

## Međuispit iz Baza podataka

20. travnja 2012.

Zadaci 1 - 3 odnose se na bazu podataka TERETANA prikazanu na **slici 1**. Na slici nisu prikazane sve n-torke koje su sadržane u relacijama.

clan				dolazak			vjezba		
sifClan	prezime	ime	datIstek	sifDolazak	sifClan	datDolazak	sifDolazak	sifSprava	minuta
1111	Horvat	Ivan	25.05.2012	55	1111	11.03.2012	55	101	25
1112	Novak	Jura	15.03.2012	56	1111	23.03.2012	55	103	25
1113	Horvat	Stipe	01.06.2012	57	1112	25.02.2012	56	104	15
1114	Herceg	Ante	17.04.2012	58	1113	23.03.2012	57	105	20
1115	Kluk	Antun	11.02.2012				58	102	10
							58	103	30

sprava	
sifSprava	naziv
101	Orbitrek
102	Butterfly
103	Abtoner
104	Bicikl
105	Legwork

Slika 1.

U relacije sa slike 1 pohranjuju se podaci o članovima teretane (relacija **clan**) te o njihovim dolascima u teretanu (relacija **dolazak**). Svaki dolazak člana vezan je uz određeni datum, a član teretane može u jednome danu i više puta doći u teretanu. U sklopu jednog dolaska član teretane vježba na spravama te se u relaciju **vjezba** zapisuje koliko je **ukupno** minuta unutar tog dolaska vježbao na određenoj spravi (relacija **sprava**). Za svakog se člana uz ime i prezime bilježi i datum kada mu ističe članstvo (datum isteka je zadnji dan kada članstvo još vrijedi).

1. Napišite po jednu SQL naredbu kojom će se obaviti sljedeće:

- Za svakog člana ispisati njegovu šifru, inicijale (npr. I.H.) te status članstva na dan izvođenja upita – ako je isteklo ispisati 'isteklo članstvo', ako nije ispisati 'aktivni član'. **(2 boda)**
- Za svaku korištenu spravu ispisati šifru i naziv sprave, šifru i prezime osobe koja je na njoj najdulje trenirala tijekom jednog dolaska te duljinu trajanja te vježbe. U slučaju da više osoba dijeli maksimalno vrijeme rezultat treba sadržavati podatke svih tih osoba. Rezultate poredati silazno prema trajanju vježbe, a unutar toga uzlazno prema šifri sprave. **(2.5 boda)**
- Ispisati šifre i prezimena svih članova s trenutno aktivnim članstvom koji su u veljači 2012. godine ukupno trenirali više od 10 sati. **Zadatak riješiti bez podupita. (2.5 boda)**
- Za **sve evidentirane sprave** ispisati šifru i naziv sprave te koliko su ukupno puta članovi čije prezime počinje na 'H' ili 'P' vježbali na njoj. **Zadatak riješiti bez podupita. (2.5 boda)**
- Svim članovima koji su u tekućem mjesecu barem jednom došli na trening a članstvo im ističe unutar sljedeća dva tjedna produljiti članstvo za 15 dana. **(2.5 boda)**

2. Napišite po jedan **izraz relacijske algebre** (ne SQL upit) koji odgovara sljedećem:

- Za sve sprave koje su od strane barem jednog korisnika u jednome danu korištene više od 45 minuta ispisati šifru i naziv sprave, te koliko su maksimalno minuta korištene na jednom treningu. **(2 boda)**
- Ispisati šifre svih sprava na kojima se vježbalo 12. travnja, ali nije 13. travnja 2012. godine. **(2 boda)**

3. Uz pretpostavku da slika 1 prikazuje sve podatke pohranjene u bazi podataka, u obliku tablice **prikažite rezultate** obavljanja sljedećih operacija:

a) 
$$\left[ \pi_{\text{prezime}} \left( \sigma_{\text{naziv}='Orbitrek'} (\text{clan} \bowtie \text{dolazak} \bowtie \text{vjezba} \bowtie \text{sprava}) \right) \right] \cup \left[ \pi_{\text{prezime}} \left( \sigma_{\text{naziv}='Abtoner'} (\text{clan} \bowtie \text{dolazak} \bowtie \text{vjezba} \bowtie \text{sprava}) \right) \right] \quad (1 \text{ bod})$$

b)  $\pi_{\text{sifClan}, \text{sifDolazak}} (\sigma_{\text{datIstek} > '01.04.2012'} (\text{clan} * \bowtie \text{dolazak})) \quad (1 \text{ bod})$

4. U bazi podataka evidentiraju se podaci o servisu sprava u jednoj teretani. Teretana ima nekoliko dvorana i određena sprava se može nalaziti samo u jednoj dvorani. Određeni servis jedne sprave obavlja jedan serviser. Ista sprava se u jednom danu može servisirati samo jednom. Svaki serviser radi u određenoj tvrtki. Moguće je da dva serviseri iz različitih tvrtki imaju istu šifru.

Relacijska shema SERVIS sastoji se od sljedećih atributa:

- sifSprava - jedinstvena šifra sprave
- nazSprava - naziv sprave
- sifDvorana - šifra dvorane
- kvadrDvorana - kvadratura dvorane
- datServis - datum obavljenog servisa
- sifServiser - šifra serviseri
- prezServiser - prezime serviseri
- sifTvrta - šifra tvrtke
- nazTvrta - naziv tvrtke

Odredite ključ relacijske sheme SERVIS tako da ona bude u 1NF, a zatim postupno normalizirajte relacijsku shemu na 2NF i 3NF. (4 boda)

5. Koji je maksimalan broj n-torki koje može imati relacija nad kojom je izgrađeno B<sup>+</sup>-stablo reda 100 dubine 4? Objasniti rješenje. (2 boda)

6. Napisati SQL naredbe koje će kreirati najmanji mogući broj indeksa za učinkovito obavljanje (pomoću B<sup>+</sup> stabla) svih dolje navedenih upita. (3 boda)

- 1) SELECT \* FROM clan WHERE datIstek = TODAY AND prezime = 'Horvat';
- 2) ELECT \* FROM clan WHERE sifclan < 1200;
- 3) SELECT \* FROM clan WHERE datIstek BETWEEN '01.04.2012.' AND '01.05.2012.';
- 4) SELECT \* FROM clan ORDER BY datIstek ASC, prezime DESC;
- 5) SELECT \* FROM clan ORDER BY prezime, ime;
- 6) SELECT \* FROM clan ORDER BY sifClan, prezime;

7. Napisati SQL naredbu koja će kreirati relacijsku shemu *BON*= { *serijskiBroj*, *sifClan*, *datIzd*, *datIskor*, *popustKn* }. Shema *BON* opisuje promotivne bonove za popust na obnovu članarine u teretani. Kod kreiranja relacijske sheme smisleno odabrati tipove podataka te osigurati sljedeće: (3 boda)

- serijski broj jedinstveno identificira bon
- podaci o bonu se evidentiraju kod izdavanja, prije nego je poznato koji član će ga iskoristiti i kada
- korištenje bona evidentirati će se unosom šifre postojećeg člana (iz relacije CLAN sa Slike 1.) te datuma iskorištavanja
- datum izdavanja i popust u kunama moraju biti poznati
- popust u kunama može poprimiti samo vrijednosti 50 ili 100
- bon se mora iskoristiti unutar 90 dana nakon izdavanja
- ukoliko se član koji je iskoristio bon izbriše, informaciju o bonu zadržati a šifru člana postaviti na NULL

## Rješenja:

1. Napišite po jednu SQL naredbu kojom će se obaviti sljedeće:

- a) Za svakog člana ispisati njegovu šifru, inicijale (npr. I.H.) te status članstva na dan izvođenja upita – ako je isteklo ispisati 'isteklo članstvo', ako nije ispisati 'aktivni član'. **(2 boda)**

```
SELECT sifclan,  
       (SUBSTRING(ime FROM 1 FOR 1))||'.'||(SUBSTRING(prezime FROM 1 FOR  
1))||'.' AS inicijali,  
       CASE  
         WHEN datistek<TODAY THEN 'isteklo članstvo'  
         ELSE 'aktivni član'  
       END AS stanje  
FROM clan;
```

- b) Za svaku korištenu spravu ispisati šifru i naziv sprave, prezime osobe koja je na njoj najdulje trenirala tokom jednog dolaska te duljinu trajanja te vježbe. U slučaju da više osoba dijeli maksimalno vrijeme rezultat treba sadržavati podatke svih tih osoba. Rezultate poredati silazno prema trajanju vježbe, a unutar toga uzlazno prema šifri sprave. **(2.5 boda)**

```
SELECT DISTINCT sprava.sifSprava, naziv, clan.sifclan, prezime,  
minuta  
FROM clan  
      JOIN dolazak ON clan.sifClan=dolazak.sifClan  
      JOIN vjezba ON vjezba.sifdolazak=dolazak.sifdolazak  
      JOIN sprava ON vjezba.sifSprava=sprava.sifSprava  
WHERE vjezba.minuta = (SELECT MAX(minuta) FROM vjezba v WHERE  
v.sifSprava=vjezba.sifSprava)  
ORDER BY minuta desc, sifSprava asc;
```

- c) Ispisati šifre i prezimena svih članova s trenutno aktivnom članarinom koji su u veljači 2012. godine ukupno trenirali više od 10 sati. **Zadatak riješiti bez podupita.** **(2.5 boda)**

```
SELECT clan.sifClan, prezime  
FROM clan  
      JOIN dolazak  
        ON clan.sifclan=dolazak.sifClan  
      JOIN vjezba  
        ON vjezba.sifDolazak=dolazak.sifDolazak  
WHERE datistek>TODAY  
      AND MONTH(datdolazak)=2 AND YEAR(datdolazak)=2012  
GROUP BY clan.sifClan, prezime  
HAVING SUM(minuta)>10*60;
```

- d) Za **sve evidentirane sprave** ispisati šifru, naziv te koliko su ukupno puta članovi čije prezime počinje na 'H' ili 'P' vježbali na njoj. **Zadatak riješiti bez podupita.** **(2.5 boda)**

```
SELECT sprava.sifSprava, naziv, COUNT(dolazak.sifDolazak) AS vjezbaoputa  
FROM clan  
      JOIN dolazak  
        ON clan.sifclan=dolazak.sifClan  
        AND (prezime like 'P%' or prezime like 'H%')  
      JOIN vjezba  
        ON vjezba.sifdolazak=dolazak.sifDolazak  
      RIGHT JOIN sprava  
        ON vjezba.sifSprava=sprava.sifsprava  
GROUP BY sprava.sifSprava, naziv;
```

Alt. rješenje:

```
SELECT sprava.sifSprava, naziv, COUNT(clan.sifClan) AS vjezbaoPut  
FROM sprava  
LEFT JOIN vjezba ON vjezba.sifSprava=sprava.sifSprava  
LEFT JOIN dolazak ON vjezba.sifDolazak=dolazak.sifDolazak  
LEFT JOIN clan ON clan.sifClan=dolazak.sifClan  
AND (prezime like 'P%' or prezime like 'H%')  
GROUP BY sprava.sifSprava, naziv  
ORDER BY sprava.sifSprava;
```

- e) Svim članovima koji su u tekućem mjesecu barem jednom došli na trening a članstvo im ističe unutar sljedeća dva tjedna produžiti članstvo za 15 dana. **(2.5 boda)**

```
UPDATE clan  
SET datistek = datistek+15  
WHERE datistek BETWEEN TODAY AND TODAY+14  
AND sifclan IN  
(SELECT sifClan FROM dolazak WHERE  
MONTH(datdolazak)=MONTH(TODAY)  
AND YEAR(datdolazak)=YEAR(TODAY));
```

2. Napišite po jedan **izraz relacijske algebre** (ne SQL upit) koji odgovara sljedećem:

- a) Za sve sprave koje je su od strane barem jednog korisnika u jednome danu korištene više od 45 minuta ispisati šifru, naziv, te koliko su maksimalno minuta korištene na jednom treningu. **(2 boda)**

-- prihvatljivo rješenje koje pretpostavlja da nijedan član nije unutar više dolazaka u jednom danu vježbao na istoj spravi:

$\rho_{\text{ispis}}(\text{sifSprava}, \text{naziv}, \text{maxMin}) \llbracket \text{sifSprava}, \text{naziv} \mathrel{G} \text{MAX}(\text{minuta}) \left( \sigma_{\text{minuta} > 45}(\text{sprava} \bowtie \text{vjezba}) \right) \rrbracket$

--potpuno ispravno rješenje:

$\rho_{\text{ispis}}(\text{sifSprava}, \text{naziv}, \text{maxMin}) \llbracket \text{sifSprava}, \text{naziv} \mathrel{G} \text{MAX}(\text{minuta}) \left[ \left( \pi_{\text{sifSprava}} \sigma_{\text{minPoDanu} > 45}(\mathbf{A}) \right) \bowtie \text{vjezba} \right] \rrbracket$

gdje je  $\mathbf{A} = \rho_{\text{ispis}}(\text{sifClan}, \text{sifSprava}, \text{datDolazak}, \text{minPoDanu}) \llbracket \text{sifClan}, \text{sifSprava}, \text{datDolazak} \mathrel{G} \text{SUM}(\text{minuta}) (\text{vjezba} \bowtie \text{dolazak}) \rrbracket$

- b) Ispisati šifre svih sprava na kojima se vježbalo 12. travnja, ali nije 13. travnja 2012. godine.

**(2 boda)**

$\pi_{\text{sifSprava}} \left[ \sigma_{\text{datDolazak} = 12.4.2012} (\text{vjezba} \bowtie \text{dolazak}) \right] \setminus \pi_{\text{sifSprava}} \left[ \sigma_{\text{datDolazak} = 13.4.2012} (\text{vjezba} \bowtie \text{dolazak}) \right]$

3. Uz pretpostavku da slika 1 prikazuje sve podatke pohranjene u bazi podataka, u obliku tablice **prikažite rezultate** obavljanja sljedećih operacije:

- a)  $\pi_{\text{prezime}} (\sigma_{\text{naziv}='Orbitrek'} (\text{clan} \triangleright \triangleleft \text{dolazak} \triangleright \triangleleft \text{vjezba} \triangleright \triangleleft \text{sprava}))$   
 $\cup \pi_{\text{prezime}} (\sigma_{\text{naziv}='Abtoner'} (\text{clan} \triangleright \triangleleft \text{dolazak} \triangleright \triangleleft \text{vjezba} \triangleright \triangleleft \text{sprava}))$  (1 bod)

prezime
Horvat

- b)  $\pi_{\text{sifClan}, \text{sifDolazak}} (\sigma_{\text{datIstek} > '01.04.2012'} (\text{clan} * \triangleright \triangleleft \text{dolazak}))$  (1 bod)

sifClan	sifDolazak
1111	55
1111	56
1113	58
1114	NULL

4. U bazi podataka evidentiraju se podaci o servisu sprava u jednoj teretani. Teretana ima nekoliko dvorana i određena sprava se može nalaziti samo u jednoj dvorani. Određeni servis jedne sprave obavlja jedan serviser. Ista sprava se u jednom danu može servisirati samo jednom. Svaki serviser radi u određenoj tvrtki. Moguće je da dva serviseri iz različitih tvrtki imaju istu šifru.

Relacijska shema SERVIS sastoji se od sljedećih atributa:

- sifSprava - jedinstvena šifra sprave
- nazSprava - naziv sprave
- sifDvorana - šifra dvorane
- kvadrDvorana - kvadratura dvorane
- datServis - datum obavljenog servisa
- sifServiser - šifra serviseri
- prezServiser - prezime serviseri
- sifTvrta - šifra tvrtke
- nazTvrta - naziv tvrtke

Odredite ključ relacijske sheme SERVIS tako da ona bude u 1NF, a zatim postupno normalizirajte relacijsku shemu na 2NF i 3NF.

1NF:  $K=\{\text{sifSprava}, \text{datServis}\}$

2NF:  $\text{SPRAVA}=\{\text{sifSprava}, \text{nazSprava}, \text{sifDvorana}, \text{kvadrDvorana}\}$   
 $\text{SERVIS}=\{\text{sifSprava}, \text{datServis}, \text{sifServiser}, \text{prezServiser}, \text{sifTvrta}, \text{nazTvrta}\}$

3NF:  $\text{SPRAVA2}=\{\text{sifSprava}, \text{nazSprava}, \text{sifDvorana}\}$   
 $\text{DVORANA}=\{\text{sifDvorana}, \text{kvadrDvorana}\}$   
 $\text{SERVISER}=\{\text{sifServiser}, \text{sifTvrta}, \text{prezServiser}\}$   
 $\text{TVRKA}=\{\text{sifTvrta}, \text{nazTvrta}\}$   
 $\text{SERVIS2}=\{\text{sifSprava}, \text{datServis}, \text{sifServiser}, \text{sifTvrta}\}$

5. Koji je maksimalan broj n-torki koje može imati relacija nad kojom je izgrađeno B<sup>+</sup>-stablo reda 100 dubine 4? Objasniti rješenje. **(2 boda)**

Max. popunjeno stablo reda 100 i dubine 4 ima  $100 \cdot 100 \cdot 100 \cdot 100 \cdot 99 = 99 \cdot 10^8$  kazaljki. To znači da relacija može imati najviše 9.9 milijardi n-torki.

6. Napisati SQL naredbe koje će kreirati najmanji mogući broj indeksa za učinkovito obavljanje (pomoću B<sup>+</sup> stabla) svih dolje navedenih upita. **(3 boda)**

- 1) SELECT \* FROM clan WHERE datlstek = TODAY AND prezime = 'Horvat';
- 2) ELECT \* FROM clan WHERE sifclan < 1200;
- 3) SELECT \* FROM clan WHERE datlstek BETWEEN '01.04.2012.' AND '01.05.2012.';
- 4) SELECT \* FROM clan ORDER BY datlstek ASC, prezime DESC;
- 5) SELECT \* FROM clan ORDER BY prezime, ime;
- 6) SELECT \* FROM clan ORDER BY sifClan, prezime;

CREATE INDEX i1 ON clan(datlstek, prezime DESC)	1, 3, 4
CREATE INDEX i2 ON clan(sifClan, prezime)	2, 6
CREATE INDEX i3 ON clan (prezime, ime)	5

7. Napisati SQL naredbu koja će kreirati relacijsku shemu  $BON = \{ \text{serijskiBroj}, \text{sifClan}, \text{datlzd}, \text{datlskor}, \text{popustKn} \}$ . Shema BON opisuje promotivne bonove za popust na obnovu članarine u teretani. Kod kreiranja relacijske sheme smisleno odabrati tipove podataka te osigurati sljedeće: **(3 boda)**

- serijski broj jedinstveno identificira bon
- podaci o bonu se evidentiraju kod izdavanja, prije nego je poznato koji član će ga iskoristiti i kada
- korištenje bona evidentirati će se unosom šifre postojećeg člana (iz relacije CLAN sa Slike 1.) te datuma iskorištavanja
- datum izdavanja i popust u kunama moraju biti poznati
- popust u kunama može poprimiti samo vrijednosti 50 ili 100
- bon se mora iskoristiti unutar 90 dana nakon izdavanja
- ukoliko se član koji je iskoristio bon izbriše, informaciju o bonu zadržati a šifru člana postaviti na NULL

```
CREATE TABLE BON (  
    serijskiBroj INTEGER PRIMARY KEY,  
    sifClan INTEGER REFERENCES clan(sifClan) ON DELETE SET NULL,  
    datlzd DATE NOT NULL,  
    datlskor DATE,  
    popustKn INTEGER NOT NULL  
        CHECK (popustKn IN (50,100)),  
    CHECK (datlskor-datlzd<=90)  
);
```

## Međuispit iz Baza podataka

26. travnja 2013.

Zadaci 1 - 8 odnose se na bazu podataka INTERNET\_TV prikazanu na **slici 1**. Na slici **nisu** prikazane sve n-torke sadržane u relacijama.

korisnik					serija			
sifKorisnik	prezime	ime	datRod	rejting	sifSerija	nazSerija	godPocEmit	prepDob
11	Perić	Pero	18.10.1978	3	101	Game of Thrones	2011	15
12	Jurić	Đurđa	11.06.1974	2	102	Big Bang Theory	2007	12
13	Jurić	Ivo	05.02.2001	4	103	Sulejman	2011	0
14	Ban	Josip	17.12.1953	3	104	The Wire	2002	18
15	Horvat	Ante	01.04.1973	2				

epizoda					gledanje			
sifEpizoda	rbrSezona	rbrEpizoda	sifSerija	trajanjeMin	sifEpizoda	sifKorisnik	datGled	ocjena
601	1	1	101	52	602	15	13.04.2013	NULL
602	1	2	101	51	604	12	29.03.2013	4
603	2	1	101	49	604	12	09.04.2013	5
604	2	4	102	19	604	15	09.04.2013	NULL
605	3	4	102	21	603	14	06.04.2013	3

Slika 1.

U relacije sa slike 1 pohranjuju se podaci o korisnicima internetske televizije (relacija **korisnik**) te o njihovom gledanju i ocjenjivanju (relacija **gledanje**) pojedinih epizoda (relacija **epizoda**) serija (relacija **serija**). Za svaku se seriju, osim šifre i naziva, evidentira godina početka emitiranja i preporučena najmanja dob gledatelja serije. Korisniku koji često gleda serije raste upisani rejting. Korisnik pojedinu epizodu serije može pogledati više puta, ali samo jednom u istom danu, a nakon svakog gledanja može je ocijeniti ocjenom od 1 do 5. Vrijednost NULL atributa *ocjena* znači da je korisnik odbio dati ocjenu.

