

SQL – Opis podataka



Neđeljko Lekić

Irena Orović

www.etf.ac.me

U OVOJ LEKCIJI

■ SQL

- SQL jezik
- SQL, relacioni model i E/R diagram
- CREATE TABLE
 - Kolone
 - Primarni ključevi
 - Spoljnji ključevi
- DROP TABLE
- ALTER TABLE

SQL

- SQL - Structured Query Language

- ANSI Standardi

- SQL-1989
- SQL-1992 (SQL2)
- SQL-1999 (SQL3)
- SQL-2003
- SQL-2006
- SQL-2008

- Različiti DBMS koriste različite SQL

SQL

■ SQL obezbjeđuje

- Jezik za opis podataka (DDL - data definition language)
- Jezik za rukovanje podacima (DML - data manipulation language)
- Jezik za kontrolu podataka (DCL - data control language)

■ Osim toga SQL

- se može koristiti iz drugih programskih jezika.
- Može se proširiti u cilju obezbjeđenja uobičajenih programskih konstrukcija (kao što su: if-then, petlje, promjenljive, itd.)

SQL - NAPOMENE

- SQL (uglavnom) nije case-sensitive, ali u cilju naglašavanja, SQL ključne riječi je preporučljivo pisati velikim slovima.
- Nizovi u SQL-u navode se pod jednostrukim navodnicima:
`'Ja sam niz'`
- Citati unutar niza navode se pod dvostrukim navodnicima.
- Prazan niz: ''

NEPROCEDURALNO PROGRAMIRANJE

- SQL je deklarativan (neproceduralni) jezik
 - Proceduralni – navodi što kompjuter tačno treba da uradi.
 - Neproceduralni – opisuje zahtijavani rezultat (ne način kako to izračunati).
- Primjer: Neka je data baza podataka sa sljedećim tabelama:
 - Student sa atributima ID, Ime, Adresa.
 - Predmeti sa atributima Šifra, Naziv.
 - Upis sa atributima ID, Šifra.
- Dobiti listu studenata koji su odabrali predmet 'Baze podataka'.

PROCEDURALNO PROGRAMIRANJE

```
Set P to be the first Predmet Record      /* Nalazenje sifre predmeta      */
Code = ''                                  /* 'Baze podataka'           */
While (P is not null) and (Code = '')
    If (P.Naziv = 'Baze podataka') Then
        Code = P.Code
Set P to be the next Predmet Record
Set IMENA to be empty                      /* Lista imena studenata */
Set S to be the first Student Record
While S is not null                        /* Za svakog studenta...   */
    Set U to be the first Upis Record
    While U is not null                    /* Za svaku instacu Upisa... */
        If (U.ID = S.ID) And              /* Ako je student          */
            (U.Code = Code) Then          /* upisao Baze podataka    */
            IMENA = IMENA + S.IME         /* dodati ga u listu      */
        Set U to be the next Upis Record
    Set S to be the next Student Record
Return IMENA
```

NEPROCEDURALNO PROGRAMIRANJE

```
SELECT Ime FROM Student, Upis
WHERE (Student.ID = Upis.ID)
AND (Upis.Sifra =
      (SELECT Sifra FROM Predmeti
        WHERE Naziv = 'Baze podataka'))
```


