

Sustavi baza podataka - Međuispit
28. travnja 2017.

1. a) Nacrtati B⁺-stablo **reda 4** tako da **ukupni broj čvorova** stabla bude **minimalan**. Stablo treba sadržavati sljedeće vrijednosti ključeva: 1, 2, 3, 10, 11, 12, 13, 14, 15, 28, 29, 30 (ukupno 12 vrijednosti ključeva).
- b) Nacrtati stablo koje nastane upisivanjem zapisa s ključem **9** u stablo koje je nacrtano u a) dijelu zadatka.
- c) Nacrtati stablo koje nastane brisanjem zapisa s ključevima **10** i zatim **11** (dakle, brisanjem dva zapisa) iz stabla koje je nacrtano u b) dijelu zadatka.

Dovoljno je nacrtati samo konačna rješenja, posebno za a), posebno za b) i posebno za c) dio zadatka. Blokove s podacima nije potrebno crtati.

2. U bazi podataka kreirane su relacije *racun* i *stavka*. Broj stavki računa redundantno je pohranjen u *brojStavki*. Napisati pohranjenu proceduru **brisanje** kojom se za jedan po jedan račun čiji je *brojStavki* veći od 5, brišu njegove stavke s rednim brojevima većim od 5, te se vrijednost atributa *brojStavki* postavlja na 5. Nakon što se se višak stavki jednog računa obriše i njegova vrijednost za *brojStavki* ažurira, rezultat tih dviju operacija mora ostati trajno zabilježen u bazi podataka.

```
CREATE TABLE racun (  
    brRac          INTEGER,  
    brojStavki     INTEGER,  
    PRIMARY KEY (brRac));  
  
CREATE TABLE stavka (  
    brRac          INTEGER,  
    rbrStavka     INTEGER,  
    PRIMARY KEY (brRac, rbrStavka));
```

Ako se bilo koja pogreška dogodi pri obavljanju operacije DELETE, procedura u pozivajući program vraća pogrešku s tekстом 'Pogreška-brisanje'. Ako se bilo koja pogreška dogodi pri obavljanju operacije UPDATE, procedura u pozivajući program vraća pogrešku s tekстом 'Pogreška-izmjena'. U slučaju pojave bilo koje druge pogreške na bilo kojem drugom mjestu u proceduri, procedura u pozivajući program prosljeđuje originalnu pogrešku. Nije dopuštena upotreba pomoćnih relacija, okidača niti SAVEPOINT mehanizma.

(4 boda)

3. U bazi podataka nalazi se relacija *r*(A, B, C). Za relaciju *r* je izgrađen samo jedan indeks za atribut A. Izvršava se SQL naredba `SELECT lista_atributa FROM r WHERE A = konstanta`. Navesti koji faktori i na koji način utječu na odluku optimizatora hoće li se pri izvršavanju naredbe koristiti navedeni indeks. Odgovor treba napisati u sljedećem obliku:
 - a) *lista_atributa* utječe na sljedeći način: ako lista atributa [nadopuniti tekst] tada se [više/manje] isplati koristiti indeks zato jer [nadopuniti tekst]pod b), c), itd, navesti daljnje faktore i pripadna obrazloženja u sličnom obliku kao pod a)

(4 boda)

4. Zadane su relacije *r*(A, B), *s*(C, D), *t*(E, F). Izvršava se upit koji je opisan sljedećim izrazom relacijske algebre:

$$\sigma_{A=D \wedge B < C \wedge D=200 \wedge F \neq 200} ((r \times s) \times t)$$

- a) nacrtati inicijalno stablo upita (*query tree*) za prikazani izraz relacijske algebre
- b) nacrtati stablo upita i **pripadni izraz relacijske algebre** nakon provedene heurističke optimizacije. Nije potrebno crtati pojedine korake optimizacije (dovoljno je konačno rješenje). Fizičke operatore i način prosljeđivanja međurezultata nije potrebno ucrtavati u stablo.
- c) navesti dva (odabrati po želji) različita pravila za transformaciju izraza relacijske algebre koji su korišteni u b) dijelu zadatka, te za svako od njih opisati kako je korišteno u konkretnom primjeru.

