

Univerzitet u Kragujevcu
Fakultet inženjerskih nauka



Baze Podataka

Projektni zadatak:

Školska biblioteka

Student:

Katarina Vuletić 656/2018

Predmetni nastavnik:

Milan Erić

Kragujevac, Jun 2021.

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	15
3 Logicka šema relacione baze podataka	15
3.1 Entiteti	15
3.2 Veze	16
3.3 Međurelaciona ograničenja	17
3.4 Kompletna logička šema	18
4 Fizička šema	18
5 Implementacija projektovane baze sa testnim podacima	20
6 Literatura	41

1. Uvod

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

Bibliotekarstvo je nauka koja se bavi potrebama biblioteka. U današnje vreme raširena je upotreba računara u bibliotekama, a knjige se klasifikuju koristeći neki od raspoloživih decimalnih sistema klasifikacije.

Biblioteka je mesto na kojem se putem knjige stiču znanja, umenja i veštine, dolazi do raznih informacija.

U ovom projektnom zadatku biće objašnjen način modelovanja baze podataka informacionog sistema koji je pogodan za skladištenje informacija koje se tipicno mogu naći u školskim bibliotekama.

1.1 Relevantni dokumenti i potrebe realnog sistema

U našoj bazi podataka odlučili smo da čuvamo informacije bitne za svaku školsku biblioteku, informacije o školi u kojoj se ta biblioteka nalazi, informacije o zaposlenima, odnosno bibliotekarima, knjigama koje izdaju svojim članovima, informacije o samim članovima, kao i o autorima koji te knjige pišu.

Bitne tačke ovog sistema su:

- škola
- biblioteka
- bibliotekar
- član
- knjiga
- autor
- tip knjige
- predmet

Veze između tačaka:

- biblioteka se nalazi u školi
- bibliotekari izdaju knjige
- članovi iznajmljuju knjige

- knjige su trenutno dostupne u biblioteci
- knjiga može predstavljati udžbenik za neki predmet, a ukoliko nije udžbenik, onda pripada nekom tipu knjige, kao što je roman, rečnik, lektira i slično
- svaki autor je napisao jednu ili veći broj knjiga

2. ER dijagram

Modeliranje baze podataka zahteva da se svi elementi realnog sveta svrstaju u neku od 3 kategorije:

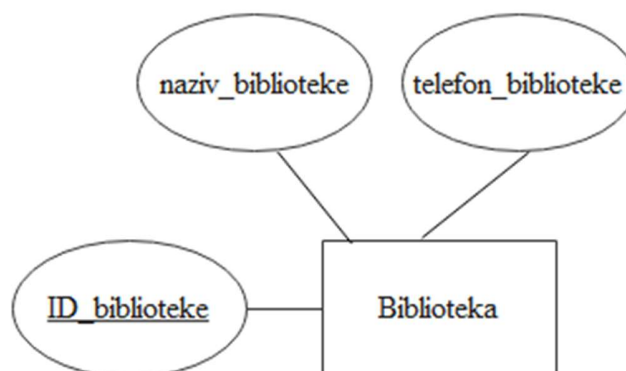
- 1) entiteti – objekti ili događaji koji su nam od interesa za BP
- 2) veze – odnosi među entitetima
- 3) atributi – svojstva entiteta i veza

U daljem tekstu su opisani i prikazani svi entiteti i atributi, kao i veze između entiteta. Za ovu realizaciju je korišćen softver "Edraw Max".

2.1 Entiteti

Biblioteka:

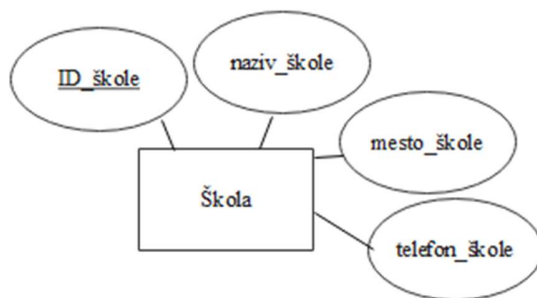
Entitet *biblioteka* predstavlja jednu biblioteku koja sadrži identifikator *ID_biblioteke*, attribute *naziv_biblioteke*, *telefon_biblioteke*.



Slika 1: ER dijagram entiteta *biblioteka*

Škola:

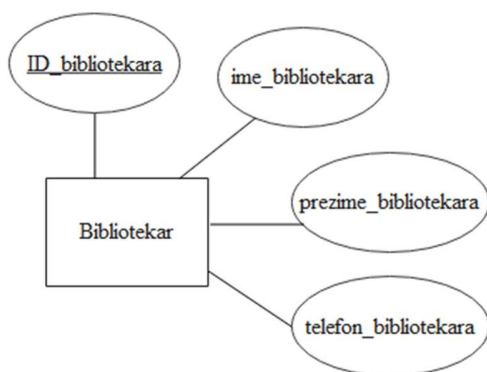
Entitet *škola* predstavlja školu u kojoj se biblioteka nalazi i koja sadrži identifikator *ID_škole*, atribute *naziv_škole*, *mesto_škole*, *telefon_škole*.



Slika 2: ER dijagram entiteta *škola*

Bibliotekar:

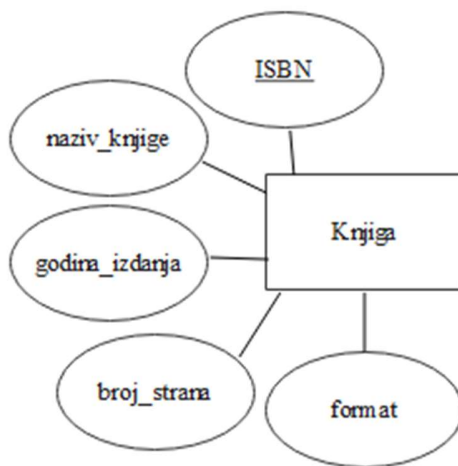
Entitet *bibliotekar* ima svoj identifikator *ID_bibliotekara* i dodatne attribute *ime_bibliotekara*, *prezime_bibliotekara* i *telefon_bibliotekara*.



Slika 3: ER dijagram entiteta *bibliotekar*

Knjiga:

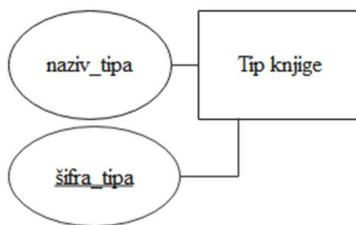
Entitet *knjiga* sadrži svoj identifikator *ISBN* i atribute *naziv_knjige*, *godina_izdanja*, *broj_strana*, *format*.



Sika 4: ER dijagram entiteta *knjiga*

Tip knjige:

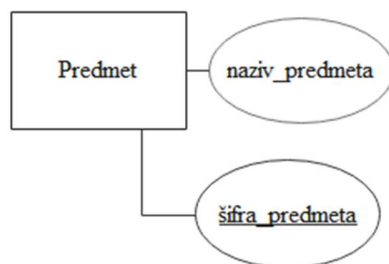
Entitet *tip knjige* sadrži ključ *šifra_tipa* i atribut *naziv_tipa*.



Sika 5: ER dijagram entiteta *tip_knjige*

Predmet:

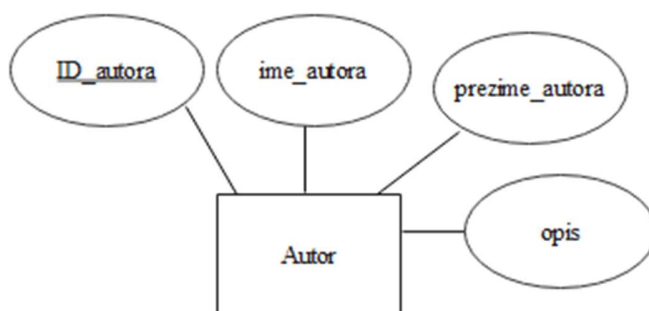
Entitet *predmet* sadrži ključ *šifra_predmeta* i atribut *naziv_predmeta*.



