

Tema: Agregatne funkcije

Ako je potrebno izvršiti neke operacije nad svim entitetima koji ulaze u rezultat upita, to se može uraditi upotrebom AGREGATNIH FUNKCIJA.

Osnovne agregatne funkcije su:

- brojanje (count)
- sumiranje (sum)
- izračunavanje
 - najveće (max)
 - najmanje (min)
 - srednje vrednosti (avg)

Sve se navode u SELECT klauzuli, tako što im se kao argument u zagradama navodi atribut ili lista atributa nad kojima želimo da operišu.

Ako se u SELECT klauzuli pojavi agregatna funkcija onda se u njoj ne sme pojaviti ništa osim još neke druge agregatne funkcije.

Agregatne funkcije se ne smeju pojaviti u WHERE klauzuli direktno (već jedino u nekom podupitu).

Funkcija COUNT broji koliko entiteta ulazi u rezultat upita. Ako želimo da izbrojimo samo različite entitete, neophodno je koristiti ključnu reč DISTINCT. Ako nije neophodno vršiti projekciju, umesto liste atributa navodi se *.

Funkcija SUM sumira vrednosti navedenog atributa za sve entitete koji zadovoljavaju uslove upita. Funkcije MAX, MIN i AVG računaju najveću, najmanju i srednju vrednost navedenog atributa za entitete koji zadovoljavaju uslove upita. STDDEV računa standardnu devijaciju, a VARIANCE varijansu po zadatom atributu. Funkcije STDDEV, VARIANCE, SUM i AVG dopuštaju kao argumente samo attribute koji su numeričkog tipa.

GRUPISANJE REZULTATA

Entiteti koji zadovoljavaju uslove upita u kome se koriste agregatne funkcije mogu se grupisati, tako da se agregatne funkcije odnose samo na pojedinačne grupe entiteta. Grupisanje se vrši po jednom ili više entiteta. Grupisanje se vrši po jednom ili više atributa, pri čemu jednu grupu čine svi oni entiteti koji zadovoljavaju uslove upita, a imaju jednake vrednosti atributa po kojima se vrši grupisanje. Atributi po kojima se entiteti grupišu moraju pripadati listi traženih atributa. Grupisanje se vrši GROUP klauzulom koja sledi iza WHERE klauzule.

Kao što se WHERE klauzulom upita biraju entiteti koji su od interesa, tako se HAVING klauzulom se biraju grupe koje su od interesa. Uslov koji se nalazi u HAVING klauzuli može sadržati i agregatne funkcije (u WHERE ne može!). Ova klauzula sledi iza GROUP klauzule (i to samo kada GROUP klauzula postoji).

REŠENJA ZADATAKA

Primer 1: Izdvojiti ukupan broj studenata.

```
select count(*) broj_studenata
from dosije;
```

Primer 2: Izdvojiti ukupan broj studenata koji bar iz jednog predmeta imaju ocenu 10.

```
select count(distinct indeks) broj_studenata
from ispit
where ocena=10;
```

Primer 3: Izdvojiti ukupan broj položenih predmeta i položenih bodova za studenta sa indeksom 25/2010.

```
select count(*), sum(bodovi) suma_bodova
from ispit i join predmet p on i.id_predmeta=p.id_predmeta
where ocena>5 and indeks=20100025;
```

U prethodnim primerima uočite razmiku između: count(*), count(ime_kolone), count(distinct ime_kolone)...

Primer 4: Izlistati ocene dobijene na ispitima i ako je ocena jednaka 5 ispisati NULL

```
SELECT NULLIF(ocena, 5)
FROM ispit;
```

Primer 5: Koliko ima različitih ocena dobijenih na ispitu a da ocena nije 5.

```
SELECT COUNT(DISTINCT NULLIF(ocena, 5))
FROM ispit;
```

Primer 6: Izdvojiti šifre, nazive i bodove predmeta čiji je broj bodova veći od prosečnog broja bodova svih predmeta.

```
select sifra, naziv, bodovi
from predmet
where bodovi > (select avg(bodovi + 0.0)
               from predmet);
```

Primer 7: Za svaki predmet izračunati koliko studenata ga je položilo.

```
select id_predmeta, (select count(*) from ispit i where i.id_predmeta=p.id_predmeta
and ocena>5) as polozilo
from predmet p;
```

Ili:

```
select p.id_predmeta, count(indeks) as polozilo
from predmet p left outer join ispit i on p.id_predmeta=i.id_predmeta and ocena>5
group by p.id_predmeta;
```

Primer 8: Za svakog studenta rođenog 1992. godine, koji ima bar jedan položen ispit, izdvojiti broj indeksa, prosečnu ocenu, najmanju i najveću ocenu.

```
select d.indeks, avg(ocena+0.0) as prosek, min(ocena) as najmanja_ocena,
max(ocena) as najveca_ocena
from dosije d join ispit i on d.indeks=i.indeks
where god_rodjenja=1992 and ocena>5
group by d.indeks;
```