U zadacima 1-4 napisati po jednu SQL naredbu kojom će se obaviti sljedeće:

1. Ispisati šifru i naziv serije te datum prvog (najranijeg) gledanja za sve serije čija je godina početka emitiranja 2011., a prosječna ocjena gledatelja je minimalno 3.0. Serije ispisati poredane od one s najkasnijim datumom prvog gledanja, prema onoj s najranijim; u slučaju kad je taj datum istovjetan navesti abecedno po nazivu. **Zadatak riješiti bez podupita.** (2 boda)
2. Za sve serije čija je preporučena dob 18 ispisati šifru, naziv i broj različitih gledatelja rođenih 1990. ili ranije koji su pogledali barem jednu epizodu te serije (ispis i u slučaju kad seriju nije pogledao nitko iz opisane dobne skupine!). **Zadatak riješiti bez podupita.** (2.5 boda)
3. Ispisati šifru i naziv svake serije čija je ovogodišnja minutaža gledanja vikendom veća od minutaže gledanja tijekom svih pet radnih dana. (2.5 boda)
4. Smanjiti za 1 rejting korisnika koji su u tekućoj godini pogledali manje od 3 različite serije ili im je ukupna minutaža gledanih serija u tom vremenu bila manja od 100 minuta. (2.5 boda)
5. Napisati **izraz relacijske algebre (ne SQL upit)** koji odgovara sljedećem: za korisnike koji su ukupno odgledali više od 200 minuta programa ispisati njihovu šifru i spomenutu minutažu odgledanog programa. (2 boda)

U zadacima 6 i 7, uz pretpostavku da slika 1 prikazuje sve podatke pohranjene u bazi podataka, u **obliku tablice prikazati rezultate** obavljanja sljedećih operacija:

6.  $[\pi_{\text{prezime, ime}} (\sigma_{\text{godPocEmit}=2011} (\text{serija} \bowtie \text{epizoda} \bowtie \text{gledanje} \bowtie \text{korisnik}))]$   
 $\setminus [\pi_{\text{prezime, ime}} (\sigma_{\text{godPocEmit}=2007} (\text{serija} \bowtie \text{epizoda} \bowtie \text{gledanje} \bowtie \text{korisnik}))]$  (1 bod)
7.  $\pi_{\text{sifKorisnik, ocjena}} (\sigma_{\text{rejting} \geq 3} (\text{korisnik} * \bowtie \text{gledanje}))$  (1 bod)

8. Napisati SQL naredbe koje će kreirati najmanji mogući broj indeksa za učinkovito obavljanje (pomoću B<sup>+</sup> stabla) svih dolje navedenih upita. (2 boda)

- 1) SELECT \* FROM korisnik WHERE datRod BETWEEN '01.01.1980.' AND '01.01.1990.';
- 2) SELECT \* FROM korisnik WHERE rejting < 4;
- 3) SELECT \* FROM korisnik WHERE prezime LIKE 'H%' AND ime LIKE 'H%';
- 4) SELECT \* FROM korisnik ORDER BY datRod ASC, prezime DESC;
- 5) SELECT \* FROM korisnik ORDER BY ime, prezime;
- 6) SELECT \* FROM korisnik ORDER BY sifKorisnik, rejting;

9. Koji uvjet mora biti zadovoljen da bi relacije r(R) i s(S) bile unijski kompatibilne? Koje se operacije relacijske algebre mogu obaviti s operandima r(R) i s(S) onda i samo onda kada su relacije r(R) i s(S) unijski kompatibilne? (2 boda)

10. Produkcijaska kuća pohranjuje podatke o TV kvizovima koje snima. Relacijska shema EPIZODA\_KVIZA sadrži sljedeće attribute:

- sifKviz - jedinstvena šifra kviza,
- nazKviz - naziv kviza,
- sifStudio - jedinstvena šifra TV studija,
- velStudio - veličina studija u m<sup>2</sup>,
- sifVod - jedinstvena šifra voditelja,
- prezVod - prezime voditelja,
- imeVod - ime voditelja,
- rbrSez - redni broj sezone kviza,
- budzetSez - budžet pojedine sezone kviza,
- rbrEpiz - redni broj epizode kviza unutar sezone (počinje od 1 za svaku novu sezonu),
- datSnimEpiz - datum snimanja epizode kviza.

Vrijede sljedeća pravila:

- kviz se uvijek snima u istom TV studiju,
- tijekom sezone kviza ne mijenja se voditelj,
- istog se dana može snimiti više epizoda pojedinog kviza, a svaka se epizoda snimi u jednom danu.

Odrediti ključ relacijske sheme EPIZODA\_KVIZA tako da ona bude u 1NF, a zatim postupno normalizirati relacijsku shemu na 2NF i 3NF. (4 boda)

11. Relacija KORISNIK (*sifKorisnik*, *prez*, *ime*) sadrži n-torke sa sljedećim vrijednostima atributa *sifKorisnik*: 11, 17, 20, 25, 40, 42, 47. Nacrtati B<sup>+</sup>-stablo reda 5 za atribut *sifKorisnik* tako da popunjenost stabla bude **minimalna**. (2,5 boda)

12. Napisati SQL naredbu koja će kreirati relacije **zabEmisija** i **producent** nad shemama ZABEMISIJA= {*sifZabEm*, *nazZabEm*, *sifProducent*} i PRODUCENT={*sifProducent*, *OIB*, *prezime*, *ime*, *datRod*, *placa*}. Relacija **zabEmisija** zapisuje opće podatke o zabavnim emisijama, a relacija **producent** detalje o producentima zabavnih emisija. Kod kreiranja relacijske sheme smisleno odabrati tipove podataka te osigurati sljedeće: (4 boda)

- zabavna emisija je jedinstveno identificirana šifrom zabavne emisije (*sifZabEm*);
- svaka zabavna emisija mora imati drugačiji naziv;
- producent je jedinstveno identificiran šifrom producenta;
- producent je jedinstveno identificiran svojim OIB-om;
- za svaki zapis relacije **producent** vrijednosti atributa *sifProducent*, *OIB*, *prezime*, *ime* i *datRod* moraju biti poznate, dok vrijednost atributa *placa* ne mora biti poznata;
- za svaki zapis relacije **zabEmisija** moraju biti poznate vrijednosti atributa *sifZabEm* i *nazZabEm*;
- plaća producenta ne smije biti veća od 12000 kuna;
- atribut *sifProducent* u relaciji **zabEmisija** smije poprimiti isključivo vrijednosti istoimenog atributa u relaciji **producent**; ako se izbriše zapis iz relacije **producent**, brišu se i zapisi o zabavnim emisijama koje je on producirao.

13. Zadana je relacijska shema R = { A, B, C, D, E, F } i skup funkcijskih zavisnosti F = { AB → C, A → DE, C → F, E → D }. Ispitati vrijedi li funkcijska zavisnost ABC → DEF. Za svaki korak dokaza napisati pravilo koje se koristi. (2 boda)



## Rješenja:

U zadacima 1-8 napisati po jednu SQL naredbu kojom će se obaviti sljedeće:

1. Ispisati šifru i naziv serije te datum prvog (najranijeg) gledanja za sve serije čija je godina početka emitiranja 2011., a prosječna ocjena gledatelja je minimalno 3.0. Serije ispisati poredane od one s najkasnijim datumom prvog gledanja, prema onoj s najranijim; u slučaju kad je taj datum istovjetan navesti abecedno po nazivu. **Zadatak riješiti bez podupita. (2 boda)**

```
SELECT serija.sifSerija, nazSerija, MIN(datGled) najranijeGledanje
FROM serija JOIN epizoda ON serija.sifserija=epizoda.sifserija
      JOIN gledanje ON epizoda.sifepizoda=gledanje.sifepizoda
WHERE godPocEmit=2011
GROUP BY serija.sifSerija, nazSerija
HAVING AVG(ocjena)>=3.0
ORDER BY najranijeGledanje DESC, nazSerija;
```

2. Za sve serije čija je preporučena dob 18 ispisati šifru, naziv i broj različitih gledatelja rođenih 1990. ili ranije koji su pogledali barem jednu epizodu te serije (ispis i u slučaju kad seriju nije pogledao nitko iz opisane dobne skupine!). **Zadatak riješiti bez podupita. (2.5 boda)**

```
SELECT serija.sifserija, nazSerija, COUNT(DISTINCT korisnik.sifkorisnik)
FROM gledanje JOIN epizoda ON gledanje.sifepizoda=epizoda.sifepizoda
      JOIN korisnik ON gledanje.sifkorisnik=korisnik.sifkorisnik
      AND datrod<='31.12.1990'
      RIGHT JOIN serija ON epizoda.sifserija=serija.sifserija
WHERE prepDob=18
GROUP BY serija.sifserija, nazSerija;
```

```
SELECT serija.sifserija, nazSerija, COUNT(DISTINCT korisnik.sifkorisnik)
FROM serija LEFT JOIN epizoda ON serija.sifserija=epizoda.sifserija
      LEFT JOIN gledanje ON gledanje.sifepizoda=epizoda.sifepizoda
      LEFT JOIN korisnik ON gledanje.sifkorisnik=korisnik.sifkorisnik
      AND datrod<='31.12.1990'
WHERE prepDob=18
GROUP BY serija.sifserija, nazSerija;
```

3. Ispisati šifru i naziv svake serije čija je ovogodišnja minutaža gledanja vikendom veća od minutaže gledanja tijekom svih pet radnih dana. **(2.5 boda)**

```
SELECT serija.sifserija, nazSerija
FROM serija
WHERE
  (SELECT SUM(trajanjemin)
   FROM epizoda JOIN gledanje ON epizoda.sifepizoda=gledanje.sifepizoda
   WHERE WEEKDAY(datgled) IN (0, 6)
   AND YEAR(datgled)=YEAR(TODAY)
   AND epizoda.sifserija=serija.sifserija)
>
  (SELECT SUM(trajanjemin)
   FROM epizoda JOIN gledanje ON epizoda.sifepizoda=gledanje.sifepizoda
   WHERE WEEKDAY(datgled) BETWEEN 1 AND 5)
  AND YEAR(datgled)=YEAR(TODAY)
  AND epizoda.sifserija=serija.sifserija);

-- može i ... WHERE WEEKDAY(datgled) IN (1, 2, 3, 4, 5))
```

4. Smanjiti za 1 rejting korisnika koji su u tekućoj godini pogledali manje od 3 različite serije ili im je ukupna minutaža gledanih serija u tom vremenu bila manja od 100 minuta. **(2.5 boda)**

Napomena: uvjetima su obuhvaćeni korisnici koji u tekućoj godini nisu pogledali ništa (0 serija, 0 minuta) tj. nema zapisa u relaciji gledanje s njihovom korisničkom šifrom i datumom u tekućoj godini.

```
UPDATE korisnik
SET rejting=rejting-1
WHERE
(
    SELECT COUNT(DISTINCT sifserija)
    FROM gledanje JOIN epizoda ON gledanje.sifepizoda=epizoda.sifepizoda
    WHERE YEAR(datgled)=YEAR(TODAY)
    AND gledanje.sifkorisnik=korisnik.sifkorisnik
)<3
OR (
    SELECT SUM(trajanjemin)
    FROM gledanje JOIN epizoda ON gledanje.sifepizoda=epizoda.sifepizoda
    WHERE YEAR(datgled)=YEAR(TODAY)
    AND gledanje.sifkorisnik=korisnik.sifkorisnik
)<100;
```

```
UPDATE korisnik
SET rejting=rejting-1
WHERE sifkorisnik NOT IN (
SELECT sifkorisnik
    FROM gledanje JOIN epizoda ON gledanje.sifepizoda=epizoda.sifepizoda
    WHERE YEAR(datgled)=YEAR(TODAY)
GROUP BY sifKorisnik
HAVING COUNT(DISTINCT sifserija)>=3 AND SUM(trajanjemin)>=100
);
```

```
UPDATE korisnik
SET rejting=rejting-1
WHERE NOT EXISTS (
    SELECT sifKorisnik
    FROM gledanje JOIN epizoda ON gledanje.sifepizoda=epizoda.sifepizoda
    WHERE YEAR(datgled)=YEAR(TODAY)
    AND gledanje.sifkorisnik=korisnik.sifkorisnik
GROUP BY sifKorisnik
HAVING COUNT(DISTINCT sifserija)>=3 AND SUM(trajanjemin)>=100
);
```

```
-- iduća dva upita smanjili bi rejting za korisnike koji jesu pogledali bar
-- jednu epizodu neke serije, a pritom je zadovoljen uvjet: manje od tri
-- različite serije ili manje od 100 minuta; rejting se ne bi smanjio
-- korisnicima koji u tekućoj godini nisu pogledali ništa
```

```
UPDATE korisnik
SET rejting=rejting-1
WHERE sifkorisnik IN(
    SELECT sifkorisnik
    FROM gledanje JOIN epizoda ON gledanje.sifepizoda=epizoda.sifepizoda
    WHERE YEAR(datgled)=YEAR(TODAY)
GROUP BY sifKorisnik
HAVING COUNT(DISTINCT sifserija)<3 OR SUM(trajanjemin)<100
);
```

```

UPDATE korisnik
SET rejting=rejting-1
WHERE EXISTS (
    SELECT sifKorisnik
    FROM gledanje JOIN epizoda ON gledanje.sifepizoda=epizoda.sifepizoda
    WHERE YEAR(datgled)=YEAR(TODAY)
    AND gledanje.sifkorisnik=korisnik.sifkorisnik
    GROUP BY sifKorisnik
    HAVING COUNT(DISTINCT sifserija)<3 OR SUM(trajanjemin)<100
);

```

5. Napisati **izraz relacijske algebre** (ne SQL upit) koji odgovara sljedećem: za korisnike koji su ukupno odgledali više od 200 minuta programa ispisati njihovu šifru i spomenutu minutažu odgledanog programa. **(2 boda)**

$\sigma_{\text{ukupnoTrajanje} \geq 200} [\rho_{\text{ispis(sifKorisnik, ukupnoTrajanje)}}[\sigma_{\text{SUM(trajanjeMin)((gledanje \triangleright \triangleleft \text{epizoda})})}] ]$

U zadacima 6 i 7, uz pretpostavku da slika 1 prikazuje sve podatke pohranjene u bazi podataka, u **obliku tablice prikazati rezultate** obavljanja sljedećih operacija:

6.  $[\pi_{\text{prezime, ime}}(\sigma_{\text{godPocEmit}=2011}(\text{serija} \triangleright \triangleleft \text{epizoda} \triangleright \triangleleft \text{gledanje} \triangleright \triangleleft \text{korisnik}))] \setminus [\pi_{\text{prezime, ime}}(\sigma_{\text{godPocEmit}=2007}(\text{serija} \triangleright \triangleleft \text{epizoda} \triangleright \triangleleft \text{gledanje} \triangleright \triangleleft \text{korisnik}))]$  **(1 bod)**

prezime	ime
Ban	Josip

7.  $\pi_{\text{sifKorisnik, ocjena}}(\sigma_{\text{rejting} \geq 3}(\text{korisnik} * \triangleright \triangleleft \text{gledanje}))$  **(1 bod)**

sifKorisnik	ocjena
11	NULL
13	NULL
14	3

8. Napisati SQL naredbe koje će kreirati najmanji mogući broj indeksa za učinkovito obavljanje (pomoću B<sup>+</sup> stabla) svih dolje navedenih upita. **(2 boda)**

- 1) SELECT \* FROM korisnik WHERE datRod BETWEEN '01.01.1980.' AND '01.01.1990.';
- 2) SELECT \* FROM korisnik WHERE rejting < 4;
- 3) SELECT \* FROM korisnik WHERE prezime LIKE 'H%' AND ime LIKE 'H%';
- 4) SELECT \* FROM korisnik ORDER BY datRod ASC, prezime DESC;
- 5) SELECT \* FROM korisnik ORDER BY ime, prezime;
- 6) SELECT \* FROM korisnik ORDER BY sifKorisnik, rejting;

```

CREATE INDEX i1 ON korisnik (datRod, prezime DESC)      1 i 4
CREATE INDEX i2 ON korisnik (ime, prezime)              3 i 5
CREATE INDEX i3 ON korisnik (rejting)                   2
CREATE INDEX i4 ON korisnik (sifKorisnik, rejting)      6

```

9. Koji uvjet mora biti zadovoljen da bi relacije  $r(R)$  i  $s(S)$  bile unijski kompatibilne? **(1 bod)**

Relacije  $r(R)$  i  $s(S)$  su unijski kompatibilne ako vrijedi:

- relacije su istog stupnja
- korespondentni atributi su definirani nad istim domenama

Koje se operacije relacijske algebre mogu obaviti s operandima  $r(R)$  i  $s(S)$  onda i samo onda kada su relacije  $r(R)$  i  $s(S)$  unijski kompatibilne? **(1 bod)**

Unija, razlika, presjek.

10. Produkcijaska kuća pohranjuje podatke o TV kvizovima koje snima. Relacijska shema EPIZODA\_KVIZA sadrži sljedeće attribute:

- sifKviz - jedinstvena šifra kviza,
- nazKviz - naziv kviza,
- sifStudio - jedinstvena šifra TV studija,
- velStudio - veličina studija u  $m^2$ ,
- sifVod - jedinstvena šifra voditelja,
- prezVod - prezime voditelja,
- imeVod - ime voditelja,
- rbrSez - redni broj sezone kviza,
- budzetSez - budžet pojedine sezone kviza,
- rbrEpiz - redni broj epizode kviza unutar sezone (počinje od 1 za svaku novu sezonu),
- datSnimEpiz - datum snimanja epizode kviza.

Vrijede sljedeća pravila:

- kviz se uvijek snima u istom TV studiju,
- tijekom sezone kviza ne mijenja se voditelj,
- istog se dana može snimiti više epizoda pojedinog kviza, a svaka se epizoda snimi u jednom danu.

Odrediti ključ relacijske sheme EPIZODA\_KVIZA tako da ona bude u 1NF, a zatim postupno normalizirati relacijsku shemu na 2NF i 3NF. **(4 boda)**

1NF

$K=\{sifKviz, rbrSez, rbrEpiz\}$

2NF

KVIZ (sifKviz, nazKviz, sifStudio, velStudio)  $K=\{sifKviz\}$

SEZONA (sifKviz, brSez, budzetSez sifVod, prezVod, imeVod)  $K=\{sifKviz, brSez\}$

EPIZODA (sifKviz, rbrSez, rbrEpiz, datSnimEpiz)  $K=\{sifKviz, rbrSez, rbrEpiz\}$

3NF

KVIZ2 (sifKviz, nazKviz, sifStudio)  $K=\{sifKviz\}$

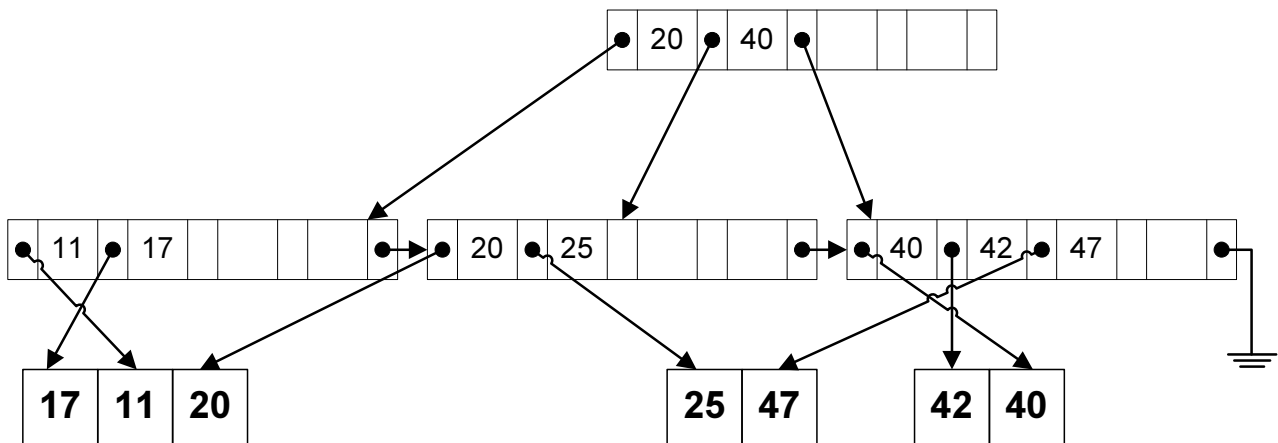
STUDIO (sifStudio, velStudio)  $K=\{sifStudio\}$

SEZONA2 (sifKviz, brSez, budzetSez sifVod)  $K=\{sifKviz, brSez\}$

VODITELJ (sifVod, prezVod, imeVod)  $K=\{sifVod\}$

EPIZODA (sifKviz, rbrSez, rbrEpiz, datSnimEpiz)  $K=\{sifKviz, rbrSez, rbrEpiz\}$

11. Relacija KORISNIK (*sifKorisnik*, *prez*, *ime*) sadrži n-torke sa sljedećim vrijednostima atributa *sifKorisnik*: 11, 17, 20, 25, 40, 42, 47. Nacrtati B<sup>+</sup>-stablo reda 5 za atribut *sifKorisnik* tako da popunjenost stabla bude minimalna. (2,5 boda)



12. Napisati SQL naredbu koja će kreirati relacije **zabEmisija** i **producent** nad shemama ZABEMISIJA= {*sifZabEm*, *nazZabEm*, *sifProducent*} i PRODUCENT={*sifProducent*, *OIB*, *prezime*, *ime*, *datRod*, *placa*}. Relacija **zabEmisija** zapisuje opće podatke o zabavnim emisijama, a relacija **producent** detalje o producentima zabavnih emisija. Kod kreiranja relacijske sheme smisleno odabrati tipove podataka te osigurati sljedeće: (4 boda)

- zabavna emisija je jedinstveno identificirana šifrom zabavne emisije (*sifZabEm*);
- svaka zabavna emisija mora imati drugačiji naziv;
- producent je jedinstveno identificiran šifrom producenta;
- producent je jedinstveno identificiran svojim OIB-om;
- za svaki zapis relacije **producent** vrijednosti atributa *sifProducent*, *OIB*, *prezime*, *ime* i *datRod* moraju biti poznate, dok vrijednost atributa *placa* ne mora biti poznata;
- za svaki zapis relacije **zabEmisija** moraju biti poznate vrijednosti atributa *sifZabEm* i *nazZabEm*;
- plaća producenta ne smije biti veća od 12000 kuna;
- atribut *sifProducent* u relaciji **zabEmisija** smije poprimiti isključivo vrijednosti istoimenog atributa u relaciji **producent**; ako se izbriše zapis iz relacije **producent**, brišu se i zapisi o zabavnim emisijama koje je on producirao.

```
CREATE TABLE producent (
    sifProducent INTEGER PRIMARY KEY,
    OIB DECIMAL(11) NOT NULL UNIQUE,
    prezime NCHAR(40) NOT NULL,
    ime NCHAR(40) NOT NULL,
    datRod DATE NOT NULL,
    placa DECIMAL(7,2) CHECK (placa <=12000)
);
```

```
CREATE TABLE zabEmisija (
    sifZabEm INTEGER PRIMARY KEY,
    nazZabEm NCHAR(50) NOT NULL UNIQUE,
    sifProducent INTEGER REFERENCES producent(sifProducent) ON DELETE CASCADE
);
```

13. Zadana je relacijska shema  $R = \{ A, B, C, D, E, F \}$  i skup funkcijskih zavisnosti  $F = \{ AB \rightarrow C, A \rightarrow DE, C \rightarrow F, E \rightarrow D \}$ . Ispitati vrijedi li funkcijska zavisnost  $ABC \rightarrow DEF$ . Za svaki korak dokaza napisati pravilo koje se koristi. **(2 boda)**

1.  $ABC \rightarrow ABC$  (refleksivnost):
2.  $ABC \rightarrow ABC \wedge A \rightarrow DE \Rightarrow ABC \rightarrow ABCDE$  (akumulacija)  
 $ABC \rightarrow ABCDE \wedge C \rightarrow F \Rightarrow ABC \rightarrow ABCDEF$  (akumulacija)
3.  $ABC \rightarrow ABCDEF \Rightarrow ABC \rightarrow DEF$  (dekompozicija):

Može i obrnutim redom kod akumulacije

1.  $ABC \rightarrow ABC$  (refleksivnost):
2.  $ABC \rightarrow ABC \wedge C \rightarrow F \Rightarrow ABC \rightarrow ABCF$  (akumulacija)  
 $ABC \rightarrow ABCF \wedge A \rightarrow DE \Rightarrow ABC \rightarrow ABCDEF$  (akumulacija)
3.  $ABC \rightarrow ABCDEF \Rightarrow ABC \rightarrow DEF$  (dekompozicija)

# 1. međuispit iz Baza podataka

2. travnja 2010.

