

SINTAKSA I OSNOVE SQL-a

I auditorne vježbe

Teme

- ▶ Osnovni pojmovi
 - ▶ Sistem za upravljanje bazama podataka
 - ▶ Jezici u sistemima za upravljanje bazama podataka
 - ▶ SQL standard
- ▶ Sintaksa i osnove upotrebe SQL jezika
 - ▶ Osnovni objekti u SQL jeziku
 - ▶ Tipovi podataka
 - ▶ Naredbe za definiciju baze podataka i relacije
 - ▶ Četiri osnovne SQL naredbe za upravljanje podacima
 - ▶ Naredbe za prenos podataka u/iz datoteke



Sistem za upravljanje bazama podataka

- ▶ Baza podataka – definicija
- ▶ Sistem za upravljanje bazama podataka – SUBP (*Database Management System – DBMS*)
 - ▶ Programski sistem koji omogućava upravljanje bazom podataka
- ▶ Najvažnije zadaće SUBP su:
 - ▶ Zaštititi bazu podataka od neovlaštenog korištenja
 - ▶ Spriječiti narušavanja pravila integriteta
 - ▶ Osigurati obnovu podataka u slučaju uništenja
 - ▶ Spriječiti štetne interferencije korisnika u višekorisničkim sistemima
 - ▶ Omogućiti korištenje rječnika podataka (podaci o podacima)
 - ▶ Optimizirati sve funkcije i obavljati ih efikasno koliko je to moguće



Jezici u sistemima za upravljanje bazama podataka

- ▶ **DDL – Data description language (jezik za opis podataka)**
 - ▶ Jezik za definiciju ili deklaraciju objekata u bazi podataka
- ▶ **DML – Data manipulation language (jezik za upravljanje podacima)**
 - ▶ Omogućava manipulaciju objektima baze podataka (dohvat podataka, unos, izmjena i brisanje)
- ▶ **Ovi se jezici mogu upotrebljavati na različite načine:**
 - ▶ Obavljanje DML i DDL naredbi uz pomoć interaktivnih alata
 - ▶ Ugrađivanjem DML i DDL naredbi unutar “jezika domaćina” (host language) – C++, C#, Java, Informix 4GL, itd.
 - ▶ Pomoću korisničkih interfejsa koji omogućavaju upotrebu DML i DDL jezika u grafičkom obliku



SQL standard

- ▶ *Structured Query Language* (SQL) – upitni jezik temeljen na relacijskoj algebri i predikatnom računu
- ▶ Neproceduralnost – opisuje šta se želi dobiti kao rezultat, ali ne i kako se do tog rezultata dolazi
- ▶ Objedinjuje funkcije DDL i DML jezika
- ▶ Zadaća je omogućiti definiciju podataka, upravljanje podacima i provođenje kontrole nad podacima u relacijskoj bazi podataka
- ▶ Standardi: SQL-89, SQL-92, SQL-99, SQL:2003



Sintaksa i osnove upotrebe SQL jezika

- ▶ Osnovni objekti u SQL-u su:
 - ▶ Baza podataka (*Database*)
 - ▶ Tablica, relacija (*Table*)
 - ▶ Kolona, stupac (*Column*)
 - ▶ Pogled ili virtuelna tablica (*View*)
 - ▶ Sinonim (*Synonym*)
 - ▶ Pravilo integriteta (*Constraint*)
 - ▶ Indeks (*Index*)
 - ▶ Pohranjena procedura (*Stored procedure*)
 - ▶ Okidač (*Trigger*)



Identifikatori i velika i mala slova

- ▶ Imena objekata (identifikatori) u SQL-u formiraju se iz slova, znaka '_' i cifri.
 - ▶ U MySQL-u može sadržavati i \$
- ▶ Prvi znak od ukupno 18 značajnih mora biti slovo
 - ▶ U MySQL-u može početi i sa donjom crtom ili cifrom, ali ne smije se sastojati samo od cifri i može sadržavati 64 karaktera
- ▶ Ime objekta ne može biti rezervisana ključna riječ
 - ▶ Ispravno formirani identifikatori: stud, ispiti2008god, stud_ispit
 - ▶ Nespravno formirani identifikatori: _I I.mjesec, I I mjesec, stud-ispit
- ▶ Case-insensitive jezik
- ▶ Radi preglednosti – ključne riječi velikim slovima, ostali elementi naredbe malim slovima
- ▶ Postoji razlika između nizova znakova napisanih malim i velikim slovima

```
SELECT * FROM mjesto  
WHERE nazMjesto='Tuzla'
```

```
SELECT * FROM mjesto  
WHERE nazMjesto='TUZLA'
```



