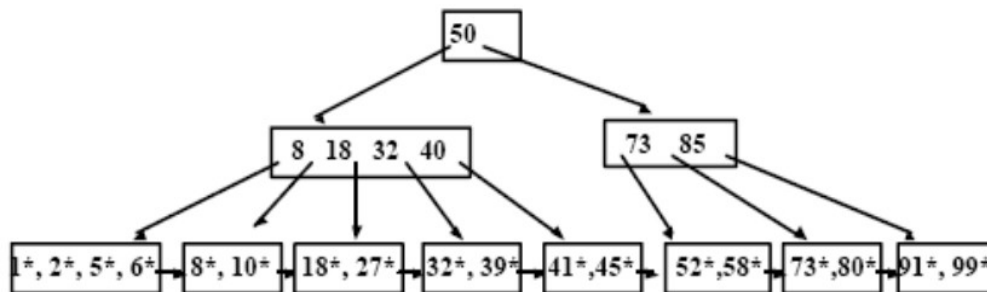


Μάθημα “Βάσεις Δεδομένων ΙΙ” - Ασκήσεις Επανάληψης

01 Εκφώνηση

Θεωρείστε το παρακάτω B+tree (κάθε κόμβος ευρετηρίου χωρά 4 καταχωρίσεις ευρετηρίου και κάθε κόμβος φύλλο χωρά 4 καταχωρίσεις δεδομένων):



(A) Ποιο είναι το μικρότερο πλήθος διαγραφών που πρέπει να γίνουν για να μειωθεί το ύψος του δέντρου κατά 1; Δώστε μια τέτοια ακολουθία διαγραφών.

(B) Ποιο είναι το μικρότερο πλήθος εισαγωγών που πρέπει να γίνουν ώστε ο κόμβος με τις τιμές 73 και 85 να γεμίσει; Δώστε μια τέτοια ακολουθία εισαγωγών;

01 Λύση

(A) Υποθέτουμε ότι κατά τη διαγραφή πρώτα γίνεται προσπάθεια ανακατανομής με το αριστερό ή το δεξιό αδέρφι κατά σειρά, και μετά συγχώνευσης με το αριστερό ή το δεξιό αδέρφι κατά σειρά.

(α) διαγραφή του 18*, συγχώνευση με τον αριστερό κόμβο, νέος κόμβος (8*, 10*, 27*), διαγραφή του 18 από τον κοινό γονιό τους.

(β) διαγραφή του 41*, συγχώνευση με τον αριστερό κόμβο, νέος κόμβος (32*, 39*, 45*), διαγραφή του 40 από τον κοινό γονιό τους. Ο γονιός τώρα είναι ο (8, 32).

(γ) διαγραφή του 80*, συγχώνευση με τον αριστερό κόμβο, νέος κόμβος (52*, 58*, 73*), διαγραφή του 73 από τον κοινό γονιό τους. Ο γονιός γίνεται (85) πέφτει κάτω από το 50% οπότε συγχωνεύεται με το αδέρφι του (8, 32) καθώς ανακατανομή δεν μπορεί να γίνει. Η ρίζα δεν έχει λόγο ύπαρξης καθώς μένει με ένα παιδί, οπότε το 50 κατεβαίνει και δημιουργείται η νέα ρίζα (8, 32, 50, 85) με πέντε παιδιά.

(B) Για να γεμίσει ο κόμβος αυτός πρέπει να δεχτεί δυο καταχωρίσεις ευρετηρίου. Αυτό θα συμβεί όταν γίνουν δυο διασπάσεις στα παιδιά του (δηλαδή αποκτήσει άλλα δυο παιδιά). Εισάγουμε τα 75* και 79* στον (73*, 80*), ο οποίος υπερχειλίζει και διασπάται στους (73*, 75*, 79*) και (80*, 83*) [υποθέτουμε ότι κατά τη διάσπαση περιττού αριθμού καταχωρίσεων η έξτρα καταχώριση πηγαίνει αριστερά]. Ανεβαίνει το 80 στο γονιό και τώρα είναι ο (73, 80, 85). Συνεχίζουμε με εισαγωγή των 76* και 78* στον (73*, 75*, 79*), ο οποίος υπερχειλίζει και διασπάται στους (73*, 75*, 76*) και (78*, 79*). Ανεβαίνει το 78* στο γονιό ο οποίος τώρα γίνεται (73, 78, 80, 85).

