|  |
| --- |
| Универзитет у крагујевцу  Факултет техничких наука |
| База података за праћење рада Рода мегамаркета |
| Пројектни рад из предмета Програмирање база података |
|  |
|  |
|  |

|  |
| --- |
| Чачак, 2019. године |

Ментор: Студент:

Катарина Митровић, асистент *Зорана Гавриловић 93-2016*

Садржај

[1. Увод 4](#_Toc18928059)

[2. Анализа процеса и информационих токова 5](#_Toc18928060)

[3. Анализа релевантних докумената 7](#_Toc18928061)

[Отпремница добављача 7](#_Toc18928062)

[Евиденција пријема 8](#_Toc18928063)

[Ценовник 9](#_Toc18928064)

[Рачун за физичка лица 10](#_Toc18928065)

[4. Модел објекти-везе 11](#_Toc18928066)

[Модел објекти-везе пре извршене нормализације 11](#_Toc18928067)

[Модел објекти-везе након извршене нормализације 12](#_Toc18928068)

[5. Денормализација 13](#_Toc18928069)

[5.1. Pre-joining денормализација која нарушава 2НФ 13](#_Toc18928070)

[5.1.1. Тригер 1 15](#_Toc18928071)

[5.1.2. Тригер 2 16](#_Toc18928072)

[5.1.3 Тригер 3 16](#_Toc18928073)

[5.2. Pre-joining денормализација која нарушава 3НФ 17](#_Toc18928074)

[5.2.1 Тригер 1 18](#_Toc18928075)

[5.2.2 Тригер 2 19](#_Toc18928076)

[5.2.3 Тригер 3 20](#_Toc18928077)

[6. Оптимизација 21](#_Toc18928078)

[6.1. Оптимизација заснована на изведеним вредностима 21](#_Toc18928079)

[6.1.1. Тригер 1 23](#_Toc18928080)

[6.1.2. Тригер 2 23](#_Toc18928081)

[6.1.3 Тригер 3 24](#_Toc18928082)

[6.1.4 Тригер 4 24](#_Toc18928083)

[6.1.5 Тригер 5 25](#_Toc18928084)

[6.2. Индекси 26](#_Toc18928085)

[6.2.1. Индекс 1 26](#_Toc18928086)

[6.2.2. Индекс 2 26](#_Toc18928087)

[6.2.3 Индекс 3 27](#_Toc18928088)

[7. Релациона шема 28](#_Toc18928089)

[7.1 Табела КУПЦИ 29](#_Toc18928090)

[7.2 Табела РАЧУНИ 29](#_Toc18928091)

[7.3 Табела АРТИКЛИ 30](#_Toc18928092)

[7.4 Табела СТАВКЕ\_РАЧУНА 31](#_Toc18928093)

[7.5 Табела ОБЈЕКТИ 31](#_Toc18928094)

[7.6 Табела ДОБАВЉАЧИ 32](#_Toc18928095)

[7.7 Табела ОТПРЕМНИЦЕ 32](#_Toc18928096)

[7.8 Табела ЕВИДЕНЦИЈЕ\_ПРИЈЕМА 33](#_Toc18928097)

[8. UML дијаграм случајева коришћења 34](#_Toc18928098)

[9. Услови за имплементацију 35](#_Toc18928099)

[10. Подаци за тестирање 37](#_Toc18928100)

[10.1 Табела КУПЦИ 37](#_Toc18928101)

[10.2 Табела РАЧУНИ 38](#_Toc18928102)

[10.3 Табела АРТИКЛИ 39](#_Toc18928103)

[10.4 Табела СТАВКЕ\_РАЧУНА 41](#_Toc18928104)

[10.5 Табела ОБЈЕКТИ 42](#_Toc18928105)

[10.6 Табела ДОБАВЉАЧИ 43](#_Toc18928106)

[10.7 Табела ОТПРЕМНИЦЕ 44](#_Toc18928107)

[10.8 Табела ЕВИДЕНЦИЈЕ\_ПРИЈЕМА 45](#_Toc18928108)

[11. Опис апликације са упутством за коришћење 48](#_Toc18928109)

[11.1 Почетна страна 48](#_Toc18928110)

[11.2 Приказ 49](#_Toc18928111)

[11.3 Унос, измена и брисање 50](#_Toc18928112)

[11.4 Израда документа 52](#_Toc18928113)

[12. Закључак 53](#_Toc18928114)

[Индекс слика 54](#_Toc18928115)

[Литература 56](#_Toc18928116)

# Увод

Циљ израде овог семинарског рада јесте изучавање пословног система и конверзија, представљање истог као информациони. Сама имплементација, превођење једног система у други састоји се од једне базе података која ће то на најпростији начин да представи и опише.

Рода, Меркатор, постоји од 1994. године и малопродајни је ланац започет у Кули. Продајни систем се састоји из 35 објеката широм Србије. Малопродајно место које се представља у овом раду је устаљено у Јагодини и то је друштво са ограниченом одговорношћу. Пословни циљ једне овакве компаније је експанзија, у свим могућим аспектима. Најважнији циљ јесте пружање јединствене услуге и задржавање муштерија, како сталних тако и придобијање нових. Рода као малопродајни објекат има доста уговора са осталим фирмама које им достављају одређене продајне експонате, а поред тога има и ланац својих производа. Продаја се врши физичким и правним лицима, издаје се рачун за обе врсте муштерија (разликују се).

Пријем робе и вредност продајних ствари пролазе кроз надлежност запосленог лица које има титулу калкуланта. То је особа која одговара за сигурност и безбедност сваког артикла који је ушао у фирму тиме што ће радити пописе на месечном и годишњем нивоу. На тим пописима не сме бити мањка, у већини случајева га има, али га профити покривају. Књига у коју се на дневном нивоу уносе подаци о издатим рачунима се зове књига евиденције и њој има приступ само овлашћено лице. У Јагодини тренутно постоји само једно запослено лице које извршава све те функције (калкулације). Овим кратким уводом је представљен пословни систем у најкраћим цртама.



# Анализа процеса и информационих токова

Један од процеса у описаном пословном систему је сам пут артикала од поручивања до продаје. Потпроцеси обухватају:

• Наручивање робе

• Дотавку робе

• Продају робе

Роба се наручује од разних добављача специфицирањем броја потребних артикала одређеног производа. Подаци о њима се могу видети на документима као што су отпремница добављача и евиденција пријема робе.

Поруџбина робе

Поручивање мејлом/лично

Отпремница добављача

Доставка робе се одвија договореног датума и добављач из одређене фирме при доставци доноси документ са списком испоручене робе.

Доставка робе

Физичко достављаље

Евиденција пријема

Након евиденције о пријему робе, она се излаже на продајном месту и постаје доступна муштеријама-купцима.

Продаја робе

Примљена роба

Рачун

Испод је табела која приказује који документи су улази и излази из ког потпроцеса и на који начин тако повезани чине једну целину.

|  |  |  |
| --- | --- | --- |
| Улаз | Подпроцес | Излаз |
| Наруџбеница | Поруџбина робе | Отпремница добављача |
| Отпремница добављача | Доставка робе | Евиденција пријема  Ценовник |
| Евиденција пријема  Ценовник | Продаја робе | Рачун за физичка лица  Рачун за правна лица |

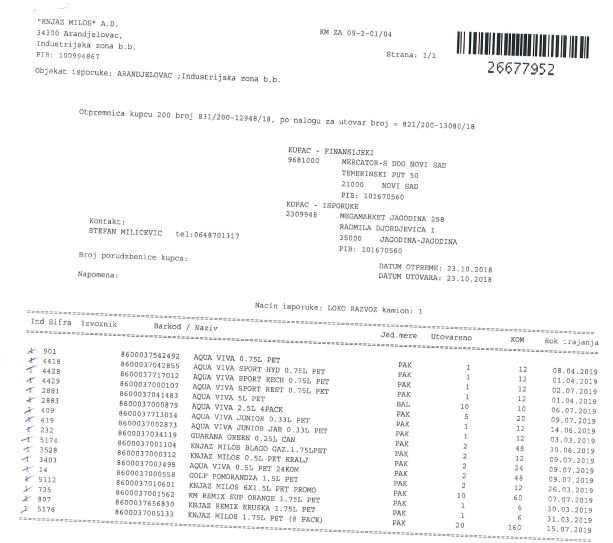
Табела 1 – Табела улаза и излаза за процес *од поручивања до продаје робе*

# Анализа релевантних докумената

У овом поглављу биће описана документа која дају информације о томе како наш реалан пословни систем функционише. Документа обухватају: отпремницу добављача, евиденцију пријема, ценовник и рачун за физичка лица.

## Отпремница добављача

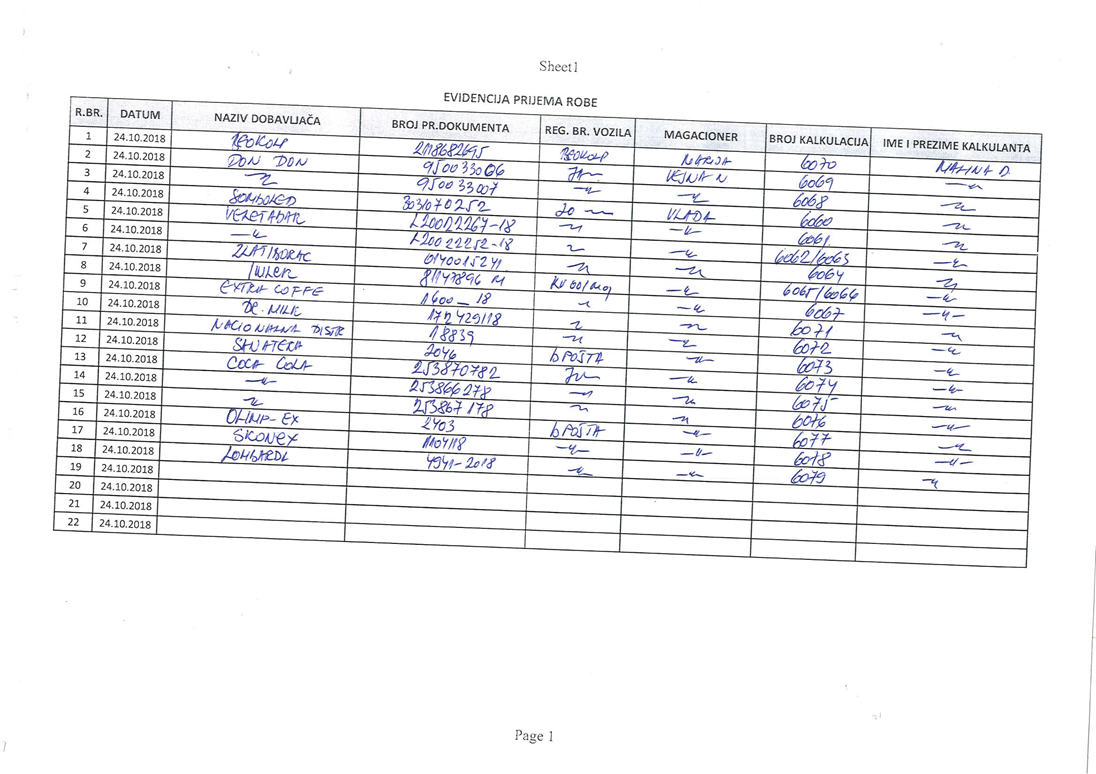
Ова отпремница даје информације о фирми која је послала артикле (назив, адреса, ПИБ), податке о фирми која наручује(назив, адреса, ПИБ), датум отпреме и утовара и податке о артиклима као што су шифра, баркод и назив, јединица мере, количина утоварене робе, количина робе у комадима и рок трајања. На слици испод приказана је отпремница добављача „Књаз Милош“ А.Д.



