

Fakultet tehničkih nauka, Novi Sad

Predmet:

Baze podataka

dr Slavica Kordić

Nikola Todorović

Marko Vještica

Miroslav Tomić

Gergelj Kiš

Aleksandar Hadžibabić

Rad u učionici (1/2)

- Baze podataka:
 - Studentska korisnička šema (user schema)
 - pod nazivom **raXY**
 - **User name:** raXY
 - **Password:** ftn
 - gde je *X* broj indeksa, a *Y* godina upisa

Rad u učionici (2/2)

- Podaci potrebni za konektovanje na bazu podataka:

	MI A2-1, MI A2-2, MI A2-3	Ostale učionice	Kući
Host Name	192.168.0.102	192.168.7.204	localhost
Port Number	1522	1521	1521
Oracle SID, database name	db2016	bp1	xe

SQL (Structured Query Language)

- **SQL**

- standardni jezik relacionih sistema za upravljanje bazama podataka
- jezik visokog nivoa deklarativnosti
- objedinjuje funkcije jezika za definiciju podataka, jezik za manipulaciju podacima i upitni jezik

SQL (Structured Query Language)

- **Namena i zadaci SQL-a u okviru sistema za upravljanje bazama podataka**
 - administratorima baze podataka za obavljanje poslova administracije
 - programerima za izradu aplikacija nad bazom podataka
 - krajnjim korisnicima, za postavljanje upita nad bazom podataka

SQL (Structured Query Language)

- **SQL se javlja u formama:**
 - interaktivnog jezika sistema za upravljanje bazama podataka
 - ugrađenog jezika u jezik III generacije
 - sastavnog dela jezika IV generacije

SQL (Structured Query Language)

- **Saglasno nameni i vrstama korisnika koji ga upotrebljavaju, SQL obezbeđuje realizaciju sledećih zadataka:**
 - izražavanje upita putem upitnog jezika (naredba SELECT)
 - ažuriranje baze podataka putem jezika za manipulaciju podacima (naredbe INSERT, DELETE i UPDATE)
 - realizacija implementacione šeme baze podataka i definisanje fizičke organizacije baze podataka (naredbe CREATE, DROP i ALTER)
 - automatsko održavanje rečnika podataka

SQL (Structured Query Language)

- **Saglasno nameni i vrstama korisnika koji ga upotrebljavaju, SQL obezbeđuje realizaciju sledećih zadataka:**
 - transakcijska obrada podataka (naredbe COMMIT, ROLLBACK, SAVEPOINT)
 - zaključavanje resursa (naredba LOCK TABLE)
 - zaštita podataka od neovlašćenog pristupa (naredbe GRANT, REVOKE)
 - praćenje zauzeća resursa i performansi rada sistema za upravljanje bazama podataka (naredbe AUDIT, EXPLAIN PLAN)
 - obezbeđenje proceduralnog načina obrade podataka "slog po slog" (naredbe za rad sa kursorom: OPEN, FETCH, CLOSE)

SQL (Structured Query Language)

- Sintaksa SQL-a zavisi od proizvođača sistema za upravljanje bazama podataka.

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 radnik

```
CREATE TABLE radnik(  
    Mbr integer NOT NULL,  
    Ime varchar(20) NOT NULL,  
    Prz varchar(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 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 projekat

```
CREATE TABLE projekat
```

```
(
```

```
    Spr integer not null,
```

```
    Ruk integer not null,
```

```
    Nap varchar(30),
```

```
    Nar varchar(30),
```

```
    CONSTRAINT projekat_PK PRIMARY KEY (Spr),
```

```
    CONSTRAINT projekat_FK FOREIGN KEY (Ruk)  
        REFERENCES Radnik (Mbr),
```

```
    CONSTRAINT projekat_UK UNIQUE (Nap)
```

```
);
```

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

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)
);
```


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

Naziv tabele i kolone

- mora početi slovom,
- mora biti između 1 i 30 znakova dužine,
- mora sadržati samo velika i mala slova, cifre, _, \$ i #,
- ne sme se poklapati sa nazivom nekog drugog objekta koji je kreirao isti korisnik,
- ne sme biti rezervisana reč Oracle servera.
- Nazivi nisu case sensitive.

SQL tipovi podataka

Tip podatka

Opis

- **VARCHAR2(size)** niz karaktera promenljive dužine, maksimalne dužine *size*; minimalna dužina je 1, maksimalna je 4000
- **CHAR(size)** Niz karaktera fiksne dužine od *size* 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
 - za kompatibilnost sa starijim verzijama Oracle-a
- **CLOB** niz karaktera promenljive dužine do 4 GB
- **BLOB** binarni podaci do 4 GB
- **BFILE** binarni podaci smešteni u eksternom fajlu do 4 GB
- **ROWID** jedinstvena adresa vrste u tabeli

Izražavanje upita i osnovna struktura naredbe SELECT

- Sve vrste upita se u SQL-u izražavaju putem naredbe SELECT. Osnovna struktura SELECT naredbe je:

```
SELECT *| <lista_obeležja>  
FROM <lista_tabela>  
[WHERE <uslov_selekcije>]
```

<lista_obeležja> sadrži obeležja nad kojima se formira rezultat upita, <lista_tabela> sadrži nazive tabela potrebne za realizaciju upita, <uslov_selekcije> izražava uslov selekcije podataka iz tabela koje su navedene iza službene reči FROM

Upiti nad jednom tabelom

- Izlistati sadržaj svih tabela.

select * from radnik;

select * from projekat;

select * from radproj;

Upiti nad jednom tabelom

- Prikazati imena i prezimena svih radnika.

```
select ime, prz  
from radnik;
```

DISTINCT

```
SELECT [DISTINCT] <lista_obeležja>  
FROM <lista_tabela>  
WHERE <uslov_selekcije>
```

- Izlistati različita imena radnika.

```
select distinct ime from radnik;
```

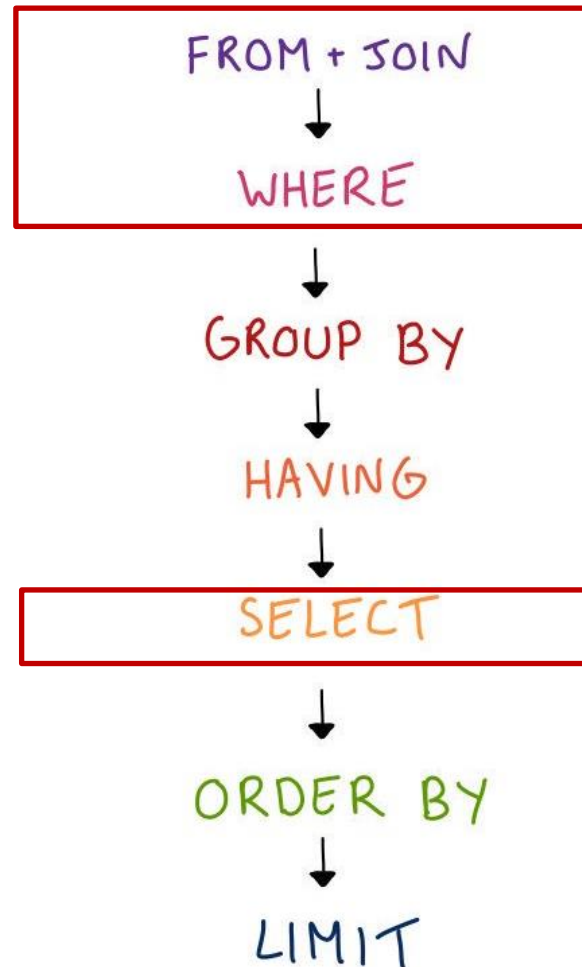
WHERE <uslov_selekcije>

- Izlistati mbr, ime i prezime radnika koji imaju platu veću od 25000.

```
select mbr, ime, prz  
from radnik  
where plt>25000;
```


Redosled izvršavanja klauzula

Redosled izvršavanja SQL klauzula



Aritmetički izrazi

- Izlistati godišnju platu svakog radnika.

```
select mbr, ime, prz, plt*12  
from radnik;
```

Null vrednost

x IS NULL – x je nula vrednost

x IS NOT NULL – x nije nula vrednost

- Izlistati mbr, ime, prz radnika koji nemaju šefa.

