



# Baze podataka 1

## Vežbe 1



Srđan Nikolić

Institut za matematiku i informatiku, Prirodno-matematički fakultet, Univerzitet u Kragujevcu

- Šta se dešava sa podacima koji se nalaze u RAM memoriji kada se ugasi računar?
- Kako možemo podatke da sačuvamo trajno?
- Kako možemo da filtriramo podatke?

Studenti prve godine informatike su došli na ideju da naprave program pomoću koga bi bilo moguće voditi evidenciju o knjigama u fakultetskoj biblioteci.

Zaključili su da bi trebalo čuvati **informacije** o:

- **autorima** – neki spisak, bar, imena bi morao da bude dostupan
- **knjigama** – ISBN, naziv, autori.

Od funkcionalnosti koje bi program trebao da zadovolji odlučili su se za sledeće:

- dodavanje, brisanje autora, kao i izmena imena  $f$
- brisanje i dodavanje knjige u evidenciju  $f$
- dodavanje podataka o tome koji je autor koju knjigu napisao, izmena i brisanje istih  $f$
- pretraga autora po knjigama – zadavanjem imena ili ISBN-a knjige dobija se spisak autora  $f$
- pretraga knjiga po naslovu sa i bez zadavanja autora  $f$
- pretraga autora po imenu sa spiskom knjiga koje je napisao

## ZADATAK ZA VAS:

- Osmislite sami strukture, odlučite kako bi čuvali podatke u toku rada programa
- Da li podatke držati u dinamičkim strukturama dok program radi pa ih na kraju rada izmenjene upisati u datoteke ili učitavati po potrebi
- u zavisnosti od prethodne dve odluke napravite spisak f-ja sa skicom njihove signature i opisom šta bi radile



```
C:\WINDOWS\system32\cmd.exe
```

```
Izaberite zeljenu opciju:
```

1. Dodaj autora
2. Izbrisi autora
3. Izmeni ime autora
4. Dodaj knjigu
5. Izbrisi knjigu
6. Izmeni knjigu
7. Prikazi autore po naziv knjige
8. Prikazi autore po ISBN-u knjige
9. Prikazi knjige po naslovu
10. Prikazi knjige po naslovu i imenu autora
11. Prikazi sve knjige koje je napisao neki autor

```
#####
```

```
Unesite opciju:
```

```
#####
```

```
Unesite opciju: 7
```

```
Unesite naziv knjige: Head First Java
```



```
C:\WINDOWS\system32\cmd.exe
Izaberite zeljenu opciju:
1. Dodaj autora
2. Izbrisi autora
3. Izmeni ime autora
4. Dodaj knjigu
5. Izbrisi knjigu
6. Izmeni knjigu
7. Prikazi autore po naziv knjige
8. Prikazi autore po ISBN-u knjige
9. Prikazi knjige po naslovu
10. Prikazi knjige po naslovu i imenu autora
11. Prikazi sve knjige koje je napisao neki autor

#####

Unesite opciju: 7
Unesite naziv knjige: Head First Java

Rezultat pretrage: Kathz Bierra, Bert Batas

#####

Unesite opciju: 1
```

Nakon nekog vremena želje su počele da rastu pa se i kompleksnost problema menjala, npr.

- trebalo je ubaciti i godinu izdavanja
- broj zapisa je porastao na 200 000 različitih knjiga
- pojavili su se različiti autori sa istim imenima
- program su počeli da koriste i službenici kojima preciznost nije jača strana pa su više puta unosili podatke o istoj knjizi, a ime jednog istog autora zapisivali u različitim knjigama na različit način

## KAKO BI TO IZGLEDALO SA BAZOM



U bazi će podaci biti čuvani u tabelama, npr. ovakvim

Autor:

	autorID	autor
	1	T. Codd
	2	P. Norvig
▶*	NULL	NULL

Knjiga:

	ISBN	naslov
	N0-01	Artificial Intellige...
	N0-02	Relational Datab...
▶*	NULL	NULL

KA:

	autorID	ISBN
	1	N0-02
	2	N0-01
▶*	NULL	NULL

Tabele su deo baze podataka Biblioteka o kojoj "brine" sistem za upravljanje bazama podataka - SUBP (npr. SQL Server, MySQL, Oracle i sl.). Isti sistem mora da vam omogući mehanizam upisa, brisanja i dodavanja novih podataka kao i dobijanja odgovora na pitanja koja se tiču podataka upisanih u bazu. "Razgovor" sa SUBP-om o tome šta želite da uradi se odvija na posebnom jeziku Structured Query Language – SQL, čije komande SUBP jeste u stanju da procesira.

