. Faktura se dostavlja odgovornoj osobi na provjeru. Odgovorna osoba vraća pozitivno ili negativno odobrenje za izdavanje fakture. Odobrene fakture se evidentiraju u sustavu, te se dostavljaju klijentu. Neodobrene fakture se također evidentiraju u sustavu, te se koriste prilikom ponovne izrade faktura.

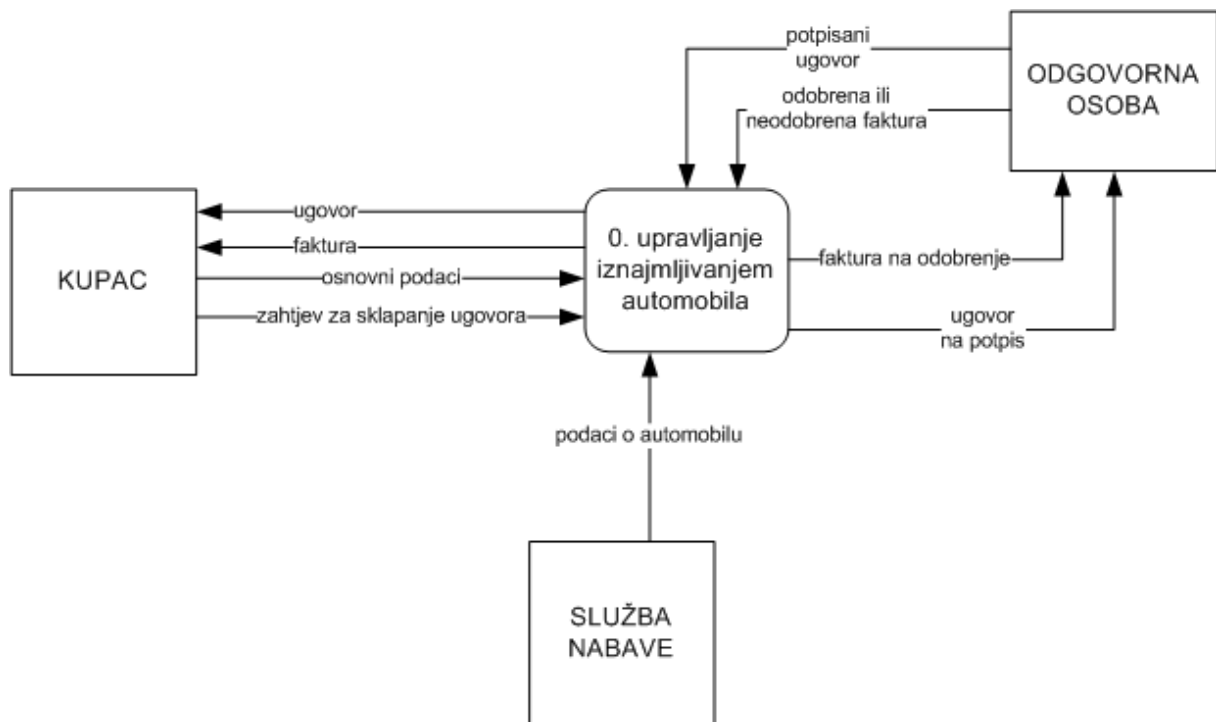
### Rješenje 17:

Dijagram dekompozicije procesa:



