1.

```
select d.indeks, ime, prezime,  naziv, ocena
     case when i.id_predmeta in (select id_predmeta
                         from obavezan_predmet op
                         where op.id_smera=d.id_smera) then 'obavezan'
          else 'izborni'
     end predmet
from dosije d join ispit i on d.indeks=i.indeks
    join predmet p on p.id_predmeta=i.id_predmeta
 where ocena>5 and status_prijave='o' and ime like 'P%' and month(dat_rodjenja) between 2 and
7;
```

2.

```
create trigger brisanje_studenta
            before delete
            on dosije
            referencing old as stari
            for each row
            when(stari.status <> 'ispisan')
            begin atomic
              signal sqlstate '75000' ('Student ciji status nije ispisan ne moze se obrisati.');
            end@

insert into dosije (indeks, id_smera, status, ime, prezime, pol, jmbg, dat_upisa)
values(20090206, 201, 'budzet', 'Marko', 'Markovic','m', '1205989035819', current_date)@

delete from dosije
where indeks=20090206@

update dosije
set status='ispisan'
where indeks=20090206@
```

3.

```
create trigger promena_bodova before update of bodovi on predmet
referencing old as staro new as nova
for each row
begin atomic
 set nova.bodovi= case
                  when staro.bodovi > nova.bodovi then staro.bodovi-1
                  when staro.bodovi < nova.bodovi then staro.bodovi+1
                  else staro.bodovi
                 end;
end@

insert into predmet(id_predmeta, sifra, naziv, semslus, bodovi)
            values(1601, 'pred1', 'predmet 1',  1, 4)@
```

```
select *
from predmet
where id_predmeta=1601@

update predmet
set bodovi=25
where id_predmeta=1601@

select *
from predmet
where id_predmeta=1601@

update predmet
set bodovi=10
where id_predmeta=1601@

select *
from predmet
where id_predmeta=1601@

update predmet
set bodovi=10, sifra='pr1'
where id_predmeta=1601@

select *
from predmet
where id_predmeta=1601@

drop trigger promena_bodova@
```

4.

```
create table broj_predmeta(broj smallint)@

insert into broj_predmeta select count(*) from predmet@

select *
from broj_predmeta@

create trigger novi_pred
        after insert
        on predmet
        for each row
        begin atomic
            update broj_predmeta
            set broj = broj + 1;
            end@
```

```
create trigger brisanje_pred
            after delete
            on predmet
            for each row
            begin atomic
                update broj_predmeta
                set broj = broj - 1;
                end@

insert into predmet (id_predmeta, sifra, naziv, semslus, bodovi)
                values(2002, 'predm13', 'predmet 13', 1, 15)@

select *
from broj_predmeta@

delete from predmet
where id_predmeta=2002@

select *
from broj_predmeta@
```

5.

```
create table predmet_student (
  id_predmeta integer,
  studenti smallint
);

insert into predmet_student
select id_predmeta, 2
from obavezan_predmet
where id_smera=201;

merge into predmet_student ps
using (select id_predmeta, count(*) n
      from ispit
      where ocena>5 and status_prijave='o'
      group by id_predmeta) as p
on ps.id_predmeta=p.id_predmeta
when matched then
  update set
    ps.studenti=p.n
when not matched then
  insert
  values (p.id_predmeta, p.n);
```

6.

```
create table student_podaci (
```

```
 indeks integer,
 broj_predmeta smallint,
 prosek float,
 dat_rodjenja date);

insert into student_podaci
select indeks, count(*), avg(ocena+0.0), null
from ispit
where ocena>5 and status_prijave='o'
group by indeks
having avg(ocena+0.0)>8
union
select indeks, 10, 10.00, null
from dosije
where status='diplomirao';

merge into student_podaci sp
using (select d.indeks, status, dat_rodjenja, count(i.id_predmeta) brp, avg(ocena+0.0) prosek
      from dosije d left outer join ispit i on d.indeks=i.indeks and ocena>5 and status_prijave='o'
      group by   d.indeks, status, dat_rodjenja) as p
on sp.indeks=p.indeks
when matched and p.status='diplomirao' then
 update set
   sp.dat_rodjenja=p.dat_rodjenja
when matched and p.status='ispisan' then
 delete
when matched and p.status='budzet' then
 update set
   (sp.broj_predmeta, sp.prosek)=(p.brp, p.prosek)
when not matched and p.status<>'ispisan' then
 insert (indeks, broj_predmeta, prosek)
 values (p.indeks, p.brp, p.prosek)
else ignore;
```

7.

```
create distinct type prosek as float with comparisons;

create table prosek (
indeks int not null,
prosek prosek,
primary key (indeks),
foreign key fk_dosije(indeks)
references dosije
on delete cascade
);

create function prosek(indeks_studenta int)
returns prosek
return select avg(cast(ocena as float))
```

```
      from ispit i
      where i.indeks=indeks_studenta
          and ocena>5 and status_prijave='o';

select indeks, dec(cast(prosek(indeks) as float), 4,2)
from dosije;

insert into prosek
select indeks, prosek(indeks)
from dosije;

delete from prosek
where indeks not in ( select indeks
                from upis_godine);

delete from ispit
where indeks=20070208 and id_predmeta=635;

create trigger unos_ispita
     after insert on ispit
     referencing new as dodati
     for each row
     begin
      update prosek
      set prosek=prosek(dodati.indeks)
      where indeks=dodati.indeks;
     end@

insert into ispit (indeks, id_predmeta, godina, semestar, godina_roka, oznaka_roka, datum_prijav,
nacin_prijav, status_prijave,bodovi_pismenog, datum_pismenog, bodovi, ocena)
values(20070208, 635, 2007, 1, 2007, 'jan', '02-02-2007', 'auto', 'o', 85, '02-15-2007', 85, 9);
```