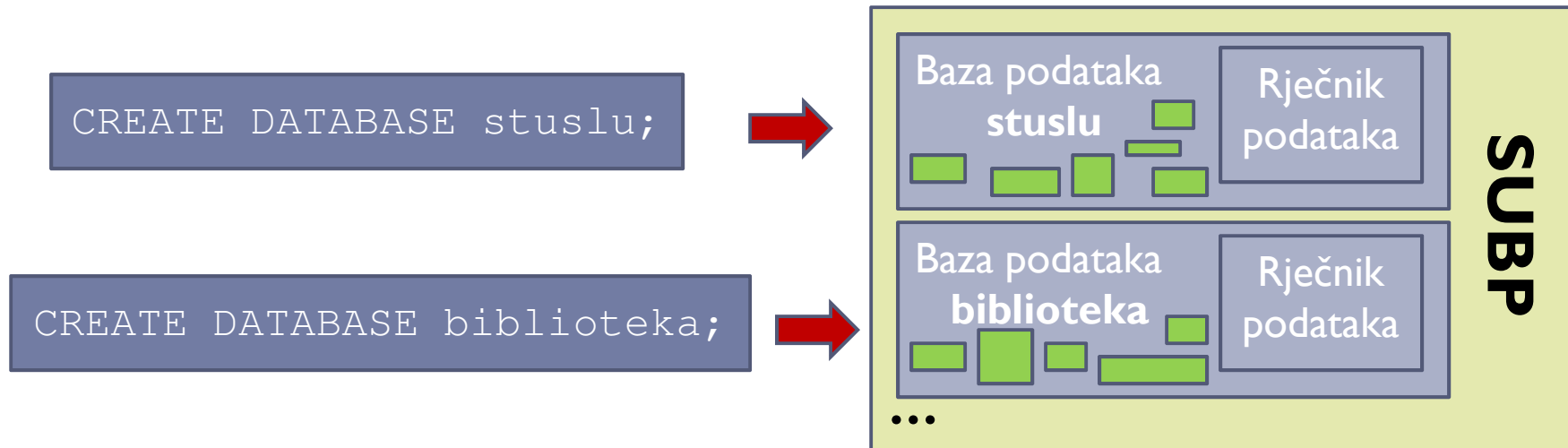
Obavljanje SQL naredbi u interaktivnom okruženju

- ▶ SQL naredba se ne mora završiti tačkom-zarez (;)
- ▶ Ukoliko se želi izvesti jedna za drugom dvije ili više naredbi, kao separator među naredbama se koristi (;)
- ▶ Komentari u SQL-u
 - ▶ Jedan ili više redova omeđenih vitičastim zagradama { }
 - ▶ Mjesto u redu na kojem se nalaze znakovi -- predstavlja početak komentara
- ▶ SQL jezik je jezik slobodnog formata naredbi



Kreiranje nove baze podataka

- ▶ Kreiranje nove instance baze podataka (kreiranje baze podataka)
 - ▶ Jedan DBMS može istovremeno upravljati sa više baza podataka



- ▶ Rječnik podataka sadrži opise relacijskih shema, integritetskih ograničenja, ...

DROP DATABASE stuslu;

Naredba za definiciju baze podataka

- ▶ Naredba namijenjena za stvaranje nove baze podataka

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name  
[create_specification] ...
```

create_specification:

```
[DEFAULT] CHARACTER SET [=] charset_name | [DEFAULT]  
COLLATE [=] collation_name
```

- ▶ *create_specification* specificira karakteristike baze podataka koje se pohranjuju u db.opt fajl
 - ▶ *CHARACTER SET* specificira podrazumjevani skup karaktera za bazu podataka – utf8, latin2, itd.
 - ▶ *COLLATE* specificira podrazumjevani skup pravila za poređenje karaktera (*collation*) baze podataka – utf8_general_ci, latin2_croatian_ci, itd.



Naredba za definiciju baze podataka

- ▶ Ukoliko ima dozvolu korisnik može pristupiti svakoj bazi podataka unutar instance sistema za upravljanje bazama podataka pomoću naredbe

```
USE imeBaze
```

- ▶ Veza sa bazom podataka ostaje otvorena do nove USE naredbe ili do kraja sesije
- ▶ Naredba kojom se bespovratno uništavaju podaci i cjelokupna struktura baze podataka

