Univerzitet u Kragujevcu

Fakultet inženjerskih nauka



**Baze Podataka**

Projektni zadatak:

APOTEKA-PRODAJA NELEKA

Student: Predmetni nastavnik:

Andjela Stanojevic 607/2019 Milan Erić

Aleksandar Đorđević

**Sadržaj:**

**1 Uvod . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .** 3

1.1 Relevantni dokumenti i potrebe realnog sistema . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

**2 ER dijagram . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4**

2.1 Entiteti . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .4

2.2 Veze . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

2.3 Kompletan Er dijagram . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

**3 Logicka šema relacione baze podataka . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14**

3.1 Entiteti . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .14

3.2 Veze . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .15

3.3 Međurelaciona ograničenja . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

3.4 Kompletna logička šema . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

**4 Fizička šema . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .17**

**5 Implementacija projektovane baze sa testnim podacima . . . . . . . . . . . 18**

**6 Literatura . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .31**

**1. Uvod**

**Apoteka**  je zdravstvena  ustanova koja nabavlja i ispituje lekove i [sanitetski materijal](https://sr.wikipedia.org/w/index.php?title=%D0%A1%D0%B0%D0%BD%D0%B8%D1%82%D0%B5%D1%82%D1%81%D0%BA%D0%B8_%D0%BC%D0%B0%D1%82%D0%B5%D1%80%D0%B8%D1%98%D0%B0%D0%BB&action=edit&redlink=1), i njima snabdeva stanovništvo. Apoteka je radno mesto farmaceuta. U apotekama farmaceuti, pored uloge zdravstvenog radnika, imaju i ulogu [prodavca](https://sr.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%BE%D0%B4%D0%B0%D0%B2%D0%B0%D1%86&action=edit&redlink=1" \o "Prodavac (stranica ne postoji)). U apoteci mora biti prisutan kvalifikovani farmaceut tokom čitavog radnog vremena.

U [Srbiji](https://sr.wikipedia.org/wiki/%D0%A1%D1%80%D0%B1%D0%B8%D1%98%D0%B0" \o "Srbija) je 1830 ovorena prva apoteka u Beogradu, 1836. je otvorena *Dvorska i voena apoteka* u [Kragujevcu](https://sr.wikipedia.org/wiki/%D0%9A%D1%80%D0%B0%D0%B3%D1%83%D1%98%D0%B5%D0%B2%D0%B0%D1%86), a 1862, u sastavu Glavne vojne bolnice u [Beogradu](https://sr.wikipedia.org/wiki/%D0%91%D0%B5%D0%BE%D0%B3%D1%80%D0%B0%D0%B4" \o "Beograd) je osnovana i prva [bolnička apoteka](https://sr.wikipedia.org/w/index.php?title=%D0%91%D0%BE%D0%BB%D0%BD%D0%B8%D1%87%D0%BA%D0%B0_%D0%B0%D0%BF%D0%BE%D1%82%D0%B5%D0%BA%D0%B0&action=edit&redlink=1" \o "Bolnička apoteka (stranica ne postoji)).

Većina apoteka je danas proširila svoj asortiman sa lekova na [kozmetiku](https://sr.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B7%D0%BC%D0%B5%D1%82%D0%B8%D0%BA%D0%B0" \o "Kozmetika), [dijetetske proizvode](https://sr.wikipedia.org/wiki/%D0%94%D0%B8%D1%98%D0%B5%D1%82%D0%B5%D1%82%D1%81%D0%BA%D0%B8_%D1%81%D1%83%D0%BF%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D0%B8" \o "Dijetetski suplementi), [zdravu hranu](https://sr.wikipedia.org/w/index.php?title=%D0%97%D0%B4%D1%80%D0%B0%D0%B2%D0%B0_%D1%85%D1%80%D0%B0%D0%BD%D0%B0&action=edit&redlink=1" \o "Zdrava hrana (stranica ne postoji)) i slične srodne proizvode, otuda i potreba za informacionim sistemom upravljanja apotekom, njihovom prodajom, nabavkom,..

U ovom projektnom zadatku biće objašnjen razvoj baze podataka informacionog sistema za upravljanje apotekom-prodajom neleka u cilju poboljšanja rada i usluga istih.

* 1. **Relevantni dokumenti i potrebe realnog sistema**

U našoj bazi podataka odlučili smo da čuvamo informacije bitne za svaku apoteku, porudzbine, kao i informacije o zaposlenima, dobavljacima, musterijama, medicinskoj opremi,kozmetici…

Bitne tačke ovog sistema su:

-apoteka

-zaposleni u apoteci

-proizvodi(higijena/nega/kozmetika, medicinska oprema)

-musterije

-dobavljaci

Veze između tačaka:

-musterije kupuju proizvode

-zaposleni narucuje proizvode

-dobavljac obezbedjuje proizvode

-zaposleni radi u apoteci

**2. ER dijagram**

Modelovanje sistema započinjemo konstruisanjem ER dijagrama. ER dijagram je kontruisan u programu Edraw Max. Zatim identifikujemo:

1) entitete – objekti ili događaji koji su nam od interesa za BP

2) veze – odnosi među entitetima

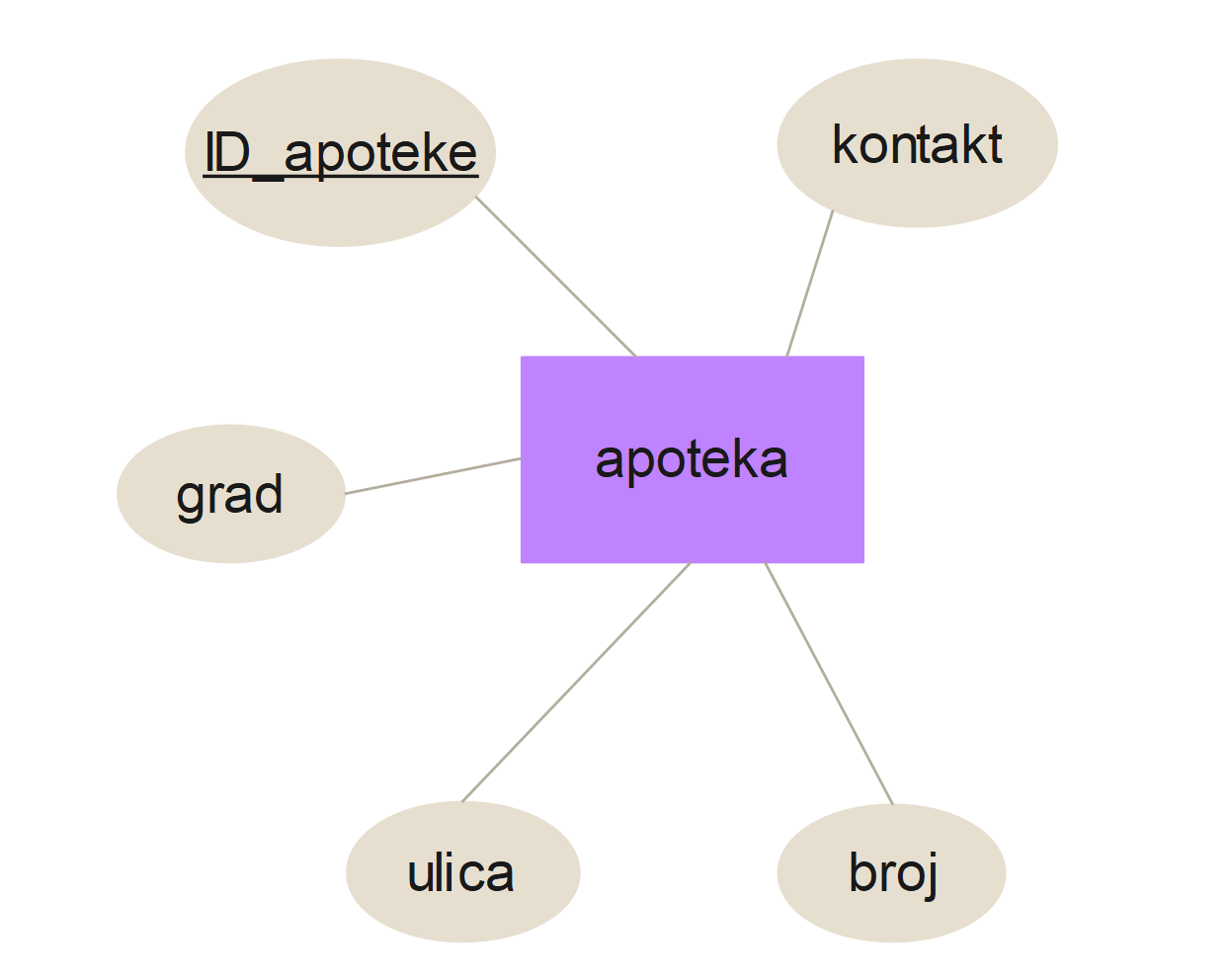
3) atribute – svojstva entiteta i veza

**2.1 Entiteti**

Identifikovali smo sledeće entitete:

* **Apoteka**

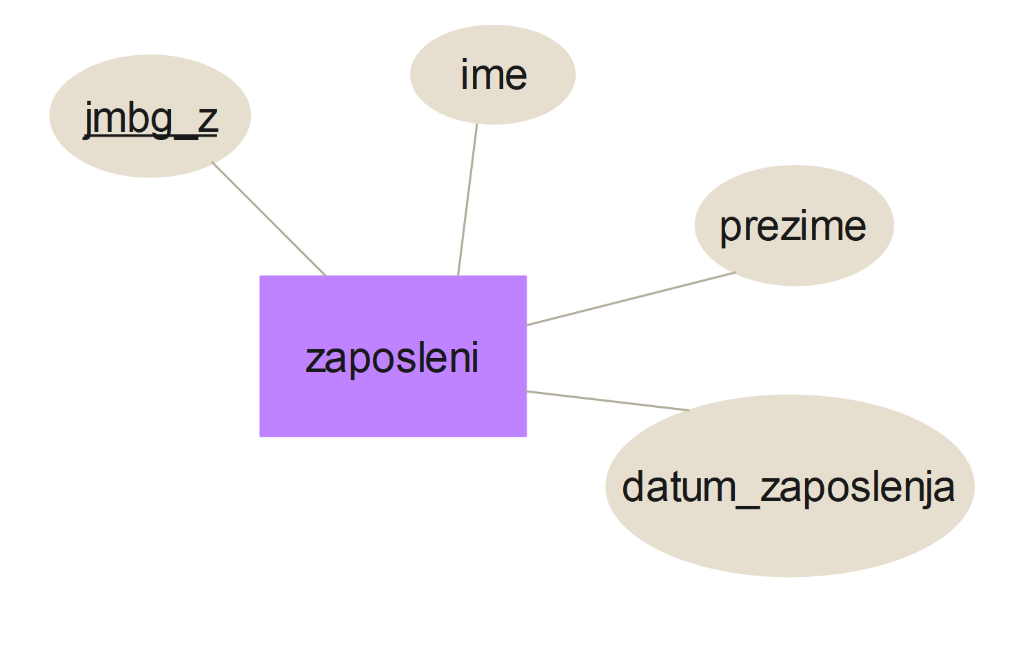
Entitet *apoteka* predstavlja jednu apoteku koja sadrži identifikator *ID\_apoteke*, atribute *kontakt, grad, ulica* i *broj* .



Slika 2.1.1: ER dijagram entiteta *apoteka*

* **Zaposleni**

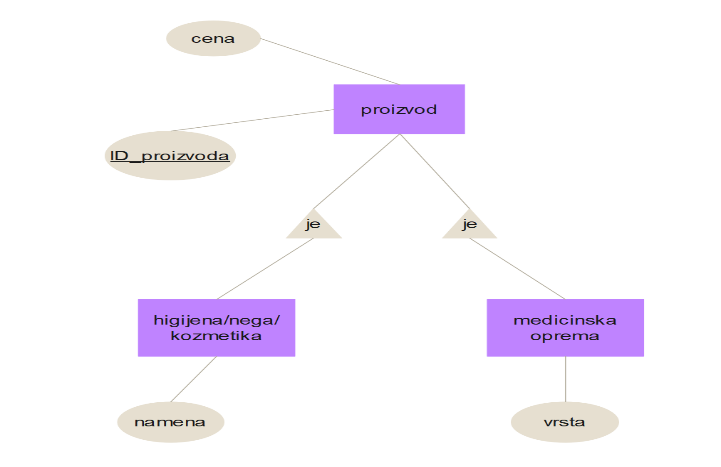
Identifikator radnika je *jmbg\_z*. Atributi su i *ime*, *prezime* i *datum\_zaposlenja*.