Primer 9: Za svaku godinu ispitnog roka i predmet pronaći najveću ocenu. Izdvojiti godinu roka, naziv predmeta i najveću ocenu.

```
select godina_roka, naziv, max(ocena) as najveca_ocena
from ispit i join predmet p on i.id_predmeta=p.id_predmeta
group by godina_roka, naziv;
```

Primer 10 : Izdvojiti predmete koje je polagalo više od 5 različitih studenata.

```
select id_predmeta, count(distinct indeks) broj_studenata
from ispit
group by id_predmeta
having count(distinct indeks)>5;
```

Međutim, ne mora se izdvajati I broj_studenata, pa ga ne moramo na voditi u select liniji.

```
select id_predmeta
from ispit
group by id_predmeta
having count(distinct indeks)>5;
```

Rešenja zadataka za vežbu:

Primer 11: Za svakog studenta izdvojiti broj indeksa i mesec u kome je položio više od dva ispita (nije važno koje godine). Izdvojiti indeks studenta, ime meseca i broj položenih predmeta. Rezultat urediti prema broju indeksa i mesecu polaganja.

```
--select indeks, monthname(datum_ispita) as mesec, count(*) as broj_predmeta
--from ispit
--where ocena>5
--group by indeks, monthname(datum_ispita)
--having count(*) > 2
--order by indeks, monthname(datum_ispita);
```

obratiti pažnju da ne može:

```
--select indeks, monthname(datum_ispita) as mesec, count(*) as broj_predmeta
--from ispit
--where ocena>5
--group by indeks, monthname(datum_ispita)
--having broj_predmeta > 2
--order by indeks, monthname(datum_ispita);
```

Ovako nešto je već moguće:

```
--select indeks, monthname(datum_ispita) as mesec, count(*) as broj_predmeta
--from ispit
--where ocena>5
--group by indeks, monthname(datum_ispita)
--having count(*) > 2
--order by indeks, mesec;
```

Primer 12: Za svaki rok koji održan 2011. godine i u kome nema neuspešnih polaganja ispita, izdvojiti oznaku roka, broj položenih ispita u tom roku i broj studenata koji su položili ispite u tom roku.

```
select oznaka_roka, count(*) as broj_ispita, count(distinct indeks) broj_studenata
from ispit
where godina_roka=2011
group by oznaka_roka
having min(ocena)>5;
```

--ili

```
select oznaka_roka, count(*) as broj_ispita, count(distinct indeks) broj_studenata
from ispit
group by godina_roka, oznaka_roka
having godina_roka=2011 and min(ocena)>5;
```

Primer 13: Za svaki ispitni rok izdvojiti naziv ispitnog roka, najveću ocenu dobijenu u tom ispitnom roku i ime i prezime studenta koji je dobio tu ocenu. Ime i prezime studenta

napisati u jednoj koloni. Za ispitne rokove u kojima nije bilo ispita, kao ime i prezime studenta ispisati *nema*, a kao ocenu 0.

```
select naziv, coalesce(ime || ' ' || prezime, 'nema'), coalesce(ocena, 0)
from ispitni_rok ir left outer join ispit i on i.godina_roka=ir.godina_roka and
i.oznaka_roka=ir.oznaka_roka
      left outer join dosije d on i.indeks=d.indeks
where ocena = (select max(ocena)
              from ispit i2
              where i.godina_roka=ir.godina_roka and ir.oznaka_roka=i2.oznaka_roka) or
ocena is null;
```

Primer 14: Prikazati naziv predmeta koji je položio samo student Milos Peric.

```
select naziv
from predmet p join ispit i on p.id_predmeta=i.id_predmeta
      join dosije d on d.indeks=i.indeks
where ime='Milos' and prezime='Peric' and ocena>5
      and not exists ( select *
                    from ispit i2
                    where ocena>5 and i2.indeks<>d.indeks and
                    i2.id_predmeta=p.id_predmeta);
```

Primer 15: Izdvojiti parove studenata čija imena počinju na slovo *M* i za koje važi da su bar dva ista predmeta položili u istom ispitnom roku.

```
select d1.indeks, d1.ime, d1.prezime, d2.indeks, d2.ime, d2.prezime
from dosije d1, dosije d2
where d1.indeks<d2.indeks and 2 <= (select count(*)
                                   from ispit i1 join ispit i2 on i1.id_predmeta=i2.id_predmeta
                                   and
                                   i1.godina_roka=i2.godina_roka and
                                   i1.oznaka_roka=i2.oznaka_roka
                                   where d1.indeks=i1.indeks and d2.indeks=i2.indeks
                                   and i1.ocena>5 and i2.ocena>5)
and d1.ime like 'M%' and d2.ime like 'M%';
```