Univerzitet u Kragujevcu

Fakultet inženjerskih nauka



**Baze Podataka**

Projektni zadatak:

Biblioteka

Student: Predmetni nastavnik:

Kristina Trajković 605/2019 Milan Erić

**Sadržaj:**

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

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

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

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

2.3 Kompletan ER dijagram . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

**3 Logicka šema relacione baze podataka . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13**

3.1 Entiteti . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .13

3.2 Veze . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .13

3.3 Međurelaciona ograničenja . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

3.4 Kompletna logička šema . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

**4 Fizička šema . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .16**

**5 Implementacija projektovane baze sa testnim podacima . . . . . . . . . . . 16**

**6 Literatura . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .26**

**1. Uvod**

Biblioteka je sređena zbirka knjiga po autorskom, predmetnom ili katalogu, koja pripada privatnim licima ili državnim institucijama, ili ustanovama u kojima se čuvaju zbirke knjiga.

U ovom projektu je izvršena analiza domena poslovanja i funkcionisanja biblioteke. Kreiran je poslovni model za ovaj domen i izvršena implementacija kroz kreiranje šeme relacione baze podataka.

Bitne tačke ovog sistema su:

-Knjiga

Za funkcionisanje biblioteke neophodna je evidencija knjiga radi praćenja izdavanja i stanja kako izdatih primeraka tako i onih raspoloživih za izdavanje. Podaci o knjizi su njen naziv, žanr, godina izdanja, izdavačka kuća.

-Korisnik

Korisnici su osnovni smisao funkcionisanja biblioteke i njen ključni faktor postojanja.Evidencija o korisnicima obuhvata ime i prezime člana, jedinstveni matični broj građanina,telefon.

-Bibliotekar

Oni predstavljaju elementarni faktor funkcionisanja biblioteke.Bibliotekar je osoba nadležna za izdavanje knjiga ali i njihovo primanje nazad u biblioteku.

**2. ER dijagram**

Modelovanje sistema započinjemo konstruisanjem ER dijagrama. Zatim identifikujemo entitete, atribute i sve veze između entiteta. ER dijagram je kontruisan u programu Edraw Max.

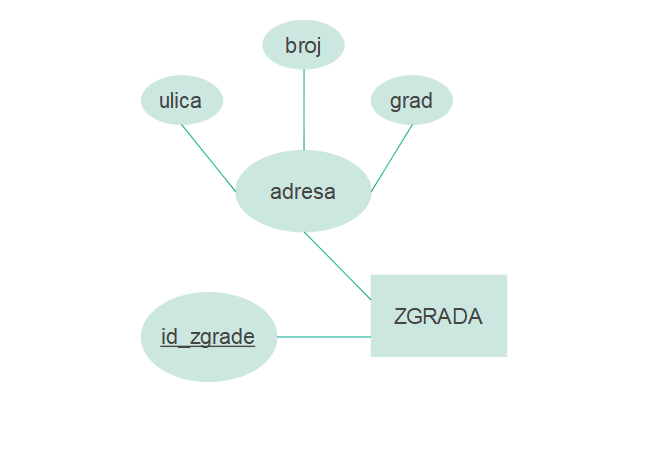
U daljem tekstu su opisani i prikazani svi entiteti i atributi, kao i veze između entiteta.

**2.1 Entiteti**

Identifikovali smo sledeće entitete:

**Zgrada:**

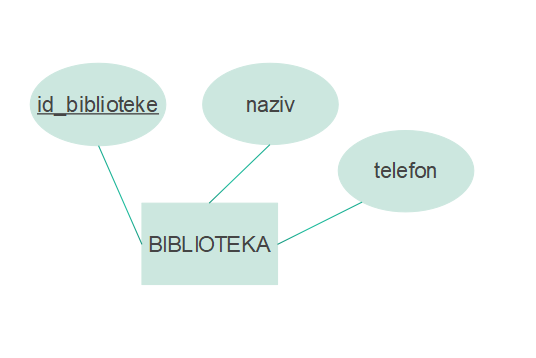
Entitet zgrada predstavlja jednu apoteku koja sadrži identifikator id\_zgrade, atribut adresa(koju čine podaci ulica, broj i grad).



Slika 1: ER dijagram entiteta *zgrada*

**Biblioteka:**

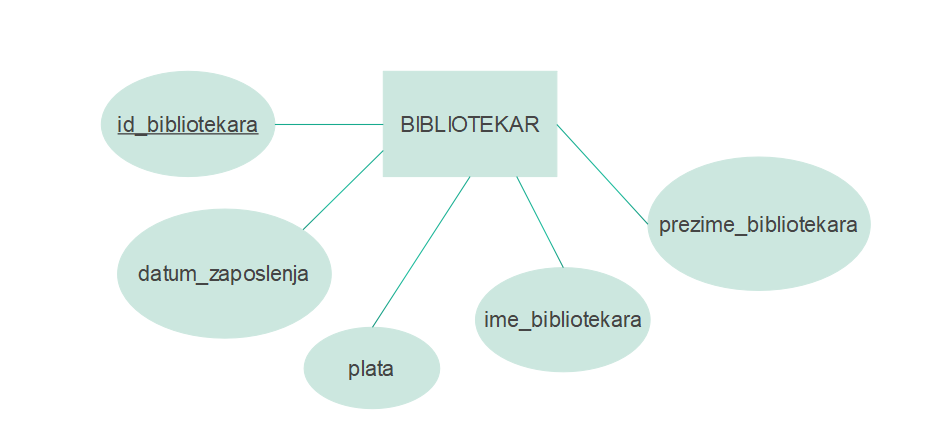
Entitet biblioteka ima svoj identifikator id\_biblioteke i atribute naziv i telefon.



