

**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
ELEKTROTEHNIČKI FAKULTET**

**Stručni studij**

**„KNJIŽNICA“**

**Projektni zadatak**

**Boris Marenčić. Axxxx**

Novska, 2010.

## Sadržaj

1. OPIS PROJEKTOG ZADATKA.....	3
2. ER DIJAGRAM .....	4
3. RELACIJSKI MODEL.....	5
4. OPIS ODABRANOG RJEŠENJA .....	6
5. SQL NAREDBE ZA KREIRANJE BAZE PODATAKA.....	7
6. PUNJENJE BAZE PODACIMA.....	9
7. UPITI KOJI ĆE BITI NAJČEŠĆE KORIŠTENI.....	11
8. ZAKLJUČAK.....	13

Marenić

# 1. OPIS PROJEKTOG ZADATKA

## Projektni zadatak - "Knjižnica"

### Opis:

Potrebno je osmisliti i realizirati bazu podataka koja može učinkovito poslužiti za potrebe fakultetske knjižnice. U fazi analize, utvrđeno je da knjižnica raspolaže s znatnom količinom literature (knjiga, časopisa, diplomskih radova, skripti, itd.), te da pojedine literature ima u nekoliko primjeraka (npr. nekoliko primjeraka iste knjige). Osnovni zahtjevi korisnika su slijedeći:

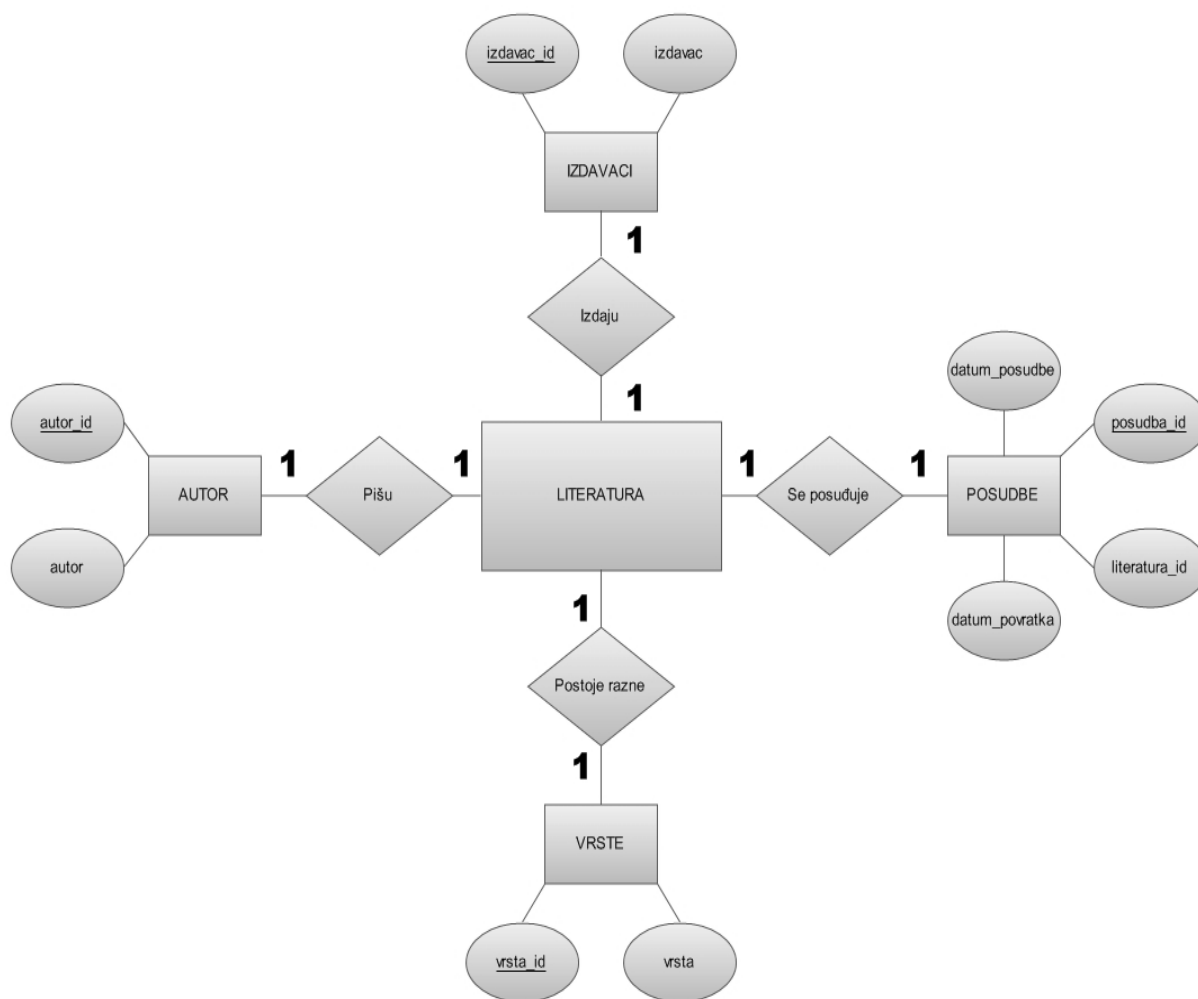
- efikasno pretraživanje raspoložive literature prema vrsti (knjiga, časopis, skripta, ...), autoru, naslovu, ISBN broju (ako je na raspolaganju) i izdavaču,
- u svakom trenutku raspoloživa informacija da li je knjiga u knjižnici ili je posuđena nekom od studenata ili profesora u kom slučaju je potrebno dobiti informaciju o datumu posudbe.
- efikasno praćenje trajanja svake posudbe

