

## Θεματολόγιο

- A) Πρακτικά Θέματα στον Έλεγχο Ταυτοχρονισμού
  - Φαντάσματα
  - Κλείδωμα δέντρων
  - Κλείδωμα σε διαφορετικά επίπεδα διακριτότητας
  - Πώς τα βάζω όλα μαζί
- B) Εναλλακτικές Τεχνικές
  - Αισιόδοξος έλεγχος ταυτοχρονισμού
  - Έλεγχος ταυτοχρονισμού με χρονόσημα
  - Έλεγχος ταυτοχρονισμού με πολλαπλές εκδόσεις
- Γ) Δοσοληψίες και SQL

1

## Θεματολόγιο

- Φαντάσματα
- Κλείδωμα δέντρων
- Κλείδωμα σε διαφορετικά επίπεδα διακριτότητας
- Πώς τα βάζω όλα μαζί
- Αισιόδοξος έλεγχος ταυτοχρονισμού
- Έλεγχος ταυτοχρονισμού με χρονόσημα
- Έλεγχος ταυτοχρονισμού με πολλαπλές εκδόσεις
- Δοσοληψίες και SQL

2

## Αρχική παρατήρηση

- Μέχρι στιγμής, κάναμε δύο μη προφανείς υποθέσεις:
  - Η ΒΔ δεν εμπλουτίζεται με νέες εγγραφές κατά τη διάρκεια των δοσοληπιών (ήτοι, υποθέταμε ότι write = UPDATE και όχι INSERT)
  - Τα αντικείμενα της ΒΔ είναι ανεξάρτητα μεταξύ τους (το οποίο καταρρέει αν υποθέσουμε π.χ., ένα B+ tree σε κάποιο πεδίο ενός πίνακα)
- Τι θα γίνει αν αναιρέσουμε τις υποθέσεις αυτές;

3

## Παράδειγμα: Sailors

SID	SNAME	RATING	AGE
1	Popay1	1	71
2	Popay2	1	43
3	Popay3	2	80
4	Popay4	2	63

T1:  
SELECT RATING,  
MAX(AGE)  
FROM SAILORS  
GROUP BY RATING

T2:  
INSERT INTO SAILORS (5,'X',1,96);  
DELETE FROM SAILORS WHERE  
RATING = 2 AND  
AGE IN (SELECT MAX(AGE) FROM  
SAILORS WHERE RATING =2)

4

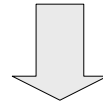
## Πιθανή Εκτέλεση

SID	SNAME	RATING	AGE
1	Popay1	1	71
2	Popay2	1	43
3	Popay3	2	80
4	Popay4	2	63

X1

X1

T1:  
SELECT RATING,  
MAX(AGE)  
FROM SAILORS  
GROUP BY RATING



RATING	AGE
1	71

T2: INSERT INTO SAILORS (5,'X',1,96);  
DELETE FROM SAILORS WHERE  
RATING = 2 AND  
AGE IN (SELECT MAX(AGE) FROM  
SAILORS WHERE RATING =2)



5

## Πιθανή Εκτέλεση

SID	SNAME	RATING	AGE
1	Popay1	1	71
2	Popay2	1	43
3	Popay3	2	80
4	Popay4	2	63
5	X	1	96

X1

X1

T1:  
SELECT RATING,  
MAX(AGE)  
FROM SAILORS  
GROUP BY RATING

RATING	AGE
1	71

T2: INSERT INTO SAILORS (5,'X',1,96);  
DELETE FROM SAILORS WHERE  
RATING = 2 AND  
AGE IN (SELECT MAX(AGE) FROM  
SAILORS WHERE RATING =2)

X2



6

## Πιθανή Εκτέλεση

SID	SNAME	RATING	AGE
1	Popay1	1	71
2	Popay2	1	43
<del>3</del>	<del>Popay3</del>	<del>2</del>	<del>80</del>
4	Popay4	2	63
5	X	1	96

X1

X1

X2

X2

X2

T1:  
SELECT RATING,  
MAX(AGE)  
FROM SAILORS  
GROUP BY RATING

RATING	AGE
1	71

T2: INSERT INTO SAILORS (5,'X',1,96);  
DELETE FROM SAILORS WHERE  
RATING = 2 AND  
AGE IN (SELECT MAX(AGE) FROM  
SAILORS WHERE RATING =2)

To DEL παίρνει  
X-lock

7

## Πιθανή Εκτέλεση

SID	SNAME	RATING	AGE
1	Popay1	1	71
2	Popay2	1	43
4	Popay4	2	63
5	X	1	96

X1

X1

T1:  
SELECT RATING,  
MAX(AGE)  
FROM SAILORS  
GROUP BY RATING

RATING	AGE
1	71

T2: INSERT INTO SAILORS (5,'X',1,96);  
DELETE FROM SAILORS WHERE  
RATING = 2 AND  
AGE IN (SELECT MAX(AGE) FROM  
SAILORS WHERE RATING =2)

T2 COMMITS!!

8

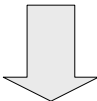
## Πιθανή Εκτέλεση

SID	SNAME	RATING	AGE
1	Popay1	1	71
2	Popay2	1	43
4	Popay4	2	63
5	X	1	96

T2: INSERT INTO SAILORS (5,'X',1,96);  
 DELETE FROM SAILORS WHERE  
 RATING = 2 AND  
 AGE IN (SELECT MAX(AGE) FROM  
 SAILORS WHERE RATING =2)

T1:  
 SELECT RATING,  
 MAX(AGE)  
 FROM SAILORS  
 GROUP BY RATING

X1



RATING	AGE
1	71
2	63

Μετά, όσοι έχουν  
 RATING=2

9

## Πιθανές Εκτελέσεις...

SID	SNAME	RATING	AGE
1	Popay1	1	71
2	Popay2	1	43
3	Popay3	2	80
4	Popay4	2	63
5	X	1	96

T1 || T2

RATING	AGE
1	71
2	63

T1 ; T2

RATING	AGE
1	71
2	80

T2 ; T1

RATING	AGE
1	96
2	63

10

## Φάντασμα

- Το φαινόμενο κατά το οποίο μια δοσοληψία ξεκινά με ένα σύνολο εγγραφών στην αρχή της, και βρίσκεται με ένα διαφορετικό σύνολο εγγραφών στο τέλος της, χωρίς να έχει κάνει αυτή τις αλλαγές.
- Ο 2PL ΔΕΝ μας προστατεύει από φαντάσματα!

11

## Αν μπορούσαμε άραγε ...

SID	SNAME	RATING	AGE
1	Popay1	1	71
2	Popay2	1	43
3	Popay3	2	80
4	Popay4	2	63
5	X	1	96



...να κλειδώναμε στην ίδια δοσοληψία όλες τις εγγραφές που πληρούν τη συνθήκη RATING=1...

12

## Κλείδωμα κατηγορήματος

- Το γεγονός ότι κλειδώνω όλες τις εγγραφές που πληρούν μια συνθήκη (όπως π.χ., RATING = 1)
- Η πιο απλοϊκή λύση, είναι να κλειδώσω όλον τον πίνακα (βλ. και στη συνέχεια)
- Καμιά πιο έξυπνη ιδέα;

13

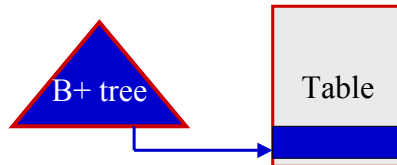
## Θεματολόγιο

- Φαντάσματα
- Κλείδωμα δέντρων
- Κλείδωμα σε διαφορετικά επίπεδα διακριτότητας
- Πώς τα βάζω όλα μαζί
- Αισιόδοξος έλεγχος ταυτοχρονισμού
- Έλεγχος ταυτοχρονισμού με χρονόσημα
- Έλεγχος ταυτοχρονισμού με πολλαπλές εκδόσεις
- Δοσοληψίες και SQL

14

## Κλείδωμα ευρετηρίων

- ✦ Αν είχαμε ένα index (π.χ., B+ tree) πάνω στο πεδίο rating, θα μπορούσαμε να κλειδώναμε γρήγορα όλες τις εγγραφές με RATING=1.
- ✦ Ακόμα κι αν δεν υπάρχουν τέτοιες εγγραφές, κλειδώνοντας τον index, μπορούμε να απαγορεύσουμε να δημιουργηθούν από άλλους κατά τη διάρκεια της δοσοληψίας!
- ✦ Ειδική περίπτωση του κλειδώματος κατηγορήματος...