Slika 2: ER dijagram entiteta *biblioteka*

**Bibliotekar:**

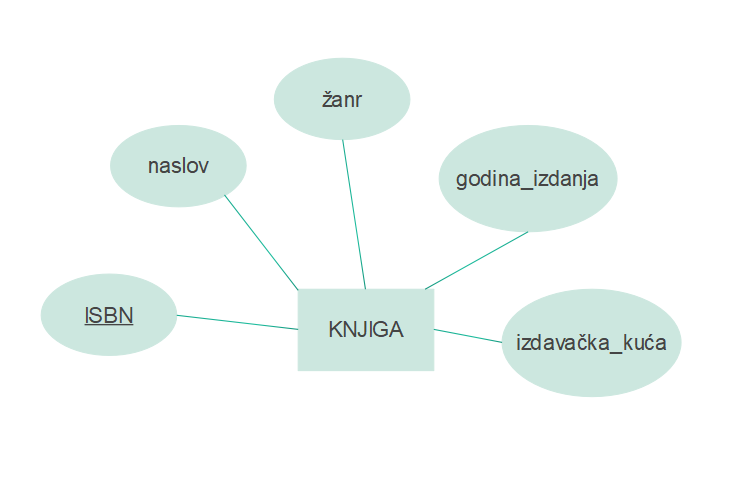
Entitet bibliotekar ima svoj identifikator id\_bibliotekara i atribute datum\_zaposlenja, plata, ime\_bibliotekara, prezime\_bibliotekara



Slika 3: ER dijagram entiteta *bibliotekar*

**Knjiga:**

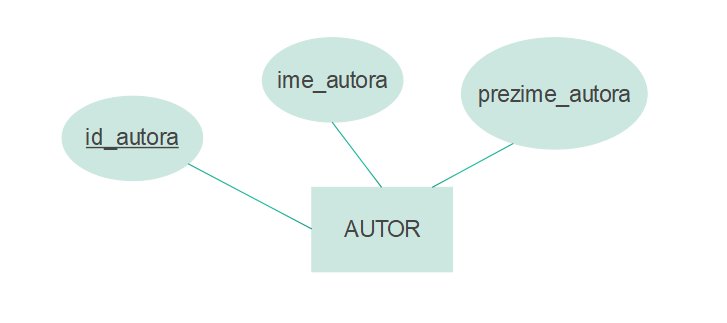
Entitet knjiga ima svoj identifikator ISBN i atribute naslov, žanr, godina\_izdanja, izdavačka\_kuća.



Slika 4: ER dijagram entiteta *knjiga*

**Autor:**

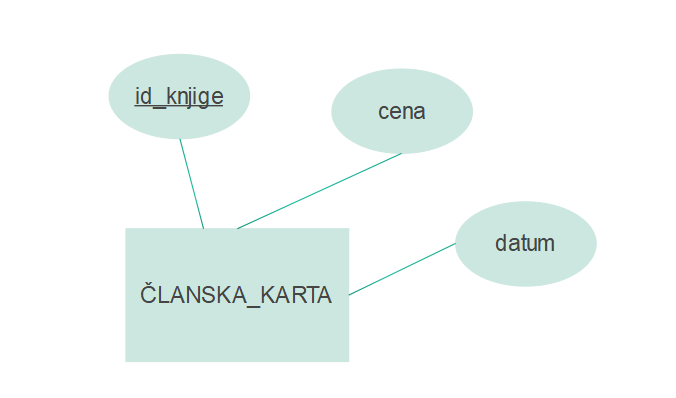
Entitet autor ima svoj identifikator id\_autora i atribute ime\_autora, prezime\_autora.



Slika 5: ER dijagram entiteta *autor*

**Članska karta:**

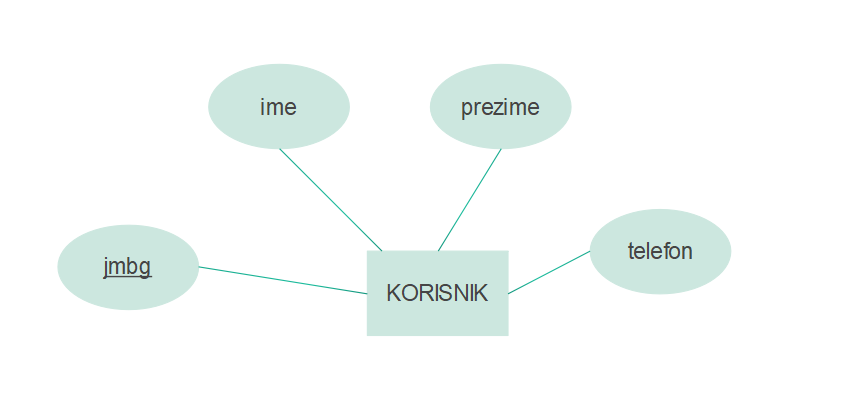
Entitet članska karta ima svoj identifikator id\_knjige i atribute cena, datum.



Slika 6: ER dijagram entiteta *člasnka karta*

**Korisnik:**

Entitet korisnik ima svoj identifikator jmbg i atributi ime, prezime, telefon.



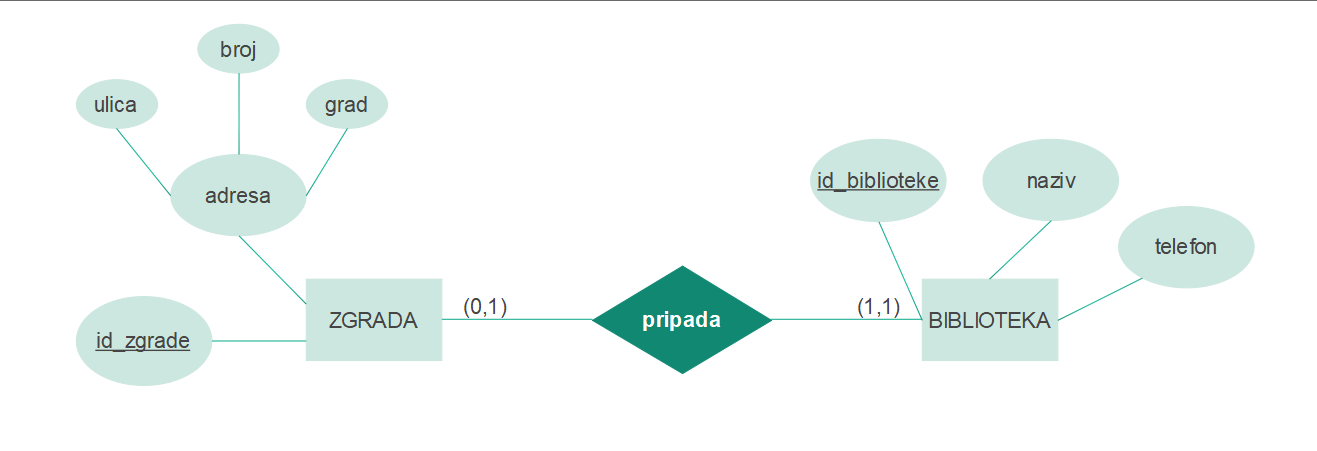
Slika 7: ER dijagram entiteta *korisnik*