Baza podataka treba biti u što većoj mjeri normalizirana.

### Zadaci:

1. Napraviti ER dijagram (entity-relationship diagram) dijagram sa svim bitnim elementima.
2. Napraviti relacijski model
3. Dati kraći tekstualni opis odabranog rješenja uz osvrt na eventualne specifičnosti, pretpostavke ili ograničenja u modelu.
4. Napraviti SQL naredbe za kreiranje baze podataka koja odgovara relacijskom modelu.
5. Napraviti SQL naredbe kojima se baza puni podacima za potrebe testiranja.
6. Napraviti primjer SQL upita za koje se očekuje da će biti najčešće upotrebljavani od strane korisnika baze podataka (prema zahtjevima u opisu zadatka), opisati ih riječima (koja je svrha upita?) i dati konkretni primjer rezultata kakav se dobije takvim upitom.

## 2. ER DIJAGRAM



### 3. RELACIJSKI MODEL

Server: localhost ▶ Database: boris

#### autor

Field	Type	Null	Default	
<u>autor_id</u>	int(11)	No		
autor	varchar(200)	No		

#### izdavaci

Field	Type	Null	Default	
<u>izdavac_id</u>	int(11)	No		
izdavac	varchar(150)	No		

#### literatura

Field	Type	Null	Default	
<u>literatura_id</u>	int(11)	No		
isbn	varchar(15)	Yes	NULL	
autor_id	int(11)	No		autor -> autor_id
naslov	varchar(200)	No		
izdavac_id	int(11)	No		izdavaci -> izdavac_id
vrsta_id	int(11)	No		vrste -> vrsta_id
dostupno	char(2)	No	Da	

#### posudbe

Field	Type	Null	Default	
<u>posudba_id</u>	int(11)	No		
literatura_id	int(11)	No		literatura -> literatura_id
datum_posudbe	date	No		
datum_povratka	date	Yes	NULL	

#### vrste

Field	Type	Null	Default	
<u>vrsta_id</u>	int(11)	No		
vrsta	varchar(100)	No		

## 4. OPIS ODABRANOG RJEŠENJA

Kao rješenje projektnog zadatka kreirana je baza sa 5 tablica. Rješenje bi trebalo zadovoljavati sve zahtjeve navedene u projektnom zadatku.