Zadaci 1 i 2 odnose se na relacije opisane na slici 1. Na slici nisu prikazane sve n-torke koje su sadržane u relacijama.

Slika 1.

osoba				osobaPrijatelj	
sifOsoba	imeOsoba	prezOsoba	datRodj	sifOsoba	sifPrijatelj
1	Ivan	Car	11.05.1976	1	2
2	Ana	Kralj	12.06.1977	2	3
3	Marko	Beg	13.07.1978	3	2
4	Ela	Ban	14.08.1979	2	4
5	Ivan	Knez	15.09.1980	4	2
				3	5

vjestina			osobaVjestina		
sifVjestina	nazVjestina		sifOsoba	sifVjestina	razina
1	Engleski jezik		1	1	4
2	SQL		2	1	5
3	Java		2	3	3
			4	1	4
			4	2	5
			5	1	5

Relacija **osoba** sadrži podatke o osobama, a relacijom **osobaPrijatelj** se definiraju veze među osobama. Takve veze nisu simetrične, odnosno jedna osoba može imati više definiranih prijatelja, a osobe koje je definirala kao svoje prijatelje ne moraju tu istu osobu smatrati svojim prijateljem. Osoba ne mora imati definiranih prijatelja. Relacija **vjestina** sadrži podatke o vještinama. Relacija **osobaVjestina** sadrži podatke o razini vještini (atribut razina s vrijednostima u intervalu [1, 5]) kojom neka osoba vlada. Osoba ne mora imati definiranih vještina.

1. Napisati po jednu SQL naredbu kojom će se obaviti sljedeće:

- Ispisati šifru, prezime, ime i datum rođenja osoba koji su prijatelji osobe sa šifrom 3.  
**Zadatak riješiti bez podupita. (1 bod)**
- Ispisati šifru, prezime, ime i datum rođenja osoba koje tekuće godine navršavaju 30 godina i barem jednom vještinom vladaju s razinom 5. U ispisu se ista osoba pojavljuje samo jednom.  
**Zadatak riješiti bez podupita. (1.5 bod)**
- Za osobe koje vladaju vještinama koje u nazivu sadrže riječ 'jezik' ispisati inicijale (npr. za osobu Ivan Car ispisati I.C.), naziv vještine i opis koliko dobro osoba vlada tom vještinom:
  - za razinu 1 i 2 ispisati „slabo“
  - za razinu 3 „srednje“
  - za razine 4 i 5 ispisati „dobro“.**Zadatak riješiti bez podupita. (1.5 bod)**
- Za svaku vještinu ispisati šifru i naziv vještine, broj osoba koje vladaju tom vještinom i prosječnu razinu. Potrebno je ispisati i vještine kojima nitko ne vlada. Rezultate poredati silazno po broju osoba, a unutar toga silazno po prosječnoj razini. **Zadatak riješiti bez podupita. (2 boda)**
- Ispisati šifru, ime i prezime onih osoba koje imaju razinu vještine „SQL“ najveću od svih osoba koje vladaju tom vještinom **(1.5 bod)**
- Za osobu i svakog njenog prijatelja ispisati šifru i prezime osobe te šifru i prezime prijatelja. **Zadatak riješiti bez podupita. (1 bod)**

2. Napisati po jedan **izraz relacijske algebre (ne SQL upit)** koji odgovara sljedećem:

- Ispisati šifru i prezime osobe te broj vještina kojima osoba vlada. Osobe koje ne vladaju niti jednom vještinom nije potrebno ispisivati. **(1 bod)**
- Ispisati šifru, ime, prezime i datum rođenja osoba koji su **zajednički** prijatelji osobama sa šifrom 1 i 3. **(1.5 bod)**

Zadaci 3 i 4 se odnose na relacije opisane na slici 2.

**Slika 2.**

osoba		
sifOsoba	imeOsoba	prezOsoba
1	Ivan	Car
2	Ana	Kralj
3	Marko	Beg
4	Ivan	Župan

vjestina		
sifVjestina	nazVjestina	sifTipVjestina
1	Engleski	1
2	SQL	NULL
3	Java	3
4	Skijanje	NULL

osobaVjestina		
sifOsoba	sifVjestina	razina
1	1	4
1	4	3
3	1	NULL
4	3	5
3	3	4

tipVjestina	
sifTipVjestina	nazTipVjestina
1	Strani jezik
2	Sport
3	Računarstvo

3. Prikažite **rezultate obavljanja** sljedećih operacija (rezultat prikazati u obliku tablice):

a)  $\sigma_{\text{razina} > 3 \vee \text{nazTipVjestina} \neq \text{'Računarstvo'}}(\text{vjestina} \bowtie \text{tipVjestina} \bowtie \text{osobaVjestina})$  (1 bod)

b)  $(\pi_{\text{sifOsoba}}(\sigma_{\text{sifVjestina}=1}(\text{osobaVjestina}))) \cup \pi_{\text{sifOsoba}}(\sigma_{\text{imeOsoba}=\text{'Ivan'}}(\text{osoba})) \bowtie \text{osoba}$  (1 bod)

4. Napišite **SQL upit** kojim će se obaviti sljedeća operacija:

$\sigma_{\text{prezOsoba}=\text{'Župan'}}(\text{osoba} * \bowtie \text{osobaVjestina})$  (1.5 bod)

5.

a) Koje binarne operacije od navedenih:

- unija
- presjek
- razlika
- dijeljenje
- prirodno spajanje

je moguće provesti između relacija *posjet* i *mjesto*? Objasniti odgovor. (1 bod)

b) Koji je stupanj i kardinalnost relacije koja nastane kao rezultat obavljanja sljedeće operacije:

$\text{mjesto} \times \rho_{\text{posjet}(\text{sifOsoba}, \text{pbrPosjet})}(\text{posjet})$  (0.5 boda)

posjet	
sifOsoba	pbr
1	20000
2	10000
3	10000
1	23000
1	21000
2	21000
2	47000

mjesto	
pbr	
	20000
	21000
	23000



## Rješenje:

1.

- a)
- ```
SELECT osoba.*
FROM osoba
JOIN osobaPrijatelj
ON osoba.sifOsoba = osobaPrijatelj.sifPrijatelj
WHERE osobaPrijatelj.sifOsoba = 3
```
- b)
- ```
SELECT DISTINCT osoba.*
FROM osoba JOIN osobaVjestina
ON osoba.sifOsoba = osobaVjestina.sifOsoba
WHERE YEAR(today) - YEAR(datRodj) = 30
AND razina = 5
```
- c)
- ```
SELECT SUBSTRING(imeOsoba FROM 1 FOR 1) || '.'
|| SUBSTRING (prezOsoba FROM 1 FOR 1) || '.'
, nazVjestina
, CASE WHEN razina < 3 THEN 'slabo'
WHEN razina = 3 THEN 'srednje'
ELSE 'dobro'
END as razina
FROM osoba JOIN osobaVjestina
ON osoba.sifOsoba = osobaVjestina.sifOsoba
JOIN vjestina
ON osobaVjestina.sifVjestina = vjestina.sifVjestina
WHERE nazVjestina LIKE '%jezik%'
```
- d)
- ```
SELECT vjestina.sifVjestina, nazVjestina, COUNT(*) as brOsoba, AVG(razina) as prRazina
FROM vjestina LEFT JOIN osobaVjestina
ON vjestina.sifVjestina = osobaVjestina.sifVjestina
GROUP BY sifVjestina, nazVjestina
ORDER BY brOsoba DESC, prRazina DESC
```
- e)
- ```
SELECT sifOsoba, imeOsoba, prezOsoba
FROM osobaVjestina JOIN vjestina
ON osobaVjestina.sifVjestina = vjestina.sifVjestina
JOIN osoba
ON osobaVjestina.sifOsoba = osoba.sifOsoba
AND vjestina.nazVjestina = 'SQL'
AND osobaVjestina.razina = (SELECT MAX(razina)
FROM osobaVjestina osobaVjestinaMax
WHERE osobaVjestinaMax.sifVjestina =
osobaVjestina.sifVjestina)
```
- f)
- ```
SELECT osoba.sifOsoba, osoba.prezOsoba, p.sifOsoba, p.prezOsoba
FROM osoba p JOIN osobaPrijatelj
ON osobaPrijatelj.sifPrijatelj = p.sifOsoba
JOIN osoba
ON osobaPrijatelj.sifOsoba = osoba.sifOsoba
```

2.

a)

$\rho_{\text{ISPIS}}(\text{sifOsoba}, \text{prezOsoba}, \text{brVjest}) \left( \text{sifOsoba}, \text{prezOsoba} \text{ } G \text{ COUNT}(\text{sifOsoba}) (\text{osoba} \bowtie \text{osobaVjestina}) \right)$

b)

$(\pi_{\text{sifPrijatelj}}(\sigma_{\text{sifOsoba}=1}(\text{osobaPrijatelj}))) \cap \pi_{\text{sifPrijatelj}}(\sigma_{\text{sifOsoba}=3}(\text{osobaPrijatelj})) \bowtie \text{osoba}$   
 $\text{sifPrijatelj} = \text{sifOsoba}$

3.

a)

<i>sifVjestina</i>	<i>nazVjestina</i>	<i>sifTipVjestina</i>	<i>nazTipVjestina</i>	<i>sifOsoba</i>	<i>razina</i>
1	Engleski	1	Strani jezik	1	4
1	Engleski	1	Strani jezik	3	NULL
3	Java	3	Računarstvo	4	5
3	Java	3	Računarstvo	3	4

b)

$(\pi_{\text{sifOsoba}}(\sigma_{\text{sifVjestina}=1}(\text{osobaVjestina}) \cup \pi_{\text{sifOsoba}}(\sigma_{\text{razina}=3}(\text{osobaVjestina})))) \bowtie \text{osoba}$

<i>sifOsoba</i>	<i>imeOsoba</i>	<i>prezOsoba</i>
1	Ivan	Car
3	Marko	Beg
4	Ivan	Župan

4.

```
SELECT osoba.*, sifVjestina, razina
FROM osoba LEFT JOIN osobaVjestina
ON osoba.sifOsoba = osobaVjestina.sifOsoba
WHERE prezOsoba = 'Župan'
```

5.

a) dijeljenje, prirodno spajanje

b) 3, 21

# 1. međuispit iz Baza podataka

1. travnja 2011.

Zadaci 1 do 7 se odnose na relacije opisane na **slici 1**. Na slici **nisu** prikazane sve n-torke koje su sadržane u relacijama.

Slika 1.

osoba				zadatak	
sifOsoba	prezOsoba	imeOsoba	datRodj	sifZadatak	nazZadatak
1	Anić	Ljerka	11.05.1986	1	Konceptualno modeliranje
2	Kraljević	Ante	12.06.1987	2	Logičko modeliranje
3	Begić	Marijana	13.07.1988	3	Fizičko modeliranje
4	Ban	Ela	14.08.1989	4	Programiranje
5	Knez	Ivana	15.09.1989		

zadatakPrethodi		osobaRadi			
sifZadatak	sifZadatakPrethodi	sifOsoba	sifZadatak	datRadi	satiRadi
2	1	1	1	09.01.2011	5
3	1	1	1	15.03.2011	6
3	2	4	3	19.03.2011	3,5
		5	1	19.02.2011	7
		1	2	15.03.2011	2
		4	2	17.03.2011	5

U relacije sa slike 1 se pohranjuju podaci o osobama (relacija **osoba**) koje rade na zadacima (relacija **zadatak**) određenog informatičkog projekta. Broj radnih sati osobe na zadatku projekta određenog radnog dana evidentira se u relaciji **osobaRadi**. Pojedini zadaci se mogu obavljati tek nakon što se obave svi definirani prethodnici (relacija **zadatakPrethodi**). Za jednu osobu i jedan dan se isti zadatak može evidentirati najviše jednom.

Napišite po jednu SQL naredbu kojom će se obaviti sljedeće:

1. Ispisati različite nazive zadataka na kojima je barem jedna osoba nekog dana radila više od 6 sati. **Zadatak riješiti bez podupita.** (1 bod)
2. Za osobe čije ime počinje slovom 'Lj' ili 'Nj' ispisati prezime i inicijal imena (npr. za osobu Anić Ljerka ispisati 'Anić Lj.'), te naziv zadatka i datum rada na zadatku. Uzeti u obzir samo zadatke koji u nazivu sadrže riječ 'model'. **Zadatak riješiti bez podupita.** (1.5 bod)
3. Za zadatke na kojima se subotama i nedjeljama ukupno radilo više od 100 sati ispisati šifru i naziv zadatka, broj različitih osoba koje su **ikada** radile na zadatku te prvi i zadnji datum rada na zadatku. (2.5 bod)
4. Za sve osobe čije prezime ne završava nizom 'ić' ispisati šifru osobe, prezime i ime te jedan od sljedeća dva teksta:
  - „točno 42 sata“ - ako je osoba u tekućem mjesecu ukupno radila točno 42 sata
  - „manje ili više od 42 sata“ - ako je osoba u tekućem mjesecu ukupno radila manje ili više od 42 sata ili **nije radila uopće** (2.5 bod)
5. Za svaki zadatak ispisati šifru i naziv te broj zadataka koji mu neposredno prethode (atribut/stupac imenovati sa brojPrethodnika) i broj zadataka kojima on neposredno prethodi (atribut/stupac imenovati sa brojNasljednika). Za zadatke koji nemaju prethodnika/nasljednika ispisati vrijednost 0 za broj zadataka prethodnika/nasljednika. Rezultate poredati silazno po broju zadataka nasljednika, a unutar toga uzlazno po nazivu zadatka. (2 boda)

Napišite po jedan **izraz relacijske algebre** (ne SQL upit) koji odgovara sljedećem:

6. Ispisati različita prezimena svih osoba koje su bilo kad radile na zadatku broj 1, ali nikada na zadatku broj 2. (1 bod)
7. Ispisati ime i prezime osobe, te ukupan broj sati koje je osoba odradila od 1.1.2011. Osobe koje nisu radile ne ispisivati. (1 bod)

Slika 1.

osoba				zadatak	
sifOsoba	prezOsoba	imeOsoba	datRodj	sifZadatak	nazZadatak
1	Anić	Ljerka	11.05.1986	1	Konceptualno modeliranje
2	Kraljević	Ante	12.06.1987	2	Logičko modeliranje
3	Begić	Marijana	13.07.1988	3	Fizičko modeliranje
4	Ban	Ela	14.08.1989	4	Programiranje
5	Knez	Ivana	15.09.1989		

zadatakPrethodi		osobaRadi			
sifZadatak	sifZadatakPrethodi	sifOsoba	sifZadatak	datRadi	satiRadi
2	1	1	1	09.01.2011	5
3	1	1	1	15.03.2011	6
3	2	4	3	19.03.2011	3,5
		5	1	19.02.2011	7
		1	2	15.03.2011	2
		4	2	17.03.2011	5

U obliku tablice prikažite rezultate obavljanja sljedećih operacija:

8.  $\pi_{\text{sifZadatak, nazZadatak}}(\sigma_{\text{prezOsoba}='Anić' \vee \text{prezOsoba}='Knez'}(\text{osoba} \bowtie \text{osobaRadi} \bowtie \text{zadatak}))$  (1 bod)

9.  $\pi_{\text{nazZadatak}}(\text{zadatak}) \setminus (\pi_{\text{nazZadatak}}(\text{zadatak} \bowtie \text{zadatakPrethodi}))$  (1 bod)

Napišite po jedan SQL upit za svaki od izraza relacijske algebre:

10.  $\pi_{\text{imeOsoba, prezOsoba}}(\text{osoba} \bowtie \sigma_{\text{sifZadatak}=1}(\text{osobaRadi}))$  (0.5 bod)

11.  $\rho_{\text{dnevniProsjeck}}(\text{imeOsoba, prezOsoba, prosjeck})(\text{imeOsoba, prezOsoba} \mathrel{\mathcal{G}} \text{AVG}(\text{satiRadi}))$   
 $(\sigma_{\text{sifZadatak}=1 \vee \text{sifZadatak}=3}(\text{osoba} * \bowtie \text{osobaRadi} * \bowtie \text{zadatak}))$  (1.5 bod)

12. Napišite tablicu istinitosti u trovalentnoj logici za operaciju **AND**. (0.5 boda)

## Rješenja:

### 1. 1 bod

```
SELECT DISTINCT nazZadatak
FROM zadatak
    JOIN osobaRadi
        ON osobaRadi.sifZadatak = zadatak.sifZadatak
WHERE satiradi >=6
```

### 2. 1.5 bod

```
SELECT TRIM(prezOsoba) || ' ' || SUBSTRING(imeOsoba FROM 1 for 2)|| '.'
    , nazZadatak
    , datRadi
FROM osobaRadi
    JOIN osoba
        ON osobaRadi.sifOsoba = osoba.sifOsoba
    JOIN zadatak
        ON osobaRadi.sifzadatak = zadatak.sifzadatak
WHERE nazZadatak LIKE '%model%'
    AND (ImeOsoba LIKE 'Lj%' OR ImeOsoba LIKE 'Nj%' )
```

### 3. 2.5 boda

```
SELECT zadatak.*
    , COUNT(DISTINCT sifOsoba), MIN (datRadi), MAX(datRadi)
FROM zadatak
    JOIN osobaRadi
        ON osobaRadi.sifZadatak = zadatak.sifZadatak
WHERE (SELECT SUM(satiRadi)
        FROM osobaRadi
        WHERE osobaRadi.sifZadatak = zadatak.SifZadatak
        AND WEEKDAY(datRadi) IN (0,6)) > 100
GROUP BY zadatak.sifZadatak, zadatak.nazZadatak
```

#### Alternativno rješenje:

```
SELECT zadatak.*
    , COUNT(DISTINCT sifOsoba), MIN (datRadi), MAX(datRadi)
FROM zadatak
    JOIN osobaRadi
        ON osobaRadi.sifZadatak = zadatak.sifZadatak
GROUP BY zadatak.sifZadatak, zadatak.nazZadatak
HAVING (SELECT SUM(satiRadi)
        FROM osobaRadi
        WHERE osobaRadi.sifZadatak = zadatak.SifZadatak
        AND WEEKDAY(datRadi) IN (0,6)) > 100
```

#### 4. 2.5 boda

```
SELECT osoba.sifOsoba, osoba.prezOsoba, osoba.imeOsoba
, CASE (SELECT SUM(satiRadi)
      FROM osobaRadi
      WHERE osobaRadi.sifOsoba = osoba.sifOsoba
        AND YEAR(datRadi) = YEAR(TODAY)
        AND MONTH(datRadi) = MONTH(TODAY))
      WHEN 42 THEN 'radi točno 42 sata'
      ELSE 'radi manje ili više od 42 sata'
END
FROM osoba
WHERE prezOsoba NOT LIKE '%ić'
```

#### Alternativno rješenje:

```
SELECT osoba.sifOsoba, osoba.prezOsoba, osoba.imeOsoba
, CASE SUM(satiRadi)
      WHEN 42 THEN 'radi točno 42 sata'
      ELSE 'radi manje ili više od 42 sata'
END
FROM osoba
LEFT JOIN osobaRadi
      ON osobaRadi.sifOsoba = osoba.sifOsoba
      AND YEAR(datRadi) = YEAR(TODAY)
      AND MONTH(datRadi) = MONTH(TODAY)
WHERE prezOsoba NOT LIKE '%ić'
GROUP BY osoba.sifOsoba, osoba.prezOsoba, osoba.imeOsoba
```

#### 5. 2 boda

```
SELECT zadatak.*
      , COUNT(njemuPrethode.sifZadatak) brojPrethodnika
      , COUNT(onPrethodi.sifZadatak) brojNasljednika
FROM zadatak
LEFT JOIN zadatakPrethodi njemuPrethode
      ON zadatak.sifZadatak = njemuPrethode.sifZadatak
LEFT JOIN zadatakPrethodi onPrethodi
      ON zadatak.sifZadatak = onPrethodi.sifZadatakPrethodi
GROUP BY zadatak.sifZadatak, zadatak.nazZadatak
ORDER BY brojNasljednika DESC, nazZadatak

ili

SELECT zadatak.*
      , (SELECT COUNT(*) FROM zadatakPrethodi
          WHERE zadatak.sifZadatak = zadatakPrethodi.sifZadatak) brojPrethodnika
      , (SELECT COUNT(*) FROM zadatakPrethodi
          WHERE zadatak.sifZadatak = zadatakPrethodi.sifZadatakPrethodi)
                                          brojNasljednika
FROM zadatak
ORDER BY brojNasljednika DESC, nazZadatak
```

6. 1 bod

$$\pi_{\text{prezOsoba}}(\pi_{\text{sifOsoba, prezOsoba}}(\sigma_{\text{sifZadatak}=1}(\text{osoba} \bowtie \text{osobaRadi})) \setminus \pi_{\text{sifOsoba, prezOsoba}}(\sigma_{\text{sifZadatak}=2}(\text{osoba} \bowtie \text{osobaRadi})))$$

Uz pretpostavku da dvije osobe nemaju isto prezime, rješenje:

$$\pi_{\text{prezOsoba}}(\sigma_{\text{sifZadatak}=1}(\text{osoba} \bowtie \text{osobaRadi})) \setminus \pi_{\text{prezOsoba}}(\sigma_{\text{sifZadatak}=2}(\text{osoba} \bowtie \text{osobaRadi}))$$

7. 1 bod

$$\rho_{\text{ispis}}(\text{sifOsoba, imeOsoba, prezOsoba, brojSati})(\text{sifOsoba, imeOsoba, prezOsoba} \mathrel{\mathcal{G}}_{\text{SUM(satiRadi)}}(\sigma_{\text{datRadi} \geq 1.1.2011}(\text{osoba} \bowtie \text{osobaRadi}))))$$

Uz pretpostavku da dvije osobe nemaju isto prezime, rješenje:

$$\rho_{\text{ispis}}(\text{imeOsoba, prezOsoba, brojSati})(\text{imeOsoba, prezOsoba} \mathrel{\mathcal{G}}_{\text{SUM(satiRadi)}}(\sigma_{\text{datRadi} \geq 1.1.2011}(\text{osoba} \bowtie \text{osobaRadi}))))$$

8. 1 bod

sifZadatak	nazZadatak
1	Konceptualno modeliranje
2	Logičko modeliranje

9. 1 bod

nazZadatak
Konceptualno modeliranje
Programiranje

10. 0.5 boda

```
SELECT DISTINCT imeOsoba, prezOsoba
FROM osoba
JOIN osobaRadi
ON osoba.sifOsoba = osobaRadi.sifOsoba
WHERE sifZadatak = 1
```

## 11. 1.5 boda

```
SELECT imeOsoba, prezOsoba, AVG(satiRadi) AS prosjek
FROM osoba
    LEFT JOIN osobaRadi
        ON osoba.sifOsoba = osobaRadi.sifOsoba
    LEFT JOIN zadatak
        ON osobaRadi.sifZadatak = zadatak.sifZadatak
WHERE zadatak.sifZadatak = 1
    OR zadatak.sifZadatak = 3
GROUP BY imeOsoba, prezOsoba
```

## 12. 0.5 boda

AND	true	unknown	false
true	true	unknown	false
unknown	unknown	unknown	false
false	false	false	false



# 1. međuispit iz Baza podataka

3. travnja 2007.