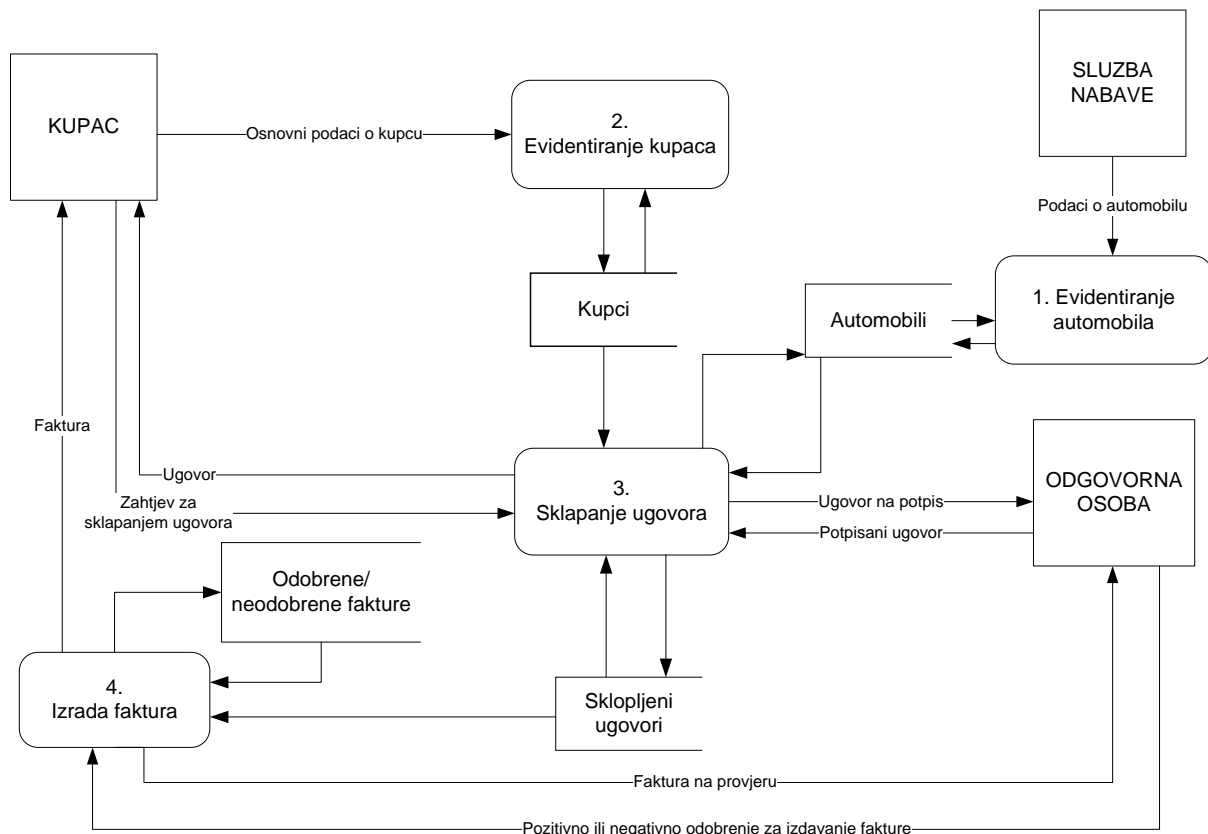
Slika 93: Rješenje zadatka 17 – dijagram dekompozicije procesa (izvor: Autori)

Dijagram konteksta:



Slika 94: Rješenje zadatka 17 – dijagram konteksta (izvor: Autori)

Dijagram 1. razine dekompozicije procesa:



Slika 95: Rješenje zadatka 17 – dijagram 1. razine (izvor: Autori)

**Zadatak 18:**

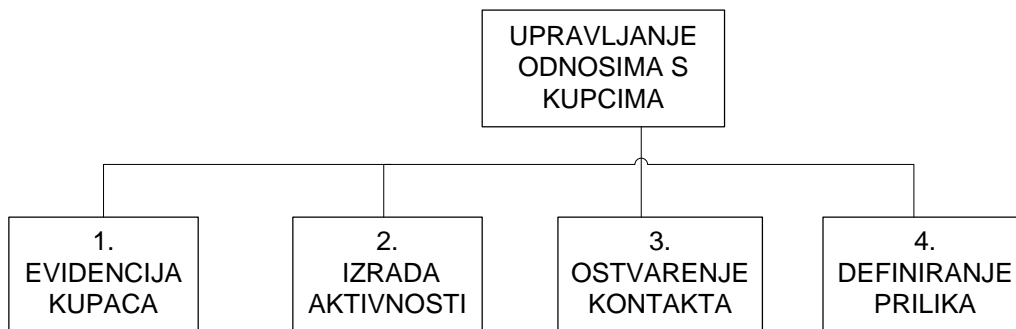
Nacrtati dijagram dekompozicije, dijagram konteksta i dijagram 1. razine koji opisuju sljedeći sustav:

*Poslovni podsustav upravljanja odnosa s kupcima*

U poslovnom podsustavu obavljaju se sljedeći poslovi: evidencija kupaca, izrada aktivnosti, ostvarenje kontakta, definiranje prilika. Evidencija kupaca obavlja se temeljem osnovnih podataka dobivenih od kupaca. Izrada aktivnosti obavlja se temeljem podataka o stanju prodaje pristiglih iz odjela prodaje. Odgovornoj osobi se dostavlja popis aktivnosti koje ona odobrava. Ostvarenje kontakta obavlja se na osnovi podataka o kupcima i na osnovi podataka o aktivnostima zbog kojih se kontakt ostvaruje. Osim toga, da bi se ostvario kontakt s kupcem, kupcu se postavljaju relevantna pitanja na koja kupac daje odgovore koji označavaju da je kontakt ostvaren. Definiranje prilika obavlja se temeljem ostvarenih kontakata. Prilike se evidentiraju, te se dostavljaju odjelu prodaje.

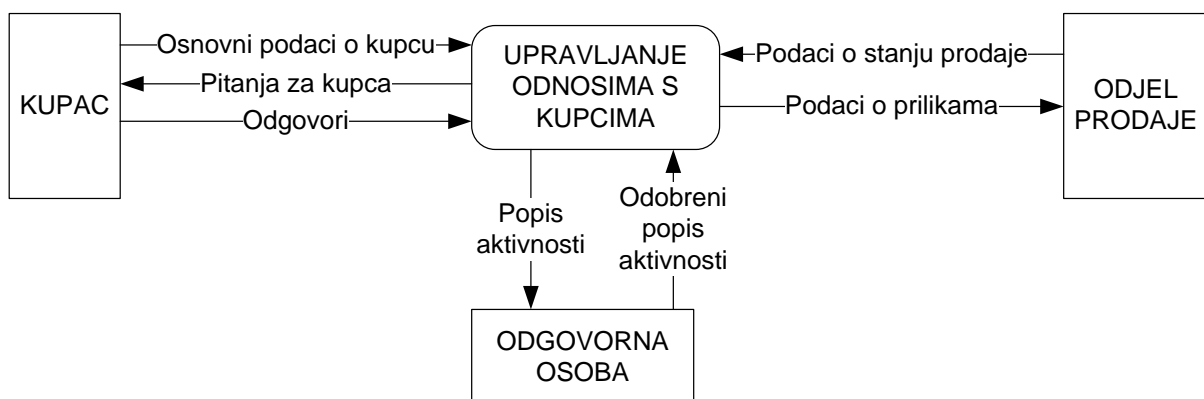
### Rješenje 18:

Dijagram dekompozicije procesa:



