

SUSTAVI BAZA PODATAKA

Zadaci i rješenja za završni ispit

Ne garantira se apsolutna točnost svih zadataka.

ZADACI

ZI 2014., 1. Zadatak

Na slici su prikazani grafovi transakcija T_1 , T_2 i T_3 .

T_1
 $r_1[x] \rightarrow w_1[z] \rightarrow c_1$
 $w_1[y] \searrow$

T_2
 $r_2[y] \rightarrow w_2[y] \rightarrow c_2$
 $r_2[x] \searrow$

T_3
 $r_3[x]$
 $r_3[y] \rightarrow c_3$
 $r_3[z] \rightarrow c_3$

a) Nacrtati graf povijesti H_1 koja obuhvaća dvije od tri prikazane transakcije i čije bi izvršavanje (kad bi se dopustilo) izazvalo anomaliju nekonzistentne analize.

b) Nacrtati graf povijesti H_2 koja obuhvaća dvije od tri prikazane transakcije i čije bi izvršavanje (kad bi se dopustilo) izazvalo anomaliju izgubljene izmjene.

c) Za svaku od povijesti H_1 i H_2 navesti koju minimalnu ANSI SQL razinu izolacije sustav treba koristiti kako bi se spriječilo neserijalizabilno izvršavanje povijesti.

d) Za dvije od tri prikazane transakcije vrijedi da je konflikt-serijalizabilna (CSR) **svaka** povijest izvršavanja tih transakcija. Navesti o kojim se transakcijama radi i objasniti zašto za taj par transakcija vrijedi ova tvrdnja. (6 bodova)

i b)

Može netko reći, zašto je potreban luk iz $r_3[z]$ u $r_1[x]$, jel on mogao ići direktno u $w_1[z]$ ako je radi tog konflikta?



7:42 AM Today

morao je

s tim da bi u a) možda čak bilo dobro staviti da nejde iz w1[y] nego iz c1 al valjda je i ovo oke
 najbitnije je za zapamtiti, 3 su problema istodobnog pristupa:

- nekonzistentna analiza
- izgubljena izmjena
- prljavo čitanje

a anomalije su:

- prljavo čitanje
- neponovljivo čitanje (uključuje nekonzistentnu analizu i izgubljenu izmjenu)
- sablasne n-torke

- na temelju tih anomalija, ANSI SQL definira četiri razine izolacije. Svakom višom razinom izolacije osigurava se izbjegavanje većeg broja anomalija

A n o m a l i j e				
		Prljavo čitanje	Neponovljivo čitanje	Sablasne n-torke
Razine izolacije	READ UNCOMMITTED	DA	DA	DA
	READ COMMITTED	NE	DA	DA
	REPEATABLE READ	NE	NE	DA
	SERIALIZABLE	NE	NE	NE

c) I za H1 i za H2 je potreban REPEATABLE READ zato jer nekonzistentna analiza i izgubljena izmjena spadaju u anomaliju neponovljivog čitanja.

d) T2 i T3 zato jer u te dvije transakcije imamo samo jedno pisanje (tj. Jedan par konfliktnih operacija). To znači da imamo samo dvije mogućnosti, ili će prvo biti w2[y] pa r3[y] ili obrnuto. Ta dva slučaja imamo i u serijskom poretku pa tako zaključujemo da je povijest CSR.

ZI 2014., 2. Zadatak

U bazi podataka kreirana je relacija *osoba* i indeks *idx*. U relaciju su upisane samo *n-torke* prikazane na slici (sadržaj relacije je važan). SUBP koristi MGL protokol (uz hijerarhiju objekata: *baza podataka* → *relacija* → *n-torka*) i protokol zaključavanja indeksa, ali samo ako su za korištenje protokola zaključavanja indeksa ostvareni potrebni preduvjeti. Ne koriste se U-ključevi.

```
CREATE TABLE osoba (
  sif INTEGER
, ime CHAR(20)
, prez CHAR(20)
) LOCK MODE ROW;

CREATE UNIQUE INDEX idx
ON osoba (sif);
```

sif	ime	prez
100	Ana	Horvat
103	Ivo	Ban
106	Jure	Ban
109	Ivo	Kolar
112	Jure	Novak
115	Ivo	Novak
118	Ana	Turk
121	Edo	Keler
125	Lino	Hertz

Redoslijed kojim SQL naredbe pristižu u sustav u skladu je s rednim brojevima naredbi. U sustav pristigla SQL naredba obavlja se odmah i u cijelosti (ako nije bilo zapreka za postavljanje potrebnih ključeva).

T ₁ BEGIN WORK; SET LOCK MODE TO WAIT; SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	T ₂ BEGIN WORK; SET LOCK MODE TO WAIT; SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
1. DELETE FROM osoba WHERE sif=118	2. SELECT * FROM osoba WHERE sif=100
3. SELECT * FROM osoba WHERE sif=100	4. UPDATE osoba SET prez='Li' WHERE sif=125
5. INSERT INTO osoba VALUES(118, 'Jo', 'Wu')	6. SELECT * FROM osoba WHERE sif=125
7. SELECT * FROM osoba WHERE prez='Wu'	8. SELECT * FROM osoba WHERE sif=118

Za svaku naredbu (1-8) koja se pokušava obaviti prije nego transakcija uđe u stanje čekanja na postavljanje ključa, napisati kojom vrstom ključa se pokušava zaključati koji element baze podataka. Ako zaključavanje nije uspjelo navesti i razlog zašto. Rješenje treba biti u obliku prikazanom u priloženom primjeru rješenja.