Umesto pisanja f-ja koje će manipulirati podacima po datotekama posao se svodi na zadavanje SQL komandi. Neki primeri su dati u tabeli:

Dodavanje autora	<code>insert autori values (3, 'T. Berners-Lee')</code>
Brisanje autora	<code>delete autori where autorID=3</code>
Izmena autora	<code>update autori set autor='T.B.Lee' where autorID=3</code>
Pretraga autora po knjigama – zadavanjem ISBN-a knjige dobija se spisak autora	<code>select * from autori where autorID in (select autorID                   from KA                   where ISBN='N0-01')</code>
Dodavanje novog podatka u tabelu knjige	<code>ALTER TABLE dbo.Knjige ADD                   godina numeric(4, 0) NULL</code>



# Informacioni sistem sajta "Polovni automobili"

← → C www.polovniautomobili.com

ŠTA JE  
**BOLJE**  
ZA MENE  
  
**MANJA  
RATA**  
  
ILI KRAĆI  
**ROK**  
OTPLATE?

**POLOVNI  
AUTOMOBILI**  
kupi dobro, prodaj brzo



PONUDA PREKO 1.200 OBJEKATA  
U SRBIJI I CRNOJ GORI  
Pronađi privatni smeštaj



Brza pretraga Ponuda vozila Kupujem Prodajem Auto osiguranje Saveti i Vesti Moj profil Postavite oglas Ponuda za korisnike

Pretraživač Marke 133980 oglasa - 50042 pripada kategoriji "Putnička vozila"

Pretraga vozila:

Marka: Sve marke Model: Svi modeli Cena do: €

Godina proizvodnje (od - do): bilo koja bilo koja Karoserija: Odaberi Gorivo: Odaberi

Region: Odaberi Pogledaj: Polovna i nova vozila

Kredit  Lizing

**PRETRAGA** **DETALJNA PRETRAGA**

Oglasi postavljeni u poslednja 24 časa: detaljno

## POLOVNA VOZILA

 6.200 € Fiat Bravo 2007 god	 7.500 € Jeep Cherokee 2006 god	 5.990 € Audi A4 2005 god	 37.850 € Nissan Navara 2016 god
 7.490 € Peugeot Partner 2010 god	 6.199 € Fiat 500 2007 god	 2.300 € Alfa Romeo 147 2003 god	 17.990 € Fiat 500X 2015 god

## NOVA VOZILA

REKLAMA

Web programiranje - PHP osno...  
25000 din cena  
Organizator: OKSchool  
Mesto: Online - putem interneta

Kurs iOS programiranja  
Cena na upit  
Organizator: eCampus IT kursevi  
Mesto: Beograd

**KURSEVI.COM**

- Saveti i vesti Serija Prvi Servis
- NOVO Istorija: Opel Corsa 0
  - NOVO Zašto je toliko problem dati „migavac“? 19
  - Ugradnja elektro-podizača stakala 2
  - Kako se paralelno parkirati? 3
  - Kako dobiti ili ukloniti ležećeg policajca? 9
- Ostali saveti i vesti

Oglasi na e-mail

Prijavite se na našu mejling listu. Pratile preko mejla najnovije oglase na našem sajtu.

Unesite vašu e-mail adresu

REKLAMA

Rukovaoci mašinama i uređaji...

KAKO  
UZETI  
POTPUNO  
  
ONLINE  
KEŠ  
  
KREDIT?

BESPLATAN POZIV  
0800 344 344



Obezbeđivanje jedinstvenosti i brze pretrage po, npr. ISBN-u, je moguće obezbediti postavljanjem ograničenja na to polje, npr.

```
ALTER TABLE dbo.KA ADD CONSTRAINT  
IX_KA UNIQUE NONCLUSTERED ( autorID ) ON [PRIMARY]
```

Tada će SUBP voditi računa o jedinstvenosti. Pored ove postoji puno različitih vrsta ograničenja o kojima sistem može voditi računa, a o kojima bi inače brinule vaše f-je (na prethodnom primeru za upis knjiga - ako uopšte imate želju da zaštitite službenike od nenamernih grešaka, odnosno f-je pretrage bi morale da imaju vrlo brze algoritme koji traže i pogodne strukture u kojima bi se traženi podaci nalazili) .

