



# Sadržaj

- 1 O kursu
  
- 2 Baze podataka
  
- 3 Relacioni model
  
- 4 SQL

# Sadržaj

1 O kursu

2 Baze podataka

3 Relacioni model

4 SQL

# Sadržaj kursa

- Uvod u baze podataka. Osnovni pojmovi.
- Relacioni model podataka. Upitni jezik SQL.
- Arhitektura sistema baza podataka
- Relaciona algebra i relacioni račun
- Integritet i bezbednost u relacionim bazama podataka
- Sigurnost podataka i autorizacija
- Funkcionalne zavisnosti
- Dodatne mogućnosti upitnog jezika SQL
- Transakcije i konkurentnost
- Optimizacija
- Administracija

# Praktični primeri

- SUBP IBM DB2
- mala studentska baza - uvodni primeri na predavanjima
- studentska baza **stud2020**

# Obaveze

Završni praktično-teorijski ispit:

- praktični deo: 40 poena
- teorijski deo: 60 poena

Preduslov za polaganje završnog ispita je osvojeno najmanje 50% poena na praktičnom i 50% poena na teorijskom delu ispita.

# Literatura

- C.J.Date: An Introduction to Database Systems, VIII ed, Addison Wesley Inc, 2004.
- C.J.Date, SQL and Relational Theory: How to Write Accurate SQL Code, O'Reilly Media, Inc, 2nd Edition, 2011
- pisani materijal koji prati predavanja

# Literatura za prvu sedmicu

- C.J.Date: An Introduction to Database Systems, VIII ed, Addison Wesley Inc, 2004.  
Poglavlja 1 i 3
- pisani materijal koji prati predavanja



# Napomena

Slajdovi su nastali obradom slajdova profesora Nenada Mitića za predmet Uvod u relacione baze podataka

# Sadržaj

- 1 O kursu
- 2 Baze podataka**
- 3 Relacioni model
- 4 SQL

# Pre baza podatka

- Tradicionalni pristup razvoju sistema za čuvanje i obradu podataka
- Neki nedostaci:
  - ponavljanje istih podataka za potrebe različitih aplikacija
  - nekonzistentnost podataka
  - otežan je pristup više korisnika istoj datoteci
  - ...

# Sistem baza podataka

Sistem baza podataka (SBP) je u osnovi sistem za računarsko zapisivanje i čuvanje slogova, tj. sistem čija je svrha da čuva informacije i dozvoli korisniku da te informacije dobije i menja

# Komponente SBP

- Podaci
- Hardver
- Softver
- Korisnici

# Komponente SBP - podaci i hardver

- Podaci
  - Integrisani
  - Deljivi
- Hardver
  - Spoljašnji memorijski uređaji
  - Procesori i glavna memorija

# Komponente SPB - Softver

## Softver

- sistem za upravljanje bazom podataka - SUBP (eng. Database management system, DBMS)
  - nivo softvera koji se nalazi između korisnika i fizičkih podataka u bazi
  - štiti korisnike baze od detalja na hardverskom nivou
  - upravlja svim zahtevima za direktan pristup bazi
  - Relacioni: Oracle RDBMS, IBM DB2, Microsoft SQL Server, MySQL
- Alati za razvoj aplikacija, pisanje izveštaja, pomoćni (utility) programi, program za upravljanje transakcijama (TP monitor)

# Komponente SPB - korisnici

## Korisnici

- Aplikativni programeri
- Krajnji korisnici
- Administratori
  - Administrator baze podataka
  - Administrator podataka



# Administrator podataka

- Administrator podataka (eng. data administrator, DA)
  - razume postojeće podatke
  - odlučuje koji podaci će biti čuvani u bazi podataka
  - ustanovljava pravila za održavanje i rad sa podacima po njihovom čuvanju u bazi
  - nije tehničko lice, već pripada upravljačkim strukturama

# Administrator baze podataka

- Administrator baze podataka (eng. database administrator, DBA)
  - profesionalac u IT
  - formira bazu
  - odgovoran za implementaciju odluka DA
  - odgovoran za rad sistema, performanse, ...