Slika 2.1.2: ER dijagram entiteta *zaposleni*

* **Nelek**

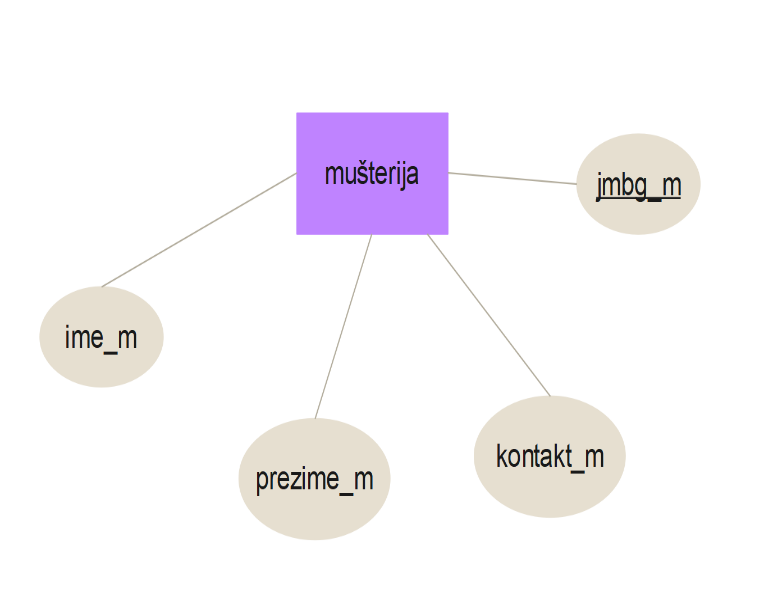
Entitet proizvod sadrži identifikator *id\_proizvoda* kao i atribut *cena*. Entitet proizvod takodje sadrzi i 2 podentiteta *medicinska oprema* i *ostalo.* Ova dva podentiteta nasledjuju od entiteta proizvod *id\_proizvoda* kao strani kljuc i preostali atribut cena. Takodje imaju i sopstvene atribute : *medicinska oprema* ima spostveni atribut *vrsta,* dok  *higijena/nega/kozmetika*  sadrzi sopstveni atribut namena.



Slika 2.1.3: ER dijagram entiteta *proizvod*

* **Musterija**

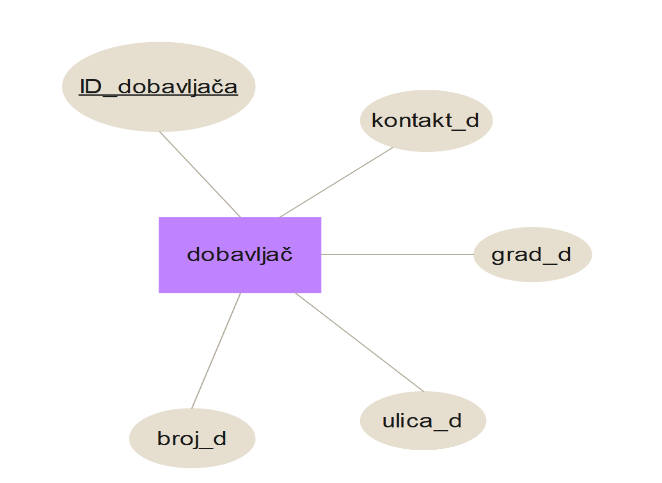
Entitet musterija ima sopstveni identifikator *jmbg\_m,* atribute *ime\_m, prezime\_m, kontakt\_m*. *Jmbg\_m* predstavlja jedinstveni maticni broj osobe koja je kupila neki nelek u apoteci.



Slika 2.1.4: ER dijagram entiteta *musterija*

* **Dobavljac**

Svaki proizvod ima određenog dobavljaca. Entitet dobavljac ima svoj jedinstveni identifikator *id\_dobavljaca* i ostale atribute *kontakt\_d, grad\_d, ulica\_d , broj\_d.* Uz pomoc dodatnih atributa dobijamo kontakt dobavljaca i njegovu adresu.



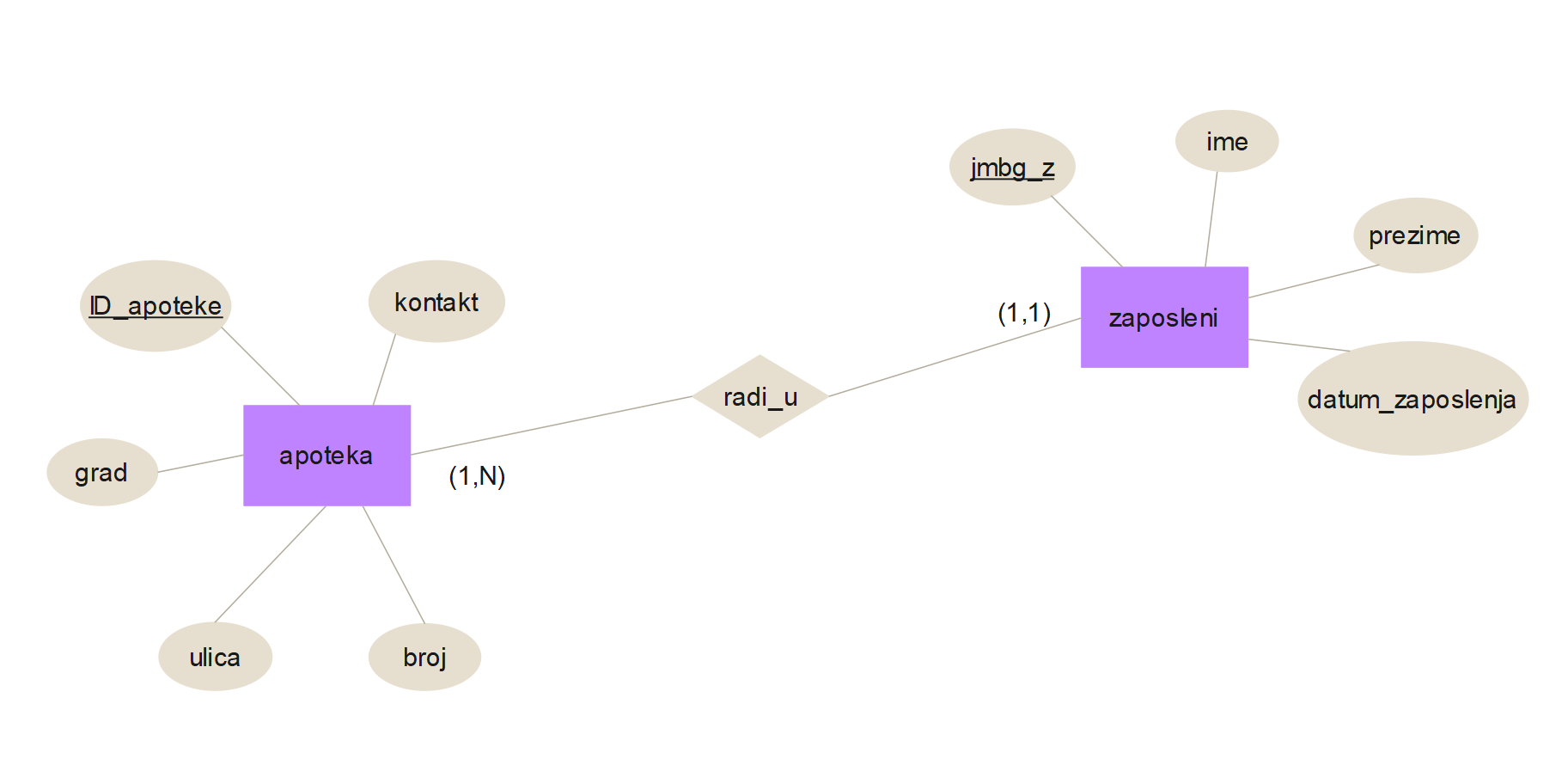
Slika 2.1.5: ER dijagram entiteta *dobavljac*