Sika 6: ER dijagram entiteta *predmet*

Autor:

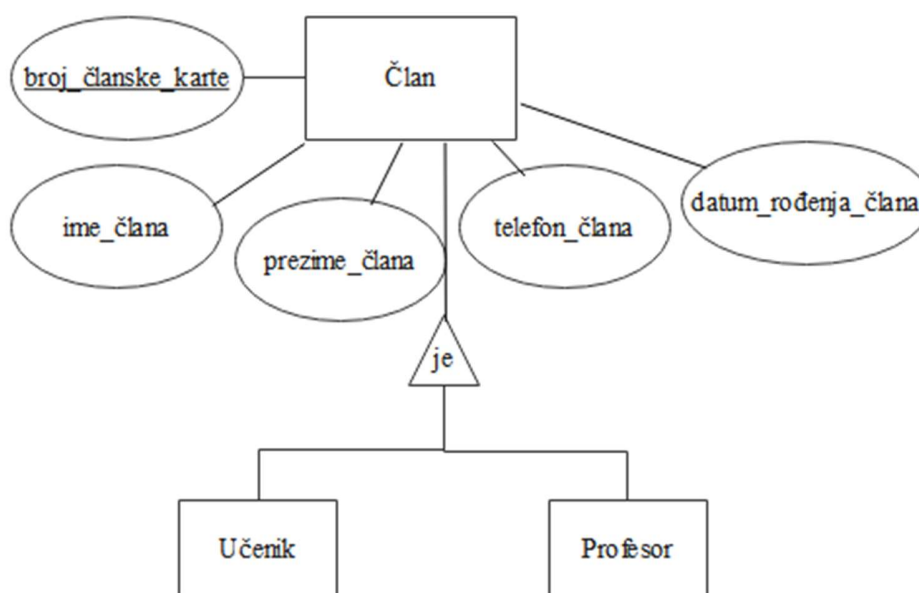
Entitet *autor* predstavlja autora određene knjige, koji ima svoj identifikator *ID_aura*, kao i attribute *ime_aura*, *prezime_aura* i *opis*.



Slika 7: ER dijagram entiteta *autor*

Član i podtipovi člana:

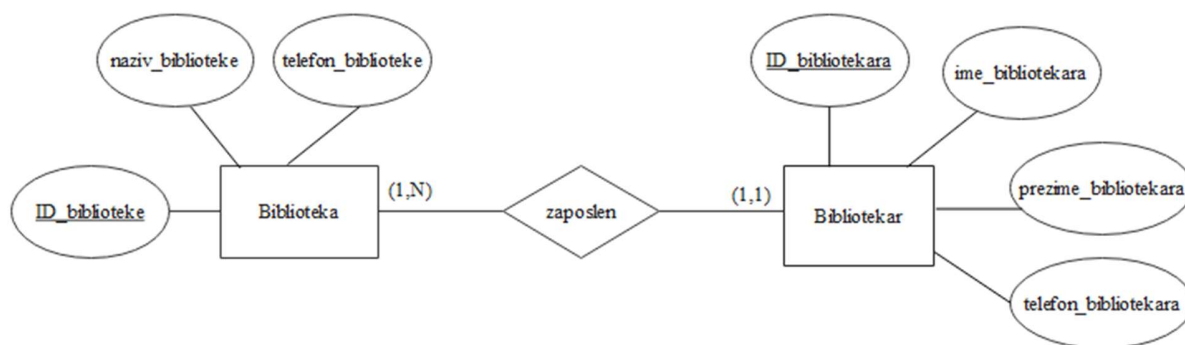
Entitet *član* pored svog identifikatora *broj_članske_karte*, sadrži i atribute *ime_člana*, *prezime_člana*, *telefon_člana*, *datum_rođenja_člana*. Ovaj entitet sadrži i dva podentiteta *učenik* i *profesor*.



Slika 8: ER dijagram entiteta *član* i podentiteta *učenik* i *profesor*

2.2 Veze

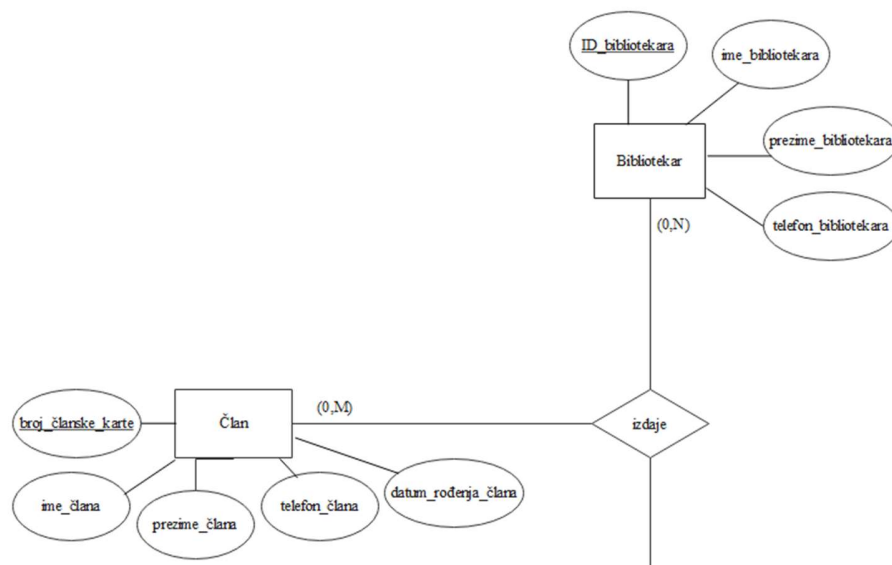
Veza bibliotekar --- biblioteka: Veza sa strane entiteta *biblioteka* je 1,N i u ovom slučaju ovaj entitet mora da ima najmanje jednog zaposlenog, a najčešće ima više. Veza sa strane entiteta *bibliotekar* je 1,1. To znači da jedan bibliotekar radi tačno i samo u jednoj biblioteci. Kardinalnost veze je 1:N. Naziv ove veze je *zaposlen*.

Slika 9: ER dijagram veze *zaposlen* entiteta *bibliotekar* i *biblioteka*

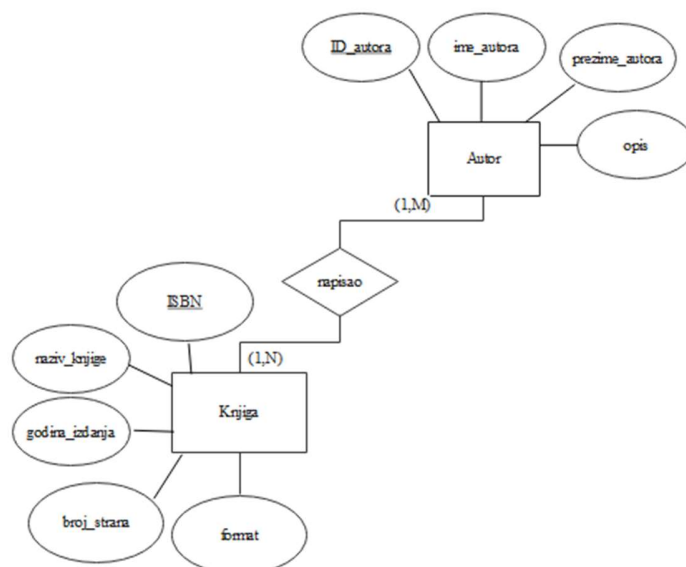
Veza knjiga--- bibliotekar: Svaku knjigu(koja se izda) će izdati jedan ili više bibliotekara ali, sa druge strane, svaki bibliotekar može da ne izda nijednu knjigu, a može da ih izda i više. Naziv veze je *izdaje* a njena kardinalnost je M:N.