02 Εκφώνηση

Ένα B+-tree έχει εσωτερικούς κόμβους και κόμβους-φύλλα που η μέγιστη χωρητικότητα τους είναι 2 κλειδιά αναζήτησης και 3 δείκτες. Σχεδιάστε το τελικό B+-tree που προκύπτει από τη μαζική φόρτωση των τιμών 3, 30, 44, 2, 14, 4, 20 σε ένα άδειο τέτοιο δέντρο.

02 Λύση

Ταξινομούμε τις καταχωρίσεις δεδομένων: 2*, 3*, 4*, 14*, 20*, 30*, 44*. Δημιουργούνται 4 κόμβοι φύλλα: (2*, 3*), (4*, 14*), (20*, 30*), (44*). Προσθέτουμε σταδιακά από τα αριστερά προς τα δεξιά τα φύλλα στο κενό δέντρο. Τα τρία πρώτα φύλλα ευρετηριοποιούνται από ένα νέο γονιό που γίνεται και η ρίζα του δέντρου (4, 20). Η προσθήκη του τέταρτου φύλλου, διασπά τη ρίζα στους κόμβους (4) και (44) και δημιουργεί τη νέα ρίζα (20).

03 Εκφώνηση

Υποθέστε ότι μόλις κατασκευάσατε ένα πυκνό B+ tree Εναλλακτικής (2) πάνω σε ένα αρχείο σωρού για έναν πίνακα με 20000 εγγραφές. Το πεδίο αναζήτησης του B+ tree είναι αλφαριθμητικό με μέγεθος 40 bytes και αποτελεί υποψήφιο κλειδί του πίνακα. Οι pointers (δηλαδή τα *pid* στις καταχωρίσεις δεδομένων στα φύλλα του B+ tree καθώς και οι διευθύνσεις των σελίδων του B+ tree στις καταχωρίσεις ευρετηρίου) είναι 10 bytes. Το μέγεθος μια σελίδας του δίσκου είναι 1000 bytes. Το ευρετήριο κατασκευάστηκε με τον αλγόριθμο μαζικής φόρτωσης από κάτω προς τα επάνω και οι κόμβοι στο κάθε επίπεδο γέμισαν στο μέγιστο βαθμό (100%).

- (A) Πόσα επίπεδα έχει το δέντρο και πόσους κόμβους έχει το κάθε επίπεδο;
- (B) Πόσα επίπεδα θα είχε το δέντρο και πόσους κόμβους θα είχε το κάθε επίπεδο αν οι κόμβοι ήταν 70% γεμάτοι;

03 Λύση

Το μέγεθος μιας καταχώρισης δεδομένων είναι 50 bytes (μέγεθος τιμής κλειδιού αναζήτησης + μέγεθος pointer σε σελίδα του δίσκου). Αφού έχουμε Εναλλακτική (2), όλες οι καταχωρίσεις δεδομένων έχουν το ίδιο (σταθερό) μέγεθος. Άρα, σε κάθε σελίδα φύλλο, όταν έχει πληρότητα 100% χωράνε $1000/50=20$ καταχωρίσεις δεδομένων.

(A) Έχω 20000 εγγραφές στον πίνακά μου, άρα θα χρειαστώ 20000 καταχωρίσεις δεδομένων, δηλαδή $20000/20=1000$ σελίδες φύλλα (επίπεδο 1). Αντίστοιχα, με την ίδια λογική, καθώς μια καταχώριση ευρετηρίου θα έχει μέγεθος πάλι 50 bytes, κάθε κόμβος του επιπέδου 2 θα χωρά 20 καταχωρίσεις ευρετηρίου όταν έχει πληρότητα 100% και θα χρειαστώ 1000 καταχωρίσεις ευρετηρίου στο επίπεδο 2 για τα 1000 φύλλα. Άρα, θα έχω $1000/20=50$ κόμβους στο επίπεδο 2. Τέλος, θα χρειαστώ 3 κόμβους στο επίπεδο 3 για να αποθηκεύσω τις 50 καταχωρίσεις ευρετηρίου για τους 50 κόμβους του 2ου επιπέδου, και έναν κόμβο ρίζα στο 4ο επίπεδο που θα έχει 3 παιδιά.

(B) Το μόνο που αλλάζει είναι το πλήθος καταχωρίσεων που αποθηκεύει κάθε κόμβος, που τώρα είναι $\text{ceil}(1000/50*70\%)=14$. Άρα, στο επίπεδο 1 θα χρειαστώ $\text{ceil}(20000/14)=1429$ κόμβους, στο επίπεδο 2 θα χρειαστώ $\text{ceil}(1429/14)=103$ κόμβους, στο επίπεδο 3 θα χρειαστώ $\text{ceil}(103/14)=8$ κόμβους, και στο επίπεδο 4 τη ρίζα που θα έχει 8 παιδιά.