1. Nadopunite započeti tekst definicije kopije n-torke.

Neka su n-torke  $t_1 = \langle a_1, a_2, \dots, a_n \rangle$  i  $t_2 = \langle b_1, b_2, \dots, b_n \rangle$  definirane na relacijskoj shemi  $R = \{ A_1, A_2, \dots, A_n \}$ . n-torka  $t_1$  je kopija n-torke  $t_2$  ako i samo ako ... (1 bod)

2. a) Koji uvjet mora biti zadovoljen da bi relacije  $r(R)$  i  $s(S)$  bile unijski kompatibilne? (1 bod)

b) Koje se operacije relacijske algebre mogu obaviti s operandima  $r(R)$  i  $s(S)$  onda i samo onda kada su relacije  $r(R)$  i  $s(S)$  unijski kompatibilne? (1 bod)

Zadaci 3 - 7 se odnose na relacije opisane na **slici 1**. Na slici **nisu** prikazane sve n-torke koje su sadržane u relacijama.

**Slika 1.**

stud					nast		
mbrStud	prez	ime	sifNastSem	sifNastProj	sifNast	prez	ime
101	Turk	Ivan	1001	1002	1001	Ribić	Matko
102	Anić	Josip	1001	1003	1002	Anić	Alen
103	Ban	Ana	1003	1002	1003	Oreb	Iva
104	Bašić	Tea	1001	1003			
...					...		

pred		ispit				
sifPred	nazPred	mbrStud	sifPred	datIsplit	sifNast	ocjena
10	Kemija	101	10	15.1.2006	1003	1
11	Mehanika	101	10	15.1.2007	1002	4
12	Fizika	102	10	15.1.2007	1001	3
		102	11	15.1.2006	1002	5
...		...				

Relacija **stud** opisuje studente: atribut *sifNastSem* je šifra nastavnika koji je studentov mentor na seminarskom radu; atribut *sifNastProj* je šifra nastavnika koji je studentov mentor na projektu. Relacija **nast** opisuje nastavnike. Relacija **pred** opisuje predmete. Relacija **ispit** opisuje ispite: studenta *mbrStud* je na ispitu iz predmeta *sifPred* na datum *datIsplit* nastavnik *sifNast* ocijenio ocjenom *ocjena*.

3. Napisati **izraz relacijske algebre (ne SQL upit)** čiji je rezultat relacija  $\text{nastE}(\{ \text{imeNastE} \})$ . Relacija **nastE** sadrži **različita** imena nastavnika koji su držali ispit iz predmeta Elektronika. (2 boda)

Za svaki zadatak 4 - 7 **napisati po jedan SQL upit** kojim će se dobiti traženi rezultat. Pri rješavanju zadataka 4 - 7 možete pretpostaviti da u relacijama nema NULL vrijednosti.

4. Ispisati matične brojeve i prezimena studenata kojima prezime započinje slovom A ili završava slovom b, te su položili ispit (tj. na ispitu dobili ocjenu veću od 1) iz predmeta sa šifrom 30. (1 bod)

5. Ispisati matične brojeve, prezimena i imena studenata kojima je mentor na seminaru nastavnik s prezimenom Kolar, a mentor na projektu **nije** nastavnik s prezimenom Novak. (1 bod)

6. Ispisati **različite** inicijale studenata (npr. za studenta **Turk Ivan** treba ispisati **T.I.**). Ime atributa (stupca) za inicijale studenta treba biti *inicijali*. U rezultatu se ne smiju više puta pojaviti isti inicijali. (1 bod)

7. Ispisati **prvih stotinu najboljih** studenata s rang liste studenata poredanih prema broju položenih ispita. Studenti koji eventualno imaju međusobno jednak broj položenih ispita poredani su abecedno po prezimenu i imenu. Ispisuje se matični broj, prezime i ime studenta, te broj položenih ispita. Ime atributa (stupca) za broj položenih ispita treba biti *brojPolozenihIspita*. (2 boda)

Zadaci 8 - 11 se odnose na relacije opisane na **slici 2**.

**Slika 2.**

stud			mjesto			zupanija	
<i>mbrSt</i>	<i>prez</i>	<i>pbrSt</i>	<i>pbr</i>	<i>nazMjesto</i>	<i>sifZup</i>	<i>sifZup</i>	<i>nazZup</i>
101	Turk	42000	52100	Pula	1	1	Istarska
102	Kolar	NULL	42000	Varaždin	2	2	Varaždinska
103	Novak	52100	52000	NULL	1	3	Karlovačka
			23000	Zadar	NULL	NULL	Zadarska

8. Napišite **rezultat obavljanja** sljedeće operacije (rezultat napisati u obliku tablice):

$$\sigma_{\text{nazMjesto} \neq \text{'Varaždin'}} ( \text{mjesto} \bowtie \text{zupanija} ) \quad (1 \text{ bod})$$

9. Napišite **rezultat obavljanja** sljedeće operacije (rezultat napisati u obliku tablice):

$$\text{stud} \underset{\text{pbrSt=pbr}}{* \bowtie *} ( \pi_{\text{pbr}} (\text{mjesto}) ) \quad (1 \text{ bod})$$

10. Napišite **rezultat obavljanja** sljedeće operacije (rezultat napisati u obliku tablice):

$$\text{zupanija} \underset{\text{pbr=pbrSt}}{* \bowtie} ( \text{mjesto} \bowtie \text{stud} ) \quad (1 \text{ bod})$$

11. Napišite **SQL upit** kojim će se obaviti sljedeća operacija:

$$\text{zupanija} \underset{\text{pbr=pbrSt}}{* \bowtie} ( \text{mjesto} \bowtie \text{stud} ) \quad (2 \text{ boda})$$

## RJEŠENJA:

1) ...  $\forall i, 1 \leq i \leq n$ , vrijedi:  $(a_i = b_i) \vee (a_i \text{ jest NULL} \wedge b_i \text{ jest NULL})$

- 2) a) relacije  $r(R)$  i  $s(S)$  su unijski kompatibilne ukoliko vrijedi:
- relacije su istog stupnja
  - korespondentni atributi su definirani nad istim domenama

b) unija, razlika, presjek

3)  $\rho_{\text{nastE(imeNastE)}} \left( \pi_{\text{ime}} \left( \sigma_{\text{nazPred}='Elektronika'} (\text{nast} \bowtie \text{ispit} \bowtie \text{pred}) \right) \right)$

4) 

```
SELECT stud.mbrStud, prez
FROM stud, ispit
WHERE stud.mbrStud = ispit.mbrStud
AND sifPred = 30
AND ocjena > 1
AND (prez LIKE 'A%' OR prez LIKE '%b');
```

5) 

```
SELECT mbrStud, stud.prez, stud.ime
FROM stud, nast AS nastS, nast AS nastP
WHERE sifNastSem = nastS.sifNast
AND sifNastProj = nastP.sifNast
AND nastS.prez = 'Kolar'
AND nastP.prez <> 'Novak';
```

6) 

```
SELECT DISTINCT SUBSTRING(prez FROM 1 FOR 1)
|| '.'
|| SUBSTRING(ime FROM 1 FOR 1)
|| '.' AS inicijali
FROM stud;
```

7) 

```
SELECT FIRST 100 stud.mbrStud
, prez
, ime
, COUNT(stud.mbrStud) AS brojPolozenihIspita
FROM stud, ispit
WHERE stud.mbrStud = ispit.mbrStud
AND ocjena > 1
GROUP BY stud.mbrStud, prez, ime
ORDER BY brojPolozenihIspita DESC, prez, ime;
```

8) 

pbr	nazMjesto	sifZup	nazZup
52100	Pula	1	Istarska

9) 

mbrSt	prez	pbrSt	pbr
101	Turk	42000	42000
102	Kolar	NULL	NULL
103	Novak	52100	52100
NULL	NULL	NULL	52000
NULL	NULL	NULL	23000

10)

<i>sifZup</i>	<i>nazZup</i>	<i>pbr</i>	<i>nazMjesto</i>	<i>mbrSt</i>	<i>prez</i>	<i>pbrSt</i>
2	Varaždinska	42000	Varaždin	101	Turk	42000
1	Istarska	52100	Pula	103	Novak	52100
3	Karlovačka	NULL	NULL	NULL	NULL	NULL
NULL	Zadarska	NULL	NULL	NULL	NULL	NULL

11)

```
SELECT zupanija.*
      , pbr
      , nazMjesto
      , stud.*
FROM stud
     INNER JOIN mjesto
       ON stud.pbrSt = mjesto.pbr
     RIGHT OUTER JOIN zupanija
       ON mjesto.sifZup = zupanija.sifZup;
```

# 1. međuispit iz Baza podataka

2. travnja 2008.

1. U kojem slučaju rezultat Kartezijeva produkta dviju relacija ne bi bila relacija? Kako se rješava taj problem? **(0.5 bodova)**

Rješenje:

Ako relacije imaju istoimene atribute. Problem se rješava preimenovanjem atributa.

2. Zadane su relacije  $r(R)$ ,  $s(S)$  i  $t(T)$ .  $R=\{A, B, C\}$ ,  $S=\{B, C, D\}$ ,  $T=\{E, F\}$ ,  $\text{card}(r)=5$ ,  $\text{card}(s)=6$ ,  $\text{card}(t)=7$ . U relacijama **nema** NULL vrijednosti.

- a) odredite **stupanj** i **kardinalnost** rezultata operacije  $r \times t$  **(0.5 bodova)**  
 b) odredite **stupanj** i **kardinalnost** rezultata operacije  $r \bowtie r$  **(0.5 bodova)**  
 c) odredite **stupanj** rezultata operacije  $r \stackrel{*}{\bowtie} s$  **(0.5 bodova)**

Rješenje:

- a)  $\text{card} = 35$ ,  $\text{deg} = 5$   
 b)  $\text{card} = 5$ ,  $\text{deg} = 3$   
 c)  $\text{deg} = 4$

Zadaci 3 i 4 odnose se na relacije opisane na **slici 1**.

**Slika 1.**

igrac			klub			mjesto	
mbri	prez	sifKlub	sifKlub	nazKlub	pbr	pbr	nazMjesto
101	Modrić	1	1	Dinamo	10000	10000	Zagreb
102	Kalinić	2	2	Hajduk	21000	21000	Split
103	Čale	1	3	Varteks	NULL	NULL	Varaždin
104	Papa	3	4	NULL	10000	51000	Rijeka

3. Napišite **rezultat obavljanja** sljedećih operacija (rezultat napisati u obliku tablice):

- a)  $(\pi_{\text{nazKlub}, \text{pbr}}(\text{klub})) \bowtie \text{mjesto}$  **(1 bod)**  
 b)  $(\text{igrac} \bowtie^* \text{klub}) \bowtie^* \text{mjesto}$  **(1 bod)**

Rješenje:

a)

nazKlub	pbr	nazMjesto
Dinamo	10000	Zagreb
Hajduk	21000	Split
NULL	10000	Zagreb

b)

mbri	prez	sifKlub	nazKlub	pbr	nazMjesto
101	Modrić	1	Dinamo	10000	Zagreb
102	Kalinić	2	Hajduk	21000	Split
103	Čale	1	Dinamo	10000	Zagreb
NULL	NULL	NULL	NULL	NULL	Varaždin
NULL	NULL	4	NULL	10000	Zagreb
NULL	NULL	NULL	NULL	51000	Rijeka

4. Napišite **SQL upit** kojim će se obaviti sljedeće operacije:

a)  $\pi_{\text{nazKlub}}(\text{igrac} \bowtie \text{klub})$  (1 bod)

b)  $(\text{klub} \bowtie \rho_{\text{mjesto}(\text{pbrM}, \text{nazM})(\text{mjesto})} * \text{igrac}) \bowtie \text{igrac}$   
 $\text{klub.pbr} > \text{mjesto.pbrM}$  (1 bod)

Rješenje:

a) `SELECT DISTINCT nazKlub  
FROM igrac JOIN klub  
ON igrac.sifKlub = klub.sifKlub`

b) `SELECT igrac.mbrI  
      , igrac.prez  
      , klub.*  
      , mjesto.pbr AS pbrM  
      , mjesto.nazMjesto AS nazM  
FROM klub JOIN mjesto ON klub.pbr > mjesto.pbr  
LEFT JOIN igrac ON igrac.sifKlub = klub.sifKlub`

Zadaci 5 i 6 odnose se na relacije opisane na **slici 2**. Na slici **nisu** prikazane sve n-torke koje su sadržane u relacijama. U relacijama nema NULL vrijednosti.

**Slika 2.**

Slika 2.

stud				
mbrStud	prez	ime	pbrRod	pbrStan
101	Turk	Ivan	10000	10000
102	Anić	Josip	10000	21000
103	Ban	Ana	21000	21000
104	Bašić	Tea	42000	10000

...

nast		
sifNast	prez	ime
1001	Ribić	Matko
1002	Anić	Alen
1003	Oreb	Iva

...

mjesto	
pbr	nazMjesto
10000	Zagreb
21000	Split
42000	Varaždin

...

pred	
sifPred	nazPred
10	Kemija
11	Mehanika
12	Fizika

...

ispit				
mbrStud	sifPred	datIsplit	sifNast	ocjena
101	10	15.1.2006	1003	1
101	10	15.1.2007	1002	4
102	10	15.1.2007	1001	3
102	11	15.1.2006	1002	5

...

Relacija **stud** opisuje studente: atribut *pbrRod* je poštanski broj mjesta u kojem je student rođen; atribut *pbrStan* je poštanski broj mjesta u kojem student stanuje. Relacija **nast** opisuje nastavnike. Relacija **pred** opisuje predmete. Relacija **mjesto** opisuje mjesta. Relacija **ispit** opisuje ispite: studenta *mbrStud* je na ispitu iz predmeta *sifPred* na datum *datIsplit* nastavnik *sifNast* ocijenio ocjenom *ocjena*.

5. Napisati **izraz relacijske algebre (ne SQL upit)** čiji je rezultat relacija  $\text{mjesto1}(\{ \text{pbr}, \text{brRod} \})$ . N-torka relacije *mjesto1* sadrži poštanski broj mjesta te broj studenata rođenih u tom mjestu. U relaciji se nalaze i mjesta u kojima nije rođen niti jedan student. (1.5 bodova)

Rješenje:

$\rho_{\text{mjesto1}(\text{pbr}, \text{brRod})}(\text{pbr} \bowtie \text{COUNT}(\text{mbrStud})(\text{mjesto} * \text{stud}))$   
 $\text{pbr}=\text{pbrRod}$

6. Napisati po jednu SQL naredbu kojom će se obaviti sljedeće:

a) Ispisati matični broj, ime, prezime, naziv mjesta stanovanja i naziv mjesta rođenja za studente koji **ne** stanuju u mjestu u kojem su rođeni. **Zadatak riješiti bez podupita.** (1 bod)

b) Ispisati šifru i naziv predmeta, te prosjek **pozitivnih** ocjena za predmete kojima je prosjek veći od 3.0 i u nazivu sadrže riječ „osnove“. **Zadatak riješiti bez podupita.** (1.5 bodova)

c) Za svakog studenta ispisati ime i prezime te naziv predmeta i ocjenu iz ispita koje je student polagao tekuće kalendarske godine. Studentima koji **tekuće** kalendarske godine nisu polagali niti jedan ispit za naziv predmeta ispisati riječ „nepoznato“, a za ocjenu ispisati 0. Rezultat treba ovisiti o datumu (godini) obavljanja upita. **Zadatak riješiti bez podupita.** (2 boda)

d) Za svako mjesto ispisati poštanski broj mjesta, naziv mjesta, prosjek pozitivnih ocjena studenata koji u njemu stanuju i prosjek pozitivnih ocjena studenata koji su u njemu rođeni. (1.5 bodova)

e) Ispisati imena i prezimena svih osoba (studenata i nastavnika) kojima ime i prezime sadrže jednak broj znakova **ILI** počinju istim slovom. Rezultate poredati uzlazno po prezimenu, a zatim po imenu. (1.5 bodova)

Rješenje:

a)

```
SELECT mbrStud, prez, ime
      , mjestoStan.nazMjesto AS nazMjestoStan
      , mjestoRod.nazMjesto AS nazMjestoRod
FROM stud, mjesto mjestoStan, mjesto mjestoRod
WHERE stud.pbrStan = mjestoStan.pbr
      AND stud.pbrRod = mjestoRod.pbr
      AND pbrStan <> pbrRod
```

b)

```
SELECT sifPred, nazPred, AVG(ocjena) AS prosjek
FROM pred JOIN ispit ON pred.sifPred = ispit.sifPred
WHERE ocjena > 1
      AND nazPred LIKE '%osnove%'
GROUP BY sifPred, nazPred
HAVING AVG(ocjena) > 3.0
```

c)

```
SELECT ime, prez,
CASE
  WHEN nazPred IS NULL THEN 'nepoznato'
  ELSE nazPred
END AS predmet,
CASE
  WHEN ocjena IS NULL THEN 0
  ELSE ocjena
END AS ocjena
FROM stud LEFT JOIN ispit ON stud.mbrStud = ispit.mbrStud
                        AND YEAR(ispit.datIspit) = YEAR(TODAY)
      LEFT JOIN pred ON ispit.sifPred = pred.sifPred
```

ili

```
SELECT ime, prez,
CASE
  WHEN nazPred IS NULL THEN 'nepoznato'
  ELSE nazPred
END AS predmet,
CASE
```

```

        WHEN ocjena IS NULL THEN 0
        ELSE ocjena
    END AS ocjena
FROM pred JOIN ispit ON ispit.sifPred = pred.sifPred
        RIGHT JOIN stud ON stud.mbrStud = ispit.mbrStud
        AND YEAR(ispit.datIspit) = YEAR(TODAY)

```

d)

```

SELECT pbr, nazMjesto,
(SELECT AVG(ocjena)
    FROM stud JOIN ispit ON stud.mbrStud = ispit.mbrStud
    WHERE stud.pbrStan = mjesto.pbr AND ocjena > 1) AS prosjekStan,
(SELECT AVG(ocjena)
    FROM stud JOIN ispit ON stud.mbrStud = ispit.mbrStud
    WHERE stud.pbrRod = mjesto.pbr AND ocjena > 1) AS prosjekRod
FROM mjesto

```

e)

```

SELECT ime, prez FROM nast
WHERE CHAR_LENGTH(TRIM(ime)) = CHAR_LENGTH(TRIM(prez))
    OR SUBSTRING(ime FROM 1 FOR 1) = SUBSTRING(prez FROM 1 FOR 1)
UNION
SELECT ime, prez FROM stud
WHERE CHAR_LENGTH(TRIM(ime)) = CHAR_LENGTH(TRIM(prez))
    OR SUBSTRING(ime FROM 1 FOR 1) = SUBSTRING(prez FROM 1 FOR 1)
ORDER BY prez, ime

```



# 1. međuispit iz Baza podataka

1. travnja 2009.

1. (2 boda) Zadana je relacija **r** nad shemom **R(x1, x2)**:

Što će se dobiti izvođenjem sljedećih upita:

- a) `SELECT AVG(x1) FROM R`
- b) `SELECT COUNT(x1) FROM R`
- c) `SELECT AVG(x1) FROM R WHERE x2 = b`
- d) `SELECT COUNT(x1) FROM R WHERE x2 = b`

**R**

x1	x2
null	null
6	a
null	b
2	null

Rješenje:

- a) 4
- b) 2
- c) NULL
- d) 0

2. (1 bod) Navedite sve razloge zašto prikazana tablica nije relacija. Što bi se moglo promijeniti da prikazana tablica postane relacija?

a	b	a
1	2	1
1	2	5
1	2	1

Rješenje:

Ima dva atributa sa jednakim imenima, te ima dva jednaka retka. Može se izbaciti zadnja ntorka i preimenovati treći stupac.

Zadatak 3 odnosi se na relacije opisane na **slici 1**.

Slika 1.

osoba				mjesto			zupanija	
mbr	prezime	ime	pbr	pbr	nazivM	sifZ	sifZ	nazivZ
101	Ivić	Ivo	10430	10000	Zagreb	21	4	Istarska
102	Perić	Pero	23232	23000	Zadar	18	5	NULL
103	Marić	Mario	NULL	10430	Samobor	NULL	18	Zadarska
104	Ivić	Ivana	23232	23232	Nin	18	21	Grad Zagreb

3. (4 boda) Napišite **SQL upite** kojima će se obaviti sljedeće operacije te **rezultat obavljanja** tih operacija (rezultat napisati u obliku tablice):

- a)  $\pi_{pbr, nazivM}(osoba \bowtie mjesto \bowtie zupanija)$
- b)  $\rho_{mjesto2(postBr,broj)}(pbr \overline{G}COUNT(mbr)(osoba * \bowtie mjesto))$

Napomena: prilikom pisanja SQL upita u b) dijelu zadatka zanemarite preimenovanje relacije u mjesto2 (attribute i dalje treba preimenovati).

Rješenje:

a)

pbr	nazivM
23232	Nin

```
SELECT DISTINCT osoba.pbr, nazivM
FROM osoba JOIN mjesto ON osoba.pbr = mjesto.pbr
JOIN zupanija ON mjesto.sifZ = zupanija.sifZ
```

b)

mjesto2

postBr	broj
10430	1
NULL	1
23232	2

```
SELECT osoba.pbr AS postBr, COUNT(mbr) AS broj
FROM osoba LEFT OUTER JOIN mjesto
ON osoba.pbr = mjesto.pbr
GROUP BY osoba.pbr
```

Zadaci 4 i 5 odnose se na relacije opisane na **slici 2**. Na slici **nisu** prikazane sve n-torke koje su sadržane u relacijama. Osim atributa **sifNadDjel** u relaciji **djelatnik** niti jedan drugi atribut ne može poprimiti NULL vrijednosti.

