

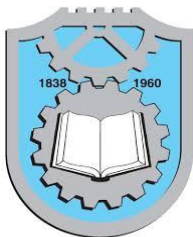
**Универзитет у Крагујевцу
Факултет Инжењерских наука
Крагујевац**

**„Развој база података и апликативног софтвера
за сегмент продаје пословног система“**

Мастер рад

Аутор: Бојана Г. Петровић

Крагујевац, 2017. године



**Универзитет у Крагујевцу
Факултет Инжењерских наука
Крагујевац**



Назив студијског програма: Индустријско инжењерство

Ниво студија: Мастер студије

Предмет: Пројектовање информационих система и база података

Број индекса: 2016/626

Бојана Г. Петровић

Мастер рад:

**Развој база података и апликативног софтвера за
сегмент продаје пословног система**

Комисија за преглед и одбрану:

1. Проф.Др. Милан Ерић, ментор

2. _____

3. _____

Датум одбране: _____

Оцена: _____

Изјава о самосталној изради рада

Изјављујем да сам овај мастер рад урадила самостално, служећи се стеченим знањем и искуством као и наведеном литературом.

626/2016 Бојана Г. Петровић

(потпис)

У оквиру мастер рада са темом “Развој база податакаи апликативног софтвера за сегмент продаје пословног система“ кандидат треба да:

За сегмент продаје пословног система изпројектује базу података и десктоп апликацију која ће омогућити обраду поруџбина, рачуна, парагон блока и пописа. Рад треба да обухвати уводна разматрања везана за моделирање као основе развоја информационих система и стандарде као подршку моделирању, функционално, информационо и апликативно моделирање као и могућности имплементације конкретног информационог система. На крају дати закључна разматрања и списак коришћене литературе.

Препоручена литература:

1. Полишчук Ј.: **Пројектовање информационих система**, Електротехнички факултет, Подгорица, 2007.
2. Rainer K., Turban E.: **Introduction to Information Systems**, John Wiley & Sons, Inc., 2009
3. Вељовић А.: **Пројектовање информационих система**, Компјутер библиотека, 2003.
4. Лазаревић Б.,: **Базе података**, ФОН Београд, Београд 2003.
5. J.Sharp, **Microsoft Visual C# 2010**, Microsoft Press, 2010.
6. Tim Anderson, **C# in easy steps**, Published by Computer Step, 2014.

Крагујевац, јун 2017.

Prof. Dr Milan D. Erić

Ментор:

Садржај

Преглед слика и табела	ii
Преглед коришћених скраћеница.....	iii
Апстракт.....	iv
1. Увод у пројектовање информационих система и база података	1
2. Систем база података	3
2.1 Врсте база података.....	6
2.1.1 Личне базе података.....	6
2.1.2 Базе података за радне групе.....	7
2.1.3 Базе података одељења	7
2.1.4 Базе података организација	8
2.1.5 Интернет, интранет и екстранет базе података	8
2.2 Циљеви увођења база података.....	9
2.3. Структура информационих система и база података	10
2.4 Алати за развој база података и њихова класификација	13
2.4.1 Алати декомпозиције	14
2.4.2 Алати тока.....	15
2.4.3 Алати нарације	16
2.4.4 Алати транзиције.....	16
2.4.5 Матични алати.....	17
2.5 Животни циклус развоја информационих система и база података	18
3. Функционално моделирање	19
3.1 Функционална декомпозиција	20
3.1.1 Дефинисање граница система	20
3.1.2 Дефинисање стабла активности.....	21
3.1.3 Верификација стабла активности	21
3.2 Дефинисање захтева корисника.....	22
3.2.1 Дефинисање захтева из докумената	22
3.2.2 Дефинисање захтева интервјуом	23
3.2.3 Дефинисање матрице односа	24
3.2.4 Анализа захтева корисника	25
3.3 Технички предуслови.....	25
3.3.1 Дефинисање архитектуре система.....	25
3.3.2 Кадровске потребе.....	26

3.3.3 Динамика реализације и трошкови.....	26
4. Модели развоја информационог система.....	27
5. Информационо моделирање.....	30
5.1 Анализа реалног система „МАЛОПРОДАЈНОГ ОБЈЕКТА"	32
5.2 Идентификовање кандидата за ентитете.....	33
5.3 Идентификација веза.....	33
5.3.1 Идентификација веза и дефинисање кардиналности за реални систем, „Малопродајног објекта.....	36
5.4 Дефинисање атрибута.....	40
5.5 Дефинисање кључева.....	41
5.5.1 Идентификација атрибута и кључева на примеру реалног система.....	42
5.5.2 ЕР дијаграм модела базе података за малопродајни објекат	44
6. Релациони модел	45
6.1 Структура релационог модела	45
6.2 Правила превођење ЕР модела у релациони модел	47
1.2. Сваки слаб објекат постаје шема релације. Назив типа објекта постаје назив	48
6.2.1 Превођење ЕР модела малопродајног објекта у релациони модел	50
7. Апликативно моделирање.....	54
7.1 Дефинисање физичког дизајна.....	54
7.2 Генерисање шеме базе података	57
7.2 Израда апликације	61
8. ЗАКЉУЧАК	67
9. ЛИТЕРАТУРА	68

Преглед слика и табела

Слика 1: СУБП као интерфејс	5
Слика 2: Структура базе података	10
Слика 3: Централизована база података	11
Слика 4: Дистрибуирана база података	12
Слика 5: Структура корисник-показатељ	12
Слика 6: Приказ дијаграма декомпозиције	14
Слика 7: Генерички модел алата тока	15
Слика 8: Графички модел алата транзиције	16
Слика 9: Активности функционалног моделирања	19
Слика 10: Модел водопада	27
Слика 11: Спирални модел	28
Слика 12: Инкрементални модел	28
Слика 13: Активности информационог моделирања	30
Слика 14: Графичко представљање ентитета	33
Слика 15: Графичко представљање везе	33
Слика 16: Графички приказ мешовите везе	34
Слика 17: Графички приказ кардиналности везе	34
Слика 18: Графички приказ тоталне везе	35
Слика 19: Приказ везе ПОСЕДУЈЕ	36
Слика 20: Приказ везе ЗАПОСЛЕН	36
Слика 21: Приказ ентитета РАДНИК и њему подређених ентитета	37
Слика 22: Приказ везе РАДИ	37
Слика 23: Приказ везе ИЗДАЈЕ	37
Слика 24: Приказ везе ЗА	37
Слика 25: Приказ везе НАБАВЉА	38
Слика 26: Приказ везе НАРУЧУЈЕ ОД	38
Слика 27: Приказ везе ДОСТАВЉА	38
Слика 28: Приказ везе У ОБЛИКУ	38
Слика 29: Приказ везе ВРШИ	39
Слика 30: Приказ везе ПРЕОСТАЛЕ	39
Слика 31: Приказ атрибута ентитета ПРОДАВНИЦА	42
Слика 32: Приказ атрибута ентитета РАДНИК	42
Слика 33: ЕР дијаграм „МАЛОПРОДАЈНОГ ОБЈЕКТА“	44
Слика 34: Microsoft Visual Studio 2010	54
Слика 35: Изглед форме након креирања новог пројекта	55
Слика 36: Изглед форме након додавања дугмета	55
Слика 37: Почетни прозор апликације	61
Слика 38: Унос података у табелу РАДНИК	62
Слика 39: Унос података у табеле КАСИР, НАРУЧИЛАЦ И МАГАЦИОНЕР	62
Слика 40: Унос података у табелу КАСА	63
Слика 41: Унос података у табелу РОБА	63
Слика 42: Унос података у табелу ДОБАВЉАЧ	64
Слика 43: Функције апликације	64
Слика 44: Функција РАЧУН	65
Слика 45: Функција ПОРУЏБИНА	65
Слика 46: Функција ПОПИС	66
Табела 1: Графички модел матичних алата	17
Табела 2: Приказ ентитета - табела, атрибута - релација.	56

Преглед коришћених скраћеница

BSP, (engleski) - *Business System Planning* – Planiranje poslovnog sistema
CASE, (engleski) - *Computer Aided Software Engineering* - компјутерско-softversko
инженјерство
DBMS, (engleski) - *DataBase Management System* – sistem za upravljanje bazama podataka
DDL, (engleski) - *Data Definition Language* – jezik za definisanje podataka
DML, (engleski) - *Data Manipulation Language* – jezik za manipulaciju podacima
ER, (engleski) - *Entity Relationship* – odnos entiteta
HIPO, (engleski) - *Hierarchy Input Proces Output* – hijerarhija procesa ulaza i izlaza
LAN, (engleski) - *Local Area Network* – lokalna računarska mreža
OOA, (engleski) - *Object Oriented Analysis* – analiza orjentisanosti objekata
SDL, (engleski) - *Storage Definition Language* – definisanje skladišta podataka
SDM, (engleski) - *Structured Design Method* – struktuiran metod dizajna
top-down, (engleski) – odozgo na dole

Апстракт

У првом делу рада описан је систем база података, њихова класификација, циљеви и структура као и алати за развој информационих система и база података.

Други део рада усмерен је на функционално моделирање, активности функционалног моделирања као и моделе развоја информационог система.

Трећи део рада усмерен је на информационо моделирање, активности информационог моделирања, креирање ЕР дијаграма као и на превођење ЕР дијаграма у релациони модел.

Четврти део рада се односи на израду апликације, опис и начин коришћења апликације.

Abstract

The first section describes the system databases, objectives and structure as well as tools for the development of information systems and databases.

The second part is directed to the modeling of the functional, activity of the functional modeling and models of the development of the information system.

The third part of the paper focuses on information modeling, information modeling activities, creation of ER diagrams and the translation of ER diagrams into the relational model.

The fourth part of the paper refers to the development of applications, description and method of use applications.

1. Увод у пројектовање информационих система и база података

У свим сферама људске делатности као и у свакодневном животу информација представља главни ресурс. Карактеристике информација као главног ресурса се односе на чињеницу да је информација неисцрпна, управо зато што нема природно порекло и не подлеже ограничењима. Такође, трошење не уништава њен садржај, већ повећава своју вредност. Информација као ресурс је потпуно индиферентна према сваком облику потрошње, односно њена експлоатација није ограничена нити временом нити интензитетом потрошње. Информација подноси мноштво конзумента истовремено, што значи да информације сачуване у бази података, на располагању су неограниченом броју корисника у исто време и са различитих локација, док је једино људска способност ограничавајући фактор њеног универзалног коришћења. Све информације су данас сачуване у облику податка у базама података.

Да би се информацијама могло манипулисати на адекватан начин, потребно је креирати адекватне информационе системе (ИС), као и системе за управљање базом података (скраћено СУБП, енгл. *DBMS - DataBase Management System*). Информациони систем се дефинише као систем у којем се везе између објеката и везе система са околином, остварују разменом информација. У основи информационог система се налази добро пројектована база података, која представља фундаменталне и стабилне системе, објекте у систему, њихове атрибуте, међусобне везе као и стање система. Систем за управљање базом података је сервер базе податка. Он обликује физички приказ базе у складу са траженом логичком структуром. Такође, обавља у има клијента све операције са подацима и брине за сигурност података, те аутоматизује административне послове са базом.

Развојем информационих система треба дефинисати што прецизнију слику реалног система, његова бивша и садашња стања као и проценити будућа понашања за даљи развој и примену информатичке технологије. Приликом описа рада пословног система врши се поступак моделирања. У процесу моделирања, елиминишу се детаљи, чиме се умањује видљива комплексност система који се проучава. Поред графичке презентације, потребно је дефинисати прецизне дефиниције, пропратни текст предмета који се појављује у моделу.

С обзиром на то, моделирање треба да:

- буде језик за комуникацију између аналитичара и корисника и
- омогући прецизну и формализовану спецификацију захтева.

Систем база података се може посматрати као хијерархија апстракција. Сваки ниво хијерархије апстракције је једна врста модела и представља скуп објеката и операција над тим објектима као и везама између објеката.

Један од ниво хијерархије апстракција је модел податка (логичка шема податка) или ER модел (скраћено од енгл. *Entity Relationship*), који проучавамо у наставку рада.

На ниво непосредно испод модела податка налази се репрезентација модела погодних структура податка односно релациони модел податка. Карактеристика другог нивоа је елиминисање утицаја рачунарског система на модел податка.

Физичка репрезентација се налази на последњем нивоу хијерархије апстракција. Подаци су представљени у бинарном облику уз присуство свих карактеристика рачунарских меморијских медија и других компонената рачунарског система.

Тема овог рада је усмерен на модел податка, релациони модел као и на превођење ЕР дијаграма у релациони модел и креирање апликације за малопродајни објекат.

2. Систем база података

Систем база података садржи четири основне компоненте:

- корисници,
- апликације над базама података,
- системи за управљање базама података и
- базе података.

База податка представља колекцију међусобно повезаних података који су организовани у табеле, а користе се за једну или више апликација. Подаци који се налазе у некој бази физички су меморисани као датотеке, па се поставља питање по чему се база података разликује од директне датотеке? Коришћењем система датотека, код којих су се подаци чували и претраживали у скупу непознатих датотека указивали су на озбиљне проблеме, као што су:

- вишеструко памћење истих података, што је доводило до проблема у ажурирању,
- високи трошкови обраде,
- великих утрошака меморијског простора,
- зависност програма од организације података,
- ниске продуктивности у развоју информационих система и
- проблема у развоју софтверских алата.

Кључна ствар која разликује базу података од директне датотеке лежи у својству независности података коју пружа Систем за управљање базама података.

Изучавање базе података и система за управљање базама података јесу основа за изучавање информационих система. Стручњак за информационе системе мора бити спреман да анализира потребе предузећа затим да дизајнира и имплементира базу података. Базе података се користе за прикупљање, чување и манипулацију подацима, на основу којих се добијају нове информације у различитим организацијама, као што су пословни системи, здравство, школство, владине институције и др. Свакодневно их користе појединци путем личних рачунара, радне групе путем мрежних сервера и сви запослени путем апликација које се налазе у пословним системима.

Постоје више начина за дефинисање база података. Базе података у ужем смислу, представљају скуп података који су организовани према потребама корисника, који се одржавају и који се користе за добијање информација. Смештене су на рачунару који омогућава лакшу и бржу обраду података и добијање жељених информација. Карактеристичан пример је ценовник у малопродајном објекту који се налази на папиру. Коришћењем базе података се могу добити подаци за све артикле са задатом ценом, могу се излистати сви артикли којима назив артикла почиње задатим словима, артикли којима се назив завршава одређеним словима и много тога.

База података смештена у рачунару, представљена је низом бита, организованих у бајтове, а са више бајтова у одговарајућем формату записују се вредности појединих података и представљају једно поље базе података.

У ширем смислу, базу податка можемо посматрати као интегрисани скуп података о неком систему и скуп поступака за њихово одржавање и коришћење, организовани према потребама корисника. То је добро структурирана колекција података, која постоји једно одређено време, која се одржава и коју користе више корисника и програмера.

Како рад са базом не значи само прикупљање података већ и њихово додавање, меморисање и брисање, те чување интегритета податка, чување сигурносних копија базе, омогућавање истовременог приступа бази од стране више клијената, проверу идентитета корисника, постаје јасно да је за то потребан СУБП. Систем за управљање базом података је софтверски систем који се користи као:

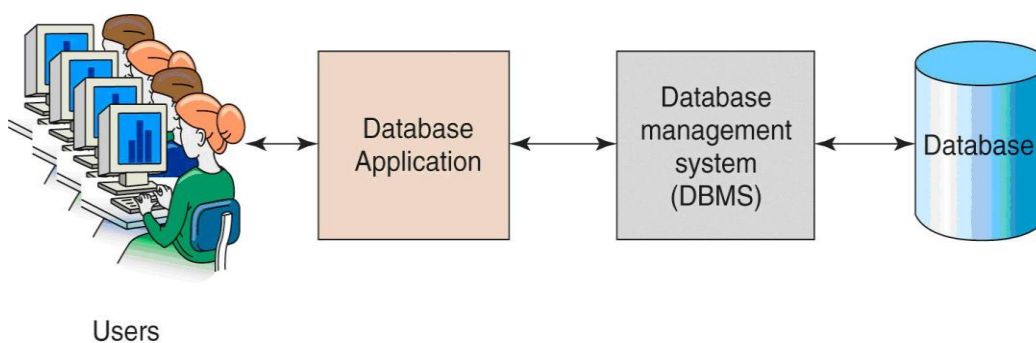
- **језик за опис података** (скраћеница енгл. *Data Definition Language – DDL*), који омогућава корисницима дефинисање типова и структура података, као и ограничења над подацима у бази података.
- **језик за манипулацију подацима** (скраћеница енгл. *Data Manipulation Language – DML*), који омогућава корисницима ажурирање, брисање и претраживање података из базе података.
- **језик за дефинисање начина меморисања података** (скраћеница енгл. *Storage Definition Language – SDL*), који се користи за специфицирање интерне шеме базе података.
- **контролисани приступ бази података**, који укључује различите функције и механизме за приступ подацима у бази података.

СУБП омогућава:

- Трајно чување података. СУБП пружа могућност чувања веће количине података, независно од процеса који те податке користи.
- Већу флексибилност, коришћењем структуре податка које подржавају приступ веома великим количинама података.
- Креирање нових база података и дефинисање нових шема, коришћењем специјализованог језика.
- Лакши приступ и измену податка, коришћењем могућих упитних језика.
- Складиштење података са минималним грешкама.
- Коришћење заједничких података од стране свих овлашћених корисника, пошто се податак меморише само једном док више програма истовремено захтева приступ истим подацима.

- Могућност опоравка базе податка. Неопходно је да СУБП, преко периодичног копирања базе на архивске меморије и памћење трансакција које су се између два копирања догодиле, омогући ефикасан опоравак базе података после дужег времена.
- Заштиту података, и то са аспекта: очувања интегритета и сигурности податка. Интегритет базе података се користи да означи тачност и коректност податка у бази и може нарушити неку грешку у програму или системски отказ. Сигурност базе податка се односи на неовлашћено коришћење и ажурирање података.

СУБП је уведен као интерфејс између корисника и базе података (слика 1). Кориснички програми не приступају подацима директно, већ комуницирају са софтвером. Када се говори о софтверу за базе података, тада се заправо мисли на СУБП. СУБП, управља структуром базе података, дефинише објекте базе (ентитете), дефинише атрибуте као и дозвољене вредности атрибута, везе између објеката, ограничења над објектима и међусобним везама.