04 Εκφώνηση

Έχουμε τους πίνακες $\pi_1(A,B,C)$, $\pi_2(C,D,E)$ και $\pi_3(E,F)$ με κύρια κλειδιά τα A, C και E αντίστοιχα. Έστω ότι ο π_1 έχει 100 εγγραφές, ο π_2 έχει 150 εγγραφές και ο π_3 έχει 75 εγγραφές. Πόσες εγγραφές θα έχει το αποτέλεσμα της τριπλής σύζευξης των πινάκων;

04 Λύση

Η σύζευξη του π_1 με τον π_2 θα γίνει στο κοινό πεδίο C (κλειδί του π_2 , ξένο κλειδί στον π_1). Αντίστοιχα, η σύζευξη του αποτελέσματος με τον π_3 θα γίνει στο κοινό πεδίο E (κλειδί του π_3 , ξένο κλειδί στον π_2 και κατ' επέκταση στο $\pi_1 \mid x \mid \pi_2$).

Άρα, το $\text{temp1} = \pi_1 \mid x \mid \pi_2$ θα έχει τόσες εγγραφές όσες έχει ο π_1 , δηλαδή 100. Επίσης, το $\text{temp1} \mid x \mid \pi_3$ θα έχει τόσες εγγραφές όσες έχει ο temp1 , δηλαδή 100.

Αν κάναμε πρώτα το $\text{temp1} = \pi_2 \mid x \mid \pi_3$ θα είχε 150 εγγραφές και το $\pi_1 \mid x \mid \text{temp1}$ θα είχε 100 εγγραφές.

05 Εκφώνηση/Λύση

Για κάθε ένα από τα παρακάτω ζεύγη αιτημάτων αποφασίστε αν είναι ισοδύναμα και απαντήστε με ένα ΝΑΙ ή ένα ΟΧΙ. Όλα τα αιτήματα αναφέρονται στο παρακάτω σχήμα:

$R(A,B)$, όπου A και B είναι υπονήφια κλειδιά.
 $S(A,B)$, όπου A είναι κλειδί.

Υποθέστε πως κανένας πίνακας δεν περιέχει τιμές NULL και πως τα πεδία με ίδιο όνομα έχουν κοινό πεδίο ορισμού.

Αίτημα 1	Αίτημα 2	Ισοδύναμα;
SELECT A FROM R	$\Pi_A(R)$	ΝΑΙ
SELECT B FROM R	SELECT B FROM R GROUP BY B	ΝΑΙ
SELECT B FROM S	SELECT B FROM S GROUP BY B	ΟΧΙ
SELECT B FROM S	SELECT DISTINCT B FROM S	ΟΧΙ
SELECT R.B FROM R, S WHERE R.A=S.A	SELECT S.B FROM R, S WHERE R.A=S.A	ΟΧΙ
SELECT R.B FROM R, S WHERE R.A=S.A	SELECT B FROM R WHERE A IN (SELECT A FROM S)	ΝΑΙ
(SELECT B FROM S) EXCEPT (SELECT B FROM R)	SELECT B FROM S WHERE B NOT IN (SELECT B FROM R)	ΝΑΙ
$\Pi_A(R - S)$	$\Pi_A(R) - \Pi_A(S)$	ΟΧΙ
$R \times S$	$R \bowtie S$ (natural join)	ΟΧΙ
$\Pi_{R,A}(R)$	$\sigma_{R.A=S}(\Pi_{R,A,R,B}(R \times S))$	ΟΧΙ

06 Εκφώνηση

Έστω το SQL αίτημα:

```
SELECT U.A, U.B, V.D  
FROM U, V  
WHERE U.A = V.A
```

με

U (A char(10), B char(20), C char(70)), όπου A πρωτεύον κλειδί
V (A char(10), D char 40), όπου A ξένο κλειδί που αναφέρεται στο U.A

Ο U αποτελείται από 10000 εγγραφές των 100 bytes και ο V από 50000 εγγραφές των 50 bytes. Έστω ότι το μέγεθος της σελίδας είναι 1000 bytes και η ενδιάμεση μνήμη έχει 20 σελίδες.

Εκτιμήστε το κόστος του κάθε ενός σχεδίου εκτέλεσης του αιτήματος που δίνεται παρακάτω.

(α) Δεν υπάρχουν ευρετήρια.

