

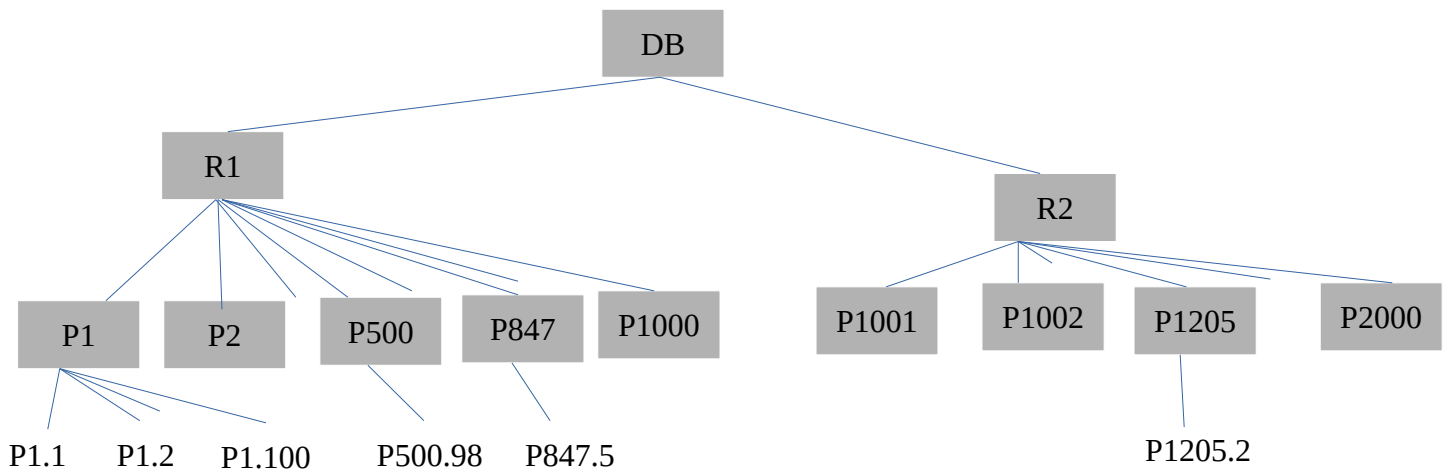
Προχωρημένα Θέματα Τεχνολογίας & Εφαρμογών Βάσεων Δεδομένων

Άσκηση 2

Όθωνας Γκαβαρδίνας AM: 2620

ΘΕΜΑ 1

(α)



T1: IS1(DB), IS1(R1), IS1(P847), S1(P847.5)

T2: IS2(DB), S2(R1)

ή εναλλακτικά, γίνεται κλιμάκωση κλειδώματος, κατά την οποία κλειδώνεται η R1 με IS-lock, και οι σελίδες P1, ..., P1000, με S-lock.

T3: IS3(DB), IS3(R1), IS3(P500), S3(P500.98), S3(P500.99), S3(P500.100), S3(P501), S3(P502), ..., S3(P1000), IS3(R2), S3(P1001), S3(P1002), ..., S3(P1204), IS3(P1205), S3(P1205.2)

T4: SIX4(DB), SIX4(R1), S(P1), S(P2), ..., S(P1000), ...

Εάν η συνθήκη ισχύει για κάποιες από τις σελίδες του R1, αυτές κλειδώνονται με X-lock.

(β)

R1(1007.35) R1(1008.12) R2(1007.35) W1(1008.12) R2(1007.42) W2(1007.42) W1(1007.35)
W1(1008.17) C2 C1

T1: R1(1007.35) R1(1008.12) W1(1008.12) W1(1007.35) W1(1008.17)

T2: R2(1007.35) R2(1007.42) W2(1007.42)