Слика 1 - отпремница добављача

## Евиденција пријема

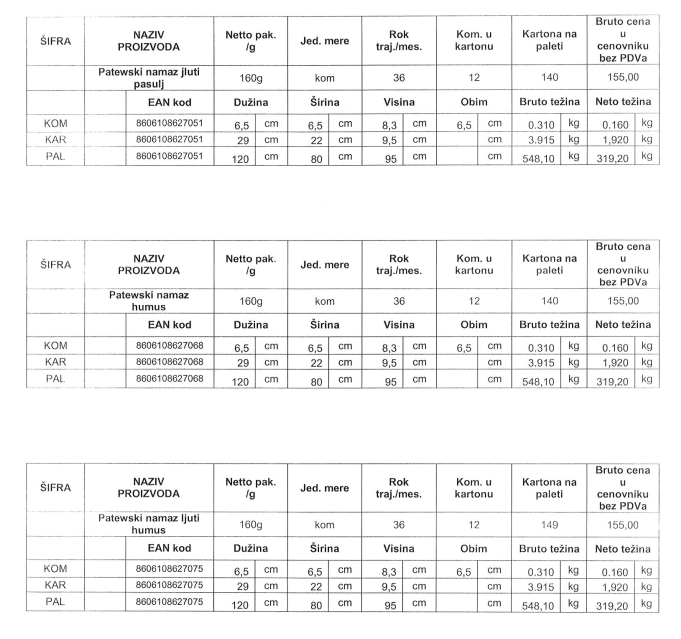
Овај документ је ручно попуњен. Приказује податке о томе када је који добављач доставио робу Рода мегамаркету. На њему се може видети редни број пријема робе, датум, назив добављача, број пријемног документа, регистарски број возила, магационер, број калкулација и име и презиме калкуланта.



Слика 2 - евиденција пријема

## Ценовник

На ценовнику можемо видети основне податке о производима као што су шифра, назив производа, јединица мере, цена, и друга поља која се односе на порезе. На наредној слици налази се део ценовника фирме Patewski која доставља намазе Рода мегамаркету.



Слика 3 – ценовник

## Рачун за физичка лица

Рачун се издаје на месту куповине и садржи информације о продајном месту и роби коју је муштерија купила (назив, количина и цена), као и о датуму и времену куповине.



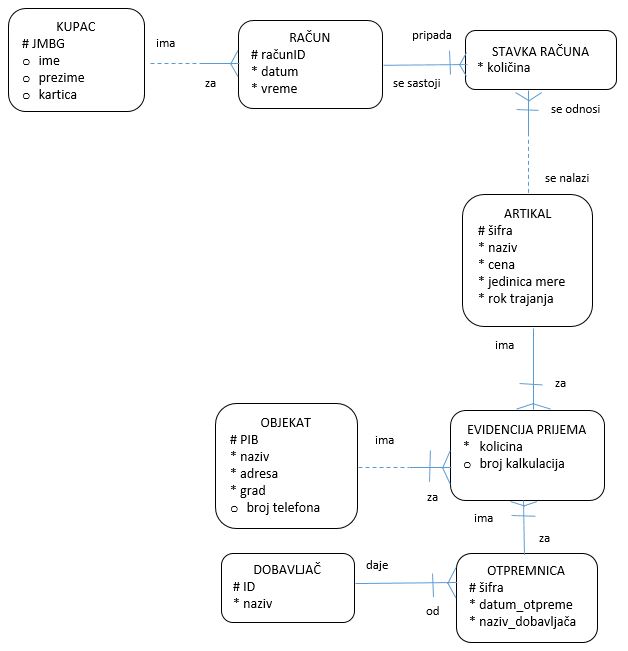
Слика 4 - рачун за физичка лица

# Модел објекти-везе

Код моделовања реалних система помоћу база података битно је тежити што нормализованијем моделу, али до граница докле је претраживање базе временски ефикасно – не треба нормализовати базу до крајности јер се на тај начин у упите уводи превише JOIN наредби које су лоше за брзину претраживања. У овом поглављу биће приказан пример денормализоване базе и један од начина на који се она може нормализовати.

## Модел објекти-везе пре извршене нормализације

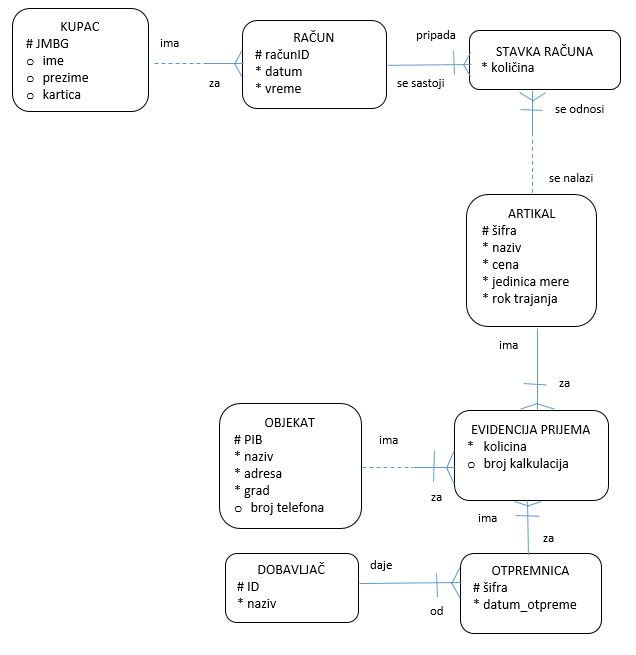
На наредном ЕР дијаграму приказан је денормализован модел базе података. Нормализација коју је потребно извршити односи се на атрибут назив\_добављача у ентитету ОТПРЕМНИЦА. Постојање овог атрибута у ентитету ОТПРЕМНИЦА нарушава другу нормалну форму јер назив\_добављача не зависи од целог сложеног примарног кључа у ентитету (шифра отпремнице + шифра добављача), већ само од шифре добављача. Како би се модел нормализовао и задовољила друга нормална форма, потребно је избрисати атрибут назив\_добављача из ентитета ОТПРЕМНИЦА и оставити га у ентитету ДОБАВЉАЧ.



Слика 5 - модел објекти-везе пре нормализације

## Модел објекти-везе након извршене нормализације

На наредном ЕР дијаграму је модел након нормализације. Ентитет ДОБАВЉАЧ има један примарни кључ и назив добављача зависи од њега. Ако је примарни кључ прост, аутоматски је задовољена друга нормална форма, што је овде случај.



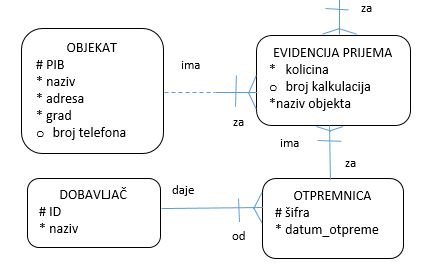
Слика 6 - модел објекти-везе након нормализације

# Денормализација

На потпуно нормализованом моделу биће извршене две денормализације, због природе система и захтева који подразумевају да се одређени скупови података често заједно претражују. Прва денормализација је pre-joining која нарушава другу нормалну форму, а друга је pre-joining која нарушава трећу нормалну форму.

## Pre-joining денормализација која нарушава 2НФ

Прва денормализација обухвата ентитете ОБЈЕКАТ и ЕВИДЕНЦИЈА ПРИЈЕМА. Када калкулант шаље евиденције пријема из једног објекта у друге, битно му је да на том документу буде тачан назив објекта на који се евиденција односи. Зато је атрибут назив објекта пренет у ентитет ЕВИДЕНЦИЈА ПРИЈЕМА, нарушавајући другу нормалну форму јер у том ентитету назив објекта не зависи од целог примарног кључа којег чине шифра добављача, шифра отпремнице, ПИБ објекта и шифра артикла.



Слика 7 - денормализација 1

Табела спецификације тригера документује шта који тригер ради над којим пољима у ентитетима. У њој је детаљно описана логика коју тригер извршава:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Табела | Тип тригера | Колона | Потребан | Шта треба да се уради? | Редни број тригера |
| Објекти | INSERT |  | НЕ |  |  |
| UPDATE | назив\_објекта | ДА | Потребно је да тригер омогући аутоматско ажурирање вредности за назив\_објекта у Евиденцијама пријема, када се он промени у Објектима | 1 |
| DELETE |  | НЕ |  |  |
| Евиденције пријема | INSERT |  | ДА | Вредност за назив\_објекта се аутоматски поставља у Евиденцијама пријема на основу унетог ПИБ-а | 2 |
| UPDATE | назив\_објекта | ДА | Забрањена је директна измена вредности овог поља у табели Евиденције пријема | 3 |
| ПИБ | ДА | Ажурирање вредности за назив\_објекта на основу унетог ПИБ-а | 2 |
| DELETE |  | НЕ |  |  |

Табела 2 – Табела спецификације тригера за денормализацију 1

### Тригер 1

ALTER TABLE evidencije\_prijema

ADD naziv\_objekta VARCHAR2(20);

UPDATE evidencije\_prijema

SET naziv\_objekta = (

SELECT naziv

FROM objekti

WHERE evidencije\_prijema.pib = objekti.pib

);

CREATE OR REPLACE TRIGGER objekti\_update\_nazivobjekta

AFTER

UPDATE OF naziv ON objekti

FOR EACH ROW

DECLARE

novi\_naziv VARCHAR2(20);

o\_pib NUMBER(10,0);

BEGIN

novi\_naziv := :NEW.naziv;

o\_pib := :NEW.pib;

UPDATE evidencije\_prijema

SET naziv\_objekta = novi\_naziv

WHERE pib = o\_pib;

END;

### Тригер 2

CREATE OR REPLACE TRIGGER evidencije\_prijema\_insert

BEFORE

INSERT ON evidencije\_prijema

FOR EACH ROW

DECLARE

novi\_naziv\_objekta VARCHAR2(20);

BEGIN

SELECT naziv

INTO novi\_naziv\_objekta

FROM objekti

WHERE pib = :NEW.pib;

:NEW.naziv\_objekta := novi\_naziv\_objekta;

END;

### 5.1.3 Тригер 3

CREATE OR REPLACE TRIGGER evidencije\_prijema\_update\_no

BEFORE

UPDATE OF naziv\_objekta ON evidencije\_prijema

FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(

num=>-20002,

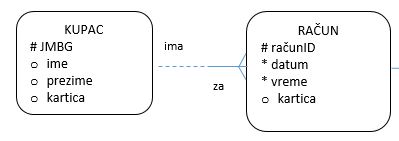
msg=> ‘Naziv objekta se ne moze direktno menjati.’

