

Uvod u relacione baze podataka

Ana Vulović *

Matematički fakultet, Univerzitet u Beogradu

12. čas

1 Merge naredba

1. Napisati SQL naredbe kojima se:

- kreira tabela *predmet_student* u kojoj se čuvaju informacije o broju studenata koji su uspešno položili ispit iz tog predmeta. Tabela sadrži kolone *id_predmeta* i *br_studenata*. Kolona *id_predmeta* je i primarni ključ. Definisati i strani ključ na tabelu predmet.
- popunjava tabela *predmet_student* sa obaveznim predmetima sa smerom sa identifikatorom 201. Prilikom popunjavanja, vrednost u koloni *br_studenata* postaviti na 5.
- dopunjava sadržaj tabele *predmet_student* sa validnim podacima za sve predmete iz tabele predmet, tako što se ažuriraju podaci za već postojeće predmete u tabeli *predmet_student*, i dodavanjem novih redova za ostale predmete.
- briše tabelu *predmet_student*.

```
create table predmet_student (  
id_predmeta integer not null primary key,  
br_studenata smallint,  
foreign key fk_predmet(id_predmeta) references predmet )
```

```
insert into predmet_student  
select id_predmeta, 5  
from obavezan_predmet  
where id_smera = 201
```

```
select * from predmet_student
```

```
merge into predmet_student ps  
using (select p.id_predmeta, count(i.indeks) br  
from ispit i right join predmet p  
on i.id_predmeta = p.id_predmeta
```

*ana_vulovic@matf.bg.ac.rs

```

        and ocena > 5 and status_prijave = 'o'
    group by p.id_predmeta) as pom
on ps.id_predmeta = pom.id_predmeta
when matched then
    update set
        ps.br_studenata = pom.br
when not matched then
    insert
        values(pom.id_predmeta, pom.br)

select * from predmet_student

drop table predmet_student

```

2. Napisati SQL naredbe kojima se:

- (a) kreira tabela *student_podaci* u kojoj se čuvaju sledeće informacije: indeks, broj položenih predmeta, prosek i datum rođenja studenta. Kolona indeks je i primarni ključ. Definisati i strani ključ na tabelu dosije.
- (b) u kreiranu tabelu unose informacije za studente čiji prosek ocena je veći od 8, pri čemu se ne navodi datum rođenja i za studente koji imaju status *diplomirao* postavlja se broj položenih predmeta na 40, prosek na 10.00 i datum rođenja ostaje nepoznat.
- (c) dopunjava sadržaj tabele *student_podaci* sa validnim podacima za sve studente iz tabele dosije, tako što se:
 - ažurira datum rođenja studentima koji su već u tabeli i imaju status diplomiranog studenta,
 - studentima koji su već u tabeli i na budžetu su, ažuriraju se podaci o broju položenih predmeta i proseku.
 - studentima koji su već u tabeli, a imaju status *ispisan*, brišu se iz tabele *predmet_student*
 - studenti koji se ne nalaze u tabeli *predmet_student* i kojima status nije *ispisan*, uneti indeks, validan broj položenih predmeta i prosek.
 - za sve ostale ne raditi ništa.

Aktuelni status je poslednji status studenta u tabeli *status*.

```

create table studenti_podaci (
    indeks integer not null primary key,
    broj_polozenih smallint,
    prosek float,
    datum_rodjenja date,
    foreign key (indeks) references dosije )

```

```

insert into studenti_podaci
with diplomirani as (

```

```

select indeks
from dosije d
where 'diplomirao' in (select status
                        from status
                        where indeks = d.indeks)
)
select indeks, count(*), avg(ocena+0.0), null
from ispit
where ocena > 5 and status_prijave = 'o'
   and indeks not in (select indeks from diplomirani)
group by indeks
having avg(ocena+0.0) > 8
union
select indeks, 40, 10.00, null
from diplomirani

merge into studenti_podaci sp
using (select d.indeks, status, datum_rodjenja, count( case
                when ocena > 5 and status_prijave = 'o' then id_predmeta
                else null
            end ) brp,
        avg( (case
                when ocena > 5 and status_prijave = 'o' then ocena
                else null
            end) +0.0) prosek
        from dosije d left join ispit i on d.indeks = i.indeks
        left join status s on s.indeks = d.indeks
        where not exists (select *
                          from status
                          where indeks = d.indeks
                          and datum > s.datum)
        -- eliminisemo studente koji su poslednji put kad su menjali status,
        -- u istom danu promenili uneli 2 ili vise statusa
        and not exists (select *
                        from status
                        where indeks = d.indeks and datum = s.datum
                        and status <> s.status)
        group by d.indeks, status, datum_rodjenja) as pom
on pom.indeks = sp.indeks
when matched and pom.status = 'diplomirao' then
update set
    sp.datum_rodjenja = pom.datum_rodjenja
when matched and status = 'budzet' then
update set
    (sp.broj_polozenih, sp.prosek) = (pom.brp, pom.prosek)
when matched and pom.status = 'ispisan' then
delete
when not matched and pom.status <> 'ispisan' then
insert (indeks, broj_polozenih, prosek)
values (pom.indeks, pom.brp, pom.prosek )