Слика 1:СУБП као интерфејс

Извор: wordpress/wp-content/uploads/2013/09/Osnovni-pojmovi-baze-podataka.pdf

2.1 Врсте база података

У зависности од тога, на који начин су подаци интерно организовани, разликујемо:

- хијерархијске,
- мрежне,
- релационе,
- објектно-орјентисане,
- прилагођене за веби
- мултимедијалне базе података.

По броју корисника који им приступају, базе података се могу класификовати на:

- личне базе података,
- базе података за радне групе,
- базе података одељења,
- базе података предузећа и
- интернет, интранет и екстранет базе података.

2.1.1 Личне базе података

Личне базе података се праве за коришћење од стране једног корисника и већ су дуго присутне у коришћењу персоналних рачунара. Личне базе података су пронашле широку примену јер могу унапредити продуктивност појединца. Међутим, оне садрже један фактор ризика који се односи на чињеницу да податке ових база није лако делити са другим корисницима. На пример, ако би менаџер продаје желео списак клијената, то се не би могло ни брзо ни лако урадити узимањем података из личне базе података сваког од продавца. Овај пример указује на чист проблем који се односи на чињеницу, да ако су подаци од интереса за једног човека, онда су вероватно и од интереса за друге људе. Из тог разлога, личне базе података треба свести на коришћење у малим предузећима, где је вероватноћа потреба за дељењем података између корисника изузетно мала.

2.1.2 Базе података за радне групе

Радну групу чини релативно мали број људи који ради на неком пројекту или апликацији. Радну групу чини десетак људи. Сваки члан радне групе има свој рачунар, а рачунари су умрежени путем LAN-а (*скраћеница* енгл. *Local Area Network*). База податка се налази на централном рачунару који се зове сервер базе податка, који је такође на мрежи. Различити чланови групе могу имати различита овлашћења приступа подацима. Размена података отвара нова питања и проблеме који нису присутни код личних база података. Главни проблеми управљања подацима код база података за радне групе јесу проблеми безбедности и очување интегритета података, с обзиром да више корисника може истовремено да обавља ажурирање података.

2.1.3 Базе података одељења

Одељење је веће од радне групе, обично се састоји и до 100 особа, и одговорно је за већи број различитих послова. Базе података одељења су направљене да подрже различите облике послова и активности које обавља то одељење. Када су сви подаци сачувани у бази података, корисници могу да претражују базу података у циљу добијања одговора на следећа питања:

- За одређену врсту занимања, какве прилике за запослење тренутно постоје у одређеној делатности?
- За ту врсту посла, која стручна спрема или вештина је неопходна?
- Које вештине и знање поседује одређени радник? И обрнуто, који радници поседују одређено знање и вештине?
- Који све радници су обављали одређени посао у организацији? И обрнуто, које све послове је одређени радник обављао у организацији?
- Које све запослене нагледа овлашћено лице?

Питања на које треба одговорити, при прављењу базе података за одељења су:

1. Како база података и њено окружење треба да буду организовани, да би се оствариле задовољавајуће перформансе, узимајући у обзир велики број корисника и велики број трансакција?
2. Како на одговарајући начин обезбедити податке од незадовољног приступа и/или дистрибуције?
3. Које алате за развој база података и апликација треба користити у случају комплексног окружења?

4. Да ли и друга одељења користе исте врсте података?
5. Да ли су корисници база података географски једни од других удаљени или је величина базе података толика да се подаци могу чувати на више рачунара, стварајући дистрибуиране базе података?
6. Да ли се бази података може приступити преко Интернета и да ли она треба да буде укључена у интранет организације?

2.1.4 Базе података организација

Базе података организација обухвата више одељења једне организације. Једна организација може имати више база података, тако да једна таква база података не садржи све податке једне организације. Током последњих деценија, развој база података организација је довео до развоја два најважнија облика:

- *ЕРП система и*
- *имплементације складишта података.*

ЕРП системи раде са текућим подацима организације, док складиште података пружа могућност корисницима да раде са прошлим подацима. Складиште података прикупља податке из оперативних база података укључујући и личне базе, базе радних група, одељења и ЕРП базе података.

2.1.5 Интернет, интранет и екстранет базе података

Све веће коришћење Интернета, светске мреже која повезује кориснике, довело је до промена у окружењу база података. Лак приступ Интернету свих врста платформи омогућава компанијама да реорганизују своје послове и развију брже апликације уз ниске трошкове. Основа развоја корисничке апликације је прикључивање базе података. Када је база података повезана са неком Интернет локацијом, корисници преко ВЕБ-а могу да постављају одређена питања на која ће добити одговоре базиране на свежим информацијама. Интернет базе података су незаменљиве у развоју сајтова за куповину преко Интернета.

Многе организације су користиле Интернет технологију за стварање приватне мреже, намењене за управљање информацијама у оквиру организације. По изгледу не постоји разлика између интранета и Интернет страница, али приступ интранет страници је ограничен само на кориснике у оквиру једне организације. Интранет може да оствари Интернет конекцију али та конекција ће бити заштићена.

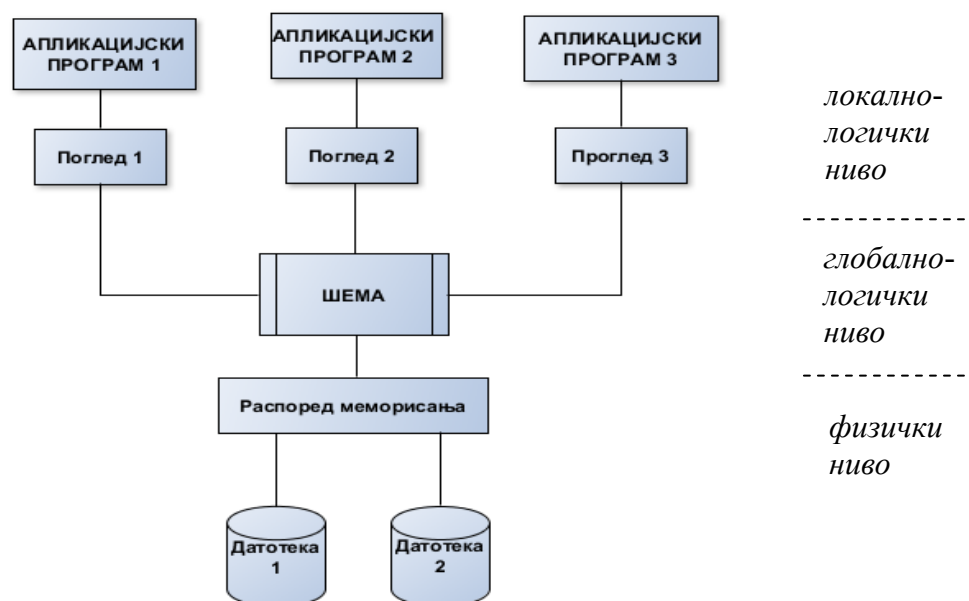
2.2 Циљеви увођења база података

Базе података представљају виши ниво рада са податцима у односу на класичне програмске језике. Виши ниво рада се огледа у примени технологија база података која настоји да задовољи следеће циљеве:

- **Физичка независност податка**, подразумева раздвајање логичке дефиниције базе податка од њене стварне физичке грађе. Значи, ако се физичка грађа промени, то неће захтевати промене у постојећим апликацијама.
- **Логичка независност податка**, подразумева раздвајање глобално-логичке дефиниције целе базе података од локално-логичке дефиниције за једну апликацију. Значи, ако се логичка дефиниција промени (нпр, уведе се нови запис или веза), то неће захтевати промене у постојећим апликацијама. Локално-логичка дефиниција се обично своди на издвајање само неких елемената из глобалне дефиниције, уз неке једноставне трансформације тих елемената.
- **Флексибилност приступа податцима**. У старијим мрежним и хијерархијским базама, приступи информацијама били су унапред дефинисани, дакле корисник је могао претраживати податке једино оним распоредом који је био предвиђен у време пројектовања и имплементирања базе. Данас је омогућено да корисник слободно приступи податцима и да по свом нахођењу успостави везу између података. Овај захтев задовољавају једино релационе базе.
- **Истовремени приступ до податка**. База мора омогућити да већи број корисника може истовремено користити исте податке, с тим што један корисник не сме ометати другог корисника и сваки од њих има утисак да сам ради са базом.
- **Чување интегритета**. Односи се на чување коректности и конзистенције података у ситуацији када постоји грешка у апликацијама, и пружа могућност опоравка након кvara. Чување интегритета омогућава поуздану заштиту базе у случају кvara хардвера или грешке у раду системског софтвера.
- **Заштита од неовлашћеног коришћења**. Пружа могућност да се корисницима ограниче права коришћења базе, што подразумева регулисање овлашћења, шта корисник може са податцима а шта не може.
- **Задовољавајућа база приступа**. Операције с подацима мора да се одвијају довољно брзо, у складу са потребама апликације. На брзину приступа утичу физичка структура податка.
- **Могућност подешавања и контроле**. Велика база захтева: праћење перформанси, мерење параметара у физичкој грађи, регулисање овлашћења корисника. Ови послови се морају одвијати централизовано. Администратору треба да стоје на располагању разни алати.

2.3. Структура информационих система и база података

Структура базе података се састоји од три слоја и интерфејса међу слојевима (слика 2). Реч је о три нивоа апстракције: локално-логички (први ниво апстракције), глобално-логички (други ниво апстракције) и физички ниво (трећи ниво апстракције).



Слика2: Структура базе података
Извор: Небојша Лукућ, SQL Server

Локално-логички ниво (први ниво апстракције), односи се на логички део базе који користи поједина апликација. Тај аспект види корисник или апликациони програмер. Запис једне локално-логичке дефиниције зове се преглед или под-шема. То је текст или дијаграм којим се именују и дефинишу сви локални типови податка и веза међу тим типовима у складу са правилима коришћења модела. Преглед задаје пресликавање којим се из глобалних података и веза изводе локални.

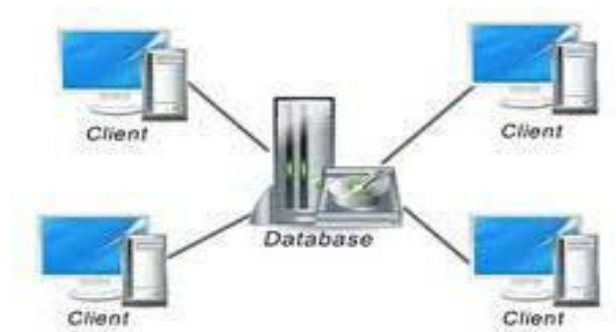
Глобално-логички ниво (други ниво апстракције), односи се на логичку структуру целе базе. Тај аспект води пројектант базе или администратор. Запис логичке дефиниције назива се шема. Шема је текст или дијаграм који дефинише логичку шему базе и подразумева именовање и дефинисање свих типова података и веза између података, у складу са правилима коришћења модела. Такође, шема уводи и ограничења којим се чува интегритет података.

Физички ниво (трећи ниво апстракције), односи се на физички приказ и распоред података на јединицама спољне меморије. Тај аспект виде само системски програмери. Физички ниво се може даље поделити на више под-нивоа апстракције, од сасвим конкретних стаза и цилиндара на диску, до већ донекле апстрактних појмова датотеке и записа какве сусрећемо у класичним програмским језицима. Распоред меморије описује како се елементи логичке дефиниције базе пресликавају на физичке уређаје.

Типови структуре базе података се могу класификовати на:

- централизоване базе података,
- дистрибуиране базе података,
- корисник- показатељ базе података и
- паралелне базе података.

Централизоване базе података користе тзв. Dummy терминале. То значи да се форматирање и процесирање садржаја базе података одвија на средишњем рачунару (слика 3). Таква структура ствара велико процесорско оптерећење средишњег рачунара, те највећи проблем је блокада целог система.



Слика 3: Централизована база података
Извор: Силвија Давила, „Дизајн база података”

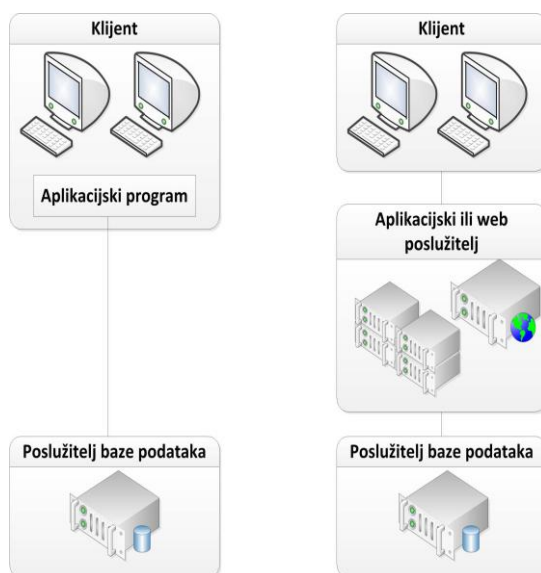
Дистрибуирана база података користи базе података које су распоређене на више различитих локација (слици 4). Код такве структуре базе, корисник мисли да све време ради са централизованом базом. Састав који подржава такав рад је СУБП.



Слика 4: Дистрибуирана база података
Извор: Силвија Давила, „Дизајн база података”

Код структуре базе података корисник-показатељ, корисник има могућност процесирања и форматирања података у бази података. Таква врста структуре може бити:

- **двослојна**—на корисничком рачунару се налазе апликацијски програми и
- **трослојна**—састоји се од апликацијског или ВЕБ претаживача и корисника који само показује резултате (слика 5).



Слика5:Структура корисник-показатељ
Извор: Силвија Давила, „Дизајн база података”

Архитектура паралелних база података користи се код великих база података и може бити:

- *архитектура дељене меморије* – процеси деле исту меморију и дискове,
- *архитектура дељених дискова* – сваки процес има приступ било којем диску и само свом делу меморије и
- *архитектура без дељења меморије и дискова* – сваки чвор садржи процес, меморију и један или више дискова.

За стварање базе података потребно је задати само шему и погледе. СУБП тада аутоматски генерише потребни распоред меморисања и физичку базу. Администратор може само донекле утицати на физичку грађу базе, подешавањем њему доступних параметара. Програми и корисници не приступају директно физичкој бази, већ меморисању података посредством СУПБ-а. Комуникација програма односно корисника са СУБП-ом обавља се на локално-логичком нивоу.

2.4 Алати за развој база података и њихова класификација

Информатички експерти настоје прикупити довољну колекцију алата за развој информационих система и база података. Неки од тих алата су једноставни и лако се користе, док су други захтевни и комплексни. Затим, неки имају веома распрострањену примену, док се други користе у специфичним околностима. Један део алата егзистира и веома дуго се примењује, док неки представљају новину у развоју база података.

Постоје више метода, као што су ООА (*скраћеница енгл. Object Oriented Analysis*), HIPO (*скраћеница енгл. Hierarchy Input Proces Output*), BSP (*скраћеница енгл. Bussiness Sistem Planing*), DFD (*скраћеница енгл. Date Flow Dijagram*), SSA (*скраћеница енгл. Sistem Structured*), SDM (*скраћеница енгл. Structured Design Method*), и др. , многе су преточене у CASE алате (*скраћеница енгл. Computer Aided Software Engineering*), за анализу и декомпозицију система. Улога самих алата је у домену проширења и побољшања комуникације између већег броја људи који су иновативни у развоју база података.

Алати који се користе у развоју база података, могу се класификовати на следећи начин:

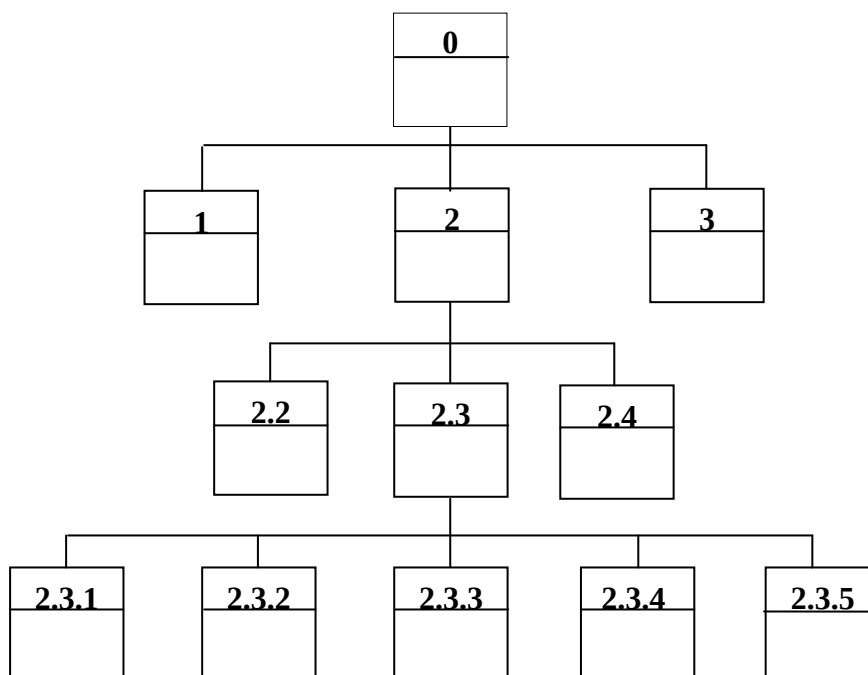
- *алати декомпозиције,*
- *алати тока,*
- *алати нарације ,*
- *алати транзиције и*
- *матични алати.*

2.4.1 Алати декомпозиције

Алати декомпозиције су графички алати који се користе у top-down (одозго на доле) приступу за рашчлањавање ентитета у што мање делова. Ова врста алата, примењује се у три случаја. Први случај примене се огледа у коришћењу од стране информатичких експерата у циљу рашчлањавања проблема у форме индивидуално одабраних делова.

Други случај примене алата декомпозиције се заснива на организовању алата тока. Оваквим приступом се постиже, унутар неких методологија развоја информационих система, да дијаграм хијерахије прераста у мапе за дијаграме тока податка. На једнак начин, структуриран дијаграм постаје мапа за програмске блок-дијаграме.

Трећи случај примене алата декомпозиције се базира на концепту реверзибилног инжењеринга. Декомпозицијски дијаграми омогућавају да се прати пут од модула најнижег нивоа до модула највишег приоритета. Овај начин примене се може илустровати примером у коме се захтева промена кода за неки програмски модул. Овај модул може бити или неки под-програм или неки сегмент унутар датог програма, као што је приказано на слици 6.



