

---

# Baze podataka

Ovaj

---

Uvod u SQL (Structured Query  
Language)

---

# Pregled predavanja

- Baze podataka
  - Predavanje 1
    - Uvod
    - Relacione baze
    - SQL komande
      - Kreiranje baze
      - Unos podataka
      - Pretraživanje podataka
  - Predavanje 2
    - Dodatne SQL komande
    - Pregled programskih paketa baziranih na SQL-u
      - Komercijalni
      - Besplatni
        - Uputstvo za korištenje

---

# Uvod

- Database - Baza podataka
  - je skup informacija (podataka) koje su prikupljene u određenom, obično dužem, vremenskom periodu.
- DBMS (Database Management System)
  - Specijalizirani softver koji služi za kreiranje i manipulaciju bazama podataka. Bitne osobine:
    - Obezbjeđuje pohranu velike količine podataka
    - Obezbjeđuje metodologiju za pristup, pretraživanje i modifikaciju podataka.
    - Omogućava istovremeni pristup podacima od strane više korisnika

---

# Primjeri primjene DBMS-a

- Sistem za rezervaciju avionskih karata
  - Neki od podataka koji se pohranjuju:
    - Informacija o rezervaciji: ime i prezime putnika preference za obrok, dodjeljeni broj sjedišta itd...
    - Informacija o letu uključuje mjesto polaska i dolaska kao i termine, tip aviona, ukupan broj sjedišta itd...
    - Informacije o cijenama, taksama itd..
  - Tipične pretrage koje izvršava agent su: Izlistavanje letova na nekoj destinaciji u određenim terminima, skupa sa listom slobodnih mjesta. Pronalaženje alternativnih ruta itd...
  - Tipični izmjene podataka su rezervisanje mjesta na određenom letu.
  - Sistem mora obezbijediti ukupni integritet baze u slučaju da više agenata istovremeno rezerviraju mjesto na letu.

---

# Primjeri primjene DBMS-a

- Bankovni sistem
  - Neki od podataka koji se pohranjuju:
    - Informacije o klijentima: ime, prezime, adresa, broj telefona, itd
    - Informacije o računima: tip računa, kredita itd...
    - Veza između računa i klijenta
- Kao i u prethodnom slučaju više subjekata istovremeno treba da može pristupiti računu npr. Klijent preko ATM aparata i službenik na šalteru preko računara.
- Sistem mora da bilježi sve promjene i da radi i u izvanrednim uslovima npr. gubitak električne energije.

---

# DBMS evolucija

- Prvi komercijalni DBMS paketi:
  - Bazirani na fajl sistemima
  - Neadekvatna metodologija pretraživanja
  - Šema baze podataka ovisna od strukture direktorija
  - Problemi kada više od jednog korisnika pokušavaju izmjeniti neki od podataka u bazi.
- 1970 Ted Codd u naučnom radu “*A relational model for large shared data banks,*” predlaže model za DBMS baziran na tabelama. Ovaj rad predstavlja osnovu današnjih relacionih baza podataka.
  - Relacioni DBMS-ovi danas čine preko 90% tržišta softvera za baze podataka.

---

# Relacione baze podataka

- U relacionim bazama podataka svi podaci smještaju se u tabele:
  - Svakoj tabeli dodjeljuje se **ime**
    - Svakoj koloni u tabeli dodjeljuje se **atribut** koji opisuje podatke koji se smještaju u toj koloni
    - Svaki red u tabeli predstavlja jedan **zapis** u bazi podataka
- Prosječna baza sadrži na stotine tabela
- Između tabela mogu postojati veze
  - Operacije nad tabelama su matematički definisane *Relacionom Algebrom* (presjek, unija razlika itd...)

# Primjer jednostavne baze

Atribut

Ime tabele

Zapis

Student			
<i>BrIndexa</i>	<i>Prezime</i>	<i>Ime</i>	<i>Godiste</i>
115	Begic	Amir	1980
200	Alic	Emina	
123	Popic	Marko	1983

Predmet		
<i>Kod</i>	<i>Naziv</i>	<i>Fakultet</i>
PMF1	Matematika	Prirodno Matematički Fakultet
EE12	Elektronika	Fakultet Elektrotenike
EF25	Sociologija	Ekonomski Fakultet

Ocjene			
<i>SID</i>	<i>PKOD</i>	<i>Ocjena</i>	<i>Datum</i>
200	PMF1	10	10.2.2001
123	EE12	10	6.6.2000
115	EF25	7	8.5.2004
123	PMF1	8	25.8.2000