```
DROP DATABASE imeBaze
```



Tipovi podataka - numerički

- ▶ BIT[(M)] – bitsko polje, M predstavlja broj bita po vrijednosti (1 do 64)
- ▶ TINYINT – vrlo mali cijeli broj pohranjen u 1 byte
- ▶ BOOL, BOOLEAN – sinonim za TINYINT(1), 0 – FALSE, ne-0 – TRUE
- ▶ SMALLINT – Mali cijeli broj pohranjen u 2 byte-a
- ▶ MEDIUMINT – Cijeli broj srednje veličine (-8388608 do 8388607)
- ▶ INTEGER – Cijeli broj pohranjen u 4 byte-a (INT)
- ▶ BIGINT – Veliki cijeli broj
- ▶ SERIAL – Generira niz cijelih brojeva a alias je za BIGINT UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE
- ▶ DECIMAL – U posebnom formatu pohranjen decimalni broj. Npr. DECIMAL(15,3) predstavlja decimalni broj sa ukupno 15 cifara, od toga se 3 nalaze iza decimalne tačke
- ▶ FLOAT[(M,D)] – broj sa pokretnom tačkom jednostruke preciznosti, podrazumjevano 7 decimalnih mjesta
- ▶ DOUBLE [(M,D)] – broj sa pokretnom tačkom dvostruke preciznosti (REAL), podrazumjevano 15 decimalnih mjesta



Tipovi podataka – datum i vrijeme

- ▶ DATE – Vrijednosti ovog tipa se uvijek prikazuju u obliku datuma 'YYYY-MM-DD'. Opseg je '1000-01-01' do '9999-12-31'
- ▶ DATETIME[(fsp)] – Pohranjuje informacije o datumu i vremenu 'YYYY-MM-DD HH:MM:SS[(fsp)]'
- ▶ TIMESTAMP[(fsp)] – trenutak u vremenu. Pohranjuje broj sekundi od '1970-01-01 00:00:00' UTC
- ▶ TIME[(fsp)] – Vrijeme u formatu 'HH:MM:SS[.fraction]'
- ▶ YEAR[(2|4)] – godina u formatu sa dvije ili četiri cifre



Tipovi podataka – stringovi

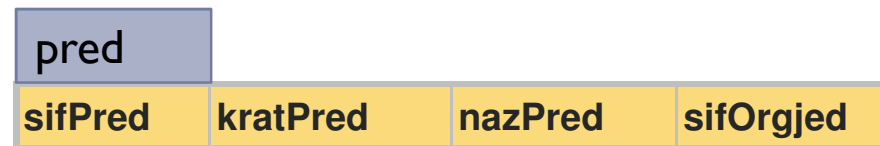
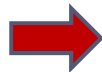
- ▶ CHAR – Znakovni niz (string). Npr. CHAR(24)
- ▶ NCHAR – Isto kao i CHAR ali uz dodatak kojim se omogućava ispravno leksikografsko uređenje niza znakova koji sadrže nacionalne kodne stranice
- ▶ VARCHAR – Niz znakova varijabilne dužine. Pohranjuje se kao prefiks dužine 1 ili 2 byte (broj byte-a u podacima) plus podaci
- ▶ NVARCHAR – U odnosu na VARCHAR isto što i NCHAR u odnosu na CHAR
- ▶ BINARY(M) – slično CHAR tipu, ali pohranjuje binarno stringove bajta umjesto nebinarno stringove karaktere
- ▶ VARBINARY(M) – Za BINARY slično kao VARCHAR za CHAR
- ▶ TINYBLOB – BLOB maksimalne dužine 255 byte-a
- ▶ TINYTEXT – TEXT maksimalne dužine 255 karaktere
- ▶ BLOB[(M)] – BLOB maksimalne dužine $2^{16}-1$ byte-a
- ▶ TEXT[(M)] – TEXT maksimalne dužine $2^{16}-1$ karaktere
- ▶ MEDIUMBLOB – BLOB maksimalne dužine $2^{24}-1$ byte-a
- ▶ MEDIUMTEXT – TEXT maksimalne dužine $2^{24}-1$ karaktere
- ▶ LONGBLOB – BLOB maksimalne dužine $2^{32}-1$ byte-a
- ▶ LONGTEXT – TEXT maksimalne dužine $2^{32}-1$ karaktere
- ▶ ENUM('val1','val2',...) – enumeracija, string objekat koji može imati samo jednu vrijednost odabranu iz liste vrijednosti 'val1','val2',..., NULL ili specijalnu vrijednost '' za grešku
- ▶ SET('val1','val2',...) – skup, string objekat koji može imati nijednu ili više vrijednosti od kojih svaka mora biti odabrana iz liste vrijednosti 'val1','val2',...



Kreiranje relacije

- ▶ Opisivanje relacijske sheme (kreiranje relacije)
 - ▶ Kreira se prazna relacija
 - ▶ Istovremeno je moguće definirati integritetska ograničenja