Слика 6: Приказ дијаграма декомпозиције

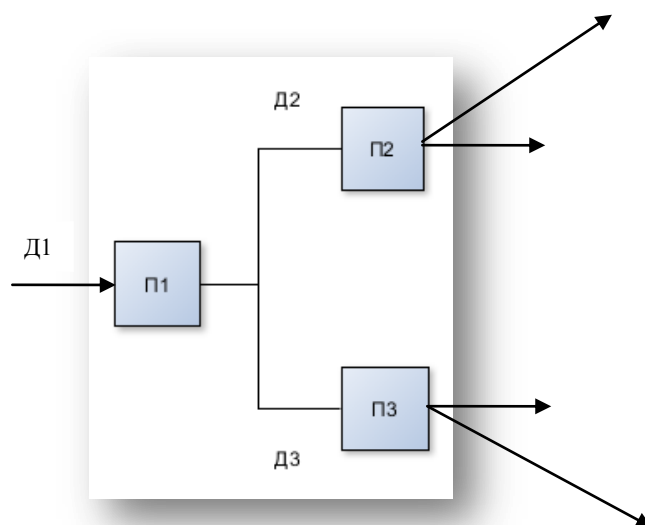
Извор: Милан Ерић, „Пројектовање информационих система и базе података”

2.4.2 Алати тока

Алати тока су такође графички алати који презентују ток информација или физичких ресурса између појединих процеса. Под физичким ресурсима се подразумевају сировински материјали који су предмет процесних обрада усмерених ка трансформацији улазних ресурса у полупроизводе. Генерички модел алата тока, приказан је на слици 7.

П - процес

Д - податак



Слика 7: Генерички модел алата тока

Извор: Милан Ерић, „Пројектовање информационих система и базе података”

Алати тока имају две намене. Прва намена се огледа у идентификацији и одређивању делова система. На такав начин, дијаграм тока податка презентује систем кроз токове податка/информација, базу података, процесе који одређују податке и ентитете који примају или шаљу податке. У том случају, дијаграм тока података постаје листа спецификација компонената система које морају бити узете у разматрање током анализе и дизајнирања информационог система.

Друга намена алата тока се темељи на приказивању распореда-секвенци и истовремености-конкурентности операција. На овај начин системски блок-дијаграм приказује секвенцу корака који се морају обавити да би се обрадила нека пословна функција.

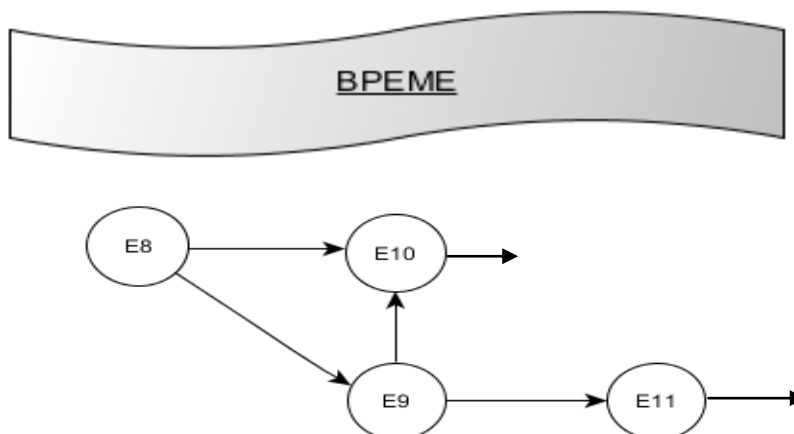
2.4.3 Алати наратије

Алати наратије имају две примене, односно могу се користити на два начина. Први начин се заснива на транслацији из графичког формата у форму која је подложна програмским језицима. На пример, веома је тешко користити дијаграме тока податка у програмирању било којим програмским језиком, тако да се мора извршити транслација дијаграма у погодније форме.

Други начин примене алата наратије је обухватнији. Алати наратије представљају једну алтернативу графичким алатима. Пример за илустрацију је писање процедура којом се описује како ради неки информациони систем. На такав начин је омогућено разумевање процедура без обзира на графичку или наративну оријентисаност оног ко је тумачи.

2.4.4 Алати транзиције

Алати транзиције су намењени за приказивање промена система или процеса, које настају током времена. Графичка форма алата транзиције, приказана је на слици 8.



Слика8: Графички модел алата транзиције

Извор: Милан Ерић, „Пројектовање информационих система и базе података”

2.4.5 Матични алати

Матични алати су у суштини табеле, док је главно обележје ових алата засновано на одређеним формама колона и редова (табела 1).

Матични алати се примењују на три начина. Први начин употребе се огледа у томе што пружају могућност брзог претраживања и проналажења најмањих делова података. На пример, могу се скенирати колоне великих табела, како би се пронашао податак.

Други начин употребе матичних алата се манифестује у креирању и осигурању једне контролне листе активности. На пример, матрица типа улаз-излаз допушта секвенцијална испитивања сваког излазног извештаја, проверу сваког поља улазних података и др.

Трећи начин употребе матичних алата се темељи на приказу комплексног сета процедура на једноставан и ефикасан начин.

Табела 1: *Графички модел матичних алата*

Извор: Милан Ерић, „Пројектовање информационих система и базе података”

2.5 Животни циклус развоја информационих система и база података

Развој информационих система и база података је веома сложен и дуготрајан процес који подразумева низ фаза и активности, од планирања развоја до застаревања система, њихове замене и повлачења из употребе. Неопходно је добро пројектовати и припремити увођење информационог система, из разлога што његов развој представља значајну финансијску инвестицију.

Да би се детаљно разрадио процес развоја софтвера, неопходно је дефинисати његов животни циклус, моделе развоја активности у свакој фази животног циклуса и резултате сваке активности. Методологија животног циклуса полази од чињенице да животни век сваког програмског производа пролази кроз исте фазе.

Развој информационих система прошао је кроз следеће фазе животног циклуса:

- *функционално моделирање,*
- *информационо моделирање,*
- *апликативно моделирање и*
- *имплементација.*

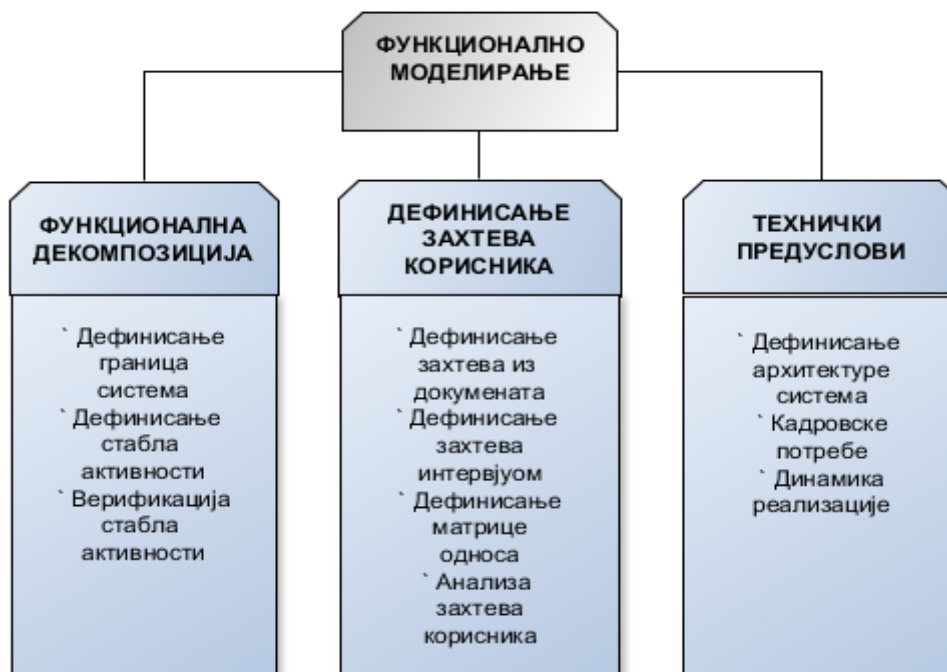
3. Функционално моделирање

Активност функционалног моделирања омогућава декомпоновање пословних функција и планирање потребних ресурса за реализацију функција. Функционално моделирање је везано за коришћење ИДЕФО технике. ИДЕФО је техника моделирања активности базираних на комбинацији графике и текста, који су представљени на организован и систематичан начин да би се повећала разумљивост.

ИДЕФО функционални модел се састоји од хијерархијског низа дијаграма који постепено показују све више детаља о функцијама и о њиховој вези са осталим деловима система. ИДЕФО моделирање омогућује анализу особина одређеног пословног процеса ради његовог максималног унапређења.

На слици 7. приказана је активност функционалног моделирања којом се дефинишу три подређене активности:

1. функционална декомпозиција,
2. дефинисање захтева корисника и
3. технички предуслови.



Слика 9: Активности функционалног моделирања

3.1 Функционална декомпозиција

Приликом реализације активности функционалне декомпозиције, посебна пажња се поклања захтевима топ-менаџмента. У предузећу, практично, треба да се рефлектује вођење пословања водећег менаџмента и да успех реинжењеринга пословних процеса зависи од њихових поставки, датих у облику визија, мисије и дефинисаних циљева. Активност функционалне декомпозиције, изводе се кроз три подређене активности:

- дефинисање граница система,
- дефинисање стабла активности и
- верификација стабла активности.

3.1.1 Дефинисање граница система

Дефинисање граница система је везано за набрајање објеката који ће у следећем кораку бити по хијерархији повезани у стабло активности.

У оквиру утврђења граница система, треба јасно дефинисати циљеве који треба да садрже следеће елементе:

- зашто се процес моделира?
- шта ће процес да прикаже?
- шта ће корисник модела направити са њим?
- чему служи модел?

Одговори на питања треба да помогну у решавању проблема. Следећа питања на која треба дати одговор су:

- који су задаци на датом радном месту?
- који је распоред извођења корака?
- како се изводи контрола?
- који се ресурси користе?

Дакле, треба идентификовати задатке сваког запосленог и схватити однос између задатака. За извођење ових активности користи се графички језик ИДЕФО.

3.1.2 Дефинисање стабла активности

На основу дефинисаних граница система прелази се на следећу активност функционалног моделирања односно дефинисање стабла активности, где се успостављају вертикалне везе између активности.

Стабло активности се дефинише применом метода решавања проблема одозго на доле, када се сложена активност раставља на више подређених активности, чија је структура типа стабла. Дакле, стабло представља хијерархију дефинисаних активности и омогућава функционалну декомпозицију и увид у дубину одвијања веза између активности.

3.1.3 Верификација стабла активности

Верификација стабла активности веома је важна одлука, јер је то стратешки моменат, тј. одлука руководства да ли је то оно што се жели постићи реинжењерингом пословних процеса. Овај елемент се мора разматрати приликом класичног начина увођења докумената за обезбеђење квалитета, по стандарду ИСО 9000.

Верификационо тело треба стално да контролише рад стручног тима, уз благовремено отклањање евентуалних грешака и недоследности у њиховом раду. Постојање овог тела подразумева да је пројекат контролисан и верификован у свим фазама израде.

Након верификације стабла активности од стране топ-менаџмента, приступа се дефинисању захтеву корисника.

3.2 Дефинисање захтева корисника

Дефинисање захтева корисника подразумева информисање и учење пројектанта односно упознавање са потребама и жељама корисника, како би пројектант могао да успостави информационе везе и донесе правилне закључке.

Активност „Дефинисање захтева корисника“ изводи се кроз четири подређене активности:

- дефинисање захтева из докумената,
- дефинисање захтева интервјуом,
- дефинисање матрице односа и
- анализа захтева корисника.

3.2.1 Дефинисање захтева из докумената

Дефинисање захтева из докумената је поглед одозго на доле и у надређеној активности дефинисање захтева корисника има глобални карактер, јер се прикупљају документа из целе фирме. У оквиру наведене активности потребно је прикупити:

- улазне документе,
- излазне документе,
- узроке извештаја,
- организационе прописе и др.

За прикупљање наведене документације, потребно је дати одговор на следећа питања:

- Одакле потичу подаци?
- Како су подаци добијени?
- Колики је максимум и минимум података добијених у пракси?
- Да ли се уносе тачни подаци?
- Како се користе?
- Које се упућују?
- Шта стартује излаз?

Поступак рада са документима захтева дефинисање одговарајућих процедура и организационих прописа које треба проучавати и иновирати постојећом праксом. Анализа докумената помаже аналитичару да формира мишљење о читкости из докумената, као и да разуме корисникову технологију, да би у следећем процесу поставио права питања приликом спровођења интервјуа.

3.2.2 Дефинисање захтева интервјуом

Дефинисање захтева интервјуом је приступ одозго на доле, која треба да омогући дефинисање:

- потреба за информацијама,
- циљева и
- проблема како их виде руководиоци и непосредни извршиоци.

Дефинисање захтева интервјуом је кључни моменат у реинжењерингу пословних процеса, јер се овде руководство изјашњава о питању будућности, односно даљем пословању. Први кораци за извођење интервјуа су везани за дефинисање:

- листе руководилаца за интервјуе,
- временског распореда интервјуа,
- тема за разговор,
- организације групе за интервјуе,
- потврде термина,
- вођење записника,
- припреме панела,
- опремање просторије и
- избор општих питања и пробни интервју.

Посебно треба нагласити организацију групе за интервјуе, где треба дефинисати:

- који ће члан тима водити записник?
- ко ће презентовани досадашње резултате?
- ко ће постављати питања?

Наведене активности треба да омогуће да се за тренутно постојеће објекте пословања (организационе јединице, процесе и др.), апликације и датотеке идентификују редундантности података, разјасне одговорности и уопште разуме пословање. Да би се оствариле ове активности приступа се поступку интервјуисања, почев од највиших руководиоца до непосредних корисника.

Општа питања интервјуисања су:

- Које су надлежности и одговорности?
- Шта су основни циљеви и какве се промене могу очекивати у одређеној области за следећу годину и даље?
- Који су критични фактори у погледу одговорности, управљања и одлучивања?
- Какви су захтеви за информацијама?
- Који су највећи проблеми били у последње време и шта је било потребно за решавање тих проблема?
- У којим процесима се могу постићи побољшања и које су потребе за подацима?
- Које од прикупљених информације су употребљиве?

По завршетку интервјуа треба одговоре размотрити, написати резиме, издвојити уочене проблеме и дефинисати закључке.

3.2.3 Дефинисање матрице односа

Дефинисање матрице односа треба да дефинише матрицу веза између активности и докумената која треба да повежу документа одређена као улазне или излазне информације са активностима из стабла активности. Документа су дефинисана одговарајућим рубрикама, чијом се анализом одређује одговарајући ентитети.

Ентитети на сваком нивоу представљају објекат који се може описати неким особинама. Сваком ентитет се придодаје начин на који активност користи тај ентитет, односно шта ради са појединим инстанцама тог ентитета преко тзв. ЦРУД матрице.

Ентитет се у оквиру неке активности: и/или креира (C-Crtate), и/или потражује (R-Retrive), и/или ажурира (U-Update), и/или брише (D-Delite), па отуда се и назива ЦРУД матрица.

На основу изведених, претходно описаних активности у наставку се приступа анализи захтева будућег корисника софтверске апликације.

3.2.4 Анализа захтева корисника

Анализа захтева корисника подразумева да активности дефинисане у претходним корацима, морају бити верификоване од стране одговарајућег овлашћеног лица. Задатак овлашћеног лица је да усваја резултате рада стручног тима у контролним тачкама. Све наведене активности и подактивности морају ићи наведеним распоредом и да дефинисање техничких предуслова тек онда долази у разматрање.

3.3 Технички предуслови

Технички предуслови се изводе кроз три подређене активности:

- дефинисање архитектуре система,
- кадровске потребе и
- динамика реализације и трошкови.

Технички предуслови подразумевају, пре свега, рачунарски систем са дефинисаним:

- **системом хардвера** (радна меморија, масовна меморија, улазне јединице, излазне јединице и централна процесорска јединица),
- **системом софтвера** (оперативни систем, језички процесори, апликативни софтвер) и
- **системом документације** (документација за систем софтвера).

3.3.1 Дефинисање архитектуре система

Дефинисање архитектуре система треба да укаже на основе претпоставке које се морају испунити да би се могао спроводити поступак реинжењеринга пословних процеса. Реинжењеринг пословних процеса захтева најновија сазнања, технике и компатибилности у мрежама рачунара и употребу уређаја за приказ информација.

При избору опреме, треба имати у виду техничко-технолошке предуслове, који су сагледани према структури референтног модела:

- квалитетан комуникациони систем,
- висок степен компатибилности рачунарске опреме,
- отвореност мрежне архитектуре,
- модуларност опреме крајњих корисника,
- ефикасност система управљања подацима и
- коришћење софтверских производа за развој апликације.

3.3.2 Кадровске потребе

Под кадровским потребама подразумева се број потребног кадра за реализацију пројекта реинжењеринга пословних процеса и потребна обука за коришћење информационе технологије.

Информатичка основа у спровођењу реинжењеринга пословних процеса, треба имати минимум потребног кадра. Потребни су:

- руководицац,
- водећи пројектант за моделирање процеса и података,
- водећи пројектант софтверских решења,
- водећи пројектант базе података,
- систем инжењер и
- референт документације.

Потребни кадар за реализацију реинжењеринга пословних процеса мора проћи одређене обуке и курсеве, како би се оспособио за рад на реализацији пројекта.

3.3.3 Динамика реализације и трошкови

Динамика реализације и трошкови подразумева коришћење неког од софтвера за управљање пројектима.