15

## Πρώτη λύση

- ✦ Ας υποθέσουμε ότι θέλουμε να εισάγουμε μια εγγραφή με RATING=1 στη διάρκεια της T1
- ✦ Κατεβαίνοντας από τη ρίζα προς τα φύλλα του B+ δέντρου, μπορούμε να κλειδώνουμε κάθε κόμβο με X-lock [του φύλλου συμπεριλαμβανομένου]

Τραγική απόδοση: ήδη από το πρώτο βήμα έχω κλειδώσει τη ρίζα, ήτοι, ΟΛΟ το δέντρο.

16



## Παρατήρηση

- ✦ Στα B+ δέντρα, οι ενδιάμεσοι κόμβοι χρησιμεύουν ΜΟΝΟ ως μονοπάτια για τα φύλλα [θυμάστε τη διαφορά με τα B δέντρα;]
- ✦ Αν κλείδωνα με **S-lock** τους **ενδιάμεσους** κόμβους και με **X-lock** τα **φύλλα**;

**Bug:** τι θα γίνει αν η εισαγωγή διασπάσει το παιδί ;

17

## Απλός αλγόριθμος κλειδώματος δέντρων

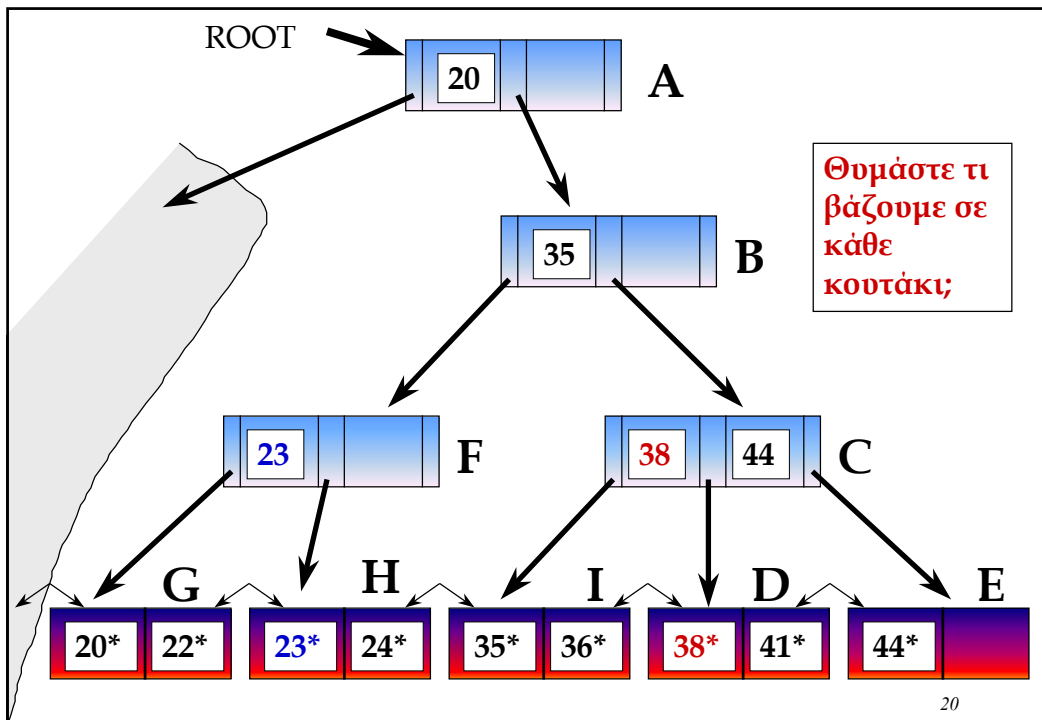
- ✦ **SELECT:** Ξεκινώντας από τη ρίζα κλειδώνουμε με S-lock κάθε κόμβο. Μετά, ξεκλειδώνουμε τον γονέα του.
- ✦ **INS/DEL:** Ξεκινώντας από τη ρίζα κλειδώνουμε με X-locks *κατά περίπτωση*. Μετά ελέγχουμε αν ο κλειδωθείς κόμβος είναι **ασφαλής**:
  - ✦ Αν ναι, ξεκλειδώνουμε τους προγόνους του.
- ✦ **Ασφαλής κόμβος:** οι όποιες αλλαγές δεν θα διαδοθούν προς τα πάνω από τον κόμβο αυτό
  - ✦ INS: ο κόμβος δεν είναι πλήρης.
  - ✦ DEL: ο κόμβος δεν είναι άδειος κατά το ήμισυ.

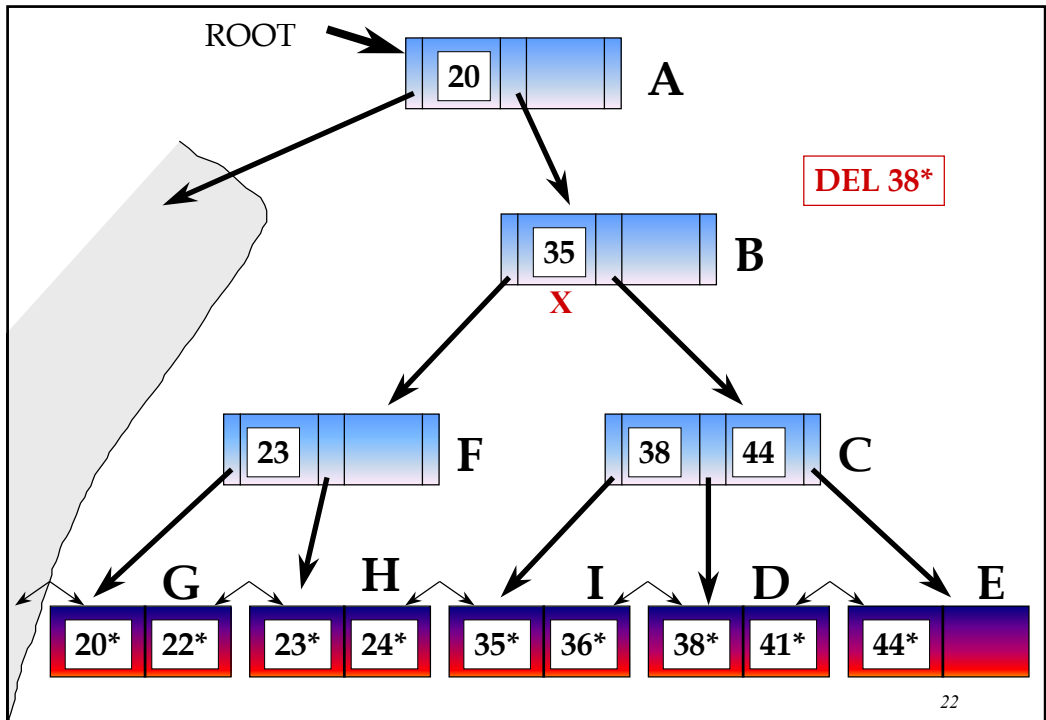
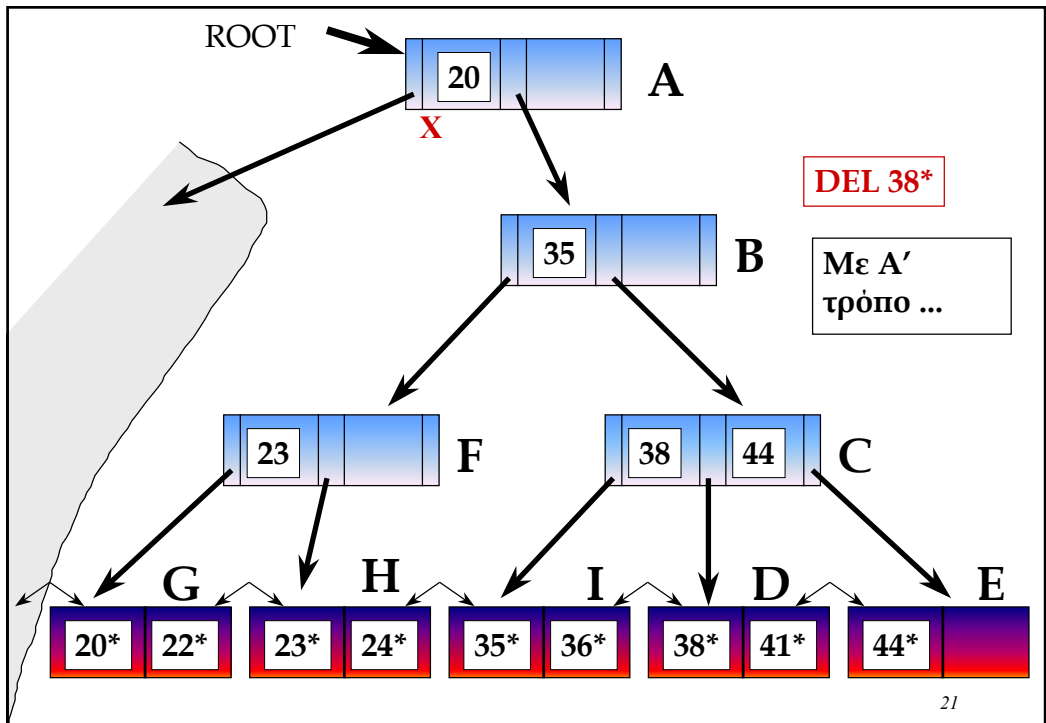
18