### Tablica „autor“

Ova tablica sadržava ime i prezime autora literature. Primarni ključ je autor\_id koji je postavljen kao autopovećavajući (engl. autoincrement) te nam služi kao ograničenje (foreign key) na unos u tablici literatura.

### Tablica „izdavaci“

Ova tablica sadrži podatke o izdavačima literature. Kao primarni ključ odabrano je polje izdavac\_id koji se automatski povećava prilikom svakog unosa u tablicu te nam kao i kod prethodne tablice služi kao strani ključ u tablici literatura.

### Tablica „literatura“

Glavna tablica naše baze podataka. Sadrži informacija kao što su: isbn, autor literature, naslov literature, izdavač, vrsta literature te podatak o dostupnosti literature. U ovoj tablici primarni ključ je literatura\_id i tipa je INTEGER, te se povećava svakim unosom. Tablica je ograničena sa 3 strana ključa: autor\_id, izdavac\_id i vrsta\_id.

### Tablica „posudbe“

Tablica sadržava informacije o literaturi koja je posuđena i kada je literatura vraćena u knjižnicu. Ove dvije informacije su nam potrebne da bi mogli dati informaciju koliko su trajale posudbe. Primarni ključ je posudba\_id koji je također autopovećavajući. Tablica je ograničena stranim ključem literatura\_id koji je referenciran na polje literatura\_id u tablici literatura.

### Tablica „vrste“

Sadržava informacije o vrstama literature (knjiga, časopis, seminar...). Primarni ključ ove tablice je vrsta\_id.

## 5. SQL NAREDBE ZA KREIRANJE BAZE PODATAKA

### Kreiranje baze podataka:

```
CREATE DATABASE IF NOT EXISTS `boris` ;
```

### Kreiranje tablice „autor“

```
CREATE TABLE IF NOT EXISTS `autor` (  
  `autor_id` int(11) NOT NULL AUTO_INCREMENT,  
  `autor` varchar(200) NOT NULL,  
  PRIMARY KEY (`autor_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=1
```

### Kreiranje tablice „izdavaci“

```
CREATE TABLE IF NOT EXISTS `izdavaci` (  
  `izdavac_id` int(11) NOT NULL AUTO_INCREMENT,  
  `izdavac` varchar(150) NOT NULL,  
  PRIMARY KEY (`izdavac_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=1
```

### Kreiranje tablice „literatura“

```
CREATE TABLE IF NOT EXISTS `literatura` (  
  `literatura_id` int(11) NOT NULL AUTO_INCREMENT,  
  `isbn` varchar(15) DEFAULT NULL,  
  `autor_id` int(11) NOT NULL,  
  `naslov` varchar(200) NOT NULL,  
  `izdavac_id` int(11) NOT NULL,  
  `vrsta_id` int(11) NOT NULL,  
  `dostupno` char(2) NOT NULL DEFAULT 'Da',
```

```

PRIMARY KEY (`literatura_id`),
KEY `autor_id` (`autor_id`),
KEY `izdavac_id` (`izdavac_id`),
KEY `vrsta_id` (`vrsta_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=1

```

### Kreiranje tablice „posudbe“

```

CREATE TABLE IF NOT EXISTS `posudbe` (
  `posudba_id` int(11) NOT NULL AUTO_INCREMENT,
  `literatura_id` int(11) NOT NULL,
  `datum_posudbe` date NOT NULL,
  `datum_povratka` date DEFAULT NULL,
  PRIMARY KEY (`posudba_id`),
  KEY `knjiga_id` (`literatura_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=1

```

### Kreiranje tablice „vrste“

```

CREATE TABLE IF NOT EXISTS `vrste` (
  `vrsta_id` int(11) NOT NULL AUTO_INCREMENT,
  `vrsta` varchar(100) NOT NULL,
  PRIMARY KEY (`vrsta_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin AUTO_INCREMENT=1