**SELECT mbr, ime, prz from radnik
where sef is null;**

BETWEEN

- Izlistati mbr, ime, prz radnika čija je plata između 20000 i 24000 dinara.

```
select mbr, ime, prz  
from radnik  
where plt between 20000 and 24000;
```

- Razlika između between i < >?

BETWEEN

- Izlistati ime, prz, god radnika rođenih između 1953 i 1975.

select ime, prz, god

from radnik

where god between '01-jan-1953' and '31-dec-1975';

NOT BETWEEN

- Izlistati ime, prz, god radnika koji nisu rođeni između 1953 i 1975.

select ime, prz, god

from radnik

**where god not between '01-jan-1953' and
'31-dec-1975';**

LIKE

<obeležje> LIKE <uzorak>

- Izlistati mbr, ime, prz radnika čije prezime počinje na slovo M.

```
select mbr, ime, prz  
from radnik  
where prz like 'M%';
```

NOT LIKE

- Izlistati mbr, ime, prz radnika čije ime ne počinje slovom A.

```
select mbr, ime, prz  
from radnik  
where ime not like 'A%';
```


LIKE

- Izlistati mbr, ime, prz radnika čije ime sadrži slovo **a** na drugoj poziciji.

```
select mbr, ime, prz  
from radnik  
where ime like '_a%';
```

LIKE

- Izlistati imena radnika koja počinju na slovo E. Imena ne bi trebalo da se ponavljaju.

```
select distinct ime  
from radnik  
where ime like 'E%';
```

LIKE

- Izlistati radnike koji u svom imenu imaju slovo E (e).

```
select mbr, ime, prz  
from radnik  
where ime like '%e%' or ime like  
'%E%';
```

IN

- Izlistati matične brojeve radnika koji rade na projektima sa šifrom 10, 20 ili 30.

```
select distinct mbr  
from radproj  
where spr in (10, 20, 30);
```

IN

- Izlistati matične brojeve radnika koji rade na projektu sa šifrom 10 ili rade 2, 4, ili 6 sati.

```
select distinct mbr  
from radproj  
where brc in (2, 4, 6) or spr='10';
```

NOT IN

- Izlistati matične brojeve radnika koji se ne zovu Ana ili Sanja.

```
select mbr, ime, prz  
from radnik  
where ime not in ('Ana', 'Sanja');
```

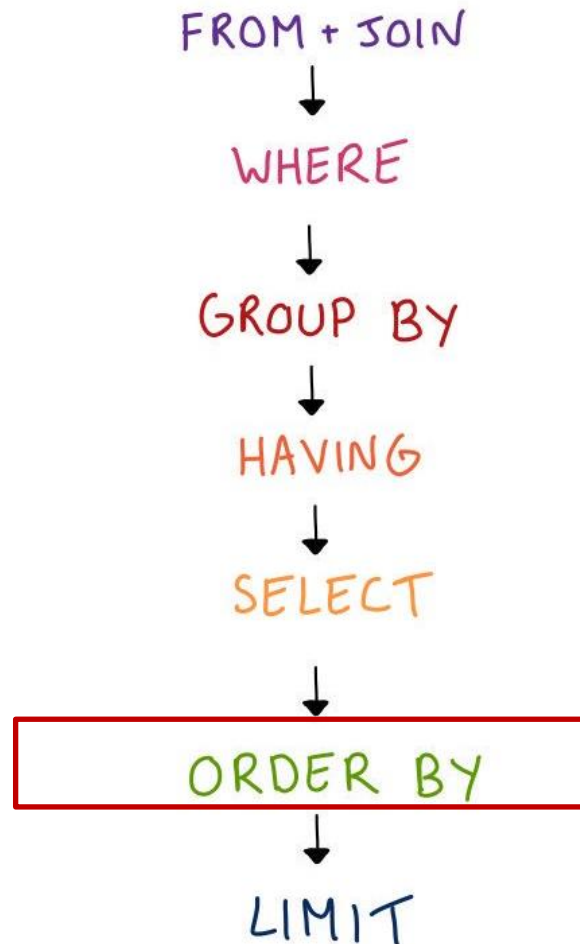
Uređivanje izlaznih rezultata

```
SELECT *| <lista_obeležja>  
FROM <lista_tabela>  
WHERE <uslov_selekcije>  
ORDER BY <podlista_obeležja>
```

ORDER BY je uvek poslednja klauzula naredbe SELECT

Redosled izvršavanja klauzula

Redosled izvršavanja SQL klauzula



ORDER BY

- Prikazati radnike koji imaju šefa sortirano po prezimenu.

```
select mbr, ime, prz, plt  
from radnik  
where sef is not null  
order by prz asc;
```

ORDER BY

- Neki primeri upotrebe klauzule ORDER BY.

```
SELECT Mbr, Ime, Prz, Plt  
FROM Radnik ORDER BY Prz, Ime;
```

```
SELECT Mbr, Prz, Ime, Plt  
FROM Radnik ORDER BY Prz ASC, Ime ASC;
```

```
SELECT Mbr, Prz, Ime, Plt  
FROM Radnik ORDER BY Prz ASC, Ime DESC;
```

ORDER BY

- Neki primeri upotrebe klauzule ORDER BY.

```
SELECT Mbr, Prz, Ime  
FROM Radnik  
ORDER BY 2, 3, Plt;
```

```
SELECT Mbr, Prz, Ime  
FROM Radnik  
ORDER BY 2, 3, Plt * 1.17;
```

ORDER BY

- Prikazati matične brojeve, imena, prezimena i plate radnika, po opadajućem redosledu iznosa plate.

```
SELECT Mbr, Ime, Prz, Plt Plata  
FROM Radnik  
ORDER BY Plata DESC;
```

Uređivanje izlaznih rezultata

- Prikazati matične brojeve, **spojena** (konkatenirana) imena i prezimena radnika, kao i plate, uvećane za 17%.

```
SELECT Mbr,  
Ime || ' ' || Prz "Ime i prezime",  
Plt * 1.17 Plata  
FROM Radnik;
```

- **Concat funkcija**

Zadatak

- Prikazati radnike čije prezime sadrži ime.
Na primer Marko Marković, ili Djordje Karadjordjevic
 - Funkcije UPPER, LOWER

```
SELECT * from radnik where LOWER(prz)  
LIKE '%' || LOWER(ime) || '%'
```

ANY

$x \Theta \text{ ANY } (<\text{lista_vrednosti}>)$

$\Theta \in \{<, >, <=, >=, !=, =\}$

Primer:

$x = \text{ANY } (<\text{lista_vrednosti}>)$

x je jednako makar jednoj vrednosti u
<listi_vrednosti>

ANY

- Prikazati matične brojeve radnika, imena i prezimena i platu radnika koji se zovu Pera ili Moma.

```
SELECT Mbr, Ime, Prz, Plt  
FROM Radnik  
WHERE Ime = ANY ('Pera', 'Moma');
```


ALL

$x \Theta \text{ALL} (<\text{lista_vrednosti}>)$

$\Theta \in \{<, >, <=, >=, !=, =\}$

Primer:

$x \neq \text{ALL} (<\text{lista_vrednosti}>)$

x je različito od svake vrednosti u
<listi_vrednosti>

ALL

- Prikazati matične brojeve radnika, imena i prezimena i platu radnika koji se ne zovu Pera ili Moma.

```
SELECT Mbr, Ime, Prz, Plt  
FROM Radnik  
WHERE Ime !=ALL ('Pera', 'Moma');
```

Upotreba skupovnih funkcija

- Prikazati matične brojeve radnika, kao i plate, uvećane za NULL vrednost.

SELECT Mbr, Plt + NULL FROM Radnik;

- Prikazati matične brojeve radnika, kao i plate, uvećane za godišnju premiju.