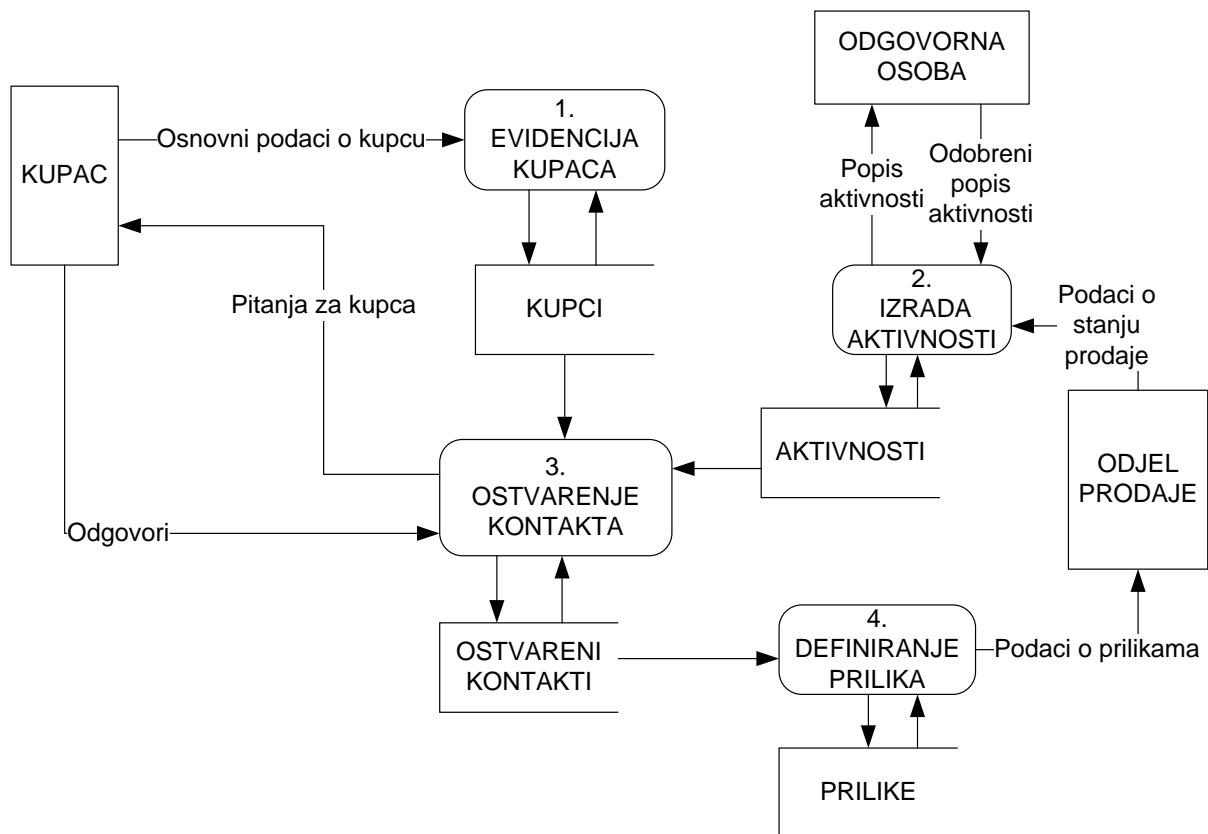
Slika 96: Rješenje zadatka 18 – dijagram dekompozicije procesa (izvor: Autori)

Dijagram konteksta:



Slika 97: Rješenje zadatka 18 – dijagram konteksta (izvor: Autori)

Dijagram 1. razine dekompozicije procesa:



Slika 98: Rješenje zadatka 18 – dijagram 1. razine (izvor: Autori)