**2.2 Veze**

Sledi opis veza među entitetima:

* **Veza *apoteka--- zaposleni***

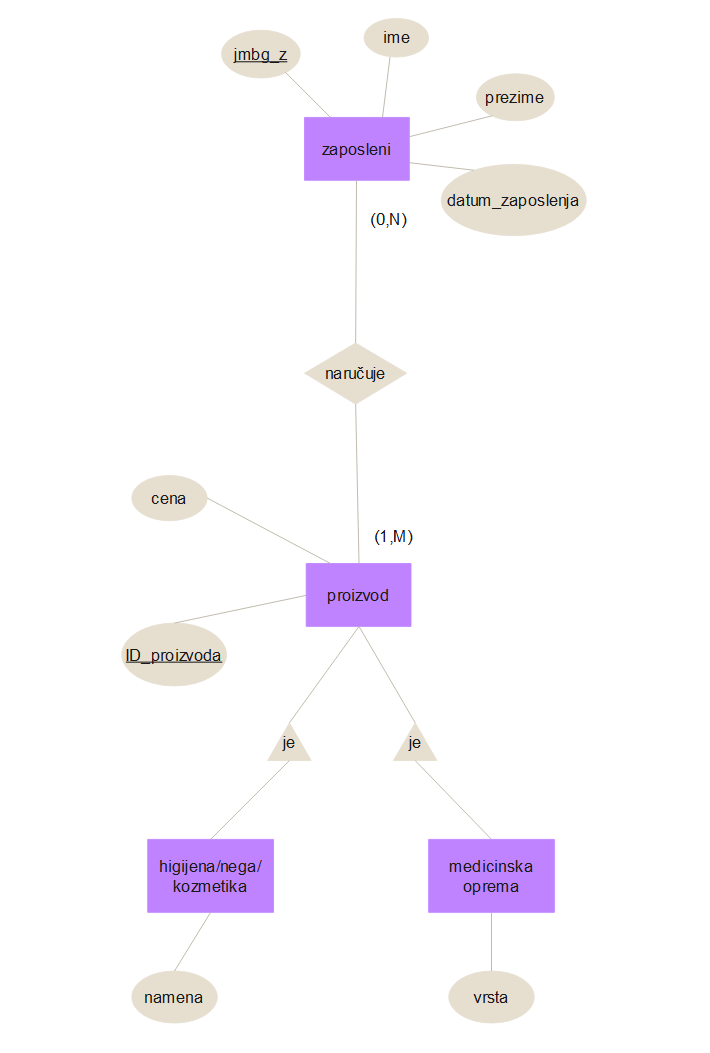
Veza sa strane entiteta *apoteka* je 1,N i u ovom slučaju ovaj entitet mora da ima najmanje jednog zaposlenog, a najčešće ima više. Veza sa strane entiteta *zaposleni* je 1,1. To znači da jedan zaposleni radi tačno i samo u jednoj apoteci . Kardinalnost veze je 1:N. Naziv ove veze je ***radi\_u****.*



Slika 2.2.1: ER dijagram veze *radi*\_u entiteta *apoteka* i *zaposleni*

* **Veza *zaposleni --- proizvod***

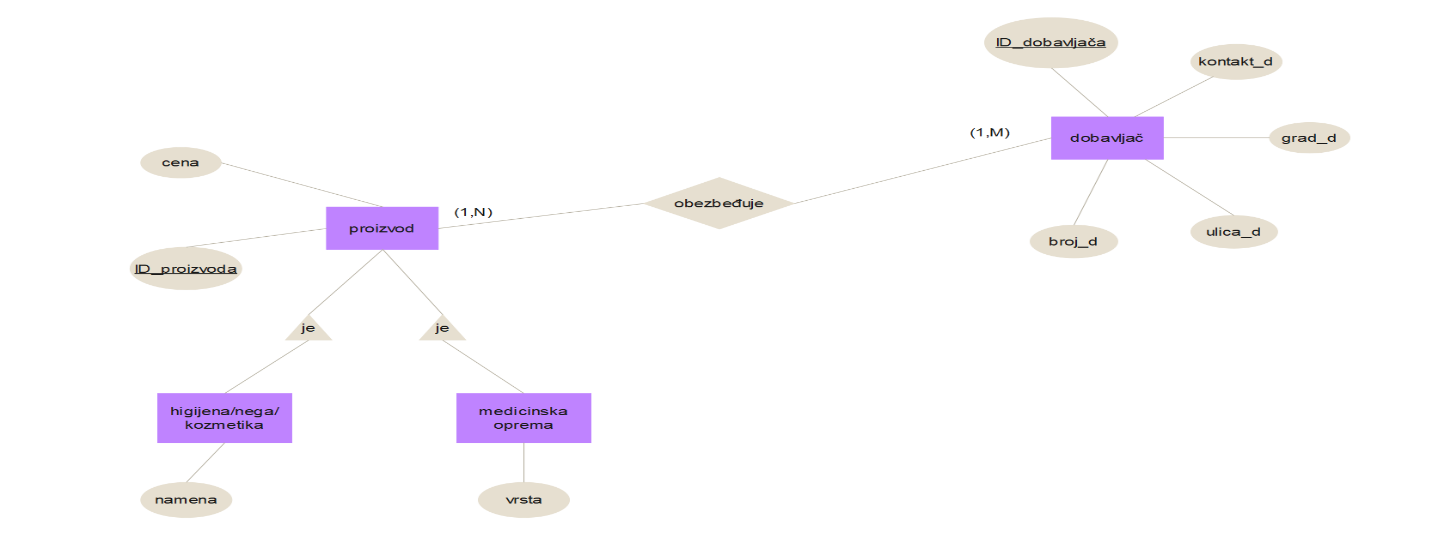
Veza sa strane entiteta *proizvod* je 1,M i u ovom slučaju ovaj entitet moze biti porucen najmanje 1 proizvod ili moze biti poruceno vise proizvoda. Veza sa strane entiteta *zapodlenog* je 0,N. To znači da jedan zaposleni ne mora da naruci nijedan proizvod ,a moze da naruci vise proizvoda. Kardinalnost veze je 1:M. Naziv ove veze je ***narucuje****.*