(5 bodova)

Primjer oblika rješenja:

- T₁ postavlja S na n-torke 100, 102, 102, postavlja X na n-torke 105, 107
- T₂ postavlja IS na n-torke 103, 104, X na osoba
- T₁ promovira S u IX na n-torci 102
- T₂ ne uspijeva postaviti S na bazu podataka jer je T₂ već prije s X zaključala relaciju, T₂ čeka
- T₁ uspješno promovira S u X nad bazom podataka

- T1 postavlja IX na bazu i relaciju osoba; X na zapis u indexu koji pokazuje na sif=118; X na n-torku sif=118
- T2 postavlja IS na bazu i relaciju osoba; S na zapis u indexu koji pokazuje na sif=100; S na n-torku sif=100
- T1 ima IX na bazi i relaciji osoba; postavlja S na zapis u indexu koji pokazuje na sif=100 i S na n-torku sif=100
- T2 promovira IS u IX na bazu i relaciju osoba; postavlja S na zapis u indexu koji pokazuje na sif=125 i X na n-torku sif=125
- T1 ima IX na bazi i relaciji osoba; ima već X na zapisu u indexu koji pokazuje na sif=118 i X na n-torci sif=118
- T2 ima IX na bazi i relaciji osoba; ima već S na zapisu u indexu koji pokazuje na sif=125 i X na n-torci sif=125
- T1 ima IX na bazi; želi SIX na relaciji osoba - odbijeno zbog IX od T2, čeka T2

8. T2 ima IX na bazi i relaciji osoba; želi S na zapisu indexa koji pokazuje sif=118 i na n-torci sif=118 - odbijeno zbog X od T1, čeka T1 --> potpuni zastoј

5. Ja sam iz 10. predavanja, slajd 31 zaključila ovako:

Kada neka transakcija obavi DELETE neke n-torke, niti n-torka, niti njen zapis u indeksu se ne brišu fizički dok se ta transakcija ne commita. Ako ta ista transakcija pokuša zapisati novu n-torku na to mjesto, ona to može napraviti (neće napraviti novi zapis u indeksu pa nije problem sa unique). Ako neka druga transakcija pokuša zapisati novu n-torku na to mjesto, onda neće uspjeti zbog X ključa na zapisu indeksa i n-torci.

ZI 2014., 3. Zadatak

Objasniti zašto je u tablici kompatibilnosti ključeva za MGL protokol u presjeku stupca X i retka IS oznaka false, te zašto je u presjeku stupca IX i retka IX oznaka true.

Presjek stupca X i retka IS je negativan jer, ako su prije svi elementi u podstablu bili zaključani ključem X, nije moguće neki od njih naknadno zaključati ključem S.

Presjek stupca IX i retka IX je pozitivan jer IX ključ kaže da će se neki elementi u podstablu zaključati ključem X, a 2 ključa IX naposljetku mogu zaključati neke druge elemente u podstablu.

ZI 2014., 4. Zadatak

Sustav koji koristi 2PL protokol istodobno izvršava transakcije T_1 i T_2 čiji su grafovi prikazani na slici.



- U obliku topološkog poretka operacija transakcija prikazati povijest H koja će izazvati potpuni zastoј u sustavu koji ne koristi ključeve za izmjenu (U-ključeve). Operacije zaključavanja i otključavanja u povijesti H nije potrebno eksplicitno navoditi.
- Objasniti na koji će način sustav koji, pored S-ključeva i X-ključeva, koristi i ključeve za izmjenu (U-ključeve) spriječiti pojavu potpunog zastoја tijekom izvršavanja povijesti H. (5 bodova)

a) $r1[x]$, $r2[x]$, $w1[x]$, $r2[y]$, $w2[x]$, $w1[z]$, $c1$, $c2$

- b) Tako što U-ključ neće dozvoliti operaciju $r_2[x]$ (T2 će u tom trenutku započeti čekanje) dok T1 ne obavi sve ostale operacije te se time sprječava potpuni zastoј.

ZI 2014., 5. Zadatak

Objasniti po čemu se razlikuju temeljni, striktni i rigorozni 2PL protokoli.

Temeljni - propisuje da transakcija Ti mora zaključati sve potrebne elemente prije nego otključa bilo koji element, odnosno transakcija koja je otpustila ključ ne može nakon toga postavljati nove ključeve

Striktni - X-ključevi smiju se otpustiti tek poslije točke potvrđivanja transakcije

Rigorozni - svi (a ne samo X-ključevi) otpuštaju se netom poslije transakcije

ZI 2014., 6. Zadatak

6. Partitionirani sustav obuhvaća čvorove S_1 i S_2 . Korisnici sustava koji pristupaju čvoru S_1 u najvećem broju slučajeva (ali ne isključivo) koriste podatke o osobama čiji je prihod manji ili jednak 50 000, a korisnici sustava koji pristupaju čvoru S_2 u najvećem broju slučajeva (ali ne isključivo) koriste podatke o osobama čiji je prihod veći od 50 000.

```
CREATE TABLE osoba (  
    sif      INTEGER  
, imepres CHAR(50)  
, prihod   INTEGER  
, PRIMARY KEY (sif));
```

- a) Opisati za ovaj slučaj prikladne sheme fragmentacije i alokacije. Shemu fragmentacije treba opisati u obliku izraza relacijske algebre.
b) Što je transparentnost fragmentacije?
c) Objasniti kako bi se u ovom slučaju pomoću SQL objekata mogla osigurati transparentnost lokacije i transparentnost fragmentacije? (4 boda)

a) horizontalna fragmentacija i partitionirana alokacija

$osoba_1 = O_{prihod \leq 50000}(osoba)$

$osoba_2 = O_{prihod > 50000}(osoba)$

- b) Transparentnost fragmentacije - korisnici (ili aplikacije) ne trebaju voditi računa o načinu na koji je relacija fragmentirana
c) Transparentnost lokacije - korisnici (ili aplikacije) ne trebaju znati u kojem je čvoru alociran koji ,fragment. Svakom podatku moraju moći pristupiti ukoliko im je poznat identifikator podatka

Transparentnost lokacije - definiranjem sinonima

CREATE PUBLIC SYNONYM osoba1 FOR baza@subp1:osoba1 i
tako i CREATE PUBLIC SYNONYM osoba2 FOR baza@subp2:osoba2