## Popis literature

1. Chen, P. P., 1976. The Entity - Relationship Model - towards a unified view of data. *ACM Transactions on Database Systems (TODS)*, Vol.1, pp. 9-36.
2. Codd, E. F., 1970. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, Volume 12, Nr. 6.
3. Data Anomalies, 2018. *Data Anomalies*. [Mrežno]  
Dostupno na: [https://databasemanagement.fandom.com/wiki/Category:Data\\_Anomalies](https://databasemanagement.fandom.com/wiki/Category:Data_Anomalies)  
[Pokušaj pristupa 14. 12. 2018.].
4. Hercigonja, Z., 2017. *Funkcijske zavisnosti u bazama podataka*. Varaždin: Vlastita naklada.
5. Jakupović, A., 2018. *Razvoj informacijskih sustava, predavanja, Veleučilište u Rijeci*. [Mrežno]  
Dostupno na: <https://www.veleri.hr/?q=node/71#>  
[Pokušaj pristupa 28. 10. 2018.].
6. Kaluža, M., 2008. *Sustavi baza podataka*. Rijeka: Veleučilište u Rijeci.
7. Kaluža, M., 2018. *Modeliranje podataka i procesa, predavanja, Veleučilište u Rijeci*. [Mrežno]  
Dostupno na:  
[https://www.veleri.hr/?q=system/files/nastavni\\_materijali/k\\_informatika\\_1/MPP\\_predavanja.pdf](https://www.veleri.hr/?q=system/files/nastavni_materijali/k_informatika_1/MPP_predavanja.pdf)  
[Pokušaj pristupa 28. 10. 2018.].
8. Maleković, M. & Schatten, M., 2017. *Teorija i primjena baza podataka*. Varaždin: Fakultet organizacije i informatike.
9. Manger, R., 2014. *Baze podataka*. 2. ur. Zagreb: Element.
10. Martin, J., 1993. *Principles of Object Oriented Analysis and Design*. New Jersey: Prentice Hall, Englewood Cliffs.
11. Pavlić, M., 1996. *Razvoj informacijskih sustava*. Zagreb: Znak.
12. Pavlić, M., 2009. *Informacijski sustavi*. s.l.:Sveučilište u Rijeci.
13. Pavlić, M., 2011. *Oblikovanje baza podataka*. Rijeka: Odjel za informatiku Sveučilišta u Rijeci.
14. Pavlić, M., Jakupović, A. & Čandrlić, S., 2014. *Modeliranje procesa*. Rijeka: Odjel za informatiku Sveučilišta u Rijeci, Ašenbrener Katić, M (ur.).



15. Srića, V. & Spremić, M., 2000. *Informacijskom tehnologijom do uspjeha*. Zagreb: Sinergija.
16. Šuman, S., Jakupović, A. & Liverić, D., 2015. Uvod u formalizaciju metoda modeliranja podataka. U: S. Hirnig, ur. *Zbornik Veleučilišta u Rijeci*. Rijeka: Veleučilište u Rijeci, pp. 83-92.
17. Varga, M., 2020. *Baze podataka: konceptualno, logičko i fizičko modeliranje podataka*. Zagreb: vlastita naklada.
18. Whitten, J. L. & Bentley, L. D., 2007. *Systems Analysis and Design Methods*. 7. ur. SAD: McGraw-Hill/Irwin.

## Popis slika

Slika 1: Veza sustava i okoline .....	2
Slika 2: Poslovni sustav (izvor: Autori).....	5
Slika 3: Načini bilježenja informacije (izvor: Autori).....	7
Slika 4: Od podatka do znanja (izvor: Autori).....	7
Slika 5: Računalom podržani informacijski sustav (izvor: Autori).....	8
Slika 6: Elementi informacijskog sustava (izvor: Autori) .....	9
Slika 7: Položaj informacijskog sustava u poslovnoj organizaciji (izvor: Autori) .....	11
Slika 8: Dijagram dekompozicije procesa (izvor: Autori).....	17
Slika 9: Primjer modela procesa: dijagram toka podataka prve razine dekompozicije (izvor: Autori) .	17
Slika 10: Primjer EVA modela podataka (izvor: Autori).....	18
Slika 11: Primjer relacijskog modela podataka (izvor: Autori) .....	19
Slika 12: Primjer modela arhitekture programskog proizvoda (izvor: Autori) .....	20
Slika 13: Prototip dijela korisničkog sučelja (izvor: Autori) .....	21
Slika 14: Prototip dijela korisničkog sučelja (izvor: Autori) .....	21
Slika 15: Položaj modela podataka (izvor: Autori).....	23
Slika 16: Simbol za tip entiteta na EVA modelu podataka (izvor: Autori) .....	27
Slika 17: Simboli za vezu na EVA modelu podataka (izvor: Autori).....	28
Slika 18: Primjeri grafičkih prikaza veze prema različitim notacijama (izvor: Autori) .....	28
Slika 19: Primjer veze između dva entiteta na EVA modelu (izvor: Autori) .....	29
Slika 20: Primjer unarne veze na EVA modelu (izvor: Autori) .....	25
Slika 21: Primjer binarne veze na EVA modelu (izvor: Autori) .....	26
Slika 22: Primjer ternarne veze na EVA modelu (izvor: Autori).....	26
Slika 23: Primjer n-arne veze na EVA modelu (izvor: Autori) .....	27
Slika 24: Primjer višestruke veze na EVA modelu (izvor: Autori) .....	27
Slika 25: Simbol za atribut entiteta na EVA modelu podataka (izvor: Autori).....	28
Slika 26: Tip entiteta Student (izvor: Autori).....	29
Slika 27: Tip entiteta Račun (izvor: Autori).....	29
Slika 28: Odnos tipova entiteta "Student" i "Studij" (izvor: Autori).....	30
Slika 29: Odnos tipova entiteta "Račun" i "Usluge računa" (izvor: Autori) .....	30
Slika 30: Primjer podcrtanog identifikatora na dijelu EVA modela (izvor: Autori).....	31
Slika 31: Simbol za identifikacijski zavisni tip entiteta na EVA modelu podataka (izvor: Autori).....	32
Slika 32: Primjer egzistencijalno slabog entiteta na EVA modelu (izvor: Autori).....	32

