

Fakultet tehničkih nauka, DRA, Novi Sad

Predmet:

Baze podataka 1

Dr Ivan Luković

Milan Čeliković

Vladimir Dimitrieski

Branko Terzić

Angelina Vujanović

Nikola Todorović

Vladimir Ivković

Aleksa Mirković

Primer

**radnik({Mbr, Ime, Prz, Sef, Plt, God, Pre},
{Mbr}),**

projekat({Spr, Ruk, Nap, Nar}, {Spr}),

radproj({Spr, Mbr, Brc}, {Spr + Mbr}),

radnik[Sef] \subseteq radnik[Mbr],

projekat[Ruk] \subseteq radnik[Mbr],

radproj[Mbr] \subseteq radnik[Mbr],

radproj[Spr] \subseteq projekat[Spr].

Tabela radnik

- **Mbr** - maticni broj radnika
- **Ime** - ime radnika
- **Prz** - prezime radnika
- **Sef** - maticni broj direktno nadređenog rukovodioca - radnika
- **Plt** - mesecni iznos plate radnika
- **God** - Datum rođenja radnika
- **Pre** – godišnja premija na platu radnika

Obeležja Mbr, Ime, Prz ne smeju imati null vrednost.
Plata ne sme biti manja od 500

Tabela projekt

- **Spr** - sifra projekta
- **Ruk** - rukovodilac projekta
- **Nap** - naziv projekta
- **Nar** - narucilac projekta

Obeležja Spr i Ruk ne smeju imati null vrednost, dok obeležje Nap mora imati jedinstvenu vrednost

Tabela radproj

- **Spr** - sifra projekta
- **Mbr** - maticni broj radnika
- **Brc** - broj casova nedeljnog angazovanja na projektu

Sva tri obeležja ne smeju da imaju null vrednost

Kreiranje tabele

```
CREATE TABLE [šema.]<naziv_tabele>  
(<naziv_kolone> <tip_podatka> [DEFAULT  
izraz] [, ...]  
    CONSTRAINT <naziv_ogranicenja>  
    <definicija_ogranicenja> [, ...]);
```

- šema – poklapa se sa nazivom korisnika
- **DEFAULT opcija:**
 - Specificira se predefinisana vrednost za kolonu, koja se koristi ukoliko se prilikom ubacivanja podataka izostavi vrednost za tu kolonu

SQL tipovi podataka

Data	TypeDescription
– VARCHAR2(size)	niz karaktera promenljive dužine, maksimalne dužine <i>size</i> ; minimalna dužina je 1, maksimalna je 4000
– CHAR(size)	Niz karaktera fiksne dužine od <i>size</i> bajtova; default i minimalna dužina je 1, maksimalna dužina je 2000
– NUMBER(p,s)	broj ukupnog broja cifara p, od čega je s cifara iza decimalnog zareza; p može imati vrednosti od 1 do 38
– DATE	vrednosti za vreme i datum
– LONG	niz karaktera promenljive dužine do 2 GB
– CLOB	Karakter do 4 GB
– BLOB	
– BFILE	binarni podaci smešteni u eksternom fajlu do 4 GB
– ROWID	jedinstvena adresa vrste u tabeli

Tabela radnik

```
CREATE TABLE radnik
(
  Mbr integer NOT NULL,
  Ime varchar2(20) NOT NULL,
  Prz varchar2(25) NOT NULL,
  Sef integer,
  Plt decimal(10, 2),
  Pre decimal(6, 2),
  God date NOT NULL,
  CONSTRAINT radnik_PK PRIMARY KEY (Mbr),
  CONSTRAINT radnik_FK FOREIGN KEY (Sef) REFERENCES
Radnik (Mbr),
  CONSTRAINT radnik_CH CHECK (Plt>500)
);
```


Tabela projekat

```
CREATE TABLE projekat
(
  Spr integer not null,
  Ruk integer not null,
  Nap varchar2(30),
  Nar varchar2(30),
  CONSTRAINT projekat_PK PRIMARY KEY (Spr),
  CONSTRAINT projekat_FK FOREIGN KEY (Ruk)
  REFERENCES Radnik (Mbr),
  CONSTRAINT projekat_UK UNIQUE (Nap)
);
```