**2.2 Veze**

Identifikovali smo sledeće veze:

**Veza zgrada-biblioteka:**

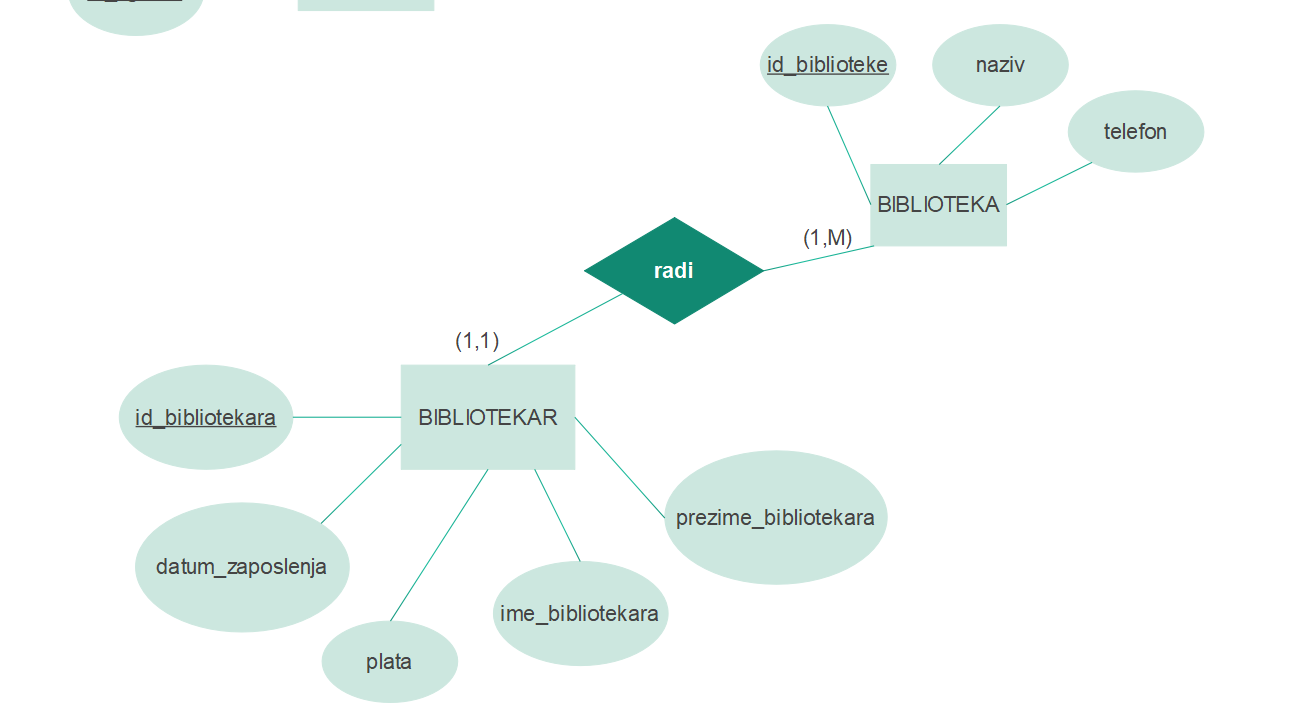
Veza sa strane entiteta zgrada je 0,1, to znači da u zgradi postiji samo jedna bibliotek, a ne mora nijedna. Veza sa strane entiteta biblioteka je 1,1, što znači da bilioteka pripada tačno jednoj zgradi. Naziv ove veze je pripada.



Slika 8: ER dijagram veze *pripada* entiteta *zgrada i biblioteka*

**Veza biblioteka-bibliotekar:**

Veza sa strane entiteta biblioteka je 1,M, što znači da u biblioteci mora bar jedan bibliotekar da radi, a može I više njih. Veza sa strane entiteta bibliotekar je 1,1, što znači da bibliotekar radi u jednoj I samo jednoj biblioteci.