```
CREATE TABLE pred (  
    sifPred    INTEGER  
, kratPred    CHAR(8)  
, nazPred    NCHAR(60)  
, sifOrgjed   INTEGER  
);
```

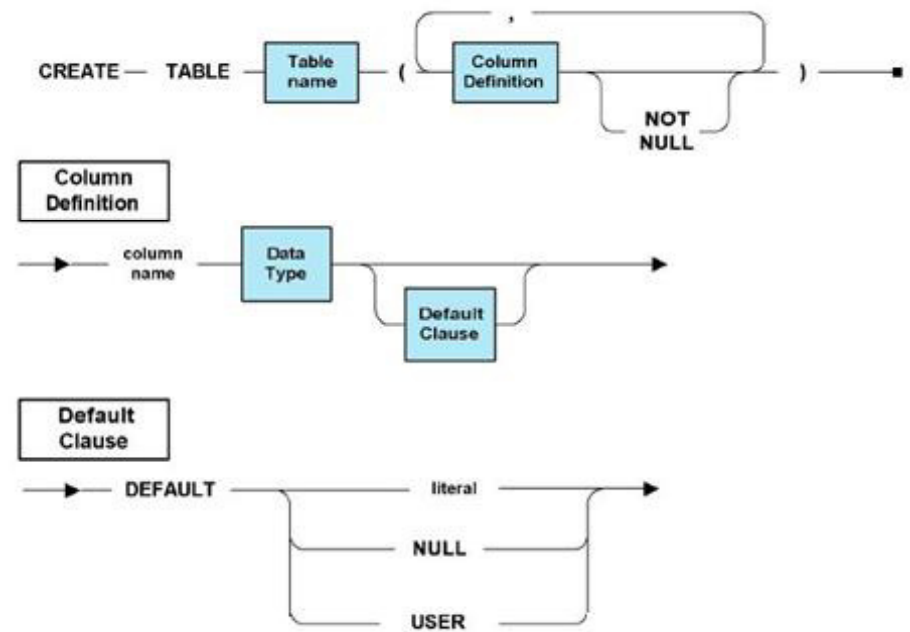


```
DROP TABLE pred;
```



Naredba za definiciju relacije

- ▶ Relacija je dvodimenzionalna tablica čiji redovi predstavljaju n-torke, a kolone attribute (svojstva)
- ▶ Ključ relacije – atribut ili skup atributa čije vrijednosti u relaciji moraju biti jedinstvene
- ▶ Struktura svake relacije mora biti definisana uz pomoć naredbe CREATE TABLE



Privremena relacija

- ▶ Privremena relacija ima sve karakteristike trajne relacije kreirane naredbom `CREATE TABLE`, osim što se ne vidi u riječniku podataka, te biva uništena u trenutku kada se završava program u kojem je kreirana
- ▶ Prvenstveno se koristi za pohranu međurezulata
- ▶ Naredba za kreiranje je `CREATE TEMPORARY TABLE`
- ▶ Naredba kojom se bespovratno uništavaju podaci i struktura relacije ili privremene relacije je:

`DROP TABLE imeTablice`

- ▶ `NULL` vrijednost – oznaka za vrijednost atributa koja je privremeno ili trajno nedostupna
- ▶ `NOT NULL` uz definiciju atributa **A** u relaciji **R** predstavlja pravilo prema kojem nije dozvoljeno obaviti operaciju koja bi u bilo kojoj n-torci relacije **R** postavila za vrijednost atributa **A** `NULL` vrijednost

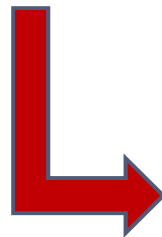


Unos podataka u relaciju

- Unos n-torke (zapisa, reda, *row*) u relaciju

pred			
sifPred	kratPred	nazPred	sifOrgjed