Tabela radproj

```
CREATE TABLE radproj
(
  Spr integer NOT NULL,
  Mbr integer NOT NULL,
  Brc integer NOT NULL,
  CONSTRAINT radproj_PK PRIMARY KEY (Spr, Mbr),
  CONSTRAINT radproj_rad_FK FOREIGN KEY (Mbr)
REFERENCES radnik(Mbr),
  CONSTRAINT radproj_prj_FK FOREIGN KEY (Spr)
REFERENCES projekat(Spr)
);
```

Tabela faze_projekta

- Kreirati tabelu faze_projekta

**faze_projekta({Spr , Sfp, Rukfp, Nafp,
Datp}, {Spr+ Sfp})**

faze_projekta[Spr] \subseteq projekat[Spr],

faze_projekta[Rukfp] \subseteq radnik[Mbr]

Tabela faze_projekta

- **Sfp** - šifra faze projekta,
- **Spr** - sifra projekta,
- **Rukfp** - rukovodilac faze projekta,
- **Nafp** - naziv faze projekta,
- **Datp** - datum početka faze projekta

Obeležja Spr i Sfp ne smeju imati null vrednost.

Obeležje Nafp mora imati jedinstvenu vrednost.

Izmena definicije tabele

- **ALTER TABLE**

Alter table iskaz služi za:

- dodavanje nove kolone,
- modifikaciju postojeće kolone,
- definisanje podrazumevane vrednosti za novu kolonu,
- brisanje kolone
- dodavanje oraničenja.

ALTER TABLE

```
ALTER TABLE <naziv_tabele>  
ADD (<naziv_kolone> <tip_podatka> [DEFAULT izraz]  
[, <naziv_kolone> <tip_podatka>]...);
```

```
ALTER TABLE <naziv_tabele>  
MODIFY (<naziv_kolone> <tip_podatka> [DEFAULT izraz] [,  
    <naziv_kolone> <tip_podatka>]...);
```

```
ALTER TABLE <naziv_tabele>  
DROP COLUMN (<naziv_kolone>);
```

```
ALTER TABLE <naziv_tabele>  
ADD CONSTRAINT <naziv_ogranicenja>  
<definicija_ogranicenja>;
```

Izmena definicije tabele

- U tabelu faze_projekta dodati atribut:

Datz - datum završetka faze projekta.

- **Datz** ne sme biti manji od **Datp**

Podaci za faze_projekta

- U tabelu faze_projekta dodati bar dve faze za jedan projekat i jednu za drugi projekat

Zadatak za vežbu

- Za svaki projekat prikazati sifru projekta, naziv projekta, ime i prezime rukovodioca projekta, prezime njegovog šefa, nazive faza projekta, imena i prezimena rukovodioca faza projekta. Ako projekat nije podeljen u faze napisati: nema faze.

Brisanje definicije tabele

DROP TABLE <naziv_tabele>;

Brisanje definicije tabele

- Izbrisati tabelu faze_projekta.

Kreiranje, izmena i brisanje definicije pogleda

**CREATE [OR REPLACE] VIEW
<naziv_pogleda> [(alias [, alias]...)]
AS podupit;**

- Podupit koji se koristi za definisanje pogleda može biti kompleksan

Modifikacija pogleda

- Pogledi se modifikuju pomoću OR REPLACE opcije (kreira se pogled, a ako pogled sa tim imenom već postoji, nova definicija zamenjuje staru).
- Dakle, pogled može biti izmenjen bez brisanja postojećeg pogleda.
- Na primer, mogu se dodati alijasi za kolone u pogledu.

Kreiranje složenog pogleda

- Ukoliko se u upitu pomoću kog se kreira pogled nalaze skupovne funkcije (min, max, avg, sum, count) ili izrazi, u pogledu se moraju definisati alternativna imena za te kolone.