SQL, RELACIONI MODEL, E/R DIZAJN

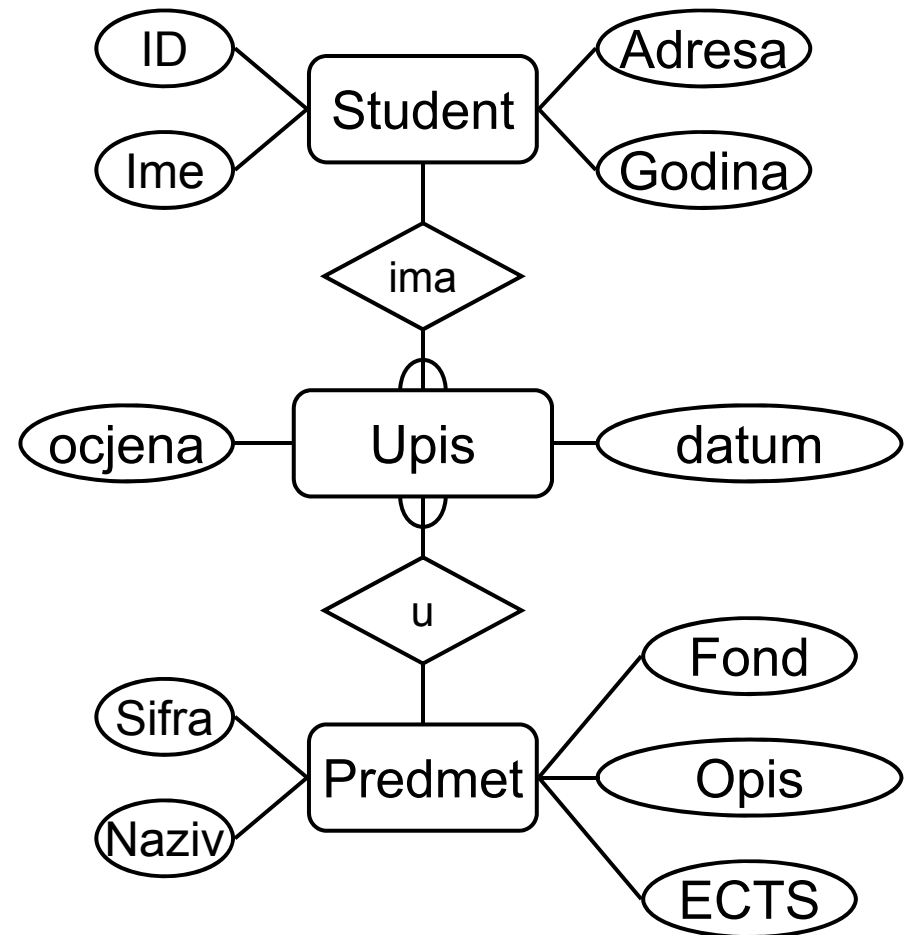
- SQL je zasnovan na relacionom modelu
 - Baze podataka koje podržavaju SQL su najčešće relacione baze.
- E/R dizajn se može implementirati SQL-om
 - Enteti, atributi, i veze mogu se izraziti terminima SQL-a.
 - M:M veze su problem i treba ih ukloniti.

RELACIJE, ENTITETI, TABELE

Relacioni model	E/R Diagram	SQL
Relacija n-torka Atribut Spoljnji ključ Primarni ključ	Entitet Instanca Atribut M:1 veza	Tabela Red Kolona ili polje Spoljnji ključ Primarni ključ

IMPLEMENTIRANJE E/R DIZAJNA

- Za dati E/R dizajn
 - Entiteti postaju SQL tabele.
 - Atributi entiteta kolone odgovarajućih tabela.
 - Veze mogu biti predstavljene spoljašnjim ključem.



ENTITETI I ATRIBUTI

- Svaki entitet postaje tabela u bazi podataka.
 - Često se za ime tabele uzima ime entiteta.
 - Atributi postaju kolone tabele sa istim imenima.

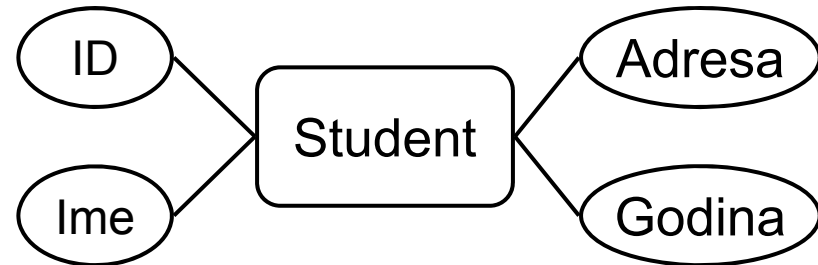


Tabela Student,
sa kolonama:
ID, Ime, Adresa i Godina.