- (α1) με Εμφώλευση Βρόχων κατά Σελίδα
- (α2) με Εμφώλευση Βρόχων κατά Μπλοκ
- (α3) με Ταξινόμηση και Συγχώνευση
- (α4) με Κατακερματισμό

(β) Υπάρχουν ευρετήρια

- (β1) Υπάρχει μη-συγκροτημένο ευρετήριο κατακερματισμού Εναλλακτικής 2 πάνω στο V.A.
- (β2) Υπάρχει συγκροτημένο ευρετήριο κατακερματισμού Εναλλακτικής 1 στο U.A.

Βοήθεια:

πλήθος σελίδων U: $(10000 \text{ εγγ} * 100 \text{ bytes/εγγ}) / 1000 \text{ bytes/σελ} = 1000 \text{ σελ}$

πλήθος σελίδων V: $(50000 \text{ εγγ} * 50 \text{ bytes/εγγ}) / 1000 \text{ bytes/σελ} = 2500 \text{ σελ}$

Ισοδύναμη σχεσιακή άλγεβρα: $\Pi_{U.A, U.B, V.D}(\sigma_{U.A=V.A}(U \bowtie V))$.

Σε μερικά σχέδια, για να ελαχιστοποιήσουμε το κόστος των συζεύξεων συμφέρει να “κρατήσουμε” για τον U μόνο τα απαραίτητα πεδία, δηλαδή τα A, B. Με άλλα λόγια, στο σχέδιο υπολογισμού εκτέλεσης θα “σπρώξουμε” μια προβολή για τον U προς τα κάτω, πριν τη σύζευξη.

Ισοδύναμη σχεσιακή άλγεβρα: $\Pi_{U.A, U.B, V.D}(\sigma_{U.A=V.A}(\Pi_{U.A, U.B}(U) \bowtie V))$.

06 Λύση

(α1) Εμφώλευση Βρόχων κατά Σελίδα. Εξ ορισμού χρησιμοποιεί μόνο 3 πλαίσια της ενδιάμεσης μνήμης. Ένα για να διαβάζει σελίδα-σελίδα τον εξωτερικό πίνακα, ένα για να διαβάζει σελίδα-σελίδα ολόκληρο τον εσωτερικό πίνακα για κάθε σελίδα του εξωτερικού και ένα ως πλαίσιο εξόδου για τις συζευγμένες (ταιριαστές) εγγραφές των δυο πινάκων.

V εξωτερικός

- Πρώτα εκτελούμε την προβολή $U' = \Pi_{U,A,U,B}(U)$.
- Το μέγεθος του πίνακα U' είναι: $(10000 \text{ εγγ} * 30 \text{ bytes/εγγ})/1000 \text{ bytes/σελ} = 300 \text{ σελ}$
- Κόστος προβολής = $1000 + 300 = 1300 \text{ I/O}$
- Κόστος σύζευξης = $2500 + 2500*300 = 2500 + 750000 = 752500 \text{ I/O}$
- Συνολικό κόστος = $1300 + 752500 = 753800 \text{ I/O}$

U εξωτερικός

- Εφόσον ο U θα διαβαστεί μόνο μια φορά, μπορούμε να κάνουμε την προβολή $\Pi_{U,A,U,B}(U)$ εν πτήση και να αποφύγουμε την αποθήκευση του προσωρινού U'
- Κόστος σύζευξης = $1000 + 300*2500 = 1000 + 750000 = 751000 \text{ I/O}$

Άρα συμφέρει να έχουμε τον U εξωτερικό.

(α2) Εμφώλευση Βρόχων κατά Μπλοκ. Χρησιμοποιούνται $B-2=18$ πλαίσια για τον εξωτερικό πίνακα, ένα για τον εσωτερικό και ένα για την έξοδο.

V εξωτερικός

- Πρώτα εκτελούμε την προβολή $U' = \Pi_{U,A,U,B}(U)$.
- Το μέγεθος του πίνακα U' είναι: $(10000 \text{ εγγ} * 30 \text{ bytes/εγγ})/1000 \text{ bytes/σελ} = 300 \text{ σελ}$
- Κόστος προβολής = $1000 + 300 = 1300 \text{ I/O}$
- Κόστος σύζευξης = $2500 + \text{ceil}(2500/18)*300 = 2500 + 139*300 = 2500 + 41700 = 44200 \text{ I/O}$
- Συνολικό κόστος = $1300 + 44200 = 45500 \text{ I/O}$

U εξωτερικός

- Εφόσον ο U θα διαβαστεί μόνο μια φορά, μπορούμε να κάνουμε την προβολή $\Pi_{U,A,U,B}(U)$ εν πτήση και να αποφύγουμε την αποθήκευση του προσωρινού U'
- Κόστος σύζευξης = $1000 + \text{ceil}(300/18)*2500 = 1000 + 17*2500 = 1000 + 42500 = 43500 \text{ I/O}$

Άρα συμφέρει να έχουμε τον U εξωτερικό.