	DB	R2	1007	1007.35	1007.42	1008	1008.12	1008.17
T1	IS	IS	IS	S		IX	X	
T2	IS	IS	IS	S	X			

-Αρχικά κλειδώνω με S-lock, γνωρίζοντας, ότι στη συνέχεια ο πόρος θα χρησιμοποιηθεί από την T2. (Γι' αυτό και δεν βάζω X-lock, μιας και υπάρχει W1(1007.35))

output: IS1(DB) IS1(R2) IS1(1007) S1(1007.35) R1(1007.35)

-Στη συνέχεια κλειδώνω με X-lock, επειδή η T1 θα χρειαστεί τον πόρο για γράψιμο.

output: IS1(DB) IS1(R2) IS1(1007) S1(1007.35) R1(1007.35) IX1(1008) X1(1008.12)
R1(1008.12)

-Ο πόρος 1007.35, έχει κλειδωθεί με S-lock κλειδαριά. Μπορώ να έχω πολλές τέτοιες κλειδαριές μαζί. Το ίδιο ισχύει για τα IS-locks.

output: IS1(DB) IS1(R2) IS1(1007) S1(1007.35) R1(1007.35) IX1(1008) X1(1008.12)
R1(1008.12) IS2(DB) IS2(R2) IS2(1007) S2(1007.35) R2(1007.35)

-Έχω ήδη κλειδώσει για την T1 τον πόρο 1008.12, με X-lock, οπότε δε χρειάζεται ξανά.

output: IS1(DB) IS1(R2) IS1(1007) S1(1007.35) R1(1007.35) IX1(1008) X1(1008.12)
R1(1008.12) IS2(DB) IS2(R2) IS2(1007) S2(1007.35) R2(1007.35) W1(1008.12)

-Γνωρίζω ότι για τον πόρο 1007.42, γίνεται από την T2 και διάβασμα και γράψιμο στη συνέχεια άρα κλειδώνω με X, SIX locks. Δεν υφίσταται σύγκρουση μεταξύ IS, και SIX locks.

output: IS1(DB) IS1(R2) IS1(1007) S1(1007.35) R1(1007.35) IX1(1008) X1(1008.12)
R1(1008.12) IS2(DB) IS2(R2) IS2(1007) S2(1007.35) R2(1007.35) W1(1008.12)
SIX2(DB) SIX2(R2) IX2(1007) SIX2(1007.42) R2(1007.42)

Ο πίνακας γίνεται:

	DB	R2	1007	1007.35	1007.42	1008	1008.12	1008.17
T1	IS	IS	IS	S		IX	X	
T2	SIX	SIX	SIX	S	X			

-Στη συνέχεια γίνεται και η εγγραφή.

output: IS1(DB) IS1(R2) IS1(1007) S1(1007.35) R1(1007.35) IX1(1008) X1(1008.12)
R1(1008.12) IS2(DB) IS2(R2) IS2(1007) S2(1007.35) R2(1007.35) W1(1008.12)
SIX2(DB) SIX2(R2) IX2(1007) SIX2(1007.42) R2(1007.42) W2(1007.42)

-Για τον πόρο 1007.35, δεν μπορώ απλά να κλειδώσω με X-lock, γιατί θα υπάρξουν συγκρούσεις. Επομένως αρχικά γίνονται τα unlocks για την T1. Συγκεκριμένα, με εμποδίζουν τα SIX-locks, τα οποία θα χρειαστούν να γίνουν αμέσως όλα τα unlocks.

output: IS1(DB) IS1(R2) IS1(1007) S1(1007.35) R1(1007.35) IX1(1008) X1(1008.12)
R1(1008.12) IS2(DB) IS2(R2) IS2(1007) S2(1007.35) R2(1007.35) W1(1008.12)
SIX2(DB) SIX2(R2) IX2(1007) SIX2(1007.42) R2(1007.42) W2(1007.42)
U2(1007.35) U2(1007.42) U2(1007) U2(R2) U2(DB)

Ο πίνακας γίνεται:

	DB	R2	1007	1007.35	1007.42	1008	1008.12	1008.17
T1	IX	IX	IX	X		IX	X	X
T2								

- Η T1 κλειδώνει τον πόρο 1007.35, για εγγραφή.