KREIRANJE TABELE

CREATE TABLE

```
<name> (  
    <col-def-1>,  
    <col-def-2>,  
        :  
    <col-def-n>,  
    <constraint-1>,  
        :  
    <constraint-k>)
```

■ Neohodno je navesti:

- ime tabele
- listu definicija kolona
- listu ograničenja (npr. ključevi)

DEFINISANJE KOLONA

```
<col-name> <type>  
[NULL|NOT NULL]  
[DEFAULT <val>]  
[constraint-1 [,  
constraint-2 [,  
...]]]
```

- Svakoj koloni se zadaje ime i tip podatka koji će sadržavati
- Najčešći tipovi:
 - INT
 - FLOAT
 - CHAR (n)
 - VARCHAR (n)
 - DATE

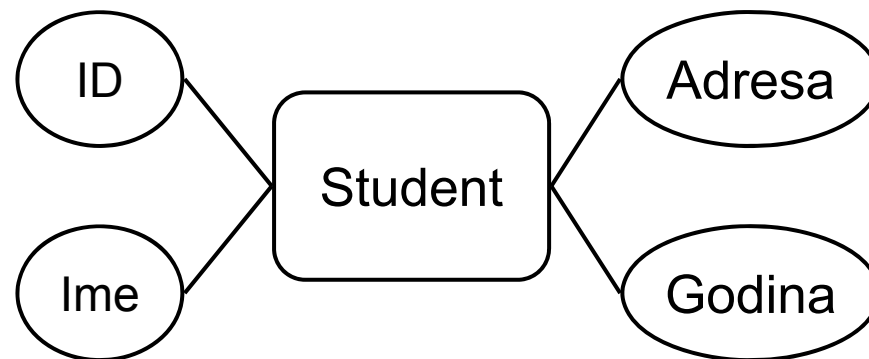
DEFINISANJE KOLONA

- Kolone se mogu navesti kao **NULL** ili **NOT NULL**.
- **NOT NULL** kolone ne mogu imati **NULL** vrijednost.
- Ako naredbom ništa nije navedeno za kolone, podrazumijeva se **NULL**.
- Kolonama se može dodijeliti podrazumijevana (default) vrijednost.
- Samo se navede ključna riječ **DEFAULT** i zatim vrijednost, primjer:

`broj INT DEFAULT 0`

PRIMJER

```
CREATE TABLE Student (  
  studID INT NOT NULL,  
  studIme VARCHAR(50) NOT NULL,  
  studAdresa VARCHAR(50),  
  studGodina INT DEFAULT 1)  
ENGINE=INNODB;
```



OGRANIČENJA

CONSTRAINT

<name>

<type>

<details>

■ Najčešći <type>:

PRIMARY KEY

UNIQUE

FOREIGN KEY

INDEX

- Ograničenja imaju ime – ograničenja pristupa zahtijevaju ime, ali neka druga ne.
- Ograničenja koja se odnose na jednu kolonu, mogu se uključiti u definiciju te kolone.

OGRANIČENJA: PRIMARY KEY

- Primarni ključ se definiše kroz ograničenja.
- **PRIMARY KEY** ograničenje uključuje **UNIQUE** i **NOT NULL** ograničenja.

- Za primarni ključ **<details>** je lista kolona koje sačinjavaju ključ.

```
CONSTRAINT <name>  
PRIMARY KEY  
(col1, col2, ...)
```

OGRANIČENJA: UNIQUE

- Isto kao **PRIMARY KEY** , grupi kolona se može zadati **UNIQUE** ograničenje
- Ovime se definiše kandidat za ključ tabele.

<details> za **UNIQUE** ograničenje je lista kolona koja predstavlja kandidat za ključ.

```
CONSTRAINT <name>  
UNIQUE  
(col1, col2, ...)
```