Slika 2.2.2: ER dijagram veze *narucuje* entiteta *zaposleni* i *proizvod*

* **Veza *proizvod --- dobavljac***

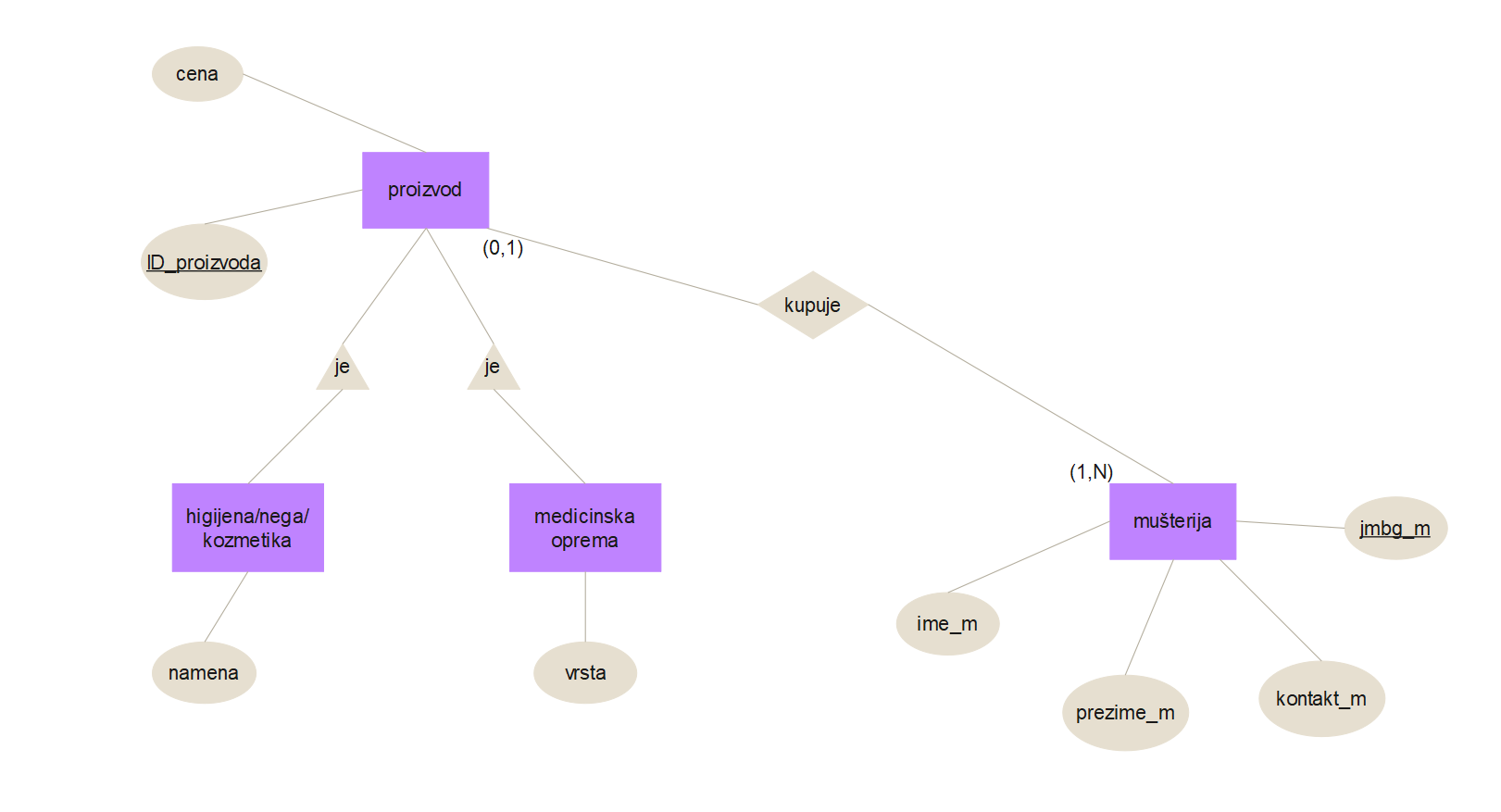
Veza sa strane entiteta *proizvod* je 1,N i to znači da se bar jedan proizvod moze obezbediti od srane dobavljaca, a moze se obezbediti i vise proizvoda. Veza sa strane entiteta *dobavljac* je 1,M sto znaci da moze postojati bar 1 dobavljac za neki proizvod, a moze postojati i vise njih (npr DUKA PHARM, FENIX, VEGA,..). Naziv ove veze je ***obezbedjuje*** .