SELECT Mbr, Plt + Pre FROM Radnik;

Funkcija NVL(*izraz, konstanta*)

- Prikazati matične brojeve radnika, kao i plate, uvećane za godišnju premiju. Ukoliko za nekog radnika vrednost premije ne postoji, smatrati da ona iznosi 0.

```
SELECT Mbr, Plt + NVL(Pre, 0)  
FROM Radnik;
```

Funkcija count

- COUNT(*) – vraća ukupan broj selektovanih torki
- COUNT(<obeležje>) – vraća ukupan broj selektovanih torki, za koje vrednost <obeležja> nije nula vrednost
- COUNT(DISTINCT <obeležje>) – vraća ukupan broj različitih torki, za koje vrednost <obeležja> nije nula vrednost

Funkcija count

- Koliko ima radnika?

```
select count(*) from radnik;
```

- Koliko ima šefova?

```
select count(distinct sef) broj_sefova  
from radnik;
```

Funkcije max i min

- $\text{MAX}(\langle \text{obeležje} \rangle)$ – vraća maksimalnu vrednost za $\langle \text{obeležje} \rangle$, uzimajući u obzir sve selektovane torke
- $\text{MIN}(\langle \text{obeležje} \rangle)$ – vraća minimalnu vrednost za $\langle \text{obeležje} \rangle$, uzimajući u obzir sve selektovane torke

Funkcije max i min

- Prikazati minimalnu i maksimalnu platu radnika.

```
select min(plt) minimalna, max(plt) maksimalna  
from radnik;
```


Funkcija sum

- SUM(<obeležje>) – vraća zbir vrednosti datog <obeležja>, za sve selektovane torke, uključujući višestruko ponavljanje istih torke
- SUM(DISTINCT <obeležje>) – vraća zbir vrednosti datog <obeležja>, za sve različite selektovane torke

Funkcija sum

- Prikazati broj radnika i ukupnu mesečnu platu svih radnika.

```
select count(*) "Broj radnika",  
sum(plt) "Ukupna mesecna plata"  
from radnik;
```

- Šta je efekat sum funkcije kada u skupu ima null vrednosti?
 - Ignoriše null vrednosti

Funkcija avg

- `AVG(<obeležje>)` – vraća srednju vrednost datog <obeležja>, za sve selektovane torke, uključujući višestruko ponavljanje istih torke
- `AVG(DISTINCT <obeležje>)` – vraća srednju vrednosti datog <obeležja>, za sve različite selektovane torke

Funkcija avg

- Prikazati broj radnika, prosečnu platu i ukupnu godišnju platu svih radnika.
**select count(*) "Broj radnika",
avg(plt) "Prosecna plata",
12*sum(plt) "Godisnja plata"
from radnik;**
- Šta je efekat avg funkcije kada u skupu ima null vrednosti?
 - Ignoriše null vrednosti

Skupovne funkcije nad isključivo null vrednostima

- Prikazati ukupnu premiju svih radnika čiji je matični broj veći od 100.

**SELECT SUM(pre) FROM radnik WHERE
mbr >100;**

- Šta je rezultat sum, avg, max, min funkcija kada su u skupu sve null vrednosti?

Rezultat je null.

- Rezultat count funkcije u tom slučaju je vrednost 0.

Funkcija round

- ROUND(<izraz>, <broj_decimala>) – vraća zaokruženu vrednost datog <izraza> na dati <broj_decimala>

Funkcija round

- Prikazati prosečnu platu svih radnika pomnoženu sa koren iz 2 (1,41) zaokruženo na dve decimale.

```
select round(avg(plt *1.41), 2)  
from radnik;
```

Select naredba u listi tabela

```
SELECT * FROM (SELECT MBR,IME  
FROM radnik);
```


ROWNUM

Prikazati 10 radnika koji imaju najveću platu, sortiranih po plati u opadajućem redosledu

```
SELECT mbr, plt, rownum  
FROM RADNIK WHERE ROWNUM <= 10  
ORDER BY PLT DESC;
```

- **Da li je ovo ispravno?**

ROWNUM

- **Pseudokolona**
 - **ROWNUM** vrednost koju torika dobija nastaje na osnovu redosleda kojim Oracle dobavlja torke
 - Vrednost za **ROWNUM** kreće od 1

```
SELECT mbr, plt, rownum FROM  
(SELECT * FROM Radnik ORDER BY plt  
desc)  
WHERE ROWNUM <= 10;
```

Zadatak avg, round – select naveden u listi kolona

- Prikazati za svakog radnika red koji sadrži njegovu platu, prosečnu platu i apsolutnu (ABS) razliku prosečne plate i njegove plate.

	PLT	PROSECNA_PLATA	RAZLIKA
1	10000	20690.91	10690.91
2	20000	20690.91	690.91
3	15000	20690.91	5690.91
4	8000	20690.91	12690.91
5	40000	20690.91	19309.09

SELECT PLT,

**(SELECT ROUND(AVG(PLT), 2) FROM
radnik) as prosecna_plata,**

**ABS((SELECT ROUND(AVG(PLT), 2) from
RADNIK) – plt) as razlika FROM radnik;**

GROUP BY - Uvod

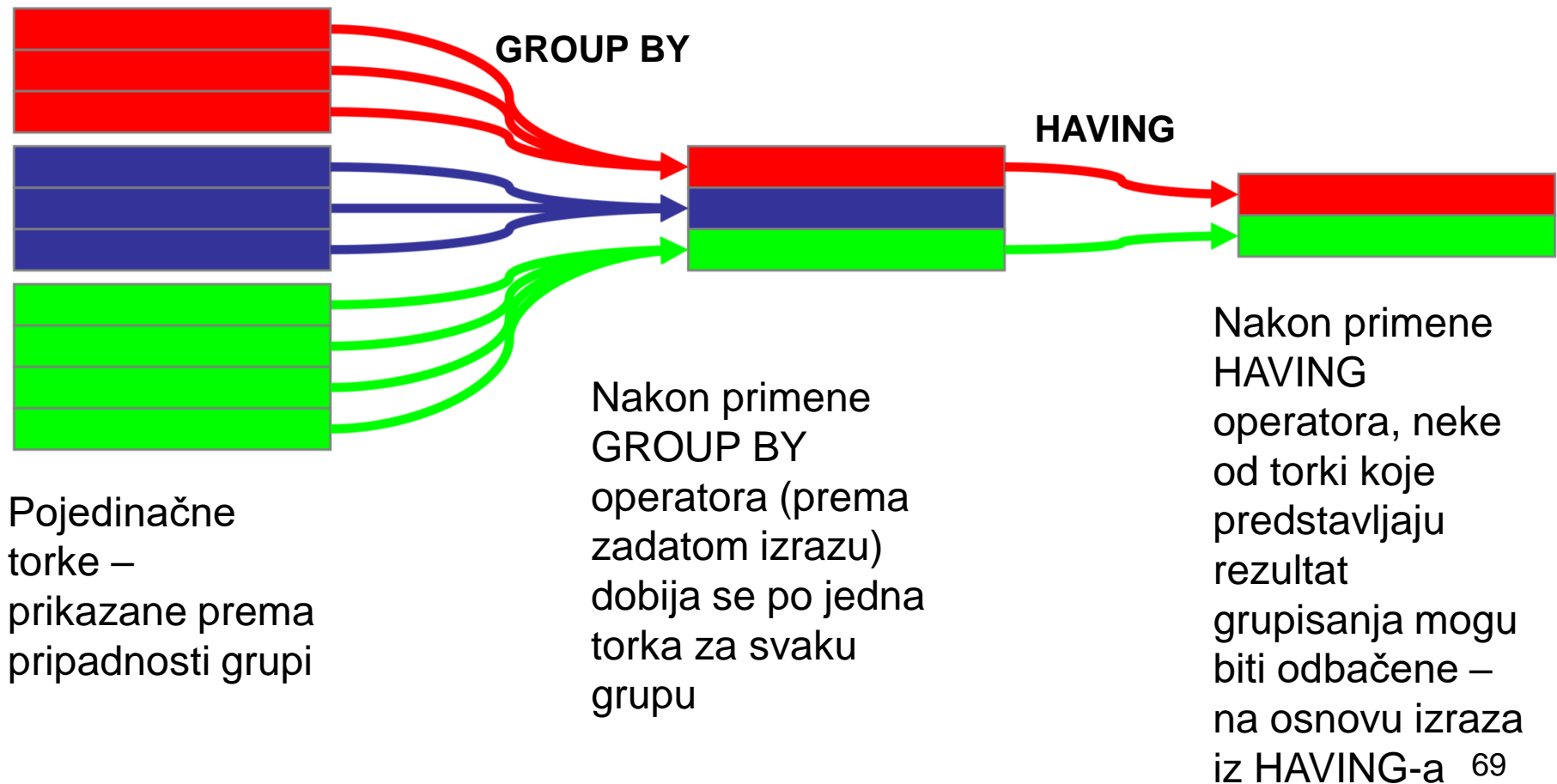
- **SELECT mbr, spr FROM radproj where mbr < 40;**