PRIMJER

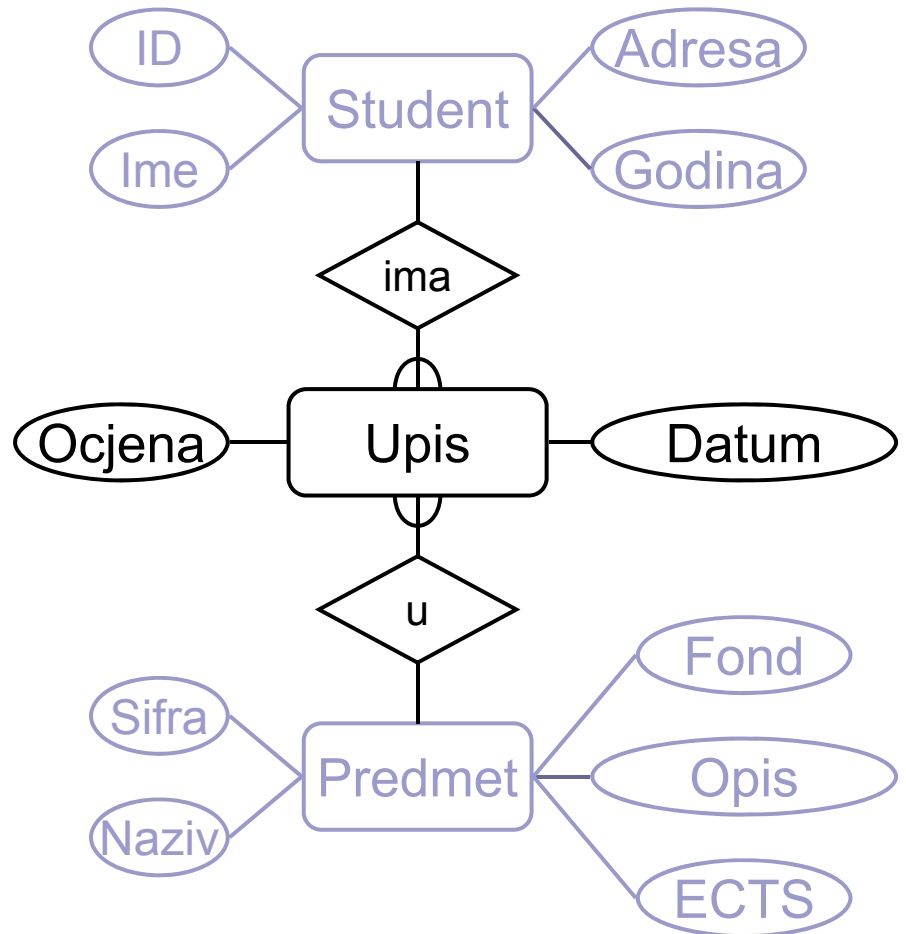
```
CREATE TABLE Student (  
    studID INT NOT NULL,  
    studIme VARCHAR(50) NOT NULL,  
    studAdresa VARCHAR(50),  
    studGodina INT DEFAULT 1,  
    CONSTRAINT pkStudent  
        PRIMARY KEY (studID)  
) ENGINE=INNODB;
```

PRIMJER

```
CREATE TABLE Predmet (  
    predSifra INT NOT NULL,  
    predNaziv VARCHAR(70) NOT NULL,  
    predFond VARCHAR(50),  
    predOpis VARCHAR(100),  
    CONSTRAINT pkPredmet  
        PRIMARY KEY (predSifra)  
)
```

VEZE

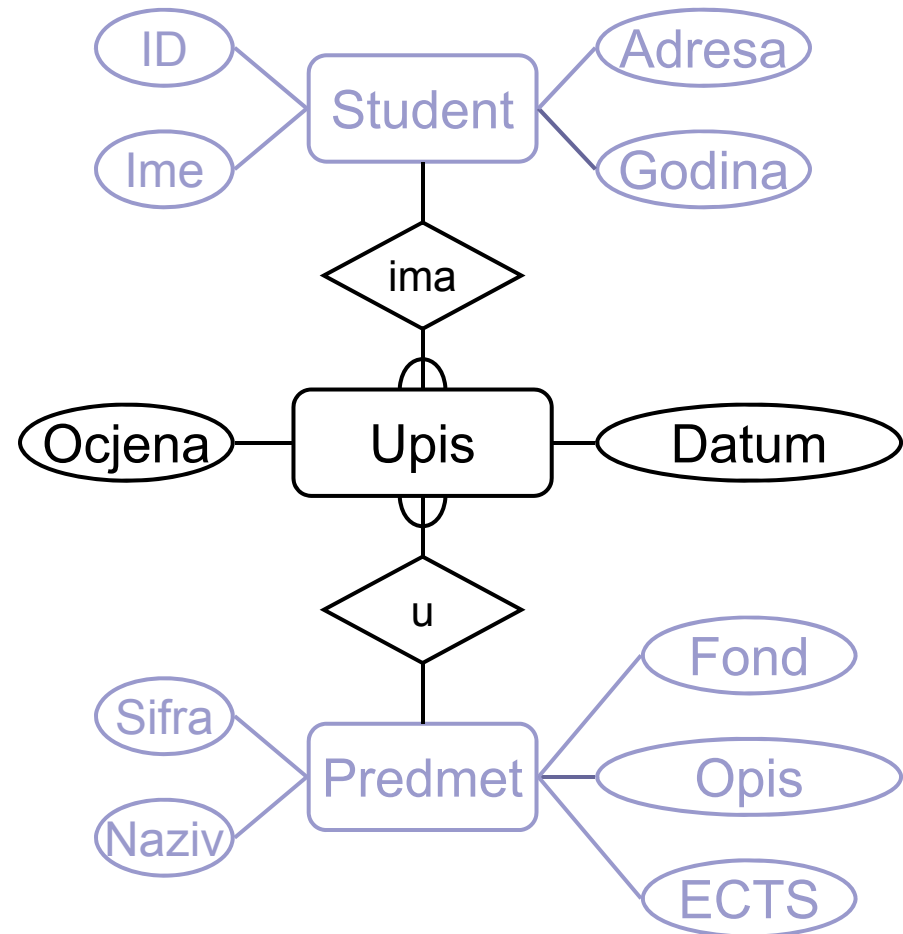
- Zavisno od tipa:
 - 1:1 se obično ne koristi ili se može tretirati kao specijalan slučaj M:1 veze
 - M:1 se predstavlja kao spoljašnji ključ iz M-strane na 1.
 - M:M se dijeli u dvije M:1 veze.



