

## 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)
  - Svakiom 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)

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

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)

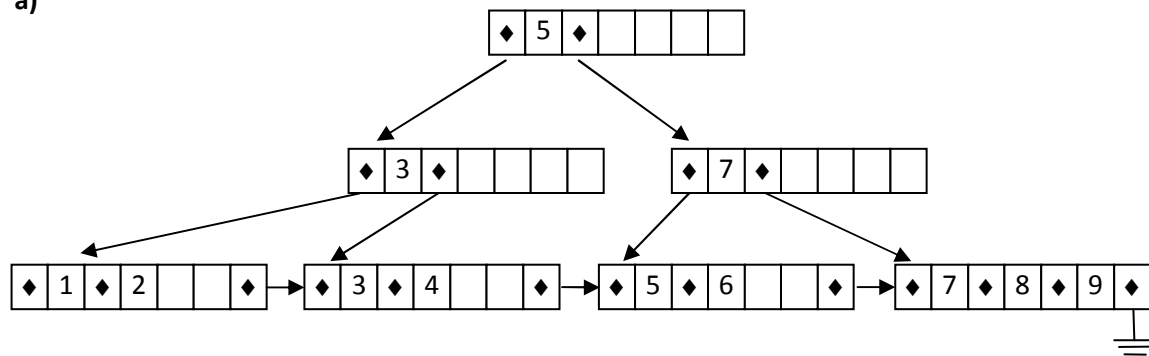
4.

moguće je više rješenja, najjednostavnije je pomoću akumulacije:

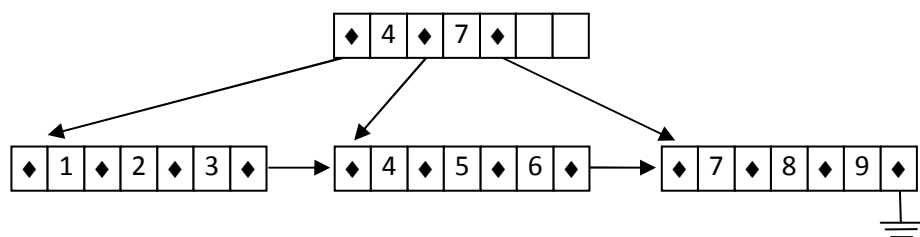
- refleksivnost  $AB \rightarrow AB$
- akumulacija  $AB \rightarrow AB \wedge AB \rightarrow D \Rightarrow AB \rightarrow ABD$
- akumulacija  $AB \rightarrow ABD \wedge D \rightarrow FG \Rightarrow AB \rightarrow ABDFG$
- akumulacija  $AB \rightarrow ABDFG \wedge AF \rightarrow C \Rightarrow AB \rightarrow ABCDFG$
- 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;