mbr	spr
10	10
20	20
10	30
30	30
30	40

- **SELECT mbr, count(spr) FROM radproj where mbr < 40 group by mbr;**

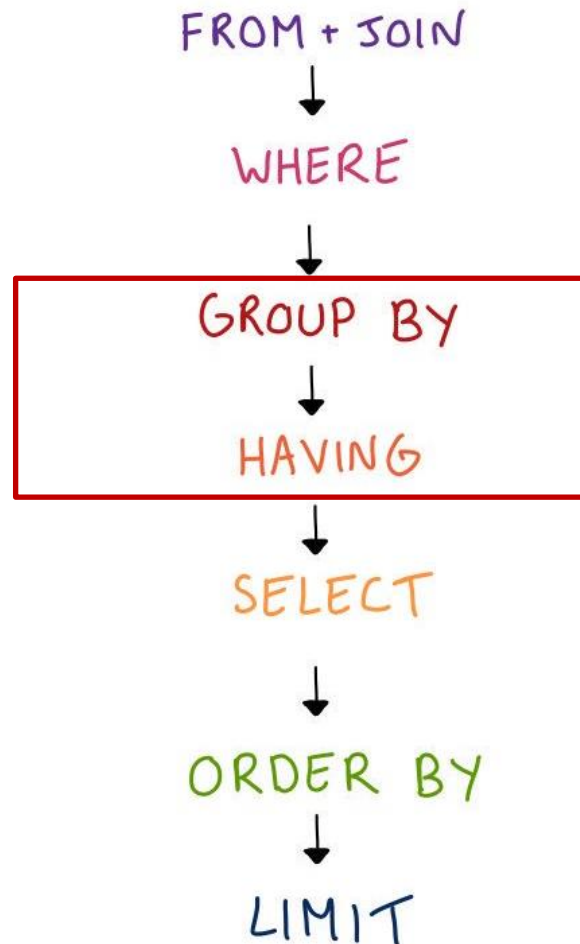
mbr	count(spr)
30	2
20	1
10	2

GROUP BY - Uvod



Redosled izvršavanja klauzula

Redosled izvršavanja SQL klauzula



GROUP BY

- Prikazati koliko radnika radi na svakom projektu i koliko je ukupno angažovanje na tom projektu?

```
select spr, count(mbr), sum(brc)  
from radproj  
group by spr;
```

HAVING

- Izlistati mbr radnika koji rade na više od dva projekta, pored mbr-a, prikazati i broj projekata na kojima radnici rade.

```
select mbr from radproj  
group by mbr  
having count(spr)>2;
```

```
select mbr, count(spr) from radproj  
group by mbr  
having count(spr)>2;
```


GROUP BY - Napomene

- Najčešće se koristi u kombinaciji sa skupovnim funkcijama (min, max, count, avg...)
- Svaka kolona koja se nađe među izrazima SELECT klauzule, osim onih kolona koji su pod skupovnom funkcijom se mora naći i u izrazima GROUP BY klauzule
 - Npr count(spr) se može naći u izrazima u select klauzuli, a spr se ne mora naći naveden u izrazima koji pripadaju group by klauzuli
 - Ovakva upotreba i jeste najčešća
- Grupe se mogu filtrirati korišćenjem having ključne reči
 - Where filtira torke, ne grupe
- Može se koristiti u kombinaciji sa ORDER BY

Nezavisni ugnježdeni upiti

- SELECT unutar WHERE druge SELECT naredbe
- Predikatski izrazi:
 - ANY, ALL, IN i EXISTS
- SQL dozvoljava višestruko ugnježdavanje upita

Nezavisni ugnježdjeni upiti

- Izlistati u rastućem redosledu plate mbr, ime, prz i plt radnika koji imaju platu veću od prosečne.

```
select mbr, ime, prz, plt  
from radnik  
where plt > (select avg(plt) from radnik)  
order by plt asc;
```

Nezavisni ugnježdeni upiti

- Izlistati imena i prezimena radnika koji rade na projektu sa šifrom 30

select ime, prz from radnik

where mbr in

(select mbr from radproj where spr=30);

Nezavisni ugnježdjeni upiti

- Izlistati mbr, ime, prz radnika koji rade na projektu sa šifrom 10, a ne rade na projektu sa šifrom 30.

```
select mbr, ime, prz  
from radnik  
where mbr in  
(select mbr from radproj where spr=10)  
and mbr not in  
(select mbr from radproj where spr=30);
```

- **Zašto ne može u jednom ugnježđenom upitu?**

Nezavisni ugnježdeni upiti

- Izlistati ime, prz i god najstarijeg radnika.

```
select mbr, ime, prz, god  
from radnik  
where god <= all(select god from radnik);
```

```
select mbr, ime, prz, god  
from radnik  
where god = (select min(god) from  
radnik);
```

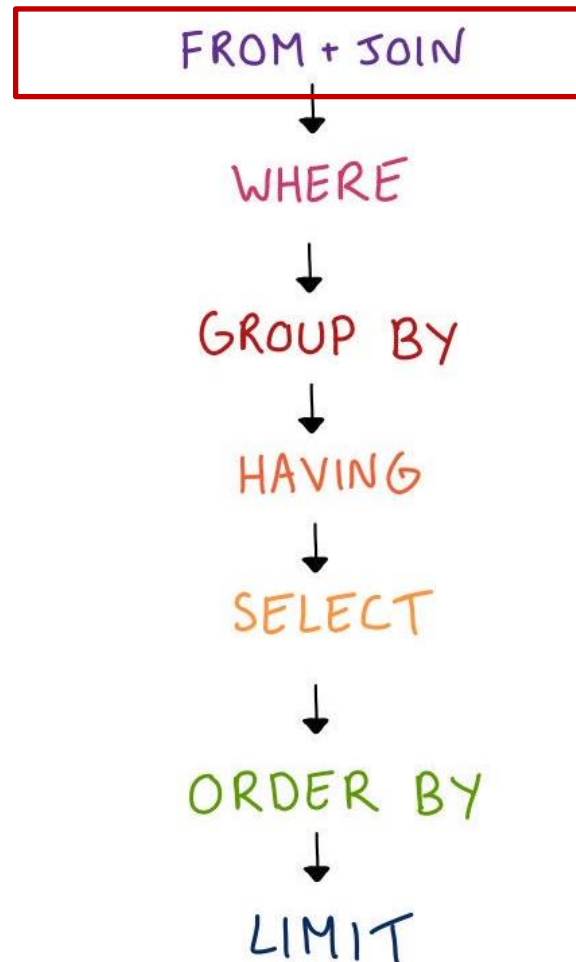
Spajanje tabela

- Prikazati mbr, prz, ime, plt i brc angažovanja svih radnika koji rade na projektu sa šifrom 10.

```
select radnik.mbr, prz, ime, plt, brc  
from radnik, radproj  
where spr=10 and  
radnik.mbr = radproj.mbr;
```

Redosled izvršavanja klauzula

Redosled izvršavanja SQL klauzula



Spajanje tabela

- Prikazati mbr, prz, ime, plt i brc angažovanja svih radnika koji rade na projektu sa šifrom 10.

```
select r.mbr, r.prz, r.ime, r.plt, rp.brc  
from radnik r, radproj rp  
where rp.spr=10 and r.mbr=rp.mbr;
```

Spajanje tabela

- Prikazati mbr, ime, prz i plt radnika koji su rukovodioci projekata.

```
select distinct mbr, ime, prz, plt  
from radnik, projekat  
where ruk=mbr;
```

Spajanje tabela

- Izlistati imena, prezimena svih radnika osim rukovodioca projekta sa šifrom 10.

```
select mbr, ime, prz  
from radnik r, projekat p  
where p.spr=10 and r.mbr!=p.ruk;
```