## Απλός αλγόριθμος κλειδώματος δέντρων

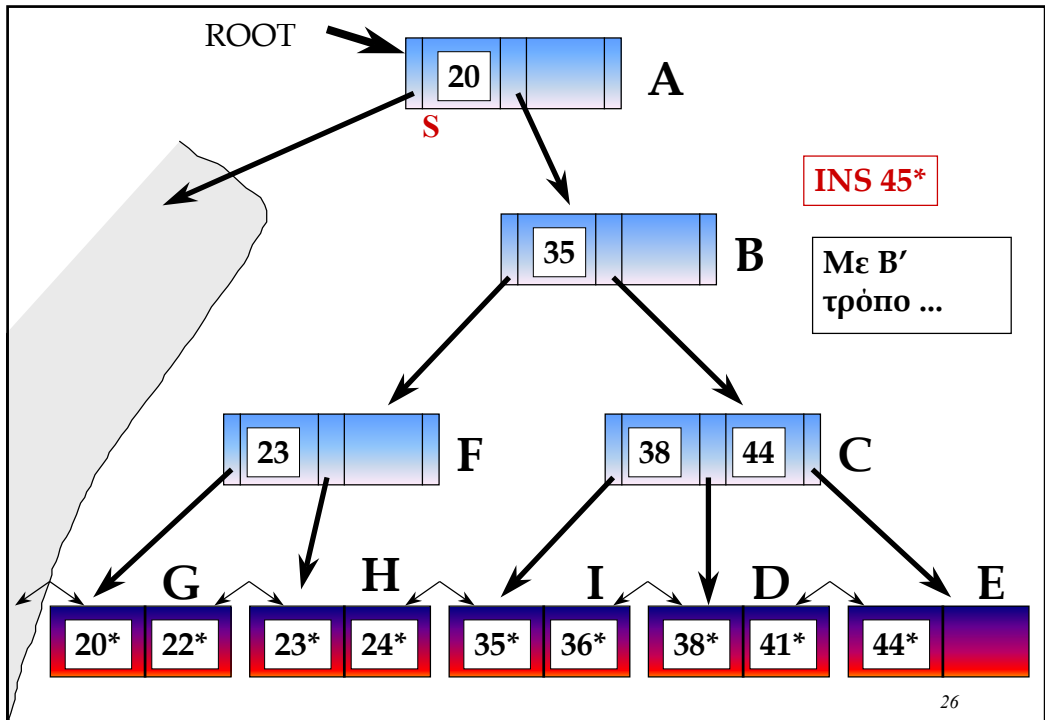
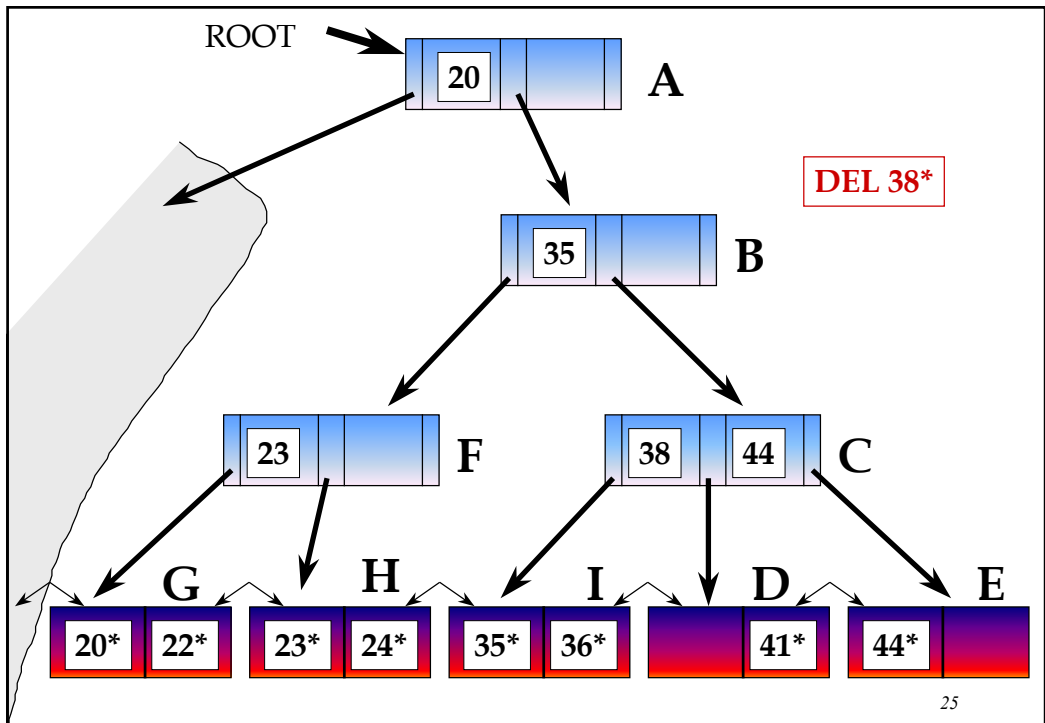
- *A' τρόπος για το «κατά περίπτωση»*
  - X-locks για όλους τους κόμβους
- *B' τρόπος για το «κατά περίπτωση»*
  - S-locks για τους ενδιάμεσους κόμβους
  - X-locks για τα φύλλα
  - Μετατροπή του S-lock σε X-lock αν υπάρχει διάσπαση