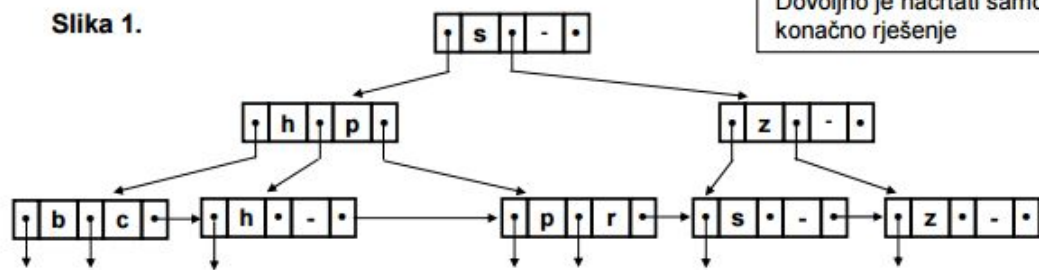
Transparentnost fragmentacije - stvaranjem virtualnih relacija i sinonima

Napravili bismo virtualnu relaciju koja sadrži oba fragmenta tako da korisnici uopće ni ne osjete da je baza particionirana

ZI 2013., 1. Zadatak

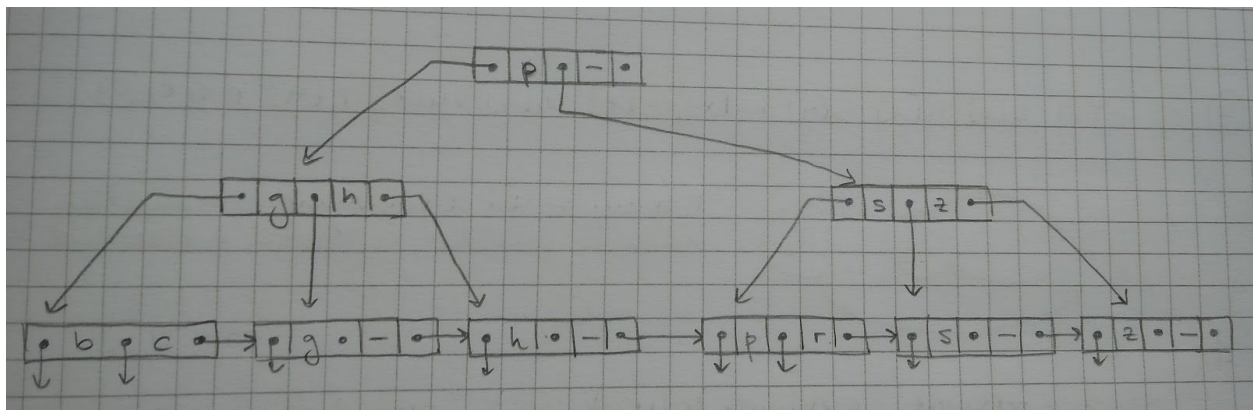
1. Nacrtati B⁺-stablo nakon unosa zapisa s ključem **g** u B⁺-stablo na slici 1.

Slika 1.



Dovoljno je nacrtati samo konačno rješenje

(3 boda)



U pravu si, nije točno

ZI 2013., 2. Zadatak

Spajaju se relacije **r** i **s** koje su pohranjene u B(**r**), odnosno B(**s**) blokova.

a) Objasniti na koji se način obavlja fizička operacija blokovsko spajanje s ugniježđenim petljama (block nested-loop join)

b) Navesti izraze (formule) kojima se procjenjuje trošak te operacije, ovisno o raspoloživoj količini primarne memorije. Objasniti na koji su način ti izrazi izvedeni.

a) Za svaki blok u B(r) te za svaki blok u B(s) spajamo njihove n-torke zajedno

b) najbolji slučaj: samo po jedan blok svake relacije stanu u glavnu memoriju

• procjena troška: $B(r) * B(s) + B(r)$ (nisu uključeni troškovi zapisivanja rezultata)

najbolji slučaj: (jednako kao u nested-loop join)

• procjena troška: $B(r) + B(s)$ (nisu uključeni troškovi zapisivanja rezultata)

1 blok za čitanje unutarnje relacije, 1 blok za izlazni međuspremnik

vanjska petlja: umjesto po jedan blok, $B(r) / (M - 2)$ puta učitava po $M-2$ blokova

procjena troška: $B(r) / (M - 2) * B(s) + B(r)$ (nisu uključeni troškovi zapisivanja rezultata)

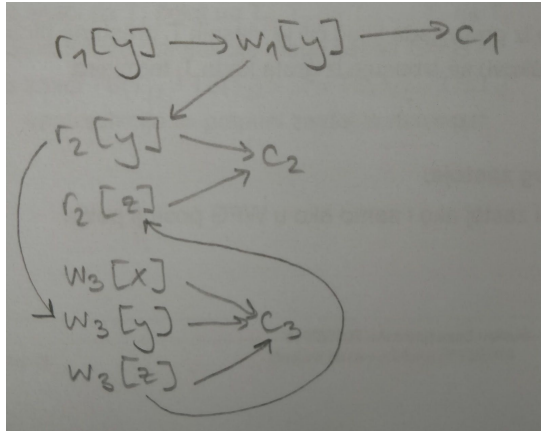
ZI 2013., 3. Zadatak

3. Povijest H_1 je prikazana u obliku topološkog poretka operacija transakcija T_1 , T_2 i T_3 :

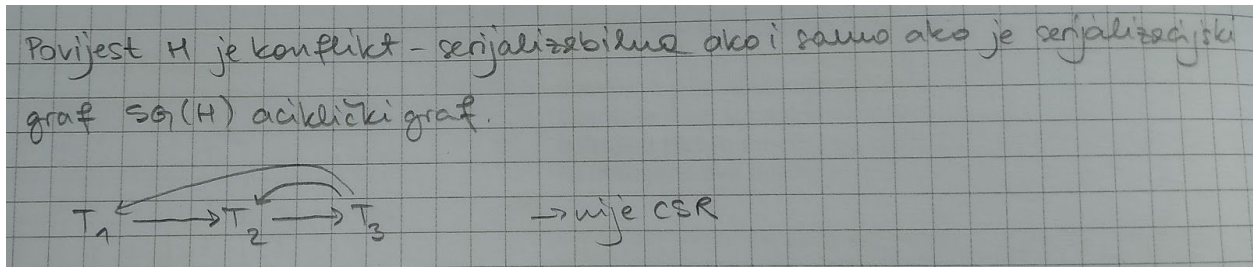
$H_1 \quad r_1[y] \quad w_1[y] \quad r_2[y] \quad w_3[x] \quad w_3[y] \quad c_1 \quad w_3[z] \quad c_3 \quad r_2[z] \quad c_2$

