

SQL

nastavak

Kreiranje tabele

```
CREATE TABLE naziv_tabele  
(obeležje tip_podatka [NOT NULL]  
{, obeležje tip_podatka [NOT NULL]} )
```

- Prikazani deo sintakse SQL naredbe je isti za sve varijante SQL-a
- Ova naredba poseduje veliki broj dodatnih klauzula, koje se odnose na deklaraciju lokalnih i međurelacionih ograničenja, kao i definisanje parametara fizičke strukture tabele, koji zavise od konkretnog sistema za upravljanje bazama podataka

Kreiranje tabele

- Za svako obeležje tabele se navodi tip podatka, koji zavisi od sistema za upravljanje bazama podataka
- Klauzula NOT NULL obezbeđuje zabranu dodele nula vrednosti obeležju tabele
- Ukoliko se ne naglasi drugačije, za obeležja je dozvoljena dodela nula vrednosti

Kreiranje tabele

```
CREATE TABLE Radnik (  
    mbr NUMERIC ( 10 ) NOT NULL,  
    ime VARCHAR ( 255 ) NOT NULL,  
    prz VARCHAR ( 255 ) NOT NULL,  
    plt NUMERIC ( 10, 2 ) NOT NULL,  
    god DATE NOT NULL,  
    sef NUMERIC ( 10))
```

```
CREATE TABLE Projekat (  
    spr NUMERIC( 10 ) NOT NULL,  
    nap VARCHAR ( 255 ) NOT NULL,  
    ruk NUMERIC( 10 ) NOT NULL)
```

```
CREATE TABLE RadProj (  
    brc NUMERIC( 10 ) NOT NULL,  
    spr NUMERIC( 10 ) NOT NULL,  
    mbr NUMERIC( 10 ) NOT NULL)
```

Ograničenja

- Ograničenja je moguće definisati u okviru same `CREATE TABLE` naredbe ili pomoću naredbe `ALTER TABLE`

Definisanje ograničenja u naredbi CREATE TABLE

- **Primarni ključ**

```
[CONSTRAINT naziv] PRIMARY KEY (obeležje {,  
obeležje})
```

- **Strani ključ**

```
[CONSTRAINT naziv] FOREIGN KEY (obeležje  
{,obeležje}) REFERENCES  
naziv_referentne_tabele (obeležje {,  
obeležje})  
[ON DELETE restrict|cascade|set null|no  
action]  
[ON UPDATE restrict|cascade|set null|no  
action]
```

Definisanje ograničenja u naredbi CREATE TABLE

- **Ograničenja vrednosti obeležja**

```
[CONSTRAINT naziv] CHECK (obelezje uslov)  
CHECK (obelezje IN (lista_vrednosti))
```

- **Jedinstvena vrednost obeležja**

```
[CONSTRAINT naziv] UNIQUE (obelezje {,  
obelezje})
```

```
CREATE TABLE Radnik (  
    mbr NUMERIC( 10 ) NOT NULL,  
    ime VARCHAR ( 255 ) NOT NULL,  
    prz VARCHAR ( 255 ) NOT NULL,  
    plt NUMERIC( 10, 2 ) NOT NULL,  
    god DATE NOT NULL,  
    sef NUMERIC( 10 ),  
    CONSTRAINT PKRadnik0 PRIMARY KEY (mbr)  
)
```

```
CREATE TABLE Projekat (  
    spr NUMERIC( 10 ) NOT NULL,  
    nap VARCHAR ( 255 ) NOT NULL,  
    ruk NUMERIC( 10 ) NOT NULL,  
    CONSTRAINT PKProjekat1 PRIMARY KEY (spr),  
    CONSTRAINT FKProjekat1 FOREIGN KEY (ruk) REFERENCES Radnik (Mbr)  
)
```


Definisanje ograničenja – naredba ALTER TABLE

- Primarni ključ

```
ALTER TABLE naziv_tabele ADD  
  [CONSTRAINT naziv_ograničenja]  
  PRIMARY KEY (obeležje {, obeležje})
```

Definisanje ograničenja – naredba ALTER TABLE

- Strani ključ

```
ALTER TABLE naziv_tabele ADD  
  [CONSTRAINT naziv_ograničenja]  
  FOREIGN KEY (obeležje {, obeležje})  
  REFERENCES naziv_referentne_tabele (obeležje {, obeležje})  
  [ON DELETE restrict|cascade|set null|no action]  
  [ON UPDATE restrict|cascade|set null|no action]
```

```
alter table PROJEKAT add constraint FK_RUKOVODI foreign key  
  (RUK) references RADNIK (MBR) on delete restrict on update  
  restrict;
```

```
alter table RADNIK add constraint FK_JE_SEF foreign key (SEF)  
  references RADNIK (MBR) on delete restrict on update  
  restrict;
```

Još malo o naredi ALTER TABLE

- Pomoću ove naredbe je moguće dodati novo obaležje u tabelu, promeniti tip podatka postojećem obeležju

```
ALTER TABLE naziv_tabele ADD (obeležje tip_podatka, ...)  
ALTER TABLE naziv_tabele MODIFY  
    (obeležje [novi_tip_podatka] [NOT NULL|NULL])
```

Bisanje tabele

```
DROP TABLE <naziv_tabele>
```

Kreiranje indeksa

- Stabla pristupa (indeksi) se zajedno sa klauzulom NOT NULL naredbe CREATE TABLE koriste u cilju implementacije ograničenja ključa i u cilju obezbeđenja efikasnog traženja i pretraživanja podataka tabele
- Osnovna sintaksa naredbe za kreiranje indeksa je:

```
CREATE [UNIQUE] INDEX <naziv_indeksa>  
ON <naziv_tabele>  
(<obeležje> [ASC|DESC]  
{, <obeležje> [ASC|DESC]})
```