Primer

- Izlistati imena, prezimena svih radnika osim rukovodioca projekta sa šifrom 10. (sa ugnježdenim upitom)

```
select ime, prz, mbr  
from radnik  
where mbr != (select ruk from projekat  
              where spr=10);
```

Primer

- Izlistati nazive projekata na kojima radi bar jedan radnik koji radi i na projektu sa šifrom 60.

```
select p.nap from projekat p  
where spr in (select spr from radproj  
where mbr in (select mbr from radproj  
where spr=60));
```

Primer

- Prikazati imena i prezimena rukovodilaca projekata i broj projekata kojima rukovode.

```
select prz, ime, count(spr)  
from radnik r, projekat p  
where ruk=mbr  
group by mbr, prz, ime;
```

Primer

- Prikazati za svakog radnika mbr, prz, ime, ukupan broj projekata i ukupno angažovanje na projektima na kojima radi.

```
select r.mbr, r.prz, r.ime, count(*),  
sum(rp.brc)  
from radnik r, radproj rp  
where r.mbr=rp.mbr  
group by r.mbr, r.prz, r.ime;
```

Primer

- Prikazati imena i prezimena rukovodilaca projekata i broj projekata na kojima **rade**.

```
select ime, prz, count(rp.spr) bp
from radnik r, radproj rp
where r.mbr=rp.mbr and
       r.mbr in (select ruk from projekat)
group by r.mbr, prz, ime;
```

```
select ime,prz,count(distinct rp.spr)
from radnik r,projekat p, radproj rp
where rp.mbr=r.mbr and p.ruk=r.mbr
group by r.mbr,ime,prz:
```


Primer

- Izlistati nazive projekata na kojima se ukupno radi više od 15 časova.

```
select nap  
from projekat p, radproj rp  
where p.spr=rp.spr  
group by p.spr, nap  
having sum(brc)>15;
```

Primer

- Izlistati šifre i nazive projekata na kojima radi više od dva radnika.

```
select p.spr, p.nap  
from projekat p, radproj rp  
where rp.spr=p.spr  
group by p.spr, p.nap  
having count(mbr)>2;
```

Primer

- Izlistati nazive i šifre projekata na kojima je prosečno angažovanje veće od prosečnog angažovanja na svim projektima.

```
select p.spr, p.nap  
from projekat p, radproj rp  
where rp.spr=p.spr  
group by p.spr, p.nap  
having avg(brc)>(select avg(brc) from radproj);
```

Primer

- Izlistati nazive i šifre projekata sa najvećim prosečnim angažovanjem.

	SPR	MBR	BRC
1	10	10	10
2	10	50	11
3	10	100	9
4	10	130	2
5	20	20	13
6	20	70	8
7	20	110	8
8	20	120	7
9	30	10	4
10	30	30	3
11	30	50	5
12	30	60	4
13	30	80	9

Primer:

SPR = 10

$$(10+11+9+2)/4 = 8$$

SPR = 20

$$(13+8+8+7)/4 = 9$$

SPR = 30

$$(4+3+4+5+9)/5 = 5$$

Upit bi u ovom slučaju trebalo da vrati
projekat sa šifrom 20

Rešenje

```
select p.spr, p.nap  
from projekat p, radproj rp  
where rp.spr=p.spr  
group by p.spr, p.nap  
having avg(brc)>=all(select avg(brc)  
from radproj group by spr);
```

Upit sa višestrukom upotrebom iste tabele

- Prikazati mbr, ime, prz, plt radnika koji zarađuju više od radnika sa matičnim brojem 40.

```
select r.mbr, r.prz, r.ime, r.plt  
from radnik r, radnik r1  
where r.plt>r1.plt and r1.mbr=40;
```

Upit sa višestrukom upotrebom iste tabele

- Prikazati imena, prezimena i plate radnika koji zarađuju bar 1000 dinara manje od rukovodioca projekta na kom radnik radi.

```
select r1.ime, r1.prz, r1.plt, p.nap from  
radnik r1, radnik r2, projekat p, radproj rp  
where r1.mbr=rp.mbr and rp.spr=p.spr and  
p.ruk=r2.mbr and r1.plt+1000<r2.plt;
```

Povezani upiti

- Prikazati mbr, ime, prz, plt radnika čiji je broj sati angažovanja na nekom projektu veći od prosečnog broja sati angažovanja na tom projektu.

```
select distinct r.mbr, ime, prz, plt, brc  
from radnik r, radproj rp1  
where r.mbr=rp1.mbr and  
rp1.brc>(select avg(brc) from radproj rp2  
where rp2.spr=rp1.spr);
```


EXISTS

EXISTS(<lista_vrednosti>) –
<lista_vrednosti> nije prazan skup
vrednosti

NOT EXISTS(<lista_vrednosti>) –
<lista_vrednosti> je prazan skup vrednosti

EXISTS

- Ko je najstariji radnik? (exist)

```
select ime, prz, god  
from radnik r  
where not exists  
(select mbr from radnik r1  
where r1.god<r.god);
```

EXISTS

- Izlistati mbr, ime, prz radnika koji ne rade na projektu sa šifrom 10. (ne postoji radnik sa projekta 10 koji je jednak traženom radniku)

```
select mbr, ime, prz  
from radnik r  
where not exists  
(select * from radproj rp  
where r.mbr=rp.mbr and rp.spr=10);
```

EXISTS

- Izlistati radnike koji ne rade ni na jednom projektu. (ne postoji projekat na kom rade)

```
select mbr, ime, prz  
from radnik r  
where not exists  
(select * from radproj rp where r.mbr=rp.mbr);
```

```
select mbr, ime, prz  
from radnik r  
where mbr not in  
(select rp.mbr from radproj rp);
```

EXISTS

- Izlistati radnike koji nisu rukovodioci projekata. (ne postoji projekat kojim rukovodi taj radnik)

```
select mbr, ime, prz  
from radnik r  
where not exists  
(select * from projekat where mbr=r.ruk);
```

```
select mbr, ime, prz  
from radnik r  
where mbr not in  
(select ruk from projekat);
```

Povezani upiti - EXISTS

- Ko je najmlađi rukovodilac projekata?

```
select distinct mbr, ime, prz, god  
from radnik r, projekat p  
where r.mbr=p.ruk and not exists  
(select mbr from radnik r1, projekat p1  
where r1.mbr=p1.ruk and r1.god>r.god);
```

Unija (UNION)

- Izlistati mbr, ime, prz radnika koji rade na projektu sa šifrom 20 ili im je plata veća od prosečne. (unija)

```
select mbr, ime, prz from radnik  
where mbr in  
(select mbr from radproj where spr=20)  
union  
select mbr, ime, prz from radnik  
where plt>(select avg(plt) from radnik);
```

Unija (UNION ALL)

- Izlistati mbr, ime, prz radnika koji rade na projektu sa šifrom 20 ili im je plata veća od prosečne. (unija)

```
select mbr, ime, prz from radnik  
where mbr in  
(select mbr from radproj where spr=20)  
union all  
select mbr, ime, prz from radnik  
where plt>(select avg(plt) from radnik);
```


Presek (INTERSECT)

- Izlistati mbr, ime, prz radnika čije prezime počinje na slovo M ili slovo R i mbr, ime, prz radnika čije prezime počinje na slovo M ili slovo P.

```
select mbr, ime, prz from radnik  
where prz like 'M%' or prz like 'R%'
```

INTERSECT

```
select mbr, ime, prz from radnik  
where prz like 'M%' or prz like 'P%';
```

Razlika (MINUS)

- Izlistati mbr, ime, prz radnika čije prezime počinje na slovo M ili slovo R i mbr, ime, prz radnika čije prezime počinje na slovo M ili slovo P.

**select mbr, ime, prz from radnik
where prz like 'M%' or prz like 'R%'
MINUS**

**select mbr, ime, prz from radnik
where prz like 'M%' or prz like 'P%';**

Prirodno spajanje (NATURAL)

- Prikazati ime i prez radnika koji rade na projektu sa šifrom 30.

```
select ime, prez  
from radnik natural join radproj  
where spr=30;
```

Spajanje se vrši na osnovu imena kolona.