Slika 2.2.3: ER dijagram veze *obezbedjuje* entiteta *proizvod* i *dobavljac*

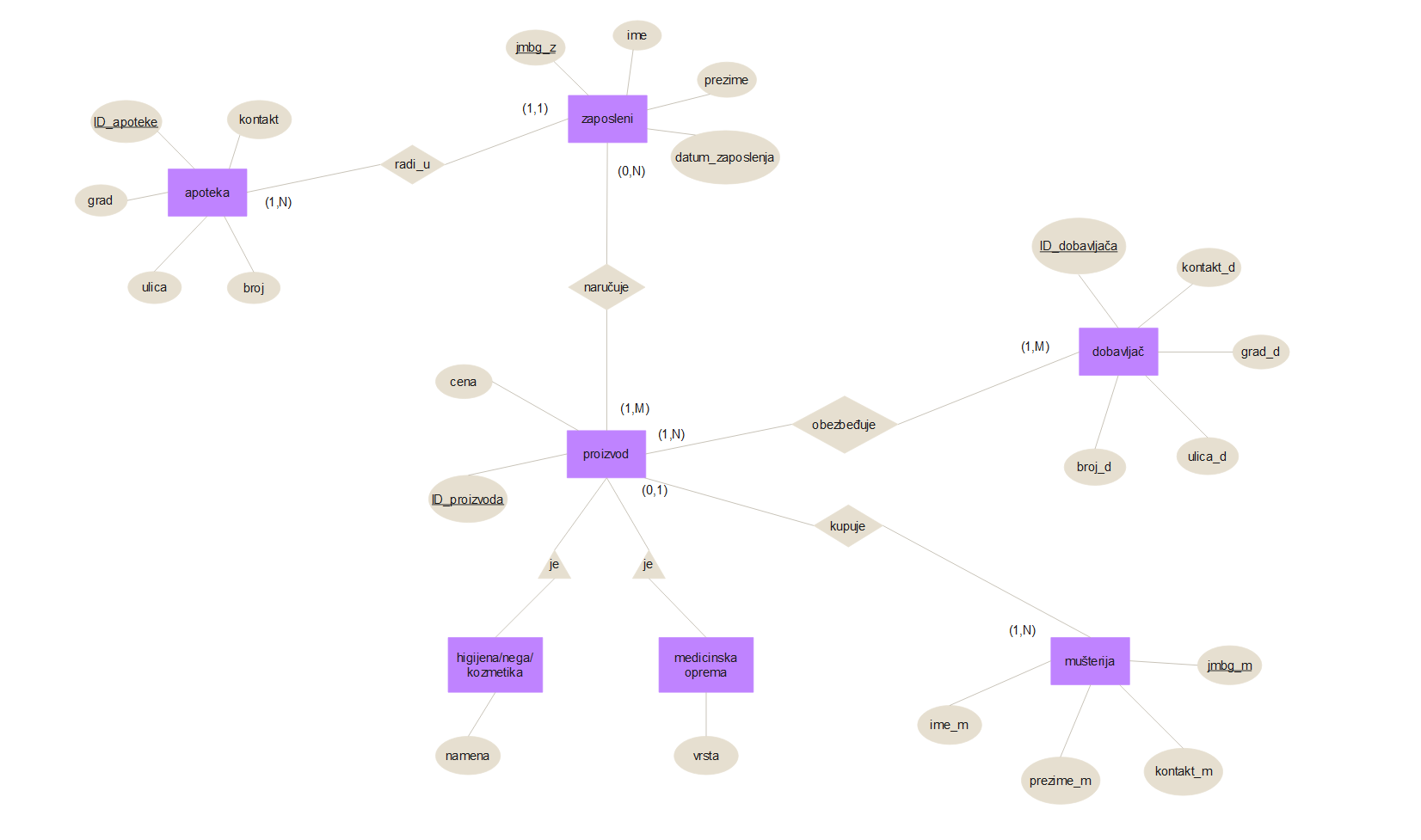
* **Veza *proizvod --- musterija***

Veza sa strane entiteta *proizvod* je 0,1 i to znači da jedan proizvod specificiran svojim ID-em moze biti kupljen , a i ne mora biti kupljen od strane neke musterije. Veza sa strane entiteta *musterija* je 1,N sto znaci da musterija moze uzeti bar 1 proizvod pri ulasku u apoteku, a moze uzeti i vise njih . Naziv ove veze je ***kupuje****.*



Slika 2.2.4: ER dijagram veze *kupuje* entiteta *proizvod* i *musterija*

**2.3 Kompletan ER dijagram**



Slika 2.3.1: ER dijagram komplentnog sistema

**3. Logička šema relacione baze podataka**

Prethodno smo dizajnirali ER model. Sada je potrebno da sve entitete i veze iz ER modela prevedemo u relacioni model i tako dobijamo logičku šemu relacione baze podataka.

**3.1 Entiteti**

Svaki enitet u našoj šemi postaje nezavisna šema relacije. Ime entiteta posta ime šeme relacije. Identifikator entiteta postaje *primarni ključ* šeme relacije. Obeležja tipa objekta su obeležja šeme relacije.

Dobijamo sledeće:

* Apoteka(**id\_apoteke**, kontakt, grad, ulica, broj)
* Dobavljac(**id\_dobavljaca** ,kontakt\_d, grad\_d, ulica\_d,broj\_d)
* Proizvod(**id\_proizvoda**, cena)
* Zaposleni(**jmbg\_z**, ime, prezime, datum\_zaposlenja, apoteka\_id\_apoteke)
* Musterija(**jmbg\_m**, kontakt\_m, ime\_m, prezime\_m)

Ukoliko imamo entitete sa podentitetima, podentitet postaje šema relacije. Ime podentiteta postaje ime šeme relacije, a identifikator entiteta predstavlja primarni ključ podentiteta.

* Medicinska\_oprema(vrsta, **proizvod\_id\_proizvoda**)
* Ostalo(namena, **proizvod\_id\_proizvoda**)

**3.2 Veze**

U zavisnosti od kardinalnosti, na veze se primenjuju različita pravila prevođenja.

Relacija *Projektuje* nastaje od veze *projektuje* između entiteta *sala* i *film*. Kardinalnost ove veze je (0,M):(0,N), što znači da u ovom slučaju veza postaje posebna šema relacije. Ključ ove relacije je složeni ključ koji se sastoji od identifikatora entiteta *sala* i *film.*

* Projektuje(id\_sale, id\_filma, dan, vreme)

Relacija *Prodaje* nastaje od veze *prodaje* između entiteta *šalterski\_radnik* i *karta*. Kardinalnost ove veze je (1,M):(0,1), što znači da u ovom slučaju veza postaje posebna šema relacije. Obeležja ove šeme relacije su identifikatori objekata koji su u vezi, a ključ šeme relacije je identifiaktor objekta za koji je gornja granica 1, tj. identifikator entiteta karta.

* Prodaje(id\_karte, id\_radnika)

Relacija *Kupuje* nastaje od veze *kupuje* entiteta *karta* i *korisnik*. Njena kardinalost je (0,1):(0,M). Relacija se dobija na isti način kao prethodna.

* Kupuje(id\_karte, jmbg)

**3.3 Međurelaciona ograničenja**

Pošto su dve relacije normalizovane, mogu se definisati sledeća međurelaciona ograničenja.

Sala[id\_bioskopa]⊆Bioskop[id\_bioskopa]

Radnik[id\_bioskopa]⊆Bioskop[id\_bioskopa]

Obezbeđenje[id\_radnika]⊆Radnik[id\_radnika]

Šalterski\_radnik[id\_radnika]⊆Radnik[id\_radnika]

Sedište[id\_sale] ⊆ Sala[id\_sale]

Karta[id\_sale, br\_sedišta] ⊆ Sedište[id\_sale, br\_sedišta]

Karta[id\_filma] ⊆ Film[id\_filma]

Projektuje[id\_sale] ⊆ Sala[id\_sale]

Projektuje[id\_filma] ⊆ Film[id\_filma]

Prodaje[id\_karte] ⊆ Karta[id\_karte]

Prodaje[id\_radnika] ⊆ Šalterski\_radnik[id\_radnika]

Kupuje[id\_karte] ⊆ Karta[id\_karte]

Kupuje[jmbg] ⊆ Korisnik[jmbg]

**3.4 Kompletna logička šema**

Na osnovu dosadašnjeg izlaganja dolazi se do kompletnog relacionog modela sistema:S={

Bioskop(id\_bioskopa, naziv, br\_sala, tip, ulica, broj, grad),

Sala(id\_sale, kapacitet, tip, **id\_bioskopa**),

Radnik(id\_radnika, datum\_zaposlenja, ime, prezime, plata, **id\_bioskopa**),

Film(id\_filma, trajanje, godina, ocena, uzrast, naziv, režiser, žanr),

Karta(id\_karte, datum, cena, **broj\_sedišta**, **id\_sale, id\_filma**),

Korisnik(jmbg, ime, prezime),

Sedište(broj\_sedišta, **id\_sale**, tip, pozicija),

Obezbeđenje(id\_radnika, br\_dozvole),

Šalterski\_radnik(id\_radnika),

Projektuje(id\_sale, id\_filma, dan, vreme),

Prodaje(id\_karte, id\_radnika),

Kupuje(id\_karte, jmbg)