Ovaj



---

# SQL

- SQL je najčešće korišteni jezik za komunikaciju sa relacionom bazom podataka i, prema ANSI-u, predstavlja standardni jezik za relacione DBMS
- SQL-92 je osnovni SQL standard
  - Mnogi DBMS podržavaju samo dio standarda a čest je slučaj da dodaju i svoje definicije
- Omogućava slijedeće:
  - Definiranje i modificiranje strukture baze (kreiranje tabela i njihovih relacija)
  - Pretraživanje baze u potrazi za informacijom
  - Dodavanje novih i promjena starih podataka u bazi
  - Sigurnosne postavke za pristup bazi spram korisničkog imena i šifre

---

# Kreiranje baze podataka

- Dizajniranje baze je proces kojim se utvrđuje struktura baze kako bi se:
  - smanjilo nepotrebno ponavljanje podataka
  - omogućila brza pretraga i pristup podacima
- Procesom dizajniranja utvrđuju se:
  - Tabele i atributi u tabelama skupa sa tipovima podataka atributa
  - Relacija između tabela
- Tabele skupa sa atributima se definišu naredbom `CREATE`

# Kreiranje tabela

**CREATE TABLE ime\_tabele (atribut1 tip-podatka, ..., atributN tip-podatka, opcionalna ograničenja na attribute)**

- **CREATE TABLE je SQL komanda za kreiranje tabele**

- ime\_tabele, atribut1, ..., atributN

- predstavljaju imena za tabelu i attribute

- tip-podatka

- Predstavlja tip podataka koji će se smještati u datu kolonu. Neki validni SQL tipovi podataka:

- `varchar(n)` – string varijabilne dužine (do maksimalno n-slova)
- `integer` – cjelobrojne vrijednosti
- `float` – realne vrijednosti
- `date` – datum

# Kreiranje tabela

**CREATE TABLE ime\_tabele (atribut1 tip-podatka, ..., atributN tip-podatka, opcionalna ograničenja na attribute)**

- Često korištena ograničenja:
  - PRIMARY KEY
    - Svaka tabela treba da ima kolonu u kojoj su vrijednosti unikatne (ne ponavljaju se). Ta kolona se definiše kao primarni ključ tabele naredbom `PRIMARY KEY`
    - Primarni ključ može biti:
      - posebna definisana kolona
      - već definisana kolona u tabeli u kojoj su unešene vrijednosti sigurno unikatne
      - kombinacija više kolona koje pojedinačno ne moraju biti unikatne
  - FOREIGN KEY
    - Ukoliko se u nekoj koloni pojavljuju vrijednosti definisane kao primarni ključ u drugoj tabeli tada se ta kolona označava sa `FOREIGN KEY`. Naredbom `REFERENCES` se označava ime druge tabele. Ova kolona ne može sadržavati vrijednosti koje nisu definisane u drugoj tabeli.
  - NOT NULL
    - Definiše da kolona ne može imati prazan zapis

# Primjer

Student			
<i>BrIndexa</i>	<i>Prezime</i>	<i>Ime</i>	<i>Godiste</i>
115	Begic	Amir	1980
200	Alic	Emina	
123	Popic	Marko	1983

Predmet		
<i>Kod</i>	<i>Naziv</i>	<i>Fakultet</i>
PMF1	Matematika	Prirodno Matematički Fakultet
EE12	Elektronika	Fakultet Elektrothenike
EF25	Sociologija	Ekonomski Fakultet

**CREATE TABLE** student (  
BrIndexa **INTEGER NOT NULL PRIMARY KEY**,  
Prezime **VARCHAR(20) NOT NULL**,  
Ime **VARCHAR(10) NOT NULL**,  
Godiste **INTEGER**  
);

**CREATE TABLE** predmet (  
Kod **VARCHAR(6) NOT NULL**,  
Naziv **VARCHAR(100) NOT NULL**,  
Fakultet **VARCHAR(100) NOT NULL**,  
**PRIMARY KEY (Kod)**  
);

**Validne definicije ograničenja**

**Komanda se terminira sa ;**

**Zarezima se odvaja definicija novog atributa**

Ovaj

# Primjer

<b>Ocjene</b>			
<i>SID</i>	<i>PKOD</i>	<i>Ocjena</i>	<i>Datum</i>
200	PMF1	10	10.2.2001
123	EE12	10	6.6.2000
115	EF25	7	8.5.2004
123	PMF1	8	25.8.2000

```
CREATE TABLE ocjene (  
  sid INTEGER,  
  pkod VARCHAR(6),  
  ocjena INTEGER NOT NULL,  
  datum DATE NOT NULL,  
  FOREIGN KEY (sid) REFERENCES student(BrIndexa),  
  FOREIGN KEY (pkod) REFERENCES predmet(Kod),  
  PRIMARY KEY (sid,pkod)  
);
```