Unutrašnje spajanje (INNER)

- Prikazati ime i prz radnika koji rade na projektu sa šifrom 30.

```
select ime, prz  
from radnik r inner join radproj rp  
on r.mbr=rp.mbr  
where spr=30;
```

Spoljno spajanje (OUTER)

- Levo (LEFT)
- Desno (RIGHT)
- Potpuno (FULL)

Spoljno spajanje (LEFT OUTER)

- Prikazati mbr, ime i prz radnika i šifre projekata na kojima rade. Prikazati, takođe, iste podatke i za radnike koji ne rade ni na jednom projektu, pri čemu za šifru projekta treba, u tom slučaju, prikazati nedostajuću vrednost.

```
select r.mbr,ime, prz, spr  
from radnik r left outer join radproj rp  
on r.mbr=rp.mbr;
```

Spoljno spajanje (LEFT OUTER)

- Prikazati mbr, ime i prz svih radnika i nazive projekata kojima rukovode. Ukoliko radnik ne rukovodi ni jednim projektom ispisati: ne rukovodi projektom.

```
select r.mbr,ime, prz, nvl(nap, 'ne rukovodi  
projektom') Projekat  
from radnik r left outer join projekat p  
on r.mbr=p.ruk;
```

Spoljno spajanje (RIGHT OUTER)

- Prikazati nazive svih projekata i mbr radnika koji rade na njima. Ukoliko na projektu ne radi ni jedan radnik ispisati nulu umesto matičnog broja.

```
select nvl(rp.mbr, 0) "Mbr radnika", nap  
from radproj rp right outer join projekat p  
on rp.spr=p.spr;
```

```
select nvl(rp.mbr, 0) "Mbr radnika", nap  
from radproj rp, projekat p  
where rp.spr(+) = p.spr;
```


Spoljno spajanje (FULL OUTER)

```
select nvl(rp.mbr, 0) "Mbr radnika", nap  
from radproj rp full outer join projekat p  
on rp.spr=p.spr;
```

Primer

- Prikazati matične brojeve, imena i prezimena radnika, zajedno sa šiframa projekata na kojima rade. Prikazati, takođe, iste podatke i za radnike koji ne rade ni na jednom projektu, pri čemu za šifru projekta treba, u tom slučaju, prikazati nedostajuću vrednost.

Rešenje

```
SELECT r.Mbr, r.Prz, r.Ime, rp.Spr  
FROM Radnik r, Radproj rp  
WHERE r.Mbr = rp.Mbr (+);
```

```
SELECT r.Mbr, r.Prz, r.Ime, rp.Spr  
FROM Radnik r LEFT OUTER JOIN  
Radproj rp ON r.Mbr = rp.Mbr;
```

Primer

- Prikazati za sve radnike i projekte na kojima rade Mbr, Prz, Ime, Spr i Nap. Za radnike koje ne rade ni na jednom projektu, treba prikazati Mbr, Prz i Ime, dok za vrednosti obeležja Spr i Nap treba zadati, redom, konstante 0 i "Ne postoji". Urediti izlazni rezultat saglasno rastućim vrednostima obeležja Mbr.

Rešenje

```
SELECT r.Mbr, r.Prz, r.Ime, NVL(p.Spr, 0) AS Spr,  
NVL(p.Nap, 'Ne postoji') AS Nap  
FROM Radnik r, Radproj rp, Projekat p  
WHERE r.Mbr = rp.Mbr (+) AND rp.Spr = p.Spr (+)  
ORDER BY Mbr;
```

```
SELECT r.Mbr, r.Prz, r.Ime, NVL(p.Spr, 0) AS Spr,  
NVL(p.Nap, 'Ne postoji') AS Nap  
FROM Radnik r LEFT OUTER JOIN Radproj rp ON  
r.Mbr = rp.Mbr LEFT OUTER JOIN Projekat p ON  
rp.Spr = p.Spr  
ORDER BY Mbr;
```

Primer

- Prikazati imena i prezimena svih radnika i prezimena njihovih šefova ako ih imaju. Ako nema šefa ispisati: nema sefa.

```
select r1.ime, r1.prz "Radnik",  
nvl(r2.prz, 'Nema sefa') Sef  
from radnik r1 left outer join radnik r2  
on r1.sef=r2.mbr  
order by r1.prz;
```

Dekartov proizvod spajanje (Cross Join)

- Koristi se ako želimo da napravimo Dekartov proizvod između dve tabele

SELECT * FROM radnik, projekat

- Ekvivalentno je sa

SELECT *

FROM radnik CROSS JOIN projekat;

- Može se dodati uslov na cross join, onda se ponaša kao inner join
- Često se zaborave uslovi spoja prilikom spajanja tabela, pa rezultat bude Dekartov proizvod torki iz spajajućih tabela

Zadatak za vežbu

- Za svaku satnicu angažovanja (brc), prikazati koliko radnika radi na nekom projektu sa tom satnicom. Rezultate urediti u opadajućem redosledu satnice.

```
SELECT brc, COUNT(mbr)  
FROM radproj GROUP BY brc  
ORDER BY brc DESC;
```


Zadatak za vežbu

- Za svakog radnika prikazati matični broj, ime, prezime, kao i broj projekata kojima rukovodi, pri čemu je potrebno prikazati isključivo one radnike koji su rukovodioci na manjem broju projekata od prosečnog broja projekata na kojima rade radnici čije se prezime ne završava na "ić".

```
SELECT mbr, ime, COUNT(spr) br_pr_rukovodi  
FROM radnik r LEFT OUTER JOIN projekat p on r.mbr=p.ruk  
GROUP BY mbr, ime HAVING COUNT(spr) < (SELECT  
AVG(COUNT(spr)) FROM radproj rp, radnik r  
WHERE rp.mbr = r.mbr  
AND prz NOT LIKE '%ić'  
GROUP BY r.mbr);
```

Selekcionni izraz (CASE)

- Prosti (*Simple*) CASE:

```
CASE expr
    WHEN expr1 THEN return_expr1
    [ WHEN expr2 THEN return_expr2
      WHEN exprn THEN return_exprn ]
    [ ELSE else_expr ]
END;
```

- Pretražujući (*Searched*) CASE:

```
CASE
    WHEN comparison_expr1 THEN return_expr1
    [ WHEN comparison_expr2 THEN return_expr2
      WHEN comparison_exprn THEN return_exprn ]
    [ ELSE else_expr ]
END;
```

Primer

- Za svakog radnika prikazati mbr, ime, prz, kao kategoriju kojoj pripada na osnovu visine plate. Kategorije po visini plate su sledeće:
 - Plata manja od 10000 – kategorija: '**mala primanja**',
 - plata između 10000 i 20000 – kategorija: '**srednje visoka primanja**',
 - plata između 20000 i 40000 – kategorija: '**visoka primanja**',
 - plata veća od 40000 – kategorija: '**izuzetno visoka primanja**'.
- Takođe, radnike urediti prema kategoriji kojoj pripadaju, u redosledu od najniže ka najvišoj kategoriji po visini plate.

Primer

```
select mbr, ime, plt,  
case  
  when plt < 10000 then 'mala primanja'  
  when plt >=10000 and plt < 20000 then 'srednja primanja'  
  when plt >=20000 and plt < 40000 then 'visoka primanja'  
  else 'izuzetno visoka primanja'  
end as visina_primanja  
from radnik  
order by  
case visina_primanja  
  when 'izuzetno visoka primanja' then 1  
  when 'visoka primanja' then 2  
  when 'srednja primanja' then 3  
  else 4  
end desc, plt asc;
```

Selekcionni izraz (CASE)

- Napomene:
 - Može se iskoristiti gde god je dozvoljeno korišćenje izraza
 - Najčešće – u okviru liste kolona ili u okviru *order by* klauzule
 - Ukoliko se ne iskoristi *else*, podrazumevana vrednost biće *null*
 - Kod prostog CASE izraza, poređenje sa *null* vrednošću **nije moguće** – podrazumevano se koristi poređenje operatorom '=', pa je takav izraz uvek netačan

Primer

- Za svakog radnika ispisati mbr, ime, prz, platu i mbr šefa. Pri ispisu treba obezbediti da su radnici uređeni saglasno visini plate, od najviše ka najnižoj, pri čemu bi direktor firme trebalo da se ispiše prvi.

```
select mbr, ime, plt, sef  
from radnik  
order by  
case  
    when sef is null then 1  
    else 2  
end, plt desc;
```