```

### Postavljanje ograničenja (FOREIGN KEY)

```

ALTER TABLE `literatura`
  ADD CONSTRAINT `literatura_ibfk_3` FOREIGN KEY (`izdavac_id`) REFERENCES `izdavaci`
  (`izdavac_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `literatura_ibfk_1` FOREIGN KEY (`autor_id`) REFERENCES `autor`
  (`autor_id`) ON DELETE CASCADE ON UPDATE CASCADE,

```



```
ADD CONSTRAINT `literatura_ibfk_2` FOREIGN KEY (`vrsta_id`) REFERENCES `vrste`  
(`vrsta_id`) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE `posudbe`
```

```
ADD CONSTRAINT `posudbe_ibfk_1` FOREIGN KEY (`literatura_id`) REFERENCES `literatura`  
(`literatura_id`) ON DELETE CASCADE ON UPDATE CASCADE;
```

## 6. PUNJENJE BAZE PODACIMA

### Unos podataka u tablicu „autor“

```
INSERT INTO autor (autor) VALUE ('Mato Lovrak');
```

```
INSERT INTO autor (autor) VALUE ('Stanko Paunović');
```

```
INSERT INTO autor (autor) VALUE ('Radoslav Galić');
```

```
INSERT INTO autor (autor) VALUE ('Damir Šljivac');
```

```
INSERT INTO autor (autor) VALUE ('Ivana Brlić Mažuranić');
```

### Unos podataka u tablicu „izdavaci“

```
INSERT INTO izdavaci (izdavac) VALUE ('ETFOS');
```

```
INSERT INTO izdavaci (izdavac) VALUE ('Školska Knjiga Zagreb');
```

```
INSERT INTO izdavaci (izdavac) VALUE ('ALGORITAM');
```

```
INSERT INTO izdavaci (izdavac) VALUE ('PROFIL');
```

```
INSERT INTO izdavaci (izdavac) VALUE ('Miš');
```

### Unos podataka u tablicu „vrste“

```
INSERT INTO vrste (vrsta) VALUE ('Knjiga');
```

```
INSERT INTO vrste (vrsta) VALUE ('Časopis');
```

```
INSERT INTO vrste (vrsta) VALUE ('Skripta');
```

```
INSERT INTO vrste (vrsta) VALUE ('Seminar');
```

```
INSERT INTO vrste (vrsta) VALUE ('Diplomski rad');
```

## Unos podataka u tablicu „literatura“

```
INSERT INTO literatura (isbn, autor_id, naslov, izdavac_id, vrsta_id) VALUES ('953-6254-39-5', 1, 'Vlak u snijegu', 2, 1);
```

```
INSERT INTO literatura (isbn ,autor_id, naslov, izdavac_id, vrsta_id, dostupno) VALUES ('953-175-133-1', 2, 'Elektronički sklopovi', 3, 1, 'Ne');
```

```
INSERT INTO literatura (autor_id, naslov, izdavac_id, vrsta_id, dostupno) VALUES ( 3, 'Vjerojatnost i statistika', 1, 3, 'Da');
```

```
INSERT INTO literatura (isbn ,autor_id, naslov, izdavac_id, vrsta_id) VALUES ('953-6981-53-X', 4, 'Osnove energetike', 1, 5);
```

```
INSERT INTO literatura (isbn ,autor_id, naslov, izdavac_id, vrsta_id, dostupno) VALUES ('953-6341-05-0', 5, 'Priče iz davnina', 5, 1, 'Ne');
```

## Unos podataka u tablicu „posudbe“

```
INSERT INTO posudbe (literatura_id, datum_posudbe) VALUES (2, '2009-11-11');
```

```
INSERT INTO posudbe (literatura_id, datum_posudbe, datum_povratka) VALUES (5, '2009-12-3', '2009-12-17');
```