Slika 33: Primjer egzistencijalno i identifikacijski slabog entiteta na EVA modelu (izvor: Autori) .....	33
Slika 34: Simbol za agregirani tip entiteta na EVA modelu podataka (izvor: Autori) .....	34
Slika 35: Primjer uvođenja agregacije na vezu M:M (izvor: Autori) .....	34
Slika 36: Od EVA modela do Relacijske baze podataka (izvor: Autori).....	39
Slika 37: EVA model podataka podsustava distribucije pošiljki (izvor: Autori) .....	40
Slika 38: Relacijski model podataka podsustava distribucije pošiljki (izvor: Autori) .....	41
Slika 39: Entiteti povezani vezom 1:1 po gornjim granicama (izvor: Autori) .....	42
Slika 40: EVA model s agregiranim tipom entiteta.....	43
Slika 41: EVA model s vezama 1:M po gornjim granicama (izvor: Autori) .....	44
Slika 42: EVA model sa slabim entitetom (izvor: Autori).....	45
Slika 43: Primjer unije (izvor: Autori).....	47
Slika 44: Primjer presjeka (izvor: Autori).....	48
Slika 45: Primjer razlike (izvor: Autori) .....	49
Slika 46: primjer projekcije (izvor: Autori).....	50
Slika 47: Primjer selekcije (izvor: Autori).....	51
Slika 48: Primjer prirodnog spoja dvaju tablica (izvor: Autori).....	53
Slika 49: Primjer lijevog vanjskog spoja dvaju tablica (izvor: Autori) .....	54
Slika 50: Primjer desnog vanjskog spoja dvaju tablica (izvor: Autori) .....	55
Slika 51: Primjer potpunog vanjskog spoja (izvor: Autori) .....	56
Slika 52: Primjer preimenovanja (izvor: Autori) .....	57
Slika 53: Primjer redundantnih podataka (izvor: Autori) .....	60
Slika 54: Primjer dekompozicije relacije (izvor: Autori).....	61
Slika 55: Primjer relacijskog modela podataka (izvor: Autori) .....	69
Slika 56: Primjer EVA modela podataka (izvor: Autori).....	69
Slika 57: Primjer relacijske baze podataka (izvor: Autori) .....	70
Slika 58: Općeniti primjer dijagrama dekompozicije (izvor: Autori).....	73
Slika 59: Simbol za proces na DTP-u (izvor: Autori).....	74
Slika 60: Simbol za tok podataka na DTP-u (izvor: Autori) .....	75
Slika 61: Simboli za spremište podataka na DTP-u (izvor: Autori) .....	75
Slika 62: Simbol za vanjski sustav na DTP-u (izvor: Autori) .....	76
Slika 63: Pravilo kreiranja DTP-a 1 (izvor: Autori).....	76
Slika 64: Pravilo kreiranja DTP-a 2 (izvor: Autori).....	76
Slika 65: Pravilo kreiranja DTP-a 3 (izvor: Autori).....	77
Slika 66: Općeniti primjer dijagrama konteksta (izvor: Autori) .....	78
Slika 67: Općeniti primjer dijagrama tokova podataka 1. razine dekompozicije (izvor: Autori).....	79