);

END;

## Pre-joining денормализација која нарушава 3НФ

У овом случају, због тога што су табеле повезане само преко спољног кључа, pre-joining денормализацијом нарушена је трећа нормална форма. Појавила се потреба да на рачунима физичким или правним лицима стоји и број кредитне картице којом су плаћени артикли, ако су плаћени картицом а не кешом. Због тога је ефикасније за базу података да у ентитету РАЧУН има редундантно поље картица.



Слика 8 - денормализација 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Табела | Тип тригера | Колона | Потребан | Шта треба да се уради? | Редни број тригера |
| Купци | INSERT |  | НЕ |  |  |
| UPDATE | картица | ДА | Када се вредност поља картица промени у табели Купци, тригер ажурира вредности за картицу у Рачунима | 1 |
| DELETE |  | НЕ |  |  |
| Рачуни | INSERT |  | ДА | Вредност за картицу се аутоматски поставља на основу унетог рачунИД-а | 2 |
| UPDATE | картица | ДА | Забрањена је директна измена вредности овог поља | 3 |
| рачунИД | ДА | Ажурирање вредности за картицу на основу унетог рачунИД-а | 2 |
| DELETE |  | НЕ |  |  |

Табела 3 - Табела спецификације тригера за денормализацију 2

### 5.2.1 Тригер 1

ALTER TABLE racuni

ADD kartica NUMBER(16,0);

UPDATE racuni

SET kartica = (

SELECT kartica

FROM kupci

WHERE kupci.jmbg = racuni.jmbg

);

CREATE OR REPLACE TRIGGER kupci\_update\_kartica

AFTER

UPDATE OF kartica ON kupci

FOR EACH ROW

DECLARE

nova\_kartica NUMBER(16,0);

k\_jmbg NUMBER(13,0);

BEGIN

nova\_kartica := :NEW.kartica;

k\_jmbg := :NEW.jmbg;

UPDATE racuni

SET kartica = nova\_kartica

WHERE jmbg = k\_jmbg;

END;

### 5.2.2 Тригер 2

CREATE OR REPLACE TRIGGER racuni\_insert

BEFORE

INSERT ON racuni

FOR EACH ROW

DECLARE

nova\_kartica NUMBER(16,0);

BEGIN

SELECT kartica

INTO nova\_kartica

FROM kupci

WHERE jmbg = :NEW.jmbg;

:NEW.kartica := nova\_kartica;

END;

### 5.2.3 Тригер 3

CREATE OR REPLACE TRIGGER racuni\_update\_kartica

BEFORE

UPDATE OF kartica ON racuni

FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(

num=>-20002,

msg=> ‘Kartica se ne moze direktno menjati.’

);

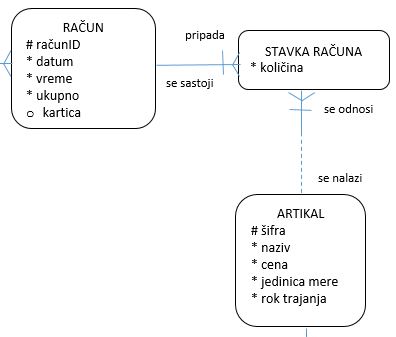
END;

# Оптимизација

Оптимизација која ће бити извршена на овој бази података заснива се на изведеним вредностима. То значи да се уводи нека нова колона чија се вредност одређује или израчунава на основу једне или више друих вредности.

## Оптимизација заснована на изведеним вредностима

Коришћење оптимизације storing derivable values подразумева додавање нове колоне која прерачунава неку вредност сваки пут када се подаци додају/промене/избришу. У овом случају додата је нова колона “укупно” у табели РАЧУН. Њена улога је да садржи укупну вредност свих ставки рачуна, која се израчунава као сума свих количина \* цена за један рачун. Ова оптимизација је извршена јер је на рачуну потребно видети укупну цену за све артикле у одређеним количинама и увођење ове колоне омогућује бржи рад базе јер није потребно у сваком упиту поново израчунавати укупну вредност артикала на рачуну. У случају оптимизције не нарушавају се нормалне форме, али се уводи редундантност података и нарушава њихов интегритет.



Слика 9 - оптимизација Storing derivable values

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Табела | Тип тригера | Колона | Потребан | Шта треба да се уради? | Редни број тригера |
| Рачуни | INSERT |  | ДА | Када се унесе нови рачун, вредност за укупно се поставља на 0 | 1 |
| UPDATE | укупно | ДА | Забрањено је директно мењати вредност за укупно | 2 |
| DELETE |  | НЕ |  |  |
| Ставке рачуна | INSERT |  | ДА | За унос нове ставке рачуна, ажурира се вредност за укупно | 3 |
| UPDATE | количина | ДА | Зависно од промене количине, мења се укупно | 4 |
| шифра\_артикла | ДА | Зависно од промене шифре артикла, тј.његове цене, ажурира се и вредност за укупно |
| рачунИД | ДА | Када се промени рачунИД, ажурира се укупно |
| DELETE |  | ДА | Ако се ставка рачуна обрише, укупно се ажурира | 5 |

Табела 4 - табела спецификације тригера за оптимизацију

### Тригер 1

ALTER TABLE racuni

ADD ukupno NUMBER(10,0);

CREATE OR REPLACE TRIGGER opt\_racuni\_insert

BEFORE INSERT ON racuni

FOR EACH ROW

DECLARE novo\_ukupno NUMBER(10,0);

BEGIN

novo\_ukupno := 0;

:NEW.ukupno := novo\_ukupno;

END;

### Тригер 2

CREATE OR REPLACE TRIGGER opt\_racuni\_update\_ukupno

BEFORE

UPDATE OF ukupno ON racuni

FOR EACH ROW

BEGIN

RAISE\_APPLICATION\_ERROR(

num=>-20002,

msg=> ‘Ukupna vrednost se ne moze direktno menjati.’

);

END;

### 6.1.3 Тригер 3

CREATE OR REPLACE TRIGGER opt\_stavke\_racuna\_insert

BEFORE INSERT ON stavke\_racuna

FOR EACH ROW

DECLARE

nova\_cena NUMBER(10,0);

BEGIN

SELECT cena

INTO nova\_cena

FROM artikli

WHERE sifra\_artikla = :NEW.sifra\_artikla;

UPDATE racuni

SET ukupno = ukupno + :NEW.kolicina \* nova\_cena

WHERE racunid = :NEW.racunid;

END;

### 6.1.4 Тригер 4

CREATE OR REPLACETRIGGER opt\_stavke\_racuna\_update

AFTER UPDATE ON stavke\_racuna

FOR EACH ROW

DECLARE

nova\_cena NUMBER(10,0);

stara\_cena NUMBER(10,0);

BEGIN

SELECT cena

INTO nova\_cena

FROM artikli

WHERE sifra\_artikla = :NEW.sifra\_artikla;

SELECT cena

INTO stara\_cena

FROM artikli

WHERE sifra\_artikla = :OLD.sifra\_artikla;

UPDATE racuni

SET ukupno = ukupno + :NEW.kolicina \* nova\_cena

WHERE racunid = :NEW.racunid;

UPDATE racuni

SET ukupno = ukupno - :OLD.kolicina \* stara\_cena

WHERE racunid = :OLD.racunid;

END;

### 6.1.5 Тригер 5

CREATE OR REPLACE TRIGGER opt\_stavke\_racuna\_delete

AFTER DELETE ON stavke\_racuna

FOR EACH ROW

DECLARE

stara\_cena NUMBER(10,0);

BEGIN

SELECT cena

INTO stara\_cena

FROM artikli

WHERE sifra\_artikla = :OLD.sifra\_artikla;

UPDATE racuni

SET ukupno = ukupno - :OLD.kolicina \* stara\_cena

WHERE racunid = :OLD.racunid;

END;

## Индекси

Још један од начина оптимизације базе података је индексирање. Увођење индекса побољшава рад базе јер у том случају није потребно скенирати целу табелу да би се нашли жељени подаци. Неке од колона које су добри кандидати за индексе су оне које се често користе уз WHERE наредбу или оне које имају широк опсег вредности (нпр.цене артикала) и индекси које ова база имплементира су фокусирани на та два правила.

### Индекс 1

Пошто постоји честа потреба да се упитима враћају вредности претражене преко имена добављача, креира се индекс на колону назив у табели ДОБАВЉАЧИ. Овај индекс је базиран на функцији за конвертовање свих слова у стрингу у мала слова (LOWER фукнција). Индекс најпре израчуна вредност функције и сачува је. Користи се LOWER функција да не би дошло до непоклапања везаних за коришћење малих или великих слова у упиту.

CREATE INDEX i\_naziv\_dobavljaca

ON dobavljaci(LOWER(naziv));

### Индекс 2

Још једна колона која се често тражи у упитима је колона укупно у табели РАЧУНИ. Ова колона се користи у оптимизацији као променљива која чува укупну вредност ставки рачуна на једном рачуну и због тога је често потребно видети њену вредност. Индекс који је имплементиран у овом случају може се касније скенирати Range index методом јер ће често бити потребно претражити податке који имају вредност за укупно у неком одређеном опсегу.

CREATE INDEX i\_ukupno

ON racuni(ukupno);

### 6.2.3 Индекс 3

У овом случају креирана су два инекса који ће се касније највише скенирати спајањем индекса (Index Join) јер се ове две колоне често налазе заједно у упиту када је потребно видети артикле и њихове цене. Index Join не подразумева праву операцију спајања али се спајају id-ови редова добијених из индекса.

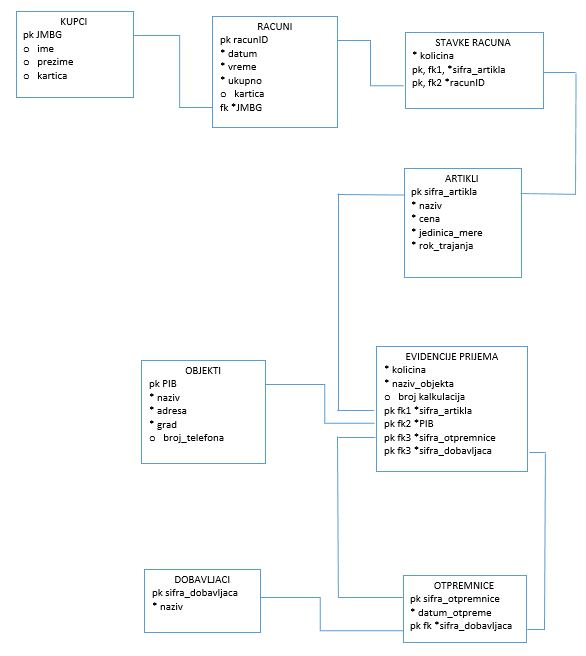
CREATE INDEX i\_naziv\_artikla

ON artikli(naziv);

CREATE INDEX i\_cena\_artikla

ON artikli(cena);