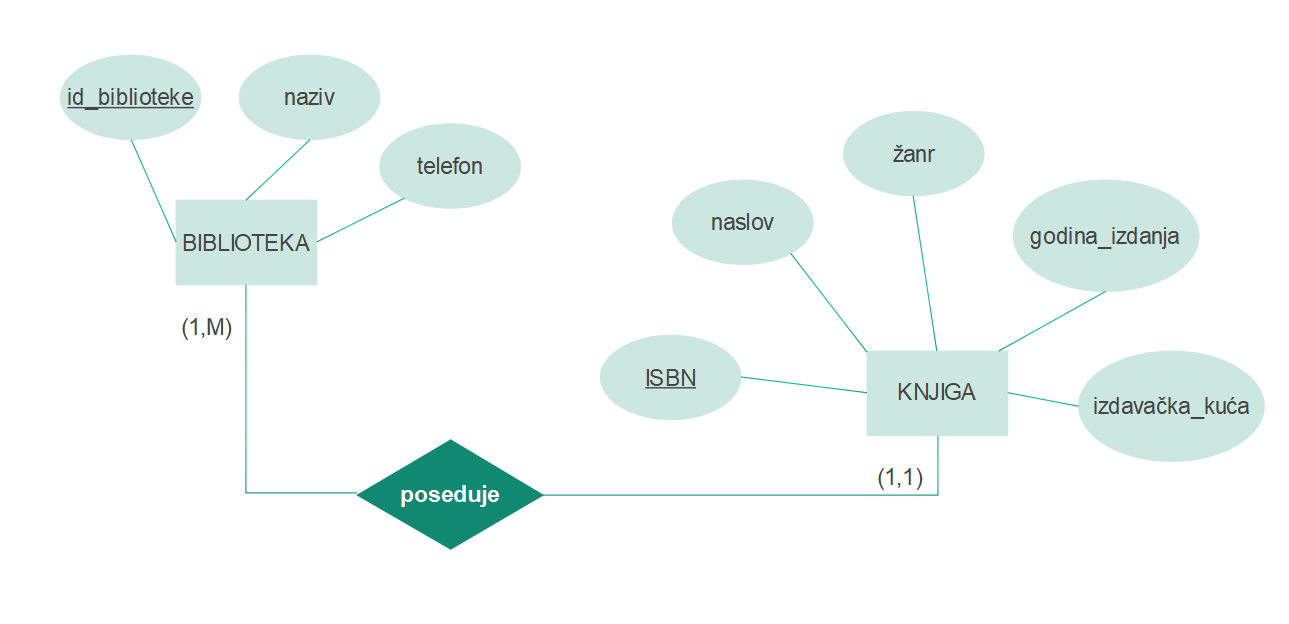


Slika 9: ER dijagram veze radientiteta bibliotekar i biblioteka

**Veza biblioteka-knjiga:**

1,M- jedna biblioteka mora da ima bar jednu knjigu, a može da ima I više knjiga.

1,1- knjiga sa svojim ISBN se nalazi u jednoj I samo jednoj biblioteci.

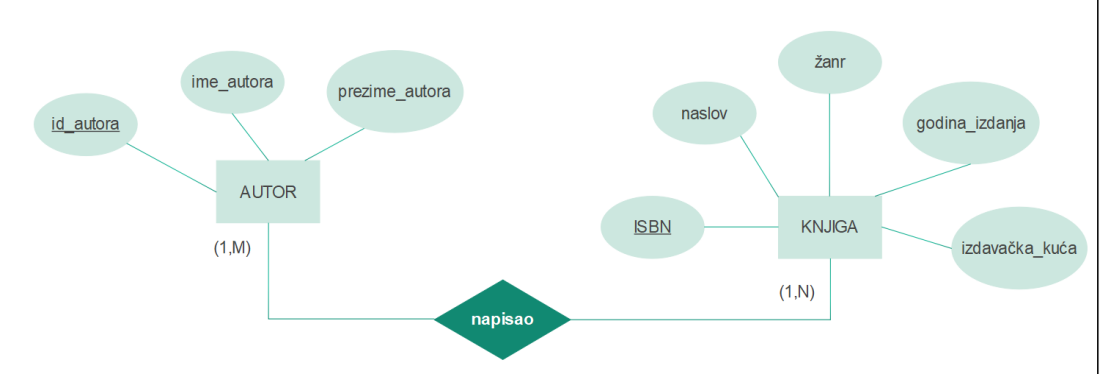


Slika 10: ER dijagram veze *poseduje* entiteta biblioteka i knjiga

**Veza autor-knjiga:**

1,M- jednu knjigu moze da napise jedan autor, a moze da napise I vise autora.

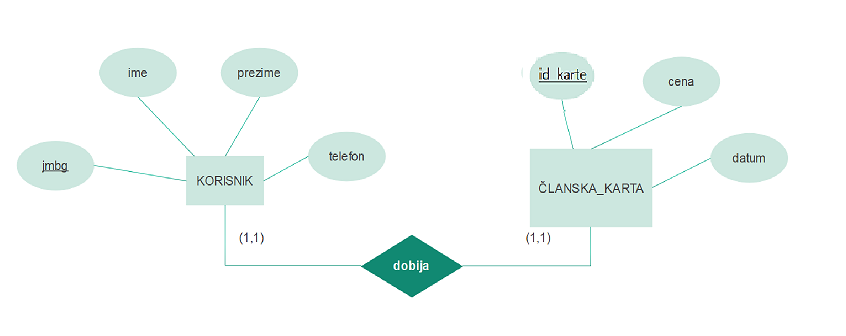
1,N-autor moze da napise jednu, a moze I vise knjiga.



Slika 11: ER dijagram veze *napisao* entiteta autor i knjiga

**Veza korisnik-članska karta:**

1,1- Članska karta ima jednog I samo jednog korisnika, I korisnik takodje ima jednu I samo jednu člasnku kartu

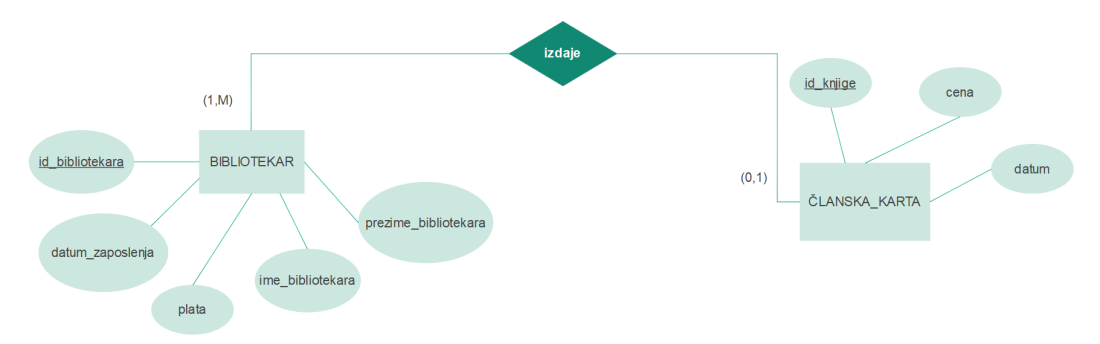


Slika 12: ER dijagram veze *dobija* entiteta korisnik i clanska karta

**Veza bibliotekar-članska karta:**

1,M- bibliotekar izdaje jednu ili više članskih karata.

0,1- članska karta može, a I ne mora da bude izdata(prazna članska karta).