Slika 68: Općeniti primjer dijagrama tokova podataka 2. razine dekompozicije (izvor: Autori).....	80
Slika 69: Rješenje zadatka 4 u Martinovoj notaciji (izvor: Autori).....	85
Slika 70: Rješenje zadatka 4 u Chenovoj notaciji (izvor: Autori) .....	85
Slika 71: Rješenje zadatka 5 u Martinovoj notaciji (izvor: Autori).....	86
Slika 72: Rješenje zadatka 5 u Chenovoj notaciji (izvor:Autori) .....	87
Slika 73: Rješenje zadatka 6 u Martinovoj notaciji (izvor: Autori).....	88
Slika 74: Rješenje zadatka 7 u Martinovoj notaciji (izvor: Autori).....	89
Slika 75: Rješenje zadatka 8 u Martinovoj notaciji (izvor: Autori).....	91
Slika 76: Rješenje zadatka 8 u Chenovoj notaciji (izvor: Autori) .....	91
Slika 77: Rješenje zadatka 9 u Martinovoj notaciji (izvor: Autori).....	92
Slika 78: Rješenje zadatka 9 u Chenovoj notaciji (izvor: Autori) .....	93
Slika 79: Rješenje zadatka 10 u Martinovoj notaciji (izvor: Autori).....	94
Slika 80: EVA model podataka podsustava fakulteta – zadatak 11 (izvor: Autori).....	95
Slika 81: EVA model podataka nekog podsustava – zadatak 12 (izvor: Autori) .....	97
Slika 82: Rješenje zadatka 12 - relacijski model (izvor: Autori) .....	97
Slika 83: EVA model podataka nekog podsustava – zadatak 13 (izvor: Autori) .....	98
Slika 84: Rješenje zadatka 13 - relacijski model (izvor: Autori) .....	98
Slika 85: EVA model podataka nekog podsustava – zadatak 14 (izvor: Autori) .....	99
Slika 86: Rješenje zadatka 14 - relacijski model (izvor: Autori) .....	100
Slika 87: Rješenje zadatka 15 – dijagram dekompozicije procesa (izvor: Autori) .....	101
Slika 88: Rješenje zadatka 15 – dijagram konteksta (izvor: Autori) .....	102
Slika 89: Rješenje zadatka 15 – dijagram 1. razine (izvor: Autori).....	103
Slika 90: Rješenje zadatka 16 – dijagram dekompozicije procesa (izvor: Autori) .....	104
Slika 91: Rješenje zadatka 16 – dijagram konteksta (izvor: Autori) .....	104
Slika 92: Rješenje zadatka 16 – dijagram 1. razine (izvor: Autori).....	105
Slika 93: Rješenje zadatka 17 – dijagram dekompozicije procesa (izvor: Autori) .....	106
Slika 94: Rješenje zadatka 17 – dijagram konteksta (izvor: Autori) .....	106
Slika 95: Rješenje zadatka 17 – dijagram 1. razine (izvor: Autori).....	107
Slika 96: Rješenje zadatka 18 – dijagram dekompozicije procesa (izvor: Autori) .....	108
Slika 97: Rješenje zadatka 18 – dijagram konteksta (izvor: Autori) .....	108
Slika 98: Rješenje zadatka 18 – dijagram 1. razine (izvor: Autori).....	109

## Popis tablica

Tablica 1: Primjer relacije s tri sloga (izvor: Autori).....	37
Tablica 2: Usporedba koncepata EVA i Relacijskog modela (izvor: Autori).....	38