(4 boda)

5. Zadane su relacije $r(A, B, C)$ i $s(D, E)$.

Primarni ključevi su potcrtani. Nema indeksa. Na raspolaganju je 100 blokova primarne memorije. Svi međurezultati se materijaliziraju (zapisuju u sekundarnu memoriju). Za operaciju spajanja koristi se spajanje raspršenim adresiranjem (*hash join*). Operacije se izvršavaju redoslijedom kako je navedeno u izrazu relacijske algebre, što znači da ne treba provoditi heurističku optimizaciju. Izvršava se sljedeća operacija relacijske algebre:

$$\sigma_{B \leq 400} (r) \bowtie_{C=D} \sigma_{E \text{ LIKE } \%T\%} (s)$$

| | | |
|--|---------------------------------|--|
| $V(B, r) = 200$ $V(C, r) = 1000$ $V(E, s) = 5000$ | $N(r) = 8000$ $N(s) = 20000$ | $\min(B, r) = 100$ $\max(B, r) = 500$ $\min(C, r) = 0$ $\max(C, r) = 5000$ $\min(E, r) = 1000$ $\max(E, r) = 10000$ |
| | $B(r) = 3000$ $B(s) = 5000$ | |
| $\text{veličina n-torke}(r) = 0.3 \text{ blokova}$ $\text{veličina n-torke}(s) = 0.2 \text{ blokova}$ | | |

Koristeći priložene podatke iz rječnika podataka, za zadanu operaciju relacijske algebre:

- Prvo procijeniti broj n-torki u međurezultatima i konačnom rezultatu
- Zatim procijeniti ukupni broj **U/I operacija** tijekom izvršavanja operacije relacijske algebre i broj blokova u konačnom rezultatu.

U rješenju prikazati postupak (ne samo konačni rezultat).

(4 boda)

6. Transakcija T je opisana sljedećim pseudokodom:

```
begin work;
  write(z, 100);
  read(x, p1);
  read(z, p2);
  p3 ← p1 + p2;
  write(y, p3);
  write(x, p2);
commit work;
```

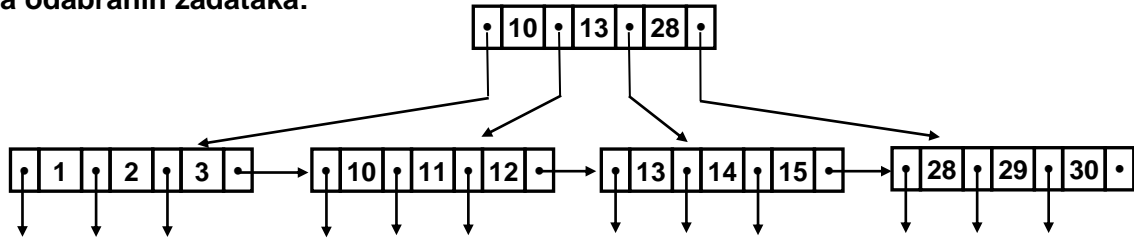
- Nacrtati graf transakcije T. U graf ucrtati isključivo one lukove koji proizlaze iz semantike transakcije ili su nužni da bi se zadovoljila pravila konstrukcije grafa transakcije.
- Transakciju T iz a) dijela zadatka prikazati u obliku topološkog poretka operacija transakcija.
- Napisati definiciju modela transakcije (nije potrebno pisati definiciju parcijalnog poretka).