# Baza podataka

- Baza podataka je skup postojećih podataka koji se koriste od strane aplikativnih sistema u nekom okruženju.
- Postojeći podaci

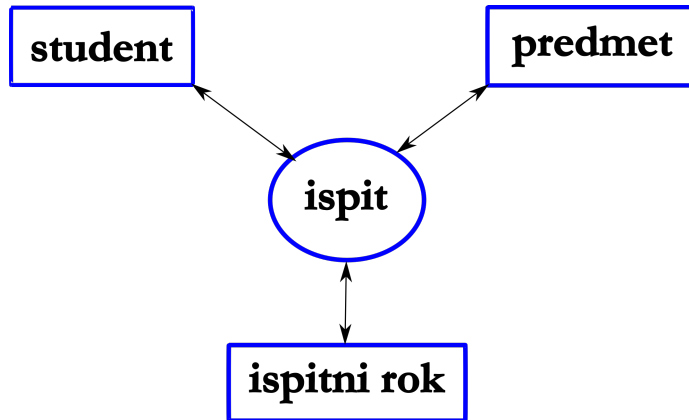
# Baza podataka

Šta bi bili podaci u studentskoj bazi podataka?

# Entiteti i odnosi

- Entiteti i odnosi
  - Entitet: osoba, mesto stvar, ...
  - Odnos: opis veze između dva ili više entiteta
  - Ako entitet definišemo kao objekat o kome želimo da sakupljamo informacije, tada odnosi zadovoljavaju ovu definiciju.

# Dijagram za malu studentsku bazu podataka



# Svojstva

- Entiteti imaju svojstva koja odgovaraju informacijama koje želimo da sačuvamo o entitetima
- Student: indeks, ime, prezime, mesto rođenja, datum upisa ...

# Model podataka

- Model podataka je apstraktna definicija objekata i operatora, .... koji zajedno čine apstraktnu mašinu sa kojom korisnik komunicira
  - Objekti dopuštaju modeliranje strukture podataka
  - Operatori dopuštaju modeliranje ponašanja
- Implementacija datog modela je fizička realizacija komponenata apstraktne mašine koje zajedno čine model na stvarnoj mašini



# Relacioni model podataka

- Relacione baze podataka se zasnivaju na relacionom modelu podataka
  - Podaci se predstavljaju korisnicima preko relacije (tabele), tj. torke u relacijama (ili redovi u tabelama)
  - Obezbeđeni su operatori koji se primenjuju nad torkama (redovima) relacije (tabele)
    - restrikcija - izdvajanje torki koje zadovoljavaju neki uslov
  - relacija je matematički termin za tabelu

# Tabela dosije

INDEKS	IME	PREZIME	DATUM_RODJENJA	MESTO_RODJENJA	DATUM_UPISA
20140021	Milos	Peric	20.01.1995	Beograd	06.07.2014
20140022	Marijana	Savkovic	11.03.1995	Kraljevo	05.07.2014
20130023	Sanja	Terzic	09.11.1994	Beograd	04.07.2013
20130024	Nikola	Vukovic	17.09.1994	NULL	04.07.2013
20140025	Marijana	Savkovic	04.02.1995	Kraljevo	06.07.2014
20140026	Zorica	Miladinovic	08.10.1995	Vranje	06.07.2014
20130027	Milena	Stankovic	NULL	NULL	NULL

# Prednosti rada sa bazom podataka

- Podaci mogu biti deljenji
- Smanjenje redundantnosti podataka
- Izbegavanje nekonzistentnosti
- Podrška za transakcioni rad
- Održavanje integriteta
- Primena zaštite podataka
- Balansiranje između konfliktnih zahteva
- Primena standarda

# Nezavisnost podataka

- Aplikacije implementirane na starim sistemima su bile zavisne od fizičke reprezentacije podataka i tehnika za pristup, što nije pogodno
  - različite aplikacije zahtevaju različite poglede nad istim podacima
  - DBA mora da ima slobodu da promeni fizicku reprezentaciju ili pristupne tehnike radi performansi
- Definicija: nezavisnost podataka je otpornost aplikacije na promene fizičke reprezentacije podatka i pristupnih tehnika

# Nezavisnost podataka

- Pojmovi na koje se odnose promene:
  - sačuvano polje (eng. stored field) je najmanja jedinica podataka koja može da se čuva
  - sačuvani slog (eng. stored record) je skup sačuvanih polja
  - sačuvana datoteka (eng. stored file) je skup svih trenutno postojećih polja sačuvanih slogova istog tipa