- Povijest H_1 prikazati u obliku grafa.
- Objasniti na koji se način utvrđuje je li povijest iz a) dijela zadatka konflikt-serijalizabilna.
- Koji se oblik karakterističnog problema istodobnog pristupa pojavljuje u povijesti H_1 ? Objasniti zašto se u povijesti H_1 pojavljuje taj problem.
- Uz pretpostavku da se povijest H_1 izvršava u sustavu koji koristi rigorozni 2PL protokol, u obliku niza operacija prikazati povijest koja će biti producirana (izvršena) u takvom sustavu.
- Uz pretpostavku da se povijest H_1 izvršava u sustavu koji koristi rigorozni 2PL protokol, ali u kojem je razina izolacije svih transakcija postavljena na READ UNCOMMITTED, u obliku niza operacija prikazati povijest koja će biti producirana (izvršena) u takvom sustavu. (9 bodova)

a)



b)



9:57 AM Yesterday

Resolve

Ovdje samo treba maknuti ovu strelicu iz T3 u T1; bez obzira na to, i dalje vrijedi da nije CSR

c) Pojavljuje se problem prljavog čitanja. On se javlja zato jer T2 čita vrijednost elementa y nakon što ju je T1 promijenila, a prije nego je T1 potvrđena.

Također se pojavljuje problem nekonzistente analize, jer T2 čita z nakon što ga T3 promijeni, a čita y prije nego ga T3 promijeni.

d)

Možda : $r1[y]$ $w1[y]$ $w3[x]$ $c1$ $w3[y]$ $w3[z]$ $c3$ $r2[y]$ $r2[z]$ $c2$ - ovo mi se čini dobro, u ostalim slučajevima imaš opasnost od izgubljene izmjene/prljavog čitanja?

I meni se čini dobro, jer tako je i meni ispalo po RIGOROZNOM 2PL-u

Možda2: $r_1[y] w_1[y] w_3[x] c_1 r_2[y] r_2[z] c_2 w_3[y] w_3[z] c_3$ -> T2 je prvi došao na čekanje, pa se prvi i pokreće nakon c_1 , a T3 je na čekanju do c_2 , jer ne može obaviti $w_3[y]$? (ovo dobijem ako radim pomoću one tablice sa predavanja (pred9, str 12))

Poredak gore nije dobar: u povijesti ima strelica iz $w_3[z]$ u $r_2[z]$, a u topološkom poretku se $r_2[z]$ izvršava prije $w_3[z]$.

Povijest bez upisanih postavljanja ključeva:

$r_1[y] w_1[y] w_3[x] c_1 r_2[y] w_3[z]$ potpuni zastoj: T2 ima S na y, T3 želi X na y; T3 ima X na z, T2 želi S na z.

Ali, pita te kakvu bi povijest producirao 2PL, a on može izmijeniti u nekim slučajevima povijest (pred 9, slajd 15), a i kod njih na 12 slajdu vidis da je poredak $r_1[x], w_2[x], r_3[x]$, a 2PL reproducira $r_1[x], r_3[x], w_2[x]$

Ali, ako se ne koriste U-ključevi, 2PL protokol bi trebao nakon $r_2[y]$ zatražiti X ključ na z od strane T3 i dobiti ga, ne zna on unaprijed da će doći do potpunog zastoja.

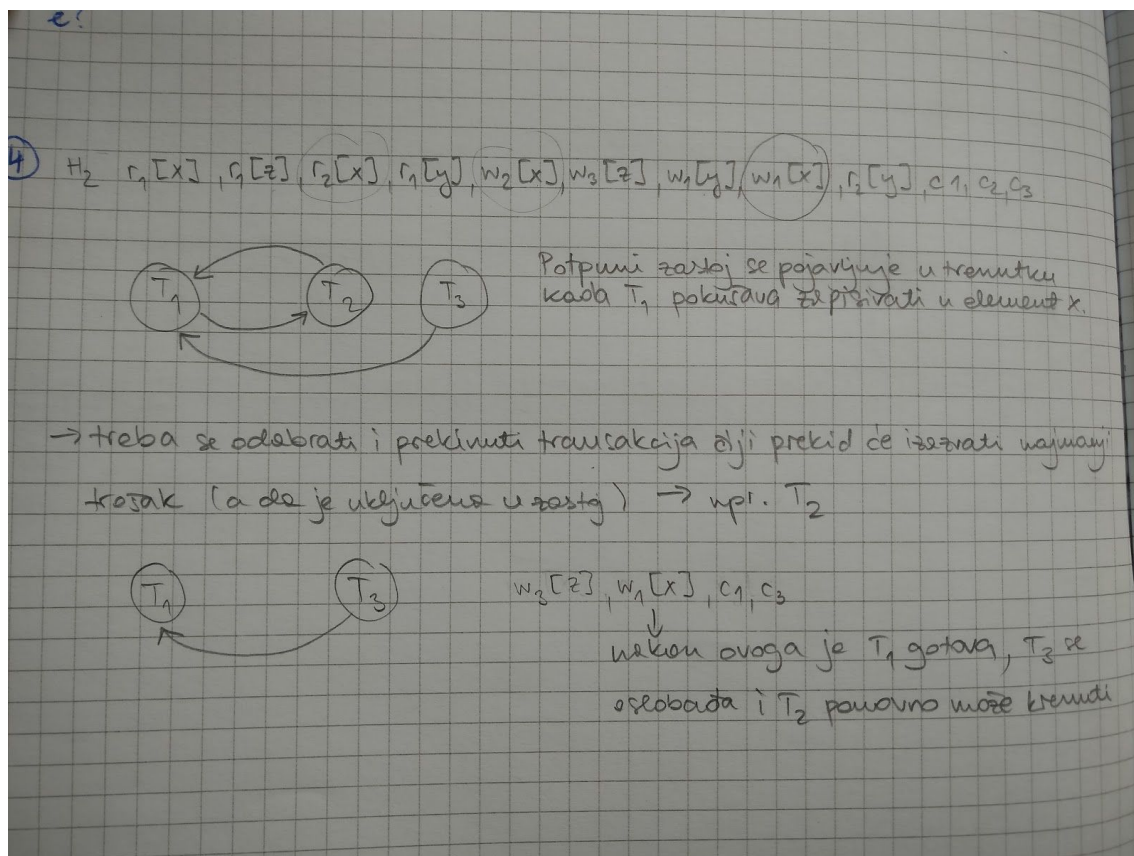
e) READ UNCOMMITTED - $r[x]$ ne postavlja S ključ i ne provjerava postoji li X ključ na x
 $r_1[y] w_1[y] r_2[y] w_3[x] c_1 w_3[y] w_3[z] c_3 r_2[z] c_2$