Трошкови реализације се могу класификовати на:

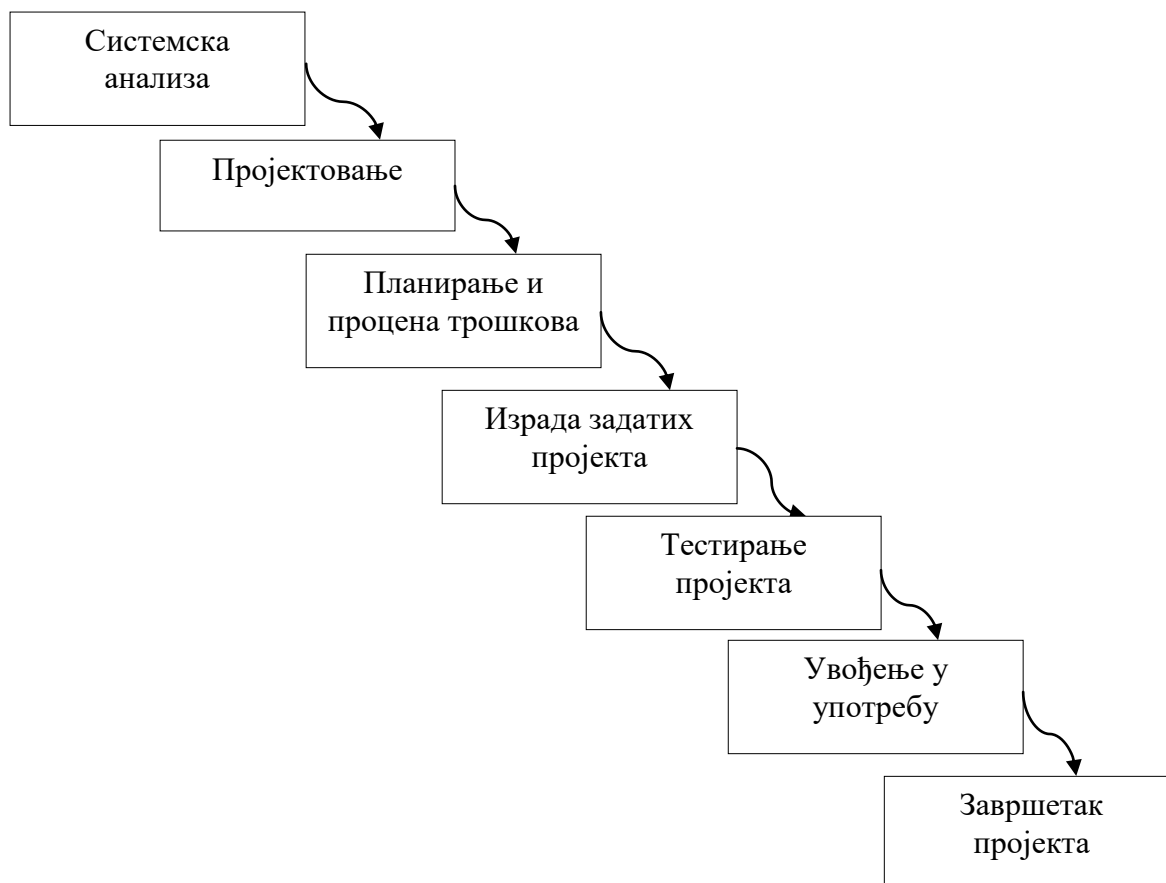
- **трошкови развоја:**
 - обука пројектног тима,
 - развој заједничких апликација,
 - стручна помоћ при развоју,
 - развој и допуна сопствених апликација и
 - софтверски производи за развој апликација.
- **трошкови техничко-технолошких ресурса:**
 - рачунарска опрема заједничких ресурса,
 - опрема комуникационог система,
 - допуна и комплетирање рачунарске опреме и
 - пратећа опрема и адаптација простора.
- **трошкови експлоатације:**
 - одржавање опреме,
 - потрошња електричне енергије,
 - коришћење комуникационих линија,
 - амортизација опреме и
 - плате радника.

4. Модели развоја информационог система

Модели развоја информационог система се могу класификовати на следеће:

- *модел водопада,*
- *спирални модел,*
- *инкрементални модел,*
- *прототипски развој и*
- *брзи развој апликација.*

Модел водопада је практично једноставан модел и естетски допадљив. Поступак модела водопада започиње системском анализом, која се назива анализа захтева, Након системске анализе, следећи корак је детаљно пројектовање целог система. Тој фази следи планирање и процена трошкова, а затим израда, тестирање и увођење целог система у употребу. Графички приказ модела водопада приказан је на слици 10.



Слика 10: Модел водопада

Извор: Милан Ерић, „Пројектовање информационог система и базе података”

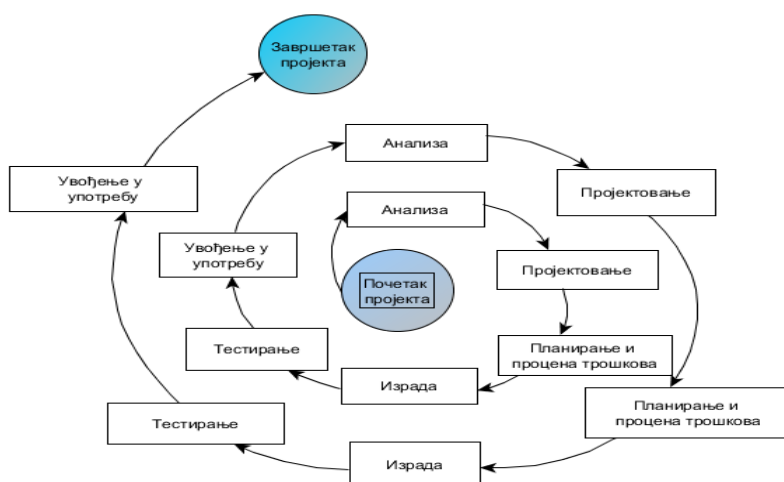
Свака појединачна активност завршава се и одобрава пре почетка следеће. Модел омогућава висок степен контроле над трошковима, радом учесника и утрошком времена. Са своје стране, свака фаза је организована као скуп активности које могу бити извршаване од стране већег броја људи истовремено.

Модел полази од претпоставке да су све информације, које су неопходне за обављање одређеног посла, познате током његовог обављања и не допушта накнадну појаву нових информација током поступка.

Модел водопада не дозвољава ни мењање пословних захтева током израде пројекта. Потпуно је нереално претпоставити да ће систем, који испуњава пословне захтеве на почетку неког пројекта, испунити те захтеве након две, три године рада. Међутим, мора се имати у виду да су активности идентификоване у моделу водопада потпуно исправне. Проблем са моделом водопада је његова линеарност, тј. претпоставка да се никад не треба враћати на одређену фазу након његовог завршетка.

Као решење проблема модела водопада, смишљено је више алтернативних модела развоја информационих система.

Спирални модел омогућава више итерација модела водопада, при чему свака међу њима проширује опсег претходне итерације, као што је приказано на слици 11.



Слика 11: Спирални модел

Извор: Милан Ерић, „Пројектовање информационих система и базе података”

Проблем са спиралним моделом је чињеница да се целокупан опсег пројекта не може разумети све до пред сам краја развоја пројекта.

Модел који се сматра најприкладнијим за сложене системе, описује се као инкрементални развој или еволутивни развој, приказан је на слици 12.



Слика 12: Инкрементални модел

Извор: Милан Ерић, „Пројектовање информационих система и базе података”

У моделу инкременталног развоја, који је у многим аспектима само варијанта спиралног модела, изводи се почетна анализа целог система, а не само једног његовог дела. Затим следи пројектовање структуре, такође целог система. Циљ пројектовања структуре је да се утврде компоненте које се могу реализовати мање или више независно и да се опишу односи и зависности између тих компонената. Затим следи детаљно пројектовање и израда сваке компоненте на основу модела који је најприкладнији. У тој фази се користи спирални модел јер омогућава већу флексибилност при пројектовању и реализовању.

Проблем инкременталног развоја јесте то што претпоставља да се сваки сложен систем може разбити на јасно раздвојене компоненте, а то није увек могуће у свим системима. Предност овог модела су измене захтева које се могу лако реализовати мењањем компонената које тек треба развити, или чак мењањем распореда којим се компоненте развијају, без губљења већ уложеног труда.

5. Информационо моделирање

Информационо моделирање се изводи на основу дефинисаних критичних функција за претходно урађену активност „Функционално моделирање“. Информационо моделирање је кључни моменат где до изражаја долазе способности и знање стручног кадра из области менаџмента и информатике. Информационо моделирање треба да омогући:

- Дефинисање системске документације, која се може користити за базу података и за развој апликација да би особље могло да дефинише системске захтеве.
- Бољу комуникацију међусобно и са крајњим корисницима.
- Јасну слику о пословним правилима.
- Логичку слику базе података који могу користити аутоматски алати за генерисање система за управљање базама података.

Информационо моделирање се изводи кроз четири подактивности, што је приказано на слици 13:

- *дефинисање детаљних захтева,*
- *креирање ЕР дијаграма,*
- *креирање атрибута и*
- *дефинисање пословних правила.*



Слика 13: Активности информационог моделирања

Прва активност у оквиру информационог моделирања представља дефинисање детаљних захтева, у оквиру ког се дефинишу послови реинжењеринга. Дефинисање детаљних захтева треба да укаже на ревизију пословних елемената из студије, уз даље детаљно продубљивање за изабрану функцију за коју се спроводи реинжењеринг пословних процеса. Дефинисање детаљних захтева се изводи посредством следећих активности:

- израда детаљног стабла активности,
- дефинисање декомпозиционог дијаграма,
- дефинисање детаљних матрица односа,
- дефинисање дијаграма тока података и
- анализа детаљних захтева.

Друга активност у оквиру информационог моделирања се односи на креирање ЕР дијаграма, коришћењем ИДЕФИХ методологије која представља креацију пројектанта информационог система. Наведена активност се односи на идентификацију ентитета, односно дефинисање објеката од интереса за посматрање и дефинисање веза и представља основну креирања ЕР модела поступком одозго на доле. Креирање ЕР дијаграма се изводи посредством следећих активности:

- идентификација кандидата за ентитете,
- идентификација веза,
- дефинисање ЕР модела и
- верификација ЕР модела.

Следећа активност се односи на креирање атрибута, која треба да описује особине претходно дефинисаних ентитета. Особине ентитета се дефинишу кроз идентификацију атрибута за сваки ентитет, дефинисање одговарајућих кључева и спровођења поступака нормализације. Наведена активност се изводи одозго на доле, тј. анализом докумената.

Активност креирања атрибута се изводи посредством следећих активности:

- дефинисање листе кандидата за атрибуте,
- дефинисање кључева,
- поступак нормализације и
- дефинисање атрибута.

Последња активност информационог моделирања се односи на дефинисање пословних правила, која представљају синтезу претходне две активности и треба да дефинише пословна ограничења и правила понашања.

5.1 Анализа реалног система „МАЛОПРОДАЈНОГ ОБЈЕКТА”

Дефинисање границе система представља тачку од које се креће при изради неког информационог система. При утврђивању граница система треба јасно дефинисати циљеве који морају да садрже следеће елементе:

- Зашто се процес моделира?
- Шта ће процес да прикаже?
- Шта ће корисник модела направити са њим?
- Чему слижи модел?

У овом раду разматрамо реални систем „Малопродајног објекта”. Тема овог рада је усмерена на свакодневне активности које се односе на радника и добављача једног малопродајног објекта, укључујући и финансијске стране. Поменуто представља границу система који ће се анализирати.

Како се у трговинским објектима писмено документују активности по питању потрошње, набавке, издавања робе и различитих активности везану за ову делатност, дошли смо на идеју да би се креирањем информационог система трајно документовали ови подаци, чиме би они постали доступни сваком запосленом раднику у сваком тренутку, а могле би се вршити и разне анализе и контроле од надређеног кадра. На такав начин запослено особље малопродајног објекта се не би оптеретило додатним послом, већ би уместо ручног попуњавања вршило унос података у рачунар, а расположивост података би тиме постала већа.

Сваки малопродајни објекат, који врши промет грађанима или потрошачима, је у обавези да поседује једну или више каса. Преко касе бележи се вредност продатог добра и пружених услуга потрошачима. У сваком малопродајном објекту (продавници) је запослено више радника (мада постоје малопродајни објекти са једним запосленим радником), различитих година старости. Сваки радник дефинисан је својим ИД бројем, именом, презименом, адресом, стручном спремом, датумом запослења и бројем телефона. Радници могу бити: магационер, касир и наручилац. Касир је задужен за рад на каси тј. пружање услуга потрошачима и за издавање рачуна за продату робу. Сваки касир дефинисан је својим ИД бројем и својом надлежношћу.

Наручилац је усмерен ка сарадњи са добављачем, у смислу наручивања потребне робе. Сарадња се може обавити са више различитих добављача. Сваки добављач доставља робу за једну или више продавница. Наручилац је дефинисан ИД бројем, описом посла и датумом наручивања.

Магационер је усмерен на вршење пописа за робу која је већ у продаји. Сваки магационер је дефинисан ИД бројем и квалификацијом.

Креирање информационог система ће пружити могућност запосленим радницима малопродајног објекта (продавнице), бржи увид у све активности везане за цео објекат, од наручивања робе, издавања рачуна, достављања робе, преко вршења пописа, до послова на каси.

Наставак рада је усмерен на описни начин функционисања реалног система, на креирање ентитета, атрибута, веза, дефинисање кардиналности веза и ограничења.

5.2 Идентификовање кандидата за ентитете

Након дефинисања граница система за који ће се пројектовати информациони систем, следи идентификација кандидата за ентитете. Идентификација кандидата за ентитете се врши детаљном анализом и разговором са запосленим особљем.

Ентитет је елемент који можемо једнозначно одредити и на такав начин га издвојити у скуп. Објекти модела који су једнозначно одређени називају се *јединствени ентитети*. Ентитети се графички представљају правоугаоником у коме се уписује назив типа ентитета у једнини, као што је приказано на слици 14.



Слика 14: Графичко представљање ентитета

Разликују се у општем случају самостални и општи ентитети. Самостални ентитет је од интереса сам за себе, његова егзистенција у бази не зависи од другог ентитета. Описни ентитет описује друге ентитете и одговара атрибутима тих ентитета чији домен није прост.

Дефинисањем границе система долазимо до идентификације објеката који ће бити од интереса за израду овог рада а који истовремено представљају кандидате за ентитете: ПРОДАВНИЦА, КАСА, РАДНИК, РАЧУН, ДОБАВЉАЧ, РОБА, КАСИР, МАГАЦИОНЕР, НАРУЧИЛАЦ, ПОПИС И ПОРУЏБИНА. У наставку рада ће бити дефинисано који од наведених ентитета спадају у самосталне, а који у описне ентитете.

5.3 Идентификација веза

Ентитети који су елементи истог скупа, као и елементи из различитих скупова могу међусобно стајати у различитим односима, тј. повезани различитим везама. Карактеристика веза између два или више ентитета је поседовање одређених односа, односно својстава. Свака веза може бити одређена једнозначно и према томе може се сматрати ентитетом.

Графички се веза приказује ромбом у коме се уписује назив типа везе, као што је приказано на слици број 15.



Слика 15: Графичко представљање везе

Извор: Милан Ерић, „Пројектовање информационих система и базе података”

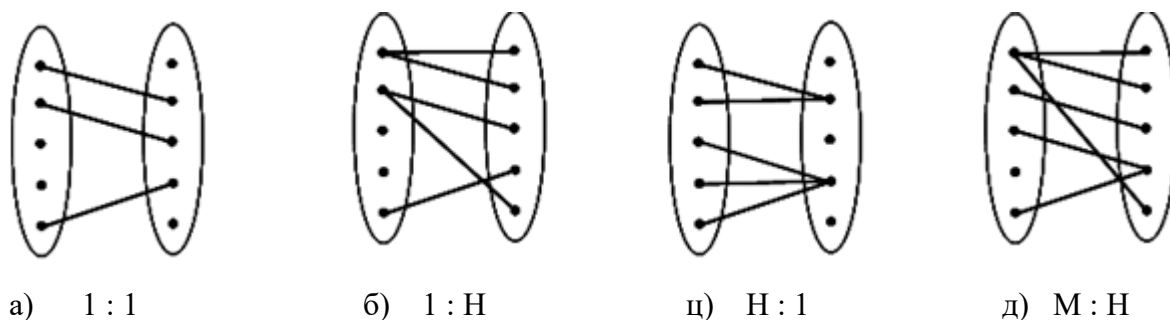
Представљањем везе ентитетом, добија се мешовита класа ентитет–веза, која се уводи у циљу решавања проблема представљања веза између веза. Врста везе се тада представља као класа ентитета што пружа могућност успоставе веза између релација. Графички се мешовита класа представља на следећи начин, слика 16.



Слика 16: Графички приказ мешовите везе

Извор: Милан Ерић, „Пројектовање информационих система и базе података”

Степен релације, кардиналност, односно степен везе између два ентитета је веома важна особина везе. Постоје три различита степена веза које могу постојати између два ентитета, слика 17.



Слика 17: Графички приказ кардиналности везе

Извор: Милан Ерић, „Пројектовање информационих система и базе података”

1:1 веза (један према један), означава да се један ентитет једног типа придружује једном ентитету другог типа.

1:N веза (један према више), означава да се један ентитет првог типа придружује већем броју ентитета другог типа, а да један ентитет другог типа само једном ентитету првог типа.

M:N веза (више према више), означава да се једном ентитету једног типа придружује више ентитета другог типа и обрнуто.

Ентитети могу бити и у вези сами са собом и такве везе називамо унарним везама. Ретко, али је ипак могуће, појављивање тројне и вишеструке везе, које повезују више ентитета. Најчешће се вишеструке везе преводе у бинарне увођењем новог ентитета уместо посматране везе.

Веза може бити тотална и парцијална. Један тип ентитета има тоталну везу, ако сваки ентитет тог типа учествује у бар једној вези, иначе, веза је парцијална.

Тотална веза се графички означава тачком (звездицом) на темену везе према објекту који има тоталну везу, слика 18.



Слика 18: Графички приказ тоталне везе

Извор: Милан Ерић, „Пројектовање информационих система и базе података”

Поред до сада дефинисаних типова веза и то чисте везе дефинисане називом, затим мешовите, постоје и специјалне везе којима се дефинише:

- егзистенцијална зависност, где је веза физичка (означава се са Е),
- идентификациона зависност, где је кључ једног ентитета део кључа слабог ентитета (означава се са И) и
- идентификационо- егзистенцијална зависност (означава се са Е&И).

У оквиру дефинисања специјалних веза дефинише се и појам слабог ентитета који се графички представља двоструким правоугаоником. Класа слабог ентитета је врста ентитета који је на неки начин завистан од другог ентитета. То су ентитети који за једнозначност тј. кључ захтевају и атрибуте (примарне кључеве) других ентитета са којима су у вези. Други извор слабих ентитета су ентитети настали заменом вишеструких веза бинарним, јер обично тако формиран ентитет захтева атрибуте других ентитета са којима је у вези за свој примарни кључ.