# Nezavisnost podataka

- Aspekti sačuvanih reprezentacija koji mogu da budu predmet promena od strane DBA
  - reprezentacija brojčanih podataka
  - reprezentacija znakovnih podataka
  - jedinice za brojčane podatke
  - kodiranje podataka
- Baza treba da bude sposobna da se širi bez promene postojećih aplikacija i bez negativnog uticaja na postojeće aplikacije



# Neformalni pogled na relaciji model

- Codd je 1970. postavio teorijsku osnovu za relaciji model podataka
- Intuitivno, relaciji model
  - predstavlja jedan način gledanja na podatke
  - sadrži pravila za predstavljanje podataka (preko tabela) i pravila za rad sa tim podacima (izdvajanje, spajanje, ....)



# Aspekti relacionog modela

- Aspekt strukture: svi podaci u bazi se korisniku prikazuju isključivo u obliku tabela
- Aspekt integriteta: tabele zadovoljavaju izvesna ograničenja (primarni i spoljašnji ključevi, ...)
- Aspekt obrade: operatori koji su na raspolaganju korisnicima za obradu tabela su takvi da izvode tabele iz tabela.

# Tabela dosije

INDEKS	IME	PREZIME	DATUM_RODJENJA	MESTO_RODJENJA	DATUM_UPISA
20140021	Milos	Peric	20.01.1995	Beograd	06.07.2014
20140022	Marijana	Savkovic	11.03.1995	Kraljevo	05.07.2014
20130023	Sanja	Terzic	09.11.1994	Beograd	04.07.2013
20130024	Nikola	Vukovic	17.09.1994	NULL	04.07.2013
20140025	Marijana	Savkovic	04.02.1995	Kraljevo	06.07.2014
20140026	Zorica	Miladinovic	08.10.1995	Vranje	06.07.2014
20130027	Milena	Stankovic	NULL	NULL	NULL

# Domeni

Domen = skup svih mogućih vrednosti (tip podataka)

- Svaki atribut ima domen
- Svaki tip može da bude ugrađen ili korisnički definisan
- Svaki tip može da bude atomski (skalarni) ili učauren (nije skalaran)
- Atributi relacije mogu da budu proizvoljnog domena

# Atributi i domeni

- u implementacijama postoje ugrađeni (sistemski predefinisani)
  - INTEGER
  - CHAR
  - ...
- ali i korisnički definisani
  - INDEKS
  - IME
  - GOD\_RODJENJA
  - .....

# Domeni - SQL podrška

SQL podržava sledeće relacione domene

- Brojevi (numbers)
- Niske karaktera (character strings)
- Niske bitova (bit strings)
- Datumi (dates)
- Vremena (times)
- Kombinacija datuma i vremena (timestamps)
- Intervali godina/mesec (year/month intervals)
- Intervali dan/vreme (day/time intervals)
- ...

# Nedostajuće vrednosti

U svakodnevnoj praksi se često javlja problem nedostatka podataka:

- ...datum rođenja nepoznat...
- ...ime supružnika nepoznato...

Potreba da se

- indikator o nedostatku vrednosti čuva u bazi
- na odgovarajući način vrši obrada takvih podataka

# Nedostajuće vrednosti

- Najčešći pristup, prihvaćen i u praksi je korišćenje *nedostajuće vrednosti* (NULL), odnosno trovalentne (3VL) logike
- Codd jer predložio korišćenje 4-valentne logike jer postoje dve vrste nedostajućih vrednosti:
  - vrednost je nepoznata
  - vrednost nije primenljiva, vrednost ne postoji, ...

# 3VL logika

Tri vrednosti

- Tačno (T, true)
- Netačno (F, false)
- Nepoznato (U, unknown)

Operatori: I, ILI, NE, MOŽDA (AND, OR, NOT, MAYBE)



## 3VL logika

AND	T	U	F
T	T	U	F
U	U	U	F
F	F	F	F

OR	T	U	F
T	T	T	T
U	T	U	U
F	T	U	F

NOT	
T	F
U	U
F	T

# 3VL logika - operator MOŽDA

MAYBE	
T	F
U	T
F	F

Operator je potreban npr. zbog ovakvih upita: Prikazati sve zaposlene koji su možda bili, ali za koje nije sasvim pouzdano da su bili, programeri rođeni pre 25. januara 1991. godine sa platom manjom od 50.000 na dan 30.09.2011.