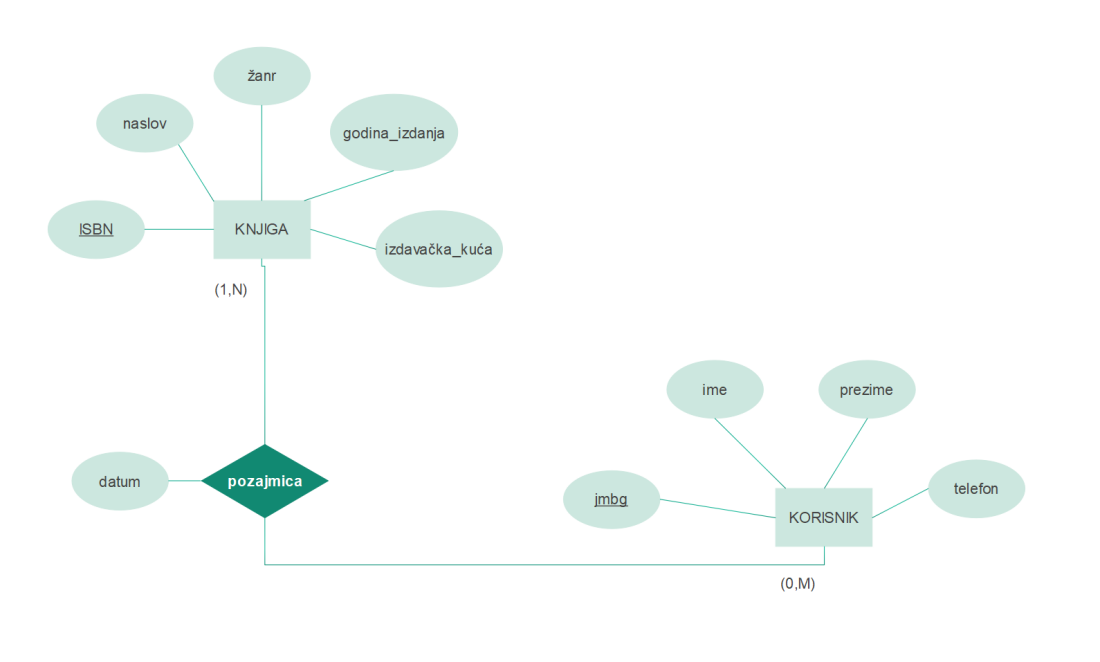


Slika 13: ER dijagram veze izdajeentiteta bibliotekar i clanka karta

**Veza knjiga-korisnik:**

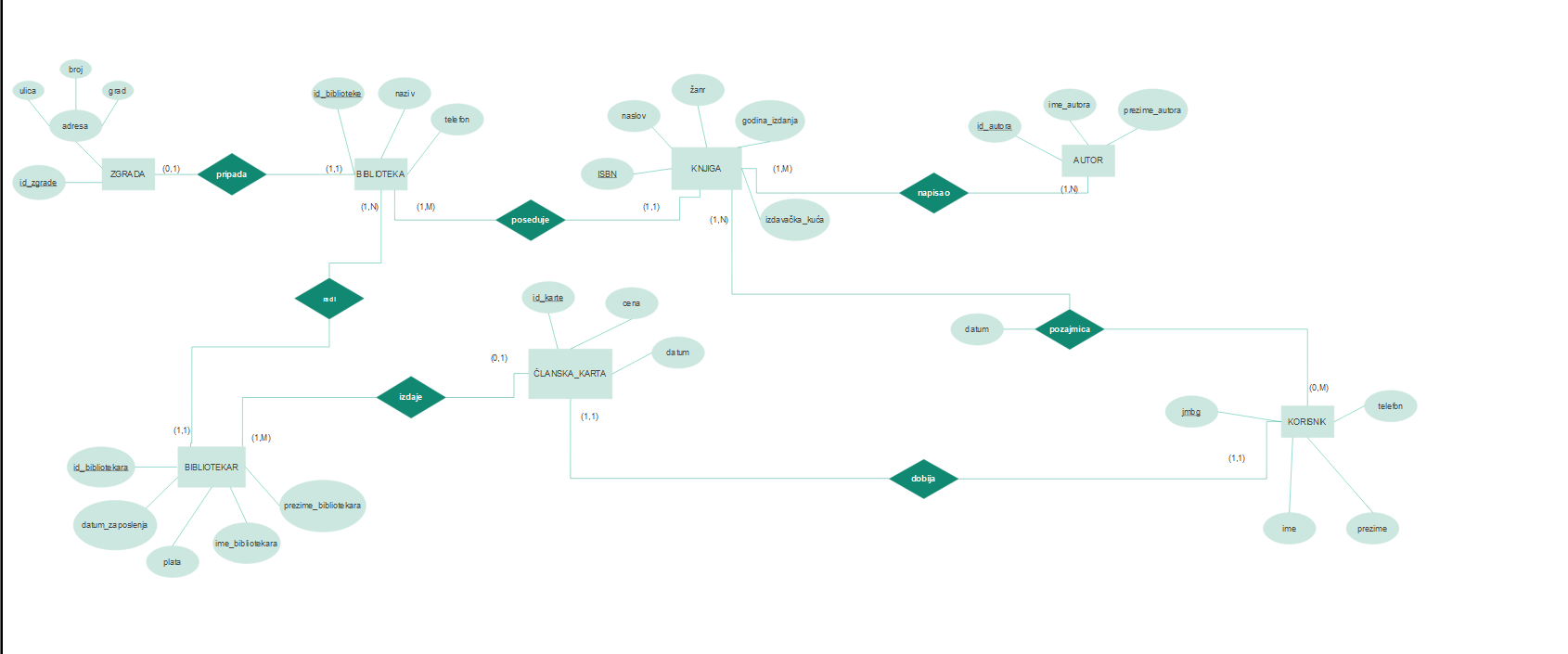
1-N- kjniga je bar jednom pozajmljena od strane nekog korisnika, a može I više puta.

0,M- korisnik ne mora da pozajmi nijednu knjigu, a može da pojazajmi I više knjiga u jednom trenutku.