Slika 10: ER dijagram veze *izdaje* entiteta *knjiga* i *bibliotekar*

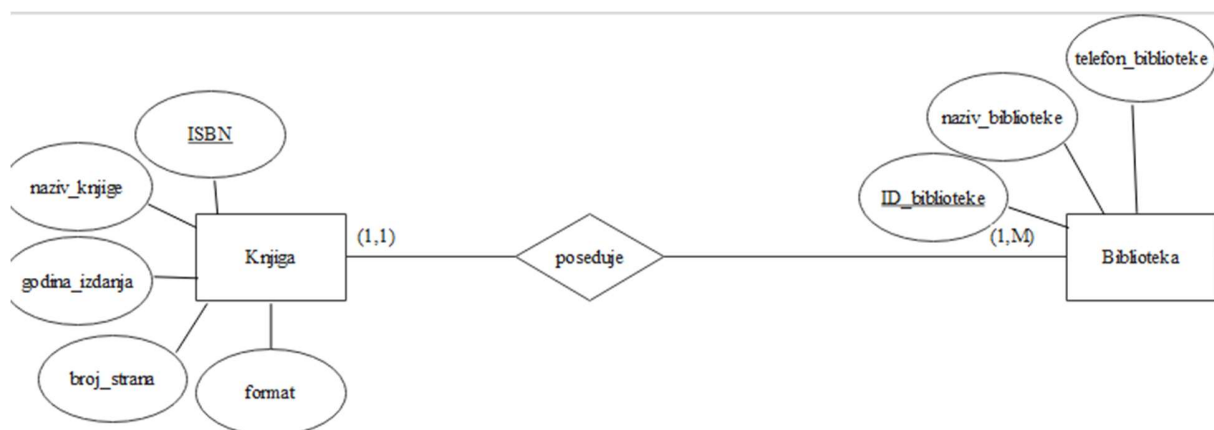
Veza bibliotekar--- član: Svaku knjigu koja se izdaje će iznajmiti nijedan ili više članova biblioteke, a svaki bibliotekar može da ne izda knjigu nijednom članu, a može da izda i većem broju članova. Naziv veze je *izdaje* a njena kardinalnost je N:M.

Slika 11: ER dijagram veze *izdaje* entiteta *bibliotekar* i *član*

Veza knjiga --- autor : Određenu knjigu, može da napiše i više različitih autora i jedan autor može da napiše više različitih knjiga. Kardinalnost ove veze je N:M. Naziv veze je *napisao*.

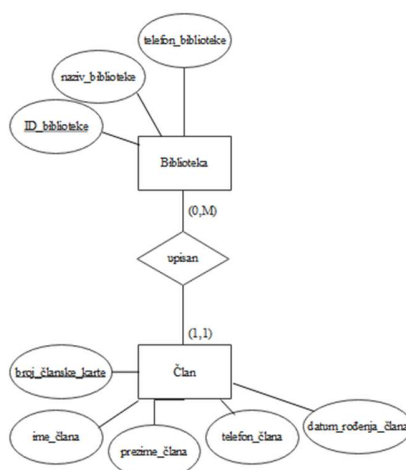
Slika 12: ER dijagram veze *napisao* entiteta *knjiga* i *autor*

Veza knjiga--- biblioteka: Veza sa strane entitet *biblioteka* nam govori da biblioteka može posedovati veći broj knjiga, pri čemu mora posedovati bar jednu knjigu da bi uopšte bila biblioteka. Veza sa strane *knjiga* je (1,1) što znači da se jedna knjiga može nalaziti samo u jednoj biblioteci. Kardinalnost veze je 1:M, dok je njen naziv *posедује*.



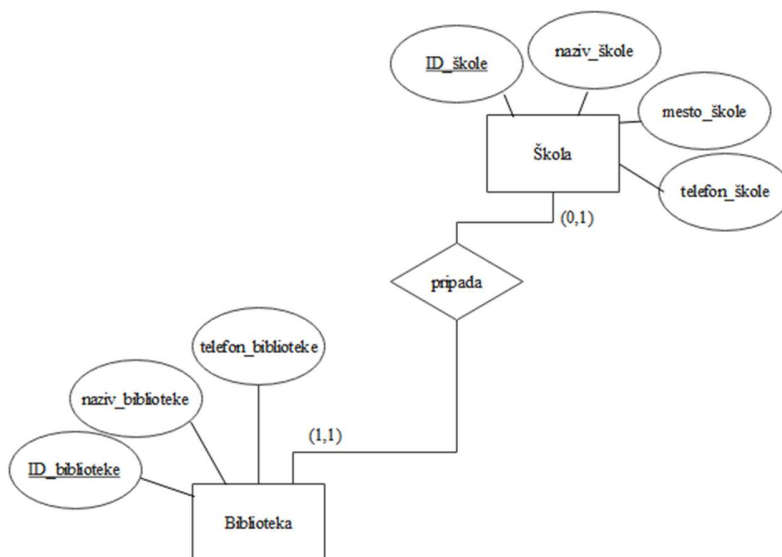
Slika 13: ER dijagram veze *poseduje* entiteta *knjiga* i *biblioteka*

Veza član---biblioteka: U biblioteci može biti učlanjen veći broj članova, a ne mora nijedan(ako je biblioteka tek počela sa radom). Veza sa strane član govori da jedan član može biti učlanjen samo u jednoj biblioteci. Kardinalnost veze je 1:M, dok je njen naziv *upisan*.



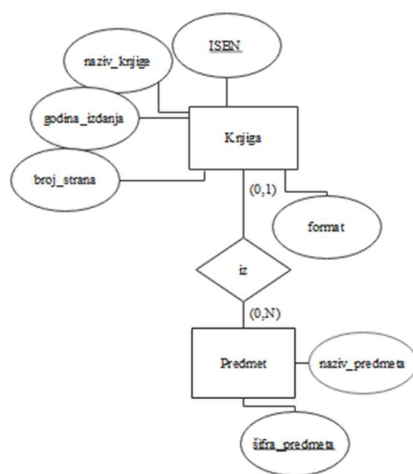
Slika 14: ER dijagram veze *upisan* entiteta *član* i *biblioteka*

Veza biblioteka --- škola : Biblioteka, da bi postojala u našoj bazi, mora da se nalazi u školi, pri čemu se jedna biblioteka može nalaziti u jednoj i samo jednoj školi. Sa druge strane, u jednoj školi postoji samo jedna biblioteka, a postoji mogućnost da imamo školu u kojoj nema nijedne biblioteke. Kardinalnost ove veze je 1:1. Njen naziv je *pripada*.