Ažuriranje baze podataka

- **INSERT**
- **DELETE**
- **UPDATE**

Ažuriranje baze podataka

- INSERT – dodavanje nove torke

```
INSERT INTO <naziv_tabele>  
[(<lista_obeležja>)] VALUES  
(<lista_konstanti>) | SELECT ...
```


Ažuriranje baze podataka

- INSERT – dodavanje nove torke

insert into Radnik (mbr, ime, prz, plt, sef, god) values (201, 'Ana', 'Savic', 30000, null, '18-aug-71');

insert into Projekat (spr, nap, ruk) values (90, 'P1', 201);

insert into RadProj (mbr, spr, brc) values (201, 90, 5);

Ažuriranje baze podataka

- Probati dodavanje nove torke sa postojećim ključem
- Probati dodavanje novog radnika sa vrednosti *null* postavljenom za IME
- Probati dodavanje nove torke sa nedozvoljenom vrednosti PLT (<500)
- Probati dodavanje novog projekta sa postojećim nazivom
- Probati dodavanje novog projekta sa nepostojećim MBR-om rukovodioca

Ažuriranje baze podataka

- DELETE – brisanje postojećih torki

**DELETE FROM <naziv_tabele>
[WHERE (<uslov_selekcije>)]**

Ažuriranje baze podataka

- DELETE – brisanje postojećih torki

delete radnik;

delete radnik where mbr=701;

Ažuriranje baze podataka

- Probati brisanje torke koja je referencirana od strane neke druge torke.

Ažuriranje baze podataka

- UPDATE – modifikacija postojećih torki

UPDATE <naziv_tabele>

SET <obeležje>= <aritm_izraz>

{,<obeležje>= <aritm_izraz>}

[WHERE (<uslov_selekcije>)]

Ažuriranje baze podataka

- UPDATE – modifikacija postojećih torki

```
update radnik  
set plt = plt*1.2;
```

```
update radnik  
set plt = plt*1.2  
where mbr = 201;
```

Ažuriranje baze podataka

- Probati ažuriranje torke sa nedozvoljenom vrednosti PLT (<500)

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

Naziv tabele i kolone

- mora početi slovom,
- mora biti između 1 i 30 znakova dužine,
- mora sadržati samo velika i mala slova, cifre, _, \$ i #,
- ne sme se poklapati sa nazivom nekog drugog objekta koji je kreirao isti korisnik,
- ne sme biti rezervisana reč Oracle servera.
- Nazivi nisu case sensitive.

SQL tipovi podataka

Tip podatka	Opis
– 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 – za kompatibilnost sa starijim verzijama Oracle-a
– CLOB	niz karaktera promenljive dužine do 2 GB
– BLOB	binarni podaci do 4 GB
– 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.

Create table as select

- Kreirati tabelu radnik2 koja ima iste kolone kao tabela radnik, pri čemu radnik2 sadrži samo podatke o radnicima koji imaju platu manju od 10000

```
CREATE TABLE radnik2 AS (SELECT * FROM  
radnik WHERE plt < 10000);
```

Tabela radnik2 neće imati indekse i sva ograničenja koja ima tabela radnik

Insert into select

- Dopuniti tabelu radnik2 podacima koji joj nedostaju iz tabele radnik,

INSERT INTO radnik2 (SELECT * FROM radnik WHERE plt>=10000);

- Proveriti da li je broj torki u tabeli radnik jednak broju torki u tabeli radnik2
- Izbrisati sadržaj i definiciju tabele radnik2

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 (rukovodioca radnika) prikazati njegov matični broj, prezime, ime, ukupan broj radnika kojima šefuje 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, pogledе, 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 karakter funkcije – primer 2

- Prikazati radnike čije prezime na početku sadrži prva 3 slova imena, na primer:
Petar **P**etric

```
SELECT * from radnik  
WHERE prz LIKE  
SUBSTR(IME,0,3) || '%';
```

Zadatak za vežbu

- Prikazati imena i prezimena radnika tako da se sva imena koja imaju poslednje slovo 'a', prikazuju bez poslednjeg slova.

```
SELECT TRIM(TRAILING 'a' FROM ime)  
FROM radnik;
```


Zadatak za vežbu

- Svim radnicima promeniti ime tako da poslednje slovo bude uvećano.
- Primer: AnA -> AnA, Marko -> MarkO

```
UPDATE radnik SET ime=  
SUBSTR(ime,1,LENGTH(ime)-1) ||  
UPPER(SUBSTR(ime,LENGTH(ime),1));
```

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

Zadatak za vežbu

- Za svakog radnika prikazati ime, prz, i projekte na kojima radi. Ako ne radi ni na jednom projektu, napisati 'Ne radi na projektu'. Imena radnika prikazati velikim slovima, a prezimena malim.

```
SELECT UPPER(ime), LOWER(prz),  
NVL(TO_CHAR(spr), 'Ne radi na projektu') broj_proj  
FROM radnik LEFT OUTER JOIN radproj  
on radnik.mbr = radproj.mbr;
```

Zadatak za vežbu

- Za svakog radnika prikazati datum rođenja u formatu yyyy/mm/dd

SELECT TO_CHAR(god,'yyyy/mm/dd') FROM radnik;

KLAUZULA WITH

Klauzula WITH

- dodela naziva bloku podupita
- blok može biti referenciran više puta unutar upita
- optimizacija upita
 - kao umetnuti pogled
 - kao privremena tabela

WITH naziv_upita

WITH – Primer

- Prikazati za svakog radnika angažovanog na projektu mbr, prz, ime, spr i broj drugih radnika koji su angažovani na istom projektu

```
select r.mbr, r.ime, r.prz, rp1.spr, count(rp2.mbr)-1 ostali  
from radnik r, radproj rp1, radproj rp2  
where r.mbr=rp1.mbr and rp1.spr=rp2.spr  
group by r.mbr, r.ime, r.prz, rp1.spr;
```

```
with projinfo as (  
  select rp.spr, count(rp.mbr) as rad_broj  
  from radproj rp group by rp.spr)  
select r.mbr, r.ime, r.prz, rp.spr, pi.rad_broj-1 ostali  
from radnik r, radproj rp, projinfo pi  
where r.mbr=rp.mbr and rp.spr=pi.spr;
```

WITH – Zadatak

- Prikazati za svakog radnika angažovanog na projektu mbr, prz, ime, spr i udeo u ukupnom broju časova rada na tom projektu (zaokruženo na dve decimale)

```
with projinfo as (  
    select rp.spr, sum(rp.brc) as cas_suma  
    from radproj rp group by rp.spr)  
select r.mbr, r.ime, r.prz, rp.spr,  
round(rp.brc/pi.cas_suma, 2) udeo  
from radnik r, radproj rp, projinfo pi  
where r.mbr=rp.mbr and rp.spr=pi.spr;
```


WITH – Zadatak

- Prikazati mbr, ime, prz, plt radnika čiji je broj sati angažovanja na nekom projektu veći od prosečnog broja sati angažovanja na tom projektu

```
with projinfo as (  
    select spr, avg(brc) prosek  
    from radproj group by spr)  
select distinct r.mbr, r.ime, r.prz, r.plt  
from radnik r, radproj rp, projinfo pi  
where r.mbr=rp.mbr and rp.spr=pi.spr  
group by r.mbr, r.ime, r.prz, r.plt, pi.spr  
having avg(rp.brc)>(select prosek from projinfo pi2  
where pi2.spr=pi.spr);
```

WITH – Zadatak

- Prikazati mbr, ime, prz, plt radnika čiji je broj sati angažovanja na nekom projektu veći od prosečnog angažovanja na svim projektima

```
with projinfo as (  
  select spr, avg(brc) pros  
  from radproj group by spr)  
select distinct r.mbr, r.ime, r.prz, r.plt  
from radnik r, radproj rp, projinfo pi  
where r.mbr=rp.mbr and rp.spr=pi.spr  
group by r.mbr, r.ime, r.prz, r.plt, pi.spr  
having avg(rp.brc)>(select avg(pros) from projinfo);
```