Slika 14: ER dijagram veze *pozajmica* entiteta knjiga i korisnik

**2.3 Kompletan ER dijagram**



Slika 15: ER dijagram komplentnog sistema

**3. Logička šema relacione baze podataka**

Potrebno je da prvo sve entitete i veze iz ER modela prevedemo u relacioni model. Formiraćemo relacije i međurelaciona ograničenja.

**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.

Jaki entiteti postoje nezavisno od nekog drugog entiteta, odnosno samostalno. Kod složenog atributa (npr *adresa*), dolazi do pretvaranja u 1NF. Svaki atribut je atomičan.

Dobijamo sledeće:

Zgrada(id\_zgrade,ulica, broj, grad)

Biblioteka(id\_biblioteke, naziv, telefon,**id\_zgrade**)

Bibliotekar(id\_bibliotekara,datum\_zaposlenja, plata, ime\_bibliotekara, prezime\_bibliotekara, **id\_biblioteke**)

Knjiga(ISBN, naslov, zanr, godina\_izdanja, izdavacka\_kuca,**id\_biblioteke**)

Autor(id\_autora, ime\_autora, prezime\_autora)

Korisnik(jmbg, ime, prezime, telefon)

Clanska karta(id\_karte, cena, datum, **jmbg**)

**3.2 Veze**

Relacija Izdajenastaje od veze izdajeizmeđu entiteta *bibliotekar* i *clanska\_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.

Izdaje(id\_karte, id\_bibliotekara)

Relacija Napisaonastaje od veze napisaoizmeđu entiteta *knjiga* i *autor*. Kardinalnost ove veze je (1,M):(1,N), što znači da u ovom slučaju veza postaje posebna šema relacije. Ključ ove relacije je složeni ključ koji se sastoji od identifikatora entiteta knjiga i autor*.*

Napisao(ISBN, id\_autora)

Relacija Pozajmicanastaje od veze pozajmicaentiteta *knjiga* i *korisnik*. Njena kardinalost je (1,N):(0,M). Relacija se dobija na isti način kao prethodna.

Pozajmica(knjiga\_ISBN, korisnik\_jmbg, datum)

**3.3 Međurelaciona ograničenja**

Biblioteka[id\_zgrade] ⊆ Zgrada[id\_zgrade]

Bibliotekar[id\_biblioteke] ⊆ Biblioteka[id\_biblioteke]

Knjiga[id\_biblioteke] ⊆ Biblioteka[id\_biblioteke]

Clanska karta[jmbg] ⊆ Korisnik[jmbg]

Napisao[ISBN] ⊆ Knjiga[ISBN]

Napisao[id\_autora] ⊆ Autor[id\_autora]

Izdaje[id\_karte] ⊆ Clanska karta[id\_karte]

Izdaje[id\_bibliotekara] ⊆ Bibliotekar[id\_bibliotekara]

Pozajmica[ISBN] ⊆ Knjiga[ISBN]

Pozajmica[jmbg] ⊆ Korisnik[jmbg]

**3.4 Kompletna logička šema**

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

Zgrada(id\_zgrade,ulica, broj, grad)

Biblioteka(id\_biblioteke, naziv, telefon,**zgrada\_id\_zgrade**)

Bibliotekar(id\_bibliotekara,datum\_zaposlenja, plata, ime\_bibliotekara, prezime\_bibliotekara, **bibliotek\_id\_biblioteke**)

Knjiga(ISBN, naslov, zanr, godina\_izdanja, izdavacka\_kuca, **bibliotek\_id\_biblioteke**)

Autor(id\_autora, ime\_autora, prezime\_autora)

Korisnik(jmbg, ime, prezime, telefon)

Clanska karta(id\_karte, cena, datum, **korisnik\_jmbg**)

Napisao(knjiga\_ISBN, autor\_id\_autora)

Izdaje(clanska\_karta\_id\_karte, bibliotekar\_id\_bibliotekara)

Pozajmica(knjiga\_ISBN, korisnik\_jmbg, datum)

}

I={

Biblioteka[id\_zgrade] ⊆ Zgrada[id\_zgrade]

Bibliotekar[id\_biblioteke] ⊆ Biblioteka[id\_biblioteke]

Knjiga[id\_biblioteke] ⊆ Biblioteka[id\_biblioteke]

Clanska karta[jmbg] ⊆ Korisnik[jmbg]

Napisao[ISBN] ⊆ Knjiga[ISBN]

Napisao[id\_autora] ⊆ Autor[id\_autora]

Izdaje[id\_karte] ⊆ Clanska karta[id\_karte]

Izdaje[id\_bibliotekara] ⊆ Bibliotekar[id\_bibliotekara]