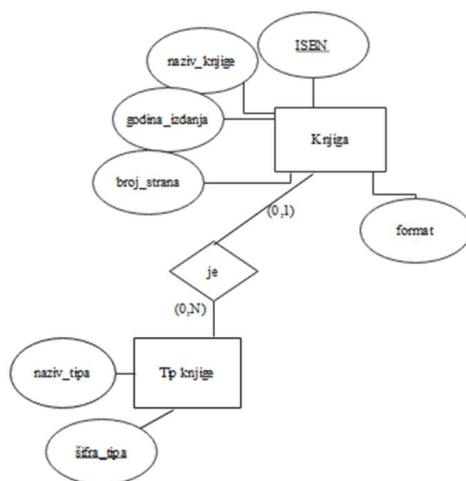


Slika 15: ER dijagram veze *pripada* entiteta *biblioteka* i *škola*

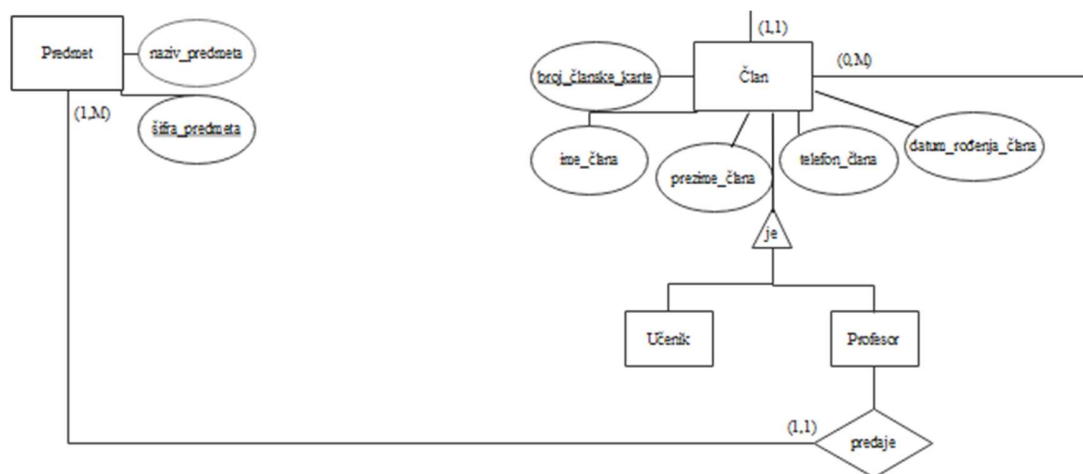
Veza knjiga---predmet: Neka knjiga iz biblioteke može biti udžbenik za neki predmet, a ako jeste, može biti samo za jedan predmet. Sa druge strane, za jedan predmet može postojati i veći broj knjiga. Prema tome, kardinalnost ove veze je 1:N, a njen naziv je *iz*.

Slika 16: ER dijagram veze *iz* entiteta *knjiga* i *predmet*

Veza knjiga---tip knjige: Neka knjiga iz biblioteke može pripadati nekom od tipova (ako nije knjiga za neki predmet), a ako pripada nekom tipu, može pripadati samo jednom. Sa druge strane, jednom tipu može pripadati i više knjiga. Prema tome, kardinalnost ove veze je 1:N, a njen naziv je *je*.

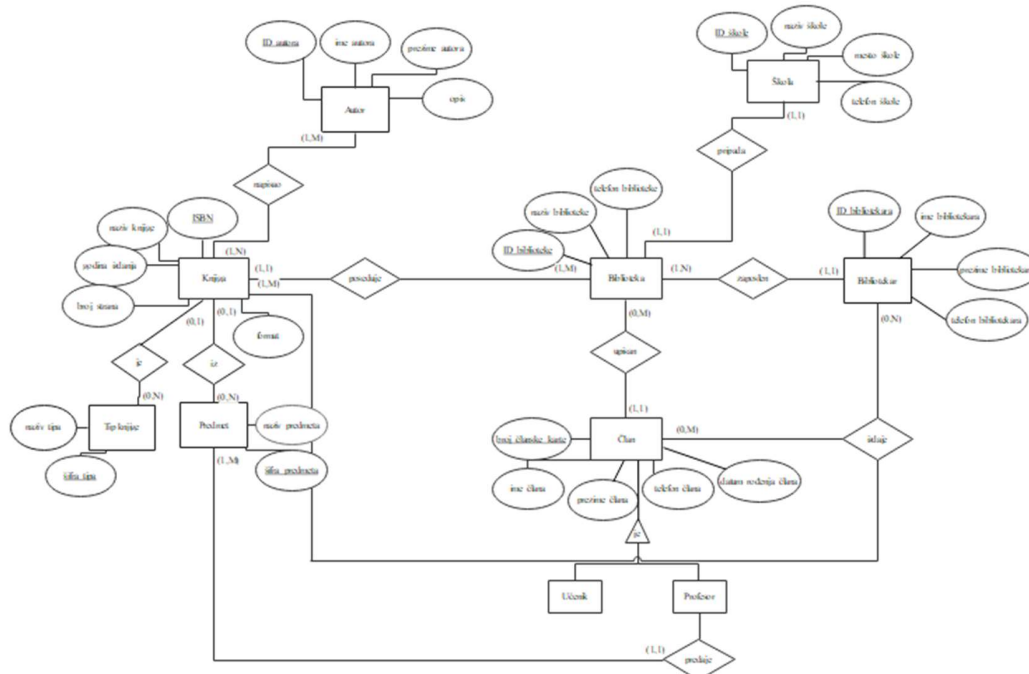
Slika 17: ER dijagram veze *je* entiteta *knjiga* i *tip knjige*

Veza profesor---predmet: Veza sa strane entiteta *predmet* nam govori da predmet može predavati i veći broj profesora. Veza sa strane *profesor* je (1,1) što znači da jedan profesor predaje samo jedan predmet. Kardinalnost veze je 1:M, dok je njen naziv *predaje*.



Slika 18: ER dijagram veze *predaje* entiteta *profesor* i *predmet*

2.3 Kompletan ER diagram



Slika 19: ER dijagram kompletnog sistema

3. Logička šema relacije 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 entitet u našoj šemi postaje nezavisna šema relacije. Ime entiteta postaje ime šeme relacije. Identifikator entiteta postaje *primarni ključ* šeme relacije. Obeležja tipa objekta su obeležja šeme relacije.

Dobijamo sledeće:

Biblioteka(ID biblioteke, naziv biblioteke, telefon biblioteke, ID škole)

Bibliotekar(ID_bibliotekara, ime_bibliotekara, prezime_bibliotekara, telefon_bibliotekara, **ID_biblioteke**)

Škola(ID_škole, naziv_škole, mesto_škole, telefon_škole)

Knjiga(ISBN, naziv_knjige, godina_izdanja, broj_strana, format, **ID_biblioteke**)

Autor(ID_autora, ime_autora, prezime_autora, opis)

Član(broj_članske_karte, ime_člana, prezime_člana, telefon_člana, datum_rođenja_člana, **ID_biblioteke**)

Predmet(šifra_predmeta, naziv_predmeta)

Tip knjige(šifra_tipa, naziv_tipa)

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

Učenik (broj_članske_karte)

Profesor(broj_članske_karte, šifra_predmeta)

3.2 Veze

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

Relacija *Izdaje* nastaje od veze *izdaje* entiteta *knjiga* i *bibliotekar*. Kardinalnost ove veze je (0,N):(1,M), 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 složeni ključ koji se sastoji od identifikatora objekata koji su u vezi.

Izdaje(ISBN, ID_bibliotekara)

Relacija *Izdaje* nastaje i od veze *izdaje* entiteta *član* i *bibliotekar*. Kardinalnost ove veze je (0,N):(0,M), što znači da i u ovom slučaju veza postaje posebna šema relacije. Obeležja ove šeme relacije su identifikatori objekata koji su u vezi, a ključ šeme relacije je složeni ključ koji se sastoji od identifikatora objekata koji su u vezi. Obzirom da ova šema relacije već postoji, i kao takva za primarni ključ ima ID_bibliotekara, njoj ćemo dodati primarni ključ entiteta član.

Izdaje(ISBN, ID_bibliotekara, broj_članske_karte)

Relacija *Napisao* nastaje od veze *napisao* entiteta *knjiga* i *autor*. Njena kardinalost je (1,N):(1,M) i relacija se dobija na potpuno isti način kao prethodne dve.

Napisao(ISBN, ID_autora)

Relacija *Iz* nastaje od veze *iz* entiteta *knjiga* i *predmet*. Kardinalnost ove veze je (0,1):(0,N), 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 ključ objekta za koji je GG=1.