(4 boda)

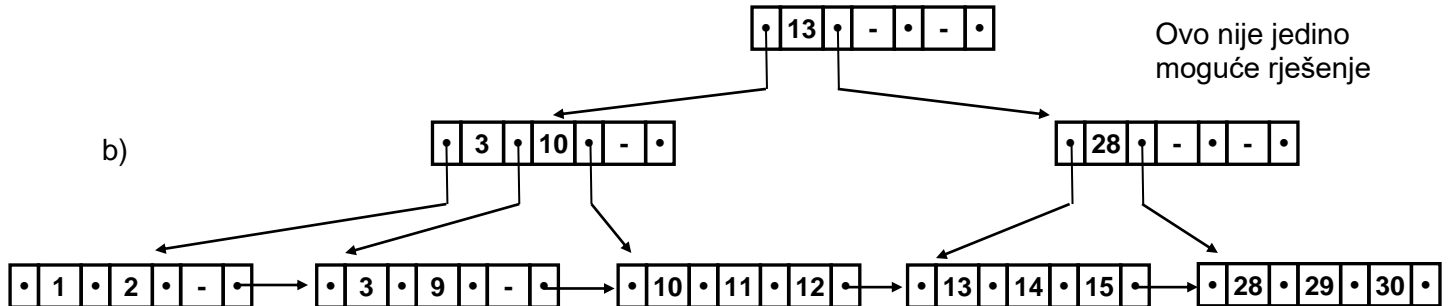
- Što je to *statement-level-rollback*? Navesti jednostavan primjer kojim će se ilustrirati važnost tog mehanizma. (2 boda)
- Objasniti koji se efekti (i zašto) postižu povećanjem, odnosno smanjenjem, učestalosti obavljanja kontrolne točke (*checkpoint*). (2 boda)
- Zašto SUBP koji koristi UNDO tehniku obnove ne smije operaciju *flush log*, kojom se zapis dnevnika <commit T> upisuje u stabilnu memoriju, obaviti prije nego su obavljene sve *output(x)* operacije transakcije T? (2 boda)

Rješenja odabranih zadataka:

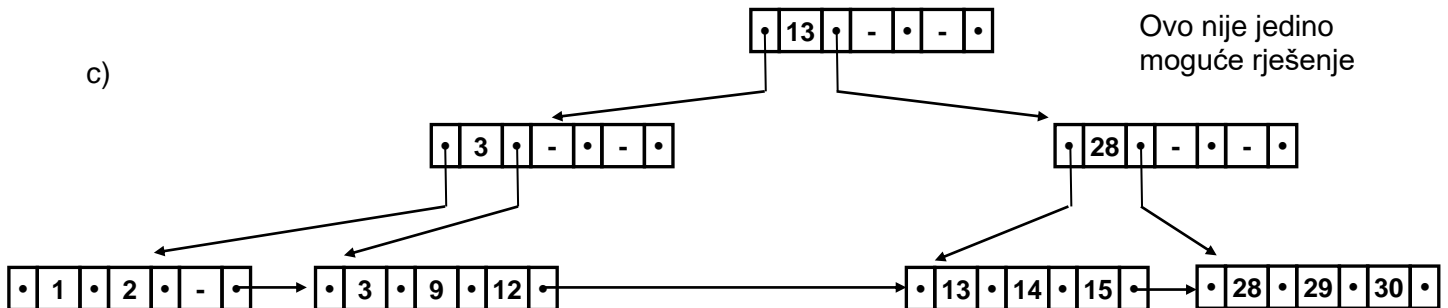
1. a)



b)



c)



2.

```
CREATE PROCEDURE brisanje ()
  DEFINE pBrRac INTEGER;
  DEFINE sqle, isame INTEGER;
  DEFINE errdata CHAR(80);
  FOREACH WITH HOLD
    SELECT brRac INTO pBrRac
    FROM racun WHERE brojStavki > 5
  BEGIN WORK;
  BEGIN
    ON EXCEPTION SET sqle, isame, errdata
      ROLLBACK WORK; -- mora biti ovdje, a ne na početku procedure!
      RAISE EXCEPTION -746, 0, "Pogreška-brisanje";
    END EXCEPTION
    DELETE FROM stavka WHERE brRac = pBrRac AND rbrStavka > 5;
  END
  BEGIN
    ON EXCEPTION SET sqle, isame, errdata
      ROLLBACK WORK; -- mora biti ovdje, a ne na početku procedure!
      RAISE EXCEPTION -746, 0, "Pogreška-izmjena";
    END EXCEPTION
    UPDATE racun SET brojStavki = 5 WHERE brRac = pBrRac;
  END
  COMMIT WORK;
END FOREACH;
END PROCEDURE;
```