Slika 2.

proizvod				djelatnik				
sifP	naziv	cijena	proizvodjac	sifDjel	sifNadDjel	prez	ime	placa
101	Nutella	28.50	Ferrero	1001	1003	Ribić	Matko	5000
102	Napolitanke	24.95	Kraš	1002	1003	Anić	Alen	6000
103	Bajadera	12.29	Kraš	1003	NULL	Oreb	Iva	10000
104	Puding	4.85	Podravka	...				
...								
racun			stavkaRacuna					
sifRacun	sifDjel	datum	sifRacun	rbrStavka	sifP	kolicina		
10	1001	1.1.2009.	10	1	101	1		
11	1001	31.3.2009.	10	2	102	2		
12	1002	31.3.2009.	10	3	104	5		
...			12	1	102	2		
			...					

Relacija **proizvod** opisuje proizvode. Relacija **djelatnik** opisuje djelatnike. Za svakog djelatnika bilježi se i šifra djelatnika koji mu je izravno nadređen. Relacija **racun** opisuje račune. Jedan račun izdaje jedan djelatnik i sadrži podatke o proizvodima koje je djelatnik prodao nekom kupcu. Jedan račun može imati više stavki koje su opisane tablicom **stavkaRacuna**. Svaka stavka odnosi se na jedan proizvod koji je prodan tim računom. Možete pretpostaviti da će količina prodanog proizvoda uvijek biti veća od nule.

4. (2 boda) Napisati izraz relacijske algebre (ne SQL upit) kojim će se ispisati šifre proizvoda koje je prodao djelatnik sa šifrom 1002, a nikada ih nije prodao djelatnik sa šifrom 1004.

Rješenje:

$$\pi_{\text{sifP}}(\sigma_{\text{sifDjel}=1002}(\text{stavkaRacuna} \bowtie \text{racun})) \setminus \pi_{\text{sifP}}(\sigma_{\text{sifDjel}=1004}(\text{stavkaRacuna} \bowtie \text{racun}))$$

5. (6 bodova) Napisati po jednu SQL naredbu kojom će se obaviti sljedeće:

a) (1 bod) Za svakog djelatnika ispisati njegovo ime i prezime, te prezime djelatnika koji mu je izravno nadređen. Za djelatnike koji nemaju nadređenog djelatnika (šifra nadređenog djelatnika je NULL), umjesto prezimena nadređenog djelatnika ispisati 'nema nadređenog'. **Zadatak riješiti bez podupita.**

b) (1 bod) Za svakog djelatnika kojem prezime počinje slovom 'R' i koji je tekuće godine prodao barem jedan proizvod, ispisati šifru, ime, prezime te ukupan broj proizvoda koje je taj djelatnik prodao tekuće godine. Upit mora dati ispravne rezultate i sljedeće godine (npr. ako se upit pokrene 2009. godine, treba gledati proizvode prodane 2009. godine; ako se upit pokrene 2010. godine, treba gledati proizvode prodane 2010. godine). **Zadatak riješiti bez podupita.**

c) (1.5 bodova) Za svaki proizvod kojem je cijena veća od 10 kn ispisati šifru i naziv proizvoda te broj različitih djelatnika koji su taj proizvod prodali u lipnju, srpnju, kolovozu ili rujnu 2008. godine. Proizvodima koji nisu prodani u zadano vrijeme kao broj djelatnika ispisati 0 (ako im je cijena veća od 10kn). **Zadatak riješiti bez podupita.**

d) (1.5 bodova) Ispisati sve podatke o djelatnicima koji su 2008. godine izdali više računa nego 2007. godine. Možete pretpostaviti da su svi djelatnici izdali barem po jedan račun i 2008. i 2007. godine.

e) (1 bod) Za svaki račun koji je izdao djelatnik s ispodprosječnom plaćom, ispisati šifru računa, datum kada je račun izdan te ukupnu vrijednost svih proizvoda prodanih tim računom.

Rješenje:

```
a)
SELECT djelatnik.ime, djelatnik.prez,
       CASE WHEN djelatnik.sifNadDjel IS NULL THEN 'nema nadređenog'
       ELSE d2.prez
       END
FROM djelatnik LEFT JOIN djelatnik d2 ON djelatnik.sifNadDjel = d2.sifDjel;
```

```
b)
SELECT djelatnik.sifDjel, prez, ime, SUM(kolicina)
FROM djelatnik, racun, stavkaRacuna
WHERE djelatnik.sifDjel = racun.sifDjel
      AND racun.sifRacun = stavkaRacuna.sifRacun
      AND prez LIKE 'R%'           -- SUBSTRING(prez FROM 1 FOR 1) = 'R'
      AND YEAR(racun.datum) = YEAR(TODAY)
GROUP BY djelatnik.sifDjel, prez, ime,
```

```
c)
SELECT proizvod.sifP, naziv, COUNT(DISTINCT sifDjel)
FROM racun JOIN stavkaRacuna ON stavkaRacuna.sifRacun = racun.sifRacun
      RIGHT JOIN proizvod
      ON proizvod.sifP = stavkaRacuna.sifP
      AND MONTH(racun.datum) IN (6,7,8,9)
      AND YEAR(racun.datum) = 2008
WHERE proizvod.cijena > 10
GROUP BY proizvod.sifP, naziv
```

```
-- može i ovo
SELECT proizvod.sifP, naziv, COUNT(DISTINCT sifDjel)
FROM proizvod LEFT JOIN stavkaRacuna ON proizvod.sifP = stavkaRacuna.sifP
      LEFT JOIN racun ON stavkaRacuna.sifRacun = racun.sifRacun
      AND MONTH(racun.datum) IN (6,7,8,9)
      AND YEAR(racun.datum) = 2008

WHERE proizvod.cijena > 10
GROUP BY proizvod.sifP, naziv
```

d)

```
SELECT * FROM djelatnik
WHERE
(SELECT COUNT(sifRacun) FROM racun
WHERE djelatnik.sifDjel = racun.sifDjel
AND YEAR(racun.datum) = 2008)
>
(SELECT COUNT(sifRacun) FROM racun
WHERE djelatnik.sifDjel = racun.sifDjel
AND YEAR(racun.datum) = 2007)
```

e)

```
SELECT racun.sifRacun, datum, SUM(cijena*kolicina)
FROM racun JOIN stavkaRacuna ON racun.sifRacun = stavkaRacuna.sifRacun
JOIN djelatnik ON djelatnik.sifDjel = racun.sifDjel
JOIN proizvod ON proizvod.sifP = stavkaRacuna.sifP
WHERE djelatnik.placa < (SELECT AVG(placa) FROM djelatnik)
GROUP BY racun.sifRacun, datum
```

## 2. međuispit iz Baza podataka

13. svibnja 2011.

Zadaci 1 do 7 se odnose na relacije opisane na **slici 1**. Na slici **nisu** prikazane sve n-torke koje su sadržane u relacijama. U relacije sa slike 1 se pohranjuju podaci o trkačima (relacija **trkac**) koji sudjeluju na utrkama (relacija **utrka**). Vrijeme (u sekundama) koje pojedini trkač ostvari u pojedinoj utrci evidentirano je u relaciji **rezultat**. Trkačima koji nisu završili utrku ostvareno vrijeme nije evidentirano (ima NULL vrijednost). Države iz kojih dolaze trkači odnosno države u kojima se održavaju utrke evidentirane su u relaciji **drzava**. Osim atributa **rezVrijeme** u relaciji **rezultat** niti jedan atribut ne može poprimiti NULL vrijednost.

Slika 1.

trkac					rezultat		
sifTrkac	prezime	ime	datRodj	sifDrzava	sifTrkac	sifUtrka	rezVrijeme
101	Petrović	Ivan	04.06.1987	1	101	21	24927
102	Jurić	Ante	12.05.1989	1	102	21	25395
103	Varga	Tamas	04.06.1991	3	102	22	35956
104	Molnar	Antal	25.11.1990	3	103	22	37937
105	Korošec	Janez	05.11.1987	2	105	22	NULL
					101	23	28299

utrka						drzava	
sifUtrka	nazUtrka	datUtrka	sifDrzava	brojKm	indeksTezine	sifDrzava	nazDrzava
21	Istra1	25.10.2009	1	75	6	1	Hrvatska
22	Velebit	15.06.2010	1	100	7	2	Slovenija
23	Red Bull	10.07.2010	4	80	3	3	Mađarska
24	Alpine	13.09.2010	2	50	1	4	Austrija

Napišite po jednu SQL naredbu kojom će se obaviti sljedeće:

1. Za svaki ostvareni rezultat u kojem je prosječna brzina trkača bila veća od 2 m/s ispisati šifru i naziv te utrke te šifru i prezime tog trkača. **Zadatak riješiti bez podupita.** (1 bod)
2. Za **svaku** državu ispisati šifru, naziv i broj utrka (nula ili više) koje su se u njoj održavale u drugoj polovici (7.-12. mjesec) 2010. godine. Ispis poredati silazno po broju održanih utrka. **Zadatak riješiti bez podupita.** (2 boda)
3. Ispisati šifru i naziv svake utrke u kojoj nije sudjelovao nijedan natjecatelj iz države s nazivom 'Italija'. (1,5 bod)
4. Za održanu utrku ispisati šifru i naziv, šifru osobe s najboljim vremenom utrke i spomenuto ostvareno najbolje vrijeme. Napomena: više osoba može ostvariti najbolje vrijeme. (2 boda)
5. Svakoju utrci koju je **završilo** više od 50 natjecatelja smanjiti indeks težine za 1, pri čemu indeks težine ne smije poprimiti vrijednost manju od 1. (2 boda)
6. Napisati SQL naredbe koje će kreirati najmanji mogući broj indeksa koji će omogućiti efikasno obavljanje (pomoću B+ stabla) svih dolje navedenih upita. (2 boda)

```
SELECT * FROM utrka WHERE datUtrka = TODAY AND nazUtrka = 'Velebit';
SELECT * FROM utrka WHERE nazUtrka = 'Red Bull';
SELECT * FROM utrka ORDER BY brojKm;
SELECT * FROM utrka ORDER BY sifUtrka, nazUtrka;
SELECT * FROM utrka ORDER BY sifUtrka DESC;
SELECT * FROM utrka ORDER BY brojKm DESC, sifUtrka ASC, nazUtrka ASC;
```

7. Napisati jedan **izraz relacijske algebre** (ne SQL upit) koji odgovara sljedećem:

- Za sve trkače ispisati šifru, prezime i broj utrka (nula ili više) na kojima su sudjelovali.

(1,5 bod)

8. Uz pretpostavku da na relacijskoj shemi  $R = PQRSTUV$  vrijede funkcijske zavisnosti iz skupa:  $F = \{PQ \rightarrow RT, R \rightarrow QS, Q \rightarrow UV, V \rightarrow U\}$ , ispitajte je li **skup atributa PQ ključ** relacijske sheme? U svakom koraku obavezno navesti koji aksiom ili pravilo se koristi. **(2,5 boda)**
9. U bazi podataka evidentiraju se utrke Formule 1 koje se održavaju tijekom više godina. Relacijska shema FORMULA sastoji se od sljedećih atributa: **(3 boda)**
- $sifVN$  – šifra Velike nagrade
  - $nazVN$  – naziv Velike nagrade (npr. Velika nagrada Australije)
  - $sifVozac$  – šifra vozača
  - $imeVozac$  – ime vozača
  - $prezVozac$  – prezime vozača
  - $sifMomcad$  – šifra momčadi za koju vozač nastupa u utrci
  - $nazMomcad$  – naziv momčadi za koju vozač nastupa u utrci
  - $datUtrka$  – datum održavanja utrke
  - $vrijeme$  – rezultat vozača u utrci (vrijeme od označenog starta do ulaska vozača u cilj)

Vrijede sljedeća pravila:

- istoga datuma održava se samo jedna utrka, a svaka se utrka vozi za jednu Veliku nagradu (npr. 14.3.2010. se održava Velika nagrada Bahreina, 28.3.2010. se održava Velika nagrada Australije,..., 27.3.2011. se održava Velika nagrada Australije ...)
- svaki vozač vozi utrku u automobilu neke momčadi i može mijenjati momčadi tijekom karijere
- moguće je da dvojica vozača u nekoj utrci ostvare posve jednako vrijeme

Određiti ključ relacijske sheme FORMULA tako da ona bude u 1NF, a zatim postupno normalizirati relacijsku shemu na 2NF i 3NF.

10. Zadano je  $B^+$  stablo reda 25 u koje je zapisano 190 000 zapisa. Koliko je UI operacija potrebno obaviti prilikom traženja jednog zapisa u **najlošijem** slučaju (uračunati i jednu UI operaciju koja je potrebna za dohvat bloka s podacima)? Objasniti kako ste došli do rezultata. **(2 boda)**
11. Napisati SQL naredbe koje će kreirati relacije **klub** i **sportas** prema relacijskim shemama  $KLUB = \{sifKlub, nazivKlub\}$  i  $SPORTAS = \{sifSportas, sifKlub, imeSportas, prezimeSportas, spol\}$ . Smisleno odaberite tipove podataka. Prilikom kreiranja relacija osigurati da:
- u relaciji **sportas** atribut **sifSportas** bude primarni ključ
  - ime i prezime sportaša ne mogu poprimiti NULL vrijednost
  - vrijednost atributa spol može biti samo Z ili M
  - šifra kluba u relaciji **sportas** poprima samo vrijednosti atributa **sifKlub** u relaciji **klub**
  - naziv kluba ne može poprimiti NULL vrijednost
  - dva kluba ne mogu imati isti naziv
  - prilikom brisanja zapisa iz relacije **klub** budu obrisani i svi zapisi o sportašima (u relaciji **sportas**) koji se nalaze u obrisanom klubu
- (3,5 boda)**
12. Koristeći **relacijsku algebru** napišite uvjet koji mora biti zadovoljen da bi se dekompozicija relacije  $r(R)$  na relacije  $r_1(R_1), r_2(R_2), \dots, r_n(R_n)$  obavila bez gubitka informacija. **(2 boda)**

1. Za svaki ostvareni rezultat u kojem je prosječna brzina trkača bila veća od 2 m/s ispisati šifru i naziv te utrke te šifru i prezime tog trkača. **Zadatak riješiti bez podupita.** (1 bod)

```
SELECT rezultat.sifutrka, nazutrka, rezultat.siftrkac, prezime
FROM trkac JOIN rezultat ON trkac.siftrkac=rezultat.siftrkac
JOIN utrka ON rezultat.sifutrka=utrka.sifutrka
WHERE (brojKm*1000)/rezVrijeme>2;
```

2. Za **svaku** državu ispisati šifru i naziv države te broj utrka (nula ili više) koje su se u njoj održale u drugoj polovici 2010. godine. Rezultat poredati silazno po broju održanih utrka. **Zadatak riješiti bez podupita.** (2 boda)

```
SELECT drzava.sifdrzava, nazdrzava, COUNT(utrka.sifutrka) AS broj_utrka
FROM drzava
LEFT JOIN utrka on drzava.sifdrzava = utrka.sifdrzava
AND month(datutrka)>6
AND year(datutrka)=2010
GROUP BY drzava.sifdrzava , nazdrzava
ORDER BY broj_utrka DESC;
```

3. Ispisati šifru i naziv svake utrke u kojoj nije sudjelovao nijedan natjecatelj iz države s nazivom 'Italija'. (1.5 boda)

```
SELECT sifutrka, nazutrka
FROM utrka
WHERE sifUtrka NOT IN (
    SELECT sifutrka
    FROM rezultat JOIN trkac ON rezultat.siftrkac=trkac.siftrkac
    JOIN drzava ON drzava.sifdrzava=trkac.sifdrzava
    WHERE nazdrzava='Italija');
```

ili

```
SELECT sifutrka, nazutrka
FROM utrka
WHERE NOT EXISTS (
    SELECT *
    FROM rezultat JOIN trkac ON rezultat.siftrkac=trkac.siftrkac
    JOIN drzava ON drzava.sifdrzava=trkac.sifdrzava
    WHERE nazdrzava='Italija'
    AND rezultat.sifutrka=utrka.sifutrka);
```

ili

```
SELECT utrka.sifutrka, nazutrka
FROM rezultat JOIN trkac ON rezultat.siftrkac=trkac.siftrkac
JOIN drzava ON drzava.sifdrzava=trkac.sifdrzava
RIGHT JOIN utrka ON rezultat.sifutrka=utrka.sifutrka
AND nazdrzava='Italija'
GROUP BY utrka.sifutrka, nazutrka
HAVING COUNT(rezultat.sifUtrka)=0;
```

4. Za održanu utrku ispisati šifru i naziv, šifru osobe (jedne ili više njih) s najboljim vremenom utrke i spomenuto ostvareno najbolje vrijeme. (2 boda)

```
SELECT utrka.sifutrka, utrka.nazutrka, siftrkac, rezvrijeme
FROM rezultat r1 JOIN utrka ON r1.sifutrka=utrka.sifutrka
WHERE r1.rezvrijeme=(
    SELECT MAX(r2.rezvrijeme)
    FROM rezultat r2
    WHERE r1.sifutrka=r2.sifutrka);
```

ili

```
SELECT utrka.sifutrka, utrka.nazutrka, siftrkac, rezvrijeme
FROM rezultat r1 JOIN utrka ON r1.sifutrka=utrka.sifutrka
WHERE (
    SELECT COUNT(*)
    FROM rezultat r2
    WHERE r1.sifutrka=r2.sifutrka AND r2.rezvrijeme>r1.rezvrijeme
)=0;
```

```
SELECT r1.sifUtrka, nazutrka, r1.sifTrkac, r1.rezvrijeme
FROM utrka JOIN rezultat r1 ON utrka.sifutrka=r1.sifutrka
LEFT JOIN rezultat r2 ON r1.sifutrka=r2.sifutrka
AND r2.rezvrijeme>r1.rezvrijeme
GROUP BY r1.sifUtrka, r1.sifTrkac, nazutrka, r1.rezvrijeme
HAVING COUNT(r2.sifUtrka)=0;
```

5. Svakoj utrci koju je **završilo** više od 50 natjecatelja smanjiti indeks težine za 1, pri čemu indeks težine ne smije poprimiti vrijednost manju od 1. (2 boda)

```
UPDATE utrka
SET indeksTezine = indeksTezine - 1
WHERE indeksTezine > 1
AND sifUtrka IN
(
SELECT sifUtrka
FROM rezultat
WHERE rezVrijeme IS NOT NULL
GROUP BY sifUtrka
HAVING COUNT(*)>50);
```

6. Napisati SQL naredbe koje će kreirati najmanji mogući broj indeksa koji će omogućiti efikasno obavljanje (pomoću B+ stabla) svih dolje navedenih upita. (2 boda)

```
CREATE INDEX i1 ON utrka (brojKm, sifUtrka DESC, nazUtrka DESC);
CREATE INDEX i2 ON utrka (nazUtrka, datUtrka);
CREATE INDEX i3 ON utrka (sifUtrka, nazUtrka);
```

7. Za sve trkače ispisati šifru, prezime i broj utrka na kojima su sudjelovali. (1,5 bod)

$P_{\text{REZULTAT(sifra, prezime, brojUtrka)}}(\text{sifTrkac, prezime, } G_{\text{COUNT(sifUtrka)}(\text{trkac}^* > < \text{rezultat})}$

8. Uz pretpostavku da na relacijskoj shemi  $R = \mathbf{PQRSTUV}$  vrijede funkcijske zavisnosti iz skupa:  $F = \{\mathbf{PQ} \rightarrow \mathbf{RT}, \mathbf{R} \rightarrow \mathbf{QS}, \mathbf{Q} \rightarrow \mathbf{UV}, \mathbf{V} \rightarrow \mathbf{U}\}$ , ispitati je li **PQ ključ** relacijske sheme? U svakom koraku obavezno navedite koji aksiom ili pravilo koristite. (2.5 boda)

Provjeriti može li **PQ** biti **ključ** relacijske sheme?

$\mathbf{PQ} \rightarrow \mathbf{PQ}$  (refleksivnost)  
 $\mathbf{PQ} \rightarrow \mathbf{PQ}$  i  $\mathbf{PQ} \rightarrow \mathbf{RT} \Rightarrow \mathbf{PQ} \rightarrow \mathbf{PQRT}$  (akumulacija)  
 $\mathbf{PQ} \rightarrow \mathbf{PQRT}$  i  $\mathbf{Q} \rightarrow \mathbf{UV} \Rightarrow \mathbf{PQ} \rightarrow \mathbf{PQRTUV}$  (akumulacija)  
 $\mathbf{PQ} \rightarrow \mathbf{PQRTUV}$  i  $\mathbf{R} \rightarrow \mathbf{QS} \Rightarrow \mathbf{PQ} \rightarrow \mathbf{PQRSTUV}$  (akumulacija)

**PQ** funkcijski određuje cijelu relacijsku shemu; potrebno je provjeriti određuju li je i podskupovi **P** odnosno **Q**.

$\mathbf{P} \rightarrow \mathbf{P}$  (refleksivnost)

Nije moguće izraz dalje dopuniti s desne strane da uključi **QRSTUV** pa funkcijska zavisnost  $\mathbf{P} \rightarrow \mathbf{QRSTUV}$  ne vrijedi tj. **P** nije ključ.

$\mathbf{Q} \rightarrow \mathbf{Q}$  (refleksivnost)  
 $\mathbf{Q} \rightarrow \mathbf{Q}$  i  $\mathbf{Q} \rightarrow \mathbf{UV} \Rightarrow \mathbf{Q} \rightarrow \mathbf{QUV}$  (akumulacija)

Nije moguće izraz dalje dopuniti s desne strane da uključi **PRSTUV** pa funkcijska zavisnost  $\mathbf{Q} \rightarrow \mathbf{PRSTUV}$  ne vrijedi tj. **Q** nije ključ.



9. U bazi podataka evidentiraju se utrke Formule 1 koje se održavaju tijekom više godina. Relacijska shema FORMULA sastoji se od sljedećih atributa: **(3 boda)**

sifVN	– šifra Velike nagrade
nazVN	– naziv Velike nagrade (npr. Velika nagrada Australije)
sifVozac	– šifra vozača
imeVozac	– ime vozača
prezVozac	– prezime vozača
sifMomcad	– šifra momčadi za koju vozač nastupa u utrci
nazMomcad	– naziv momčadi za koju vozač nastupa u utrci
datUtrka	– datum održavanja utrke
vrijeme	– rezultat vozača u utrci (vrijeme od označenog starta do ulaska vozača u cilj)

Vrijede sljedeća pravila:

- istoga datuma održava se samo jedno natjecanje
- svaki vozač vozi utrku u automobilu neke momčadi i može mijenjati momčadi tijekom karijere
- moguće je da dvojica vozača u nekoj utrci ostvare posve jednako vrijeme

Odaberite ključ relacijske sheme FORMULA tako da ona bude u 1NF a zatim postupno normalizirajte relacijsku shemu na 2NF i 3NF.