Iz(ISBN , šifra_predmeta)

Relacija *Je* nastaje od veze *je* entiteta *knjiga* i *tip knjige*. Kardinalnost ove veze je (0,1):(0,N), što znači da je prevodimo identično kao u prethodnom slučaju.

Je(ISBN,šifra_tipa)

3.3 Međurelaciona ograničenja

Mogu se definisati sledeća međurelaciona ograničenja:

Biblioteka[ID_škole] \subseteq Škola[ID_škole]
 Bibliotekar[ID_biblioteke] \subseteq Biblioteka[ID_biblioteke]
 Knjiga[ID_biblioteke] \subseteq Biblioteka[ID_biblioteke]
 Član[ID_biblioteke] \subseteq Biblioteka[ID_biblioteke]
 Napisao[ID_autora] \subseteq Autor[ID_autora]
 Napisao[ISBN] \subseteq Knjiga[ISBN]
 Izdaje[broj_članske_karte] \subseteq Član[broj_članske_karte]
 Izdaje[ISBN] \subseteq Knjiga[ISBN]
 Izdaje[ID_bibliotekara] \subseteq Bibliotekar[ID_bibliotekara]
 Iz[šifra_predmeta] \subseteq Predmet[šifra_predmeta]
 Iz[ISBN] \subseteq Knjiga[ISBN]
 Je[šifra_tipa] \subseteq Tip knjige[šifra_tipa]
 Je[ISBN] \subseteq Knjiga[ISBN]
 Učenik[broj_članske_karte] \subseteq Član[broj_članske_karte]
 Profesor[broj_članske_karte] \subseteq Član[broj_članske_karte]
 Profesor[šifra_predmeta] \subseteq Predmet[šifra_predmeta]

3.4 Kompletna logička šema

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

S={

Biblioteka(ID_biblioteke, naziv_biblioteke, telefon_biblioteke, **ID_škole**)
 Bibliotekar(ID_bibliotekara, ime_bibliotekara, prezime_bibliotekara, telefon_bibliotekara, **ID_biblioteke**)
 Škola(ID_škole, naziv_škole, mesto_škole, telefon_škole)
 Knjiga(ISBN, naziv_knjige, godina_izdanja, broj_strana, format, **ID_biblioteke**)
 Autor(ID_autora, ime_autora, prezime_autora, opis)
 Član(broj_članske_karte, ime_člana, prezime_člana, telefon_člana, datum_rođenja_člana, **ID_biblioteke**)
 Predmet(šifra_predmeta, naziv_predmeta)
 Tip_knjige(šifra_tipa, naziv_tipa)
 Učenik (broj_članske_karte)
 Profesor(broj_članske_karte, **šifra_predmeta**)
 Izdaje(ISBN, ID_bibliotekara, broj_članske_karte)
 Napisao(ISBN, ID_autora)
 Iz(ISBN , šifra_predmeta)
 Je(ISBN, šifra_tipa)

}

I={

Biblioteka[ID_škole] ⊆ Škola[ID_škole]
 Bibliotekar[ID_biblioteke] ⊆ Biblioteka[ID_biblioteke]
 Knjiga[ID_biblioteke] ⊆ Biblioteka[ID_biblioteke]
 Član[ID_biblioteke] ⊆ Biblioteka[ID_biblioteke]
 Napisao[ID_autora] ⊆ Autor[ID_autora]
 Napisao[ISBN] ⊆ Knjiga[ISBN]
 Izdaje[broj_članske_karte] ⊆ Član[broj_članske_karte]
 Izdaje[ISBN] ⊆ Knjiga[ISBN]
 Izdaje[ID_bibliotekara] ⊆ Bibliotekar[ID_bibliotekara]
 Iz[šifra_predmeta] ⊆ Predmet[šifra_predmeta]
 Iz[ISBN] ⊆ Knjiga[ISBN]
 Je[šifra_tipa] ⊆ Tip_knjige[šifra_tipa]

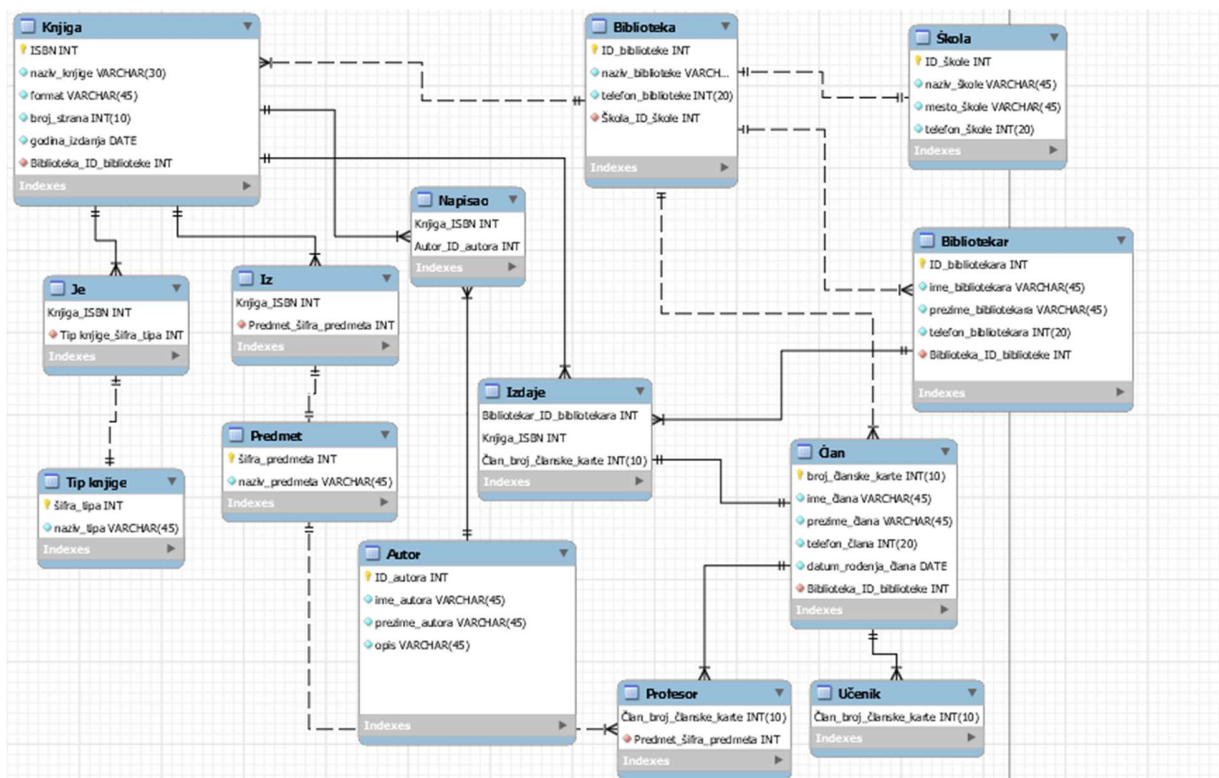
```

Je[ISBN] ⊆ Knjiga[ISBN]
Učenik[broj_članske_karte] ⊆ Član[broj_članske_karte]
Profesor[broj_članske_karte] ⊆ Član[broj_članske_karte]
Profesor[šifra_predmeta] ⊆ Predmet[šifra_predmeta]
}

```

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 20: 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 je sledeće što će biti prikazano, zajedno sa testnim podacima koji su ručno ubačeni.

```
-- MySQL Script generated by MySQL Workbench
```

```
-- Fri Jun 11 11:03:56 2021
```

```
-- 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 mydb
```

```
-- -----
```

```
-- -----
```

```
-- Schema mydb
```

```
-- -----
```