# Relacije

Neka je dat skup od  $n$  tipova ili domena  $T_i$  ( $i = 1, 2, \dots, n$ ), pri čemu ne moraju svi tipovi da budu međusobno različiti.  $R$  je relacija nad tim domenima ako se sastoji od dva dela, zaglavlja i tela, gde važi:

- Zaglavlje je skup od  $n$  atributa oblika  $A_i : T_i$  ( $i=1,2,\dots,n$ ) gde su  $A_i$  imena atributa relacije  $R$ , a  $T_i$  odgovarajuća imena domena
- Telo je skup od  $m$  torki  $t$  gde je  $t$  skup komponenti oblika  $A_i : v_i$  u kojima je  $v_i$  vrednost iz domena  $T_i$

$m$  se naziva kardinalnost, a  $n$  stepen (arnost) relacije  $R$

# Osobine relacije

- Nema ponovljenih (duplih) torki
- Torke su neuređene, od vrha ka dnu
- Atributi su neuređeni, sa leva u desno
- Svaka toraka sadrži tačno jednu vrednost za svaki atribut. Za relaciju koja zadovoljava ovu osobinu se kaže da je normalizovana, odnosno da je u prvoj normalnoj formi.

# Relacija = tabela?

Nisu jednake, jer

- Tabela može da sadrži duplirane redove dok relacija ne može da sadrži duplirane torke
- Redovi u tabeli su uređeni u redosledu od vrha ka dnu, dok za relaciju to ne važi
- Kolone u tabeli su uređene u redosledu sa leva u desno, dok za relaciju to ne važi

# Relacije

Definicija relacije ima sledeći oblik:

RELATION {<lista atributa razdvojenih zarezima>}

gde je atribut je uređen par oblika

<ime atributa><ime tipa>

Primer:

DOSIJE {Indeks INDEKS, Ime IME, Prezime PREZIME, God\_rodjenja  
GOD\_RODJENJA, Mesto\_rodjenja MESTO\_RODJENJA}

# Relacije

- Relaciona baza podataka je skup relacija.
- Relaciona shema je opis strukture relacija

# Primer relacionih operatora - projekcija

Projekcija izdvaja pojedinačne kolone iz tabele

DOSIJE [Indeks, Ime, Prezime]



# Primer relacionih operatora - restrikcija

Restrikcija (selekcija) izdvaja pojedinačne redove iz tabele

DOSIJE WHERE mesto\_rodjenja='Beograd'

# Primer relacionih operatora - spajanje

Prirodno spajanje kombinuje dve tabele na osnovu zajedničkih vrednosti u zajedničkoj koloni

DOSIJE JOIN ISPIT

# Primer relacionih operatora - spajanje

Kod spajanja se u rezultujućoj tabeli

- zajedničke vrednosti javljaju samo jednom u redu
- javljaju sve kombinacije uparenih vrednosti (npr. 20140021 vrednost za Indeks)
- ne javljaju se redovi sa vrednostima koje se ne nalaze u obe tabele koje učestvuju u spajanju

# Efekat relacionih operatora

- Rezultat primene svakog od operatora je tabela - osobina zatvorenja relacionih sistema
- Pošto je rezultat primene operatora istog tipa kao i njegov argument mogu da se pišu ugneždeni relacioni izrazi
- Sve operacije ce primenjuju na ceo skup istovremeno, a ne samo na pojedinačni red
- Rezultat operacije nije nikada pojedinačni red već je uvek kompletna tabela koja sadrži skup redova

# Osnovna ograničenja u relacionim bazama podataka

- primarni ključ relacije - skup atributa (kolona) čije vrednosti mogu jedinstveno da odrede n-torku (red) u relaciji (tabeli)
  - atribut indeks u relaciji dosije
- strani ključa - veza između dve tabele; vrednosti u atributima stranog ključa mogu biti samo one koje se javljaju u drugoj (baznoj) relaciji sa kojom je povezana relacija u kojoj je definisan strani ključ
  - strani ključ u relaciji ispit: indeks, bazna relacija: dosije