## 7. UPITI KOJI ĆE BITI NAJČEŠĆE KORIŠTENI

### Ispis svih podataka iz tablice „literatura“:

```
SELECT * FROM literatura;
```

Upit služi za prikaz svih podataka u tablici literatura. Pošto ovakvom naredbom ne dobijemo „lijepo“ napisane podatke koristit ćemo sljedeću naredbu:

```
SELECT literatura_id AS ID, autor AS Autor, naslov AS Naslov, izdavac AS Izdavač, vrsta AS Vrsta FROM literatura JOIN (autor, vrste, izdavaci) ON (literatura.autor_id = autor.autor_id AND literatura.izdavac_id = izdavaci.izdavac_id AND literatura.vrsta_id = vrste.vrsta_id);
```

Ovim upitom dobijemo podatke prikazane kao da se nalaze u jednoj tablici. Naredba JOIN nam to omogućuje. Ovaj prikaz je puno prihvatljiviji za gledanje, pošto nemamo potrebe razmišljati koji autor se krije pod kojim brojem, kao ni o kojem se izdavaču i vrsti radi.

### Ispis svih podataka iz tablice „posudba“:

```
SELECT * FROM posudba;
```

Upitom dobijemo prikazane sve podatke iz tablice posudba.

```
SELECT posudba_id AS ID, autor AS Autor, naslov AS Naslov, datum_posudbe AS 'Datum posudbe', datum_povratka AS 'Datum povratka' FROM posudbe JOIN (autor, literatura) ON (posudbe.literatura_id = literatura.literatura_id AND literatura.autor_id = autor.autor_id);
```

Ovim upitom dobijemo lijepo prikazane podatke o posudbama knjiga.

### Trajanje posudbe u danima:

```
SELECT posudba_id AS ID, autor AS Autor, naslov AS Naslov, DATEDIFF(datum_povratka, datum_posudbe) AS 'Broj dana posudjena' FROM posudbe JOIN (autor, literatura) ON (posudbe.literatura_id = literatura.literatura_id AND literatura.autor_id = autor.autor_id);
```

Gornji upit vraća nam podatke o autoru, naslovu i broju dana koliko je određena literatura posuđena. Naredba koja nam vraća broj proteklih dana – DATEDIFF.

### Literatura koja nije dostupna:

```
SELECT literatura_id AS ID, autor AS Autor, naslov AS Naslov, izdavac AS Izdavač, vrsta AS Vrsta FROM literatura JOIN (autor, vrste, izdavaci) ON (literatura.autor_id = autor.autor_id AND literatura.izdavac_id = izdavaci.izdavac_id AND literatura.vrsta_id = vrste.vrsta_id AND literatura.dostupno = 'Ne');
```

Gornji upit vraća nam informaciju o literaturi koja trenutno nije dostupna.

### **Popis svih izdavača:**

```
SELECT * FROM izdavaci;
```

Upit vraća sve izdavače koji su pohranjeni u tablici „izdavaci“.

### **Popis svih autora:**

```
SELECT * FROM autor;
```

Upit vraća sve autore koji su pohranjeni u tablici „autor“.

### **Izmjena podatka u tablici „posudbe“:**

```
UPDATE posudbe SET datum_povratka='2009-12-03' WHERE posudba_id= '1';
```

Gornji upit služi nam da bismo upisali datum kad je netko vratio knjigu.

### **Brisanje literature:**

```
DELETE FROM literatura WHERE literatura_id='2';
```

Gornji upit briše jedan red podataka iz tablice literatura koji ima id = 2.

## 8. ZAKLJUČAK

Rješavanjem ovoga projektnog zadatka upoznali smo se sa osnovama baza podataka. Uvidjeli smo kako u praksi radi mysql baza (XAMPP). Naučili smo koje su osnovne naredbe SQL jezika kao što su SELECT, INSERT, UPDATE, JOIN, AS, DELETE ...

Uvidjeli smo kako se postavljaju ograničenja na unos podataka korištenjem stranih ključeva.

XAMPP sam koristio jer je vrlo mali i jednostavan program sa svim potrebnim dodacima. Ponajviše je zanimljiv jer ima odličan Query View koji je vrlo jednostavan. Nije potrebno navesti što da se prikaže, samo klikom miša na ponuđene opcije dobije se rezultat.

Marenić