PREDSTAVLJANJE VEZE

■ Tabela Upis:

- Imaće kolone za attribute Ocjena i Datum.
- Imaće spoljašnji ključ za tabelu Student, za 'ima' vezu.
- Imaće spoljašnji ključ za tabelu Predmet, za 'u' vezu.



OGRANIČENJA: FOREIGN KEYS

- Spoljašnji ključevi (Foreign Keys - FK) takođe se definišu kao ograničenje.
- Morate da imate:
 - kolone koje čine spoljašnji ključ,
 - referenciranu tabelu,
 - kolone koje su referancirane spoljašnjim ključem.

```
CONSTRAINT <name>  
    FOREIGN KEY  
    (col1, col2, ...)  
    REFERENCES  
    <table>  
    [ (ref1, ref2, ...) ]
```

- Ako se spoljašnji ključ referencira na primarni ključ **<table>** ne treba se navoditi lista kolona.

PRIMJER

```
CREATE TABLE Upis (  
    studID INT NOT NULL,  
    predSifra INT NOT NULL,  
    Ocjena CHAR(1),  
    Datum DATE,  
    CONSTRAINT uPK  
        PRIMARY KEY (studID, predSifra),  
    CONSTRAINT fkStud FOREIGN KEY (studID)  
        REFERENCES Student (studID),  
    CONSTRAINT fkPred FOREIGN KEY (predSifra)  
        REFERENCES Predmet (predSifra))
```

BRISANJE TABELA

- Za brisanje tabele koristi se:

```
DROP TABLE  
[IF EXISTS]  
<name>
```

- Primjer:

```
DROP TABLE Predmet
```

- **BUDITE PAŽLJIVI** sa **DROP**:

- Izbrisaćete i sve podatke iz tabele.
- Obično nećete biti pitani za potvrdu.
- Nama jednostavnog načina za opoziv.

IZMJENE U STRUKTURI TABELA

- Ponekad je potrebno izmijeniti strukturu postojećih tabela.
 - Jedan način je izbrisati je (DROP) i ponovo kreirati.
 - To se ne preporučuje, već umjesto toga postoji komanda ALTER TABLE.
- ALTER TABLE može
 - Dodati novu kolonu.
 - Ukloniti postojeću kolonu.
 - Dodati novo ograničenje.
 - Ukloniti postojeće ograničenje.
 - Iznijeliti osobine postojeće kolone.

BRISANJE I DODAVANJE KOLONA

Za dodavanje i brisanje kolone koristi se:

```
ALTER TABLE <table>  
  ADD COLUMN <col>
```

```
ALTER TABLE <table>  
  DROP COLUMN <name>
```

Primjeri:

```
ALTER TABLE Student  
  ADD COLUMN  
  Stepen VARCHAR(50)
```

```
ALTER TABLE Student  
  DROP COLUMN Stepen
```

IZMJENA OSOBINA KOLONA

- Promjena imena i definicije:

```
ALTER TABLE <table>  
CHANGE COLUMN <old  
col-name> <new col-  
name> <type>
```