```

```
else ignore
```

2 SQL - zadaci za vežbanje

1. Napisati upit na SQL-u koji za školsku godinu i smer osnovnih studija izračunava koliki je procenat studenata koji su te godine upisali fakultet, upisan upravo na taj smer, kao i koja je prosečna ocena studenata tog smera na ispitima u toku te školske godine. Izdvojiti oznaku i naziv smera, školsku godinu, ukupan broj upisanih studenata na fakultet te godine i prosečnu ocenu smera u toj generaciji na kursevima slušanim te školske godine. Izveštaj urediti po godini i prosečnoj oceni.

```
with smerovi as(
  select d.id_smera, indeks/10000 godina, count(*) br_studenata
  from dosije d join smer s
    on s.id_smera = d.id_smera
    join nivo_kvalifikacije nk
    on s.id_nivoa = nk.id_nivoa
  where nk.naziv like 'Osnovne%'
  group by d.id_smera, indeks/10000 ),
upisani_po_godini as(
  select godina, sum(br_studenata) br_studenata
  from smerovi
  group by godina ),
prosek_po_smeru as(
  select id_smera, godina, avg(ocena+0.0) prosek
  from ispit i join dosije d
    on d.indeks= i.indeks
  where ocena >5 and status_prijave ='o'
  group by id_smera, godina )
select s.oznaka, s.naziv, sm.godina,
  100.0*sm.br_studenata/ upg.br_studenata procenat, prosek
from smerovi sm join smer s
  on s.id_smera = sm.id_smera
  join upisani_po_godini upg on sm.godina = upg.godina
  left join prosek_po_smeru pps
    on pps.id_smera = sm.id_smera and pps.godina = sm.godina
order by godina, prosek desc
```

2. * Za sve studente čije ime počinje sa „Mil“ izdvojiti spisak do sada položenih predmeta. Izdvojiti: broj indeksa, ime, prezime, oznaku smera, šifru predmeta, naziv predmeta, školsku godinu, semestar i ocenu. Izveštaj urediti prema broju indeksa, školskoj godini, semestru i šifri predmeta.

```
select d.indeks,d.ime, d.prezime, s.oznaka, p.sifra, p.naziv,
  i.godina, i.semestar, ocena
from dosije d left outer join ispit i
```

```

    on d.indeks = i.indeks and ocena > 5 and status_prijave = 'o'
  join smer s on s.id_smera = d.id_smera
  left outer join predmet p on p.id_predmeta = i.id_predmeta
where substr(d.ime, 1,3) = 'Mil'
order by d.indeks, i.godina, i.semestar, p.sifra

```

Uslov za ocenu i status_prijave stavljemo u uslov spajanja jer želimo da se izdvoje i oni koji nemaju nijedno uspešno polaganje, iako imaju već neko polaganje koje nije bilo uspešno. Ako bismo spajali samo po indeksu takvo jedno polaganje bi moglo da se spoji, ali bi posle bilo izbačeno uslovom po oceni, statusu prijave ili da id_predmeta je NULL. Neophodno je da se spoje dva reda i po indeksu i da važi uslov po oceni i statusu prijave, tj. da se spoji samo sa uspešnim polaganjem, ako je moguće.

- * Izdvojiti podatke o uspešnosti polaganja na smeru „Racunarstvo i informatika“ u školskoj godini 2006/2007. Izdvojiti šifru predmeta, naziv predmeta, ispitni rok, broj prijavljivanih studenta, broj studenata koji su ga položili. Izveštaj urediti prema prosečnoj oceni.

```

select p.sifra, substr(p.naziv,1, 30), i.oznaka_roka,
       count(*) prijavilo,
       sum(case when ocena > 5 and status_prijave = 'o' then 1
              else 0 end ) polozilo,
       avg(case when ocena > 5 and status_prijave = 'o'
             then ocena + 0.0
             else null end) prosek
from ispit i join predmet p on p.id_predmeta = i.id_predmeta
  join dosije d on d.indeks = i.indeks
  join smer s on s.id_smera = d.id_smera
where s.naziv = 'Racunarstvo i informatika' and i.godina = 2006
group by p.sifra, p.naziv, i.oznaka_roka
order by prosek desc