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

```
USE `mydb` ;
```

```
-- -----  
-- Table `mydb`.`Škola`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Škola` (  
  `ID_škole` INT NOT NULL,  
  `naziv_škole` VARCHAR(45) NOT NULL,  
  `mesto_škole` VARCHAR(45) NOT NULL,  
  `telefon_škole` INT(20) NOT NULL,  
  PRIMARY KEY (`ID_škole`))  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `mydb`.`Biblioteka`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Biblioteka` (  
  `ID_biblioteke` INT NOT NULL,  
  `naziv_biblioteke` VARCHAR(45) NOT NULL,  
  `telefon_biblioteke` INT(20) NOT NULL,  
  `Škola_ID_škole` INT NOT NULL,  
  PRIMARY KEY (`ID_biblioteke`),  
  INDEX `fk_Biblioteka_Škola1_idx` (`Škola_ID_škole` ASC) VISIBLE,  
  CONSTRAINT `fk_Biblioteka_Škola1`  
    FOREIGN KEY (`Škola_ID_škole`)
```

```
REFERENCES `mydb`.`Škola` (`ID_škole`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-----

-- Table `mydb`.`Bibliotekar`

-----

CREATE TABLE IF NOT EXISTS `mydb`.`Bibliotekar` (

  `ID_bibliotekara` INT NOT NULL,

  `ime_bibliotekara` VARCHAR(45) NOT NULL,

  `prezime_bibliotekara` VARCHAR(45) NOT NULL,

  `telefon_bibliotekara` INT(20) NOT NULL,

  `Biblioteka_ID_biblioteke` INT NOT NULL,

  PRIMARY KEY (`ID_bibliotekara`),

  INDEX `fk_Bibliotekar_Biblioteka_idx` (`Biblioteka_ID_biblioteke` ASC) VISIBLE,

  CONSTRAINT `fk_Bibliotekar_Biblioteka`

    FOREIGN KEY (`Biblioteka_ID_biblioteke`)

      REFERENCES `mydb`.`Biblioteka` (`ID_biblioteke`)

        ON DELETE NO ACTION

        ON UPDATE NO ACTION)

ENGINE = InnoDB;
```

```
-- -----  
-- Table `mydb`.`Knjiga`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Knjiga` (  
  `ISBN` INT NOT NULL,  
  `naziv_knjige` VARCHAR(30) NOT NULL,  
  `format` VARCHAR(45) NOT NULL,  
  `broj_strana` INT(10) NOT NULL,  
  `godina_izdanja` DATE NOT 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 `mydb`.`Biblioteka` (`ID_biblioteke`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `mydb`.`Autor`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Autor` (  
  `ID_autora` INT NOT NULL,
```

```
`ime_autora` VARCHAR(45) NOT NULL,  
`prezime_autora` VARCHAR(45) NOT NULL,  
`opis` VARCHAR(45) NOT NULL,  
PRIMARY KEY (`ID_autora`))  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `mydb`.`Član`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Član` (  
  `broj_članske_karte` INT(10) NOT NULL,  
  `ime_člana` VARCHAR(45) NOT NULL,  
  `prezime_člana` VARCHAR(45) NOT NULL,  
  `telefon_člana` INT(20) NOT NULL,  
  `datum_rođenja_člana` DATE NOT NULL,  
  `Biblioteka_ID_biblioteke` INT NOT NULL,  
  PRIMARY KEY (`broj_članske_karte`),  
  INDEX `fk_Član_Biblioteka1_idx` (`Biblioteka_ID_biblioteke` ASC) VISIBLE,  
  CONSTRAINT `fk_Član_Biblioteka1`  
    FOREIGN KEY (`Biblioteka_ID_biblioteke`)  
    REFERENCES `mydb`.`Biblioteka` (`ID_biblioteke`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```



```
-- -----  
-- Table `mydb`.`Izdaje`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Izdaje` (  
  `Bibliotekar_ID_bibliotekara` INT NOT NULL,  
  `Knjiga_ISBN` INT NOT NULL,  
  `Član_broj_članske_karte` INT(10) NOT NULL,  
  PRIMARY KEY (`Bibliotekar_ID_bibliotekara`, `Knjiga_ISBN`, `Član_broj_članske_karte`),  
  INDEX `fk_Bibliotekar_has_Knjiga_Knjiga1_idx` (`Knjiga_ISBN` ASC) VISIBLE,  
  INDEX `fk_Bibliotekar_has_Knjiga_Bibliotekar1_idx` (`Bibliotekar_ID_bibliotekara` ASC)  
  VISIBLE,  
  INDEX `fk_Izdaje_Član1_idx` (`Član_broj_članske_karte` ASC) VISIBLE,  
  CONSTRAINT `fk_Bibliotekar_has_Knjiga_Bibliotekar1`  
    FOREIGN KEY (`Bibliotekar_ID_bibliotekara`)  
    REFERENCES `mydb`.`Bibliotekar` (`ID_bibliotekara`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Bibliotekar_has_Knjiga_Knjiga1`  
    FOREIGN KEY (`Knjiga_ISBN`)  
    REFERENCES `mydb`.`Knjiga` (`ISBN`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Izdaje_Član1`
```

```
FOREIGN KEY (`Član_broj_članske_karte`)  
REFERENCES `mydb`.`Član` (`broj_članske_karte`)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
  
-----  
-- Table `mydb`.`Napisao`  
-----  
  
CREATE TABLE IF NOT EXISTS `mydb`.`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 `mydb`.`Knjiga` (`ISBN`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Knjiga_has_Autor_Autor1`  
    FOREIGN KEY (`Autor_ID_autora`)  
    REFERENCES `mydb`.`Autor` (`ID_autora`)  
    ON DELETE NO ACTION
```

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Table `mydb`.`Profesor`

```
CREATE TABLE IF NOT EXISTS `mydb`.`Profesor` (  
  `Član_broj_članske_karte` INT(10) NOT NULL,  
  `Predmet_šifra_predmeta` INT NOT NULL,  
  PRIMARY KEY (`Član_broj_članske_karte`),  
  INDEX `fk_Profesor_Predmet1_idx` (`Predmet_šifra_predmeta` ASC) VISIBLE,  
  CONSTRAINT `fk_Profesor_Član1`  
    FOREIGN KEY (`Član_broj_članske_karte`)  
      REFERENCES `mydb`.`Član` (`broj_članske_karte`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Profesor_Predmet1`  
    FOREIGN KEY (`Predmet_šifra_predmeta`)  
      REFERENCES `mydb`.`Predmet` (`šifra_predmeta`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `mydb`.`Učenik`

CREATE TABLE IF NOT EXISTS `mydb`.`Učenik` (
 `Član_broj_članske_karte` INT(10) NOT NULL,
 PRIMARY KEY (`Član_broj_članske_karte`),
 CONSTRAINT `fk_Učenik_Član1`
 FOREIGN KEY (`Član_broj_članske_karte`)
 REFERENCES `mydb`.`Član` (`broj_članske_karte`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- Table `mydb`.`Tip knjige`

CREATE TABLE IF NOT EXISTS `mydb`.`Tip knjige` (
 `šifra_tipa` INT NOT NULL,
 `naziv_tipa` VARCHAR(45) NOT NULL,
 PRIMARY KEY (`šifra_tipa`))
ENGINE = InnoDB;