# Karakteristike relacione baze

- Relacioni sistemi zahtevaju samo da se baza prikaže korisniku u obliku tabele (način smeštanja i čuvanja na medijumima nije specificiran)
- Informacioni princip: Celokupan informacioni kontekst baze se prikazuje na tačno jedan način kao eksplicitne vrednosti u pozicijama vrsta i kolona tabele
- Posledica informacionog principa: nema pokazivača koji međusobno povezuju tabele

# Formalni pogled na relacioni model

Relacioni model se sastoji od:

- Otvorenog skupa skalarnih tipova (koji uključuje i tip logičkih vrednosti boolean)
- Generators relacionih tipova i njihove odgovarajuće interpretacije
- Mogućnosti definisanja relacionih promenljivih za generisane relacione tipove
- Operacije relacione dodele kojom se dodeljuju relacione vrednosti definisanim relacionim promenljivim
- Otvorenog skupa opštih relacionih operatora ("relaciona algebra") za izvođenje relacionih vrednosti iz drugih relacionih vrednosti

# Terminologija

Codd je pri formulisanju principa relacionih baza uveo novu terminologiju

Termin	Znacenje
Relacija	matematički termin za tabelu
Torka	red u tabeli
Atribut	kolona u tabeli
Kardinalnost	broj torki
Stepen	broj atributa
Primarni ključ	atribut ili kombinacija atributa čije vrednosti jedinstveno identifikuju torku
Domen	skup važećih vrednosti (tipova)



# Tabela dosije

INDEKS	IME	PREZIME	DATUM_RODJENJA	MESTO_RODJENJA	DATUM_UPISA
20140021	Milos	Peric	20.01.1995	Beograd	06.07.2014
20140022	Marijana	Savkovic	11.03.1995	Kraljevo	05.07.2014
20130023	Sanja	Terzic	09.11.1994	Beograd	04.07.2013
20130024	Nikola	Vukovic	17.09.1994	NULL	04.07.2013
20140025	Marijana	Savkovic	04.02.1995	Kraljevo	06.07.2014
20140026	Zorica	Miladinovic	08.10.1995	Vranje	06.07.2014
20130027	Milena	Stankovic	NULL	NULL	NULL

# Terminologija

## Nad tabelom dosije

Relacija	je kompletna tabela (ime relacije DOSIJE)
Torka	je svaki red u tabeli
Atributi	tabele su kolone Indeks, Ime, Prezime, Datum_rodenja, Mesto_rodenja, Datum_upisa
Kardinalnost	tabele je 7 (tabela poseduje 7 torki)
Stepen	tabele je 5 (tabela ima 6 atributa)
Primarni ključ	tabele je kolona Indeks
Domeni	u tabeli su skup mogućih brojeva indeksa (INDEKS), skup mogućih imena (IME), ..



# Relacioni upitni jezik

- za komunikaciju sa relacionom bazom podataka
- relacioni sistem SYSTEM R i upitni jezik SQUARE (Specifying QUeries As Relational Expressions)
- SEQUEL (Structured English QUery Language)
- SQL (Structured Query Language)
- prvi standard za upitne jezike je ANSI SQL/86, od tada je objavljeno više standarda

# Podjezici SQL-a

- jezik za definisanje podataka (eng. data definition language (DDL))
- jezik za manipulisanje podacima (eng. data manipulation language (DML))
- jezik za kontrolu pristupa podacima (eng. data control language (DCL))

# Naredbe DDL

- create *objekat ime\_objekta*
- drop *objekat ime\_objekta*
- alter *objekat ime\_objekta*
- *objekat*
  - database
  - schema
  - table
  - view
  - ...