(α3) Ταξινόμηση και Συγχώνευση. Πρώτα ταξινομούμε τους U και V και μετά συγχωνεύουμε με ένα πέρασμα.

Ταξινόμηση U (με ταυτόχρονη εφαρμογή της προβολής).

Πέρασμα 0: Χρησιμοποιώ 20 πλαίσια και παράγω 50 συρμούς των 6 σελίδων (και όχι των 20 επειδή εξαιτίας της προβολής “ξεφορτώνομαι” το πεδίο B). Μετά χρειάζομαι ακόμα 2 περάσματα που χρησιμοποιούν $B-1=19$ πλαίσια ως είσοδο και ένα ως έξοδο. Το Πέρασμα 1 παράγει $\text{ceil}(50/19)=3$ συρμούς των $19*6, 19*6, 12*6$ (ή 114, 114, 72) σελίδων, και το Πέρασμα 2 παράγει τον τελικό ταξινομημένο πίνακα U.

Ταξινόμηση V.

Πέρασμα 0: Χρησιμοποιώ 20 πλαίσια και παράγω 125 συρμούς των 20 σελίδων. Μετά χρειάζομαι ακόμα 2 περάσματα που χρησιμοποιούν $B-1=19$ πλαίσια ως είσοδο και ένα ως έξοδο. Το Πέρασμα 1 παράγει $\text{ceil}(125/19)=7$ συρμούς, και το Πέρασμα 2 παράγει τον τελικό ταξινομημένο πίνακα V.

$$\text{Συνολικό κόστος} = \text{κόστος ταξ } U + \text{κόστος ταξ } V + \text{κόστος συγχώνευσης} = \\ (1000+300+2*300*2) + (2*2500*3) + (300 + 2500) = 2500 + 12500 + 2800 = 17800 \text{ I/O}$$

[Σημείωση: στην ταξινόμηση του U, στο Πέρασμα 0 διαβάζουμε 1000 σελίδες αλλά λόγω της προβολής γράφουμε 300. Στα επόμενα περάσματα γράφουμε και διαβάζουμε 300 σελίδες]

(α4) με Κατακερματισμό

Κατακερματίζουμε τον U ενώ εφαρμόζουμε την προβολή $\Pi_{U,A,U,B}(U)$. Χρησιμοποιούμε ένα πλαίσιο εισόδου και $B-1=19$ πλαίσια εξόδου (κάδους) για τη δημιουργία των 19 διαμερισμάτων. Υποθέτοντας “καλή” συνάρτηση κατακερματισμού θα πάρουμε 19 διαμερίσματα των $\text{ceil}(300/19)=16$ σελίδων (οι σελίδες που δημιουργούνται είναι 300 και όχι 1000 επειδή εφαρμόζουμε την προβολή).

Κατακερματίζουμε τον V. Χρησιμοποιούμε ένα πλαίσιο εισόδου και $B-1=19$ πλαίσια εξόδου (κάδους) για τη δημιουργία των 19 διαμερισμάτων. Υποθέτοντας “καλή” συνάρτηση κατακερματισμού θα πάρουμε 19 διαμερίσματα των $\text{ceil}(2500/19)=132$ σελίδων.

Αφού $B > \sqrt{300}$ ή $20 > 18$ θα χρησιμοποιήσουμε τον U ως εξωτερικό πίνακα. Χρησιμοποιούμε $B-1=19$ πλαίσια για να φορτώνουμε κάθε ένα διαμέρισμα του U και ένα πλαίσιο για να σαρώνουμε το αντίστοιχο διαμέρισμα του V. Έτσι οι U και V σαρώνονται μια φορά για τη σύζευξη.

$$\text{Συνολικό Κόστος} = \text{κόστος κατακερματισμού } U + \text{κόστος κατακερματισμού } V + \text{κόστος σύζευξης} = \\ (1000+300) + (2500+2500) + (300+2500) = 1300 + 5000 + 2800 = 9100 \text{ I/O}$$

(β1) Υπάρχει μη-συγκροτημένο ευρετήριο κατακερματισμού Εναλλακτικής 2 πάνω στο V.A.