-- Table `mydb`.`Predmet`

```
-- -----  
CREATE TABLE IF NOT EXISTS `mydb`.`Predmet` (  
  `šifra_predmeta` INT NOT NULL,  
  `naziv_predmeta` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`šifra_predmeta`))  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `mydb`.`Je`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Je` (  
  `Tip knjige_šifra_tipa` INT NOT NULL,  
  `Knjiga_ISBN` INT NOT NULL,  
  PRIMARY KEY (`Tip knjige_šifra_tipa`),  
  INDEX `fk_Tip knjige_has_Knjiga_Knjiga1_idx` (`Knjiga_ISBN` ASC) VISIBLE,  
  INDEX `fk_Tip knjige_has_Knjiga_Tip knjige1_idx` (`Tip knjige_šifra_tipa` ASC) VISIBLE,  
  CONSTRAINT `fk_Tip knjige_has_Knjiga_Tip knjige1`  
    FOREIGN KEY (`Tip knjige_šifra_tipa`)  
      REFERENCES `mydb`.`Tip knjige` (`šifra_tipa`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Tip knjige_has_Knjiga_Knjiga1`  
    FOREIGN KEY (`Knjiga_ISBN`)  
      REFERENCES `mydb`.`Knjiga` (`ISBN`)
```

```
ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-----

-- Table `mydb`.`Iz`

-----

CREATE TABLE IF NOT EXISTS `mydb`.`Iz` (
  `Predmet_šifra_predmeta` INT NOT NULL,
  `Knjiga_ISBN` INT NOT NULL,
  PRIMARY KEY (`Predmet_šifra_predmeta`),
  INDEX `fk_Predmet_has_Knjiga_Knjiga1_idx` (`Knjiga_ISBN` ASC) VISIBLE,
  INDEX `fk_Predmet_has_Knjiga_Predmet1_idx` (`Predmet_šifra_predmeta` ASC) VISIBLE,
  CONSTRAINT `fk_Predmet_has_Knjiga_Predmet1`
    FOREIGN KEY (`Predmet_šifra_predmeta`)
      REFERENCES `mydb`.`Predmet` (`šifra_predmeta`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `fk_Predmet_has_Knjiga_Knjiga1`
    FOREIGN KEY (`Knjiga_ISBN`)
      REFERENCES `mydb`.`Knjiga` (`ISBN`)
        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;

-- -----

-- Data for table `mydb`.`Škola`

-- -----

START TRANSACTION;

USE `mydb`;

INSERT INTO `mydb`.`Škola` (`ID_škole`, `naziv_škole`, `telefon_škole`, `mesto_škole`)
VALUES (1, 'Desanka Maksimović', 037723332, 'Kragujevac');

INSERT INTO `mydb`.`Škola` (`ID_škole`, `naziv_škole`, `telefon_škole`, `mesto_škole`)
VALUES (2, 'Dobrica Ćosić', 037723452, 'Kraljevo');

INSERT INTO `mydb`.`Škola` (`ID_škole`, `naziv_škole`, `telefon_škole`, `mesto_škole`)
VALUES (3, 'Vuk Karadžić', 037723922, 'Novi Sad');

INSERT INTO `mydb`.`Škola` (`ID_škole`, `naziv_škole`, `telefon_škole`, `mesto_škole`)
VALUES (4, 'Branislav Nušić', 037723652, 'Beograd');

COMMIT;

-- -----

-- -----

-- Data for table `mydb`.`Biblioteka`
```

```
-----  
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`Biblioteka` (`ID_biblioteke`, `naziv_biblioteke`, `telefon_biblioteke`,  
`Škola_ID_škole`) VALUES (1, 'Desanka Maksimović', 037723331, 1);
```

```
INSERT INTO `mydb`.`Biblioteka` (`ID_biblioteke`, `naziv_biblioteke`, `telefon_biblioteke`,  
`Škola_ID_škole`) VALUES (2, 'Dobrica Ćosić', 03773652, 2);
```

```
INSERT INTO `mydb`.`Biblioteka` (`ID_biblioteke`, `naziv_biblioteke`, `telefon_biblioteke`,  
`Škola_ID_škole`) VALUES (3, 'Vuk Karadžić', 03774331, 3);
```

```
INSERT INTO `mydb`.`Biblioteka` (`ID_biblioteke`, `naziv_biblioteke`, `telefon_biblioteke`,  
`Škola_ID_škole`) VALUES (4, 'Branislav Nušić', 037721735, 4);
```

```
COMMIT;
```

```
-----  
-- Data for table `mydb`.`Bibliotekar`  
-----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`Bibliotekar` (`ID_bibliotekara`, `ime_bibliotekara`,  
`prezime_bibliotekara`, `telefon_bibliotekara`, `Biblioteka_ID_biblioteke`) VALUES (11, 'Ana',  
'Ivović', 063354231, 1);
```

```
INSERT INTO `mydb`.`Bibliotekar` (`ID_bibliotekara`, `ime_bibliotekara`,  
`prezime_bibliotekara`, `telefon_bibliotekara`, `Biblioteka_ID_biblioteke`) VALUES (22,  
'Mirko', 'Nikolić', 062865213, 2);
```

```
INSERT INTO `mydb`.`Bibliotekar` (`ID_bibliotekara`, `ime_bibliotekara`,  
`prezime_bibliotekara`, `telefon_bibliotekara`, `Biblioteka_ID_biblioteke`) VALUES (33,  
'Milica', 'Živadinović', 065432176, 3);
```



```
INSERT INTO `mydb`.`Bibliotekar` (`ID_bibliotekara`, `ime_bibliotekara`,  
`prezime_bibliotekara`, `telefon_bibliotekara`, `Biblioteka_ID_biblioteke`) VALUES (44,  
'Andrijana', 'Tomašević', 065543213, 4);
```

```
COMMIT;
```

```
-- -----  
-- Data for table `mydb`.`Knjiga`  
-- -----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`Knjiga` (`ISBN`, `naziv_knjige`, `format`, `broj_strana`,  
`godina_izdanja`, `Biblioteka_ID_biblioteke`) VALUES (1554321, 'Tvrđica', 'A5', 120, 1989.,  
1);
```

```
INSERT INTO `mydb`.`Knjiga` (`ISBN`, `naziv_knjige`, `format`, `broj_strana`,  
`godina_izdanja`, `Biblioteka_ID_biblioteke`) VALUES (223456, 'Kir Janja', 'A5', 200, 1995.,  
2);
```

```
INSERT INTO `mydb`.`Knjiga` (`ISBN`, `naziv_knjige`, `format`, `broj_strana`,  
`godina_izdanja`, `Biblioteka_ID_biblioteke`) VALUES (397652, 'Ana Karenjina', 'A4', 400,  
1999., 3);
```

```
INSERT INTO `mydb`.`Knjiga` (`ISBN`, `naziv_knjige`, `format`, `broj_strana`,  
`godina_izdanja`, `Biblioteka_ID_biblioteke`) VALUES (498765, 'Znakovi pored puta', 'A4',  
120, 1999., 4);
```

```
COMMIT;
```

```
-- -----
```

-- Data for table `mydb`.`Autor`

START TRANSACTION;

USE `mydb`;

INSERT INTO `mydb`.`Autor` (`ID_aura`, `ime_aura`, `prezime_aura`, `opis`) VALUES
(1, 'Jovan Sterija', 'Popović', 'Komediograf i pisac');

INSERT INTO `mydb`.`Autor` (`ID_aura`, `ime_aura`, `prezime_aura`, `opis`) VALUES
(2, 'Jovan Sterija', 'Popović', 'Komediograf i pisac');

INSERT INTO `mydb`.`Autor` (`ID_aura`, `ime_aura`, `prezime_aura`, `opis`) VALUES
(3, 'Lav', 'Nikolajevič', 'Lav Nikolajevič Tolstoj, ruski grof, književnik i mislilac');

INSERT INTO `mydb`.`Autor` (`ID_aura`, `ime_aura`, `prezime_aura`, `opis`) VALUES
(4, 'Ivo', 'Andrić', 'Ivo Andrić bio je jugoslavenski književnik');

COMMIT;

-- Data for table `mydb`.`Član`

START TRANSACTION;

USE `mydb`;

INSERT INTO `mydb`.`Član` (`broj_članske_karte`, `ime_člana`, `prezime_člana`,
`telefon_člana`, `datum_rođenja_člana`, `Biblioteka_ID_biblioteke`) VALUES (101, 'Ana',
Popović', 0638004321', '01.11.1999.', 1);

