1. Napisati pohranjenu proceduru bez ulaznih argumenata koji će na temelju sadržaja tablice *param* u tablicu *test* upisati podatke za testiranje. Primjer mogućeg sadržaja tablice param prikazan je na slici ispod.

Tablica *test* na početku je prazna.

Za svaki zapis <brZapisa, tekst> tablice param u tablicu test se upisuje brZapisa čiji je sadržaj određen s tekst te briše pripadni zapis iz param. Ne smije se dogoditi da neki zapis iz param bude obrisan, a da pripadni zapisi nisu upisani u test, niti obrnuto. Međutim, ako se za neki zapis iz param uspješno obavi unos predviđenog broja zapisa u test i taj zapis uspješno obriše iz param, tada se promjene više ne smiju izgubiti.

Ako se iz brisanja iz param dogodi pogreška bilo koje vrste, u pozivajući program dojaviti pogrešku s tekstom „Pogreška brisanja“. U slučaju pojave bilo koje druge pogreške na bilo kojem drugom mjestu u proceduri, procedura u pozivajući program mora proslijediti originalnu pogrešku.

Primjer: procedura nakon pokretanja za sadržaj na slici obavlja sljedeće:

* U tablicu test upiše 50 zapisa „Ana“ i obriše zapis „Ana“ iz param
* U tablicu test upiše 25 zapisa „Ivo“ i obriše zapis „Ivo“ iz param
* U tablicu test uspije upisati samo 12 zapisa „Pero“ i tada se dogodi pogreška

Procedura u ovom slučaju treba završiti dojavom originalne pogreške, u test se po završetku procedure treba nalaziti 50 zapisa „Ana“, 25 zapisa „Ivo“, niti jedan zapis „Pero“, a param treba sadržavati sve preostale zapise osim zapisa „Ana“ i „Ivo“.

Nije dozvoljena uporaba okidača, pomoćnih relacija i procedura niti SAVEPOINT mehanizma.

|  |
| --- |
| CREATE TABLE test(  Ime CHAR(20)  ); |

|  |
| --- |
| CREATE TABLE param (  brZapisa INTEGER,  tekst CHAR(20),  PRIMARY KEY (tekst)  ); |

Tablica param:

|  |  |
| --- | --- |
| brZapisa | tekst |
| 50 | Ana |
| 25 | Ivo |
| 40 | Pero |
| . | . |
| . | . |
| 70 | Sara |

2. Zadane su relacije r(A, B, C) i s(D, C, E, G). Primarni ključevi su podcrtani. Nema indeksa. Na raspolaganju je 1000 blokova primarne memorije. Svi međurezultati se materijaliziraju. Za operaciju prirodnog spajanja koristi se nested loop join. Ne treba provoditi herističku optimizaciju.

Q B NOT LIKE '%m' (r) >< Q E ≤ 600 v G = 20(s)

|  |
| --- |
| V(c,r) = 200 |
| V(c,s) = 800 |
| V(E,s) = 100 |
| V(G,s) = 4 |

|  |
| --- |
| N(r) = 2000 |
| N(s) = 10 000 |

|  |
| --- |
| B(r) = 1000 |
| B(s) = 6000 |

|  |
| --- |
| Min(C,r) = 0 |
| Max(C,r) = 1000 |
| Min(C,s) = 0 |
| Max(C,s) = 1000 |
| Min(E,s) = 500 |
| Max(E,s) = 1000 |
| Min(G,s) = 10 |
| Max(G,s) = 40 |

Veličina n-torke(r) = 0.4 blokova

Veličina n-torke(s) = 0.5 blokova

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

1. Prvo procjeniti broj n-torki u međurezultatima i konačnom rezultatu
2. Zatim procjeniti ukupan broj U/I operacija tijekom izvršavanja operacije relacijske algebre i broj blokova u konačnom rezultatu

3. Objasniti zašto i u kojim situacijama SUBP koristi protokol zaključavanja indeksa (Index locking protocol)

4. 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 ostvareni preduvjeti. Ne koriste se U-ključevi.

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)

|  |  |
| --- | --- |
| T1 BEGIN WORK;  SET LOCK MODE TO WAIT  SET TRANSACTION ISOLATION LEVEL  1. SELECT \* FROM osoba WHERE sif=100  3. DELETE FROM osoba WHERE sif=106;  5. UPDATE osoba SET prez='Bo' WHERE ime='Ivo' | T1 BEGIN WORK;  SET LOCK MODE TO WAIT  SET TRANSACTION ISOLATION LEVEL  2. INSERT INTO osoba VALUES (130, 'Jo', 'Wu')  4. SELECT \* FROM osoba WHERE sif=100  6. SELECT \* FROM osoba WHERE prez='Wu' |

Za svaku naredbu (1-6) koja se pokuša 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 primjeru rješenja.