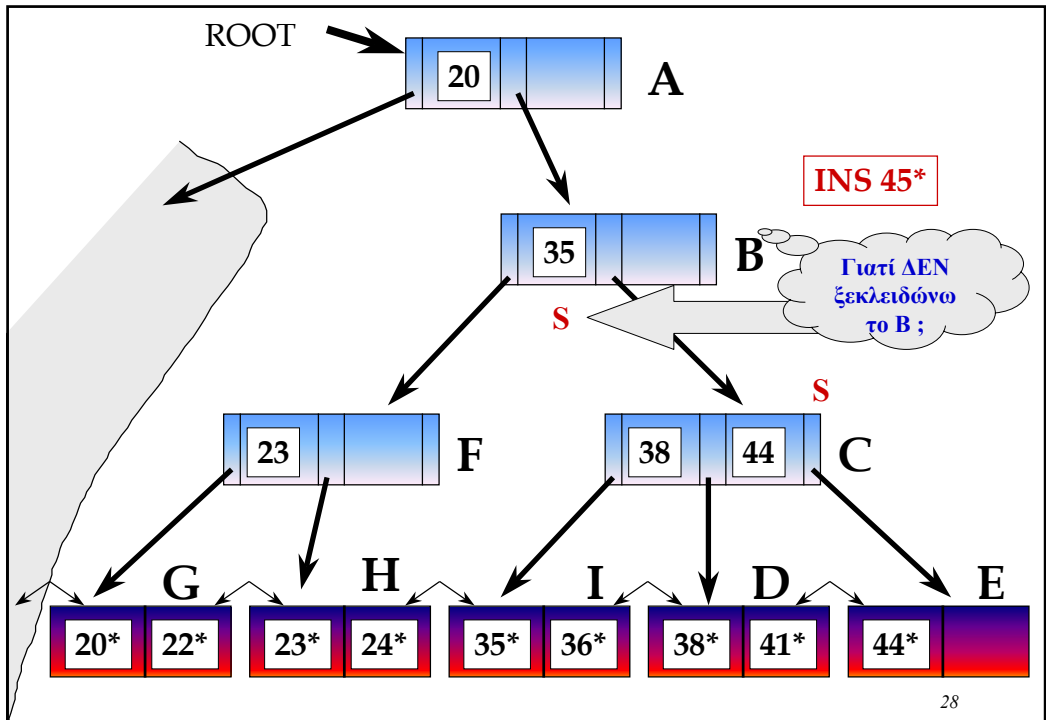
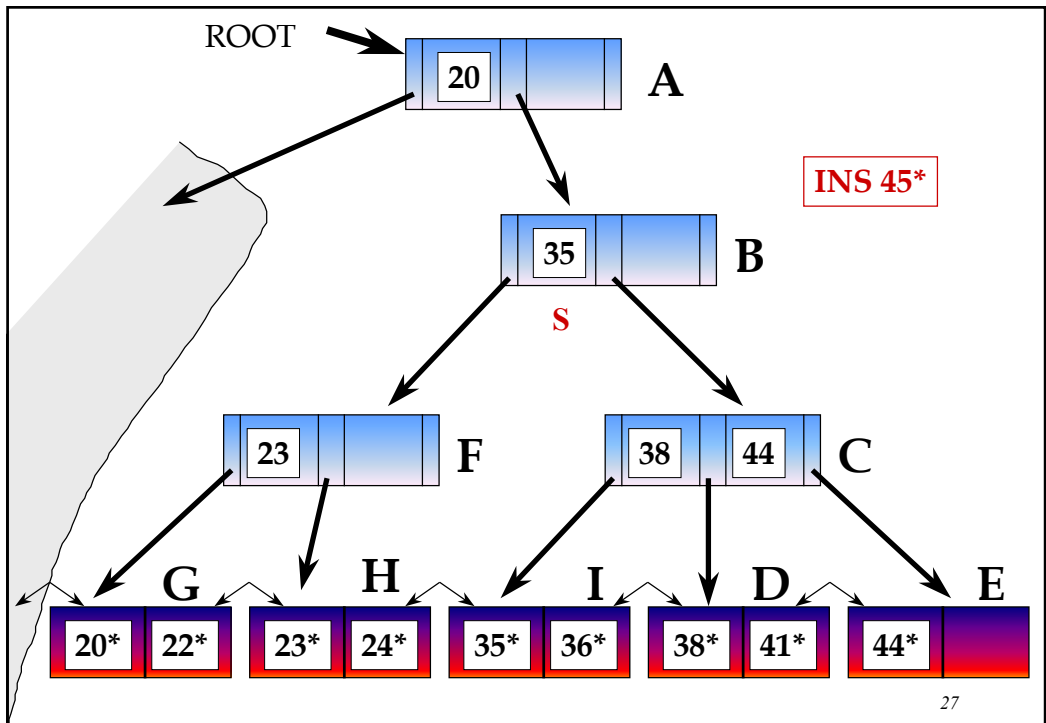
19

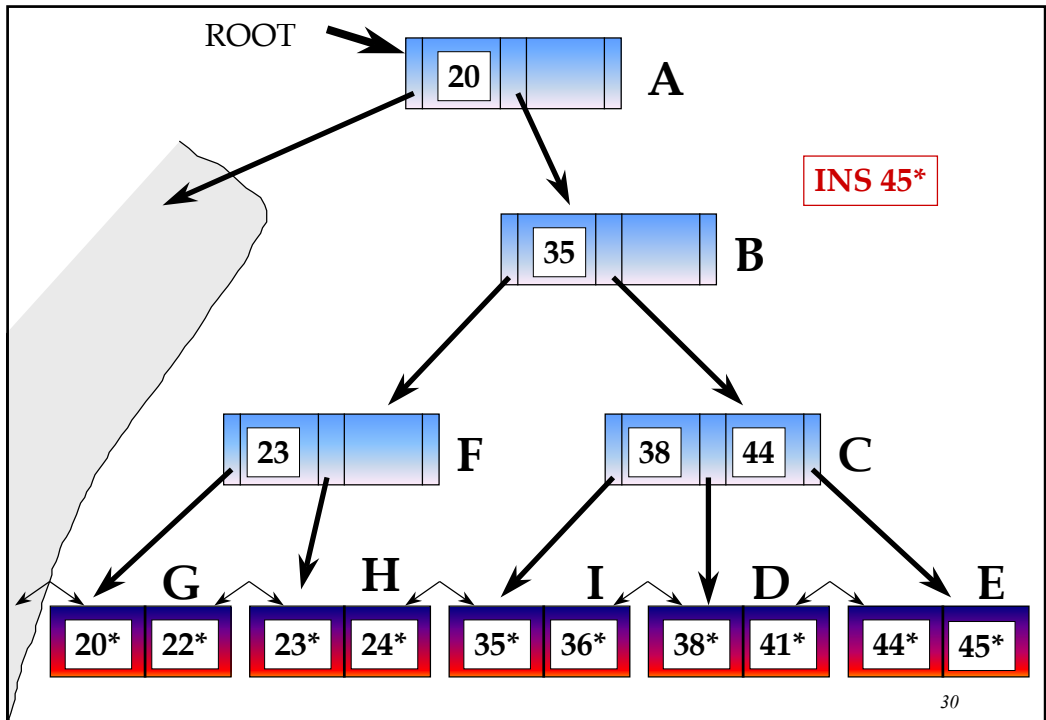
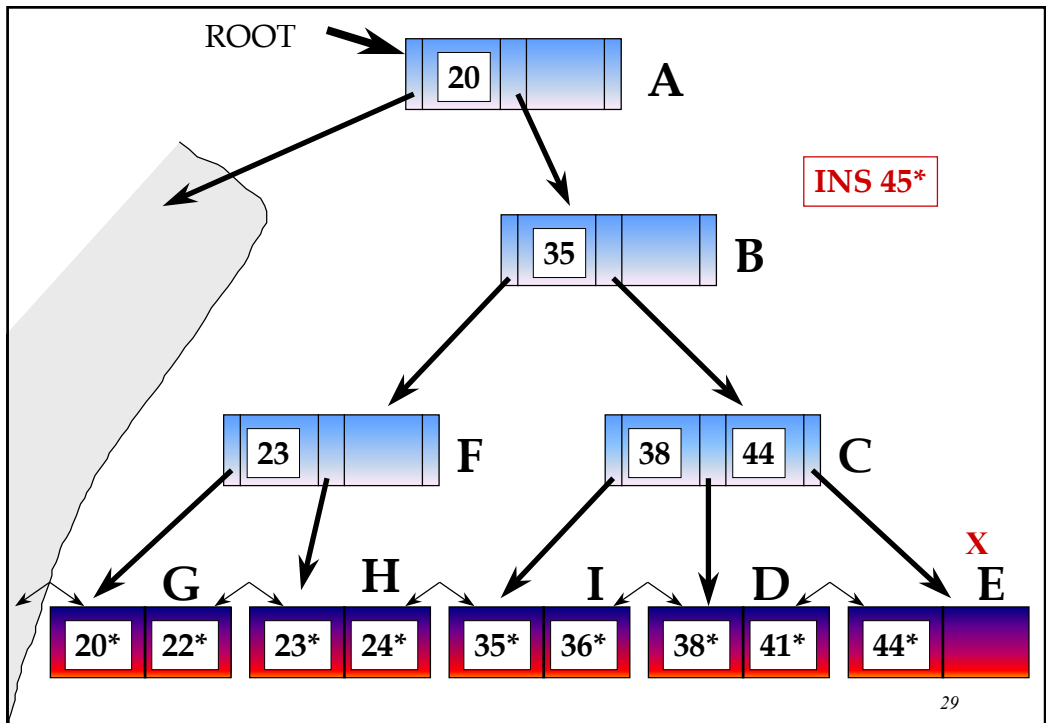