output: IS1(DB) IS1(R2) IS1(1007) S1(1007.35) R1(1007.35) IX1(1008) X1(1008.12)
R1(1008.12) IS2(DB) IS2(R2) IS2(1007) S2(1007.35) R2(1007.35) W1(1008.12)
SIX2(DB) SIX2(R2) IX2(1007) SIX2(1007.42) R2(1007.42) W2(1007.42)
U2(1007.35) U2(1007.42) U2(1007) U2(R2) U2(DB) IX1(DB) IX1(R2) IX1(1007)
X1(1007.35) W1(1007.35)

- Η T1 κλειδώνει τον πόρο 1008.17, για εγγραφή.

output: IS1(DB) IS1(R2) IS1(1007) S1(1007.35) R1(1007.35) IX1(1008) X1(1008.12)
R1(1008.12) IS2(DB) IS2(R2) IS2(1007) S2(1007.35) R2(1007.35) W1(1008.12)
SIX2(DB) SIX2(R2) IX2(1007) SIX2(1007.42) R2(1007.42) W2(1007.42)
U2(1007.35) U2(1007.42) U2(1007) U2(R2) U2(DB) IX1(DB) IX1(R2) IX1(1007)
X1(1007.35) W1(1007.35)

-Τέλος γίνονται τα unlocks και τα commits.

output: IS1(DB) IS1(R2) IS1(1007) S1(1007.35) R1(1007.35) IX1(1008) X1(1008.12)
R1(1008.12) IS2(DB) IS2(R2) IS2(1007) S2(1007.35) R2(1007.35) W1(1008.12)
SIX2(DB) SIX2(R2) IX2(1007) SIX2(1007.42) R2(1007.42) W2(1007.42)
U2(1007.35) U2(1007.42) U2(1007) U2(R2) U2(DB) IX1(DB) IX1(R2) IX1(1007)
X1(1007.35) W1(1007.35) U1(1008.17) U1(1008.12) U1(1008) U1(1007.35)
U1(1007) U1(R2) U1(DB) C2 C1

Απάντηση:

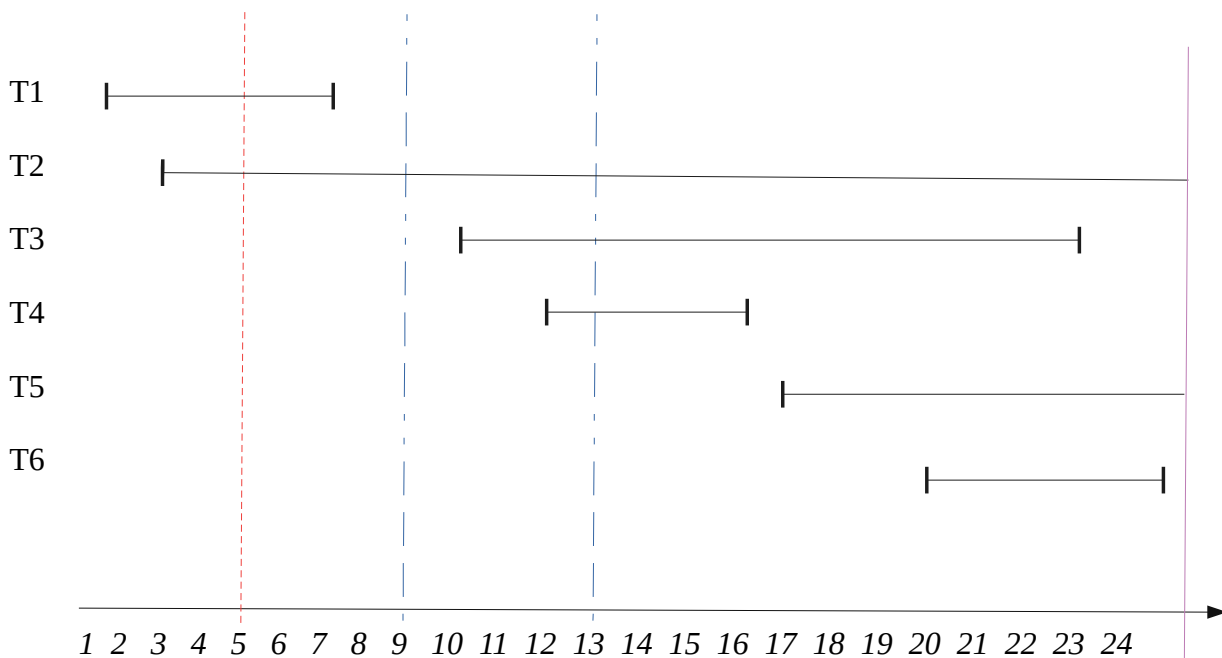
Ναι, μπορεί το δοθέν χρονοπρόγραμμα να έχει προκύψει από simple 2PL scheduler με ιεραρχίες.