1.NF

KLJUČ: (sifVozac, datUtrka)

2.NF

VOZAC (sifVozac, imeVozac, prezVozac)

TERMIN (datUtrka, sifVN, nazVN)

UTRKA (sifVozac, datUtrka, sifMomcad, nazMomcad, vrijeme)

3.NF

VOZAC1 (sifVozac, imeVozac, prezVozac)

TERMIN1 (datUtrka, sifVN)

NATJECANJE(sifNatj, nazVNj)

UTRKA1 (sifVozac, datUtrka, sifMomcad, vrijeme)

MOMCAD (sifMomcad, nazMomcad)

10. Zadano je B+ stablo reda 25 u koje je zapisano 190 000 zapisa. Koliko će biti potrebno obaviti UI operacija prilikom traženja jednog zapisa u najgorem slučaju (uračunati i jednu UI operaciju koja je potrebna za dohvrat bloka s podacima)? Objasniti kako ste došli do rezultata. **(2 boda)**

Najgori slučaj:

Broj kazaljki u internim čvorovima minimalno 13 (osim u korijenu 2), u listovima minimalno 12.

Za 5 razina:  $2 \cdot 13 \cdot 13 \cdot 13 \cdot 12 = 52\,728$

Za 6 razina:  $2 \cdot 13 \cdot 13 \cdot 13 \cdot 13 \cdot 12 = 685\,464$

Sa 6 razina stablo mora obuhvatiti BAREM 685 464 zapisa, što je previše; s 5 razina stablo mora obuhvatiti BAREM 52 728 zapisa, ali može imati i više od toga, tj. potrebnih 190 000.

Stablo treba imati 5 razina; ukupno  $5+1 = 6$  UI operacija

11. Napisati SQL naredbe koje će kreirati relacije **klub** i **sportas** prema relacijskim shemama KLUB {sifKlub, nazivKlub} i SPORTAS {sifSportas, sifKlub, imeSportas, prezimeSportas, spol}. Smisleno odaberite tipove podataka. Prilikom kreiranja relacija osigurati da:

- u relaciji **sportas** atribut **sifSportas** bude primarni ključ
- ime i prezime sportaša ne mogu poprimiti NULL vrijednost
- vrijednost atributa spol može biti samo Z ili M
- šifra kluba u relaciji **sportas** poprima samo vrijednosti atributa **sifKlub** u relaciji **klub**
- naziv kluba ne može poprimiti NULL vrijednost
- dva kluba ne mogu imati isti naziv
- prilikom brisanja zapisa iz relacije **klub** budu obrisani i svi zapisi o sportašima (u relaciji **sportas**) koji se nalaze u obrisanom klubu

(3.5 boda)

```
CREATE TABLE klub(
    sifKlub INTEGER PRIMARY KEY
, nazivKlub NCHAR(25) NOT NULL UNIQUE
);
```

```
CREATE TABLE sportas(
    sifKlub INTEGER NOT NULL REFERENCES klub(sifKluba) ON DELETE CASCADE
, imeSportas NCHAR(25) NOT NULL
, prezimeSportas NCHAR (25) NOT NULL
, spol CHAR(1) CHECK (spol IN ('Z', 'M')) -
, mbrSportas CHAR(11) PRIMARY KEY);
```

12. . Koristeći **relacijsku algebru** napišite uvjet koji mora biti zadovoljen da bi se dekompozicija relacije **r(R)** na relacije **r<sub>1</sub>(R<sub>1</sub>)**, **r<sub>2</sub>(R<sub>2</sub>)**, ..., **r<sub>n</sub>(R<sub>n</sub>)** obavila bez gubitka informacija. (2 boda)

- Relacija **r(R)** se dekomponira na relacije **r<sub>1</sub>(R<sub>1</sub>)**, **r<sub>2</sub>(R<sub>2</sub>)**, ..., **r<sub>n</sub>(R<sub>n</sub>)** bez gubitaka informacija (lossless decomposition) ako vrijedi:

$$r_1(R_1) \bowtie r_2(R_2) \bowtie \dots \bowtie r_n(R_n) = r(R)$$

odnosno

$$\pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \dots \pi_{R_n}(r) = r(R)$$

## 2. međuispit iz Baza podataka

14. svibnja 2010.

Zadaci 1., 2. i 3. odnose se na relacije opisane na **Slici 1.** Relacija **planinar** sadrži osnovne podatke o planinarima, dok relacija **vrh** opisuje planinske vrhove. Nazivi vrhova su jedinstveni. Relacija **put** sadrži podatke o putevima na te vrhove, pri čemu se zapisuje duljina puta te njegova težina (opisana cijelim brojem iz intervala [1,5]). Konačno, relacija **uspon** evidentira prolazak planinara određenim putem, datum kada je uspon izveden te vrijeme koliko je uspon trajao (izraženo u minutama). **NAPOMENA: na slici nisu prikazane sve n-torke navedenih relacija!**

Slika 1.

planinar			
sifPlaninar	prezime	ime	datRodj
1	Car	Pero	13.04.1967
2	Kralj	Ivan	21.02.1972
3	Beg	Marko	15.09.1978
4	Vojvoda	Đuro	21.12.1985
5	Kmet	Slavko	01.02.1973

put			
sifPut	sifVrh	duljinaKm	tezina
1	1	10	2
2	1	12,5	3
3	2	6	1
4	2	7	2
5	3	4	4
6	3	11	2

uspon			
sifPlaninar	sifPut	datumUspon	vrijemeMin
1	1	15.05.2009	120
1	3	21.02.2010	70
2	1	12.07.2009	150
2	2	28.10.2009	210
3	2	15.10.2009	240
3	4	17.04.2010	120

vrh		
sifVrh	nazivVrh	visina
1	Dinara	1831
2	Risnjak	1528
3	Sv. Jure	1762
4	Vojak	1401

- Napišite po jednu SQL naredbu kojom će se obaviti sljedeće:
  - Za svaki vrh viši od 1500 m ispišite naziv i prosječnu težinu svih puteva koji vode do njega. Ispišite samo one vrhove do kojih vodi barem jedan put. **Zadatak riješite bez podupita.** (1 bod)
  - Za **svakog** planinara koji se penjao putem kojeg nijedan drugi planinar nije prošao ispišite šifru planinara i šifru tog puta. (1,5 bod)
  - Za **svakog** planinara ispišite prezime i ukupan broj kilometara koje je prošao planinareći u godini 2009. Ako planinar u godini 2009. nije napravio niti jedan uspon ispišite nulu (0). (NAPOMENA: moguće je da više planinara ima isto prezime; potrebno je izvršiti zaseban ispis za svakoga od njih). (2 boda)
  - Svakom putu kojeg je prošlo više od 3 različita planinara i čija je težina veća od 1 smanjite težinu za 1. (1,5 bod)
  - Ispišite naziv vrha i šifru planinara (jednog ili više njih) koji su se posljednji popeli na taj vrh. (2 boda)
- Napišite po jedan izraz **relacijske algebre (ne SQL upit!)** koji odgovara sljedećem:
  - Ispisati šifru i naziv vrhova na koje ne postoji put kraći od 10 km. (2 boda)
  - Načiniti relaciju **r** sa shemom {sifVrh, nazivVrh, ukupnaDuljina} u kojoj se za svaki vrh ispisuje njegova šifra i naziv te ukupna duljina svih putova koji vode do tog vrha. (1,5 bod)
- Napišite SQL naredbe koje će kreirati najmanji mogući broj indeksa koji će omogućiti efikasno obavljanje (pomoću B+ stabla) svih dolje navedenih upita. (2 boda)
  - SELECT \* FROM planinar WHERE ime='Pero' AND prezime='Car';
  - SELECT \* FROM planinar WHERE ime='Pero' AND datrodj='13.04.1967';
  - SELECT \* FROM planinar ORDER BY prezime DESC, ime;
  - SELECT \* FROM planinar ORDER BY prezime DESC, ime DESC;
  - SELECT \* FROM planinar ORDER BY ime DESC;
  - SELECT \* FROM planinar ORDER BY prezime, ime, datrodj;
- Uz pretpostavku da na relacijskoj shemi **R = ABCDEFG** vrijede funkcijske zavisnosti iz skupa:  
**F = {AB→D, C→EF, AF→C, D→FG}**, ispitajte vrijedi li funkcijska zavisnost **AB→E**.  
U svakom koraku obavezno navedite koji aksiom ili pravilo koristite. (2 boda)

5. Relacija sadrži n-torke sa sljedećim vrijednostima nekog atributa: 1, 2, 3, 4, 5, 6, 7, 8, 9.

Nacrtati B+-stablo reda 4 tako da popunjenost stabla bude:

- a) minimalna (1,5 bod)
- b) maksimalna (1,5 bod)

6. U bazi podataka evidentiraju se poslovi restauracije (obnove) umjetnina. Relacijska shema RESTAURACIJA sastoji se od sljedećih atributa:

- sifUmjet – šifra umjetnine
- nazivUmjet – naziv umjetnine
- datNastUmjet - datum nastanka umjetnine
- OIBPodn – osobni identifikacijski broj podnosioca zahtjeva za restauracijom
- prezPodn – prezime podnosioca zahtjeva za restauracijom
- pbrMjStanPodn – poštanski broj mjesta gdje stanuje podnositelj zahtjeva
- nazivMjStanPodn – naziv mjesta stanovanja podnosioca zahtjeva
- datPreuz – datum kad je umjetnina preuzeta na restauraciju
- sifRest – šifra restauratorskog obrta koji je restaurirao umjetninu
- nazivRest – naziv restauratorskog obrta koji je restaurirao umjetninu

Vrijede sljedeća pravila:

- ista umjetnina se istog dana na restauraciju može preuzeti samo jednom
- istu umjetninu na restauraciju može predati ista ili različita osoba više puta
- svaku pojedinačnu restauraciju umjetnine obavlja samo jedan restauratorski obrt

Odaberite ključ relacijske sheme RESTAURACIJA tako da ona bude u 1NF a zatim postupno normalizirajte relacijsku shemu na 2NF i 3NF. (3 boda)

7. Obavljen je sljedeći niz SQL naredbi:

```
CREATE TABLE skola (  
    sifSkola    INTEGER    PRIMARY KEY CONSTRAINT pkSkola  
    , nazivSkola CHAR (50) NOT NULL );  
  
CREATE TABLE ucenik (  
    OIBUcenik   CHAR(11)    PRIMARY KEY CONSTRAINT pkUcenik  
    , imeUcenik  CHAR (25)   NOT NULL  
    , prezimeUcenik CHAR (25) NOT NULL  
    , spol       CHAR (1)    NOT NULL CHECK (spol IN ('Ž', 'M')) CONSTRAINT cSpol  
    , sifSkola    INTEGER    NOT NULL REFERENCES skola(sifSkola) CONSTRAINT fkUcenikSkola);  
  
INSERT INTO skola (1, 'III gimnazija');  
INSERT INTO ucenik ('11111111119', 'Ana', 'Pavić', 'Ž', 1);
```

Za zadatke a) – c) napisati po jednu SQL naredbu čijim izvođenjem će se pokušati narušiti:

(3,5 boda)

- a) entitetski integritet u relaciji **skola**
- b) integritet ključa u relaciji **ucenik**
- c) domenski integritet atributa spol u relaciji **ucenik**
- d) Napisati po jednu INSERT, DELETE i UPDATE naredbu kojima će se pokušati narušiti referencijski integritet **fkUcenikSkola**
- e) Nadopunite donju SQL naredbu kojom će se osigurati da u relaciji **skola** ne postoje 2 zapisa s jednakim nazivom škole  
ALTER TABLE skola ADD CONSTRAINT

## Rješenja:

1.

a) 

```
SELECT nazivVrha, AVG(tezina) as pros_tezina
FROM vrh, put
WHERE visina>1500
AND put.sifvrh=vrh.sifvrh
GROUP BY nazivVrha
```

b)

```
SELECT DISTINCT sifPlaninar, sifPut
FROM uspon
WHERE sifput NOT IN
    (SELECT sifPut
     FROM uspon u2
     WHERE u2.sifplaninar<>uspon.sifplaninar);
```

ili

```
SELECT DISTINCT sifPlaninar, sifPut
FROM uspon
WHERE NOT EXISTS
    (SELECT sifPut
     FROM uspon u2
     WHERE u2.sifplaninar<>uspon.sifplaninar AND u2.sifPut=uspon.sifPut);
```

c) 

```
SELECT planinar.prezime,
CASE
    WHEN SUM(duljinakm) IS NOT NULL THEN SUM(duljinakm)
    ELSE 0
END
FROM uspon JOIN put ON uspon.sifput=put.sifput
RIGHT JOIN planinar ON planinar.sifplaninar=uspon.sifplaninar
AND YEAR(datumuspona)=2009
GROUP BY planinar.sifPlaninar, planinar.prezime;
```

d)

```
UPDATE put
SET tezina=tezina-1
WHERE tezina>1
AND 3< (SELECT COUNT(distinct sifplaninar)
FROM uspon
WHERE put.sifput=uspon.sifput);
```

e)

```
SELECT nazivvrha, sifplaninar
FROM put JOIN uspon ON put.sifput=uspon.sifput
JOIN vrh ON put.sifvrh=vrh.sifvrh
WHERE datumuspona=(
    SELECT MAX(datumuspona)
    FROM put p2 JOIN uspon u2 ON p2.sifput=u2.sifput
    WHERE p2.sifvrh=put.sifvrh
)
```

2.

a)

$\pi_{\text{sifVrh, nazivVrha}}(\text{vrh}) \setminus \pi_{\text{sifVrh, nazivVrha}}(\sigma_{(\text{duljinaKm} < 10)}(\text{put} \bowtie \text{vrh}))$

b)

$\rho_{\text{r(sifVrh, nazivVrha, ukupnaDuljina)}}(\text{sifVrh, nazivVrha} \bowtie \text{SUM(duljinaKm)}(\text{put} \bowtie \text{vrh}))$

3.

CREATE INDEX i1 ON planinar (prezime, ime, datrodj) (pokriva 1, 4, 6)

CREATE INDEX i2 ON planinar (prezime DESC, ime) (pokriva 3)

CREATE INDEX i3 ON planinar (ime, datrodj) (pokriva 2, 5)

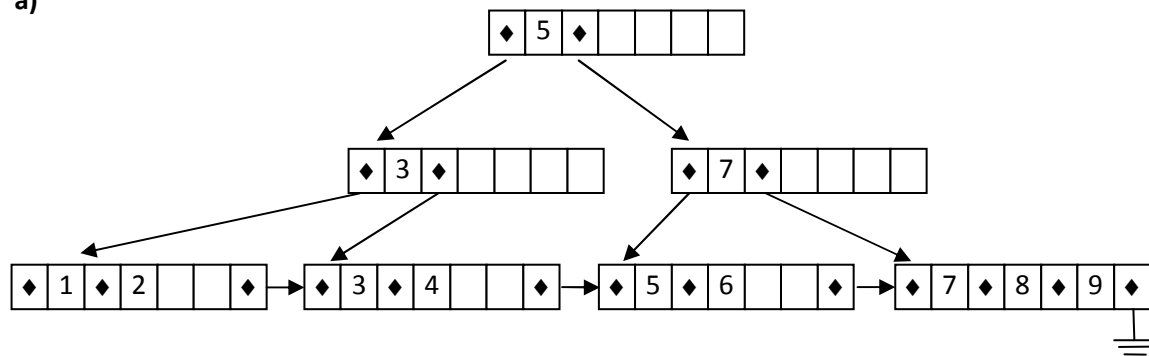
4.

moguće je više rješenja, najjednostavnije je pomoću akumulacije:

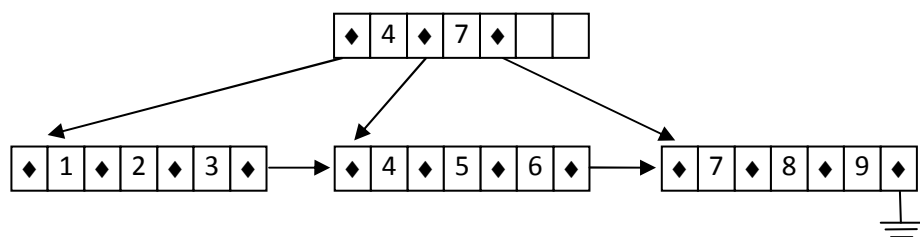
- refleksivnost  $AB \rightarrow AB$
- akumulacija  $AB \rightarrow AB \wedge AB \rightarrow D \Rightarrow AB \rightarrow ABD$
- akumulacija  $AB \rightarrow ABD \wedge D \rightarrow FG \Rightarrow AB \rightarrow ABDFG$
- akumulacija  $AB \rightarrow ABDFG \wedge AF \rightarrow C \Rightarrow AB \rightarrow ABCDFG$
- akumulacija  $AB \rightarrow ABCDFG \wedge C \rightarrow EF \Rightarrow AB \rightarrow ABCDEFG$
- dekompozicija  $AB \rightarrow ABCDEFG \Rightarrow AB \rightarrow E$

5.

a)



b)



## 6.

- sifUmjet – šifra umjetnine
- nazivUmjet – naziv umjetnine
- datNastUmjet - datum nastanka umjetnine
- OIBPodn – osobni identifikacijski broj podnositelja zahtjeva za restauracijom
- prezPodn – prezime podnositelja zahtjeva za restauracijom
- pbrMjStanPodn – poštanski broj mjesta gdje stanuje podnositelj zahtjeva
- nazivMjStanPodn – naziv mjesta stanovanja podnositelja zahtjeva
- datPreuz – datum kad je umjetnina preuzeta na restauraciju
- sifRest – šifra restauratorskog obrta koji je restaurirao umjetninu
- nazivRest – naziv restauratorskog obrta koji je restaurirao umjetninu

### 1NF:

K = { sifUmjet, datPreuz }

### 2NF:

RESTAURACIJA = { sifUmjet, datPreuz, OIBPodn, prezPodn, pbrMjStanPodn, nazivMjStanPodn, sifRest, nazivRest }

UMJETNINA = { sifUmjet, datNastUmjet, nazivUmjet }

### 3NF:

RESTAURACIJA = { sifUmjet, datPreuz, OIBPodn, sifRest }

UMJETNINA = { sifUmjet, datNastUmjet, nazivUmjet }

RESTAURATOR = { sifRest, nazivRest }

PODNOŠITELJ = { OIBPodn, prezPodn, pbrMjStanPodn }

MJESTO = { pbrMjStanPodn, nazivMjStanPodn }

## 7.

a)

INSERT INTO skola VALUES (NULL, 'V gimnazija');

ili

UPDATE skola SET sifSkola = NULL;

b)

INSERT INTO ucenik VALUES (1111111119, 'Ivan', 'Marić', 'M', 1);

c)

INSERT INTO ucenik VALUES (2222222229, 'Ivan', 'Marić', 'L', 1);

ili

UPDATE ucenik SET spol = 'B';

d)

INSERT INTO ucenik VALUES (2222222229, 'Ivan', 'Marić', 'Ž', 3);

UPDATE ucenik SET sifSkola = 5;

DELETE FROM skola WHERE sifSkola = 1;

e)

ALTER TABLE skola ADD CONSTRAINT UNIQUE (nazivSkola) CONSTRAINT uiSkola;

## 2. međuispit iz Baza podataka

15. svibnja 2007.

1. (1 bod) Navedite definiciju tranzitivne funkcijske zavisnosti.

X, Y i Z su skupovi atributa na relacijskoj shemi R. Skup atributa Z je tranzitivno ovisan o X ako vrijedi:

- $X \rightarrow Y$ ,  $Y \not\rightarrow X$  i  $Y \rightarrow Z$
- $Z \not\subseteq XY$

2. (1 bod) Napišite definiciju stranog ključa.

Zadane su relacije  $r(R)$  s primarnim ključem  $PK_R$  i  $s(S)$  s primarnim ključem  $PK_S$ . Skup atributa FK,  $FK \subseteq R$ , je strani ključ u relaciji  $r(R)$  koji se poziva na relaciju  $s(S)$  ukoliko vrijedi:

- atributi u skupu FK imaju domene jednake domenama odgovarajućih atributa u skupu  $PK_S$
- za svaku n-torku  $t_1 \in r(R)$ 
  - postoji n-torka  $t_2 \in s(S)$  takva da je  $t_2[PK_S] = t_1[FK]$

iii

- barem jedna vrijednost atributa iz  $t_1[FK]$  je NULL vrijednost

Zadatak 3 se odnosi na relacije opisane na **slici 1**. Na slici **nisu** prikazane sve n-torke koje su sadržane u relacijama.

Slika 1.

stud	nast
<i>mbrStud</i> <i>prezS</i> <i>imeS</i> <i>sifNastSem</i> <i>sifNastProj</i>	<i>sifNast</i> <i>prezN</i> <i>imeN</i>
101 Turk Ivan 1001 1002	1001 Ribić Matko
102 Anić Josip 1001 1003	1002 Anić Alen
103 Ban Ana 1003 1002	1003 Oreb Iva
104 Bašić Tea 1001 1003	...
...	
pred	ispit
<i>sifPred</i> <i>nazPred</i>	<i>mbrStud</i> <i>sifPred</i> <i>datIspit</i> <i>sifNast</i> <i>ocjena</i>
10 Kemija	101 10 15.1.2006 1003 1
11 Mehanika	101 10 15.1.2007 1002 4
12 Fizika	102 10 15.1.2007 1001 3
...	102 11 15.1.2006 1002 5
	...