ZI 2013., 4. Zadatak

4. Povijest H_2 je prikazana u obliku topološkog poretka operacija transakcija T_1, T_2 i T_3 :

$H_2 \quad r_1[x] \ r_1[z] \ r_2[x] \ r_1[y] \ w_2[x] \ w_3[z] \ w_1[y] \ w_1[x] \ r_2[y] \ c_1 \ c_2 \ c_3$

Sustav koristi 2PL protokol. Navesti u kojem točno trenutku izvršavanjem povijesti H_2 dolazi do potpunog zastoja te opisati kako će se potpuni zastoj razriješiti ako se povijest H_2 izvršava u sustavu koji koristi graf čekanja (WFG graf). (5 bodova)



ZI 2013., 5. Zadatak

5. U bazi podataka kreirana je relacija *osoba* i indeks *idx*. U relaciju su upisane samo *n-torke* prikazane na slici (sadržaj relacije je važan). SUBP koristi MGL protokol (uz hijerarhiju objekata: *baza podataka* → *relacija* → *n-torka*) i protokol zaključavanja indeksa, ali samo ako su za korištenje protokola zaključavanja indeksa ostvareni potrebni preduvjeti. Ne koriste se U-ključevi.

```
CREATE TABLE osoba (
  sif INTEGER
  , ime CHAR(20)
  , prez CHAR(20)
) LOCK MODE ROW;

CREATE UNIQUE INDEX idx
ON osoba (sif);
```

sif	ime	prez
100	Ana	Horvat
103	Ivo	Ban
106	Jure	Ban
109	Ivo	Kolar
112	Jure	Novak
115	Ivo	Novak
118	Ana	Turk
121	Edo	Keler
125	Lino	Hertz

Redoslijed kojim SQL naredbe pristižu u sustav u skladu je s rednim brojevima naredbi. U sustav pristigla SQL naredba obavlja se odmah i u cijelosti (ako nije bilo zapreka za postavljanje potrebnih ključeva).

T_1	T_2
BEGIN WORK; SET LOCK MODE TO WAIT; SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	BEGIN WORK; SET LOCK MODE TO WAIT; SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
1. SELECT * FROM osoba WHERE sif=100	2. UPDATE osoba SET prez='Zu' WHERE sif=106
3. UPDATE osoba SET prez='Ho' WHERE sif=100	4. SELECT * FROM osoba WHERE sif=112
5. SELECT * FROM osoba WHERE sif=100	6. DELETE FROM osoba WHERE sif=118
7. INSERT INTO osoba VALUES(118, 'Jo', 'Wu')	8. SELECT * FROM osoba WHERE ime='Lino'

1. T_1 postavlja IS na bazu i relaciju *osoba* te S na zapis u indeksu koji pokazuje na sif=100 i *n-torku* sif=100

2. T2 postavlja IX na bazu i relaciju osoba te S na zapis u indeksu koji pokazuje na sif=106 i X na n-torku sif=106
3. T1 promovira IS u IX za bazu i relaciju, ima S na zapis u indeksu koji pokazuje na sif=100 i promovira S u X na n-torci sif=100
4. T2 ima IX na bazi i relaciji, postavlja S na zapis u indeksu koji pokazuje na sif=112 i n-torku sif=112
5. T1 ima IX na bazi i relaciji te S na zapisu u indeksu koji pokazuje na sif=100 i X n-torci sif=100
6. T2 ima IX na bazi i relaciji, postavlja X na zapis u indeksu koji pokazuje na sif=118 i n-torku sif=118
7. T1 ima IX na bazi i relaciji, želi postaviti X na zapis u indeksu koji pokazuje na sif=118 i n-torku sif=118, odbijeno zbog X na zapisu u indeksu sif=118 od T2 --> T1 čeka
8. T2 ima IX na bazi, želi SIX na relaciji osoba, odbijeno zbog IX od T1 --> T2 čeka ==> potpuni zastoј

ZI 2013., 6. Zadatak

Objasniti zašto je u tablici kompatibilnosti ključeva za MGL protokol u presjeku stupca IS i retka IX oznaka true, te zašto je u presjeku stupca IX i retka S oznaka false.

Presjek stupca **IS** i retka **IX** je true jer transakcija koja postavlja ključ **IX** neće nužno pokušati postaviti **X** ključeve u elementima podstabla koji otprije imaju postavljene **S** ključeve.