# Релациона шема



Слика 10 - релациона шема базе података

# 7.1 Табела КУПЦИ

CREATE TABLE KUPCI

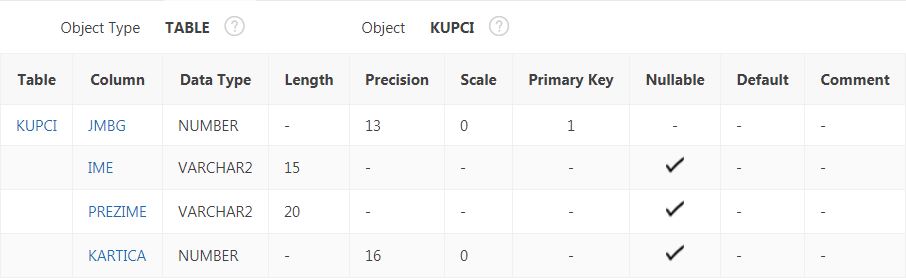
(JMBG NUMBER(13) NOT NULL,

ime VARCHAR2(15),

prezime VARCHAR2(20),

kartica NUMBER(16),

CONSTRAINT pk\_JMBG PRIMARY KEY(JMBG));



Слика 11 - табела КУПЦИ

## 7.2 Табела РАЧУНИ

CREATE TABLE RACUNI

(racunID NUMBER(10) NOT NULL,

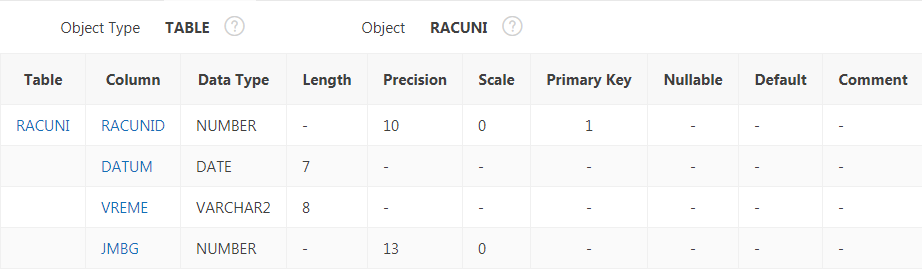
datum DATE NOT NULL,

vreme VARCHAR2(8) NOT NULL,

JMBG NUMBER(13) NOT NULL,

CONSTRAINT fk\_JMBG FOREIGN KEY (JMBG) REFERENCES KUPCI(JMBG),

CONSTRAINT pk\_racunID PRIMARY KEY(racunID));



Слика 12 - табела РАЧУНИ

## 7.3 Табела АРТИКЛИ

CREATE TABLE ARTIKLI

(sifra\_artikla NUMBER(12) NOT NULL,

naziv VARCHAR2(15) NOT NULL,

cena NUMBER(10) NOT NULL,

jedinica\_mere VARCHAR2(8) NOT NULL,

rok\_trajanja DATE NOT NULL,

CONSTRAINT pk\_sifra\_artikla PRIMARY KEY(sifra\_artikla));



Слика 13 - табела АРТИКЛИ

## 7.4 Табела СТАВКЕ\_РАЧУНА

CREATE TABLE STAVKE\_RACUNA

(kolicina NUMBER(6) NOT NULL,

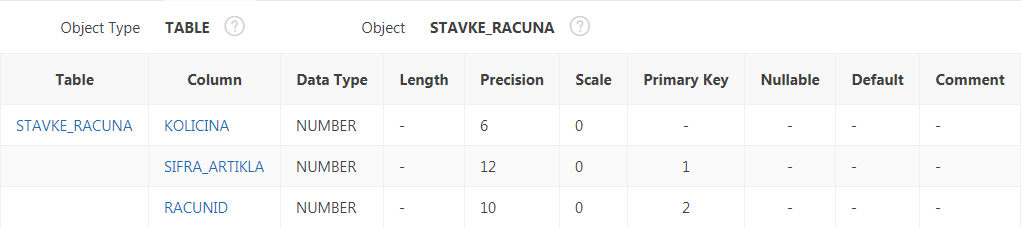
sifra\_artikla NUMBER(12) NOT NULL,

racunID NUMBER(10) NOT NULL,

CONSTRAINT fk1\_sifra\_artikla FOREIGN KEY(sifra\_artikla) REFERENCES ARTIKLI(sifra\_artikla),

CONSTRAINT fk2\_racunID FOREIGN KEY(racunID) REFERENCES RACUNI(racunID),

CONSTRAINT pk\_stavke\_racuna PRIMARY KEY(sifra\_artikla,racunID));



Слика 14 - табела СТАВКЕ\_РАЧУНА

## 7.5 Табела ОБЈЕКТИ

CREATE TABLE OBJEKTI

(PIB NUMBER(10) NOT NULL,

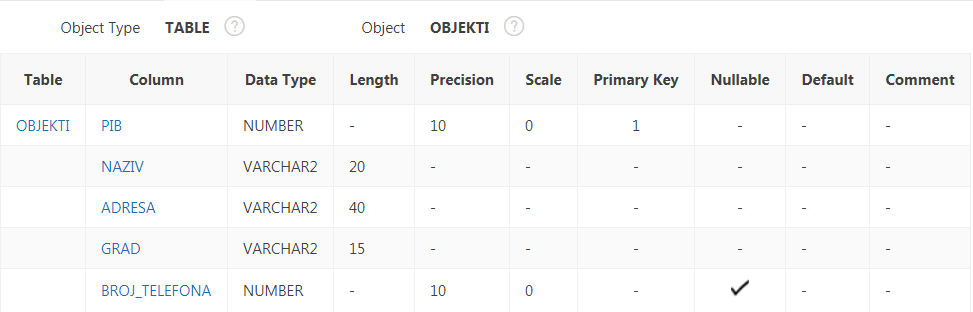
naziv VARCHAR2(20) NOT NULL,

adresa VARCHAR2(40) NOT NULL,

grad VARCHAR2(15) NOT NULL,

broj\_telefona NUMBER(10),

CONSTRAINT pk\_PIB PRIMARY KEY(PIB));



Слика 15 - табела ОБЈЕКТИ

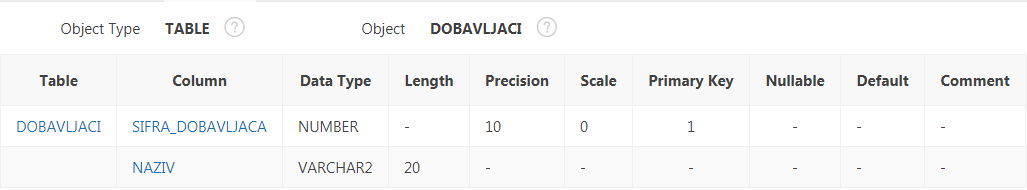
## 7.6 Табела ДОБАВЉАЧИ

CREATE TABLE DOBAVLJACI

(sifra\_dobavljaca NUMBER(10) NOT NULL,

naziv VARCHAR2(20) NOT NULL,

CONSTRAINT pk\_sifra\_dobavljaca PRIMARY KEY(sifra\_dobavljaca));



Слика 16 - табела ДОБАВЉАЧИ

## 7.7 Табела ОТПРЕМНИЦЕ

CREATE TABLE OTPREMNICE

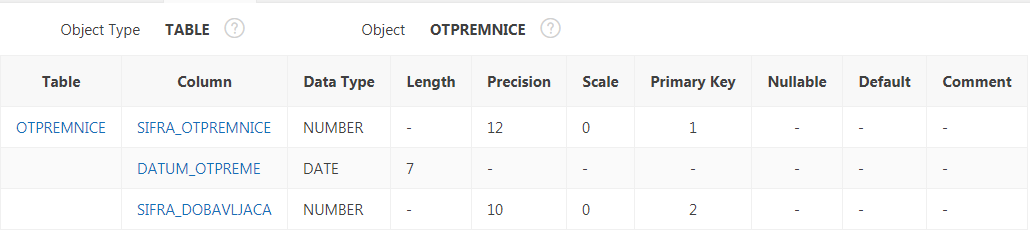
(sifra\_otpremnice NUMBER(12) NOT NULL,

datum\_otpreme DATE NOT NULL,

sifra\_dobavljaca NUMBER(10) NOT NULL,

CONSTRAINT fk\_sifra\_dobavljaca FOREIGN KEY(sifra\_dobavljaca) REFERENCES DOBAVLJACI(sifra\_dobavljaca),

CONSTRAINT pk\_otpremnice PRIMARY KEY(sifra\_otpremnice,sifra\_dobavljaca));



Слика 17 - табела ОТПРЕМНИЦЕ

## 7.8 Табела ЕВИДЕНЦИЈЕ\_ПРИЈЕМА

CREATE TABLE EVIDENCIJE\_PRIJEMA

(kolicina NUMBER(6) NOT NULL,

broj\_kalkulacija NUMBER(4),

sifra\_artikla NUMBER(12) NOT NULL,

PIB NUMBER(10) NOT NULL,

sifra\_otpremnice NUMBER(12) NOT NULL,

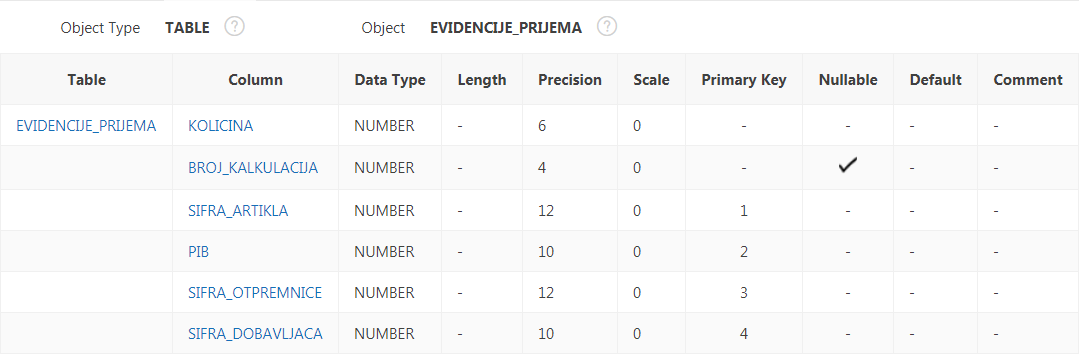
sifra\_dobavljaca NUMBER(10) NOT NULL,

CONSTRAINT fk1\_sif\_artikla FOREIGN KEY(sifra\_artikla) REFERENCES ARTIKLI(sifra\_artikla),

CONSTRAINT fk2\_PIB FOREIGN KEY(PIB) REFERENCES OBJEKTI(PIB),

CONSTRAINT fk3\_otpremnice FOREIGN KEY(sifra\_otpremnice,sifra\_dobavljaca) REFERENCES OTPREMNICE(sifra\_otpremnice,sifra\_dobavljaca),