Dakle, pojednostavljeno predstavljeno vaše je da:  $f$

- osmislite kako će podaci biti grupisani (po tabelama),  $f$
- popišete ograničenja,  $f$
- naučite da govorite SQL jezikom i ono što ste osmislili kažete SUBP-u da napravi, održava i  $f$
- da kada vam je potrebno postavite i po neko pitanje istom

**Kako će podaci fizički biti zapisani, kako će biti obezbeđen odgovor na vaša pitanja, kako će vaša ograničenja biti obezbeđena i sl. nije vaš problem.**

Relacioni model ima sledeće bitne karakteristike:

- Struktura modela je veoma jednostavna (skup tabela)
- Operacije koje iz skupa datih tabela generišu izlaz su jednostavne i lako prihvatljive
- Tabele se mogu definisati kao matematička relacija i zatim iskoristiti bogata teorijska osnova odgovarajućeg matematičkog aparata
- Minimalna redudansa podataka
- Redosled kolona i redova ne utiče na informacioni sadržaj tabele
- Svaki red se može jednoznačno odrediti, što znači da u tabeli ne mogu postojati dva identična reda

Autor:

	autorID	autor
	1	T. Codd
	2	P. Norvig
▶*	NULL	NULL

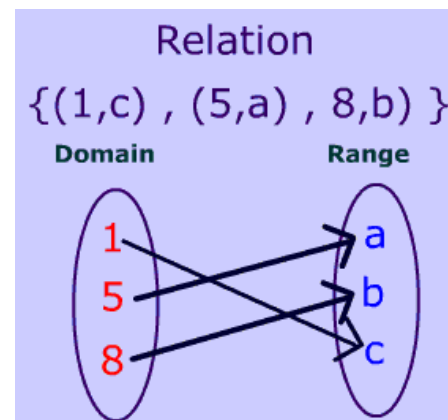
Knjiga:

	ISBN	naslov
	N0-01	Artificial Intellige...
	N0-02	Relational Datab...
▶*	NULL	NULL

KA:

	autorID	ISBN
	1	N0-02
	2	N0-01
▶*	NULL	NULL

- Dekartov proizvod i matematička relacija?
- Kartezijanski (Dekartov proizvod) – Neka je data kolekcija skupova  $D_1, D_2, \dots, D_n$  (ne neophodno različitih), Kartezijanski proizvod ovih  $n$  skupova  $D_1 \times D_2 \times \dots \times D_n$  je skup svih mogućih  $n$ -torki  $\langle d_1, d_2, \dots, d_n \rangle$  tako da je  $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$   
**Primer:  $A=\{1,5,8\}, B=\{a,b,c\}$**
- Relacija definisana na  $n$  skupova je podskup Dekartovog proizvoda tih  $n$  skupova.





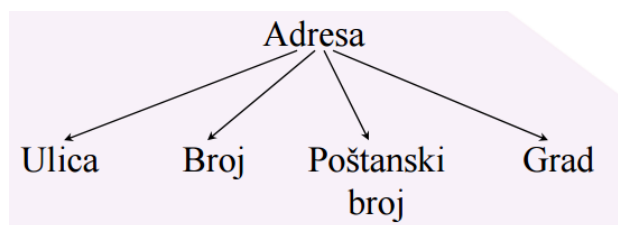
- **Domen relacije.** Skupovi  $D_1, D_2, \dots, D_n$  se nazivaju domenima relacije  $R$ .
- **Stepen relacije.** Broj domena na kojima je definisana neka relacija se naziva stepen relacije. Razlikujemo:
  - Unarne
  - Binarne
  - N-arne
- **Kardinalnost relacije** je broj  $n$ -torki u relaciji.



- **Atribut relacije.** Imenovani domen sa imenom koje definiše ulogu domena u relaciji. Atributi relacije **STUDENT** su **BRIND**, **IME** i **STAROST**.
- Koncept atributa omogućava predstavljanje relacije kao tabele.
- Uslovi koje tabela mora da zadovolji da bi bila relacija:
  - Ne postoje duplikati vrsta tabele
  - Redosled vrsta nije značajan
  - Redosled kolona nije značajan



- Atribut može biti:
  - Prost - ne može se rastavljati na delove bez gubitka svakog značenja – atomska vrednost.
  - Složen – sastoji se od više prostih atributa. Može se rastaviti na jednostavnije.