---

# Unošenje podataka u bazu

**INSERT INTO TABLE ime\_tabele (lista\_atributa) VALUES (lista\_vrijednosti)**

- ime\_tabele
  - Ime tabela u koju se unose podaci
- lista\_atributa
  - Lista sa imenima kolona u koje se upisuju podaci
- lista\_vrijednosti
  - Lista sa vrijednostima koje poprimaju atributi navedeni u lista\_atributa
  - Vrijednosti moraju odgovarati po tipu deklariranim tipovima podataka u definiciji tabele
  - Atributi koji su definirani sa ograničenjem NOT NULL moraju poprimiti neku vrijednost

# Primjer

Student			
<i>BrIndexa</i>	<i>Prezime</i>	<i>Ime</i>	<i>Godiste</i>
115	Begic	Amir	1980
200	Alic	Emina	
123	Popic	Marko	1983

```
INSERT INTO student (BrIndexa, godiste, ime, prezime) VALUES (115, 1980, 'Amir', 'Begic');  
INSERT INTO student (BrIndexa, prezime, ime) VALUES (200, 'Alic', 'Emina');  
INSERT INTO student (BrIndexa, godiste, ime, prezime) VALUES (123, 1983, 'Marko', 'Popic');
```



# Primjer

Predmet		
<i>Kod</i>	<i>Naziv</i>	<i>Fakultet</i>
PMF1	Matematika	Prirodno Matematicki Fakultet
EE12	Elektronika	Fakultet Elektrothenike
EF25	Sociologija	Ekonomski Fakultet

Ocjene			
<i>SID</i>	<i>PKOD</i>	<i>Ocjena</i>	<i>Datum</i>
200	PMF1	10	10.2.2001
123	EE12	10	6.6.2000
115	EF25	7	8.5.2004
123	PMF1	8	25.8.2000

Ovaj

```
INSERT INTO predmet (kod,naziv,fakultet)
VALUES('PMF1','Matematika','Prirodno Matematicki Fakultet');
INSERT INTO predmet (kod,naziv,fakultet)
VALUES('EE12','Elektronika','Fakultet Elektrotehnike');
INSERT INTO predmet (kod,naziv,fakultet)
VALUES('EF25','Sociologija','Ekonomski Fakultet');
```

```
INSERT INTO ocjene (sid,pkod,ocjena,datum) VALUES(200,'PMF1',10,'2001-2-10');
INSERT INTO ocjene (sid,pkod,ocjena,datum) VALUES(123,'EE12',10,'2000-6-6');
INSERT INTO ocjene (sid,pkod,ocjena,datum) VALUES(115,'EF25',7,'2004-5-8');
INSERT INTO ocjene (sid,pkod,ocjena,datum) VALUES(123,'PMF1',8,'2000-8-25');
```

---

# Pretraživanje baze

**SELECT lista\_kolona FROM lista\_tabela opcione komande**

- lista\_kolona
  - Određuje koje će kolone biti sadržane u rezultatu pretrage
  - Ukoliko je umjesto imena kolone korištena \* rezultat je kompletna tabela
- lista\_tabela
  - Određuje koje se tabele pretražuju
- Opcione komande
  - WHERE logički\_uslov
    - Određuje koji redovi će biti sadržani u rezultatu pretrage na osnovu ispunjenja logičkog uslova
      - Logički uslovi se formiraju logičkim operatorima
        - <, >, <>, =, >=, <=
        - LIKE, IN
  - ORDER BY ime\_kolone
    - Omogućava da se rezultat pretrage prikazuje sortiran prema koloni ime\_kolone

# Primjer

```
SELECT * FROM student;
```

BRINDEXA	PREZIME	IME	GODISTE
115	Begic	Amir	1980
123	Popic	Marko	1983
200	Alic	Emina	(null)

```
SELECT ime, prezime FROM student;
```

IME	PREZIME
Amir	Begic
Marko	Popic
Emina	Alic

Ovaj

```
SELECT ime, prezime FROM student ORDER BY prezime;
```

IME	PREZIME
Emina	Alic
Amir	Begic
Marko	Popic

```
SELECT ime, godiste FROM student WHERE godiste > 1980;
```

IME	GODISTE
Marko	1983

```
SELECT brindexa FROM student WHERE ime LIKE '%mi%';
```

BRINDEXA
115
200

# Primjer

```
SELECT * FROM predmet
```

KOD	NAZIV	FAKULTET
EE12	Elektronika	Fakultet Elektrotehnike
EF25	Sociologija	Ekonomski Fakultet
PMF1	Matematika	Prirodno Matematički Fakultet

```
SELECT kod FROM predmet WHERE naziv LIKE '%ka' AND fakultet LIKE 'F%';
```

KOD
EE12

```
SELECT sid,ocjena FROM ocjene WHERE (ocjena IN (6,9,10));
```

SID	OCJENA
123	10
200	10

```
SELECT ime,brindexa FROM student WHERE  
(brindexa IN (SELECT sid FROM ocjene WHERE ocjena=10));
```

IME	BRINDEXA
Marko	123
Emina	200