Зависност подтипа и зависност претхођења су још два типа веза између ентитета.

Генерализациона зависност или зависност подтипа је зависност између ентитета подтипа и њему надређеног ентитета. Ова веза (зависност) се означава са с или "ис" или "ис а" и графички се приказује троуглом.

5.3.1 Идентификација веза и дефинисање кардиналности за реални систем, „Малопродајног објекта“

На основу дефинисаних правила, тј. веза између објеката, у наставку рада је истакнуто дефинисање веза између ентитета и одговарајућих кардиналности која важе за те везе.

Полазимо од ентитета ПРОДАВНИЦА и његове везе са ентитетом КАСА, која је дефинисана називом ПОСЕДУЈЕ, као што је приказано на слици 19. Једна продавница може поседовати више каса, одакле следи да је кардиналност везе са стране КАСА (1,М), док бар једну касу мора поседовати једна ПРОДАВНИЦА, да би продавница радила, одакле следи кардиналност везе ПОСЕДУЈЕ са стране ентитета ПРОДАВНИЦА (1,1).



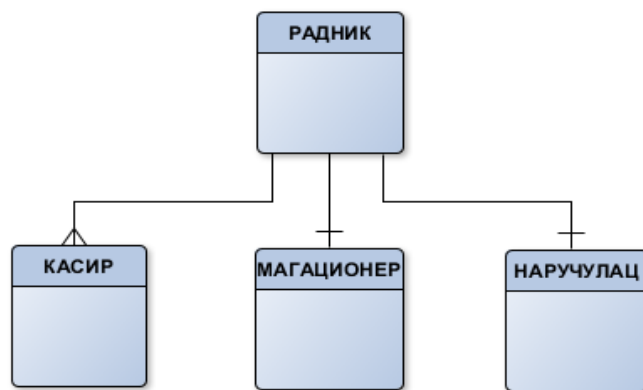
Слика 19: Приказ везе ПОСЕДУЈЕ

У једној продавници може бити ЗАПОСЛЕН-о више радника, одакле следи да је кардиналност везе са стране ентитета РАДНИК (1,М), док бар један радник мора бити запослен у једној продавници да би продавница радила, одакле следи кардиналност везе ЗАПОСЛЕН са стране ПРОДАВНИЦА (1,1), као што је приказано на слици 20.



Слика 20: Приказ везе ЗАПОСЛЕН

Ентитети КАСИР, МАГАЦИОНЕР и НАРУЧИЛАЦ су зависни ентитети од надређеног ентитета РАДНИК. У једној продавници је запослено више РАДНИКА, од којих један или више КАСИРА. Поред касира (с обзиром да се ради о малопродајном објекту) може бити запослен један (више) НАРУЧИЛАЦ и један (више) МАГАЦИОНЕР, као што је приказано на слици 21.



Слика 21: Приказ ентитета РАДНИК и њему подређених ентитета

Један касира може радити на више каса, одакле следи да је кардиналност везе са стране ентитета КАСА (1,М), док на једној каси може радити више касира, што указује на кардиналност везе РАДИ са стране ентитета КАСИР (1,Н), слика 22.



Слика22: Приказ везе РАДИ

Објекат (ентитет) из реалног света који смо узели у разматрање је РАЧУН. Један касир може да изда више рачуна, одакле следи да је кардиналност везе са стране ентитета РАЧУН (1,М), док један рачун може бити издат од стране више касир, што указује на кардиналност везе ИЗДАЈЕ са стране ентитета КАСИР (1,Н), што је приказано на слици 23.



Слика 23: Приказ везе ИЗДАЈЕ

Касир може издати један рачун за већу количину робе, одакле следи да је кардиналност везе са стране ентитета РОБА (1,М), док за мању количину робе се може издати више рачуна, што указује на кардиналност везе ЗА са стране ентитета РАЧУН (1,Н), што је приказано на слици 24.



Слика 24: Приказ везе ЗА

Једна продавница може набављати већу количину робе, што указује на кардиналност везе са стране РОБА (1,М), док мању количину робе може набављати више ПРОДАВНИЦА, што подразумева кардиналност везе НАБАВЉА са стране ентитета ПРОДАВНИЦА (1,Н), слика 25.



Слика 25: Приказ везе НАБАВЉА

Један наручилац наручује робу од више добављача, одакле следи да је кардиналност везе са стране ентитета ДОБАВЉАЧ (1,М), док један добављач може снабдевати једног наручиоца, одакле следи кардиналност везе НАРУЧУЈЕ ОД са стране ентитета НАРУЧИЛАЦ (1,1), слика 26.



Слика 26: Приказ везе НАРУЧУЈЕ ОД

Веза између ентитета ДОБАВЉАЧ и ПОРУЏБИНА дефинисана је називом ДОСТАВЉА. Један добављач може достављати више поруџбина (у облику робе), што указује на кардиналност везе ентитета ПОРУЏБИНА (1,М), док једна поруџбина може бити достављена од стране више добављача, што указује на кардиналност везе ДОСТАВЉА са стране ентитета ДОБАВЉАЧ (1,Н), као што је приказано на слици 27 и 28.



Слика 27: Приказ везе ДОСТАВЉА



Слика 28: Приказ везе У ОБЛИКУ

Између ентитета МАГАЦИОНЕР и ПОПИС постоји веза ВРШИ који је са стране ентитета МАГАЦИОНЕР кардиналности (1,1), а са стране ентитета ПОПИС кардиналности (1,M). Један магационер може вршити више попис, док један попис може вршити један магационер, што је приказано на слици 29.



Слика 29: Приказ везе ВРШИ

Веза између ентитета ПОПИС и ентитета РОБА, дефинисана је називом ПРЕОСТАЛЕ, која са своје стране ентитета ПОПИС има кардиналност (1,M), док са стране ентитета РОБА има кардиналност (1,M). Попис се може вршити за већу количину преостале робе, док такође се може вршити и за мање количине робе, као што је приказано на слици 30.



Слика 30: Приказ везе ПРЕОСТАЛЕ

5.4 Дефинисање атрибута

Сваки ентитет дефинисан је особинама који се називају атрибути. Заједно, атрибути и вредности атрибута представљају особине ентитета. Атрибути се одређују из докумената који се прикупљају у посматраном реалном систему а који је предмет проучавања. Дефинисање атрибута се изводи кроз три корака:

- идентификација атрибута,
- извршење ревизије атрибута и
- дефинисање атрибута.

Идентификација атрибута, дефинише се на основу захтева корисника и пословне документације за развој ИС. Алокација атрибута се изводи у зависности да ли је атрибут завијан од кључа или је описни.

Основна правила која се користе при дефинисању атрибута су:

- Сваки ентитет има произвољан број атрибута, што значи да нема ограничења у броју атрибута.
- Одређени атрибут припада један и само једном ентитету, тако да исти атрибут не може бити описан у оквиру два или више ентитета.
- Свако појављивање ентитета има вредност за све атрибуте тог ентитета.
- Атрибут одређеног појављивања ентитета може имати само једну вредност, па примерак ентитета за одређени атрибут има једну вредност.
- Сваки атрибут представља једну одређену чињеницу, тако да и свако значење вредности атрибута мора имати једно доследно значење.

5.5 Дефинисање кључева

Кључ је атрибут или комбинација атрибута који једнозначно одређују ентитет из скупа. Обично је то један, *прост* или елементаран кључ, или комбинација два или више атрибута, *сложен кључ*. Избор кључа је битан за пројектовање целе базе података јер мора једнозначно да идентификује чињенице сачуване у бази података и једино помоћу њега се може сигурно приступити жељеном ентитету.

У једном ентитету може постојати више комбинација атрибута који задовољавају дефиницију кључа. Један од кандидата који се изабере да практично служи за једнозначну идентификацију ентитета се назива *примарни кључ*, а сви остали неизабрани кандидати се тада називају *алтернативни кључеви*.

У многим базама података, у нашој земљи се као примарни кључ користи ЈМБГ (јединствени матични број грађана), јер он једнозначно одређује сваког човека тј. не постоји два човека исте земље са истим ЈМБГ. Графички, примарни кључ се од осталих атрибута разликује по томе што је подвучен.

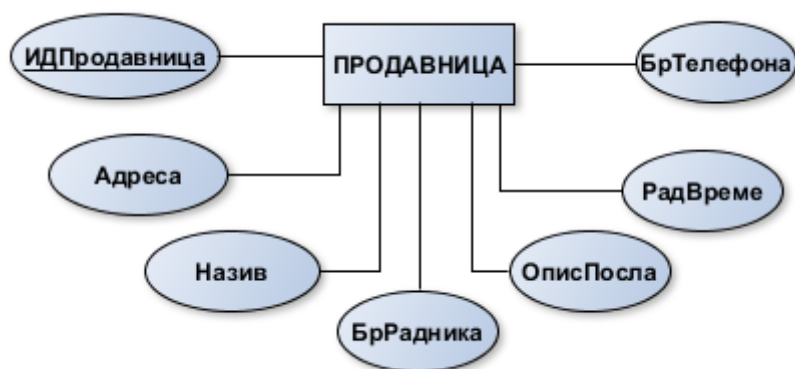
Особине примарног кључа:

- *Једнозначност* - две појаве ентитета у скупу не смеју имати исту вредност примарног кључа, без обзира да ли је кључ прост или сложен и
- *Минималност* - ако је кључ сложен и ако се из њега изостави било који атрибут, губи се особина једнозначности.

За кључ се често, у недостатку јединственог идентификатора, уводи нови – ИД број, који се додељује свакој појави ентитета, водећи рачуна да нема дуплирања.

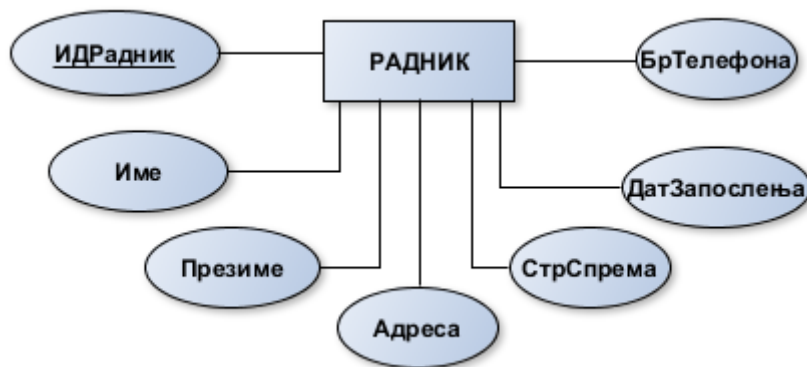
5.5.1 Идентификација атрибута и кључева на примеру реалног система

Полазимо од ентитета ПРОДАВНИЦА. За сваку продавницу је позната њена адреса, назив, број телефона, број радника, радно време, опис посла, који уједно и означавају њене атрибуте. Поред наведених атрибута (као што је већ споменуто) уводи се и атрибут ИД_Продавница који ће бити примарни кључ ентитета ПРОДАВНИЦА. Графички, примарни кључ се од осталих атрибута разликује по томе што је подвучен, као на слици 31.



Слика31: Приказ атрибута ентитета ПРОДАВНИЦА

Ентитет РАДНИК има следеће атрибуте: Име, Презиме, Адреса, СтрСпрема, БрТелефона, ДатЗапослења и ИД_Радник који једнозначно одређује радника из скупа радника и означава примарни кључ, слика 32.



Слика 32: Приказ атрибута ентитета РАДНИК

Ентитет КАСА има следеће атрибуте: ДатПроизводње, Произвођач, КапМеморије, Језик. ИД_Каса је атрибут који је уведен и који представља примарни кључ.

Ентитет РОБА има следеће атрибуте: ИД_Роба, Назив, Количина, Цена, МахКоличина, МинКоличина. Као и код ентитета КАСА, уведен је атрибут ИД_Роба који представља примарни кључ ентитета РОБА.

Ентитет ДОБАВЉАЧ има следеће атрибуте: ИД_Добављач, НазДобављач, ТелДобављач и Адреса.

Ентитет РАЧУН има следеће атрибуте: ИД_Рачун, ВреРачун, ДатРачун, ПДВ, Основица и Износ.

Ентитет МАГАЦИОНЕР има следеће атрибуте: ИД_Магационер и Квалификација.

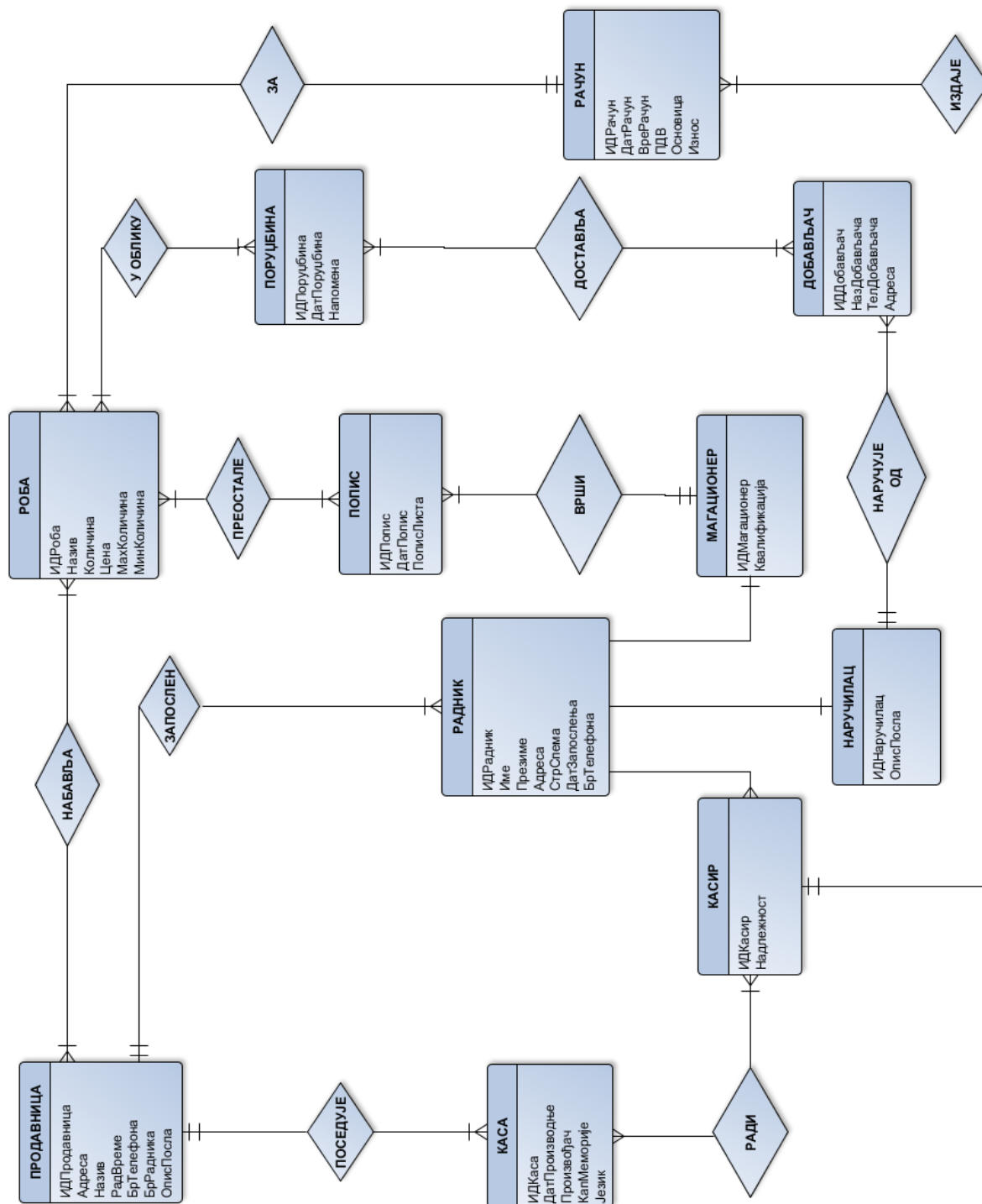
Ентитет НАРУЧИЛАЦ има следеће атрибуте: ИД_Наручилац и ОписПосла.

Ентитет КАСИР има следеће атрибуте: ИД_Касир и Надлежност.

Ентитет ПОПИС има следеће атрибуте: ИД_Попис, ДатПопис и ПописЛиста.

Ентитет ПОРУЏБИНА има следеће атрибуте: ИД_Поруџбина, ДатПоруџбина и Напомена.

5.5.2 ЕР дијаграм модела базе података за малопродајни објект



Слика 33: ЕР дијаграм „МАЛОПРОДАЈНОГ ОБЈЕКТА“

Наставак рада је усмерен ка релационим моделу базе података и правилима за преводње ЕР модела у релациони модел. На основу поменутих правила ће се извршити преводње ЕР модела малопродајног објекта у релациони модел базе података.

6. Релациони модел

Основне карактеристике релационог модела су:

- *једноставна и природна репрезентација* (модел је представљен скупом табела, а табеле су структура најприхватљивија за комуникацију са корисницима),
- могућа је *формално-математичка интерпретација модела* (табела се може дефинисати као математичка релација, а затим искористити све предности ове формалне дефиниције) и
- *богат математички алат* служи као основа за развој базе.

6.1 Структура релационог модела

Концепт структуре релационог модела се састоји од:

- **домена** – скуп дозвољених вредности истог типа.
- **појам релација** – релација је један скуп Декартових производа n домена атрибута ($n > 0$). Релација P степена n се дефинише као произвољни коначни подскуп Декартовог производа. Релација се представља у облику дводимензионалне табеле у којој n атрибута одговара колонама табеле, док шема релације прецизира име релације, листу њених атрибута и њихових домена.
- **атрибута** – представљају особине ентитета.
- **степен релација/кардиналност** – степен веза између ентитета.
- **n -торка** – дефинисана не само доменом већ и примарним кључем, представља ред у табели релације.
- **примарни кључ** – X је кључ релације P , ако испуњава следеће услове:
 - *X функционално одређује све атрибуте релације P ,*
 - *Ни један прави подскуп од X нема ту особину.*
- **страни кључ (секундарни, спољни)**, - је атрибут који у датој релацији није примарни кључ, али је примарни кључ у некој другој релацији. Страни кључ не мора да буде кандидат за кључ те исте релације.
- **Интегритет податка** – односи се на услове које треба да задовоље подаци у бази, да би стање било ваљано. Услови интегритета могу бити:
 - **Специфични**
 - **Општи**

У наставку рада ће се дефинисати домен атрибута који се појављује код ентитета анализираног ЕР дијаграма.