# Pravljenje i brisanje baze podataka

## Naredba

- za pravljenje baze podataka
  - create database *ime\_baze\_podataka*
  - Primer: create database stud2020
- za brisanje baze podataka
  - drop database *ime\_baze\_podataka*
  - Primer: drop database stud2020

# Pravljenje tabele

```
create table ime-tabele (  
def-kolone [, def-kolone]+  
[, def-primarni-ključ]?  
[, def-strani-ključ]*  
...)
```

- def-kolone  
ime-kolone tip-podatka [not null]?
  - tip-podatka: smallint, integer, float, dec(p,q), varchar(n), date ...
- def-primarni-ključ  
[CONSTRAINT ime]? PRIMARY KEY (ime-kolone [, ime-kolone]\*)
- def-strani-ključ  
[CONSTRAINT ime]? FOREIGN KEY ( ime-kolone [, ime-kolone]\*)  
REFERENCES bazna-tabela



# Pravljenje tabele

```
create table dosije (  
  indeks integer not null,  
  ime varchar(25) not null,  
  prezime varchar(25) not null,  
  datum_upisa date not null,  
  datum_rodjenja date,  
  mesto_rodjenja varchar(100) ,  
  primary key (indeks)  
)
```

# Pravljenje tabele

```
create table ispit (  
indeks integer not null,  
id_predmeta integer not null,  
godina_roka smallint not null,  
oznaka_roka char(5) not null,  
ocena smallint not null with default 5,  
datum_ispita date,  
bodovi smallint,  
primary key (indeks, id_predmeta, godina_roka, oznaka_roka),  
foreign key (godina_roka, oznaka_roka) references ispitni_rok,  
foreign key (indeks) references dosije,  
foreign key (id_predmeta) references predmet  
)
```

# Pretraživanje tabele

Osnovni oblik **select** naredbe je

**select** lista-kolona **3**

**from** ime-tabele **1**

**where** uslovi **2**

U okviru

- **select** klauzule navode se kolone, izrazi ili \* (izdvajanje svih kolona tabele)
- **where** klauzule navode se uslovi koji:
  - sadrže vrednosti kolone, konstanti i rezultata izraza i relacijske operacije =, <, >, <>, <=, >=
  - povezuju se sa logičkim operacijama AND i OR
  - mogu da sadrže logičku operaciju NOT (oblik NOT izraz)

# Pretraživanje tabele

- **select** klauzula - za projekciju
- **where** klauzula - za restrikciju

# Primeri

## Primer 1: Prikazati kompletan sadržaj tabele dosije

```
select *  
from dosije
```

ili

```
select indeks, ime, prezime, datum_rodjenja, mesto_rodjenja  
from dosije
```

ili

```
select indeks, prezime, ime, mesto_rodjenja, datum_rodjenja  
from dosije
```



# Primeri

## Primer 3: Prikazati informacije o studentima koji su rođeni u Beogradu (restrikcija)

```
select *  
from dosije  
where mesto_rodjenja = 'Beograd'
```

## Primer 4: Prikazati informacije o studentima koji su nisu rođeni u Beogradu (restrikcija)

```
select *  
from dosije  
where mesto_rodjenja <> 'Beograd'
```

Pitanje: šta se dešava sa redovima koji u koloni mesto\_rodjenja imaju nedostajuću vrednost?

# Primeri

**Primer 5: Prikazati imena i prezimena studenata koji su rođeni 01.01.1994. godine u Beogradu(restrikcija)**

```
select ime, prezime  
from dosije  
where datum_rodjenja>='01.01.1994' and mesto_rodjenja='Beograd'
```

**Primer 6: Prikazati imena i prezimena studenata koji su rođeni 01.01.1994. godine ili su rođeni u Beogradu (restrikcija)**

```
select ime, prezime  
from dosije  
where datum_rodjenja>='01.01.1994' or mesto_rodjenja='Beograd'
```



## Nedostajuće vrednosti

- pri primeni WHERE klauzule na tabelu T eliminišu se svi redove za koje uslov u WHERE ima vrednost netačno ili nedefinisano
- Test za proveru da li je vrednost izraza null: izraz **is null**
- Test za proveru da li vrednost izraza nije null: izraz **is not null**

# Primeri

**Primer 7: Prikazati imena i prezimena studenata za koje nije poznato mesto rođenja**

```
select ime, prezime, mesto_rodjenja  
from dosije  
where mesto_rodjenja is null
```

**Primer 8: Prikazati imena i prezimena studenata za koje je poznato mesto rođenja**

```
select ime, prezime, mesto_rodjenja  
from dosije  
where mesto_rodjenja is not null
```