```
INSERT INTO pred
VALUES (123, 'FIZ1', 'Fizika 1', 1);
INSERT INTO pred
VALUES (234, 'MAT1', 'Matematika 1', 1);
INSERT INTO pred
VALUES (345, 'BP', 'Baze podataka', 2);
```



pred			
sifPred	kratPred	nazPred	sifOrgjed
234	MAT1	Matematika 1	1
345	BP	Baze podataka	2
123	FIZ1	Fizika 1	1

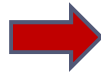


Dohvat podataka iz relacije

pred			
sifPred	kratPred	nazPred	sifOrgjed
234	MAT1	Matematika 1	1
345	BP	Baze podataka	2
123	FIZ1	Fizika 1	1

- ▶ Dohvat podataka o predmetima iz organizacione jedinice sa šifrom 1

```
SELECT * FROM pred  
WHERE sifOrgjed = 1
```



pred			
sifPred	kratPred	nazPred	sifOrgjed
234	MAT1	Matematika 1	1
123	FIZ1	Fizika 1	1

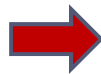


Izmjena podataka u relaciji

pred			
sifPred	kratPred	nazPred	sifOrgjed
234	MAT1	Matematika 1	1
345	BP	Baze podataka	2
123	FIZ1	Fizika 1	1

- Kraticu predmeta sa šifrom 345 promijeniti u BAPO

```
UPDATE pred  
  SET kratPred = 'BAPO'  
 WHERE sifPred = 345
```



pred			
sifPred	kratPred	nazPred	sifOrgjed
234	MAT1	Matematika 1	1
345	BAPO	Baze podataka	2
123	FIZ1	Fizika 1	1

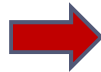


Brisanje podataka iz relacije

pred			
sifPred	kratPred	nazPred	sifOrgjed
234	MAT1	Matematika 1	1
345	BP	Baze podataka	2
123	FIZ1	Fizika 1	1

- Obrisati predmete iz organizacione jedinice sa šifrom 1

```
DELETE FROM pred  
WHERE sifOrgjed = 1
```



pred			
sifPred	kratPred	nazPred	sifOrgjed
345	BP	Baze podataka	2



Četiri osnovne naredbe za upravljanje podacima

- ▶ Postavljanje SQL upita (*query*) nad bazom podataka (pregled podataka)

```
SELECT nazMjesto FROM mjesto  
WHERE pbr=75000 OR pbr=71000
```

- ▶ Unos n-torke (zapisa, reda, *row*) u relaciju

```
INSERT INTO mjesto VALUES (75000,  
'Tuzla')
```

- ▶ Ažuriranje n-torke (n-torki) u relaciji

```
UPDATE mjesto SET nazMjesto='Zenica'  
WHERE pbr=72000
```

- ▶ Brisanje n-torke (n-torki) iz relacije

```
DELETE FROM mjesto  
WHERE pbr=71000
```



Naredba za prenos podataka u datoteku

- ▶ Primjer sadržaja ASCII datoteke

```
71000#Sarajevo#  
75000#Tuzla#
```

- ▶ Opšti oblik naredbe za prenos podataka iz relacije u datoteku u ASCII formatu

```
SELECT  
    [INTO OUTFILE 'file_name'  
    [CHARACTER SET charset_name]  
    export_options  
    | INTO DUMPFILE 'file_name'  
    | INTO var_name [, var_name]]
```

- ▶ Primjer:

```
SELECT * FROM stud INTO OUTFILE  
'C:\\tmp\\studenti.unl' FIELDS TERMINATED BY '#'  
LINES TERMINATED BY '\\n';
```



Naredba za prenos podataka iz datoteke

- ▶ Za prebacivanje podataka iz datoteke u ASCII formatu u relaciju u bazi podataka koristi se naredba `LOAD`

```
LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL] INFILE 'file_name'
  [REPLACE | IGNORE]
  INTO TABLE tbl_name
  [PARTITION (partition_name,...)]
  [CHARACTER SET charset_name]
  [{FIELDS | COLUMNS}
    [TERMINATED BY 'string']
    [[OPTIONALLY] ENCLOSED BY 'char']
    [ESCAPED BY 'char']]
  ]
  [LINES
    [STARTING BY 'string']
    [TERMINATED BY 'string']]
  ]
  [IGNORE number {LINES | ROWS}]
  [(col_name_or_user_var,...)] [
  SET col_name = expr,...]
```

- ▶ Primjer:

```
LOAD DATA INFILE '/tmp/zupanija.unl'
  INTO TABLE zupanija
  FIELDS TERMINATED BY '#'
  LINES STARTING BY '\n' TERMINATED BY '#\r';
```