INSERT INTO `mydb`.`Član` (`broj_članske_karte`, `ime_člana`, `prezime_člana`,
`telefon_člana`, `datum_rođenja_člana`, `Biblioteka_ID_biblioteke`) VALUES (102, 'Ana',
Ivović', 06380043222', '10.11.1998.', 2);

```
INSERT INTO `mydb`.`Član` (`broj_članske_karte`, `ime_člana`, `prezime_člana`,  
`telefon_člana`, `datum_rođenja_člana`, Biblioteka_ID_biblioteke) VALUES (103, 'Anastasija',  
'Popović', 063100321, '01.01.1998.', 3);
```

```
INSERT INTO `mydb`.`Član` (`broj_članske_karte`, `ime_člana`, `prezime_člana`,  
`telefon_člana`, `datum_rođenja_člana`, Biblioteka_ID_biblioteke) VALUES (104, 'Jana', '  
Jovović', 0638144321, '11.11.1999.', 4);
```

```
COMMIT;
```

```
-- -----  
-- Data for table `mydb`.`Izdaje`  
-- -----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`Izdaje` (`Knjiga_ISBN`, `Bibliotekar_ID_bibliotekara`,  
`Član_broj_članske_karte`) VALUES (1554321, 11, 101);
```

```
INSERT INTO `mydb`.`Izdaje` (`Knjiga_ISBN`, `Bibliotekar_ID_bibliotekara`,  
`Član_broj_članske_karte`) VALUES (223456, 22, 102);
```

```
INSERT INTO `mydb`.`Izdaje` (`Knjiga_ISBN`, `Bibliotekar_ID_bibliotekara`,  
`Član_broj_članske_karte`) VALUES (397652, 33, 103);
```

```
INSERT INTO `mydb`.`Izdaje` (`Knjiga_ISBN`, `Bibliotekar_ID_bibliotekara`,  
`Član_broj_članske_karte`) VALUES (498765, 44, 104);
```

```
COMMIT;
```

```
-- Data for table `mydb`.`Napisao`
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`Napisao` (`Autor_ID_autora`, `Knjiga_ISBN`) VALUES (1, 1554321);
```

```
INSERT INTO `mydb`.`Napisao` (`Autor_ID_autora`, `Knjiga_ISBN`) VALUES (2, 223456);
```

```
INSERT INTO `mydb`.`Napisao` (`Autor_ID_autora`, `Knjiga_ISBN`) VALUES (3, 397652);
```

```
INSERT INTO `mydb`.`Napisao` (`Autor_ID_autora`, `Knjiga_ISBN`) VALUES (4, 498765);
```

```
COMMIT;
```

```
-- Data for table `mydb`.`Profesor`
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`Profesor` (`Predmet_šifra_predmeta`, `Član_broj_članske_karte`) VALUES (10,101);
```

```
INSERT INTO `mydb`.`Profesor` (`Predmet_šifra_predmeta`, `Član_broj_članske_karte`) VALUES (20,102);
```

```
INSERT INTO `mydb`.`Profesor` (`Predmet_šifra_predmeta`, `Član_broj_članske_karte`) VALUES (30,103);
```

```
INSERT INTO `mydb`.`Profesor` (`Predmet_šifra_predmeta`, `Član_broj_članske_karte`) VALUES (40,104);
```

```
COMMIT;
```

```
-- -----  
-- Data for table `mydb`.`Učenik`  
-- -----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`Učenik` (`Član_broj_članske_karte`) VALUES (101);
```

```
INSERT INTO `mydb`.`Učenik` (`Član_broj_članske_karte`) VALUES (102);
```

```
INSERT INTO `mydb`.`Učenik` (`Član_broj_članske_karte`) VALUES (103);
```

```
INSERT INTO `mydb`.`Učenik` (`Član_broj_članske_karte`) VALUES (104);
```

```
COMMIT;
```

```
-- -----  
-- Data for table `mydb`.`Tip knjige`  
-- -----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`Tip knjige` (`šifra_tipa`, `naziv_tipa`) VALUES (10, 'lektira');
```

```
INSERT INTO `mydb`.`Tip knjige` (`šifra_tipa`, `naziv_tipa`) VALUES (20, 'roman');
```

```
INSERT INTO `mydb`.`Tip knjige` (`šifra_tipa`, `naziv_tipa`) VALUES (30, 'roman');
```

```
INSERT INTO `mydb`.`Tip knjige` (`šifra_tipa`, `naziv_tipa`) VALUES (40, '/');
```

```
COMMIT;
```

```
-- -----  
-- Data for table `mydb`.`Predmet`  
-- -----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`Predmet` (`šifra_predmeta`, `naziv_predmeta`) VALUES (10, '/');
```

```
INSERT INTO `mydb`.`Predmet` (`šifra_predmeta`, `naziv_predmeta`) VALUES (20, '/');
```

```
INSERT INTO `mydb`.`Predmet` (`šifra_predmeta`, `naziv_predmeta`) VALUES (30, '/');
```

```
INSERT INTO `mydb`.`Predmet` (`šifra_predmeta`, `naziv_predmeta`) VALUES (40,  
'biologija');
```

```
COMMIT;
```

```
-- -----  
-- Data for table `mydb`.`Je`  
-- -----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`Je` (`Tip knjige_šifra_tipa`, `ISBN`) VALUES (10, 1);
```

```
INSERT INTO `mydb`.`Je` (`šifra_predmeta`, `naziv_predmeta`) VALUES (20, 2);
```

```
INSERT INTO `mydb`.`Je` (`šifra_predmeta`, `naziv_predmeta`) VALUES (30, 3);
```

```
INSERT INTO `mydb`.`Je` (`šifra_predmeta`, `naziv_predmeta`) VALUES (40, 4);
```

```
COMMIT;
```

```
-- -----  
-- Data for table `mydb`.`Iz`  
-- -----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`Iz` (`Predmet_šifra_predmeta`, `ISBN`) VALUES (10, 1);
```

```
INSERT INTO `mydb`.`Iz` (`šifra_predmeta`, `naziv_predmeta`) VALUES (20, 2);
```

```
INSERT INTO `mydb`.`Iz` (`šifra_predmeta`, `naziv_predmeta`) VALUES (30, 3);
```

```
INSERT INTO `mydb`.`Iz` (`šifra_predmeta`, `naziv_predmeta`) VALUES (40, 4);
```

```
COMMIT;
```

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. Prikazati podatke o knjizi koju je iznajmila Jana Jovović:

```
SELECT * FROM Knjiga
WHERE ISBN IN (SELECT ISBN FROM Izdaje
WHERE broj_članske_karte IN(
SELECT broj_članske_karte FROM Član
WHERE ime_člana = "Jana"
WHERE prezime_člana = "Jovović"));
```

3. Prikazati imena svih članova koji su rođeni 1999. godine i sortirati ih u alfabetskom poretku:

```
SELECT ime_člana FROM Član
WHERE datum_rođenja_člana=1999.
ORDER BY ime_člana;
```

4. Prikazati naziv škole u kojoj radi bibliotekar Mirko Nikolić:

```
SELECT naziv_škole FROM Škola
WHERE ID_škole IN(
SELECT ID_škole FROM Biblioteka WHERE ID_biblioteke IN(
SELECT ID_biblioteke FROM Bibliotekar
WHERE ime_bibliotekara = "Mirko" and prezime_bibliotekara = "Nikolić"));
```

5. Prikazati autore knjiga čiji je broj strana veći od 200:

```
SELECT ime_autora, prezime_autora FROM Autor
WHERE ID_autora IN(
SELECT ID_autora FROM Napisao WHERE ISBN IN(
SELECT ISBN FROM Knjiga
WHERE broj_strana>200));
```


6. Literatura

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