- Skup svih mogućih vrednosti nekog atributa  $A_i$  naziva se domenom tog atributa i označava se sa  $D_i$ .
- Domen - tip podataka u programiranju
- Jedan domen za više atributa. Obrnuto ne.
- Primeri:
  - Atribut: Visina (cm)  
D1: skup celih pozitivnih brojeva
  - Atribut: Boja  
D3: {"žut", "crven", "zelen", "plav"}



- Šema relacije  $R$  je konačan skup atributa  $\{A_i\}$  i konačan skup  $\{O\}$  ograničenja nad vrednostima tih atributa
- Ograničenja: atributi ne mogu uzimati bilo koje vrednosti
- Podrazumeva se da kada su zadati atributi, zadati su i njihovi domeni
- Bitne osobine šeme relacije:
  - Nazivi atributa moraju biti različiti
  - Redosled atributa nije bitan
  - Šema relacije mora da sadrži bar jedan atribut
- Šemom relacije se predstavljaju svojstva klase objekata ili veza nekog sistema

Terminologija - RBP	Terminologija - tabele
Relaciona BP	Skup tabela
Relacija	Pojedinačna tabela
Atribut	Naziv kolone u tabeli
Instanca (n-torka)	Red podataka u tabeli
Kardinalnost relacije	Broj redova u tabeli
Stepen relacije	Broj kolona u tabeli
Domen	Skup dozvoljenih vrednosti za podatke u kolonama

Autor:

	autorID	autor
	1	T. Codd
	2	P. Norvig
►*	NULL	NULL

Knjiga:

	ISBN	naslov
	N0-01	Artificial Intellige...
	N0-02	Relational Datab...
►*	NULL	NULL

KA:

	autorID	ISBN
	1	N0-02
	2	N0-01
►*	NULL	NULL



- Jedan ili više atributa jedinstveno identifikuju jednu n-torku
- Ključ relacije  $R$  je takva kolekcija  $K$  njenih atributa koja zadovoljava dva sledeća uslova:
  - Osobina jedinstvenosti
  - Osobina neredudantnosti
- Kolekcija atributa  $K$  koja zadovoljava **samo osobinu jedinstvenosti** naziva se **nadključ relacije**

Data je relacija **DNEVNIK(ucenik, predmet, ocena, datum )** sledećom ekstenzijom:

Ucenik	Predmet	Ocena	Datum
Petar	Analiza	5	1.3.2015.
Petar	Numerička matematika	4	2.3.2015.
Lazar	Algebra	3	1.3.2015.

Ako imaš u vidu **isključivo** datu ekstenziju ispitaj osobine jedinstvenosti i neredudantnosti na datim kolekcijama i navedi da li neka od kolekcija poseduje osobine ključa ili nadključa. **Svaki odgovor obrazložiti!**

- a) (ucenik, predmet)
- b) (ocena, datum)

Data je relacija sledećom ekstenzijom:

Radnik	Projekat	Zadatak	Broj_dana
R1	pr2	z2	10
R2	pr1	z1	45
R3	pr2	z2	18
R4	pr2	z2	25
R5	pr1	z1	30

Navedene kolekcije atributa imaju osobine ključa, nadključa ili ni jedno ni drugo?

**Dati obrazloženje odgovora!!!**

- a) (zadatak, broj\_dana)
- b) (projekat, zadatak)
- c) (radnik)

Neka je data relacija

**promet(barKod, datumPrometa, kupac, kolicina)**

Ako je u datoj relaciji ključ

1. PK(barKod, kupac)
2. PK(barKod, datumPrometa)

Odgovori na sledeća pitanja:

- a) Koliko najviše različitih kupaca može da kupi isti proizvod u jednom danu?
- b) Koliko najviše različitih proizvoda jedan kupac može kupiti u istom danu?

Neka je data relacija:

**osoba(matBroj, ime, prezime, adresa)**

Posmatrajmo relaciju

**roditelj\_dete(matBrojRod, matBrojDete)**

Ako je ključ relacije roditelj\_dete:

1. PK(matBrojRod)
2. PK(matBrojDete)
3. PK(matBrojRod, matBrojDete)

Odgovori na sledeća pitanja uz obrazloženje:

- a) Da li je moguće upisati odnos gde je osoba sama sebi roditelj?
- b) Da li jedan roditelj može imati više dece?
- c) Da li se o jednom detetu mogu čuvati informacije o oba roditelja?
- d) Da li je moguće upisati odnos gde je osoba sama sebi deda (baba 😊)?

Data je relacija **R(Radnik, Projekat, Zadatak, Broj\_dana)**. Ako je primarni ključ

1. PK(Radnik, Projekat)
2. PK(Projekat, Zadatak)

Odgovoriti na sledeća pitanja:

- a) Da li radnici mogu raditi na više projekata?
- b) Koliko radnika može raditi na jednom projektu? Ima li nekih posebnih ograničenja?

**OBAVEZNO DATI PRIMER EKSTENZIJE, OBRAZLOŽENJE I NAVESTI OGRANIČENJA (AKO POSTOJE)!!!**