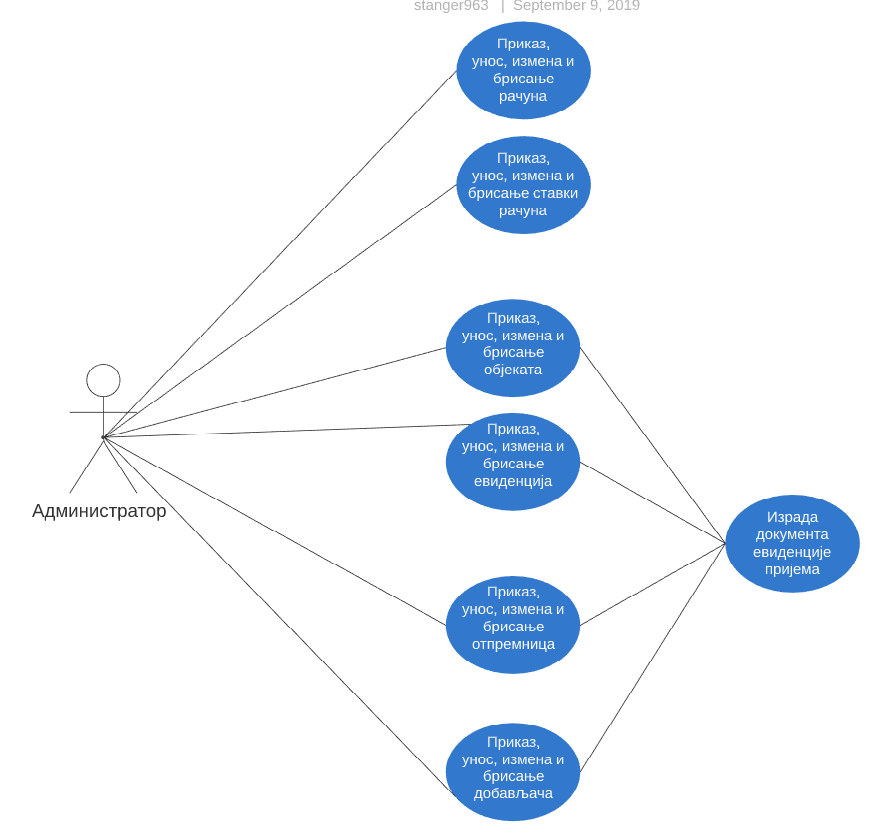
CONSTRAINT pk\_evidencije\_prijema PRIMARY KEY(sifra\_artikla, PIB, sifra\_otpremnice, sifra\_dobavljaca));



Слика 18 - табела ЕВИДЕНЦИЈЕ\_ПРИЈЕМА

# UML дијаграм случајева коришћења

Дијаграм случајева коришћења је приказ интеракције корисника са системом који показује однос између корисника и различитих случајева коришћења у којима је корисник укључен. На дијаграму испод приказан је корисник (администратор) са случајевима коришћења ове апликације.



Слика 19 - дијаграм случајева коришћења

# Услови за имплементацију

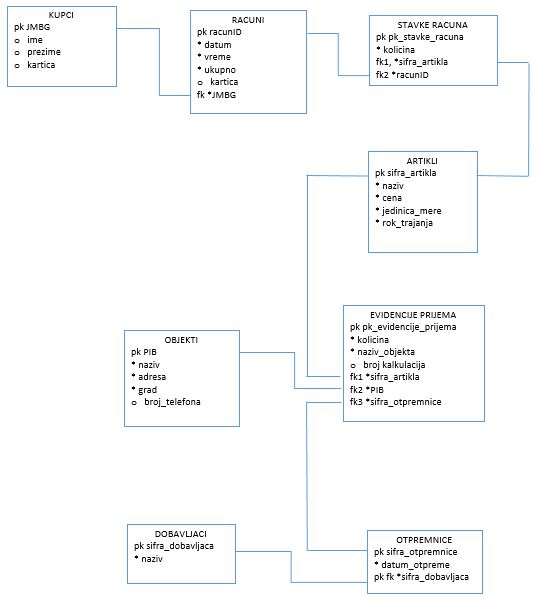
База података креирана је у виду Oracle базе, користећи Sql developer (верзија 17.2) као кориснички интерфејс. Едиција Oracle базе је Oracle Database 18c Express Edition Release 18.0.0.0.0. Апликација је написана у Python програмском језику верзије 3.6.5 са Django framework-ом верзије 2.2.5. За писање апликације коришћен је Atom едитор текста. Django framework користи model-template-view (MTV) архитектуру.

Како би Oracle база могла да се користи уз Python, било је потребно преузети cx\_Oracle модул за проширење, верзије 6.0 или изнад. Верзија коришћена у овом пројекту је 7.2.2. Такође, у settings.py фајлу у стаблу апликације било је потребно подесити податке о бази за коју је апликација везана. Име конекције се мора поклапати са именом конекције у Sql developeru, као и корисничко име, шифра, хост и порт који су подешени за ту базу података. Адреса хоста се мора написати у виду IP адресе, а не као “localhost”. Сваки пут када се у Djangu изврши нека измена над базом, потребно је мигрирати базу на сервер користећи команде python manage.py makemigrations approda и python manage.py migrate.



Слика 20 - подешавања базе

Важно је напоменути да је раније креирана база података морала бити измењена због потреба апликације – Django не подржава коришћење сложених примарних (нити спољних) кључева у моделима, а сваки модел мора имати свој примарни кључ. Због тога су child табеле које нису имале свој примарни кључ (евиденције пријема и ствке рачуна) већ пренесене кључеве морале бити измењене – пренесени примарни кључеви постали су само спољни, а креиране су нове колоне које престављају целобројне auto-increment примарне кључеве (pk\_evidencije\_prijema и pk\_stavke\_racuna). Такође, морала је бити отклоњена веза од шифре добављача из табеле отпремнице ка табели евиденције пријема, јер ни сложени спољни кључеви не могу постојати у Django моделима. Финални модел базе коришћен за апликацију изгледа овако:



Слика 21 - финални модел базе

# Подаци за тестирање

## 10.1 Табела КУПЦИ

INSERT INTO KUPCI (jmbg, ime, prezime, kartica)

VALUES (2806966554444, 'Mona', 'Liza', 2222333344445555);

INSERT INTO KUPCI (jmbg, ime, prezime, kartica)

VALUES (2006966552444, 'Marko', 'Nikolic', 9999888899998888);

INSERT INTO KUPCI (jmbg, ime, prezime, kartica)

VALUES (2706966553444, 'Nikola', 'Urosevic', 1111000011110000);

INSERT INTO KUPCI (jmbg, ime, prezime, kartica)

VALUES (2106966551444, 'Lazar', 'Matic', 2345234567896789);

INSERT INTO KUPCI (jmbg, ime, prezime, kartica)

VALUES (1806966554444, 'Sanja', 'Lalovic', 2323232323232323);

INSERT INTO KUPCI (jmbg, ime, prezime, kartica)

VALUES (2306966553444, 'Nina', 'Milosevic',5555444466667777);

INSERT INTO KUPCI (jmbg, ime, prezime, kartica)

VALUES (2206966533444, 'Jovan', 'Milosavljevic', 7777666655554444);

INSERT INTO KUPCI (jmbg, ime, prezime, kartica)

VALUES (1206966533444, 'Jovana', 'Janjic', 1111666644442222);

INSERT INTO KUPCI (jmbg, ime, prezime, kartica)

VALUES (1406966533444, 'Nikolina', 'Kutlesic', 7566344212236667);

INSERT INTO KUPCI (jmbg, ime, prezime, kartica)

VALUES (1606966533444, 'Lola', 'Gavrilovic', 8880999844436661);



Слика 22 - попуњена табела КУПЦИ

## 10.2 Табела РАЧУНИ

INSERT INTO RACUNI (racunID, datum, vreme, JMBG)

VALUES (1111111111, '10/10/2018', '10:53:44',1606966533444);

INSERT INTO RACUNI (racunID, datum, vreme, JMBG)

VALUES (1111111112, '10/10/2018', '10:54:22',1606966533444);

INSERT INTO RACUNI (racunID, datum, vreme, JMBG)

VALUES (1111111113, '10/10/2018', '10:55:23',2006966552444);

INSERT INTO RACUNI (racunID, datum, vreme, JMBG)

VALUES (1111111114, '10/10/2018', '10:57:13',2006966552444);

INSERT INTO RACUNI (racunID, datum, vreme, JMBG)

VALUES (1111111115, '10/10/2018', '11:01:14',2106966551444);

INSERT INTO RACUNI (racunID, datum, vreme, JMBG)

VALUES (1111111116, '10/10/2018', '11:11:16',2806966554444);

INSERT INTO RACUNI (racunID, datum, vreme, JMBG)

VALUES (1111111117, '10/10/2018', '11:12:24',2806966554444);

INSERT INTO RACUNI (racunID, datum, vreme, JMBG)

VALUES (1111111118, '10/10/2018', '11:13:12',2106966551444);

INSERT INTO RACUNI (racunID, datum, vreme, JMBG)

VALUES (1111111119, '10/10/2018', '11:15:25',2806966554444);

INSERT INTO RACUNI (racunID, datum, vreme, JMBG)

VALUES (1111111110, '10/10/2018', '11:22:22',2006966552444);



Слика 23 - попуњена табела РАЧУНИ

## 10.3 Табела АРТИКЛИ

INSERT INTO ARTIKLI (sifra\_artikla, naziv, cena, jedinica\_mere, rok\_trajanja)

VALUES (222222222221, 'Nutela', 294, 'gram', '10/10/2020');

INSERT INTO ARTIKLI (sifra\_artikla, naziv, cena, jedinica\_mere, rok\_trajanja)

VALUES (222222222222, 'Eurokrem', 294, 'gram', '10/10/2019');

INSERT INTO ARTIKLI (sifra\_artikla, naziv, cena, jedinica\_mere, rok\_trajanja)

VALUES (222222222223, 'Fanta', 120, 'litar', '10/10/2019');

INSERT INTO ARTIKLI (sifra\_artikla, naziv, cena, jedinica\_mere, rok\_trajanja)

VALUES (222222222224, 'Koka Kola', 120, 'litar', '10/10/2018');

INSERT INTO ARTIKLI (sifra\_artikla, naziv, cena, jedinica\_mere, rok\_trajanja)

VALUES (222222222225, 'Salama', 85, 'gram', '12/31/2018');

INSERT INTO ARTIKLI (sifra\_artikla, naziv, cena, jedinica\_mere, rok\_trajanja)

VALUES (222222222226, 'Hleb', 40, 'komad', '12/31/2018');

INSERT INTO ARTIKLI (sifra\_artikla, naziv, cena, jedinica\_mere, rok\_trajanja)

VALUES (222222222227, 'Sir', 60, 'gram', '10/02/2019');

INSERT INTO ARTIKLI (sifra\_artikla, naziv, cena, jedinica\_mere, rok\_trajanja)