(γ)

Στην περίπτωση αυστηρού 2PL όλα τα unlocks γίνονται ακριβώς πριν γίνει commit. Στο παραπάνω χρονοπρόγραμμα, δε θα μπορούσε να προκύψει αυστηρός 2PL γιατί η στιγμή που γίνονται τα unlocks για την T2, αναγκαστικά πρέπει να διαφέρει από τη στιγμή που γίνει το commit γι' αυτή.

ΘEMA 2

1. trans begin T1
2. write T1
3. trans begin T2
4. write T2
5. Physical, full system backup
6. write T2
7. commit T1
8. write T2
9. start fuzzy ckeckpoint
10. trans begin T3
11. write T3
12. trans begin T4
13. end fuzzy ckeckpoint
14. write T4
15. write T2
16. commit T4
17. trans begin T5
18. write T5
19. write T3
20. trans begin T6
21. write T6
22. write T2
23. abort T3
24. ...
- XX. commit T6
- crash – end of log
- XX+1.
- XX+2.



1. Τα prevLSN έχουν τοποθετηθεί στο παραπάνω log.

2.

24. CLR: Undo T3 LSN 19

25. CLR: Undo T3 LSN 11

26. CLR: Undo T3 END

27. commit T6

3.

Winner List:

T1, T4, T6

Loser List:

T2, T5, (**T3 abort)

27. commit T6

CRASH

28. CLR: Undo T2 LSN 15

29. CLR: Undo T5 LSN 17

30. CLR: Undo T5 END

31. CLR: Undo T2 LSN 8

32. CLR: Undo T2 LSN 6

33. CLR: Undo T2 LSN 4

34. CLR: Undo T2 LSN 3

35. CLR: Undo T2 END

36. Recovery Process Complete

4. Στην περίπτωση No Steal – No Force, δε χρειάζεται Undo, γιατί στο δίσκο δεν γράφονται δεδομένα που δεν έχουν γίνει commit.

5. Στην περίπτωση No Steal – Force, δε χρειάζεται Undo, και ούτε Redo. Αυτό γιατί δε γράφονται δεδομένα χωρίς να έχει γίνει commit, και επίσης πριν από κάθε commit γράφονται στο δίσκο όλες οι dirty pages.

!. Σε περίπτωση που δε χάνεται η βάση δεδομένων, και απλά πέφτει ο server, ελέγχω τα checkpoints, αντί για τα backups. Επίσης, έχω fuzzy checkpoints, τα οποία συμπεριφέρονται όπως τα sharp checkpoints στο σημείο που γίνεται το start.

Winner List:

(*T1 dont care), T4, T6

Loser List:

T2, T5, (**T3 abort)

Οι losers είναι οι ίδιοι με αποτέλεσμα, οι εγγραφές στο log να είναι οι ίδιες με τις παραπάνω (του ερωτήματος 3)

<<Σημείωση: Για το abort, δεν έγραψα κάποιο Undo στην ανάνηψη από το crush, το θεωρώ πως δεν υπήρξε.>>