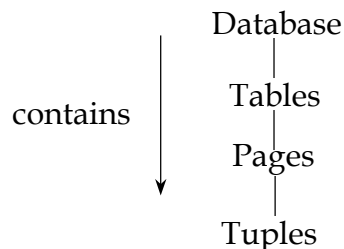
## Θεματολόγιο

- Φαντάσματα
- Κλείδωμα δέντρων
- Κλείδωμα σε διαφορετικά επίπεδα διακριτότητας
- Πώς τα βάζω όλα μαζί
- Αισιόδοξος έλεγχος ταυτοχρονισμού
- Έλεγχος ταυτοχρονισμού με χρονόσημα
- Έλεγχος ταυτοχρονισμού με πολλαπλές εκδόσεις
- Δοσοληψίες και SQL

31

## Τι κλειδώνουμε στην πράξη;

- Μέχρι τώρα θεωρούσαμε εγγραφές
- Στην πράξη, συνήθως μιλάμε για σελίδες
- Μπορούμε να κλειδώσουμε τη βάση σε πολλά «επίπεδα» που ορίζουν μια «ιεραρχία διακριτότητας»



32



## Ορισμοί

- ✦ **Διακριτότητα** (granularity): πόσο μικρό είναι ένα αντικείμενο υπό παρατήρηση
- ✦ Μικρή διακριτότητα = μεγάλο μέγεθος στοιχείου (π.χ., πίνακας ή ΒΔ)
- ✦ Μεγάλη διακριτότητα = μικρό μέγεθος στοιχείου (π.χ., σελίδα ή εγγραφή)
- ✦ **Πρόγονος ενός αντικειμένου**: οποιοδήποτε αντικείμενο στην ιεραρχία διακριτότητας το περικλείει (π.χ., η ΒΔ είναι πρόγονος μιας σελίδας)

33

## Κλείδωμα σε διαφορετικά επίπεδα διακριτότητας

- ✦ Κλείδωμα σε **μικρή** διακριτότητα: **αποκλεισμός** και καθυστέρηση των δοσοληψιών που τρέχουν εν παραλλήλω (π.χ., τι θα γίνει αν κλειδώσω όλη τη ΒΔ;)
- ✦ Κλείδωμα σε **μεγάλη** διακριτότητα: **μεγάλος αριθμός κλειδωμάτων** (χώρος μνήμης, καθυστέρηση διαχείρισης)

*Ερώτηση Κρίσεως: Ποια η καλύτερη διακριτότητα (& υπό ποιες συνθήκες);*

34

## Παράδειγμα

- ✦ T1: UPDATE EMP WHERE EMP\_ID = 1 SET...
- ✦ T2: SELECT \* FROM EMP
- ✦ Η T1 για να γράψει μία και μόνο εγγραφή, μπλοκάρει την T2.
- ✦ **Ιδέα:**
  - ✦ T1: κλειδώνει την εγγραφή της T1
  - ✦ T2: ελέγχει για κλειδώμα όλες τις εγγραφές μία μία;
- ✦ **Καλύτερη Ιδέα:** Μήπως να κλειδώναμε μόνο τη σελίδα που περιέχει την εγγραφή της T1;

35

## Εκμετάλλευση της ιεραρχίας

- ✦ Η ιδέα είναι να κλειδώνουμε με κατάλληλο τρόπο σε διαφορετικά επίπεδα διακριτότητας
- ✦ Χρειαζόμαστε ένα νέο είδος κλειδώματος (γιατί;)
- ✦ **Κλειδώμα πρόθεσης (intention lock):** εφαρμόζεται σε ένα πρόγονο, χαρακτηρίζοντας τι πράξη (για την ακρίβεια: κλειδώμα) θα εφαρμοστεί στον απόγονό του
- ✦ Τριών ειδών κλειδώματα πρόθεσης ...

36

## Κλειδώματα Πρόθεσης

- Ένας πρόγονος (π.χ., σελίδα) λαμβάνει
  - **IS**: αν η δοσοληψία θέλει να **διαβάσει** (πάρει **S-lock**) για κάποιους απογόνους (π.χ., μια εγγραφή)
  - **IX**: αν η δοσοληψία θέλει να **γράψει** (πάρει **X-lock**) για κάποιους απογόνους (π.χ., μια εγγραφή)
  - **SIX**: αν η δοσοληψία θέλει να **διαβάσει** (πάρει S-lock) όλους τους απογόνους του και να **γράψει** (πάρει X-lock) και κάποιους από αυτούς.
- Πριν πάρεις ένα κλειδώμα για κάποιο αντικείμενο, πρέπει να πάρεις κλειδώμα πρόθεσης για όλους τους προγόνους του

37

## Συγκρούσεις

- Ο πίνακας συγκρούσεων γίνεται πλέον:

**T1**: έχει κάποιο lock σε ένα αντικείμενο

**T2**: αιτείται κάποιο lock στο ίδιο αντικείμενο

- Ότι δεν έχει ☒ είναι επιτρεπτό

T1 \ T2	IS	IX	SIX	S	X
IS					☒
IX			☒	☒	☒
SIX		☒	☒	☒	☒
S		☒	☒		☒
X	☒	☒	☒	☒	☒

38

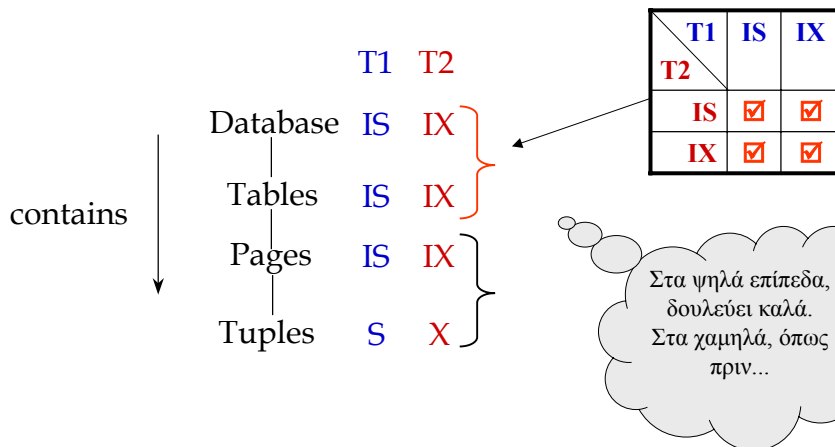
## Παράδειγμα: «σύγκρουση» για μια σελίδα

- **T1**: έχει κλειδώσει με IS τη σελίδα, ώστε να διαβάσει μία (κάποιες) εγγραφή(ες) της
- **T2**: αντιστοίχως, αλλά για να γράψει
- Αν πρόκειται για διαφορετικές εγγραφές, κανείς δεν περιμένει
- Αλλιώς, η σύγκρουση συμβαίνει στο χαμηλότερο επίπεδο με τα κλειδώματα S- & X- locks

