

Neka je dat sledeći model baze podataka:

tip_vozila (**idTipaVozila**, **naziv**) - Spisak svih tipova vozila.

marka_vozila (**idMarkeVozila**, **naziv**) - Spisak svih marki vozila.

modeli_vozila (**idModela**, **idTipaVozila**, **idMarkeVozila**, **nazivModela**) - Spisak svih modela vozila.

osobe (**idOsobe**, **ime**) - Spisak svih osoba koje mogu biti vlasnici ili kupci vozila.

vozila (**idVozila**, **idModela**, **kubikaža**, **snaga**, **tipGoriva**, **godinaProizvodnje**, **idVlasnika**) - Spisak svih proizvedenih vozila.

oglasi (**idOglasa**, **idVozila**, **datumPostavljanja**, **datumZavršetka**, **početnaCena**) - Spisak svih vozila koja se prodaju. Oglas je aktivan u periodu [**datumPostavljanja** - **datumZavršetka**].

ponude (**idPonude**, **idOglasa**, **datumPonude**, **ponudjenaCena**, **idPonuđača**) - Spisak svih ponuda za vozila.

Relaciona algebra (7 poena)

Ključevi (1 poen)

Ako su primarni ključevi na relaciji **ponude**:

- PK(**idOglasa**)
- PK(**idPonude**, **idOglasa**)
- PK(**idOglasa**, **ponudjenaCena**)

Koliko različitih ponuda može da postoji za isti oglas i pod kojim ograničenjima?

Relaciona algebra

Operacijama relacione algebre definisati relaciju koja sadrži:

- (1 poen)** Spisak vozila (**idVozila**, **kubikaža**, **snaga**) koja još nisu postavljena na oglas.
- (2 poen)** Spisak modela vozila koja nikada nisu postavljena u oglasima (**idModela**, **nazivModela**).
- (3 poena)** Najveće jedinstvene ponude za svaki oglas (**idOglasa**, **ponudjenaCena**).

SQL (30 poena)

1. (3.5 poena) Napisati SQL upit koji za svakog vlasnika vozila određuje koliko je vozila u njegovom vlasništvu.

```
select v.idVlasnika, o.ime, count(*) brojVozila
from vozila v join osobe o on v.idVlasnika = o.idOsobe
group by v.idVlasnika, o.ime
```

	idVlasnika	ime	brojVozila
1	1	Marko	19
2	2	Andreja	11
3	3	Gavra	13

2. (4.5 poena) Napisati SQL upit koji određuje koliko ima vozila sa jedinstvenom godinom proizvodnje.

```
create view jedinstvena as
select count(*) as jedinstvena
from vozila
group by godinaProizvodnje
having count(*) = 1

select sum(jedinstvena) as 'Broj jedinstvenih'
from jedinstvena
```

	Broj jedinstvenih
1	2

3. (5.5 poena) Napisati SQL upit kojim se određuju najveće jedinstvene ponude sa svaki oglas. Obavezno je korišćenje ugnježđenih upita i **nije** dozvoljeno koristiti **JOIN** mehanizme.

```
select p1.idOglasa, max(p1.ponudjenaCena) as 'Najveca jedinstvena'
from ponude p1
where not exists (
    select *
    from ponude p2
    where p1.idOglasa = p2.idOglasa and
          p1.idPonude != p2.idPonude and
          p1.ponudjenaCena = p2.ponudjenaCena
)
group by p1.idOglasa
```

	idOglasa	Najveća jedinstvena
1	1	2500
2	2	2400
3	3	3000
4	4	1500
5	5	2560
6	6	2890
7	7	11100
8	10	5000
9	11	700
10	12	2300
11	13	2400
12	14	2300
13	15	1600

4. (8 poena) Napisati SQL upit kojim se pronalazi osoba koja **kupuje** vozila sa najvećom kubikažom u proseku.

```
create view potencijalni_kupci as
select p1.*
from ponude p1 left join ponude p2 on p1.idOglasa = p2.idOglasa and p1.ponudjenaCena < p2.ponudjenaCena
where p2.idPonude is null
--order by p1.idPonudjaca, p1.datumPonude
order by idOglasa, ponudjenaCena

create view kupci_vozila as
select p1.*
from potencijalni_kupci p1 left join potencijalni_kupci p2 on p1.idOglasa = p2.idOglasa and p1.datumPonude > p2.datumPonude
where p2.idOglasa is null
order by p1.idPonudjaca, p1.datumPonude

create view prosecne_kubikaze as
select kv.idPonudjaca, avg(v.kubikaza * 1.8) prosecnaKubikaza
from kupci_vozila kv join oglasi o on kv.idOglasa = o.idOglasa join vozila v on o.idVozila = v.idVozila
group by kv.idPonudjaca

select idPonudjaca
from prosecne_kubikaze
where prosecnaKubikaza = (select max(prosecnaKubikaza) from prosecne_kubikaze)
```

	idPonudjaca
1	5

5. (8.5 poena) Napisati SQL upit kojim se određuju osobe koje su svako novo vozilo koje su kupili, platili više od prethodnog.

```
--5
create view losi_ponudjaci as
select p1.idPonudjaca
from kupci_vozila p1 join kupci_vozila p2 on p1.idPonudjaca = p2.idPonudjaca and
p1.ponudjenaCena < p2.ponudjenaCena and
p1.datumPonude > p2.datumPonude

select distinct p1.idPonudjaca
from kupci_vozila p1
where idPonudjaca not in (select idPonudjaca
from losi_ponudjaci
)
```

Results		Messages
	idPonudjaca	
1	5	
2	7	

Napomena: Osoba koja je kupila vozilo je ponudila najveću cenu za njega. Ukoliko je više osoba ponudilo najveću cenu za jedno vozilo, smatrati da ga je kupila ona osoba koja je prva ponudila tu cenu.