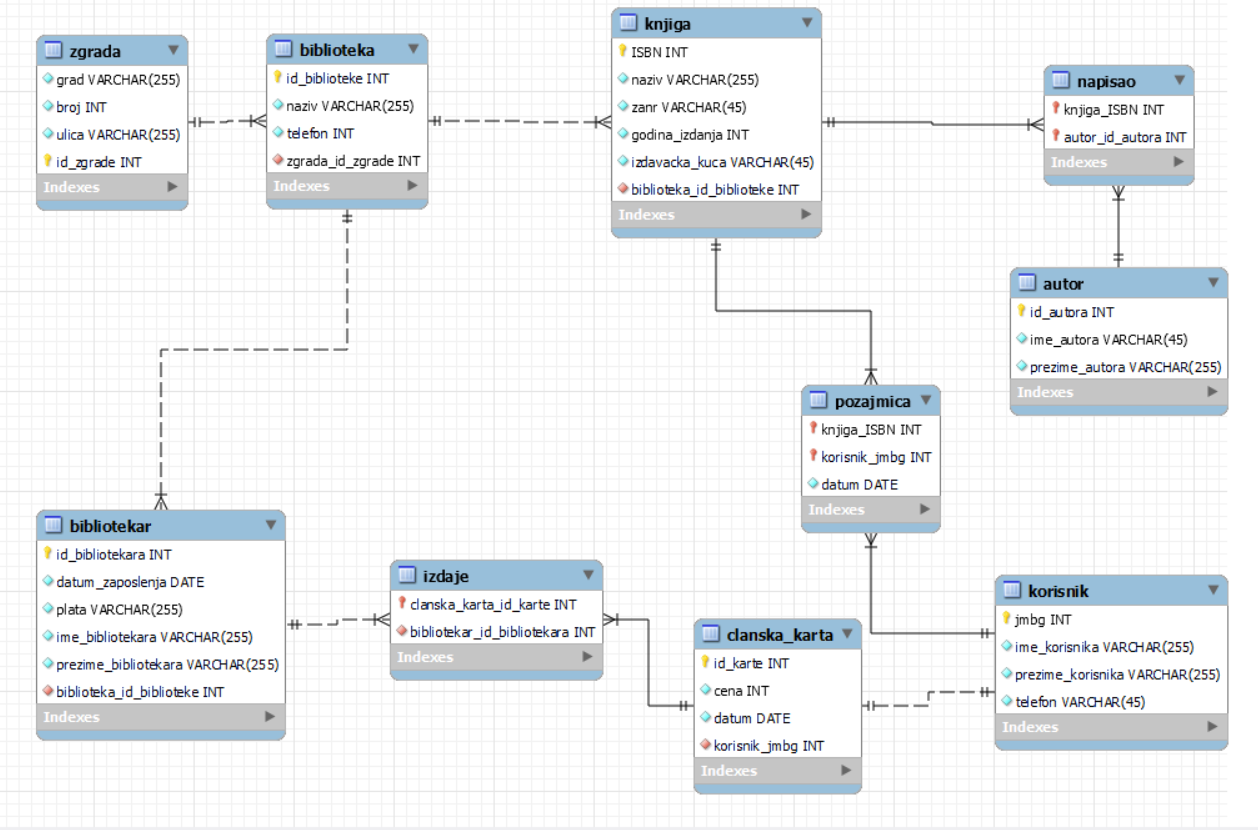
Pozajmica[ISBN] ⊆ Knjiga[ISBN]

Pozajmica[jmbg] ⊆ Korisnik[jmbg]

}

**4. Fizička šema relacione baze podataka**

Fizička šema predstavlja najniži nivo apstrakcije podataka u bazi. Ona je jedan korak iza implementacije samog sistema. Za izadu fizičke šeme korišćen je program MySQL Workbench.



Slika 16: Fizička šema realnog sistema

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

Pošto MySQL ima opciju da sam generiše SQL kod koji implementira projektovanu fizičku šemu baze podataka, to će biti prikazano na sledećim stranama, zajedno sa testnim podacima koji su ručno ubačeni.

-- MySQL Script generated by MySQL Workbench

-- Sun Jun 5 18:22:35 2022

-- Model: New Model Version: 1.0

-- 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 biblioteka

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

DROP SCHEMA IF EXISTS `biblioteka` ;

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

-- Schema biblioteka

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

CREATE SCHEMA IF NOT EXISTS `biblioteka` DEFAULT CHARACTER SET utf8 ;

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

-- Schema test

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

USE `biblioteka` ;

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

-- Table `biblioteka`.`zgrada`

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

DROP TABLE IF EXISTS `biblioteka`.`zgrada` ;

CREATE TABLE IF NOT EXISTS `biblioteka`.`zgrada` (

`grad` VARCHAR(255) NULL,

`broj` INT NULL,

`ulica` VARCHAR(255) NULL,

`id\_zgrade` INT NOT NULL,

PRIMARY KEY (`id\_zgrade`))

ENGINE = InnoDB;

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

-- Table `biblioteka`.`biblioteka`

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

DROP TABLE IF EXISTS `biblioteka`.`biblioteka` ;

CREATE TABLE IF NOT EXISTS `biblioteka`.`biblioteka` (

`id\_biblioteke` INT NOT NULL,

`naziv` VARCHAR(255) NOT NULL,

`telefon` INT NULL,

`zgrada\_id\_zgrade` INT NOT NULL,

PRIMARY KEY (`id\_biblioteke`),

INDEX `fk\_biblioteka\_zgrada\_idx` (`zgrada\_id\_zgrade` ASC) VISIBLE,

CONSTRAINT `fk\_biblioteka\_zgrada`

FOREIGN KEY (`zgrada\_id\_zgrade`)

REFERENCES `biblioteka`.`zgrada` (`id\_zgrade`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `biblioteka`.`bibliotekar`

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

DROP TABLE IF EXISTS `biblioteka`.`bibliotekar` ;

CREATE TABLE IF NOT EXISTS `biblioteka`.`bibliotekar` (

`id\_bibliotekara` INT NOT NULL,

`datum\_zaposlenja` DATE NULL,

`plata` VARCHAR(255) NULL,

`ime\_bibliotekara` VARCHAR(255) NULL,

`prezime\_bibliotekara` VARCHAR(255) NULL,

`biblioteka\_id\_biblioteke` INT NOT NULL,

PRIMARY KEY (`id\_bibliotekara`),

INDEX `fk\_bibliotekar\_biblioteka1\_idx` (`biblioteka\_id\_biblioteke` ASC) VISIBLE,

CONSTRAINT `fk\_bibliotekar\_biblioteka1`

FOREIGN KEY (`biblioteka\_id\_biblioteke`)

REFERENCES `biblioteka`.`biblioteka` (`id\_biblioteke`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `biblioteka`.`knjiga`

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

DROP TABLE IF EXISTS `biblioteka`.`knjiga` ;

CREATE TABLE IF NOT EXISTS `biblioteka`.`knjiga` (

`ISBN` INT NOT NULL,

`naziv` VARCHAR(255) NULL,

`zanr` VARCHAR(45) NULL,

`godina\_izdanja` INT NULL,

`izdavacka\_kuca` VARCHAR(45) NULL,

`biblioteka\_id\_biblioteke` INT NOT NULL,

PRIMARY KEY (`ISBN`),

INDEX `fk\_knjiga\_biblioteka1\_idx` (`biblioteka\_id\_biblioteke` ASC) VISIBLE,

CONSTRAINT `fk\_knjiga\_biblioteka1`

FOREIGN KEY (`biblioteka\_id\_biblioteke`)