Kreiranje indeksa

- Stablo pristupa se realizuje kao jedna od varijanti B-stabla
- Klauzula UNIQUE ukazuje da će indeks, a samim tim i tabela, sadržati jedinstvene vrednosti nad navedenim nazivom obeležja
- Ukoliko se drugačije ne naglasi, indeks, odnosno tabela, može sadržati višestruke pojave istih vrednosti navedenih obeležja
- Redosled obeležja u naredbi CREATE INDEX je važan i ukazuje na uređenje podataka, pokrivenih indeksom, koje može biti rastuće ili opadajuće

Ažuriranje baze podataka putem jezika SQL

- Ažuriranje baze podataka se realizuje putem naredbi:
 - za dodavanje novih toriki u tabelu (INSERT)
 - za brisanje toriki iz tabele (DELETE)
 - za modifikaciju toriki u tabeli (UPDATE)

Dodavanje nove torke

```
INSERT INTO naziv_tabele [(lista_obeležja)]  
VALUES (lista_konstanti) | SELECT upit
```

- lista_obeležja predstavlja skup obeležja tabele sa nazivom naziv_tabele
- Ukoliko se lista_obeležja ne navede, podrazumeva se redosled obeležja koji je utvrđen prilikom kreiranja tabele
- Vrednosti svih obeležja koja nemaju dozvoljene nula vrednosti, moraju biti navedene

Dodavanje nove torke

```
insert into radnik (mbr, ime, prz, plt, sef, god)
  values (101, 'Ana', 'Savic', 30000, null,
    '1971-08-18')
```

```
insert into projekat (spr, nap, ruk)
  values ('10', 'P1', 201)
```

```
insert into radproj (mbr, spr, brc)
  values (101, '10', 5)
```

Brisanje postojećih torki

```
DELETE FROM naziv_tabele  
  [WHERE uslov_selekcije]
```

- Ukoliko se uslov selekcije ne navede, sve torke iz tabele će biti izbrisane
- Brišu se samo kompletne torke, nije moguće izbrisati vrednosti samo određenih obeležja
- ```
DELETE FROM radnik WHERE plt<20000
```

# Modifikacija postojećih toraki

- **Primenjuje se u slučajevima kada se menja vrednost određenog obeležja u torci, a ne kompletna toraka**

```
UPDATE naziv_tabele
SET obeležje=aritmetički_izraz
{, obeležje=aritmetički_izraz}
[WHERE uslov_selekcije]
```

- **U aritmetičkom izrazu se koriste samo obeležja iz tabele nad kojom se vrši promena**

```
UPDATE naziv_tabele
SET (lista_obeležja)=(SELECT upit)
{, (lista_obeležja)=(SELECT upit)}
[WHERE uslov_selekcije]
```

# Modifikacija postojećih toraki

- Povećati platu radnicima za 20%.

```
UPDATE radnik
SET PLT=PLT*1.2
```

- Povećati platu za 20% onim radnicima čija je plata manja od 25000.

```
UPDATE radnik
SET PLT=PLT*1.2
WHERE PLT<25000
```

# Transakcije

- `commit` – potvrđuje tekuću transakciju, promene koje su se dogodile se zapisuju u bazu podataka
- `rollback` – opoziva tekuću transakciju, promene koje su se dogodile neće biti upisane u bazu, odnosno, baza podataka se vraća u stanje u kom je bila pre početka tekuće transakcije

# Kreiranje pogleda

- Pogled predstavlja šemu relacije koja služi za informaciono pokrivanje jednog korisničkog zahteva

```
CREATE VIEW naziv_pogleda [(lista_obeležja)] AS
SELECT upit
```

# Kreiranje pogleda

- Kreirati pogled Šefovi(mbr, ime, prezime) koji će sadržati podatke o šefovima radnika.

```
create view Sefovi(mat_broj, ime, prezime) as
select distinct r1.mbr, r1.ime, r1.prz from radnik r1, radnik
 r2
where r1.mbr=r2.sef;
select * from sefovi;
```

- Kreirati pogled RukovodiociProjekata.

```
create view RukovodiociProjekata(mat_broj, ime, prezime) as
select distinct mbr, ime, prz from radnik, projekat
where mbr=ruk;
```

```
select * from RukovodiociProjekata;
```

- Prikazati mbr, ime, i prz rukovodilaca projekata, kao i broj projekata kojima rukovode.

```
select rp.mat_broj, rp.ime, rp.prezime, count(*) from
 rukovodiociprojekata rp, projekat p
where rp.mat_broj=p.ruk
group by rp.mat_broj, rp.ime, rp.prezime;
```

# Kreiranje pogleda

- Kreirati pogled VelikeZarade koji će sadržati podatke o radnicima koji zarađuju više od proseka.

```
create view VelikeZarade
as select mbr, ime, prz, plt from radnik
where plt > (select avg(plt) from radnik);
select * from VelikeZarade;
```

- Kreirati pogled VredniRadnici koji će sadržati podatke o radnicima koji rade na više od dva projekta.

```
create view VredniRadnici(ime, prezime, uk_br_proj, uk_ang) as
select ime, prz, count(spr), sum(brc)
from radnik r, radproj rp
where r.mbr=rp.mbr
group by r.mbr, ime, prz
having count(spr) > 2;
```

- Izlistati ime i prezime vrednog radnika koji je angažovan više od 12 časova.

```
select ime, prezime from vredniradnici vr
where uk_ang > 12;
```



# Brisanje pogleda

```
DROP VIEW <naziv_pogleda>
```