DML operacije sa pogledima

- DML (Data Manipulation Language) operacije se mogu primenjivati na jednostavnim pogledima.
- Ako pogled sadrži **skupovne funkcije**, **group by** kaluzulu, **distinct** rezervisanu reč ili **rownum** rezervisanu reč, vrsta iz pogleda se ne može izbrisati.
- Isto važi i za modifikaciju podataka, s tim što dodatno važi i da se **kolone definisane izrazima** ne mogu modifikovati (npr, salary*12).
- U pogled se ne mogu dodavati podaci ako pogled sadrži **skupovne funkcije**, **group by** kaluzulu, **distinct** rezervisanu reč, **rownum** rezervisanu reč, **kolonu koja je definisana izrazom**, ili **not null** kolonu u baznoj tabeli koja nije selektovana u pogledu.
- Dodavanjem vrednosti u pogled, one se dodaju direktno u baznu tabelu.

Brisanje pogleda

DROP VIEW pogled;

Pogledi obezbeđuju sledeće prednosti

- ograničavaju pristup bazi podataka
- pojednostavljaju upite
- obezbeđuju nezavisnost podataka
- obezbeđuju višestruke poglede nad istim podacima
- mogu se brisati bez uklanjanja podataka u osnovnim tabelama.

Pogled

- Napraviti pogled koji će za sve radnike prikazati samo njihova imena, prezimena i platu.

```
CREATE OR REPLACE VIEW  
plate_radnika (Ime, Prezime, Plata) AS  
SELECT Ime, Prz, Plt  
FROM radnik;
```

Pogled

- Napraviti pogled koji će za sve radnike prikazati Mbr i ukupan broj sati angažovanja radnika na projektima na kojima radi.

Pogled

```
CREATE OR REPLACE VIEW angaz_po_radnicima  
(Mbr, SBrc) AS  
SELECT r.Mbr, NVL(SUM(rp.Brc), 0)  
FROM radnik r, radproj rp  
WHERE r.Mbr = rp.Mbr (+)  
GROUP BY r.Mbr
```

Pogled

- Napraviti pogled koji će za svakog šefa prikazati njegov matični broj, prezime, ime, ukupan broj radnika kojima rukovodi i njegovo ukupno angažovanje na svim projektima, na kojima radi. Koristiti prethodno definisani pogled.

Pogled

```
CREATE VIEW angaz_sefova (Mbr, Prz, Ime, BrRad,  
BrSat) AS  
SELECT r.Sef, r1.Prz, r1.Ime, COUNT(*), a.SBrc  
FROM radnik r, radnik r1, angaz_po_radnicima a  
WHERE r.Sef = r1.Mbr AND r.Sef = a.Mbr  
GROUP BY r.Sef, r1.Prz, r1.Ime, a.SBrc
```

Pogled

- Koliko je ukupno angažovanje svih šefova na projektima?

```
SELECT SUM(BrSat) AS UkAngSef  
FROM angaz_sefova
```

Sekvenca

- automatski generiše jedinstvene brojeve
- najčešće se koristi za kreiranje primarnih ključeva
- sekvenca se generiše i čuva nezavisno od tabele, tako da se jedna sekvenca može koristiti za više tabela

Sekvencer (Generator sekvence vrednosti)

```
CREATE SEQUENCE sequence  
  [INCREMENT BY n]  
  [START WITH n]  
  [{MAXVALUE n | NOMAXVALUE}]  
  [{MINVALUE n | NOMINVALUE}]  
  [{CYCLE | NOCYCLE}]  
  [{CACHE n | NOCACHE}]
```

```
ALTER SEQUENCE sequence ...
```

```
DROP SEQUENCE sequence
```

Primer upotrebe sekvencera

```
CREATE SEQUENCE SEQ_Mbr  
    INCREMENT BY 10  
    START WITH 240  
    NOCYCLE  
    CACHE 10;
```

```
INSERT INTO radnik (Mbr, Prz, Ime, God)  
VALUES (SEQ_Mbr.NEXTVAL, 'Misic',  
    'Petar', SYSDATE);
```