Σαρώνουμε τον U και για κάθε εγγραφή του χρησιμοποιούμε το ευρετήριο για να βρούμε τις ταιριαστές εγγραφές στον V. Θεωρώντας ομοιόμορφη κατανομή στις τιμές του πεδίου V.A κάθε εγγραφή του U θα έχει 5 ταιριαστές εγγραφές στον V (10000 εγγραφές έχει ο U με το U.A κλειδί, 50000 εγγραφές έχει ο V). Θεωρώντας ότι έχουμε ευρετήριο κατακερματισμού χρειάζονται 1,2 I/O για την εύρεση μιας καταχώρισης δεδομένων και 1 I/O για την εύρεση της αντίστοιχης εγγραφής (σύνολο $1,2 + 5*1 = 6,2$ I/O ανά 5 ταιριάσματα).

$$\text{Κόστος σύζευξης} = 1000 + 10000*(1,2 + 5*1) = 63000 \text{ I/O}$$

(β2) Υπάρχει συγκροτημένο ευρετήριο κατακερματισμού Εναλλακτικής 1 στο U.A.

Σαρώνουμε τον V και για κάθε εγγραφή του χρησιμοποιούμε το ευρετήριο για να βρούμε την ταιριαστή εγγραφή στον U. Αφού το ευρετήριο είναι Εναλλακτικής 1, το κόστος εύρεσης ταιριαστής εγγραφής είναι 1,2 I/O.

$$\text{Κόστος σύζευξης} = 2500 + 50000*1,2 = 62500 \text{ I/O}$$

07 Εκφώνηση

Είναι τα παρακάτω χρονοπρογράμματα που εμπλέκουν τις συναλλαγές T1 και T2 σειριοποιήσιμα; Αν κάποιο χρονοπρόγραμμα είναι σειριοποιήσιμο, προσθέστε τις κατάλληλες εντολές κλειδώματος (X ή S), ξεκλειδώματος (U) και Commit, ώστε να έχουμε ένα πλήρες χρονοπρόγραμμα σύμφωνα με το 2PL (αυστηρό ή απλό). Αν κάποιο χρονοπρόγραμμα δεν είναι σειριοποιήσιμο, προτείνετε τροποποίησή του ώστε να μπορεί να προκύψει (μη-σειριακό) σειριοποιήσιμο χρονοπρόγραμμα.

(1)

T1	W(A)	R(B)			R(C)									
T2			R(C)	W(B)		W(A)								

(2)

T1	W(A)		W(B)											
T2		R(B)		W(B)										

(3)

T1	R(A)	R(B)	W(C)			W(D)		W(A)						
T2				R(B)	R(D)		R(A)							

08 Εκφώνηση

Στο παρακάτω χρονοπρόγραμμα υπάρχει αδιέξοδος.

(α) Σχεδιάστε το γράφημα waits-for.

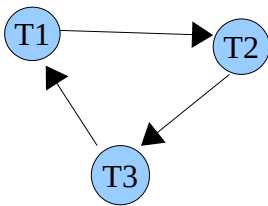
(β) Αν το σύστημα εφαρμόζει Ανίχνευση Αδιεξόδου, εξηγήστε τί θα συμβεί.

(γ) Αν το σύστημα εφαρμόζει Πρόληψη Αδιεξόδου, εξηγήστε τί θα συμβεί σε κάθε μια από τις 2 στρατηγικές πρόληψης, δηλ., (γ1) wait-die και (γ2) wound-wait.

T1	R(A)				R(D)		R(C)							
T2			W(C)	W(B)										
T3		R(B)				W(A)								

08 Λύση

(α)



(β) Το σύστημα θα ανιχνεύσει την ύπαρξη κύκλου (αμέσως μετά το R(C)) και θα επανεκκινήσει μια από τις συναλλαγές (π.χ., τη νεότερη T2, ή μια στην τύχη). Έτσι, με την αποδέσμευση των αντίστοιχων κλειδαριών, ο κύκλος θα σπάσει και κάποια(ες) από τις υπόλοιπες συναλλαγές θα συνεχίσει(ουν).

(γ1) wait-die

Όταν η T2 ζητήσει την X(B), θα πεθάνει (die) επειδή είναι νεότερη από την T3 που έχει την S(B). Έτσι, η T2 επανεκκινείται. Λίγο μετά, όταν η T3 ζητήσει την X(A), θα πεθάνει (die) και αυτή επειδή είναι νεότερη από την T1 που έχει την S(A).

(γ2) wound-wait

Όταν η T2 ζητήσει την X(B), θα περιμένει (wait) επειδή είναι νεότερη από την T3 που έχει την S(B). Λίγο αργότερα, όταν η T3 ζητήσει την X(A), επίσης θα περιμένει (wait) επειδή είναι νεότερη από την T1 που έχει την S(A). Έτσι, όταν η T1 ζητήσει την S(C), θα σκοτώσει (wound) την T2 επειδή είναι αρχαιότερη της T2 που έχει την X(C). Έτσι, θα ξεμπλοκάρει και η T3 και μέχρι να επανεκκινηθεί η T2 πιθανόν να έχουν τερματίσει οι T1 και T3.