T1 \ T2	IS	IX	SIX	S	X
IS					☒
IX	☒		☒	☒	☒
SIX		☒	☒	☒	☒
S		☒	☒		☒
X	☒	☒	☒	☒	☒

39

## Τα κλειδώματα είναι σε όλη την ιεραρχία...



40

## Αλγόριθμος Κλειδωμάτων

- Ξεκινάς στην κορυφή της ιεραρχίας
- Για να πάρω ένα κλείδωμα S- (ή IS-) σε ένα αντικείμενο, πρέπει να έχω IS- (ή IX-) στον πρόγονό του [αλλιώς, περιμένω]
- Για να πάρω ένα κλείδωμα X- (ή IX- ή SIX) σε ένα αντικείμενο, πρέπει να έχω IX- (ή SIX-) στον πρόγονό του [αλλιώς, περιμένω]
- Ξεκλειδώνω με την αντίστροφη σειρά (bottom-up)

41

## Παραδείγματα (και «τι σημαίνει S-lock σε ένα πίνακα;»)

- T1 διαβάζει τον πίνακα R, και **update** κάποιες εγγραφές:
  - T1 παίρνει **SIX-lock** στον R, μετά για κάθε εγγραφή του R παίρνει και ένα **S-lock**, και όπου δη, αναβαθμίζει το S-lock αυτό σε **X-lock** (στα updates στις κρίσιμες εγγραφές, δηλαδή).
- T2 χρησιμοποιεί ένα index για να διαβάσει τμήμα της R:
  - T2 παίρνει **IS** στον R και μετά για κάθε εγγραφή του R παίρνει και ένα **S-lock**.
- T3 διαβάζει όλον τον R: **//ΠΡΟΣΟΧΗ!**
  - T3 παίρνει **S-lock** στο R. **//S-lock σε πίνακα;**
  - ή, η T3 κάνει ότι και η T2 & μπορεί να χρησιμοποιήσει «**κλιμάκωση κλειδώματος**» (lock escalation) για να αποφασίσει αν θα πάρει S lock στο R.

42

## Κλιμάκωση κλειδωμάτων

- ✦ Όταν μια δοσοληψία προσπαθήσει να πάρει κλειδώματα π.χ., για έναν ολόκληρο πίνακα, πέφτουμε στην περίπτωση του προβλήματος χώρου & χρόνου στη διαχείριση των κλειδωμάτων
- ✦ **Κλιμάκωση Κλειδωμάτων:** Αποφασίζουμε ότι πάνω από κάποιο threshold, αντί να κλειδώνω μία μία σελίδα (με S- ή X-lock), κλειδώνω μόνο τον πίνακα (με S- ή X-lock) και τίποτε άλλο.
- ✦ **ΠΡΟΣΟΧΗ:** Αδιέξοδα!!! (αν π.χ., ο ένας είχε SIX και ο άλλος IS στον πίνακα πριν την κλιμάκωση)

43

## Θεματολόγιο

- ✦ Φαντάσματα
- ✦ Κλείδωμα δέντρων
- ✦ Κλείδωμα σε διαφορετικά επίπεδα διακριτότητας
- ✦ Πώς τα βάζω όλα μαζί
- ✦ Αισιόδοξος έλεγχος ταυτοχρονισμού
- ✦ Έλεγχος ταυτοχρονισμού με χρονόσημα
- ✦ Έλεγχος ταυτοχρονισμού με πολλαπλές εκδόσεις
- ✦ Δοσοληψίες και SQL

44

## Το ζήτημα είναι ...

- Όπως είπαμε, υπάρχει το πρόβλημα των φαντασμάτων
- Ο σκοπός είναι να το αποτρέψω και θα χρησιμοποιήσουμε την τεχνική κλειδώματος κατηγορήματος (predicate locking)
- Θα εξετάσουμε με ποιον τρόπο, γίνεται αυτό στην πράξη!

45

## Φαντάσματα

SID	SNAME	RATING	AGE
1	Popay1	1	71
2	Popay2	1	43
3	Popay3	2	80
4	Popay4	2	63
5	X	1	96

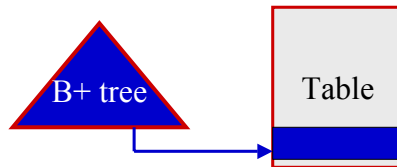


...να κλειδώναμε στην ίδια δοσοληψία όλες τις εγγραφές που πληρούν τη συνθήκη RATING=1...

46

## Κλείδωμα κατηγορήματος / ευρετηρίων

- ✦ Το γεγονός ότι κλειδώνω όλες τις εγγραφές που πληρούν μια συνθήκη (όπως π.χ., RATING = 1)
- ✦ Αν είχαμε ένα index (π.χ., B+ tree) πάνω στο πεδίο rating, θα μπορούσαμε να κλειδώναμε γρήγορα όλες τις εγγραφές με RATING=1.



47

## Αν ΔΕΝ έχω κάποιον index

- ✦ Για να αποφύγω φαντάσματα
  - ✦ SELECT statements: S-lock σε όλον τον πίνακα
  - ✦ UPDATE / DELETE statements: SIX-lock στον πίνακα και X-locks στις κρίσιμες σελίδες με τις υπό τροποποίηση εγγραφές
- ✦ Αν κάποιος πάει να γράψει (X-lock στη σελίδα), θα συγκρουστεί με το εν λόγω κλείδωμα

48



## Αν έχω B+ trees

- Στον **index**, S-locks στα φύλλα όταν κάνω SELECT, αλλιώς X-locks στα φύλλα
- Στον **πίνακα**:
  - SELECT statements: IS-lock στον πίνακα, S-locks στις σελίδες με εγγραφές που ικανοποιούν το WHERE clause
  - UPDATE / DELETE statements: IX-lock στον πίνακα και X-locks στις κρίσιμες σελίδες με τις υπό τροποποίηση εγγραφές
- **Πρώτα κλειδώνω τον index** και μετά τον πίνακα
- Η σύγκρουση εντοπίζεται στον index

49

## Tuning for performance

- Γράψτε **μικρές** δοσοληψίες. Εν ανάγκη, σπάστε τις μεγάλες σε πολλές μικρές δοσοληψίες [το default στην SQL είναι δοσοληψία ανά εντολή]!
- Χρησιμοποιήστε **indexes**. Όχι μόνο ψάχνουμε πιο γρήγορα [αν και γράφουμε πιο αργά] αλλά ο ταυτοχρονισμός αυξάνεται!
- **Αποφύγετε το lock escalation**. Τα νέα DBMS's συνήθως δεν το κάνουν, εκτός κι αν τους το πείτε (RTFM)...
- Το **granule** (επίπεδο διακριτότητας ;) ήταν για καιρό η σελίδα. Σιγά σιγά, κατεβαίνουμε σε επίπεδο **εγγραφής** (π.χ., Oracle 9i)...

50

## Θεματολόγιο

- Φαντάσματα
- Κλείδωμα δέντρων
- Κλείδωμα σε διαφορετικά επίπεδα διακριτότητας
- Πώς τα βάζω όλα μαζί
- Αισιόδοξος έλεγχος ταυτοχρονισμού
- Έλεγχος ταυτοχρονισμού με χρονόσημα
- Έλεγχος ταυτοχρονισμού με πολλαπλές εκδόσεις
- Δοσοληψίες και SQL

51

## Προβλήματα των κλειδωμάτων

- Οι τεχνικές κλειδώματος είναι **συντηρητικές** (αποφεύγονται οι συγκρούσεις)
- **Μειονεκτήματα:**
  - επιβάρυνση (overhead) χειρισμού κλειδώματος
  - έλεγχος προς ανίχνευση αδιεξόδων

52

## Αισιόδοξη εκτέλεση δοσοληψιών

- Κάθε δοσοληψία έχει τρεις φάσεις
  - **ΑΝΑΓΝΩΣΗ**: η δοσοληψία διαβάζει από τη ΒΔ, αλλά τροποποιεί τοπικά αντίγραφα των αντικειμένων σε δικό της χώρο
  - **ΕΠΙΚΥΡΩΣΗ**: έλεγχος για συγκρούσεις – αν υπάρχει σύγκρουση, abort & restart
  - **ΕΓΓΡΑΦΗ**: γράφει τα τοπικά αντίγραφα στη ΒΔ



53

## Πώς δουλεύει ;

- **Χρονόσημο δοσοληψίας**: η στιγμή που ξεκινά η φάση επικύρωσής της
- Έστω ότι έχουμε μια δοσοληψία **T<sub>j</sub>** που θέλει να κάνει Επικύρωση
- Θα την ελέγξουμε σε σχέση με όλες τις δοσοληψίες **T<sub>i</sub>** με χρονόσημο μικρότερο της T<sub>j</sub>.
- Για όλες τις δοσοληψίες T<sub>i</sub>, τ.ω.  $T_i < T_j$ , αρκεί να ισχύει μία από τις παρακάτω συνθήκες ελέγχου
- Αν δεν ισχύει καμία εξ αυτών, η T<sub>j</sub> πρέπει να κάνει ABORT και να επανεκκινηθεί.