}

I={

Sala[id\_bioskopa]⊆Bioskop[id\_bioskopa],

Radnik[id\_bioskopa]⊆Bioskop[id\_bioskopa],

Obezbeđenje[id\_radnika]⊆Radnik[id\_radnika],

Šalterski\_radnik[id\_radnika]⊆Radnik[id\_radnika],

Sedište[id\_sale] ⊆ Sala[id\_sale],

Karta[id\_sale, br\_sedišta] ⊆ Sedište[id\_sale, br\_sedišta],

Karta[id\_filma] ⊆ Film[id\_filma],

Projektuje[id\_sale] ⊆ Sala[id\_sale],

Projektuje[id\_filma] ⊆ Film[id\_filma],

Prodaje[id\_karte] ⊆ Karta[id\_karte],

Prodaje[id\_radnika] ⊆ Šalterski\_radnik[id\_radnika],

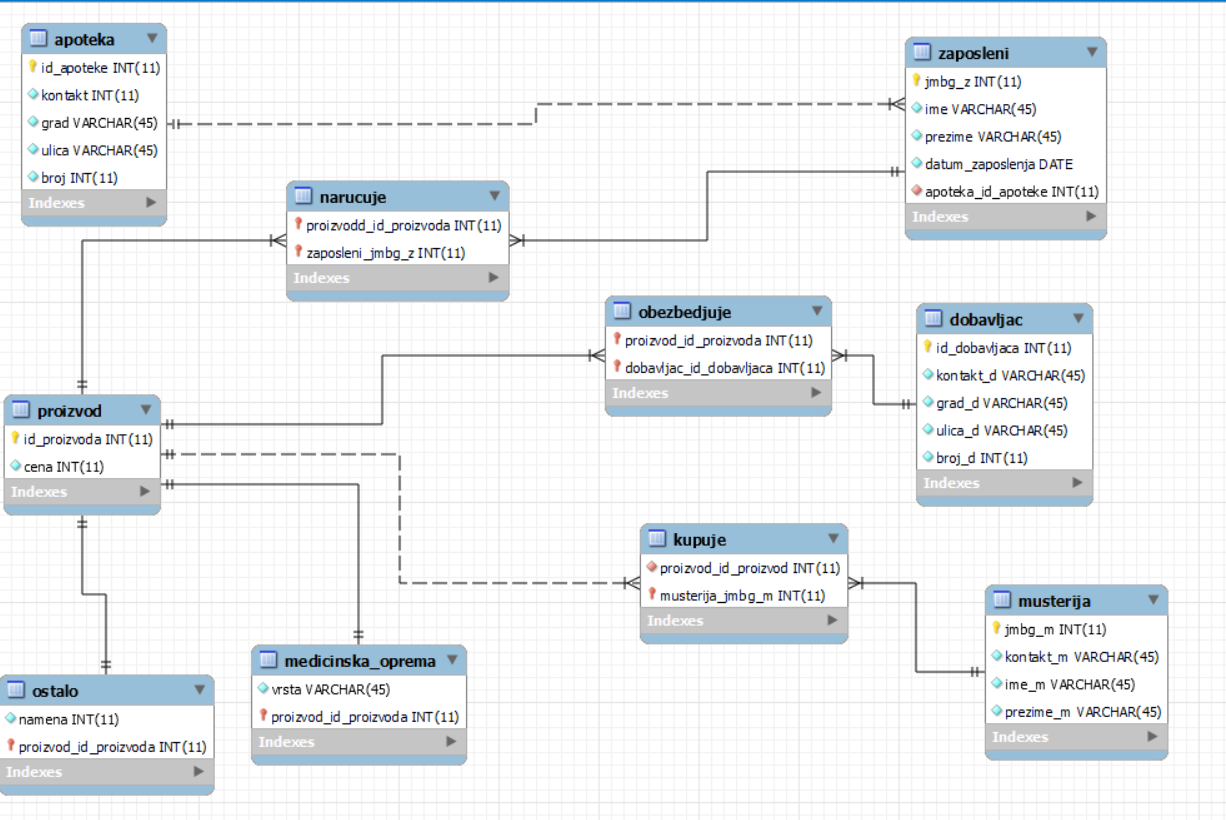
Kupuje[id\_karte] ⊆ Karta[id\_karte],

Kupuje[jmbg] ⊆ Korisnik[jmbg]

}

**4.Fizička šema relacione baze podataka**

Fizička šema relacione baze podataka opisuje realan sistem i predstavlja najniži nivo apstrakcije podataka u bazi. Identična je logičkoj šemi. Realizovana je okviru alata “MySQL Workbench”.



Slika 4.1.1: Fizička šema relacione baze podataka

**5. Implementacija projektovane baze podataka sa testnim podacima u MySQL**

SQL kod je generisan od strane MySQL Workbench-a, dok su testni podaci ručno uneseni.

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema apoteka\_nelek

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema apoteka\_nelek

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `apoteka\_nelek` DEFAULT CHARACTER SET utf8mb4 ;

USE `apoteka\_nelek` ;

-- -----------------------------------------------------

-- Table `apoteka\_nelek`.`apoteka`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `apoteka\_nelek`.`apoteka` (

`id\_apoteke` INT(11) NOT NULL,

`kontakt` INT(11) NOT NULL,

`grad` VARCHAR(45) NOT NULL,

`ulica` VARCHAR(45) NOT NULL,

`broj` INT(11) NOT NULL,

PRIMARY KEY (`id\_apoteke`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `apoteka\_nelek`.`dobavljac`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `apoteka\_nelek`.`dobavljac` (

`id\_dobavljaca` INT(11) NOT NULL,

`kontakt\_d` VARCHAR(45) NOT NULL,

`grad\_d` VARCHAR(45) NOT NULL,

`ulica\_d` VARCHAR(45) NOT NULL,

`broj\_d` INT(11) NOT NULL,

PRIMARY KEY (`id\_dobavljaca`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `apoteka\_nelek`.`proizvod`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `apoteka\_nelek`.`proizvod` (

`id\_proizvoda` INT(11) NOT NULL,

`cena` INT(11) NOT NULL,

PRIMARY KEY (`id\_proizvoda`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `apoteka\_nelek`.`medicinska\_oprema`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `apoteka\_nelek`.`medicinska\_oprema` (

`vrsta` VARCHAR(45) NOT NULL,

`proizvod\_id\_proizvoda` INT(11) NOT NULL,

PRIMARY KEY (`proizvod\_id\_proizvoda`),

CONSTRAINT `fk\_medicinska\_oprema\_proizvod1`

FOREIGN KEY (`proizvod\_id\_proizvoda`)