Presjek stupca **IX** i retka **S** je false jer bi **S** ključ pokušao postaviti **S** ključeve u sve elemente podstabla. IX nam govori da u podstablu postoji element na koji je postavljen ključ X zbog čega bi postavljanje S ključa na taj element bilo odbijeno.

ZI 2013., 7. Zadatak

Sustav s potpuno repliciranom bazom podataka obuhvaća čvorove S1, S2, ..., S5.

Odrediti parametre za quorum consensus protokol uz koje će taj protokol djelovati kao:

a) protokol zaključavanja uz pomoć primarne kopije koja se uvijek nalazi u čvoru S5

b) većinski protokol (majority protocol)

a) $w_1..w_4=1, w_5=5, \text{ suma}=9$

$Q_r = 5, Q_w = 5$

b) $w_1..w_5=1, \text{ suma} = 5$

$Q_r = 3, Q_w = 3$

ZI 2012., 1. Zadatak

1. Na slici su prikazani grafovi transakcija T_1 i T_2 .



- Nacrtati graf povijesti H_1 koja obuhvaća T_1 i T_2 i čije bi izvršavanje (kad bi se dopustilo) izazvalo nekonzistentnu analizu.
- Što je to prljavo čitanje? Ako je u H_1 moguća pojava prljavog čitanja, navesti koje ga točno operacije i zbog čega mogu uzrokovati?
- U općem slučaju, koje od svojstava povijesti (RC, ACA, ST) sprečava prljavo čitanje? Objasniti zašto. (6 bodova)

ZI 2012

a) neko vrsta anket

b) Priljava čitanje je anomalija u kojoj T_1 čita element x ili je vrijednost mijenjala transakcija T_2 , koja još nije potvrđena.

Primer: T_1 mijenja element z , a T_2 čita tu vrijednost, prije nego je T_1 napravila potvrdu

```

graph LR
    T1((T1)) --> w1x[w1[x]]
    T1 --> w1z[w1[z]]
    T1 --> r1y[r1[y]]
    T2((T2)) --> r2x[r2[x]]
    T2 --> r2z[r2[z]]
    T2 --> w2y[w2[y]]
    r1y --> w2y
  
```

U a) dijelu treba dodati luk između $r1[y]$ i $w2[y]$ (ja i frend gledali taman pa se složili da fali)

c) ACA sprječava prljavo čitanje (to svojstvo nalaže da T_2 ne smije čitati stvari koje je mijenjala T_1 dok T_1 nije potvrđena, pa mi je nekako logično, plus to implicira i na ST).

Moje rješenje za c):

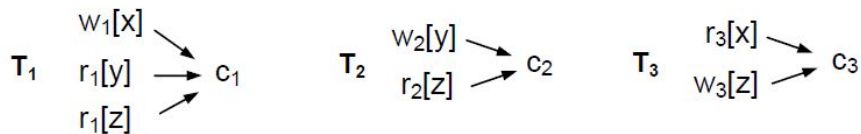
RC sprječava prljavo čitanje jer transakcija smije biti potvrđena tek nakon što su potvrđene sve transakcije iz kojih je čitala, a to zapravo znači da prema RC-u sve ide kaskadno => ne bi se moglo dogoditi prljavo čitanje jer bi se transakcije kaskadno poništile (43. Slajd pod primjerom)

ACA sprječava prljavo čitanje jer neka transakcija ne može čitati one vrijednosti transakcije koje nisu potvrđene, a moći će čitati tek kad bude potvrđena transakcija zbog koje nije mogla čitati.

ST sprječava prljavo čitanje jer neka transakcija T ne može ni čitati ni pisati neki element sve dok sve transakcije koje su pisale u njega ne budu terminirane (abort/commit).

ZI 2012., 2. Zadatak

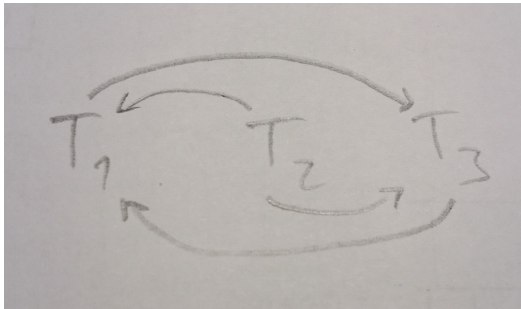
2. Na slici su prikazani grafovi transakcija T_1 , T_2 i T_3 .



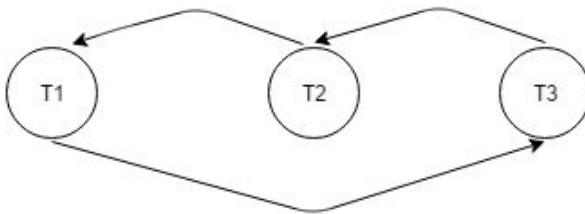
- U obliku niza operacija (ne u obliku grafa) napisati povijest H_2 koja će dovesti do potpunog zastoja u kojem će sudjelovati sve tri transakcije.
- Nacrtati WFG graf u trenutku nastanka potpunog zastoja.
- Što će sustav poduzeti u cilju razrješavanja potpunog zastoja i kako WFG izgleda u trenutku neposredno nakon razrješenja potpunog zastoja?
- Ispitati je li povijest H_2 konflikt-serijalizabilna. (5 bodova)

a) $r_3[x] w_1[x] r_1[z] w_3[z] r_1[y] w_2[y] r_2[z] c_1 c_2 c_3$

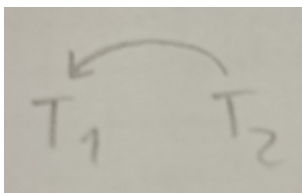
Drugi odgovor: $r_1[y] r_2[z] r_3[x] w_2[y] w_3[z] w_1[x] r_1[z] c_1 c_2 c_3$



b)



- Poništiti će transakciju koja će mu omogućiti minimalan trošak, npr T_3



- Ova povijest nije konflikt-serijalizabilna jer serijalizacijski graf ima ciklus

ZI 2012., 3. Zadatak

Što to znači da je 2PC protokol blokirajući protokol? U kojem se trenutku i zašto dešava blokiranje 2PC protokola?

