

# SQL

2. deo

# Grupisanje podataka

- Klauzula

`GROUP BY <lista_obeležja>`

omogućava particioniranje skupa seletovanih torki saglasno istim vrednostima skupa obeležja datog pomoću `<lista_obeležja>`

- Posle klauzule `GROUP BY` može biti navedena klauzula

`HAVING <uslov_selekcije_grupe>`

koja obezbeđuje zadavanje kriterijuma za selekciju grupa torki

- `<uslov_selekcije_grupe>` je isti logički izraz kao kod klauzule `WHERE`

# Grupisanje podataka

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

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

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

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

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

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

# Grupisanje podataka

- Izlistati mbr radnika koji rade na više od dva projekta.

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

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

# Grupisanje podataka

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

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

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

```
select p.spr, p.naziv
from projekat as p, radproj as rp
where rp.spr=p.spr
group by p.spr, p.naziv
having count(mbr)>1;
```

# Ugnježdjeni upiti

- Upiti kod kojih je jedna `SELECT` naredba ugrađena u `WHERE` klauzulu druge `SELECT` naredbe
- Realizuju se putem predikatskih izraza `ANY`, `ALL`, `IN`, `EXISTS` i njihovih negacija
- Odgovor na upit se generiše tako što se prvo selektuju podaci ugnježdenog upita, a potom se za svaku torku glavnog upita primenjuje kriterijum naveden u klauzuli `WHERE`

# Nezavisni ugnježdeni upit

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

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

# Zavisni ugnježdeni upit

- 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, prezime,  
plata, brc  
from radnik as r, radproj as rp1  
where r.mbr=rp1.mbr and  
      rp1.brc > (select avg(brc)  
                from radproj as rp2  
                where rp2.spr=rp1.spr);
```



# Ugnježdeni upiti

- Izlistati imena i prezimena radnika koji rade na projektu sa šifrom 70 (pomoću ugnježdenog upita).

```
select ime, prz from radnik r
where mbr in (select mbr from radproj where spr=70);
```

- Izlistati mbr, ime, prz radnika koji rade na projektu sa šifrom 20, a ne rade na projektu sa šifrom 10.

```
select mbr, ime, prz
from radnik
where mbr in (select mbr from radproj where spr=20)
and mbr not in (select mbr from radproj where spr=10);
```

- Izlistati ime, prz i god najstarijeg radnika.

```
select mbr, ime, prezime, datumr
from radnik
where datumr <= all(select datumr from radnik);
```

III

```
select mbr, ime, prezime, datumr
from radnik
where datumr <= (select min(datumr) from radnik);
```

# Ugnježdeni upiti

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

```
select r.ime, r.prz, r.mbr from radnik r
where r.mbr not in (select ruk
                    from projekat p
                    where p.spr=10);
```

- 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));
```

# Grupisanje podataka (korišćenje ugnježdenih upita)

- 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);
```

- Izlistati nazive i šifre projekata na kojima je prosečno angažovanje veće od prosečnog angažovanja na svakom projektu (najveće prosečno angažovanje).

```
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);
```

# Grupisanje podataka (korišćenje ugnježdenih upita)

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

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

# Unija

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

```
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);
```

# Exists, not exists

- Ko je najstariji radnik?

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

- Ko je najstariji 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);
```

- 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, not 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);
```

ILI

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

- 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.mbr);
```

ILI

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

# 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>;
```

# Update

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

# Modifikacija

- 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
```

# Modifikacija

- Povećati platu za 10% radnicima koji rade na više od 5 projekata.
- Povećati platu za 10% rukovodiocima bar dva projekta.