Домени атрибута за ентитет ПРОДАВНИЦА:

ИД_Продавница – позитиван цео број,
Адреса – скуп коначних низова алфабетских знакова,
Назив – скуп коначних низова алфабетских знакова,
РадВреме – скуп бројева,
БрТелефона – скуп коначних низова алфабетских знакова,
БрРадника – скуп бројева и
ОписПосла — скуп коначних низова алфабетских знакова.

Домени атрибута за ентитет КАСА:

ИД_Каса – позитиван цео број,
ДатПроизводње – скуп бројева,
КапМеморије – скуп бројева и
Језик – скуп коначних низова алфабетских знакова.

Домени атрибута за ентитет РАДНИК:

ИД_Радник – позитиван цео број,
Име – скуп коначних низова алфабетских знакова,
Презиме – скуп коначних низова алфабетских знакова,
Адреса – скуп коначних низова алфабетских знакова,
СтрСпрена – скуп коначних низова алфабетских знакова,
ДатЗапослења – скуп бројева и
БрТелефона – скуп коначних низова алфабетских знакова.

Домени атрибута за ентитет КАСИР:

ИД_Касир – позитиван цео број и
Надлежност – скуп коначних низова алфабетских знакова.

Домени атрибута за ентитет МАГАЦИОНЕР:

ИД_Магационер – позитиван цео број и
Квалификације – скуп коначних низова алфабетских знакова,

Домени атрибута за ентитет НАРУЧИЛАЦ:

ИД_Наручилац – позитиван цео број и
ОписПосла – скуп коначних низова алфабетских знакова.

Домени атрибута за ентитет РОБА:

ИД_Роба – позитиван цео број,
Назив – скуп коначних низова алфабетских знакова,
Количина – скуп бројева,
Цена – скуп бројева,
МахКоличина – скуп бројева и
МинКоличина – скуп бројева.

Домени атрибута за ентитет ДОБАВЉАЧ:

ИД_Добављач - позитиван цео број,
НазДобављач - скуп коначних низова алфабетских знакова,
ТелДобављач - скуп бројева и
Адреса – скуп коначних низова алфабетских знакова и бројева.

Домени атрибута за ентитет РАЧУН:

ИД_Рачун –позитиван цео број,

ДатРачун – скуп бројева,

ВреРачун – скуп бројева,

Износ – скуп бројева,

НазАртикла – скуп коначних низова алфаветских знакова и

ЦенаАртикла – скуп бројева.

Домени атрибута за ентитет ПОПИС:

ИД_Попис – позитиван цео број,

ДатПопис – скуп бројева и

ПописЛиста - скуп коначних низова алфаветских знакова.

Домени атрибута за ентитет ПОРУЏБИНА:

ИД_Поруџбина- позитиван цео број,

ДатПоруџбина- скуп бројева и

Напомена - скуп коначних низова алфаветских знакова.

6.2 Правила превођење ЕР модела у релациони модел

При превођењу једног модела податка у други не значи да сваком концептуструктуре једног модела мора одговарати концепт структуре другог модела, свакој операцији једног модела да мора одговарати операција другог модела и сваком експлицитном ограничењу једног модела мора одговарати експлицитно ограничење другог модела.

Превођење ЕР модела у релациони модел података се изводи на следећи начин:

- Део структуре ЕР дијаграма се представља релационом шемом.
- Ограничења, операције и део структуре ЕР модела се представљају операцијама за очување интегритета дефинисања над релационим моделом, а имплементирају коришћењем средстава одабраног СУБП.

1. Правила за превођења ентитета су:

1.1. Сваки објекат (ентитет) из ЕР дијаграма постаје шема релације. Назив типа ентитета постаје назив шеме релације. Обележја (атрибути) објекта (ентитета) постају обележја шеме релације. За основне објекте, идентификатор објекта постаје примарни кључ шеме релације. Односно:

- Сваки јак ентитет (објекат, класа ентитета) ЕР модела постаје шема релације са истим именом.
- Сваки атрибут ентитета постаје атрибут шеме релације под истим именом.
- Кључни атрибут (примарни кључ) односно идентификатор ентитета постаје примарни кључ шеме релације.

1.2. Сваки слаб објекат постаје шема релације. Назив типа објекта постаје назив шеме релације. Примарни кључ надређеног ентитета постаје једно од обележја шеме релације, која одговара слабом типу ентитета.

1.3. Објекат надтипа постаје шема релације. Назив надтипа постаје назив шеме релације. Обележја надтипа постају обележја шеме релације. Идентификатор надтипа постаје кључ шеме релације.

1.4. Објекат подтипа постаје шема релације. Назив подтипа постаје назив шеме релације. Обележја подтипа постају обележја шеме релације. Идентификатор надтипа постаје кључ шеме релације.

2. Правила за превођење бинарне везе

2.1. Везе кардиналности 1:1 по правилу немају обележја. Обележја која би могла бити приписана самој вези, у суштини су обележја једног од објекта који учествују у тој вези.

2.1.1. Везе кардиналности (1,1) : (1,1) и оба објекта која у њој учествују преводимо у једну шему релације, чија су обележја једног и другог објекта. Кандидат за кључ у овој шеми релације су идентификатори једног и другог објекта који су у вези.

2.1.2. Везе са кардиналношћу (0,1) : (1,1) и објекте у вези, преводимо у две шеме релације, с тим што се идентификатор једног од објекта, који су у вези, убаци за обележје друге шеме релације. Веза се представља спољним кључем.

2.1.3. Везе са кардиналношћу (0,1) : (0,1) креирају се у три шеме. Обележја у шеми релације која одговара вези су и идентификатори објекта који су у вези и оба су кандидати за кључ.

2.2. Везе кардиналности 1:M

2.2.1. Веза кардиналитета $(1,1) : (0,M)$ и $(1,1) : (1,M)$ не постоји посебна шема релације. Идентификатор објекта са стране за коју је горња граница кардиналитета пресликавања M постаје обележје шеме релације која одговара објектима са чије стране је максимална кардиналност 1.

2.2.2. Веза између надређеног и слабог објекта као и веза између подтипа и надтипа не постају посебне шеме релације. Оне су већ остварене правилима 1.2, 1.3 и 1.4.

2.2.3. Веза кардиналитета $(0,1) : (0,M)$ и $(0,1) : (1,M)$ постаје посебна шема релације. Обележја ове шеме релације су идентификатори обележја који су у вези док кључ шеме релације је идентификатор обележја са чије стране је максимална кардиналност 1.

2.3. Веза кардиналности N : M

2.3.1. Веза кардиналитета $(0,M) : (0,M)$, $(1,M) : (0,M)$ и $(1,M) : (1,M)$ постаје посебна шема релације. Обележја ове шеме релације су идентификатори објекта који су у вези, а кључ шеме релације је сложен кључ који се састоји од идентификатора објекта који су у вези.

2.3.2. Герунд или мешовити тип објекта - везе се посматра на исти начин као и одговарајућа веза. Уколико веза постаје обележје, она постаје обележје шеме релације када се веза уводи у посебну шему релације или се укључи у шему релације у коју се уписује спољни кључ.

2.4. Правила за ударне везе

2.4.1. Ударном везом се сматра веза између два објекта истог типа. Код ударних веза типа $(1:1)$ парцијалност само на једној страни везе, односно тоталност само на једној страни везе, не би имало смисла. При превођењу ударних веза с обзира да би спољни кључ у шеми релације имао исти назив као и примарни кључ, потребно је његово преименовање.

6.2.1 Превођење ЕР модела малопродајног објекта у релациони модел

На основу правила 1.1 сви типови ентитета постају шеме релације. Име типа ентитета постаје име шема релације. Обележја типа објекта постају обезбеђење шеме релације. За основне објекте идентификатор објекта (ентитета) постаје кључ шеме релације.

ПРОДАВНИЦА (ИД_Продавница, Назив, Адреса, РадВреме, БрТелефона, БрРадника, ОписПосла)

КАСА (ИД_Каса, ДатПроизводње, Произвођач, КапМеморије, Језик).

ДОБАВЉАЧ (ИД_Добављач, НазДобављач, ТелДобављач, Адреса).

РАЧУН(ИД_Рачун, ДатРачун, ВреРачун, ПДВ, Основица, Износ).

РОБА (ИД_Роба, Назив, Количина, Цена, МахКоличина, МинКоличина).

РАДНИК (ИД_Радник, Име, Презиме, Адреса, СтрСпрема, ДатЗапослења, БрТелефона).

ПОПИС (ИД_Попис, ДатПопис, ПописЛиста).

ПОРУЏБИНА (ИД_Поруџбина, ДатПоруџбина, Напомена).

Према правилу 1.2 сваки слаб објекат постаје шема релације. Назив типа објекта постаје назив шеме релације. Примарни кључ надређеног ентитета постаје једно од обележја шеме релације, која одговара слабом типу ентитета.

КАСИР (ИД_Касир, ИД_Радник, Надлежност).

МАГАЦИОНЕР (ИД_Магационер, ИД_Радник, Квалификација).

НАРУЧИЛАЦ (ИД_Наручилац, ИД_Радник, ОписПосла)

У зависности од кардиналности, на типове веза се примењују различита правила за превођење.

Веза „ ПОСЕДУЈЕ “

На слици 19. приказана је веза између ентитета ПРОДАВНИЦА и КАСА и дефинисана називом „ПОСЕДУЈЕ“, кардиналности (1,1) : (1,М), која по правилу 2.2.1 не постаје посебна шема релације. Идентификатор објекта КАСА овом приликом постаје атрибут шеме ПРОДАВНИЦА.

ПРОДАВНИЦА (ИД_Продавница, Назив, Адреса, РадВреме, БрТелефона, БрРадника, ОписПосла, ИД_Каса)

КАСА (ИД_Каса, ДатПроизводње, Произвођач, КапМеморије, Језик)

Веза „ ЗАПОСЛЕН “

На слици 20. приказана је веза ентитета ПРОДАВНИЦА и РАДНИК, и дефинисана називом „ЗАПОСЛЕН“, кардиналности (1,1) : (1,М), која по правилу 2.2.1 не постаје посебна шема релације. Идентификатор објекта РАДНИК овом приликом постаје атрибут шеме ПРОДАВНИЦА.

ПРОДАВНИЦА (ИД_Продавница, Назив, Адреса, РадВреме, БрТелефона, БрРадника, ОписПосла, ИД_Радник)

РАДНИК(ИД_Радник, Име, Презиме, Адреса, СтрСпрема, ДатЗапослења, БрТелефона)

Веза „ РАДИ “

На слици 22. приказана је веза ентитета КАСА и КАСИР и означена са „РАДИ“, кардиналности (1,Н) : (1,М), која по правилу 2.3.1. постаје посебна шема релације. Обележја ове шеме релације су идентификатори објекта који су у вези, а кључ шеме релације је сложен кључ који се састоји од идентификатора објекта који су у вези.

РАДИ (ИД_Каса, ИД_Касир)

КАСА (ИД_Каса, ДатПроизводње, Произвођач, КапМеморије, Језик)

КАСИР (ИД_Касир, Надлежност)

Веза „ ИЗДАЈЕ “

На слици 23. приказана је веза ентитета КАСИР и РАЧУН и дефинисана називом „ИЗДАЈЕ“, кардиналности (1,1) : (1,М), која по правилу 2.2.1 не постаје посебна шема релације. Идентификатор објекта РАЧУН овом приликом постаје атрибут шеме КАСИР.

КАСИР (ИД_Касир, Надлежност, ИД_Рачун)

РАЧУН(ИД_Рачун, ДатРачун, ВреРачун, ПДВ, Основица, Износ)

Веза „ЗА“

На слици 24. приказана је веза ентитета РАЧУН и РОБА и дефинисана називом „ЗА“, кардиналности (1,1) : (1,М), која по правилу 2.2.1 не постаје посебна шема релације. Идентификатор објекта РОБА овом приликом постаје атрибут шеме РАЧУН.

РАЧУН(ИД_Рачун, ДатРачун, ВреРачун, ПДВ, Основица, Износ, ИД_Роба)
РОБА (ИД_Роба, Назив, Количина, Цена, МахКоличина, МинКоличина)

Веза „НАБАВЉА“

На слици 25. приказана је веза ентитета РОБА и ПРОДАВНИЦА и дефинисана називом „НАБАВЉА“, кардиналности (1,Н) : (1,М), која по правилу 2.3.1. постаје посебна шема релације. Обележја ове шеме релације су идентификатори објекта који су у вези, а кључ шеме релације је сложен кључ који се састоји од идентификатора ентитета који су у вези.

НАБАВЉА (ИД_Роба, ИД_Продавница)
ПРОДАВНИЦА (ИД_Продавница, Назив, Адреса, РадВреме, БрТелефона, БрРадника, ОписПосла)
РОБА (ИД_Роба, Назив, Количина, Цена, МахКоличина, МинКоличина)

Веза „НАРУЧУЈЕ ОД“

На слици 26. приказана је веза ентитета НАРУЧИЛАЦ и ДОБАВЉАЧ и дефинисана називом „НАРУЧУЈЕ ОД“, кардиналности (1,1) : (1,М), која по правилу 2.2.1 не постаје посебна шема релације. Идентификатор објекта ДОБАВЉАЧ овом приликом постаје атрибут шеме НАРУЧИЛАЦ.

НАРУЧИЛАЦ (ИД_Наручилац, ДатНаручивања, ОписПосла, ИД_Добављач)
ДОБАВЉАЧ (ИД_Добављач, НазДобављач, ТелДобављач, Адреса)

Веза „ДОСТАВЉА“

На слици 27. приказана је веза ентитета ДОБАВЉАЧ и ПОРУЏБИНА и дефинисана називом „ДОСТАВЉА“ кардиналности (1,Н) : (1,М), која по правилу 2.3.1 постаје посебна шема релације. Идентификатор објекта ПОРУЏБИНА приликом постаје атрибут шеме ДОБАВЉАЧ.

ДОСТАВЉА (ИД_Добављач, ИД_Поруџбина)
ДОБАВЉАЧ (ИД_Добављач, НазДобављач, ТелДобављач, Адреса)
ПОРУЏБИНА (ИД_Поруџбина, ДатПоруџбина, Напомена).

Веза „У ОБЛИКУ“

На слици 28. приказана је веза ентитета ПОРУЏБИНА и РОБА и дефинисана називом „У ОБЛИКУ“ кардиналности (1,Н) : (1,М), која по правилу 2.3.1 постаје посебна шема релације. Идентификатор објекта РОБА овом приликом постаје атрибут шеме ПОРУЏБИНА.

У ОБЛИКУ (ИД Поручбина, ИД Роба)

ПОРУЏБИНА (ИД Поручбина, ДатПоручбина, Напомена)

РОБА (ИД Роба, Назив, Количина, Цена, МахКоличина, МинКоличина)

Веза „ВРШИ“

На слици 29. приказана је веза ентитета МАГАЦИОНЕР и ПОПИС и дефинисана називом „ВРШИ“ кардиналности (1,1) : (1,М), која по правилу 2.2.1 не постаје посебна шема релације. Идентификатор објекта ПОПИС овом приликом постаје атрибут шеме МАГАЦИОНЕР.

МАГАЦИОНЕР (ИД Магационер, Квалификација, ИД Попис)

ПОПИС (ИД Попис, ДатПопис, ПописЛиста).

Веза „ПРЕОСТАЛЕ“

На слици 30. приказана је веза ентитета ПОПИС и РОБА и дефинисана називом „ПРЕОСТАЛЕ“ кардиналности (1,Н) : (1,М), која по правилу 2.3.1 постаје посебна шема релације. Идентификатор објекта РОБА овом приликом постаје атрибут шеме ПОПИС.

ПРЕОСТАЛЕ (ИД Попис, ИД Роба)

ПОПИС (ИД Попис, ДатПопис, ПописЛиста)

РОБА (ИД Роба, Назив, Количина, Цена, МахКоличина, МинКоличина)

7. Апликативно моделирање

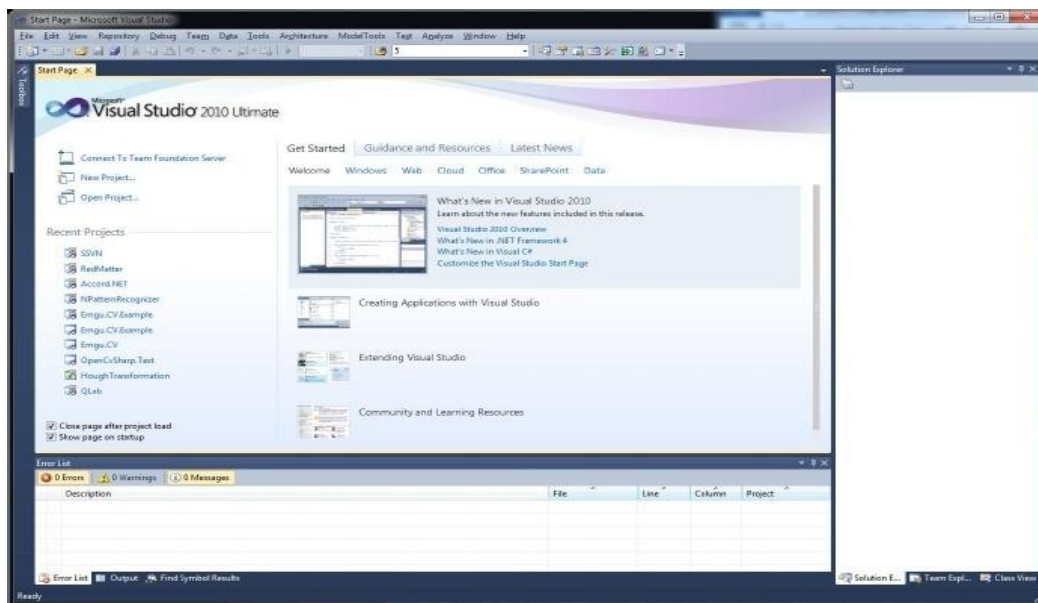
Апликативно моделирање посматра се са становишта изабраног система за управљање базама података. Ова активност се посматра кроз следеће три подактивности:

- Дефинисање физичког дизајна,
- Генерисање шеме базе података и
- Израда апликације.