54

## 1η συνθήκη ελέγχου

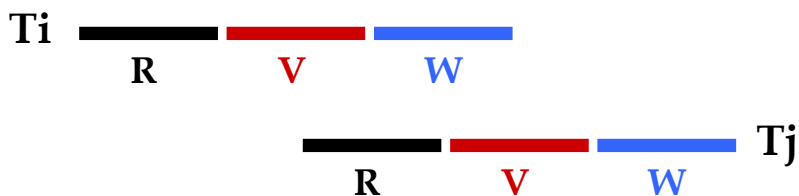
- Η  $T_i$  τελειώνει πριν αρχίσει η  $T_j$ .



*Σειριακή εκτέλεση των δοσοληψιών*

## 2η συνθήκη ελέγχου

- Η  $T_i$  τελειώνει πριν αρχίσει η φάση εγγραφής της  $T_j$
- $WriteSet(T_i) \cap ReadSet(T_j) = \emptyset$



*Η  $T_j$  δεν διαβάζει κανένα αντικείμενο που τροποποιεί η  $T_i$   
Όλες οι εγγραφές της  $T_i$  προηγούνται αυτών της  $T_j$*



## Σχόλια

- Overhead διατήρησης λίστας προσπελασθέντων αντικειμένων
- Αν οι συγκρούσεις είναι σπάνιες έχουμε μεγαλύτερη απόδοση.
- Αν οι συγκρούσεις είναι συχνές, το overhead του abort & restart είναι μεγαλύτερο.

59

## Θεματολόγιο

- Φαντάσματα
- Κλείδωμα δέντρων
- Κλείδωμα σε διαφορετικά επίπεδα διακριτότητας
- Πώς τα βάζω όλα μαζί
- Αισιόδοξος έλεγχος ταυτοχρονισμού
- Έλεγχος ταυτοχρονισμού με χρονόσημα
- Έλεγχος ταυτοχρονισμού με πολλαπλές εκδόσεις
- Δοσοληψίες και SQL

60

## Ορολογία

- ✦ Χρονόσημο δοσοληψίας **TS(T)**: η χρονική στιγμή που ξεκινά η δοσοληψία T
- ✦ Χρονόσημο ανάγνωσης αντικειμένου **RTS(O)**: η χρονική στιγμή της τελευταίας **ανάγνωσης** του **αντικειμένου O** (ασχέτως δοσοληψίας)
- ✦ Χρονόσημο εγγραφής αντικειμένου **WTS(O)**: η χρονική στιγμή της τελευταίας **εγγραφής** του **αντικειμένου O** (ασχέτως δοσοληψίας)

61

## Κεντρική Ιδέα

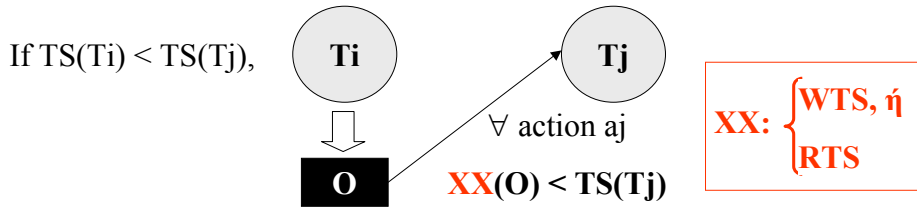
- ✦ Αν η **T<sub>i</sub>** ξεκινά ΠΙΠΙΝ την **T<sub>j</sub>**, όλες οι συγκρούσεις πράξεων μεταξύ **T<sub>i</sub>** και **T<sub>j</sub>**, πρέπει να έχουν την πράξη της **T<sub>i</sub>** (έστω **α<sub>i</sub>**) ΠΙΠΙΝ την πράξη της **T<sub>j</sub>** (έστω **α<sub>j</sub>**)



62

## Κεντρική Ιδέα

- Αντί να συγκρίνω δοσοληψίες, θεωρώ ότι η  $T_i$  διαβάσει/γράφει το αντικείμενο  $O$
- Κάθε φορά, μια δοσοληψία  $T_j$  συγκρίνει το  $TS(T_j)$  με τα χρονόσημα των αντικειμένων

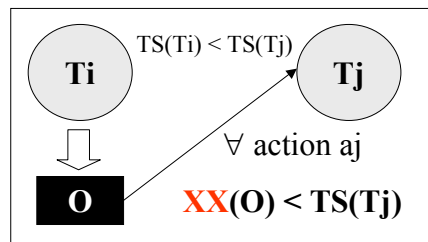


63

## $T_j$ θέλει να κάνει read( $O$ )

- Αν  $WTS(O) > TS(T_j)$  [ήτοι, η  $T_j$  ξεκίνησε πριν γραφτεί το  $O$ ], **abort**  $T_j$  & restart με νέο μεγαλύτερο  $TS$ .
- Αν  $WTS(O) < TS(T_j)$ , επέτρεψε στην  $T_j$  να διαβάσει και  $RTS(O) := \max(RTS(O), TS(T_j))$

Overhead!

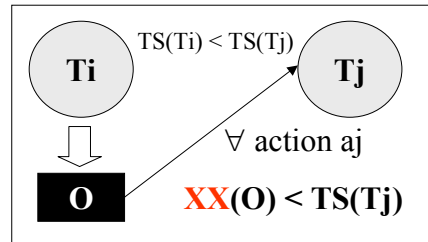


64



## Tj θέλει να κάνει write(O)

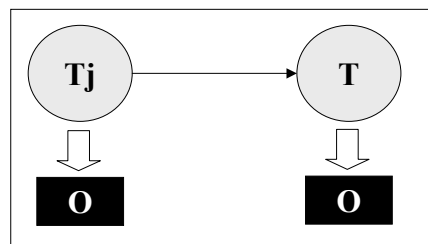
- Αν  $RTS(O) > TS(Tj)$  [ήτοι, η Tj ξεκίνησε πριν διαβαστεί το O], **abort & restart Tj**.
- Αν  $WTS(O) > TS(Tj)$ , ΜΗΝ ΕΚΤΕΛΕΣΕΙΣ ΤΟ write(O), και συνέχισε την Tj με την επόμενη εντολή (Thomas Write Rule)
- Αλλιώς, **επέτρεψε το write(O)**



65

## Thomas write rule

- Αν  $WTS(O) > TS(Tj)$  σημαίνει ότι η Tj ξεκίνησε ΠΡΙΝ από τη δοσοληψία που έγραψε το O (έστω T). Έτσι, η εγγραφή της Tj δεν έχει καμιά αξία (σε ένα σειριακό πρόγραμμα, η T θα ξαναέγραφε πάνω στην τιμή της Tj)
- Thomas Write Rule: Μπορούμε να αγνοήσουμε με ασφάλεια καθυστερημένες εγγραφές!




Tj	T
R(A)	W(A) <b>Commit</b>
W(A) <b>Commit</b>	

66

## ΠΡΟΣΟΧΗ

- Πρέπει η T να έχει κάνει **COMMIT**
- Το παράδειγμα του βιβλίου είναι παραπλανητικό: αντί για T<sub>j</sub> και T έχει T1 και T2. Αυτό που αγνοείται είναι το W(A) της T1!