3. (4 boda) Relacija **stud** opisuje studente: atribut *sifNastSem* je šifra nastavnika koji je studentov mentor na seminarskom radu; atribut *sifNastProj* je šifra nastavnika koji je studentov mentor na projektu. Relacija **nast** opisuje nastavnike. Relacija **pred** opisuje predmete. Relacija **ispit** opisuje ispite: studenta *mbrStud* je na ispitu iz predmeta *sifPred* na datum *datIspit* nastavnik *sifNast* ocijenio ocjenom *ocjena*. Napisati po jednu SQL naredbu kojom će se obaviti sljedeće:

- Za svaki predmet ispisati šifru predmeta, broj položenih ispita iz tog predmeta (*ocjena* > 1) i broj nepoloženih ispita iz tog predmeta (*ocjena* = 1).
- Ispisati matični broj, ime i prezime za sve studente koji nisu položili niti jedan ispit. Studente poredati po abecedi od A do Ž (prvo po prezimenu, a zatim po imenu).
- Studentima kojima je mentor na seminaru nastavnik Ivo Ivić za mentora na seminaru postaviti nastavnika Peru Perića. Može se pretpostaviti da u relaciji **nast** postoji točno jedan nastavnik Ivo Ivić i točno jedan nastavnik Pero Perić.
- Ispisati podatke o nastavnicima koji su ocijenili barem 100 ispita. Ispisati šifru, prezime i ime nastavnika, te broj ocijenjenih ispita. Zadatak riješiti bez upotrebe podupita.



a)

```
SELECT sifPred
      , (SELECT COUNT(*) FROM ispit
         WHERE ispit.sifPred = pred.sifPred
         AND ocjena > 1)
      , (SELECT COUNT(*) FROM ispit
         WHERE ispit.sifPred = pred.sifPred
         AND ocjena = 1)
FROM pred
```

b)

```
SELECT mbrStud, imeS, prezS
FROM stud
WHERE NOT EXISTS (SELECT * FROM ispit
                  WHERE ispit.mbrStud = stud.mbrStud
                  AND ocjena > 1)
ORDER BY prezS, imeS
```

Može se riješiti sa NOT IN

```
SELECT mbrStud, imeS, prezS
FROM stud
WHERE mbrStud NOT IN (SELECT DISTINCT mbrStud FROM ispit
                      WHERE ocjena > 1)
ORDER BY prezS, imeS
```

c)

```
UPDATE stud SET sifNastSem =
      (SELECT sifNast FROM nast WHERE imeN = 'Pero' AND prezN = 'Perić')
WHERE sifNastSem =
      (SELECT sifNast FROM nast WHERE imeN = 'Ivo' AND prezN = 'Ivić')
```

d)

```
SELECT sifNast, prezN, imeN, COUNT(*) AS brojIspita
FROM nast, ispit
WHERE nast.sifNast = ispit.sifNast
GROUP BY sifNast, imeN, prezN
HAVING COUNT(*) >= 100
```

4. (3 boda) Napisati SQL naredbe koje će kreirati relacije osoba i mjesto prema relacijskim shemama:

OSOBA={ mbr, ime, prez, pbrstan } i MJESTO={ pbr, naziv }.

Smisleno odaberite tipove podataka. Prilikom kreiranja relacija atribut mbr postaviti kao primarni ključ u relaciji osoba i osigurati da se:

- za vrijednost atributa pbrstan u relaciji osoba ne može unijeti NULL
- za vrijednost atributa pbrstan u relaciji osoba mogu unijeti samo mjesta koja postoje u relaciji mjesto
- prilikom brisanja zapisa iz relacije mjesto obrišu i svi zapisi u relaciji osoba koji se odnose na obrisano mjesto
- za vrijednost atributa pbr u relaciji mjesto mora unijeti broj u intervalu [10000, 99999]
- u relaciji mjesto za niti jedan atribut ne može unijeti NULL
- u relaciju mjesto ne mogu unijeti dva mjesta istog naziva

```
CREATE TABLE mjesto(
  pbr INTEGER
, naziv CHAR(30) NOT NULL
, PRIMARY KEY (pbr)
, UNIQUE (naziv)
, CHECK (pbr BETWEEN 10000 AND 99999)
)
```

```

CREATE TABLE osoba(
    mbr INTEGER
  , ime CHAR(30)
  , prezime CHAR(30)
  , pbrstan INTEGER NOT NULL
  , PRIMARY KEY (mbr)
  , FOREIGN KEY (pbrstan) REFERENCES mjesto(pbr)
    ON DELETE CASCADE
)

```

5. (2 bod) Zadano je B+ stablo reda 30 u koje je zapisano 25 000 zapisa. Koliko će biti potrebno obaviti UI operacija prilikom traženja jednog zapisa u **najboljem** i **najgorem** slučaju (uračunati i jednu UI operaciju koja je potrebna za dohvrat bloka s podacima)? Objasniti kako ste došli do rezultata.

Najbolji slučaj – maksimalno popunjeno stablo:

$30 \cdot 30 \cdot 29 = 26\,100$

3 razine + 1 operacija za čitanje bloka s podacima = 4 operacije

Najgori slučaj:

$2 \cdot 15 \cdot 15 \cdot 15 = 6750$

$2 \cdot 15 \cdot 15 \cdot 15 \cdot 15 = 101250$

4 razine + 1 operacija za čitanje bloka s podacima = 5 operacija

6. (2 boda) Zadana je relacija nastavnik (JMBG, ime, prezime, postbr, siforgjed). Napisati SQL naredbe koje će kreirati najmanji mogući broj indeksa koji će omogućiti efikasno obavljanje (pomoću B+ stabla) svih dolje navedenih upita.

```

SELECT * FROM nastavnik ORDER BY prezime, ime;
SELECT * FROM nastavnik WHERE prezime = 'Horvat';
SELECT * FROM nastavnik ORDER BY prezime, ime DESC;
SELECT * FROM nastavnik ORDER BY prezime, ime, JMBG;
SELECT * FROM nastavnik WHERE postbr = 10000 AND siforgjed = 36;
SELECT * FROM nastavnik WHERE siforgjed > 36 ORDER BY siforgjed;

```

Napišite barem jednu SELECT naredbu nad relacijom nastavnik koji se ne može efikasno obaviti kreiranim indeksima.

```

CREATE INDEX i1 ON nastavnik (prezime, ime, JMBG);
CREATE INDEX i2 ON nastavnik (prezime, ime DESC);
CREATE INDEX i3 ON nastavnik (siforgjed, postbr);

```

```

SELECT * FROM nastavnik;

```

```

SELECT * FROM nastavnik
WHERE prezime = 'Horvat' AND siforgjed = 36;

```

7. (3 boda) Zadana je relacijska shema **R = ABCDEFGHI** i skup funkcijskih zavisnosti koji na njoj vrijede:

**F={AC→D, F→GH, E→I, ABC→EF}**. Domene atributa sadrže samo jednostavne vrijednosti, vrijednost svakog atributa je samo jedna vrijednost iz domene tog atributa.

- Ako se za primarni ključ relacije odabere **K={ABC}**, je li zadana relacijska shema u 1NF. Ako nije objasnite zašto i predložite novi ključ tako da zadana relacijska shema bude u 1NF.
- Normalizirajte zadanu relacijsku shemu na 2NF
- Normalizirajte zadanu relacijsku shemu na 3NF

a) Relacijska shema je u 1NF

R = ABCDEFGHI, K<sub>R</sub> = ABC

b)

R1 = ABCEFGHI, K<sub>R1</sub> = ABC

R2 = ACD, K<sub>R2</sub> = AC

c)

R1 = ABCEF, K<sub>R1</sub> = ABC

R2 = ACD, K<sub>R2</sub> = AC

R3 = FGH, K<sub>R3</sub> = F

R4 = EI, K<sub>R4</sub> = E

8. (4 boda) U bazu podataka spremaju se podaci o šahistima i njihovom uspjehu na šahovskim turnirima. Relacijska shema SAH sastoji se od slijedećih atributa:

- mbr – matični broj igrača
- ime – ime igrača
- prez – prezime igrača
- oznKlub – oznaka šahovskog kluba
- nazKlub – naziv šahovskog kluba
- sifTur – šifra turnira
- godTur – godina u kojoj se turnir održava (pretpostavlja se da se u pravilu isti turnir održava

svake godine)

- nazTur – nazivTurnira
- pozicija – mjesto koje je igrač osvojio na turniru (npr. 3. mjesto)
- bodovi – rating bodovi koje je igrač za turniru dobio ili izgubio (može biti pozitivno ili negativno)

Vrijede slijedeća pravila:

- Jedan igrač je član samo jednog kluba
- Turniri se održavaju redovito, ali se isti turnir može održati najviše jednom godišnje
- Igrač jedne godine može nastupiti na više turnira
- Igrač može na istom turniru nastupiti više puta, u više različitih godina

Odaberite ključ relacijske sheme SAH tako da ona bude u 1NF. Postupno normalizirajte relacijsku shemu SAH u 2NF i 3NF.

1NF

K = {mbr, sifTur, godTur}

2NF

IGRAC = {mbr, ime, prez, oznKlub, nazKlub}

TURNIR = {sifTur, nazTur}

REZULTAT = {mbr, sifTur, godTur, pozicija, bodovi}

3NF

IGRAC = {mbr, ime, prez, oznKlub}

KLUB = {oznKlub, nazKlub}

TURNIR = {sifTur, nazTur}

REZULTAT = {mbr, sifTur, godTur, pozicija, bodovi}

Ako je netko pretpostavio da pozicija određuje bodove onda je 3NF

3NF

IGRAC = {mbr, ime, prez, oznKlub}

KLUB = {oznKlub, nazKlub}

TURNIR = {sifTur, nazTur }

REZULTAT = {mbr, sifTur, godTur, pozicija}

BODOVI = {pozicija, bodovi}

## 2. međuispit iz Baza podataka

14. svibnja 2008.

Zadaci 1, 2 i 3 odnose se na relacije opisane na slici 1. Na slici **nisu** prikazane sve n-torke koje su sadržane u relacijama. Relacija **stud** sadrži podatke o studentima. Relacija **pred** sadrži podatke o predmetima. Za svaki predmet definiran je broj ECTS bodova. Relacija **upisanPredmet** sadrži podatke o predmetima koje su studenti upisali određene akademske godine. Ako je student iz predmeta dobio negativnu ocjenu (1), predmet treba upisati ponovno. Ocjena NULL znači da student predmet još uvijek sluša i još nije dobio ocjenu iz predmeta. Relacije postoje u bazi podataka i atributi koji čine primarne ključeve u relacijama su podcrtani.

Slika 1.

stud			pred			upisanPredmet			
<u>mbr</u>	<u>prezS</u>	<u>imeS</u>	<u>sifP</u>	<u>nazP</u>	<u>ECTS</u>	<u>mbr</u>	<u>sifP</u>	<u>akGod</u>	<u>ocjena</u>
101	Turk	Ivan	51	PiPI	4	101	51	2005	4
102	Anić	Josip	52	ASP	5	102	51	2006	5
103	Ban	Ana	53	Fizika I	6	103	53	2005	1
104	Bašić	Tea	...	...	...	103	53	2006	2
105	Jurić	Ante	...	...	...	104	53	2007	NULL

- Napisati naredbu za kreiranje nove relacije u bazi: **studNagrada**, koja će sadržavati sljedeće attribute: **mbr** (matični broj studenta), **akGod** (akademska godina u kojoj je student kandidat za nagradu) **prosjeck** (prosjeck položenih predmeta s dvije znamenke iza decimalne točke), **sumECTS** (ukupan broj ECTS bodova koje je student osvojio) te **oznDobio** (oznaka je li student dobio nagradu). Smisleno odaberite tipove podataka. Atributi **mbr** i **akGod** čine primarni ključ relacije. Osigurati sljedeće:
  - entitetski integritet i jedinstvenost ključa
  - vrijednost niti jednog atributa u relaciji ne može poprimiti NULL vrijednost
  - atribut **mbr** smije poprimiti samo vrijednosti atributa **mbr** u relaciji **stud**
  - atribut **oznDobio** može poprimiti vrijednost 'D' ili 'N'

(2 boda)
- Napisati po jednu SQL naredbu kojom će se obaviti sljedeće:
  - Predmetima koje nikada nije upisao niti jedan student broj ECTS bodova smanjiti za 2.

(1 bod)
  - Iz tablice student obrisati podatke o studentima koji su upisali barem 10 različitih predmeta te dobili ocjenu 1 iz svih predmeta koje su upisali.

(2 boda)
  - Za svakog studenta koji je osvojio ukupno barem 100 ECTS bodova ispisati mbr, ime, prez te prosjeck ocjena položenih predmeta. Zapiše uzlazno poredati po izračunatom prosjeku.

Zadatak riješiti bez upotrebe podupita. (1.5 bodova)
  - Za svaki predmet ispisati podatke o studentima koji taj predmet trenutno slušaju (ocjena je NULL). Ispisati šifru i naziv predmeta te matični broj studenta koji taj predmet trenutno sluša. Ispisati i podatke o predmetima koje trenutno ne sluša niti jedan student. U tom slučaju za matični broj studenta ispisati NULL vrijednost.

(1 bod)
- Napisati izraz relacijske algebre kojim će se dobiti sljedeće:
  - Relacija sa shemom  $R = \{ \text{sifP}, \text{nazP} \}$  koja sadrži predmete koje nikad nije upisao niti jedan student.

(1.5 bodova)
  - Relacija sa shemom  $S = \{ \text{mbr}, \text{prosjeck} \}$  koja za svakog studenta sadrži prosjeck položenih predmeta.

(1 bod)
- Zadana je relacijska shema  $R = \{ A, B, C, D \}$ . Domena svakog od atributa je skup prirodnih brojeva. Prikažite relaciju  $r(R)$  koja sadrži **točno četiri** n-torke. Sadržaj relacije mora biti takav da upućuje na sljedeće zaključke:
  - na shemi  $R$  **sigurno ne vrijedi** funkcijska zavisnost  $A \rightarrow CD$
  - **moгуće je** da na shemi  $R$  **vrijedi** funkcijska zavisnost  $AB \rightarrow CD$

(1 bod)

5. **Izračunajte i obrazložite postupak:** koliko n-torki sadrži relacija ako je nad njom izgrađeno B<sup>+</sup>-stablo reda 11 s ukupno 4 razine.
- a) s **minimalno** popunjenim svim čvorovima (1 bod)
- b) s **maksimalno** popunjenim svim čvorovima (1 bod)

6. Zadana je relacija nastavnik (jmbg, ime, prezime). Napisati SQL naredbe za kreiranje najmanjeg mogućeg broja indeksa koji će omogućiti efikasno obavljanje (pomoću B<sup>+</sup> stabla) navedenih upita.

```
SELECT * FROM nastavnik ORDER BY prezime, ime;
SELECT * FROM nastavnik WHERE ime = 'Marko' AND prezime = 'Horvat';
SELECT * FROM nastavnik ORDER BY prezime, ime DESC;
SELECT * FROM nastavnik ORDER BY prezime DESC, ime, jmbg DESC
```

Napišite jednu SELECT naredbu nad relacijom nastavnik koja se **NE MOŽE** efikasno obaviti kreiranim indeksima. Naredba mora u WHERE dijelu koristiti barem jedan, a u ORDER BY dijelu barem dva atributa. (1.5 bodova)

7. Definirajte pravilo o **akumulaciji**. Uz pretpostavku da na relacijskoj shemi R = XYZUVWQ vrijede funkcijske zavisnosti iz skupa: F = { YZW → Q, XQ → YQ, XY → WY, ZU → Q } korištenjem samo pravila o akumulaciji (uz refleksivnost u prvom koraku i dekompoziciju u zadnjem) dokažite da vrijedi funkcijska zavisnost XYZ → QW. (1.5 bodova)

8. U relaciji inspek(INSPEK) evidentiraju se podaci o inspekcijama koje inspektori provode u tvrtkama. Svaki inspektor je specijaliziran isključivo za jednu vrstu inspekcije (npr. inspektor sa šifrom 100 uvijek provodi samo financijsku inspekciju). Isti inspektor tijekom istog dana najviše jednom dolazi u istu tvrtku.

inspek(INSPEK)								
sifInsp	ime	prez	oznVrInsp	nazVrInsp	sifTvr	nazTvr	datInsp	utrSati
100	Ivan	Horvat	F	Financijska	10	Impexp	1.4.2008.	3
101	Ana	Ban	F	Financijska	10	Impexp	1.4.2008.	4
101	Ana	Ban	F	Financijska	10	Impexp	2.4.2008.	2
100	Ivan	Horvat	F	Financijska	20	Eurotrans	2.4.2008.	5
103	Marko	Kolar	G	Građevinska	20	Eurotrans	2.4.2008.	7
100	Ivan	Horvat	F	Financijska	30	Crotext	1.4.2008.	4

Opis atributa relacijske sheme INSPEK:

- sifInsp            šifra inspektora, jedinstveno određuje inspektora
- ime              ime inspektora
- prez             prezime inspektora
- oznVrInsp       oznaka vrste inspekcije, jedinstveno određuje vrstu inspekcije
- nazVrInsp       naziv vrste inspekcije
- sifTvr            šifra tvrtke, jedinstveno određuje tvrtku
- nazTvr           naziv tvrtke
- datInsp          datum inspekcije
- utrSati          broj sati koje je inspektor utrošio tijekom inspekcije

Odaberite ključ relacijske sheme INSPEK tako da ona bude u 1NF, a zatim postupno normalizirajte relacijsku shemu na 2NF i 3NF.

(4 boda)

## Rješenja:

Zadaci 1, 2 i 3 odnose se na relacije opisane na slici 1. Na slici **nisu** prikazane sve n-torke koje su sadržane u relacijama. Relacija **stud** sadrži podatke o studentima. Relacija **pred** sadrži podatke o predmetima. Za svaki predmet definiran je broj ECTS bodova. Relacija **upisanPredmet** sadrži podatke o predmetima koje su studenti upisali određene akademske godine. Ako je student iz predmeta dobio negativnu ocjenu (1), predmet treba upisati ponovno. Ocjena NULL znači da student predmet još uvijek sluša i još nije dobio ocjenu iz predmeta. Relacije postoje u bazi podataka i atributi koji čine primarne ključeve u relacijama su podcrtani.

Slika 1.

stud			pred			upisanPredmet			
<u>mbr</u>	<u>prezS</u>	<u>imeS</u>	<u>sifP</u>	<u>nazP</u>	<u>ects</u>	<u>mbr</u>	<u>sifP</u>	<u>akGod</u>	<u>ocjena</u>
101	Turk	Ivan	51	PiPI	4	101	51	2005	4
102	Anić	Josip	52	ASP	5	102	51	2006	5
103	Ban	Ana	53	Fizika I	6	103	53	2005	1
104	Bašić	Tea				103	53	2006	2
105	Jurić	Ante				104	53	2007	(null)
...			...			...			

1. Napisati naredbu za kreiranje nove relacije u bazi: **studNagrada**, koja će sadržavati sljedeće attribute: **mbr** (matični broj studenta), **akGod** (akademska godina u kojoj je student kandidat za nagradu) **prosje**k (prosjeak položenih predmeta s dvije znamenke iza decimalne točke), **sumECTS** (ukupan broj ECTS bodova koje je student osvojio) te **oznDobio** (oznaka je li student dobio nagradu). Smisljeno odaberite tipove podataka. Atributi **mbr** i **akGod** čine primarni ključ relacije. Osigurati sljedeće:

- entitetski integritet i jedinstvenost ključa
- vrijednost niti jednog atributa u relaciji ne može poprimiti NULL vrijednost
- atribut **mbr** smije poprimiti samo vrijednosti atributa **mbr** u relaciji **stud**
- atribut **oznDobio** može poprimiti vrijednost 'D' ili 'N'

```
CREATE TABLE studNagrada (  
    mbr INTEGER  
    , akGod SMALLINT  
    , prosjek DECIMAL(4,2) NOT NULL  
    , sumECTS SMALLINT NOT NULL  
    , oznDobio CHAR(1) NOT NULL  
  
    , PRIMARY KEY(mbr, akGod)  
    , FOREIGN KEY(mbr) REFERENCES stud(mbr)  
    , CHECK (oznDobio IN ('D', 'N'))  
)
```

2. Napisati po jednu SQL naredbu kojom će se obaviti sljedeće:

- a) Predmetima koje nikada nije upisao niti jedan student broj ECTS bodova smanjiti za 2.

```
UPDATE pred SET ects = ects - 2
WHERE sifP NOT IN
    (SELECT DISTINCT sifP FROM upisanPredmet)
```

- b) Iz tablice student obrisati podatke o studentima koji su upisali barem 10 različitih predmeta te dobili ocjenu 1 iz svih predmeta koje su upisali.

```
DELETE FROM stud
WHERE (SELECT COUNT(DISTINCT sifP) FROM upisanPredmet
      WHERE stud.mbr = upisanPredmet.mbr) >= 10
AND NOT EXISTS (SELECT * FROM upisanPredmet
                WHERE stud.mbr = upisanPredmet.mbr
                (AND ocjena > 1 OR ocjena IS NULL))

DELETE FROM stud
WHERE (SELECT COUNT(DISTINCT sifP) FROM upisanPredmet
      WHERE stud.mbr = upisanPredmet.mbr) >= 10
AND mbr NOT IN (SELECT DISTINCT mbr FROM upisanPredmet
                WHERE ocjena > 1 OR ocjena IS NULL)
```

- c) Za svakog studenta koji je osvojio ukupno barem 100 ECTS bodova ispisati mbr, ime, prez te prosjek ocjena položenih predmeta. Zapise uzlazno poredati po izračunatom prosjeku. **Zadatak riješiti bez upotrebe podupita.**

```
SELECT stud.mbr, ime, prez, AVG(ocjena) AS prosjek
FROM stud JOIN upisanPredmet ON stud.mbr = upisanPredmet.mbr
      JOIN pred ON upisanPredmet.sifP = pred.sifP
WHERE ocjena > 1
GROUP BY stud.mbr, ime, prez
HAVING SUM(ECTS) >= 100
ORDER BY prosjek
```

- d) Za svaki predmet ispisati podatke o studentima koji taj predmet trenutno slušaju (ocjena je NULL). Ispisati šifru i naziv predmeta te matični broj studenta koji taj predmet trenutno sluša. Ispisati i podatke o predmetima koje trenutno ne sluša niti jedan student. U tom slučaju za matični broj studenta ispisati NULL vrijednost.

```
SELECT sifP, nazP, stud.mbr
FROM pred LEFT JOIN upisanpredmet
      ON pred.sifP = upisanpredmet.sifP
      AND ocjena IS NULL
```

3. Napisati izraz relacijske algebre kojim će se dobiti sljedeće:

- a) Relacija sa shemom  $R = \{ \text{sifP}, \text{nazP} \}$  koja sadrži predmete koje nikad nije upisao niti jedan student.

$$\pi_{\text{sifP}, \text{nazP}}(\text{pred}) \setminus \pi_{\text{sifP}, \text{nazP}}(\text{pred} \bowtie \text{upisanPredmet})$$

- b) Relacija sa shemom  $S = \{ \text{mbr}, \text{prosjek} \}$  koja za svakog studenta sadrži prosjek položenih predmeta.

$$\rho_{S(\text{mbr}, \text{prosjek})}(\text{mbr} \bowtie \text{AVG}(\text{ocjena})(\sigma_{\text{ocjena} > 1}(\text{upisanPredmet})))$$



4. Zadana je relacijska shema  $R = \{ A, B, C, D \}$ . Domena svakog od atributa je skup prirodnih brojeva. Prikažite relaciju  $r(R)$  koja sadrži **točno četiri** n-torke. Sadržaj relacije mora biti takav da upućuje na sljedeće zaključke:

- na shemi  $R$  **sigurno ne vrijedi** funkcijska zavisnost  $A \rightarrow CD$
- **moguće je** da na shemi  $R$  **vrijedi** funkcijska zavisnost  $AB \rightarrow CD$

A	B	C	D
1	2	6	7
1	4	6	8
3	4	9	7
4	3	8	9

5. **Izračunajte i obrazložite postupak:** koliko n-torki sadrži relacija ako je nad njom izgrađeno  $B^+$ -stablo reda 11 s ukupno 4 razine.