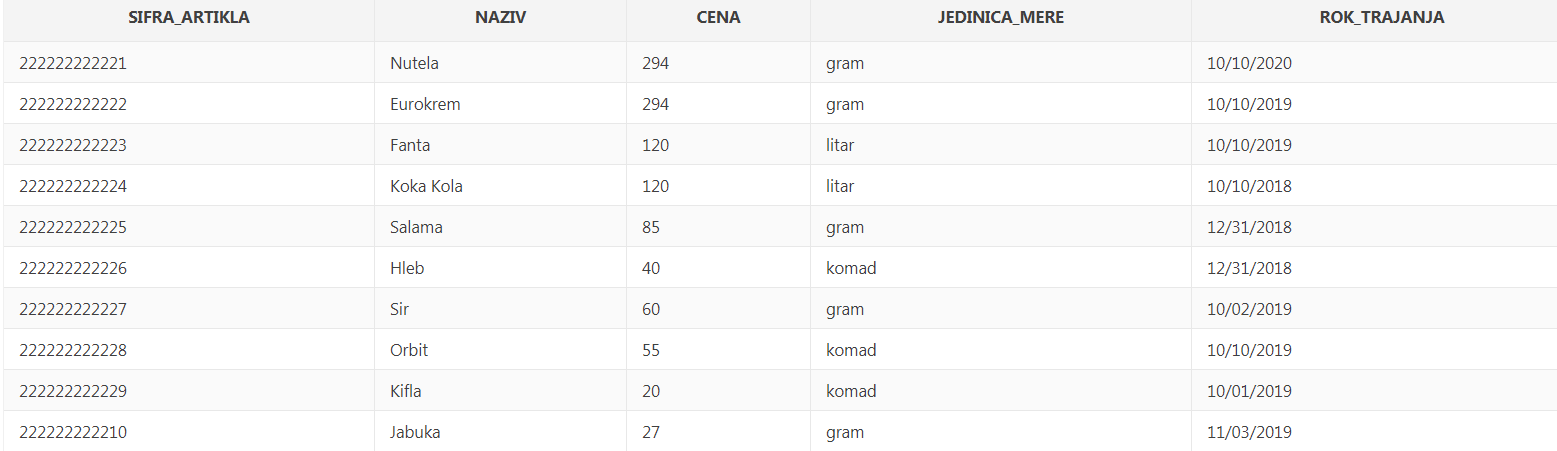
VALUES (222222222228, 'Orbit', 55, 'komad', ' 10/10/2019');

INSERT INTO ARTIKLI (sifra\_artikla, naziv, cena, jedinica\_mere, rok\_trajanja)

VALUES (222222222229, 'Kifla', 20, 'komad', ' 10/01/2019');

INSERT INTO ARTIKLI (sifra\_artikla, naziv, cena, jedinica\_mere, rok\_trajanja)

VALUES (222222222210, 'Jabuka', 27, 'gram', ' 11/03/2019');



Слика 24 - попуњена табела АРТИКЛИ

## 10.4 Табела СТАВКЕ\_РАЧУНА

INSERT INTO STAVKE\_RACUNA (kolicina, sifra\_artikla, racunID)

VALUES (2, 222222222221, 1111111113);

INSERT INTO STAVKE\_RACUNA (kolicina, sifra\_artikla, racunID)

VALUES (1, 222222222222, 1111111114);

INSERT INTO STAVKE\_RACUNA (kolicina, sifra\_artikla, racunID)

VALUES (3, 222222222224, 1111111111);

INSERT INTO STAVKE\_RACUNA (kolicina, sifra\_artikla, racunID)

VALUES (5, 222222222224, 1111111112);

INSERT INTO STAVKE\_RACUNA (kolicina, sifra\_artikla, racunID)

VALUES (7, 222222222227, 1111111112);

INSERT INTO STAVKE\_RACUNA (kolicina, sifra\_artikla, racunID)

VALUES (50, 222222222228, 1111111112);

INSERT INTO STAVKE\_RACUNA (kolicina, sifra\_artikla, racunID)

VALUES (3, 222222222228, 1111111116);

INSERT INTO STAVKE\_RACUNA (kolicina, sifra\_artikla, racunID)

VALUES (1, 222222222229, 1111111117);

INSERT INTO STAVKE\_RACUNA (kolicina, sifra\_artikla, racunID)

VALUES (3, 222222222227, 1111111116);

INSERT INTO STAVKE\_RACUNA (kolicina, sifra\_artikla, racunID)

VALUES (4, 222222222228, 1111111113);



Слика 25 - попуњена табела СТАВКЕ\_РАЧУНА

## 10.5 Табела ОБЈЕКТИ

INSERT INTO OBJEKTI (PIB, naziv, adresa, grad, broj\_telefona)

VALUES (3333333331, 'Roda D.O.O', ' Svetosavska 37a', 'Beočin', 011245246);

INSERT INTO OBJEKTI (PIB, naziv, adresa, grad, broj\_telefona)

VALUES (3333333332, 'Roda Megamarket', ' Maršala Tita 79', 'Beograd', 011245247);

INSERT INTO OBJEKTI (PIB, naziv, adresa, grad)

VALUES (3333333333, 'Roda, Stadion', ' Đure Jakšića BB', 'Beograd');

INSERT INTO OBJEKTI (PIB, naziv, adresa, grad)

VALUES (3333333334, 'Roda, Ušće', ' Niska 44', 'Beograd');

INSERT INTO OBJEKTI (PIB, naziv, adresa, grad)

VALUES (3333333335, 'Roda Headquarters', ' Sarajevska 55 ', 'Beograd');

INSERT INTO OBJEKTI (PIB, naziv, adresa, grad)

VALUES (3333333336, 'Roda Merkator', ' Vojvode Stepe 14 ', 'Beograd');

INSERT INTO OBJEKTI (PIB, naziv, adresa, grad)o.

VALUES (3333333337, 'Roda', ' Makenzijeva 53 ', 'Beograd');

INSERT INTO OBJEKTI (PIB, naziv, adresa, grad)

VALUES (3333333338, 'Roda Maloprodaja', ' Braničevska 2 ', 'Beograd');

INSERT INTO OBJEKTI (PIB, naziv, adresa, grad)

VALUES (3333333339, 'Roda', ' Njegoševa 14 ', 'Cacak');

INSERT INTO OBJEKTI (PIB, naziv, adresa, grad, broj\_telefona)

VALUES (3333333310, 'Roda Centar', ' Kneginje Milice bb ', 'Jagodina',035212244);



Слика 26 - попуњена табела ОБЈЕКТИ

## 10.6 Табела ДОБАВЉАЧИ

INSERT INTO DOBAVLJACI (sifra\_dobavljaca, naziv)

VALUES (4444444441, 'Yumis');

INSERT INTO DOBAVLJACI (sifra\_dobavljaca, naziv)

VALUES (4444444442, 'Swisslion');

INSERT INTO DOBAVLJACI (sifra\_dobavljaca, naziv)

VALUES (4444444443, 'Nectar');

INSERT INTO DOBAVLJACI (sifra\_dobavljaca, naziv)

VALUES (4444444444, 'Centroproizvod');

INSERT INTO DOBAVLJACI (sifra\_dobavljaca, naziv)

VALUES (4444444445, 'Pionir');

INSERT INTO DOBAVLJACI (sifra\_dobavljaca, naziv)

VALUES (4444444446, 'FHI Zdravlje AD');

INSERT INTO DOBAVLJACI (sifra\_dobavljaca, naziv)

VALUES (4444444447, 'ALFA-PLAM');

INSERT INTO DOBAVLJACI (sifra\_dobavljaca, naziv)

VALUES (4444444448, 'HIP');

INSERT INTO DOBAVLJACI (sifra\_dobavljaca, naziv)

VALUES (4444444449, 'Impol');

INSERT INTO DOBAVLJACI (sifra\_dobavljaca, naziv)

VALUES (4444444410, 'MIK-KOMERC');



Слика 27 - попуњена табела ДОБАВЉАЧИ

## 10.7 Табела ОТПРЕМНИЦЕ

INSERT INTO OTPREMNICE (sifra\_otpremnice, datum\_otpreme, sifra\_dobavljaca)

VALUES (555555555551, '10/10/2018', 4444444441);

INSERT INTO OTPREMNICE (sifra\_otpremnice, datum\_otpreme, sifra\_dobavljaca)

VALUES (555555555552, '10/10/2018', 4444444442);

INSERT INTO OTPREMNICE (sifra\_otpremnice, datum\_otpreme, sifra\_dobavljaca)

VALUES (555555555553, '10/10/2018', 4444444443);

INSERT INTO OTPREMNICE (sifra\_otpremnice, datum\_otpreme, sifra\_dobavljaca)

VALUES (555555555554, '10/10/2018', 4444444444);

INSERT INTO OTPREMNICE (sifra\_otpremnice, datum\_otpreme, sifra\_dobavljaca)

VALUES (555555555555, '10/10/2018', 4444444445);

INSERT INTO OTPREMNICE (sifra\_otpremnice, datum\_otpreme, sifra\_dobavljaca)

VALUES (555555555556, '10/10/2018', 4444444446);

INSERT INTO OTPREMNICE (sifra\_otpremnice, datum\_otpreme, sifra\_dobavljaca)

VALUES (555555555557, '10/10/2018', 4444444447);

INSERT INTO OTPREMNICE (sifra\_otpremnice, datum\_otpreme, sifra\_dobavljaca)

VALUES (555555555558, '10/10/2018', 4444444448);

INSERT INTO OTPREMNICE (sifra\_otpremnice, datum\_otpreme, sifra\_dobavljaca)

VALUES (555555555559, '10/10/2018', 4444444449);

INSERT INTO OTPREMNICE (sifra\_otpremnice, datum\_otpreme, sifra\_dobavljaca)

VALUES (555555555510, '10/10/2018', 4444444410);



Слика 28 - попуњена табела ОТПРЕМНИЦЕ

## 10.8 Табела ЕВИДЕНЦИЈЕ\_ПРИЈЕМА

INSERT INTO EVIDENCIJE\_PRIJEMA (kolicina, sifra\_artikla, PIB, sifra\_otpremnice)

VALUES (000202, 222222222210, 3333333331, 555555555551);

INSERT INTO EVIDENCIJE\_PRIJEMA (kolicina, sifra\_artikla, PIB, sifra\_otpremnice)

VALUES (001001, 222222222210, 3333333332, 555555555551);

INSERT INTO EVIDENCIJE\_PRIJEMA (kolicina, sifra\_artikla, PIB, sifra\_otpremnice)

VALUES (001000, 222222222226, 3333333333, 555555555552);

INSERT INTO EVIDENCIJE\_PRIJEMA (kolicina, sifra\_artikla, PIB, sifra\_otpremnice)

VALUES (001005, 222222222226, 3333333335, 555555555552);

INSERT INTO EVIDENCIJE\_PRIJEMA (kolicina, sifra\_artikla, PIB, sifra\_otpremnice)

VALUES (000037, 222222222229, 3333333334, 555555555555);

INSERT INTO EVIDENCIJE\_PRIJEMA (kolicina, sifra\_artikla, PIB, sifra\_otpremnice)

VALUES (000051, 222222222229, 3333333334, 555555555554);

INSERT INTO EVIDENCIJE\_PRIJEMA (kolicina, sifra\_artikla, PIB, sifra\_otpremnice)

VALUES (000103, 222222222221, 3333333337, 555555555558);

INSERT INTO EVIDENCIJE\_PRIJEMA (kolicina, sifra\_artikla, PIB, sifra\_otpremnice)

VALUES (000251, 222222222221, 3333333337, 555555555557);

INSERT INTO EVIDENCIJE\_PRIJEMA (kolicina, sifra\_artikla, PIB, sifra\_otpremnice)

VALUES (000303, 222222222223, 3333333338, 555555555510);

INSERT INTO EVIDENCIJE\_PRIJEMA (kolicina, sifra\_artikla, PIB, sifra\_otpremnice)

VALUES (000504, 222222222223, 3333333338, 555555555559);



Слика 29 - попуњена табела ЕВИДЕНЦИЈЕ\_ПРИЈЕМА

# Опис апликације са упутством за коришћење

Апликација која је наисана у Python-у уз Django framework је осмишљена да буде администраторски сајт Рода мегамаркета, што значи да не постоје функционалности које може користи неко неовлашћено лице, тј. обичан корисник/купац. У овом поглављу биће у кратким цртама објашњене опције које има апликација.