REFERENCES `apoteka\_nelek`.`proizvod` (`id\_proizvoda`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `apoteka\_nelek`.`musterija`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `apoteka\_nelek`.`musterija` (

`jmbg\_m` INT(11) NOT NULL,

`kontakt\_m` VARCHAR(45) NOT NULL,

`ime\_m` VARCHAR(45) NOT NULL,

`prezime\_m` VARCHAR(45) NOT NULL,

PRIMARY KEY (`jmbg\_m`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `apoteka\_nelek`.`ostalo`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `apoteka\_nelek`.`ostalo` (

`namena` INT(11) NOT NULL,

`proizvod\_id\_proizvoda` INT(11) NOT NULL,

PRIMARY KEY (`proizvod\_id\_proizvoda`),

CONSTRAINT `fk\_ostalo\_proizvod1`

FOREIGN KEY (`proizvod\_id\_proizvoda`)

REFERENCES `apoteka\_nelek`.`proizvod` (`id\_proizvoda`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `apoteka\_nelek`.`zaposleni`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `apoteka\_nelek`.`zaposleni` (

`jmbg\_z` INT(11) NOT NULL,

`ime` VARCHAR(45) NOT NULL,

`prezime` VARCHAR(45) NOT NULL,

`datum\_zaposlenja` DATE NOT NULL,

`apoteka\_id\_apoteke` INT(11) NOT NULL,

PRIMARY KEY (`jmbg\_z`),

INDEX `fk\_zaposleni\_apoteka\_idx` (`apoteka\_id\_apoteke` ASC) VISIBLE,

CONSTRAINT `fk\_zaposleni\_apoteka`

FOREIGN KEY (`apoteka\_id\_apoteke`)

REFERENCES `apoteka\_nelek`.`apoteka` (`id\_apoteke`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `apoteka\_nelek`.`narucuje`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `apoteka\_nelek`.`narucuje` (

`proizvodd\_id\_proizvoda` INT(11) NOT NULL,

`zaposleni\_jmbg\_z` INT(11) NOT NULL,

PRIMARY KEY (`proizvodd\_id\_proizvoda`, `zaposleni\_jmbg\_z`),

INDEX `fk\_proizvod\_has\_zaposleni\_zaposleni1\_idx` (`zaposleni\_jmbg\_z` ASC) VISIBLE,

INDEX `fk\_proizvod\_has\_zaposleni\_proizvod1\_idx` (`proizvodd\_id\_proizvoda` ASC) VISIBLE,

CONSTRAINT `fk\_proizvod\_has\_zaposleni\_proizvod1`

FOREIGN KEY (`proizvodd\_id\_proizvoda`)

REFERENCES `apoteka\_nelek`.`proizvod` (`id\_proizvoda`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_proizvod\_has\_zaposleni\_zaposleni1`

FOREIGN KEY (`zaposleni\_jmbg\_z`)

REFERENCES `apoteka\_nelek`.`zaposleni` (`jmbg\_z`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `apoteka\_nelek`.`obezbedjuje`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `apoteka\_nelek`.`obezbedjuje` (

`proizvod\_id\_proizvoda` INT(11) NOT NULL,

`dobavljac\_id\_dobavljaca` INT(11) NOT NULL,

PRIMARY KEY (`proizvod\_id\_proizvoda`, `dobavljac\_id\_dobavljaca`),

INDEX `fk\_proizvod\_has\_dobavljac\_dobavljac1\_idx` (`dobavljac\_id\_dobavljaca` ASC) VISIBLE,

INDEX `fk\_proizvod\_has\_dobavljac\_proizvod1\_idx` (`proizvod\_id\_proizvoda` ASC) VISIBLE,

CONSTRAINT `fk\_proizvod\_has\_dobavljac\_proizvod1`

FOREIGN KEY (`proizvod\_id\_proizvoda`)

REFERENCES `apoteka\_nelek`.`proizvod` (`id\_proizvoda`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_proizvod\_has\_dobavljac\_dobavljac1`

FOREIGN KEY (`dobavljac\_id\_dobavljaca`)

REFERENCES `apoteka\_nelek`.`dobavljac` (`id\_dobavljaca`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4;

-- -----------------------------------------------------

-- Table `apoteka\_nelek`.`kupuje`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `apoteka\_nelek`.`kupuje` (

`proizvod\_id\_proizvod` INT(11) NOT NULL,

`musterija\_jmbg\_m` INT(11) NOT NULL,

PRIMARY KEY (`musterija\_jmbg\_m`),

INDEX `fk\_proizvod\_has\_musterija\_musterija1\_idx` (`musterija\_jmbg\_m` ASC) VISIBLE,

INDEX `fk\_proizvod\_has\_musterija\_proizvod1\_idx` (`proizvod\_id\_proizvod` ASC) VISIBLE,

CONSTRAINT `fk\_proizvod\_has\_musterija\_proizvod1`

FOREIGN KEY (`proizvod\_id\_proizvod`)

REFERENCES `apoteka\_nelek`.`proizvod` (`id\_proizvoda`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_proizvod\_has\_musterija\_musterija1`

FOREIGN KEY (`musterija\_jmbg\_m`)

REFERENCES `apoteka\_nelek`.`musterija` (`jmbg\_m`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

**Upiti za pretraživanje baze podataka:**

1. Prikazati podatke o radnicima koji rade u apoteci u Njegosevoj ulici:

SELECT \* FROM Zaposleni INNER JOIN Apoteka  
ON Zaposleni.apoteka\_id\_apoteke = Apoteka.id\_apoteke  
WHERE ulica= “Njegoseva”;

1. Koliko proizvoda je porucio zaposleni a da im je cena manja od 1000 din?

SELECT id\_proizvoda, count(\*) FROM Proizvod

WHERE Proizvod.id\_proizvoda=Zaposleni.jmbg\_z and Proizvod.cena<=1000

GROUP BY id\_proizvoda;

1. Prikazati podatke o proizvodima ciji dobavljac je VEGA id=123 :

SELECT id\_proizvoda, cena FROM Proizvod, Dobavljac

WHERE Obezbedjuje.proizvod\_id\_proizvoda = Proizvod.id\_proizvoda

AND Obezbedjuje.dobavljac\_id\_dobavljaca=Dobavljac.id\_dobavljaca

AND Dobavljac.id\_dobavljaca=123;

1. Prikaz svih radnika koji su se zaposlili pre 2000. godine :

SELECT \*

FROM Radnik

WHERE EXTRACT (year FROM datum\_zaposlenja) < 2000 ;

**6. Literatura**

[1] Materijali sa moodle portala Fakulteta inženjerskih nauka: Baze podataka

[2] Lazarevic B. : Baze podataka, FON Beograd