- s **minimalno** popunjenim svim čvorovima
- s **maksimalno** popunjenim svim čvorovima

a)  $2 \cdot 6 \cdot 6 \cdot 5 = 360$

U korijenu se nalaze minimalno 2 zapisa. U internim čvorovima se nalazi minimalno  $\lceil n/2 \rceil$  zapisa što je 6. U listovima se nalazi minimalno  $\lceil (n-1)/2 \rceil$  zapisa što je 5.

b)  $11 \cdot 11 \cdot 11 \cdot 10 = 13310$

U korijenu i internim čvorovima se nalazi maksimalno  $n$  zapisa što je 11. U listovima se nalazi maksimalno  $n-1$  zapisa što je 10.

6. Zadana je relacija nastavnik (jmbg, ime, prezime). Napisati SQL naredbe za kreiranje najmanjeg mogućeg broja indeksa koji će omogućiti efikasno obavljanje (pomoću  $B^+$  stabla) navedenih upita.

```
SELECT * FROM nastavnik ORDER BY prezime, ime;
SELECT * FROM nastavnik WHERE ime = 'Marko' AND prezime = 'Horvat';
SELECT * FROM nastavnik ORDER BY prezime, ime DESC;
SELECT * FROM nastavnik ORDER BY prezime DESC, ime, jmbg DESC
```

Napišite jednu SELECT naredbu nad relacijom nastavnik koja se **NE MOŽE** efikasno obaviti kreiranim indeksima. Naredba mora u WHERE dijelu koristiti barem jedan, a u ORDER BY dijelu barem dva atributa.

```
CREATE INDEX i1 ON nastavnik(prezime, ime);
CREATE INDEX i2 ON nastavnik(prezime, ime DESC, JMBG);
```

```
SELECT * FROM nastavnik
WHERE ime = 'Marko'
ORDER BY ime, prezime
```

7. Definirajte pravilo o **akumulaciji**. Uz pretpostavku da na relacijskoj shemi  $R = XYZUVWQ$  vrijede funkcijske zavisnosti iz skupa:  $F = \{ YZW \rightarrow Q, XQ \rightarrow YQ, XY \rightarrow WY, ZU \rightarrow Q \}$  korištenjem samo pravila o akumulaciji (uz refleksivnost u prvom koraku i dekompoziciju u zadnjem) dokažite da vrijedi funkcijska zavisnost  $XYZ \rightarrow QW$ .

Pravilo o akumulaciji:

Ako na shemi  $R$  vrijedi  $X \rightarrow VZ$  i  $Z \rightarrow W$ , tada vrijedi i  $X \rightarrow VZW$

Dokaz:

$XYZ \rightarrow XYZ$ ;

refleksivnost

$XYZ \rightarrow XYZ + XY \rightarrow WY \Rightarrow XYZ \rightarrow XYZW$

akumulacija

$XYZ \rightarrow XYZW + YZW \rightarrow Q \Rightarrow XYZ \rightarrow XYZWQ$

akumulacija

$XYZ \rightarrow QW$

dekompozicija

8. U relaciji inspek(INSPEK) evidentiraju se podaci o inspekcijama koje inspektori provode u tvrtkama. Svaki inspektor je specijaliziran isključivo za jednu vrstu inspekcije (npr. inspektor sa šifrom 100 uvijek provodi samo financijsku inspekciju). Isti inspektor tijekom istog dana najviše jednom dolazi u istu tvrtku.

inspek(INSPEK)

sifInsp	ime	prez	oznVrInsp	nazVrInsp	sifTvr	nazTvr	datInsp	utrSati
100	Ivan	Horvat	F	Financijska	10	Impexp	1.4.2008.	3
101	Ana	Ban	F	Financijska	10	Impexp	1.4.2008.	4
101	Ana	Ban	F	Financijska	10	Impexp	2.4.2008.	2
100	Ivan	Horvat	F	Financijska	20	Eurotrans	2.4.2008.	5
103	Marko	Kolar	G	Građevinska	20	Eurotrans	2.4.2008.	7
100	Ivan	Horvat	F	Financijska	30	Crotext	1.4.2008.	4

Opis atributa relacijske sheme INSPEK:

- sifInsp            šifra inspektora, jedinstveno određuje inspektora
- ime              ime inspektora
- prez             prezime inspektora
- oznVrInsp       oznaka vrste inspekcije, jedinstveno određuje vrstu inspekcije
- nazVrInsp       naziv vrste inspekcije
- sifTvr            šifra tvrtke, jedinstveno određuje tvrtku
- nazTvr           naziv tvrtke
- datInsp          datum inspekcije
- utrSati          broj sati koje je inspektor utrošio tijekom inspekcije

Odaberite ključ relacijske sheme INSPEK tako da ona bude u 1NF, a zatim postupno normalizirajte relacijsku shemu na 2NF i 3NF.

**1NF:**

$K = \{sifInsp, sifTvr, datInsp\}$

**2NF:**

INSPEKTOR =  $\{sifInsp, ime, prez, oznVrInsp, nazVrInsp\}$ ,  $K = \{sifInsp\}$

TVRTKA =  $\{sifTvr, nazTvr\}$ ,  $K = \{sifTvr\}$

INSPEKCIJA =  $\{sifInsp, sifTvr, datInsp, utrosenoSati\}$ ,  $K = \{sifInsp, sifTvr, datInsp\}$

**3NF:**

INSPEKTOR2 =  $\{sifInsp, ime, prez, oznVrInsp\}$ ,  $K = \{sifInsp\}$

VRINSPEKCIJA =  $\{oznVrInsp, nazVrInsp\}$ ,  $K = \{oznVrInsp\}$

TVRTKA =  $\{sifTvr, nazTvr\}$ ,  $K = \{sifTvr\}$

INSPEKCIJA =  $\{sifInsp, sifTvr, datInsp, utrosenoSati\}$ ,  $K = \{sifInsp, sifTvr, datInsp\}$

## 2. međuispit iz Baza podataka

13. svibnja 2009.

1. Definirajte Armstrongove aksiome pomoću relacijske sheme R(X,Y,Z).

(1,5 bodova)

### REFLEKSIVNOST

Ako je  $X \subseteq Y$ , tada vrijedi  $Y \rightarrow X$

### UVEĆANJE

Ako u shemi R vrijedi  $X \rightarrow Y$ , tada vrijedi i  $XZ \rightarrow Y$

### TRANZITIVNOST

Ako u shemi R vrijedi  $X \rightarrow Y$  i  $Y \rightarrow Z$ , tada vrijedi i  $X \rightarrow Z$

Zadaci 2.,3., 4. i 5. se odnose na relacije opisane na slici 1. Na slici nisu prikazane sve n-torke koje su sadržane u relacijama.

Slika 1.

Slika 1.

pizza		
sifPiz	nazivPiz	cijena
1	Margarita	35,00
2	Capricciosa	40,00
3	Al Tonno	40,00
4	Calzone	45,00
...		

sastojak			
sifSas	nazivSas	sifKat	calPoKg
1	tijesto	1	1000
2	rajčica	2	100
3	podravec	5	5000
4	šunka	4	2000
5	feferon	6	150
...			

recept		
sifPiz	sifSas	kolicina
1	1	0.5
1	2	0.2
1	3	0.15
2	1	0.5
2	2	0.25
2	3	0.15
2	4	0.15
3	1	0.5
3	2	0.2
3	7	0.2
...		

kategorija	
sifKat	nazivKat
1	tijesto
2	povrće
3	meso
...	

b) U svakom receptu količinu sastojka kategorije 'riba' smanjite za 20%.

**(1.5 bodova)**

```
UPDATE recept
  SET kolicina = 0.8 * kolicina
  WHERE sifSas IN (SELECT sifSas
                   FROM sastojak, kategorija
                   WHERE sastojak.sifKat=kategorija.sifKat
                      AND nazivKat = 'riba');
```

c) Ispisati sve podatke o sastojcima koji se koriste u svim pizzama.

**(1.5 bodova)**

```
SELECT * FROM sastojak
WHERE (SELECT COUNT(*) FROM Pizzza) =
      (SELECT COUNT(DISTINCT sifPiz) FROM recept
       WHERE recept.sifSas = sastojak.sifSas)
```

```
SELECT * FROM sastojak
WHERE NOT EXISTS (SELECT * FROM pizza
                  WHERE NOT EXISTS (SELECT * FROM recept
                                    WHERE recept.sifPiz = pizza.sifPiz
                                       AND recept.sifSas = sastojak.sifSas
                                    )
                  )
```

```
SELECT * FROM sastojak
WHERE NOT EXISTS (SELECT * FROM pizza
                  WHERE sifPiz NOT IN
                      (SELECT sifPiz FROM recept
                       WHERE recept.sifSas = sastojak.sifSas)
```

d) Za svaku kategoriju ispisati šifru i naziv, broj sastojaka koji pripadaju toj kategoriji te broj različitih pizza u kojima se koriste sastojci te kategorije.

**(1.5 bodova)**

```
SELECT sifKat, nazivKat,
(SELECT COUNT(*) FROM sastojak WHERE sastojak.sifKat = kategorija.sifKat),
(SELECT COUNT (DISTINCT sifPiz)
 FROM recept JOIN sastojak ON recept.sifSas = sastojak.sifSas
 WHERE sastojak.sifKat = kategorija.sifKat)
FROM kategorija
```

```
SELECT kategorija.sifKat, kategorija.nazivKat,
COUNT(DISTINCT sastojak.sifSas), COUNT(DISTINCT sifPiz)
FROM kategorija LEFT JOIN sastojak ON kategorija.sifKat = sastojak.sifKat
LEFT JOIN recept ON sastojak.sifSas = recept.sifSas
GROUP BY sifKat, nazivKat
```

3. Napišite SELECT naredbu kojom biste mogli ispitati da li u relaciji *SASTOJAK* eventualno vrijedi funkcijska zavisnost: **nazivSas → sifKat calPoKg**

Objasnite što se iz rezultata izvođenja SELECT naredbe može zaključiti.

(1,5 bodova)

```
SELECT *  
FROM sastojak s1, sastojak s2  
WHERE s1.nazivSas=s2. nazivSas  
AND (s1.sifKat <> s2. sifKat  
OR s1.calPoKg <> s2.calPoKg);
```

Objašnjenje:

SELECT naredba ispisuje (ili broji) sve one n-torke koje ne zadovoljavaju funkcijsku zavisnost. Ako takve postoje → FZ sigurno ne vrijedi. Ako takve n-torke ne postoje → FZ bi mogla vrijediti (ali to sa sigurnošću ne možemo tvrditi).

4. Napisati izraz relacijske algebre kojim će se dobiti relacija ISPIS. Relacija ISPIS sadrži šifru, naziv i broj sastojaka za sve pizze koje koštaju manje od 50kn. (2 boda)

$\rho_{ISPIS(sifra, naziv, brojSas)} (sifPiz, nazivPiz \bowtie \sigma_{cijena < 50} (pizza \bowtie \sigma_{recept}))$

5. Napisati SQL naredbe koje će kreirati najmanji mogući broj indeksa koji će omogućiti efikasno obavljanje (pomoću B+ stabla) svih dolje navedenih upita. (2 boda)

```
SELECT * FROM sastojak ORDER BY nazivSas DESC, sifSas DESC, CalPoKg DESC;  
SELECT * FROM sastojak ORDER BY nazivSas , sifSas  
SELECT * FROM sastojak ORDER BY nazivSas DESC;  
SELECT * FROM sastojak ORDER BY sifKat DESC;  
SELECT * FROM sastojak ORDER BY nazivSas DESC, CalPoKg DESC;  
SELECT * FROM sastojak WHERE CalPoKg >20 AND CalPoKg <2000;  
SELECT * FROM sastojak WHERE sifKat = 2;
```

Napišite barem jednu SELECT naredbu nad relacijom **sastojak** koji se ne može efikasno obaviti kreiranim indeksima. Naredba mora sadržavati barem jedan uvjet u WHERE dijelu.

```
CREATE INDEX i1 ON sastojak (nazivSas, sifSas, CalPoKg);  
CREATE INDEX i2 ON sastojak (nazivSas, CalPoKg);  
CREATE INDEX i3 ON sastojak (sifKat);  
CREATE INDEX i4 ON sastojak (CalPoKg);
```

```
SELECT * FROM sastojak  
WHERE sifSas = 3  
ORDER BY sifKat, CalPoKg
```

6. Uz pretpostavku da na relacijskoj shemi  $R = ABCDEFG$  vrijede funkcijske zavisnosti iz skupa:

$F = \{E \rightarrow B, D \rightarrow C, B \rightarrow DA, GC \rightarrow DE\}$ , ispitajte vrijedi li funkcijska zavisnost  $ADG \rightarrow B$ .

U svakom koraku obavezno navedite koji aksiom ili pravilo koristite.

(1,5 boda)

Rješenje:

moгуće je više rješenja, najjednostavnije je pomoću akumulacije:

- refleksivnost  $ADG \rightarrow ADG$
- akumulacija  $ADG \rightarrow ADG \wedge D \rightarrow C \Rightarrow ADG \rightarrow ADGC$
- akumulacija  $ADG \rightarrow ADGC \wedge GC \rightarrow DE \Rightarrow ADG \rightarrow ADGCE$
- akumulacija  $ADG \rightarrow ADGCE \wedge E \rightarrow B \Rightarrow ADG \rightarrow ADGCEB$
- dekompozicija  $ADG \rightarrow ADGCEB \Rightarrow ADG \rightarrow B$

7. Zadano je B+ stablo reda  $n$  ( $n$  je paran broj). Ako stablo uz minimalnu popunjenost ima 20 000 kazaljki i ako je u tom slučaju za dohvat zapisa prema vrijednosti ključa potrebno točno 6 U/I operacija (uključujući operaciju za dohvat bloka podataka), odredite kojeg je reda stablo. Napomena: zbog činjenice da je  $n$  paran broj vrijedi  $\lceil (n-1)/2 \rceil = \lceil n/2 \rceil$ .

(2 boda)

Rješenje:

Najgori slučaj:

6 operacija = 6-1=5 razina u stablu

S obzirom da je  $n$  parni broj  $\lceil (n-1)/2 \rceil = \lceil n/2 \rceil$  nakon zaokruživanja pa vrijedi:

$$2 * n/2 * n/2 * n/2 * n/2 = 20000$$

$$N * n * n * n = 160\,000$$

$$N = 20$$

**Zadaci 8. i 9.** se odnose na relacijske sheme

VIZA = {oznDrzava, brPutovnica, datumIzdavanje, datumIstek} i

DRZAVA = {oznDrzava, nazivDrzava, clanEU}.

Sheme opisuju proces izdavanja viza za boravak u Hrvatskoj stranim državljanima. Atribut **clanEU** poprima vrijednost 1 za države koje su članovi EU, a za sve ostale države vrijednost 0. Putovnici sa brojem **brPutovnica** iz države **oznDrzava** izdana je viza na dan **datumIzdavanje**. Viza istječe na dan **datumIstek**.

8. Napisati SQL naredbe koje će kreirati relacije **viza** i **drzava** prema relacijskim shemama VIZA i DRZAVA.

Smisleno odaberite tipove podataka. Prilikom kreiranja relacija kombinaciju atributa **oznDrzava**, **brPutovnica** i **datumIzdavanje** postaviti kao primarni ključ u relaciji **viza** i osigurati da:

- o datum isteka vize ne može poprimiti nul-vrijednosti
- o dvije države ne mogu imati isti naziv
- o naziv države i atribut članstva u EU ne mogu poprimiti nul-vrijednosti
- o vrijednost atributa članstva u EU može biti samo 0 ili 1
- o oznaka države u relaciji **viza** poprima samo vrijednosti atributa **oznDrzava** u relaciji **drzava**
- o viza vrijedi najmanje 30, a najviše 90 dana
- o prilikom brisanja zapisa iz relacije **drzava** budu obrisani i svi zapisi u relaciji **viza** koji se odnose na obrisanu državu

(3 boda)

```
CREATE TABLE drzava(  
    oznDrzava nchar(3)  
    , nazivDrzava nchar(70) NOT NULL  
    , clanEU smallint NOT NULL  
    , PRIMARY KEY(oznDrzava)  
    , UNIQUE(nazivDrzava)  
    , CHECK(clanEU IN (1, 0))  
);
```

```
CREATE TABLE viza(  
    oznDrzava nchar(3)  
    , brPutovnica nchar(15)  
    , datumIzdavanje date  
    , datumIstek date NOT NULL  
    , PRIMARY KEY(oznDrzava, brPutovnica, datumIzdavanje)  
    , FOREIGN KEY(oznDrzava) REFERENCES drzava(oznDrzava) ON DELETE CASCADE  
    , CHECK(datumIstek-datumIzdavanje>=30 AND datumIstek-datumIzdavanje<=90)  
);
```

Moguća je i nešto drugačija sintaksa

```
CREATE TABLE drzava(  
    oznDrzava nchar(3) PRIMARY KEY  
    , nazivDrzava nchar(70) NOT NULL UNIQUE  
    , clanEU smallint NOT NULL CHECK(clanEU IN (1, 0))  
);
```

```
CREATE TABLE viza(  
    oznDrzava nchar(3) REFERENCES drzava(oznDrzava) ON DELETE CASCADE  
    , brPutovnica nchar(15)  
    , datumIzdavanje date  
    , datumIstek date NOT NULL  
    , PRIMARY KEY(oznDrzava, brPutovnica, datumIzdavanje)  
    , CHECK(datumIstek-datumIzdavanje>=30 AND datumIstek-datumIzdavanje<=90)  
);
```

9. Napisati SQL naredbu za kreiranje virtualne relacije **unija** kojom će se korisnicima omogućiti pregled, unos, izmjenu i brisanje svih podataka iz relacije viza za osobe koje žive u Europskoj Uniji. **(1.5 bodova)**

```
CREATE VIEW unija (oznDrzava, brPutovnica, datumIzdavanje, datumIstek) AS
    SELECT oznDrzava, brPutovnica, datumIzdavanje, datumIstek
    FROM viza
    WHERE oznDrzava IN
        (
            SELECT oznDrzava FROM drzava WHERE clanEU=1
        )
WITH CHECK OPTION;
```

Ili bez liste atributa:

```
CREATE VIEW unija AS
    SELECT oznDrzava, brPutovnica, datumIzdavanje, datumIstek
    FROM viza
    WHERE oznDrzava IN
        (
            SELECT oznDrzava FROM drzava WHERE clanEU=1
        )
WITH CHECK OPTION;
```

Ili samo sa \*:

```
CREATE VIEW unija AS
    SELECT *
    FROM viza
    WHERE oznDrzava IN
        (
            SELECT oznDrzava FROM drzava WHERE clanEU=1
        )
WITH CHECK OPTION;
```



**10.** U bazi podataka RAZMJENA spremaju se podaci o razmjeni studenata FER-a u sklopu ERASMUS projekta. Relacijska shema baze sastoji se od sljedećih atributa:

JMBAG – matični broj studenta koji se prijavljuje za razmjenu  
sifSveuciliste – šifra sveučilišta na koje se student prijavljuje  
nazSveuciliste – naziv sveučilišta  
sifDržava – šifra države u kojoj se nalazi sveučilište  
nazDržava – naziv države  
pbrGrad – poštanski broj grada u kojem se nalazi sveučilište  
nazGrad – naziv grada  
akGodina – akademske godina za koju se student prijavljuje  
semestar – semestar za koji se student prijavljuje (zimski ili ljetni)  
odobreno – oznaka je li prijava za razmjenu odobrena ili nije

Vrijede sljedeća pravila:

- Student se za istu godinu i semestar može prijaviti na više sveučilišta
- Student se na isto sveučilište može prijaviti više puta, ali svaki puta u različitoj akademskoj godini i semestru
- poštanski broj je jedinstven za gradove unutar jedne države ali dva grada u različitim državama mogu imati jednak poštanski broj

Odaberite ključ relacijske sheme RAZMJENA tako da ona bude u 1NF, a zatim postupno normalizirajte relacijsku shemu na 2NF i 3NF.

Rješenje:

**1NF:**

$K = \{JMBAG, sifSveuciliste, akGodina, semestar\}$

**2NF:**

$PRIJAVA = \{JMBAG, sifSveuciliste, akGodina, semestar, odobreno\}$

$K = \{JMBAG, sifSveuciliste, akGodina, semestar\}$

$SVEUCILISTE = \{sifSveuciliste, nazSveuciliste, pbrGrad, nazGrad, sifDrzava, nazDrzava\}$

$K = \{sifSveuciliste\}$

**3NF:**

$PRIJAVA = \{JMBAG, sifSveuciliste, akGodina, semestar, odobreno\}$

$K = \{JMBAG, sifSveuciliste, akGodina, semestar\}$

$SVEUCILISTE = \{sifSveuciliste, nazSveuciliste, pbrGrad, sifDrzava\}$

$K = \{sifSveuciliste\}$

$DRZAVA = \{sifDrzava, nazDrzava\}, K = \{sifDrzava\}$

$GRAD = \{sifDrzava, pbrGrad, nazGrad\}, K = \{sifDrzava, pbrGrad\}$

Rješenje ako se pretpostavi da je šifra sveučilišta jedinstvena unutar jedne države:

**1NF:**

$K = \{JMBAG, sifSveuciliste, sifDrzava, akGodina, semestar\}$

**2NF:**

$PRIJAVA = \{JMBAG, sifSveuciliste, sifDrzava, akGodina, semestar, odobreno\}$

$K = \{JMBAG, sifSveuciliste, sifDrzava, akGodina, semestar\}$

$SVEUCILISTE = \{sifSveuciliste, nazSveuciliste, pbrGrad, nazGrad, sifDrzava\}$

$K = \{sifSveuciliste, sifDrzava\}$

$DRZAVA = \{sifDrzava, nazDrzava\}, K = \{sifDrzava\}$

**3NF:**

$PRIJAVA = \{JMBAG, sifSveuciliste, sifDrzava, akGodina, semestar, odobreno\}$

$K = \{JMBAG, sifSveuciliste, sifDrzava, akGodina, semestar\}$

$SVEUCILISTE = \{sifSveuciliste, nazSveuciliste, pbrGrad, sifDrzava\}$

$K = \{sifSveuciliste, sifDrzava\}$

$DRZAVA = \{sifDrzava, nazDrzava\}, K = \{sifDrzava\}$

$\text{GRAD} = \{\text{sifDrzava}, \text{pbrGrad}, \text{nazGrad}\}, K = \{\text{sifDrzava}, \text{pbrGrad}\}$