## 11.1 Почетна страна

Како би се покренула апликација, потребно је бити у директоријуму апликације, покренути command

prompt и откуцати наредбу python manage.py runserver. Отвара се страница са URL-ом <http://localhost:8000/approda/> - почетна страница. На њој је могуће преусмерити се на форму за додавање евиденција пријема, објеката, рачуна, ставки рачуна, отпремница или добављача путем buttona. Такође се може отићи на приказ било које од наведених табела одабиром опција у navbar-у.



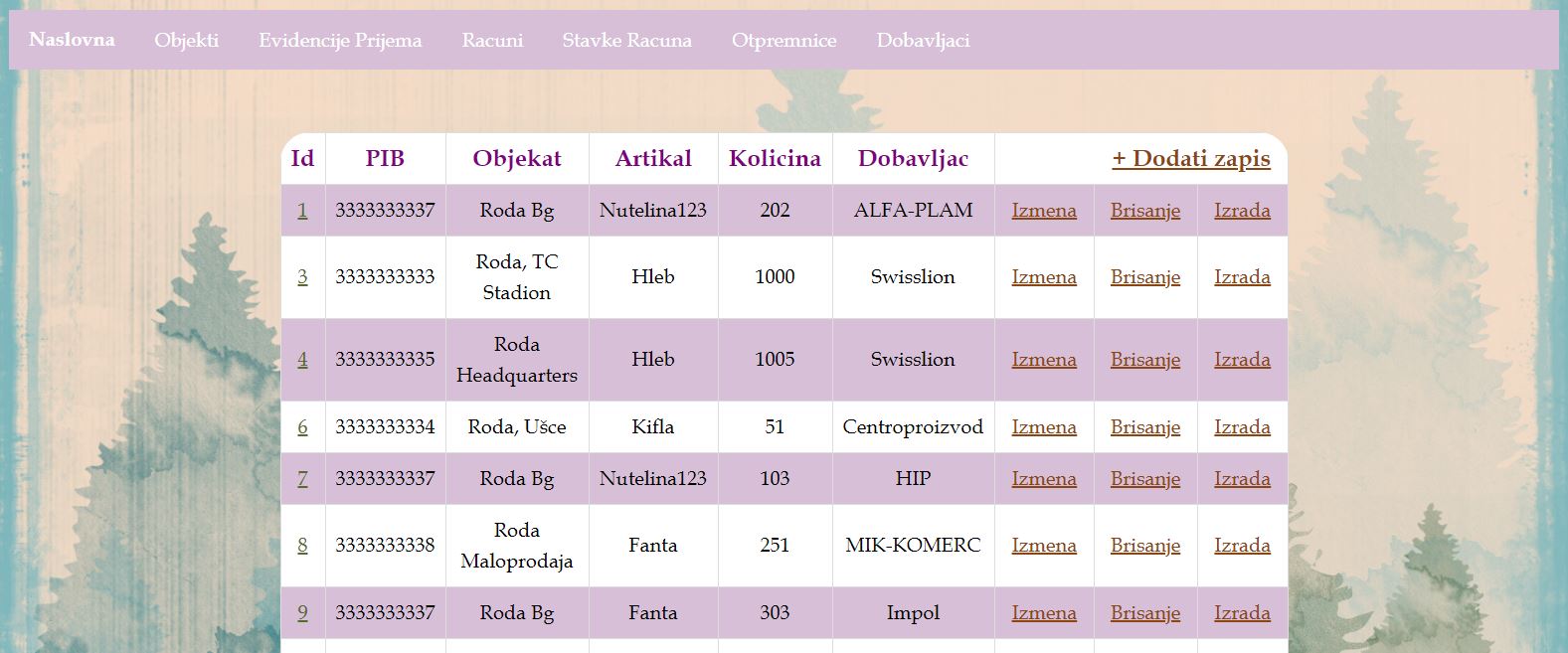
Слика 30 - почетна страна

Све наведене табеле имају своју страницу за приказ свих записа, приказ података о једном запису, додавање, измену и брисање. Евиденције пријема такође имају могућност “израде” документа. Сајт је направљен са циљем приказивања рада тригера за оптимизацију (табеле рачуни + ставке рачуна) и једну денормализацију (објекти + евиденције пријема).

Упутства у овом поглављу биће приказана на примеру за евиденције пријема, а остале табеле имају исте функционалности.

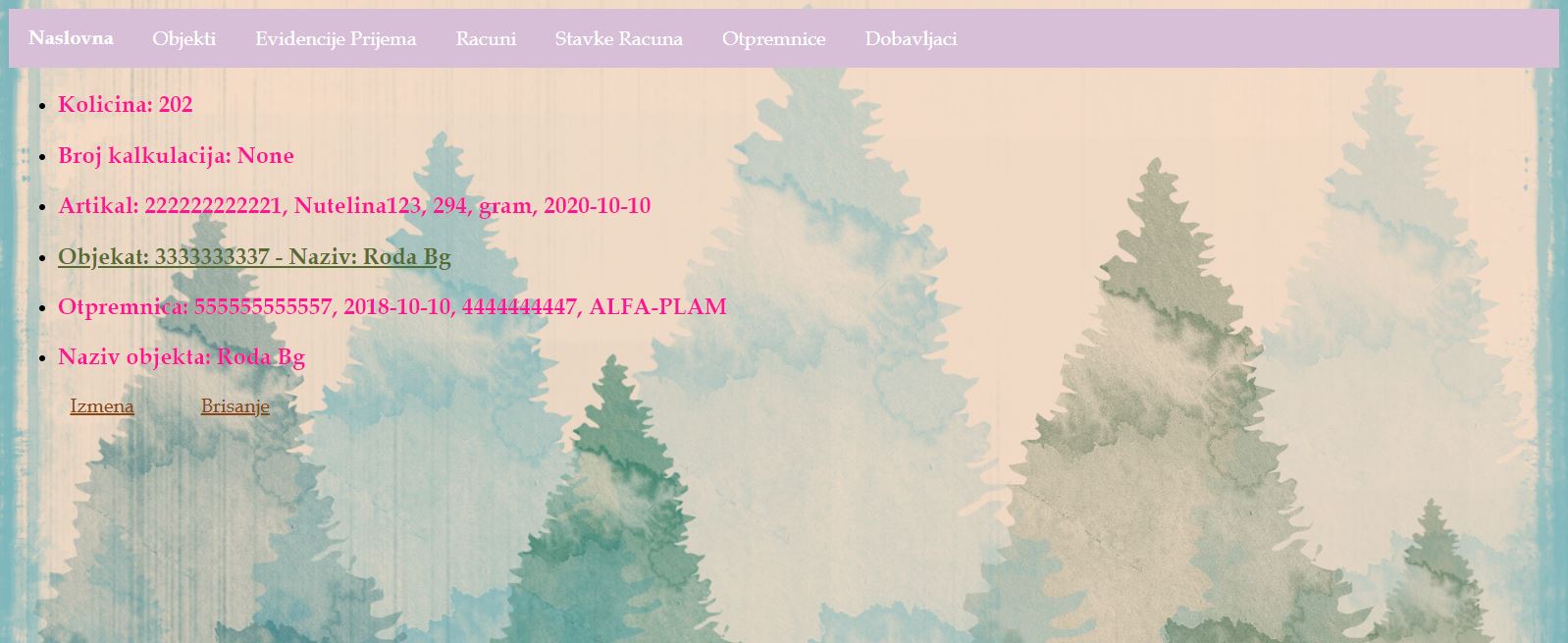
## 11.2 Приказ

Кликом на опцију Евиденције пријема из навигационог бара отвара се приказ свих евиденција пријема са опцијама за измену, брисање и израду документа сваког записа, као и додавање новог записа.



Слика 31 - приказ свих записа

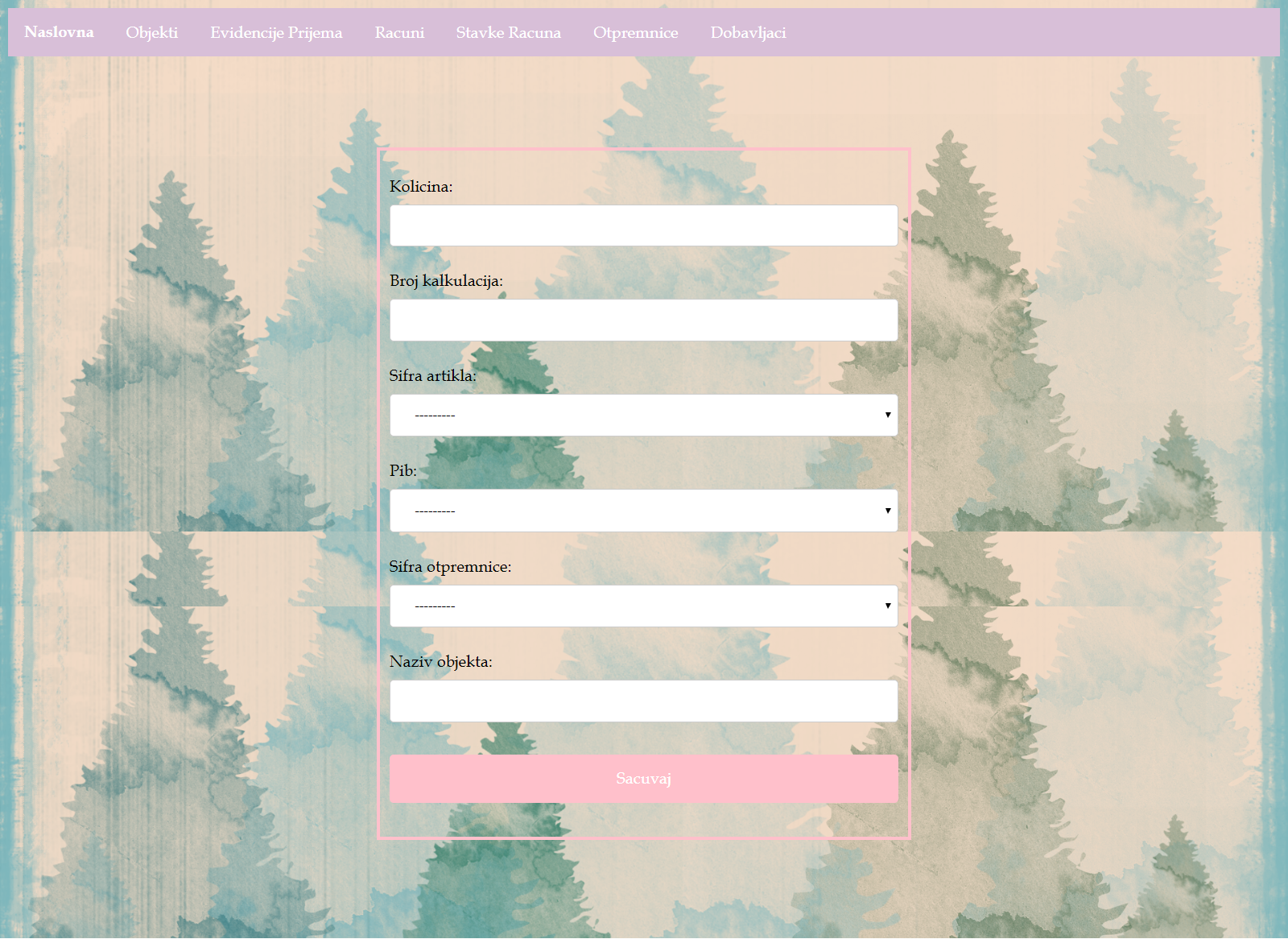
Кликом на линковани Id за сваки запис отвара се страница која приказује детаље о том запису. Детаљи су исписани у виду неуређене листе и међу њима је линк ка објекту за који је евиденција пријема везана. Као и у табели приказа свих записа, и овде постоји могућност измене и брисања записа.