2PC je blokirajući protokol, što znači da neki čvor (TC ili TM) neće moći izvršiti transakciju ukoliko se dogodila pogreška ili kvar na nekom drugom čvoru. To se događa kada koordinator čeka odgovor od ostalih čvorova jesu li spremni za početak posla ili kod trenutka kada iščekuje globalnu potvrdu transakcije.

Preciznije: blokiranje se događa u trenutku kada npr. TM prema TC pošalje poruku *spreman* i nalazi se u stanju čekanja na odluku TC o sudbini globalne transakcije. U tom trenutku dogodi se kvar na vezi prema TC. TM tada ostaje blokiran do ponovne uspostave veze sa TC-om i njegove odluke o nastavku transakcije.

ZI 2012., 4. Zadatak

4. Potpuno replicirani sustav obuhvaća čvorove S_1 i S_2 . Shema baze podataka sadrži shemu samo jedne relacije. Shema i sadržaj relacije u trenutku t_0 su prikazani na slikama.

```
CREATE TABLE zgrada (  
    sif      INTEGER  
    , visina INTEGER  
    , PRIMARY KEY (sif));
```

U trenutku t_0 baza podataka je konzistentna: sve fizičke kopije logičkih elemenata su međusobno jednake.

sif	visina
10	22
11	50
12	22

- a) Uz pretpostavku korištenja asinkrone dvosmjerne replikacije, napisati jedan primjer niza akcija koje će korisnici i sustav obaviti nakon trenutka t_0 , a koji će dovesti do nekonzistentnosti (*system delusion*), odnosno konflikta kojeg će morati razriješiti čovjek.
- b) Objasniti kako bi primjena sinkrone (*eager*) dvosmjerne replikacije sigurno spriječila neželjeni slijed događaja iz a) dijela zadatka. (3 boda)

- a) S_1 - SELECT * FROM zgrada; DELETE FROM zgrada WHERE sif = 10; (T1)
 S_2 - SELECT * FROM zgrada; UPDATE zgrada SET visina= 28 WHERE sif = 10; (T2)

- b) Unutar jedne (nedjeljive) transakcije bi se promijenile sve kopije na svim čvorovima.

ZI 2012., 5. Zadatak

Objasniti zašto je u tablici kompatibilnosti ključeva za MGL protokol u presjeku stupca IX i retka IS oznaka true, te zašto je u presjeku stupca SIX i retka IX oznaka false.

U tablici kompatibilnosti presjek stupca IX i IS je true, jer kada T1 postavi IX ključ nad relacijom, znači da će nad nekim n-torkama u relaciji postaviti X ključ, a ne nad svima, tako da se smije postaviti IS ključ na relaciju od strane T2 jer ne mora nužno postaviti S ključ nad n-torkama gdje je postavljen ključ X, već nad nekim drugim n-torkama.

Presjek SIX stupca i IX retka je false jer je postavljen S ključ nad n-torkama pri čemu bilo koja izmjena od strane druge transakcije nije dopuštena.

ZI 2012 6.

6. U bazi podataka kreirana je relacija *osoba* i indeks *idx*. U relaciju su upisane samo n-torke prikazane na slici (sadržaj relacije je važan). SUBP koristi MGL protokol (uz hijerarhiju objekata: *baza podataka* → *relacija* → *n-torka*) i protokol zaključavanja indeksa, ali samo ako su za korištenje protokola zaključavanja indeksa ostvareni potrebni preduvjeti. Ne koriste se U-ključevi.

```
CREATE TABLE osoba (
  sif INTEGER
, ime CHAR(20)
, prez CHAR(20)
) LOCK MODE ROW;

CREATE UNIQUE INDEX idx
ON osoba (sif);
```

sif	ime	prez
100	Ana	Horvat
103	Ivo	Ban
106	Jure	Ban
109	Ivo	Kolar
112	Jure	Novak
115	Ivo	Novak
118	Ana	Turk
121	Edo	Keler

Redosljed kojim SQL naredbe pristižu u sustav u skladu je s rednim brojevima naredbi. U sustav pristigla SQL naredba obavlja se odmah i u cijelosti (ako nije bilo zapreka za postavljanje potrebnih ključeva).

T ₁	T ₂
BEGIN WORK; SET LOCK MODE TO WAIT; SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	BEGIN WORK; SET LOCK MODE TO WAIT; SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
1. SELECT * FROM osoba WHERE sif=100	2. SELECT * FROM osoba WHERE sif=100
3. SELECT * FROM osoba WHERE sif=118	4. UPDATE osoba SET prez='Li' WHERE sif=112
5. UPDATE osoba SET prez='Ho' WHERE sif=118	6. DELETE FROM osoba WHERE sif=106
7. INSERT INTO osoba VALUES(106, 'Jo', 'Wu')	8. SELECT * FROM osoba WHERE sif=106
9. SELECT * FROM osoba	10. SELECT * FROM osoba

- a) Za svaku naredbu (1-10) koja se uspije obaviti prije nego transakcija uđe u stanje čekanja na postavljanje ključa, napisati kojom vrstom ključa se pokušava zaključati koji element baze podataka. Ako zaključavanje nije uspjelo navesti i razlog zašto. Rješenje treba biti u obliku prikazanom u priloženom primjeru rješenja.
- b) Ponovo riješiti zadatak pod a), ali ovog puta uz pretpostavku da nad relacijom *osoba* nije kreiran indeks *idx*.