7.1 Дефинисање физичког дизајна

Дефинисање физичког дизајна разматра проблематику везану за изградњу система за управљање базама података. Циљ ове фазе је избор програма за израду апликације као и превођење логичког у физички дизајн, односно превођење ентитета у табеле, атрибуте у колоне. За креирање апликације користимо **Microsoft Visual Studio**.

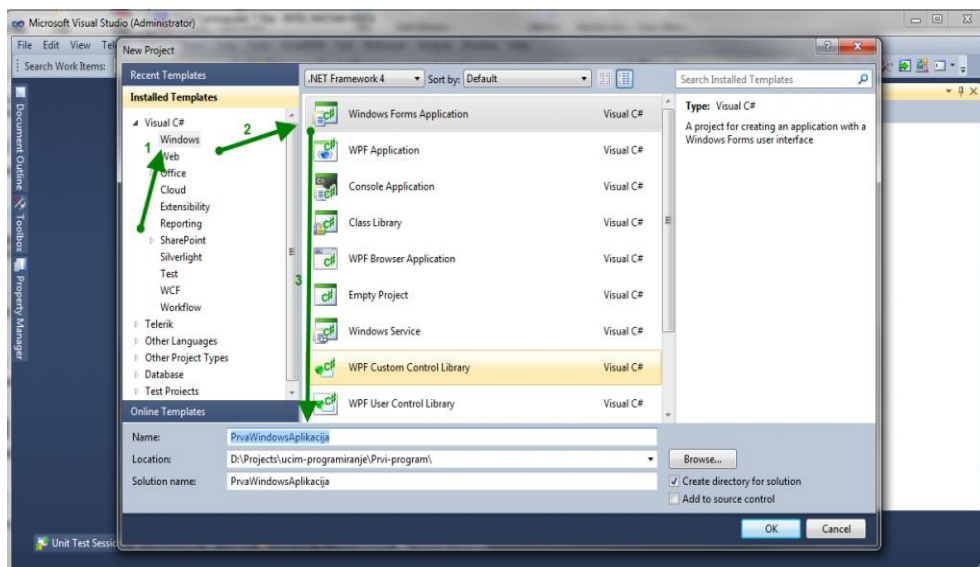
Microsoft Visual Studio 2010 представља радно окружење за развој софтверских апликација. Покретање окружења се врши кроз главни мени: : Start → All Programs → Microsoft Visual Studio 2010 → Microsoft Visual Studio 2010. Након чега се отвара страна која је приказана на слици:



Слика 34: Microsoft Visual Studio 2010

Почетна страна у свом централном делу садржи листу линкова ка ресурсима који се налазе на интернету. Она такође садржи и линкове ка последње реализованим пројектима у развојном окружењу. Уколико желимо да поново отворимо ову страницу након што смо је затворили, потребно је из менија отворити View → Start Page. Да би смо креирали нови пројекат из менија треба одабрати File → New → Project.

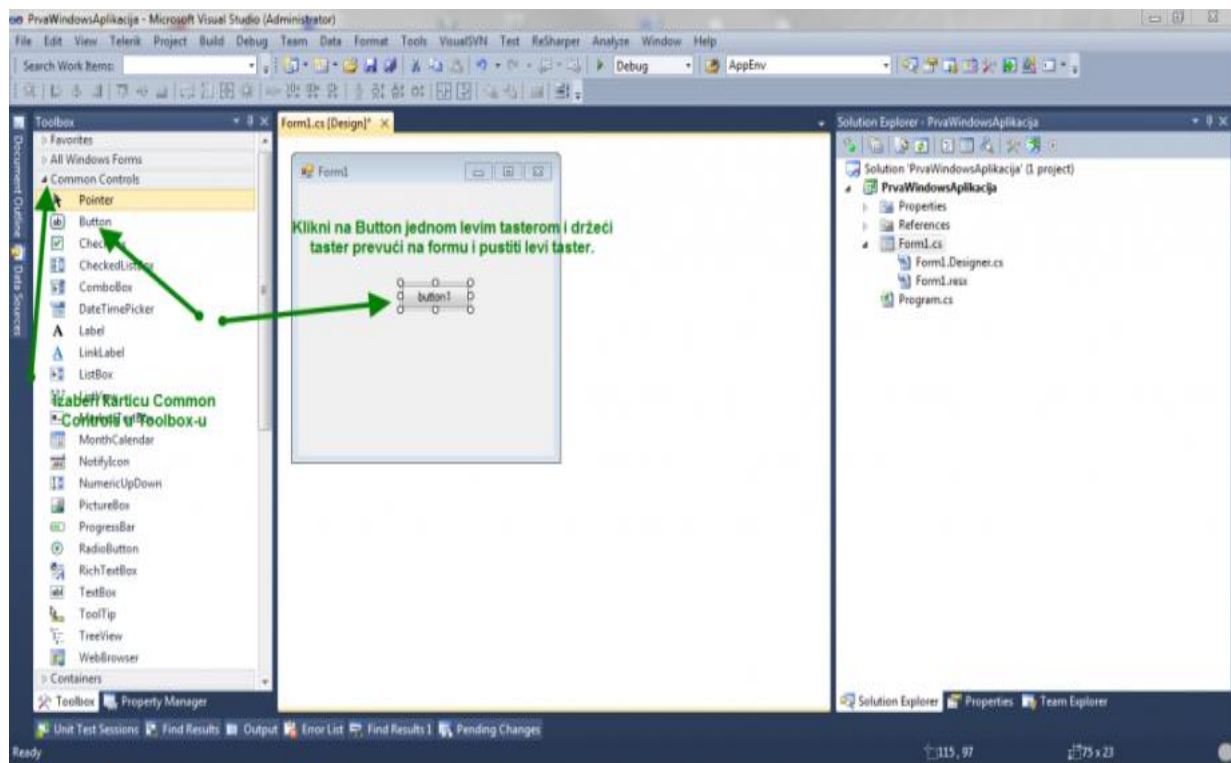
У прозору који се након тога отвара у дела Installed Templates бира се Visual C#, а у централном прозору бирамо Windows Forms Application. У пољу Name уписује се име новог пројекта (РАДНИК). На крају потврђујемо одабир кликом на тастер ОК, као што је приказано на слици 35:



Слика 35: Изглед форме након креирања новог пројекта

Такође, неопходно је сачувати локацију до креираног пројекта, тако што означимо целу путању, као што је приказано на слици 35, за поље Name и ископирамо у претходно отворен *Notepad*.

Путања се налази у пољу *Location*.



Слика36: Изглед форме након додавања дугмета

Слика 36, приказује изглед радне површине Visual Studio-а након превлачења дугмета из картице *Toolbox* и његово позиционирање на форми. Након овога потребно је два пута кликнути на дугме *button1* након чега ће Visual studio отворити прозор за куцање кода. Дакле, након што два пута кликнемо на дугме и упишемо код.

Након избора програма, прелазимо на превођење ентитета у табеле, атрибута у колоне, као и постављање одговарајућих ограничења (табела 2).

Табела 2: Приказ ентитета - табела, атрибута - релација.

Ентитети	Табеле	Атрибути	Колоне
Продавница	Продавница	ИДПродавница (ПК)	ИДПродавница - integer, Not null
		Адреса	Адреса – text (100)
		Назив	Назив – text (100)
		Радно време	РадВреме – datetime
		Број телефона	БрТелефон– text(50)
		Број радника	БрРадника– integer
		Опис посла	ОписПосла – text (100)
Каса	Каса	ИДКаса (ПК)	ИДКаса – integer, Not null
		Датум производње	ДатПроизводње – datetime
		Произвођач	Произвођач - text (100)
		Капацитет меморије	КапМеморија – integer
		Језик	Језик – text (100)
Касир	Касир	ИДКасир (ПК)	ИДКаса – integer, Not null
		Надлежност	Надлежност - text (100)
Наручилац	Наручилац	ИДНаручилац (ПК)	ИДНаручилац – integer, Not null
		Опис посла	ОписПосла - text (100)
Магационер	Магационер	ИДМагационер (ПК)	ИДМагационер – integer, Not null
		Квалификација	Квалификација - text (100)
Радник	Радник	ИДРадник (ПК)	ИДРадник – integer, Not null
		Име	Име - text (100)
		Презиме	Презиме - text (100)
		Адреса	Адреса –text (100)
		Стручна спрема	СтрСпрема - text (100)
		Датум запослења	ДатЗапослења – datetime
		Број телефона	БрТелефона –text(50)

Попис	Попис	ИДПопис (ПК)	ИДПопис –integer, Not null
		Датум пописа	ДатПопис – datetime
		Пописна листа	ПописЛиста –text (100)
Роба	Роба	ИДРоба (ПК)	ИДРоба –integer, Not Null
		Назив	Назив –text (100)
		Количина	Количина –float
		Цена	Цена –float
		Максимална количина	МахКоличина - float
		Минимална количина	МинКоличина - float
Добављачи	Добављачи	ИДДобављач (ПК)	ИДДобављач – integer, Not Null
		Назив доваљвача	НазДобављач – text (100)
		Телефон добављача	ТелДобављач – text(50)
		Адреса	Адреса – text (100)
Поруџбина	Поруџбина	ИДПоруџбина	ИДПоруџбина – integer, Not null
		Датум поруџбине	ДатПоруџбина- datetime
		Напомена	Напомена – text (100)
Рачун	Рачун	ИДРачун (ПК)	ИДРачун – integer, Not null
		Датум рачуна	ДатРачун – datetime
		Време рачуна	ВреРачун – date time
		Пдв	Пдв – float (100)
		Основица	Основица – float
		Износ	Износ – float

7.2 Генерисање шеме базе података

Шему базе података чине физичке табеле, колоне и релације. Процес генерисања шеме базе података из логичког модела податка назива се директни инжињеринг. Када се генерише шема базе података, ентитети прелазе у табеле, атрибути у колоне, а везе у релације идефинише се референцијални интегритет.

У наставку је дат код којим је креирана база:

```

/*****/
/*      TabelaProdavnica      */
/*****/
CREATE TABLE [Prodavnica](
    [IDProdavnica] int NOT NULL,
    [Adresa] nvarchar(100),

```

```
[Naziv] nvarchar(100),
[RadnoVreme] nvarchar(100),
[BrTelefona] nvarchar(50),
[BrRadnika] int,
[OpisPosla] nvarchar(100),
CONSTRAINT [PK_Prodavnica] PRIMARY KEY ([IDProdavnica])
)

/*****/
/*      TabelaRadnik      */
/*****/
CREATE TABLE [Radnik](
    [IDRadnik] int NOT NULL,
    [Ime] nvarchar(100),
    [Prezime] nvarchar(100),
    [Adresa] nvarchar(100),
    [StrSprema] nvarchar(100),
    [DatumZaposlenja] datetime,
    [BrTelefona] nvarchar(100),
    CONSTRAINT [PK_Radnik] PRIMARY KEY ([IDRadnik])
)

/*****/
/*      TabelaKasir      */
/*****/
CREATE TABLE [Kasir](
    [IDKasir] int NOT NULL,
    [Nadleznost] nvarchar(100),
    CONSTRAINT [PK_Kasir] PRIMARY KEY ([IDKasir])
)

/*****/
/*      TabelaMagacioner      */
/*****/
CREATE TABLE [Magacioner](
    [IDMagacioner] int NOT NULL,
    [Kvalifikacija] nvarchar(100),
    CONSTRAINT [PK_Magacioner] PRIMARY KEY ([IDMagacioner])
)

/*****/
/*      TabelaNaručilac      */
/*****/
CREATE TABLE [Narucilac](
    [IDNarucilac] int NOT NULL,
    [OpisPosla] nvarchar(100),
    CONSTRAINT [PK_Narucilac] PRIMARY KEY ([IDNarucilac])
```

)

```
CREATE TABLE [Kasa](
    [IDKasa] int NOT NULL,
    [DatumProizvodnje] datetime,
    [Proizvodjac] nvarchar(100),
    [KapMemorije] int,
    [Jezik] nvarchar(100),
    CONSTRAINT [PK_Kasa] PRIMARY KEY ([IDKasa])
)
```

```
CREATE TABLE [Dobavljac](
    [IDDobavljac] int NOT NULL,
    [NazivDobavljava] nvarchar(100),
    [TelefonDobavljava] nvarchar(100),
    [Adresa] nvarchar(100),
    CONSTRAINT [PK_Dobavljac] PRIMARY KEY ([IDDobavljac])
)
```

```
CREATE TABLE [Roba](
    [IDRoba] int NOT NULL,
    [Naziv] nvarchar(100),
    [Cena] decimal(18.5),
    [MaxKolicina] decimal(18.5),
    [MinKolicina] decimal(18.5),
    CONSTRAINT [PK_Roba] PRIMARY KEY ([IDRoba])
)
```

```
CREATE TABLE [Roba](
    [IDPopis] int NOT NULL,
    [DatumPopisa] datetime,
    [PopisnaListaNaziv] nvarchar(100),
    CONSTRAINT [PK_Popis] PRIMARY KEY ([IDPopis])
)
```

```
/*      TabelaPorudžbina      */
/*****/
CREATE TABLE [Porudzbina](
    [IDPopis] int NOT NULL,
    [DatumPorudzbine] datetime,
    [Napomena] nvarchar(100),
    CONSTRAINT [PK_Porudzbina] PRIMARY KEY ([IDPorudzbina])
)

/*****/
/*      TabelaRačun      */
/*****/
CREATE TABLE [Racun](
    [IDracun] int NOT NULL,
    [DatumRacuna] datetime,
    [Vreme] datetime,
    [Iznos] decimal(18.5),
    [PDV] decimal(18.5),
    [Osnovica] decimal(18.5),
    CONSTRAINT [PK_Racun] PRIMARY KEY ([IDRacun])
)

ALTER TABLE [Kasir] ADD CONSTRAINT [Radnik_Kasir]
    FOREIGN KEY [S1IDRadnik] REFERENCES [Radnik][IDRadnik]

ALTER TABLE [Magacioner] ADD CONSTRAINT [Radnik_Magacioner]
    FOREIGN KEY [S1IDRadnik] REFERENCES [Radnik][IDRadnik]

ALTER TABLE [Narucilac] ADD CONSTRAINT [Radnik_Narucilac]
    FOREIGN KEY [S1IDRadnik] REFERENCES [Radnik][IDRadnik]

ALTER TABLE [Racun] ADD CONSTRAINT [Kasir_Racun]
    FOREIGN KEY [S1IDKasir] REFERENCES [Kasir][IDKasir]

ALTER TABLE [Racun] ADD CONSTRAINT [Kasa_Racun]
    FOREIGN KEY [S2IDKasa] REFERENCES [Kasa][IDKasa]

ALTER TABLE [Racun] ADD CONSTRAINT [Roba_Racun]
    FOREIGN KEY [S3IDRoba] REFERENCES [IDRoba][IDRoba]

ALTER TABLE [Popis] ADD CONSTRAINT [Magacioner_Popis]
    FOREIGN KEY [S1IDMagacioner] REFERENCES [Magacioner][IDMagacioner]

ALTER TABLE [Popis] ADD CONSTRAINT [Roba_Popis]
    FOREIGN KEY [S2IDRoba] REFERENCES [Roba][IDRoba]

ALTER TABLE [Porudzbina] ADD CONSTRAINT [Narucilac_Porudzbina]
    FOREIGN KEY [S1IDNarucilac] REFERENCES [Narucilac][IDNarucilac]
```

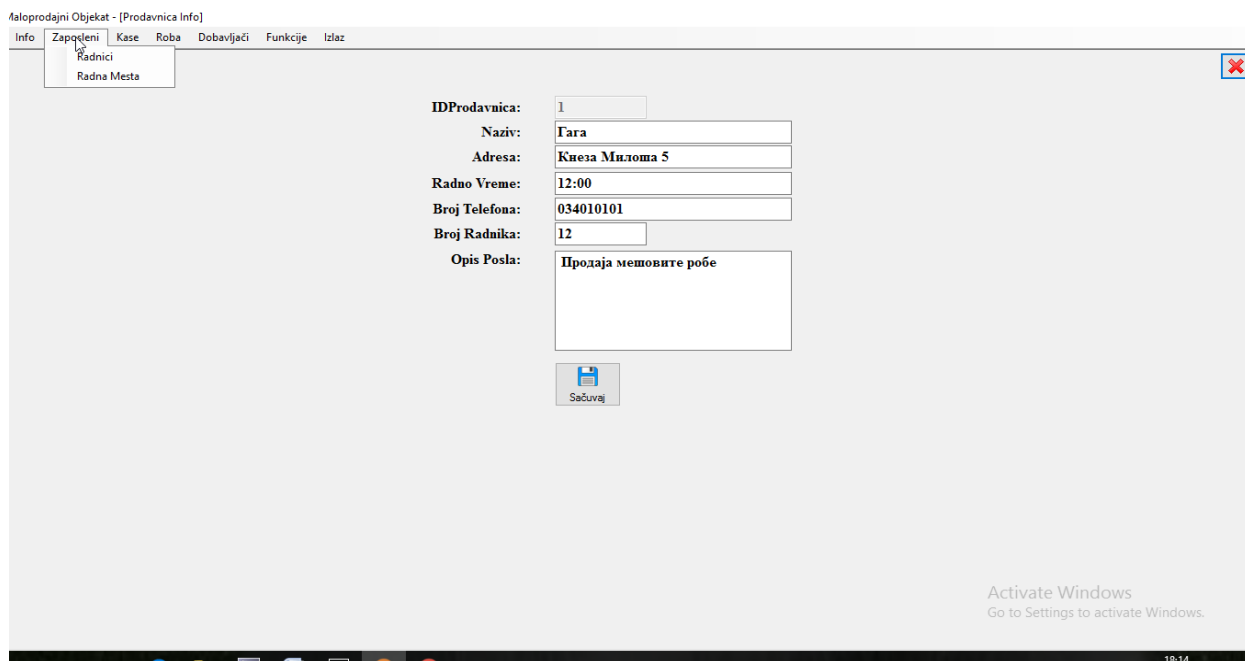
```
ALTER TABLE [Porudzbina] ADD CONSTRAINT [Dobavljac_Porudzbina]  
FOREIGN KEY [S2IDDobavljac] REFERENCES [Dobavljac][IDDobavljac]
```

```
ALTER TABLE [Porudzbina] ADD CONSTRAINT [Roba_Porudzbina]  
FOREIGN KEY [S3IDRoba] REFERENCES [Roba][IDRoba]
```

7.2 Израда апликације

Ова фаза се изводи на основу предходно утврђене шеме базе података, као и конкретних захтева будућих корисника.