09 Εκφώνηση

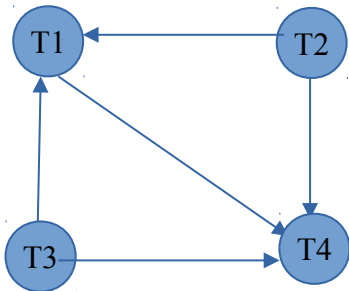
Αποφασίστε αν το ακόλουθο χρονοπρόγραμμα είναι συγκρουσιακά σειριοποιήσιμο κατασκευάζοντας το σχετικό γράφημα προτεραιότητας. Αν είναι, δώστε το αντίστοιχο σειριακό χρονοπρόγραμμα. Ο αριθμός δίπλα στο R ή W υποδηλώνει τη συναλλαγή που εκτελεί την ενέργεια.

R2(X), W3(Y), R1(Y), W2(X), W1(X), W1(Y), W4(Y), W4(X)

09 Λύση

T1			R(Y)		W(X)	W(Y)		
T2	R(X)			W(X)				
T3		W(Y)						
T4							W(Y)	W(X)

Γράφημα Προτεραιότητας:



Δεν υπάρχει κύκλος, άρα το χρονοπρόγραμμα είναι συγκρουσιακά σειριοποιήσιμο. Ισοδύναμο σειριακό: T2, T3, T1, T4 ή T3, T2, T1, T4.

10 Εκφώνηση

Αποφασίστε ποιες από τις ακόλουθες ενέργειες δεν μπορούν να εκτελεστούν και προσδιορίστε αν υπάρχει αδιέξοδο ή όχι. Κάθε ενέργεια X αναπαριστά τη λήψη μιας αποκλειστικής κλειδαριάς, κάθε ενέργεια U αναπαριστά την αποδέσμευση μιας κλειδαριάς και οι αριθμοί, τη συναλλαγή που εκτελεί την κάθε ενέργεια. Πρέπει να δώσετε το γράφημα waits-for και τον πίνακα κλειδαριών (lock table) για κάθε χρονική στιγμή που λαμβάνεται ή αποδεσμεύεται μια κλειδαριά.

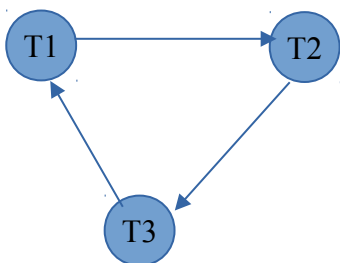
X2(B) R2(B) X1(A) W2(B) R1(A) X3(C) W3(C) X1(B) R1(B) W1(A) X2(C) R2(C) U2(B) X3(A) W2(C) R3(A) W3(A) U3(A) U2(C) U1(A) U1(B) U3(C)

10 Λύση

	1	2	3	4	5	6	7	8	9	10	11	12
T1		X(A)		X(B)						U(A)	U(B)	
T2	X(B)				X(C)	U(B)			U(C)			
T3			X(C)				X(A)	U(A)				U(C)

Έχω καταγράψει μόνο τις αιτήσεις και αποδεσμεύσεις κλειδαριών που είναι οι ενέργειες που μας ενδιαφέρουν. Προφανώς, τα γαλάζια κελιά περιέχουν εντολές που δεν μπορούν να εκτελεστούν τις δεδομένες χρονικές στιγμές, αλλά αργότερα, αν όλα πάνε καλά και “ξεμπλοκάρουν” οι αντίστοιχες συναλλαγές..

Γράφημα αναμονής με κύκλο. Αδιέξοδο τη χρονική στιγμή 7.



Πίνακας Κλειδαριών

Αντικείμενο	Κάτοχος κλειδαριάς	Τύπος κλειδαριάς	Ουρά αναμονής
A	T1	X	T3(X)
B	T2	X	T1(X)
C	T3	X	T2(X)

11 Εκφώνηση

Θεωρείστε το παρακάτω αρχείο ιστορικού (log file). Οι κωδικοί συναλλαγών είναι οι T_i και οι κωδικοί σελίδων είναι οι P_i. Ο αριθμός στην αρχή της κάθε εγγραφής ιστορικού είναι ο LSN.