Primjer oblika rješenja:

- T₁ postavlja S na n-torke 100, 102, 102, postavlja X na n-torke 105, 107
- T₂ postavlja IS na n-torke 103, 104, X na osoba
- T₁ promovira S u IX na n-torci 102
- T₂ ne uspijeva postaviti S na bazu podataka jer je T₂ već prije s X zaključala relaciju, T₂ čeka
- T₁ uspješno promovira S u X nad bazom podataka

(6 bodova)

Rješenje:

- a) BP - REL - IDX - NTORKA

- IS na BP, IS na rel, S na zapis idx sif = 100, S na ntorku
- IS na BP, IS na rel, S na zapis idx sif = 100, S na ntorku
- IS ima, IS ima, S na zapis idx sif = 118, S na ntorku
- IS -> IX, IS -> IX, S na zapis idx sif = 112, X na ntorku
- IS -> IX, IS -> IX, S na zapis idx ima, S -> X na ntorku
- IX ima, IX ima, X na zapis idx sif = 106, X na ntorku
- IX ima, IX ima, X na zapis indeksa T1 ČEKA

8 - IS na b.p. i relaciju (već ima IX), S na index sa zapisom sif = 106 (već ima X),
S na n-torku sif = 106 (već ima X)

9 - čeka

10 - IS na b.p.(već ima IX), IX -> SIX na relaciju (me može zbog ključa IX
transakcije T1), čeka; potpuni zastoj

b) BP - REL - NTORKA

Operacija unosa i sablasna n-torka

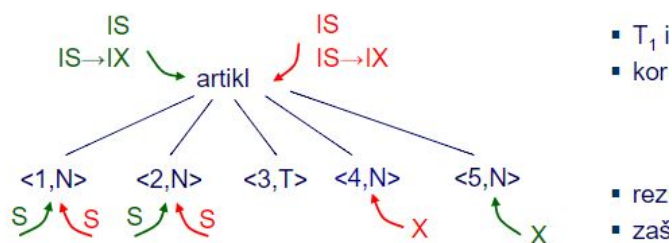
Primjer: ▪ ograničenje: ne smije postojati više

artikl

sif	grupa
1	N
2	N
3	T

T₁

```
1. SELECT COUNT(*) INTO v
   FROM artikl
   WHERE grupa = 'N';
5. if v < 3
   INSERT INTO artikl
   VALUES (5, 'N');
6. COMMIT;
```



3 komm: zato jer je to primjer pojavljivanja sablasne n-torke. Tu vidiš da ako staviš IS na relaciju, a nemaš index, onda ti se stvari počnu raspadati. HVALA

2 komm: Zašto onda ovdje gore nije odmah na relaciju stavljen S nego IS ?

^

|

1 komm: Ovo gore nije dobro, kada nemaš indeks onda kod čitanja moraš zaključati cijelu relaciju, nema zaključavanja jedne po jedne n-torke. Pogledaj prošlogodišnji

završni, ima službeno rješenje ovakvog tipa. Kao i uvijek, **molim kolege da kontroliraju da ne pričam/pišem gluposti!!!**

- 1 - T1 IS na b.p. S na relaciju
- 2 - T2 IS na b.p. S na relaciju
- 3 - T1 IS na b.p. već ima, S na relaciju već ima
- 4 - T2 IS -> IX na b.p., S -> SIX na relaciju (ne može zbog ključa S transakcije T1), čeka
- 5 - T1 IS -> IX na b.p., S -> SIX na relaciju (ne može zbog ključa S transakcije T2), čeka; potpuni zastoj

Možemo li čitati neku vrijednost u relaciji koja je označena s X ili moramo promovirati u SIX ? Jel SIX ide samo kada S i X idu na različite elemente? **SIX ide kada čitaš sve elemente a mijenjaš/pišeš samo neke. Ako neke čitaš a neke brišeš onda ide IX** HVALA
Slaven je turbo zgodan - Agreed
gerontofil/filka?

Ukratko:

Izvorni post od **MauSonja** »

Jel bi moglo neko napisati za zadatke sa kvorumima kak se točno računaju Q_r i Q_w za sve protokole zaključavanja jer iz prezentacija se baš i neda to zaključiti?

u zadacima većinom imaš zadan određen broj čvorova

ako imamo čvorove $S_1 \dots S_5$ a želimo:

a) većinski protokol (moraju se zaključati kopije elementa x u više od polovice čvorova u kojima je x alociran, operacija nad elementom x se ne obavlja sve dok se ne dobije ključ za većinu kopija elementa x)

znači težine po čvorovima bi bile:

Source kod:

S_1	S_2	S_3	S_4	S_5
1	1	1	1	1

za pisanje i za čitanje mora biti element zaključan u većini čvorova ($>$ pola ukupnog zbroja težina)

za ovo je $Q_r = 3$ i $Q_w = 3$

b) pristrani protokol - prednost se daje operacijama čitanja (protokol je "pristran")

- za obavljanje operacije čitanja dovoljno je postaviti S -ključ nad jednom kopijom
- za obavljanje operacije pisanja X -ključevi se moraju postaviti na svim kopijama

znači težine po čvorovima bi bile jednake:

Source kod:

S_1	S_2	S_3	S_4	S_5
1	1	1	1	1

samo bi $Q_r = 1$, a $Q_w = 5$

c) primarna kopija - Protokol usuglašavanja kvorumom (moraju vrijediti pravila : $Q_r + Q_w > \text{suma}$ i $Q_w + Q_w > \text{suma}$)

znači težine po čvorovima bi bile:

Source kod:

S_1	S_2	S_3	S_4	S_5
1	1	1	1	5

S_5 ima najveći težinu jer je tamo smještena primarna kopija, te se kod zaključavanja uvijek mora ona zaključati (mogu se i neke druge uz nju al nije potrebno)
sada je $Q_r = 5$, a $Q_w = 5$

TEORIJA

<http://www.fer2.net/showpost.php?p=2179842&postcount=34>

Izvršavanje koje (kakve?) povijesti vodi do izgubljene izmjene? RC

SUDJELOVALI: Ivyclover, zoxparty, Vronski, tamagoĆi, Wizzle, zackfair93, Houndolon, Legenda.