Начин стартовања апликације, појављује се прозор као на слици 37.



Слика 37: Почетни прозор апликације

Оно што прво треба урадити по покретању апликације јесте унос података запослених радника (атрибути ентитета РАДНИК). Кликом "Додај" отвара се нови прозор, у оквиру кога се уносе подаци о новом раднику (слика 38).

Maloprodajni Objekt - [FrmRadnik]

Info Zaposleni Kase Roba Dobavljači Funkcije Izlaz

Ime	Prezime	Adresa	Str. Sprema	Datum.Zaposlenja	Broj Telef.
Марко	Марковић	Јанка Веселиновић	Средња трговац...	28.08.2017.	062111111
Јанко	Јанковић	Хајдук Вељка 2	Средња трговац...	01.09.2017.	061234565
Марија	Маринковић	Илинданска 1	Средња трговац...	01.08.2017.	061123456
Ана	Бојић	Светогорска 3	Средња трговац...	01.07.2017.	064123456
Драгана	Јовић	Хаџи Милентије...	Средња трговац...	01.06.2017.	063123456
Бојана	Анђић	Петра Петровић...	Средња трговац...	01.04.2017.	06123456
Јовица	Јелић	Анђићева 5	Средња трговац...	12.07.2017.	06423456
Јована	Петровић	Знај Јовина 9	Средња трговац...	12.08.2017.	034123456
Мико	Рајић	Његовева 19	Средња трговац...	11.09.2017.	066 414 623
Марија	Анчић	Николе Пашића 1	Средња трговац...	10.05.2017.	060456123

IDRadnik: 1
Ime: Марко
Prezime: Марковић
Adresa: Јанка Веселиновић 2
Stručna Sprema: Средња трговачка школа
Datum Zapošlenja: 28.08.2017.
Br. Telefona: 062111111
Radno Mesto: Magacioner

Activate Windows
Go to Settings to activate Windows.

Слика 38: Унос података у табелу РАДНИК

У претходном кораку смо дефинисали основне податке радника, док у следећем кораку се дефинишу њихове квалификације, надлежности и опис посла и на основу тога врши се подела послова у малопродајном објекту (слика 39).

Maloprodajni Objekt - [FrmRadnaMesta]

Info Zaposleni Kase Roba Dobavljači Funkcije Izlaz

RADNIK

Ime	Prezime	Adresa	Str. Sprema	Datum.Zaposlenja	Broj Telef.
-----	---------	--------	-------------	------------------	-------------

KASIR

Radnik	Nadležnost
Драгана Јовић	рад на каси
Јовица Јелић	рад на каси
Јована Петровић	рад на каси
Марија Анчић	рад на каси

MAGACIONER

Radnik	Kvalifikacija
Марко Марковић	kval 1
Јанко Јанковић	kval 2
Марија Маринковић	kval 3
Ана Бојић	Касир
Бојана Анђић	магационер

NARUČILAC

Radnik	Opis Posla
--------	------------

Слика 39: Унос података у табеле КАСИР, НАРУЧИЛАЦ И МАГАЦИОНЕР

Након уноса података у табелу запослени, врши се унос података у табелу КАСА (атрибути ентитета КАСА). Кликом на дугме „ДОДАЈ” подаци се додају у базу података док кликом на дугме "САЧУВАЈ" подаци се трајно чувају (слика 40).

Maloprodajni Objekat - [Kasa]

Info Zaposleni Kase Roba Dobavljači Funkcije Izlaz

IDKasa	DatumProizvodnje	Proizvodjac	KapMemorije	Jezik
1	05.06.2017.	Прима 1	200	Српски
2	11.09.2017.	Прима 2	850	Српски
3	11.09.2017.	Прима 3	300	Српски
4	11.09.2017.	Прима 4	400	Српски

Dodaj Obrisi

IDKasa: 1

Datum Proizvodnje: 05.06.2017.

Proizvođač: Прима 1

Kapacitet memorije: 200

Jezik: Српски

Sačuvaj

Activate Windows
Go to Settings to activate Windows.

Слика 40: Унос података у табелу КАСА

Након уноса података у табелу КАСА, следи унос података у табелу РОБА и ДОБАВЉАЧ (слика 41,42).

Maloprodajni Objekat - [Roba]

Info Zaposleni Kase Roba Dobavljači Funkcije Izlaz

IDRoba	Naziv	Cena	MaxKolicina	MinKolicina
1	Свит	40	100	10
2	Крем банана	20	100	10
3	Смоки	30	50	5
4	Чипс	58	30	5
5	Jelen Pivo	65	100	20
6	Фанта	130	50	5
7	Koka Kola 0.33	58	40	10
8	Sprite	119	5698	5
9	Јафа кекс	100	30	5

Dodaj Obrisi

IDRoba: 1

Naziv: Свит

Cena: 40

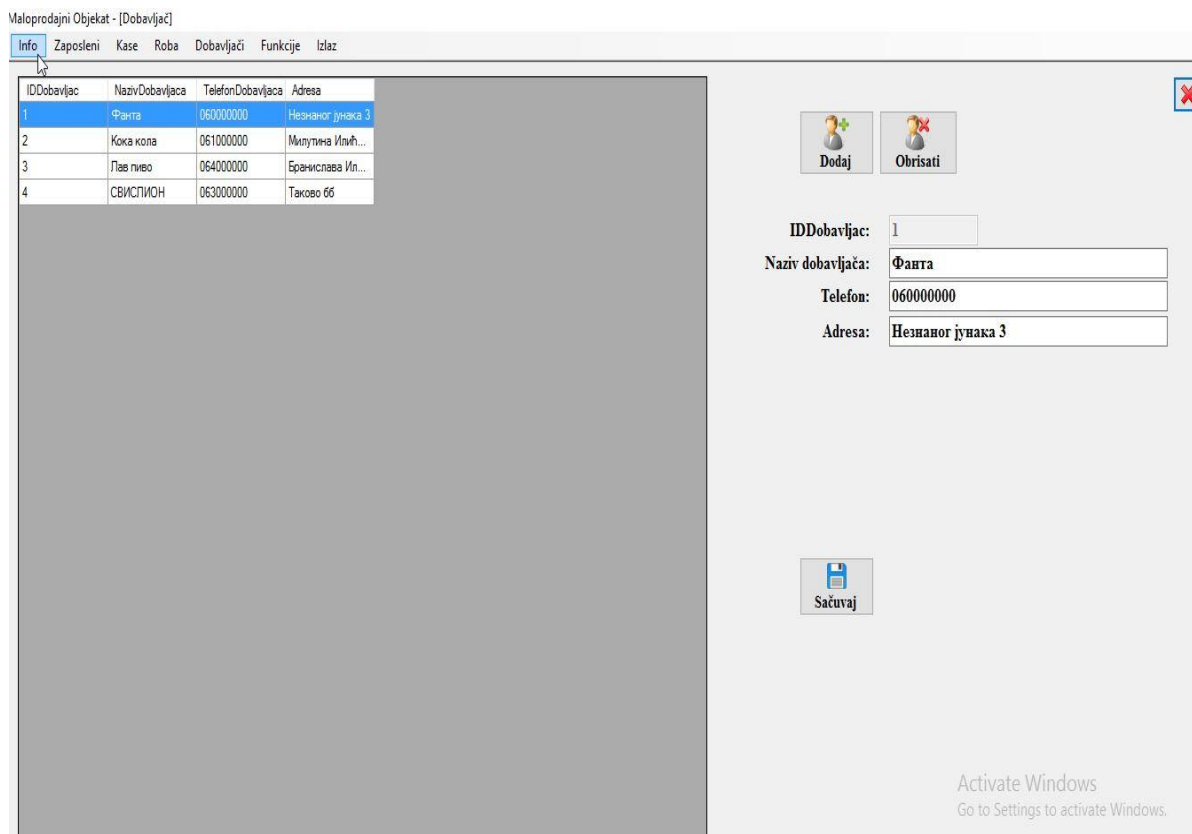
MaxKolicina: 100

MinKolicina: 10

Sačuvaj

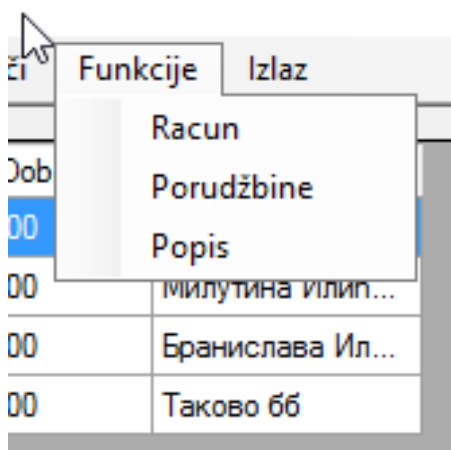
Activate Windows
Go to Settings to activate Windows.

Слика 41: Унос података у табелу РОБА



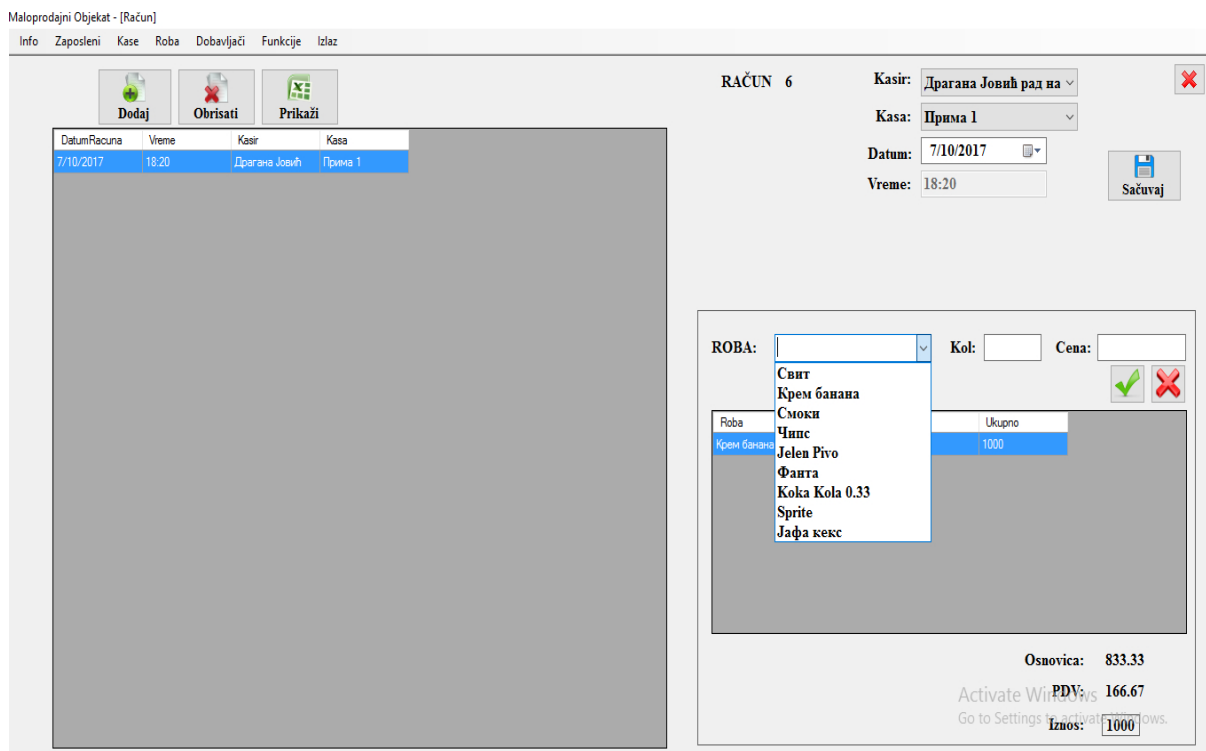
Слика 42: Унос података у табелу ДОБАВЉАЧ

Након уноса основних података у табеле ЗАПОСЛЕНИ, КАСА, ДОБАВЉАЧ И РОБА, следи избор функција (слика 43).



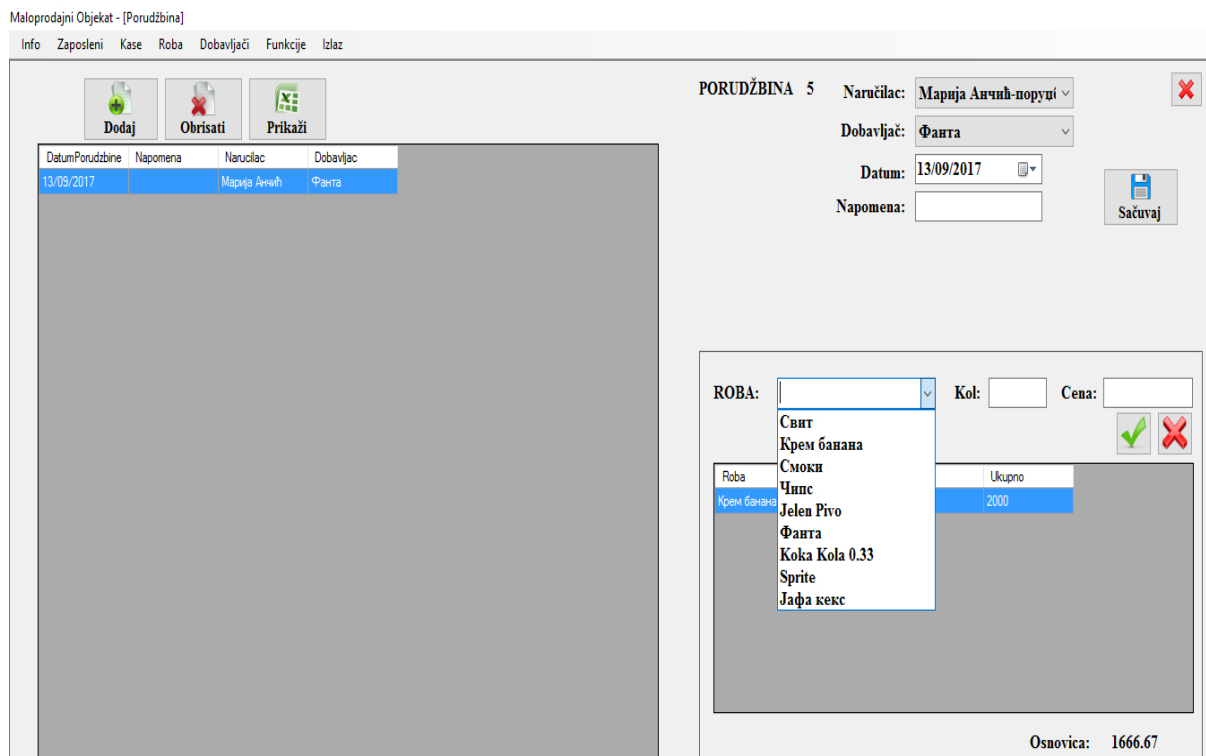
Слика 43: Функције апликације

Након стартовања функције РАЧУН, врши се избор касира, касе и уноси датум издавања рачуна. Након што се унети подаци сачувају у бази, врши се унос назива робе као и тражена количина робе, где аутоматски добијамо цену робе, основицу, пев и укупан износ тј. рачун (слика 44). Рачун се може штампати, брисати и додавати.



Слика 44: Функција РАЧУН

Након стартовања функције ПОРУЏБИНА, врши се избор наручиоца, добављача и уноси датум наручивања. Након што се унети подаци сачувају у бази, врши се унос назива робе као и тражена количина робе, где аутоматски добијамо цену робе, основицу, пдв и укупан износ наручене робе. Наружбеница се може штампати, брисати и пружа могућност додавања изостављених артикала (слика 45).



Слика 45: Функција ПОРУЏБИНА

Након стартовања функције ПОПИС, врши се избор магационера, уноси датум пописа и попуњава пописна листа. Након што се унети подаци сачувају у бази, врши се унос података преостале робе који се односе на назив робе и количину (слика 46). Такође, садржи исте функције.

Maloprodajni Objekat - [Popis]

Info Zaposleni Kase Roba Dobavljači Funkcije Izlaz

Dodaj **Obrisati** **Prikaži**

POPIS 5 Magacioner: **Mirko Pajuh-Kvalifikovan**

Datum: **11/09/2017**

Popisna Lista: **попис 3**

Sačuvaj

DatumPopisa	Radnik	Lista Naziv	PopisnaListaNaziv
11/09/2017	Mirko Pajuh-Kvali...	попис 3	

ROBA: Kol:

Roba	Kolicina
Koka Kola 0.33	58
Sprite	120
Чинц	600

Слика 46: Функција ПОПИС

8. ЗАКЉУЧАК

Како се у трговинским објектима писмено документују активности по питању потрошње, набавке, издавања робе и различитих активности везане за ову делатност, дошли смо на идеју да би се креирањем информационог система трајно документовали ови подаци, чиме би постали доступни сваком запосленом раднику у сваком тренутку, а могле би се вршити и разне анализе и контроле од надређеног кадра.

Након идентификације кандидата за ентитете, дефинисања веза, кандидата за кључеве и атрибуте, следи креирање ЕР дијаграма. Превођењем ЕР дијаграма у релациони модел долазимо до апликативног моделирања односно креирања базе података (која у суштини представља каталог ентитета Продавница, Каса, Радник, Касир, Наручилац, Додављач, Магационер, Попис, Поруџбина, Роба и Рачун) у Microsoft Visual Studio2010. На крају је развијена апликација, у програмском језикуC#, који омогућава унос и обраду података у креираној бази.

У раду је описан и начин функционисања апликације. Апликација је веома једноставна и сви будући корисници могу брзо да се упознају са основним принципима и лако је могу користити.

9. ЛИТЕРАТУРА

1. Давила С.: **Дизајн базе података**, Алгебра, 2010.
2. Ерић М.: **Пројектовање информационих система и базе података**- скрипта, Факултет Инжењерских наука, Крагујевац 2016.
3. Лазаревић Б.: **Базе података**, ФОН Београд, Београд 2003.
4. Лукић Н.: **Базе података**, Бијељина, 2007.
5. Полишчук Ј.: **Пројектовање информационих система**, Електротехнички факултет, Подгорица 2007.
6. Вељовић А.: **Пројектовање информационих система**, Компјутер библиотека, 2003.
7. Вијеновић М., Шимић Г.: **Увод у базе података**, Универзитет Сингидунум, Београд 2010.
8. www.algebra.hr/wp-content/uploads/2015/12/Administrator-baze-podataka-preview.pdf (15.07.2017.)