REFERENCES `biblioteka`.`biblioteka` (`id\_biblioteke`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `biblioteka`.`autor`

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

DROP TABLE IF EXISTS `biblioteka`.`autor` ;

CREATE TABLE IF NOT EXISTS `biblioteka`.`autor` (

`id\_autora` INT NOT NULL,

`ime\_autora` VARCHAR(45) NULL,

`prezime\_autora` VARCHAR(255) NULL,

PRIMARY KEY (`id\_autora`))

ENGINE = InnoDB;

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

-- Table `biblioteka`.`korisnik`

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

DROP TABLE IF EXISTS `biblioteka`.`korisnik` ;

CREATE TABLE IF NOT EXISTS `biblioteka`.`korisnik` (

`jmbg` INT NOT NULL,

`ime\_korisnika` VARCHAR(255) NULL,

`prezime\_korisnika` VARCHAR(255) NULL,

`telefon` VARCHAR(45) NULL,

PRIMARY KEY (`jmbg`))

ENGINE = InnoDB;

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

-- Table `biblioteka`.`clanska\_karta`

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

DROP TABLE IF EXISTS `biblioteka`.`clanska\_karta` ;

CREATE TABLE IF NOT EXISTS `biblioteka`.`clanska\_karta` (

`id\_karte` INT NOT NULL,

`cena` INT NULL,

`datum` DATE NULL,

`korisnik\_jmbg` INT NOT NULL,

PRIMARY KEY (`id\_karte`),

INDEX `fk\_clanska\_karta\_korisnik1\_idx` (`korisnik\_jmbg` ASC) VISIBLE,

CONSTRAINT `fk\_clanska\_karta\_korisnik1`

FOREIGN KEY (`korisnik\_jmbg`)

REFERENCES `biblioteka`.`korisnik` (`jmbg`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `biblioteka`.`napisao`

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

DROP TABLE IF EXISTS `biblioteka`.`napisao` ;

CREATE TABLE IF NOT EXISTS `biblioteka`.`napisao` (

`knjiga\_ISBN` INT NOT NULL,

`autor\_id\_autora` INT NOT NULL,

PRIMARY KEY (`knjiga\_ISBN`, `autor\_id\_autora`),

INDEX `fk\_knjiga\_has\_autor\_autor1\_idx` (`autor\_id\_autora` ASC) VISIBLE,

INDEX `fk\_knjiga\_has\_autor\_knjiga1\_idx` (`knjiga\_ISBN` ASC) VISIBLE,

CONSTRAINT `fk\_knjiga\_has\_autor\_knjiga1`

FOREIGN KEY (`knjiga\_ISBN`)

REFERENCES `biblioteka`.`knjiga` (`ISBN`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_knjiga\_has\_autor\_autor1`

FOREIGN KEY (`autor\_id\_autora`)

REFERENCES `biblioteka`.`autor` (`id\_autora`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `biblioteka`.`pozajmica`

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

DROP TABLE IF EXISTS `biblioteka`.`pozajmica` ;

CREATE TABLE IF NOT EXISTS `biblioteka`.`pozajmica` (

`knjiga\_ISBN` INT NOT NULL,

`korisnik\_jmbg` INT NOT NULL,

PRIMARY KEY (`knjiga\_ISBN`, `korisnik\_jmbg`),

INDEX `fk\_knjiga\_has\_korisnik\_korisnik1\_idx` (`korisnik\_jmbg` ASC) VISIBLE,

INDEX `fk\_knjiga\_has\_korisnik\_knjiga1\_idx` (`knjiga\_ISBN` ASC) VISIBLE,

CONSTRAINT `fk\_knjiga\_has\_korisnik\_knjiga1`

FOREIGN KEY (`knjiga\_ISBN`)

REFERENCES `biblioteka`.`knjiga` (`ISBN`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_knjiga\_has\_korisnik\_korisnik1`

FOREIGN KEY (`korisnik\_jmbg`)

REFERENCES `biblioteka`.`korisnik` (`jmbg`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `biblioteka`.`izdaje`

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

DROP TABLE IF EXISTS `biblioteka`.`izdaje` ;

CREATE TABLE IF NOT EXISTS `biblioteka`.`izdaje` (

`clanska\_karta\_id\_karte` INT NOT NULL,

`bibliotekar\_id\_bibliotekara` INT NOT NULL,

INDEX `fk\_clanska\_karta\_has\_bibliotekar\_clanska\_karta1\_idx` (`clanska\_karta\_id\_karte` ASC) VISIBLE,

INDEX `fk\_izdaje\_bibliotekar1\_idx` (`bibliotekar\_id\_bibliotekara` ASC) VISIBLE,

PRIMARY KEY (`clanska\_karta\_id\_karte`),

CONSTRAINT `fk\_clanska\_karta\_has\_bibliotekar\_clanska\_karta1`

FOREIGN KEY (`clanska\_karta\_id\_karte`)

REFERENCES `biblioteka`.`clanska\_karta` (`id\_karte`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_izdaje\_bibliotekar1`

FOREIGN KEY (`bibliotekar\_id\_bibliotekara`)

REFERENCES `biblioteka`.`bibliotekar` (`id\_bibliotekara`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

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

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

**Upiti**

1. Prikazati podatke o bibliotekaru koji radi u biblioteci Branislav Nušić:

SELECT \* FROM Bibliotekar INNER JOIN Biblioteka

ON Biblioteka.ID\_biblioteke = Bibliotekar.ID\_biblioteke

WHERE naziv\_biblioteke= “Branislav Nušić”;

2.Prikaz svih bibliotekara koji su se zaposlili pre 2010. godine :

SELECT \*

FROM bibliotekar

WHERE EXTRACT (year FROM datum\_zaposlenja) < 2010 ;

3.Prikazati autore knjiga čiji žanr horor:

SELECT ime\_autora, prezime\_autora FROM Autor

WHERE ID\_autora IN(

SELECT ID\_autora FROM napisao WHERE ISBN IN(

SELECT ISBN FROM Knjiga

WHERE zanr=”horor”));

**8. Literatura**

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