WITH – Primer

- Prikazati mbr, ime i prz rukovodilaca projekata kao i ukupan broj radnika kojima rukovode na projektima

```
with rukovodilac as (  
    select mbr, ime, prz, plt, spr  
    from radnik, projekat where mbr=ruk),  
projinfo as (  
    select spr, count(mbr) ljudi  
    from radproj group by spr)  
select ru.mbr, ru.ime, ru.prz, sum(pi.ljudi) ljudi  
from rukovodilac ru, projinfo pi  
where ru.spr=pi.spr  
group by ru.mbr, ru.ime, ru.prz;
```

WITH – Zadatak

- Koliko je ukupno angažovanje svih šefova na projektima?

```
with 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),  
angaz_sefova (mbr, prz, ime, brrad, brsat) as (  
    select distinct 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)  
select sum(brsat) as ukangsef  
from angaz_sefova;
```

WITH – Rekurzija

- blok podupita pomoću WITH
- blok sadrži dva upita vezana preko UNION ALL
 - prvi upit određuje početni skup podataka
 - drugi upit obezbeđuje rekurzivno proširenje skupa putem unije sa tekućim skupom
- postupak se zaustavlja kada ne dođe do promene skupa prilikom proširenja

```
WITH naziv_upita(lista_obeležja) as  
(  
    upit1  
    UNION ALL  
    upit2  
)
```

WITH – Rekurzija – Primer

- Prikazati za svakog radnika sve direktno i indirektno nadređene radnike

```
with hijerarhija(mbr,sef) as  
( select mbr, sef  
  from radnik  
  union all  
  select r.mbr, h.sef  
  from radnik r, hijerarhija h  
  where r.sef = h.mbr and h.sef is not null)  
select * from hijerarhija order by mbr, sef;
```

WITH – Rekurzija – Zadatak

- Prikazati za svakog radnika sve direktno i indirektno podređene radnike

```
with hijerarhija(mbr,pod) as  
( select sef, mbr  
  from radnik  
  union all  
  select h.mbr, r.mbr  
  from hijerarhija h, radnik r  
  where h.pod = r.sef and h.mbr is not null)  
select * from hijerarhija order by mbr, pod;
```

WITH – Rekurzija – Zadatak

- Prikazati za svakog radnika sve direktno i indirektno podređene radnike, ako nema podređenih prikazati null umesto oznake podređenog

```
with hijerarhija(mbr,pod) as  
( select sef, mbr  
  from radnik  
  union all  
  select h.mbr, r.mbr  
  from hijerarhija h, radnik r  
  where h.pod = r.sef and h.mbr is not null)  
select r.mbr, h.pod from hijerarhija h, radnik r  
where r.mbr = h.mbr(+) order by mbr, pod;
```


WITH – Rekurzija – Zadatak

- Prikazati za svakog radnika oznaku šefa

```
with hijerarhija(mbr,imeprz,sef) as
( select mbr, ime||' '||prz, sef
  from radnik
 where sef is null
 union all
  select r.mbr, r.ime||' '||r.prz, h.mbr
  from radnik r, hijerarhija h
  where r.sef = h.mbr )
select * from hijerarhija;
```

WITH – Rekurzija – SEARCH

- klauzula SEARCH
- definiše poredak redova
 - BREADTH FIRST, DEPTH FIRST
- BY – poredak redova na istom nivou
- SET – vrednost pseudo-obeležja po redosledu redova
 - pseudo-obeležje automatski postaje deo rezultata

SEARCH BREADTH | DEPTH FIRST
BY *lista_obeležja*
SET *pseudo-obeležje*

WITH – Rekurzija – Primer

- Prikazati hijerarhiju rukovođenja po nivoima

```
with hijerarhija(mbr,imeprz,sef,nivo) as
( select mbr, ime||' '||prz, sef, 1 as nivo
  from radnik
 where sef is null
 union all
  select r.mbr, r.ime||' '||r.prz, h.mbr, nivo+1
  from radnik r, hijerarhija h
  where r.sef = h.mbr)
search depth first by imeprz set poredak
select nivo, rpad(' ',3*nivo)||imeprz as imeprz,
mbr, sef
from hijerarhija
order by poredak;
```

WITH – Rekurzija – Primer

- Prikazati hijerarhiju rukovođenja po nivoima

```
with hijerarhija(mbr,imeprz,sef,nivo) as
( select mbr, ime||' '||prz, sef, 1 as nivo
  from radnik
 where sef is null
 union all
  select r.mbr, r.ime||' '||r.prz, h.mbr, nivo+1
  from radnik r, hijerarhija h
  where r.sef = h.mbr)
search breadth first by imeprz set poredak
select nivo, rpad(' ',3*nivo)||imeprz as imeprz,
mbr, sef
from hijerarhija
order by poredak;
```

WITH – Rekurzija – Primer

- Prikazati za svakog radnika lanac rukovođenja

```
with hijerarhija(imeprz,mbr,sef,lanac,glavni) as
( select ime||' '||prz, mbr, sef,
  '/'||ime||' '||prz as lanac,
  ime||' '||prz as glavni
  from radnik
  where sef is null
  union all
  select r.ime||' '||r.prz, r.mbr, r.sef,
  h.lanac||'/'||r.ime||' '||r.prz as lanac, h.glavni
  from radnik r, hijerarhija h
  where r.sef = h.mbr )
select imeprz,mbr,sef,lanac,glavni from hijerarhija;
```

WITH – Rekurzija – CYCLE

- klauzula CYCLE
- označava cikluse u rekurziji
 - prema proveru zadate liste obeležja
- oznaka prisustva ili odsustva ciklusa
 - pseudo-obeležje automatski postaje deo rezultata
 - jedan karakter

CYCLE *lista_obeležja*

SET *pseudo_obeležje*

TO *oznaka_ciklusa*

DEFAULT *oznaka_odsustva_ciklusa*

WITH – Rekurzija – Primer

- Prikazati za svakog radnika lanac rukovođenja, uz proveru postojanja ciklusa

```
with hijerarhija(imeprz,mbr,sef,lanac,glavni) as
( select ime||' '||prz, mbr, sef,
  '/'||ime||' '||prz as lanac, ime||' '||prz as glavni
  from radnik where sef is null
  union all
  select r.ime||' '||r.prz, r.mbr, r.sef,
  h.lanac||'/'||r.ime||' '||r.prz as lanac, h.glavni
  from radnik r, hijerarhija h where r.sef = h.mbr )
search breadth first by imeprz set poredak
cycle mbr set ciklus to 'x' default 'o'
select imeprz,mbr,sef,lanac,glavni, ciklus from
hijerarhija;
```

WITH – Rekurzija – Zadatak

- Prikazati sve podređene za radnika 70

```
with hijerarhija(imeprz,mbr,sef,lanac,glavni) as
( select ime||' '||prz, mbr, sef,
  '/'||ime||' '||prz as lanac,
  ime||' '||prz as glavni
  from radnik where mbr = 70
  union all
  select r.ime||' '||r.prz, r.mbr, r.sef,
  h.lanac||'/'||r.ime||' '||r.prz as lanac, h.glavni
  from radnik r, hijerarhija h where r.sef = h.mbr )
search breadth first by imeprz set poredak
cycle mbr set ciklus to 'x' default 'o'
select imeprz,mbr,sef,lanac,glavni, ciklus from
hijerarhija;
```


WITH – Rekurzija – Zadatak

- Promeniti šefa radnika 70 da bude radnik 140

```
update radnik  
set sef = 140  
where mbr = 70;
```

WITH – Rekurzija – Zadatak

- Ponovo prikazati sve podređene za radnika 70

```
with hijerarhija(imeprz,mbr,sef,lanac,glavni) as
( select ime||' '||prz, mbr, sef,
  '/'||ime||' '||prz as lanac, ime||' '||prz as glavni
  from radnik where mbr = 70
  union all
  select r.ime||' '||r.prz, r.mbr, r.sef,
  h.lanac||'/'||r.ime||' '||r.prz as lanac, h.glavni
  from radnik r, hijerarhija h
  where r.sef = h.mbr )
search breadth first by imeprz set poredak
cycle mbr set ciklus to 'x' default 'o'
select imeprz, mbr, sef, lanac, glavni, ciklus
from hijerarhija;
```

WITH – Rekurzija – Zadatak

- Poništiti promenu šefa radnika 70
rollback;

SPARSE MATRICE

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;**

KRAJ PREZENTACIJE O SQL-U