Primer upotrebe sekvencera

```
SELECT SEQ_Mbr.CURRVAL  
FROM SYS.DUAL;
```

Tabele u Oracle bazi podataka

- korisničke tabele
 - kolekcije tabela koje kreira i održava korisnik
 - sadrže korisničke informacije
- **Data Dictionary** (rečnik podataka)
 - kolekcija tabela koje kreira i održava Oracle server
 - sadrže informacije baze podataka
 - vlasnik svih tabela u rečniku je SYS korisnik
 - informacije smeštene u rečniku podataka obuhvataju imena korisnika Oracle servera, privilegije dodeljene korisnicima, nazive objekata baze podataka, ograničenja.
 - postoji nekoliko kategorija pogleda rečnika podataka; svaka od njih ima odgovarajući prefiks:
 - USER_ - ovi pogledi sadrže informacije o objektima čiji je vlasnik korisnik
 - ALL_ - ovi pogledi sadrže informacije o svim tabelama (objektnim i relacionim) koje su dostupne korisniku
 - DBA_ - ovi pogledi su zabranjeni, tj. dostupni su samo korisnicima koji imaju DBA ulogu

Tabele u Oracle bazi podataka

- Upiti u rečniku podataka se postavljaju kao i svi ostali upiti.
- Prikazati nazive tabela čiji je vlasnik korisnik.

```
SELECT table_name  
FROM user_tables;
```
- Prikazati različite tipove objekata čiji je vlasnik korisnik.

```
SELECT DISTINCT object_type  
FROM user_objects;
```
- Prikazati tabele, poglede, sinonime i sekvence čiji je vlasnik korisnik.

```
SELECT *  
FROM user_catalog;
```

Neke karakter funkcije

- LOWER(char) – za konvertovanje svih znakova u mala slova
- UPPER(char) – za konvertovanje svih znakova u velika slova
- INITCAP(char) – prvo slovo svake reči u nizu znakova pretvara u veliko slovo, a ostatak reči u mala slova
- SUBSTR(char, m [,n]) – koristi se za izdvajanje dela niza znakova
- TRIM(LEADING | TRAILING | BOTH trim_character FROM trim_source) – uklanja početne ili prateće znakove sa početka ili kraja niza znakova
- LENGTH(char) – vraća broj znakova u nizu

Neke karakter funkcije - primeri

LOWER ('Sva mala slova') → 'sva mala slova'

UPPER ('Sva velika slova') → 'SVA VELIKA SLOVA'

INITCAP('Velika početna slova') → ' Velika Početna Slova'

SUBSTR('DobroJutro',1,5) → 'Dobro'

TRIM('D' FROM 'DobroJutro') → 'obroJutro'

LENGTH('DobroJutro') → 10

Neke karakter funkcije - primer

```
SELECT Mbr, Prz, Ime  
FROM Radnik  
WHERE UPPER(Prz) = 'PETRIC';
```


Neke funkcije za konverziju podataka

- `TO_CHAR(d [, fmt])` – transformiše vrednosti tipa `DATE` u `VARCHAR2`, po izboru uz navedeni format datuma
- `TO_CHAR(n [, fmt])` – transformiše vrednost brojanog tipa u `VARCHAR2`, po izboru uz navedeni format broja
- `TO_DATE(char [, fmt])` – za konvertovanje niza znakova u ekvivalentni datum
- `TO_NUMBER(char [,fmt])` – za konvertovanje znakovnih vrednosti u numeričke

Sparse matrice

- Predstavljaju matrice gde većina elemenata sadrži vrednost 0.
- Velike sparse matrice se pojavljuju u naučnim proračunima prilikom rešavanja parcijalnih diferencijalnih jednačina

Sparse matrice

- Primer množenja dve sparse matrice

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 9 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Množenje sparse matrica

```
SELECT a.row_num, b.col_num,  
SUM(a.value*b.value)  
FROM a, b  
WHERE a.col_num = b.row_num  
GROUP BY a.row_num, b.col_num;
```

Sabiranje sparse matrica

```
SELECT a.row_num, a.col_num, a.value  
FROM a  
WHERE NOT EXISTS (SELECT 0 FROM b  
WHERE a.col_num = b.col_num and  
a.row_num = b.row_num)
```

Sabiranje sparse matrica

UNION

SELECT b.row_num, b.col_num, b.value

FROM b

WHERE NOT EXISTS (SELECT 0 FROM a

WHERE a.col_num = b.col_num and

a.row_num = b.row_num)

Sabiranje sparse matrica

UNION

**SELECT a.row_num, a.col_num, a.value +
b.value**

FROM a,b

**WHERE a.col_num = b.col_num and
a.row_num = b.row_num;**