- Promjena samo definicije:

```
ALTER TABLE <table>  
MODIFY <col-name>  
<type>
```

- PRIMJERI:

```
ALTER TABLE Student  
CHANGE COLUMN studIme  
sPIME VARCHAR(50)  
NOT NULL
```

```
ALTER TABLE Student  
MODIFY sPIME  
VARCHAR(40) NOT NULL
```

DODAVANJE OGRANIČENJA

Dodavanje
ograničenja:

```
ALTER TABLE  
<table>  
ADD CONSTRAINT  
  <definition>
```

Examples

```
ALTER TABLE  
Predmet  
ADD CONSTRAINT  
ck UNIQUE  
(predNaziv)
```

BRISANJE OGRANIČENJA

- U MySQL-u prilikom brisanja ograničenje mora se navesti tip ograničenja:
 - PRIMARY KEY za primarne ključeve;
 - FOREIGN KEY za spoljnje ključeve;
 - INDEX za UNIQUE ograničenje.

- PRIMJER:

```
ALTER TABLE
```

```
  Predmet
```

```
  DROP INDEX ck
```

```
ALTER TABLE
```

```
  Predmet
```

```
  DROP PRIMARY KEY
```

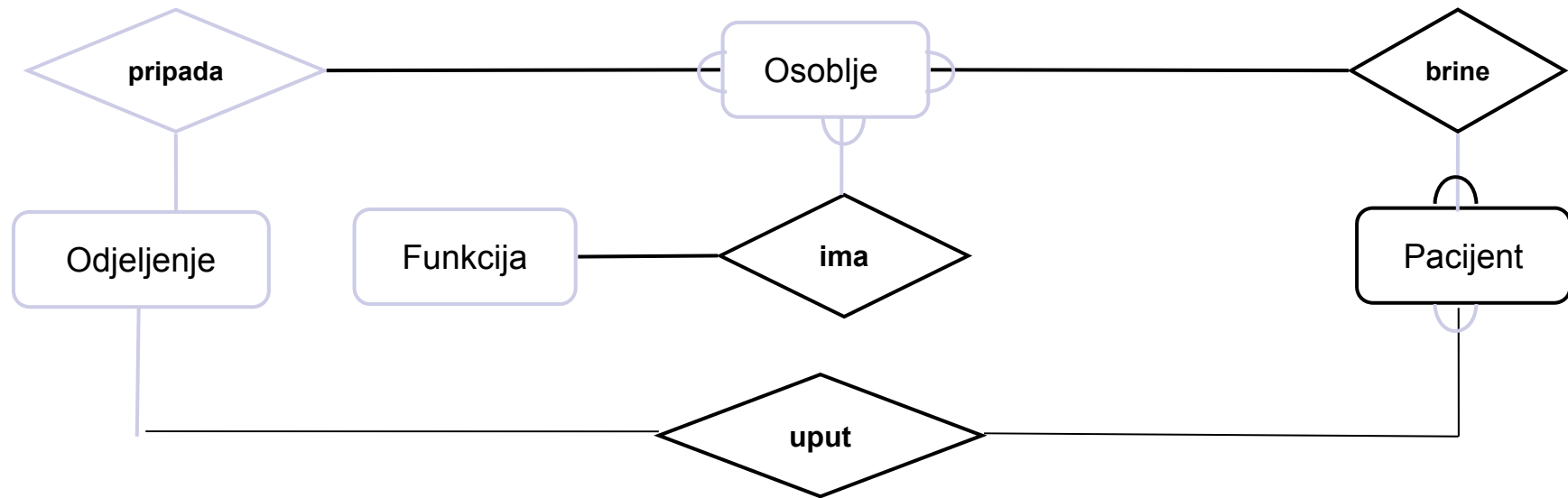
```
ALTER TABLE upis
```

```
  DROP FOREIGN KEY
```

```
  fkStud
```

ZA VJEŽBU

Na osnovu modela sa slike, kreirati bazu podataka.



Atributi postojećih entiteta su:

OSOBLJE(osID#, Ime, Prezime, DatRodj, DatZap, odjID, funID)

FUNKCIJA(funID#, Naziv)

PACIJENT(pctID#, Ime, Prezime, DatRodj, DatPrijem, odjID)

ODJELJENJE(odjID#, Naziv)

Realizovati vezu *brine*.