3. a) **lista atributa** utječe na sljedeći način: ako lista atributa sadrži samo atribut A tada se više isplati koristiti indeks zato jer se podaci ne trebaju dohvaćati iz blokova s podacima (dovoljan je *key-only-index-scan*)

b) **stupanj grupiranja** utječe na sljedeći način: ako je stupanj grupiranja nepovoljan, tada se manje isplati koristiti indeks jer su n-torke koje zadovoljavaju uvjet selekcije raspršene po više blokova (treba obaviti više U/I operacija za dohvat blokova s podacima). *[ima/nema cluster indeks nije točan odgovor]*

c) **procijenjeni broj n-torki rezultata selekcije** utječe na sljedeći način: ako je taj broj relativno velik, tada se manje isplati koristiti indeks jer trošak prolaska kroz indeks + dohvat blokova s podacima može nadmašiti trošak čitanja svih blokova (linearne pretrage) *[broj različitih vrijednosti, raspršenje vrijednosti, ima/nema histogram, broj n-torki koje vraća upit, itd. nisu točni ili nisu dovoljni odgovori]* /* dubina stabla nije bitna */

4. a) -

b) rezultat optimizacije (stablo): -
 rezultat optimizacije (izraz relacijske algebre): $(\sigma_{A=200}(r) \bowtie_{A=D \wedge B < C} \sigma_{D=200}(r)) \times \sigma_{F \neq 200}(t)$
 c) -

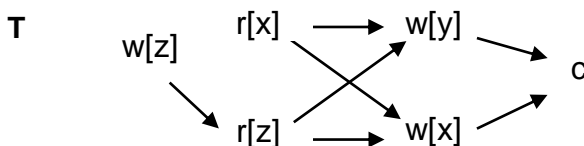
5. a) Broj n-torki

- $t_1 = \sigma_{B \leq 400}(r) = N(t_1) \Rightarrow 8\,000 \cdot (400-100)/(500-100) = 6\,000$
- $t_2 = \sigma_{E \text{ LIKE } \%T\%}(s) \Rightarrow N(t_2) = 20\,000 / 5 = 4\,000$
- $t_3 = t_1 \bowtie_{C=D} t_2 \Rightarrow D \text{ je ključ u } t_2 \Rightarrow N(t_3) = N(t_1) = 6\,000$

b) Broj U/I operacija

- veličina n-torke u konačnom rezultatu $0.3 + 0.2 = 0.5$
- veličina konačnog rezultata $B(t_3) = N(t_3) \cdot 0.5 = 6\,000 \cdot 0.5 = 3\,000$
- čitanje r pri obavljanju $\sigma_{B \leq 400}(r) \Rightarrow 3\,000$
- zapisivanje međurezultata $t_1 \Rightarrow N(t_1) \cdot 0.3 = 6\,000 \cdot 0.3 = 1\,800$
- čitanje s pri obavljanju $\sigma_{E \text{ LIKE } \%T\%}(s) \Rightarrow 5\,000$
- zapisivanje međurezultata $t_2 \Rightarrow N(t_2) \cdot 0.2 = 4\,000 \cdot 0.2 = 800$
- obavljanje $t_1 \bowtie_{C=D} t_2$
 niti jedna ne stane u međuspremnik $\Rightarrow 3 \cdot (B(t_1) + B(t_2)) = 3 \cdot (1\,800 + 800) = 7\,800$
- ukupno (bez zapisivanja konačnog rezultata) $\Rightarrow 3\,000 + 1\,800 + 5\,000 + 800 + 7\,800 = 18\,400$ U/I operacija

6. a)



b) npr. w[z] r[z] r[x] w[x] w[y] c (nije jedino moguće rješenje)

c) -

7. *Statement-level-rollback* nije "korištenje implicitne transakcije za svaku operaciju"; *statement-level-rollback* nije "SAVEPOINT mehanizam". Za točan odgovor pogledati predavanja.

8. -

9. -