```

- Za smerove na osnovnim studijama izdvojiti podatke o obaveznim predmetima koje su u školskoj 2007/2008. godini neki studenti ponovljeno upisali. Izdvojiti oznaku smeru, naziv smeru, naziv predmeta, ukupan broj studenta upisanih na kurs iz tog predmeta u školskoj 2007/2008. godini, broj studenta kojima to nije prvi put da su upisali kurs iz tog predmeta, procenat studenata koji su ponovo upisali kurs. Izveštaj urediti prema procentu studenata koji su ponovili upis u opadajućem poretku.

```

with upisali as(
  select s.id_smera, p.id_predmeta, d.indeks
  from nivo_kvalifikacije nk join smer s
    on nk.id_nivoa = s.id_nivoa
  join obavezan_predmet op on op.id_smera = s.id_smera
  join predmet p on op.id_predmeta = p.id_predmeta
  join upisan_kurs uk on uk.id_predmeta = op.id_predmeta

```

```

        join dosije d on d.indeks = uk.indeks
        and d.id_smera = s.id_smera
where nk.naziv like 'Osnovne%' and uk.godina = 2007
),
ponovili as(
select id_smera, id_predmeta, count(indeks) br_ponovo
from upisali u
where exists( select * from upisan_kurs
              where indeks = u.indeks and
                    id_predmeta = u.id_predmeta and godina < 2007 )
group by id_smera, id_predmeta
)
select s.oznaka, s.naziv, p.naziv, count(u.indeks) br_upisanih,
br_ponovo,
case
when count(u.indeks) <> 0
then coalesce(br_ponovo,0) * 100.0 / count(u.indeks)
else null
end as procenat
from upisali u right join smer s
on u.id_smera = s.id_smera
left join predmet p on u.id_predmeta = p.id_predmeta
left join ponovili pon on u.id_smera = pon.id_smera
and u.id_predmeta = pon.id_predmeta
where s.id_nivoa in (select id_nivoa
                   from nivo_kvalifikacije
                   where naziv like 'Osnovne%')
group by s.id_smera, p.id_predmeta, br_ponovo,
s.oznaka, s.naziv, p.naziv
order by br_upisanih desc

```

5. Neka je data tabela prvi_ispit:

```

create table prvi_ispit(
indeks integer not null primary key,
datum date)

```

koja sadži podatke o prvim položenim ispitima za sve studente.
Napisati naredbu SQL-a kojom se:

- a) studentima koji imaju neki položen ispit unose ispravni podaci u tabelu prvi_ispit
- b) iz tabele prvi_ispit brišu podaci za studente osnovnih studija
- c) za sve studente doktorskih studija umesto prvog upisuje datum poslednjeg položenog ispita
- d) objedinjuje posao iz b) i c)

```

create table prvi_ispit(

```

```

indeks integer not null primary key,
datum date)

insert into prvi_ispit
select indeks, min(coalesce(datum_usmenog, datum_pismenog) )
from ispit
where ocena > 5 and status_prijave = 'o'
group by indeks

delete from prvi_ispit
where indeks in (select indeks
                 from dosije d join smer s on s.id_smera = d.id_smera
                 join nivo_kvalifikacije nk on nk.id_nivoa = s.id_nivoa
                 where nk.naziv like 'Osnovne%')

update prvi_ispit as pi
set datum = (select max(coalesce(datum_usmenog, datum_pismenog))
             from ispit where indeks = pi.indeks and ocena > 5
             and status_prijave = 'o')
where indeks in (select indeks
                 from dosije d join smer s on s.id_smera = d.id_smera
                 join nivo_kvalifikacije nk on nk.id_nivoa = s.id_nivoa
                 where nk.naziv like 'Doktorske%')

merge into prvi_ispit pi
using(
select d.indeks, nk.naziv,
       max(coalesce(datum_usmenog, datum_pismenog)) as najveci_datum
from (dosije d join smer s on d.id_smera = s.id_smera
     join nivo_kvalifikacije nk on s.id_nivoa = nk.id_nivoa)
     join ispit i on d.indeks = i.indeks
where ocena > 5 and status_prijave = 'o'
     and (nk.naziv like 'Osnovne%' or nk.naziv like 'Doktorske%')
group by d.indeks, nk.naziv) as pom
on pi.indeks = pom.indeks
when matched and pom.naziv like 'Osnovne%' then
delete
when matched and pom.naziv like 'Doktorske%' then
update
set pi.datum = pom.najveci_datum
else
ignore

drop table prvi_ispit

```