|  |
| --- |
| CREATE TABLE osoba(  sif INTEGER,  ime CHAR(20),  prez CHAR(20),  ) LOCK MODE ROW |

|  |
| --- |
| CREATE UNIQUE INDEX idx ON osoba (sif) |

Tablica osoba:

|  |  |  |
| --- | --- | --- |
| **sif** | **ime** | **prez** |
| 100 | Ana | Horvat |
| 103 | Ivo | Ban |
| 106 | Jure | Toth |
| 109 | Pero | Kolar |
| 112 | Dino | Hesse |
| 115 | Tea | Novak |
| 118 | Nives | Turk |
| 121 | Edo | Keler |

|  |
| --- |
| Primjer oblik rješenja:   1. T1 postavlja S na osobu, IS na n-torke 100, 101, 102, postavlja X na n-torke 105, 107 2. T2 postavlja IS na n-torke 103, 104, X na osoba 3. T1 promovira S u IX na n-torci 102 4. T2 ne uspjeva postaviti S na bazu podataka jer je T2 već prije X ključem zaključala relaciju, T2 čeka. |

5.

r1[u]

w1[y]

w1[z]

c1

r1[x]

w1[x]

c2

w2[x]

r2[z]

r2[y]

T2

T1

1. Nacrtati graf povijesti H1 koja obuhvaća T1 i T2 i čije bi izvršavanje (kad bi se dopustilo) izazvalo problem nekonzistentnosti analize, ali ne i problem izgubljene izmjene
2. Nacrtati graf povijesti H2 koja obuhvaća T1 i T2 i čije bi izvršavanje (kad bi se dopustilo) izazvalo problem izgubljene izmjene, ali ne i problem nekonzistentnosti analize
3. Nacrtati graf povijesti H3 koja obuhvaća T1 i T2 čije bi izvršavanje izazvalo prljavo čitanje, ali ne i problem nekonzistentnosti analize niti problem izgubljene izmjene
4. Zasebno za svaku od povijesti H1, H2, H3 navesti najnižu ANSI SQL razinu izolacije pri kojoj bi SUBP uspio spriječiti neserijalizirano izvršavanje dotične povijesti

6. a) Za povijest H koja je prikazana u obliku topološkog poretka operacija transakcija, odrediti radi li se o RC (recoverable) i/ili ACA (Avoids cascading aborts) i/ili ST povijesti.

|  |
| --- |
| H: w1[x] w2[y] a1 r3[x] r2[z] r3[y] c2 r3[z] c3 |

Odogovor mora biti u obliku: „Jest RC zato jer..., nije ACA zato jer,... jest ST zato jer...

b) Koji problem bi se mogao pojaviti ako bi sustav dopustio izvršavanje neke povijesti koja jest RC i ACA, ali nije ST?

7. Sustav s potpuno repliciranom bazom podataka obuhvaća čvorove S1, S2,...S8.

a) Odrediti parametre za quorum consensus protokol uz koje će se S i X ključeve tretirati ravnopravno, kao kod većinskog protokola (majority protocol), ali će za zaključavanje elemenata X biti dovoljno zaključati:

- ili elemente X u čvoru S8 i jednom (bilo kojom) od čvorova S1-S7

- ili elemente X u svim čvorovima S1-S7

b) Objasni pojam logičkog elementa, što je zaključavanje logičkog elementa i zašto izrazi (formule) koje se primjenjuju za određivanje parametra quorum consensus protokola (u općem slučaju) garantiraju ispravno zaključavanje logičkog elementa.

***Teorija na zaokruživanje (***točno prepisani zadaci, ali negdje bez ponuđenih odgovora, fali još jedan teorijski***)***

1. Upotreba kontrolne točke (check point):
   1. –
   2. –
   3. –
   4. –
   5. –

(Odgovor s fer2.net: manje posla kod obnove (manje trans potrebno obaviti ako je čest checkpoint))

1. Kako bi spriječio problem do kojeg može dovesti pojava dugih transakcija u sustavi IBM Informix IDS, admin sustava za upravljanje baze podataka će:
   1. –
   2. –
   3. –
   4. –
   5. –

(Odgovor s fer2.net: povećati veličinu dnevnika)

1. Za obnovu sustava će se morati koristiti oboje, dnevnik (log) i arhivska kopija BP ako se dogodi:
   1. Pogreška transakcija i sustava
   2. Pogreška sustava
   3. Pogreška sustava i blokiranje 2PC protokola
   4. Pogreška sustava i pogreška medija
   5. Pogreška medija

Odgovor: pred7, slajd 19, možda e), kvar medija?

1. Korektna transakcija:
   1. –
   2. –
   3. –
   4. –
   5. –
2. Po zaključivanju na više razina granulacije, ključevi upozorenja (warning locks, intention locks) koriste se radi:
   1. –
   2. –
   3. –
   4. –
   5. –

(Odgovor s fer2.net:  lakše provjeravanje kompatibilnosit kod zaključavanja)

1. SUBP obavlja operaciju SELECT c FROM r WHERE b=5000. Relacije r sadrži 10 000 n-torki. Pri tome, vrijedi V(a,r) = 3, V(b,r) = 9900. Postoji samo jedan složeni (kompozitni) indeks nad atributima (a,b) relacije r. Optimizator je dobro obavio svoju zadaću ako u planu izvršavanja za dotičnu naredbu koristi:
   1. Index scan
   2. Table scan
   3. Key-only index scan
   4. Autoindex scan
   5. Index self-join

(Odgovor s fer2.net: koristi se index self join)

1. Kada se ne koristi LRU:
   1. -
   2. –
   3. –
   4. –
   5. –

(Odgovor s fer2.net: čitanje velike relacije)