T1	T2
R(A)	
W(A) Commit	W(A) Commit

67

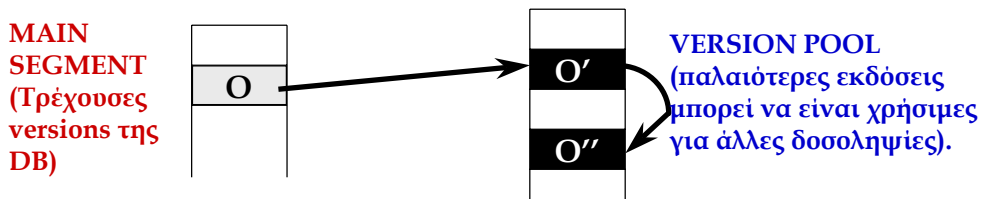
## Θεματολόγιο

- Φαντάσματα
- Κλείδωμα δέντρων
- Κλείδωμα σε διαφορετικά επίπεδα διακρίτοτητας
- Πώς τα βάζω όλα μαζί
- Αισιόδοξος έλεγχος ταυτοχρονισμού
- Έλεγχος ταυτοχρονισμού με χρονόσημα
- Έλεγχος ταυτοχρονισμού με πολλαπλές εκδόσεις
- Δοσοληψίες και SQL

68

## Έλεγχος ταυτοχρονισμού με πολλαπλές εκδόσεις

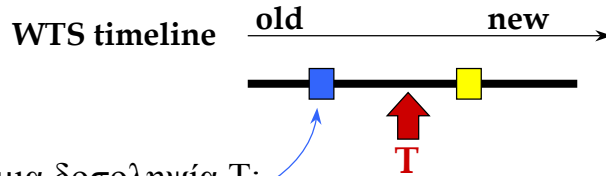
- **Ιδέα:** Κάθε δοσοληψία που γράφει ένα αντικείμενο O, κατασκευάζει μια ιδιωτική της κópια του O που αποθηκεύεται σε μια “Version pool”



## Έλεγχος ταυτοχρονισμού με πολλαπλές εκδόσεις

- Κάθε **δοσοληψία** έχει χρονόσημο **TS** τη στιγμή εκκίνησης της
- Κάθε **version** ενός αντικειμένου έχει **WTS** το χρονόσημο της δοσοληψίας που την έγραψε και **RTS** το χρονόσημο της δοσοληψίας που τη διάβασε τελευταία.
- Οι versions του ίδιου αντικειμένου συνδέονται με pointers σε μια λίστα.

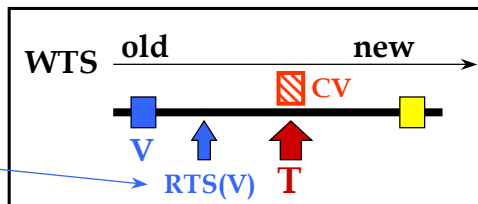
## Ανάγνωση



- ✦ Για να διαβάσει μια δοσοληψία  $T$ :
  - ✦ Βρίσκει την **τελευταία version** με  $WTS < TS(T)$ .  
(Αν δεν της κάνει η τρέχουσα version ψάχνει για προηγούμενες)
- ✦ Αν υποθέσουμε ότι για κάθε αντικείμενο υπάρχει και μια version οι δοσοληψίες που διαβάζουν μόνο (χωρίς να γράφουν ποτέ) δεν αποτυγχάνουν ποτέ.

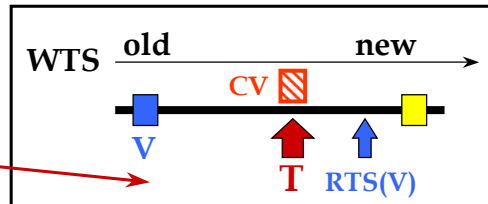
## Εγγραφή

- ✦ Για να γράψει μια δοσοληψία  $T$ :
  - ✦ Βρίσκει τη **νεότερη version**  $V$  τ.ω.  $WTS < TS(T)$ .
  - ✦ Αν  $RTS(V) < TS(T)$ ,  $T$  κάνει μια κόπια **CV** του  $V$ , με  $WTS(CV) = RTS(CV) = TS(T)$  (η εγγραφή αναμένει μέχρι το commit της  $T$  - οι άλλες δοσοληψίες δεν μπορούν να δουν την version **CV**)
  - ✦ **Αλλιώς, abort & restart με νέο TS.**



## Εγγραφή

- Αν  $RTS(V) < TS(T)$ , πάει να πει ότι κάποια  $T'$  διάβασε πριν ξεκινήσει η δοσοληψία  $T \Rightarrow$  η  $T$  μπορεί να γράψει [το πολύ πολύ να κάνει overwrite ότι θα έγραφε η  $T'$ ]
- Αν  $RTS(V) > TS(T)$  προφανώς κάποια  $T'$  διάβασε αφού ξεκίνησε η  $T \Rightarrow$  κάποια πρέπει να κάνει **abort** & επιλέγουμε την  $T$



## Θεματολόγιο

- Φαντάσματα
- Κλείδωμα δέντρων
- Κλείδωμα σε διαφορετικά επίπεδα διακριτότητας
- Πώς τα βάζω όλα μαζί
- Αισιόδοξος έλεγχος ταυτοχρονισμού
- Έλεγχος ταυτοχρονισμού με χρονόσημα
- Έλεγχος ταυτοχρονισμού με πολλαπλές εκδόσεις
- Δοσοληψίες και SQL

## Δοσοληψίες και SQL

- Στην SQL-92 κάθε δοσοληψία έχει:
  - Μέθοδο πρόσβασης: **READ ONLY** vs. **READ WRITE**
  - Ένα από 4 επιτρεπτά επίπεδα απομόνωσης:
    - **READ UNCOMMITTED**
    - **READ COMMITTED**
    - **REPEATABLE READ**
    - **SERIALIZABLE**
- Παράδειγμα χρήσης:  
**SET ISOLATION LEVEL SERIALIZABLE READ WRITE**

75

## Δοσοληψίες και SQL

- **READ UNCOMMITTED:**
  - Δεν υποστηρίζει κλειδώματα.
  - Επιτρέπεται μόνο σε **READ ONLY** δοσοληψίες
- **READ COMMITTED:**
  - Κρατά X-locks για writes, και τα ελευθερώνει στο EOT
  - Κρατά S-locks για reads, και τα ελευθερώνει αμέσως
  - Δεν έχει πρόβλημα με dirty reads (W-R συγκρούσεις), αλλά έχει πρόβλημα σε unrepeatable reads (R-W συγκρούσεις)

76

## Δοσοληψίες και SQL

### ➤ REPEATABLE READ:

- Διαβάζει μόνο ότι προκύπτει από committed δοσοληψίες
- Ελευθερώνει locks στο τέλος (S-locks & X-locks)
- Ευάλωτη σε φαντάσματα

### ➤ SERIALIZABLE:

- Αυστηρό 2PL
- Ανεπηρέαστη από φαντάσματα

77

## SQL-92

Επίπεδο	Dirty Read	Μη επαναλαμβανόμενη ανάγνωση	Πρόβλημα φαντάσματος
Read Uncommitted	Ίσως	Ίσως	Ίσως
Read Committed	Όχι	Ίσως	Ίσως
Repeatable Reads	Όχι	Όχι	Ίσως
Serializable	Όχι	Όχι	Όχι

78

## Σήμερα ...

- ✦ IBM DB2, Informix, Sybase MS SQL Server: Strict 2PL (ή πιο χαλαρό πρωτόκολλο αν δεν ζητηθεί SERIALIZABLE isolation level)
- ✦ Oracle 8: multiversion concurrency control
- ✦ MS SQL Server: επιπλέον, επιτρέπει και optimistic concurrency control

79

## Για εμβάθυνση, εμπέδωση και επέκταση...

- ✦ Hal Berenson, Philip A. Bernstein, Jim Gray, Jim Melton, Elizabeth J. O'Neil, Patrick E. O'Neil: A Critique of ANSI SQL Isolation Levels. SIGMOD Conference 1995

80