LSN	PREV_LSN	TRANS_ID	TYPE	PAGE_ID	UNDO_NEXT_LSN
10			BEGIN_CHECKPOINT		
20	-	T1	UPDATE	P1	
30			END_CHECKPOINT		
40	-	T2	UPDATE	P1	
50	-	T3	UPDATE	P2	
60			BEGIN_CHECKPOINT		
70	20	T1	ABORT		
80	70	T1	CLR 20		-
90	80	T1	END		
100	40	T2	UPDATE	P3	
110			END_CHECKPOINT		
120	100	T2	UPDATE	P2	
130	120	T2	COMMIT		
140	50	T3	UPDATE	P1	
150	140	T3	UPDATE	P1	

Υποθέστε ότι το σύστημα καταρρέει αμέσως μετά την εγγραφή στο δίσκο της εγγραφής ιστορικού με LSN 130 (δηλαδή, τη στιγμή της κατάρρευσης είχαμε FLUSHED_LSN=130).

(α) Από ποιο LSN θα ξεκινήσει η φάση Ανάλυσης του ARIES; Δώστε τα περιεχόμενα του πίνακα συναλλαγών και του πίνακα τροποποιημένων σελίδων αμέσως μετά την φάση Ανάλυσης.

(β) Από από ποιο LSN θα ξεκινήσει η φάση REDO;

(γ) Μέχρι ποιο LSN θα φτάσει η φάση UNDO;

(δ) Δώστε με τη σειρά τις εγγραφές ιστορικού που θα προκύψουν κατά τη φάση UNDO του ARIES.

11 Λύση

(α) 60

Πίνακας συναλλαγών

TRANS_ID	LAST_LSN	STATUS
T2	130	C
T3	50	R

Πίνακας τροποποιημένων σελίδων

PAGE_ID	REC_LSN
P1	20
P2	50
P3	100

(β) 20

(γ) 50

(δ) Η πρώτη από τις παρακάτω εγγραφές (140) προκύπτει κατά τη φάση REDO.

LSN	PREV_LSN	TRANS_ID	TYPE	PAGE_ID	UNDO_NEXT_LSN
140	130	T2	END		
150	50	T3	CLR 50		-
160	150	T3	END		

12 Εκφώνηση

Θεωρείστε το παρακάτω αρχείο ιστορικού (log file). Κάθε εγγραφή του ιστορικού που αντιστοιχεί σε ενημέρωση (update log record) περιλαμβάνει τον κωδικό συναλλαγής, τον κωδικό αντικειμένου, την παλιά τιμή και τη νέα τιμή του αντικειμένου. Οι κωδικοί συναλλαγών είναι οι T1, T2, T3 και T4. Οι κωδικοί αντικειμένων είναι οι A, B, C, D, και E. Ο αριθμός στην αρχή της κάθε εγγραφής ιστορικού είναι ο LSN.

LSN	PREV_LSN	TRANS_ID	TYPE	PAGE_ID	UNDO_NEXT_LSN
1			Begin checkpoint		
2			End checkpoint		
3	null	T1	Update	A	
4	3	T1	Abort		
5	null	T2	Update	B	
6	null	T3	Update	C	
7	5	T2	Update	D	
8	6	T3	Update	E	
9	4	T1	CLR LSN3		null
10	8	T3	Commit		
11	10	T3	End		
12	9	T1	End		
13	null	T4	Update	A	
14			Begin checkpoint		
15	13	T4	Update	F	
16			End checkpoint		

Υποθέστε ότι το σύστημα καταρρέει αμέσως μετά την εγγραφή στο δίσκο της εγγραφής ιστορικού με LSN 16.

(α) Ποιες συναλλαγές πρέπει να αναιρεθούν (UNDO);

(β) Ποιες συναλλαγές πρέπει να επανεκτελεστούν (REDO);

(γ) Δώστε τα περιεχόμενα του πίνακα συναλλαγών και του πίνακα τροποποιημένων σελίδων αμέσως μετά την φάση της ανάλυσης του ARIES.

(δ) Δώστε με τη σειρά τις εγγραφές ιστορικού που θα προκύψουν κατά τη φάση UNDO του ARIES.

12 Λύση

(α) T2, T4

(β) Όλες

(γ)

Πίνακας συναλλαγών

TRANS_ID	LAST_LSN	STATUS
T2	7	running
T4	15	running

Πίνακας τροποποιημένων σελίδων

PAGE_ID	REC_LSN
A	3
B	5
C	6
D	7
E	8
F	15

(δ)

LSN	PREV_LSN	TRANS_ID	TYPE	PAGE_ID	UNDO_NEXT_LSN
17		T4	CLR 15		13
18		T4	CLR 13		null
19		T4	End		
20		T2	CLR 7		5
21		T2	CLR 6		null
22		T2	End		