Слика 32 - приказ детаља записа

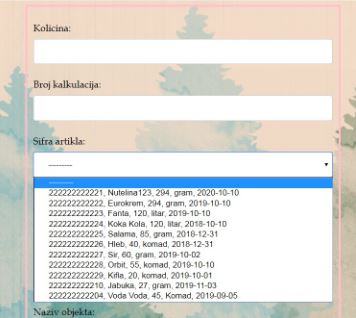
## 11.3 Унос, измена и брисање

Кликом на “+ Dodati zapis”, или на одговарајуће дугме на почетној страни, отвара се форма за унос новог записа у бази података. Све форме садрже сва поља из табела и обавезно их је попунити. Испод је форма за унос нове евиденције пријема:



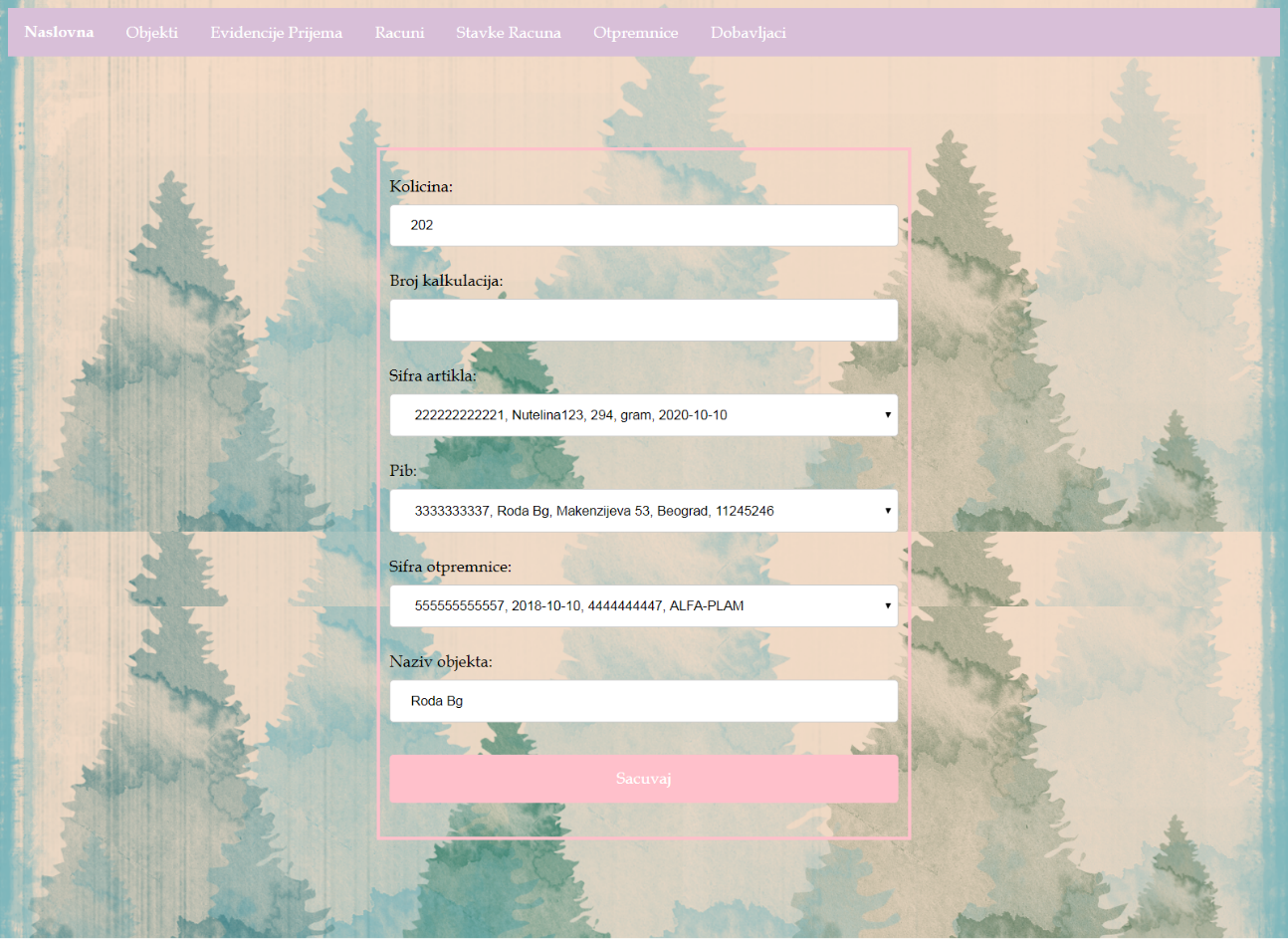
Слика 33 - форма за унос

Спољне кључеве није потребно напамет куцати, већ се бирају из падајуће листе које садрже примарне кључеве, као и остале податке о објекту. Кликом на дугме “Sacuvaj”, нови запис се уноси у базу података.



Слика 34 - одабир спољног кључа

Када се кликне на линк “Izmena” из табеле свих објеката или из детаља о објекту, отвара се форма за измену података записа која је идентична оној за унос, само што су у овом случају сетовани подаци записа који се могу мењати.



Слика 35 - форма за измену

Када је потребно обрисати објекат, отвара се страница са питањем за потрврду брисања. Кликом на “Da” објекат се брише из базе, а кликом на “Nazad” корисник се враћа на приказ свих објеката у табели.



Слика 36 - брисање записа

## 11.4 Израда документа

Евиденције пријема имају додатну могућност прављења документа евиденције пријема, који повезује податке из табеле добављачи, отпремнице, објекти и евиденције пријема. Ова функционалност је направљњна као пример једног документа који је излаз из потпроцеса овог система.



Слика 37 - израда документа

# Закључак

Овај пројекат је направљен као кратак опис пословног система Рода мегамаркета и неких од специфичности процеса и потпроцеса који се извршавају на дневном нивоу. База података је осмишљена да приближно прикаже како тај реалан систем функционише, на основу добијених докумената од калкуланта Роде мегамаркета. Апликација направљена за демонстрацију рада базе гледа на систем из угла администратора и стога има одговарајуће опције. Нека од унапређења која се могу извшити у апликацији укључују:

* Додавање могућности управљања табелама КУПЦИ и АРТИКЛИ на сајту
* Додавање могућности израде осталих докумената система
* Побољшати изглед сајта

# Индекс слика

[Слика 1 - отпремница добављача 7](#_Toc18927624)

[Слика 2 - евиденција пријема 8](#_Toc18927625)

[Слика 3 – ценовник 9](#_Toc18927626)

[Слика 4 - рачун за физичка лица 10](#_Toc18927627)

[Слика 5 - модел објекти-везе пре нормализације 11](#_Toc18927628)

[Слика 6 - модел објекти-везе након нормализације 12](#_Toc18927629)

[Слика 7 - денормализација 1 13](#_Toc18927630)

[Слика 8 - денормализација 2 17](#_Toc18927631)

[Слика 9 - оптимизација Storing derivable values 21](#_Toc18927632)

[Слика 10 - релациона шема базе података 28](#_Toc18927633)

[Слика 11 - табела КУПЦИ 29](#_Toc18927634)

[Слика 12 - табела РАЧУНИ 30](#_Toc18927635)

[Слика 13 - табела АРТИКЛИ 30](#_Toc18927636)

[Слика 14 - табела СТАВКЕ\_РАЧУНА 31](#_Toc18927637)

[Слика 15 - табела ОБЈЕКТИ 32](#_Toc18927638)

[Слика 16 - табела ДОБАВЉАЧИ 32](#_Toc18927639)

[Слика 17 - табела ОТПРЕМНИЦЕ 33](#_Toc18927640)

[Слика 18 - табела ЕВИДЕНЦИЈЕ\_ПРИЈЕМА 33](#_Toc18927641)

[Слика 19 - дијаграм случајева коришћења 34](#_Toc18927642)

[Слика 20 - подешавања базе 35](#_Toc18927643)

[Слика 21 - финални модел базе 36](#_Toc18927644)

[Слика 22 - попуњена табела КУПЦИ 38](#_Toc18927645)

[Слика 23 - попуњена табела РАЧУНИ 39](#_Toc18927646)

[Слика 24 - попуњена табела АРТИКЛИ 40](#_Toc18927647)

[Слика 25 - попуњена табела СТАВКЕ\_РАЧУНА 42](#_Toc18927648)

[Слика 26 - попуњена табела ОБЈЕКТИ 43](#_Toc18927649)

[Слика 27 - попуњена табела ДОБАВЉАЧИ 44](#_Toc18927650)

[Слика 28 - попуњена табела ОТПРЕМНИЦЕ 45](#_Toc18927651)

[Слика 29 - попуњена табела ЕВИДЕНЦИЈЕ\_ПРИЈЕМА 47](#_Toc18927652)

[Слика 30 - почетна страна 48](#_Toc18927653)

[Слика 31 - приказ свих записа 49](#_Toc18927654)

[Слика 32 - приказ детаља записа 49](#_Toc18927655)

[Слика 33 - форма за унос 50](#_Toc18927656)

[Слика 34 - одабир спољног кључа 51](#_Toc18927657)

[Слика 35 - форма за измену 51](#_Toc18927658)

[Слика 36 - брисање записа 52](#_Toc18927659)

[Слика 37 - израда документа 52](#_Toc18927660)

# Литература

[1] Wikipedia, <https://sr.wikipedia.org/wiki/Roda_(megamarketi)> , 01.09.2019.

[2] Roda – zvanični sajt, <https://www.roda.rs/> , 01.09.2019.

[3] Eučenje, FTN Čačak, Programiranje baza podataka, [https://eucenje.ftn.kg.ac.rs/](https://eucenje.ftn.kg.ac.rs/course/view.php?id=110) , 08.09.2019.

[4] Try DJANGO Tutorial, <https://www.youtube.com/playlist?list=PLEsfXFp6DpzTD1BD1aWNxS2Ep06vIkaeW> , 09.09.2019.

[5] Oracle Docs, <https://docs.oracle.com/cd/E18283_01/appdev.112/e10766/tdddg_connecting.htm#CEGDIFBC> , 09.09.2019.

[6] Django Docs, <https://docs.djangoproject.com/en/2.2/topics/> , 09.09.2019.