

Τεχνικές Ελέγχου Συνδρομικότητας

Τεχνικές Ελέγχου Συνδρομικότητας

- Ο χρήστης δεν ασχολείται με τη συνδρομικότητα
- Το ΣΔΒΔ εξασφαλίζει «σωστή συνδρομικότητα», γενικά *δρομολογεί* τις πράξεις των δοσοληψιών ώστε να προκύπτουν χρονοπρογράμματα σειριοποιησίμα βάσει συγκρούσεων
- Μέσω τεχνικών ελέγχου συνδρομικότητας

Δηλαδή, ψάχνουμε αλγόριθμους (πρωτόκολλα) που θα δρομολογούν τις πράξεις των δοσοληψιών

Τεχνικές Ελέγχου Συνδρομικότητας

Τεχνικές

1. **Κλειδώματος (locking)** για να αποτρέψουν τη συνδρομική (ταυτόχρονη) προσπέλαση των δεδομένων από πολλές δοσοληψίες
2. **Διάταξης χρονοσημάτων (timestamps)**
3. **Πιστοποίησης (validation)** μιας δοσοληψίας (αισιόδοξα πρωτόκολλα)

Τεχνικές Κλειδώματος

Η πιο απλή εκδοχή: ένα κλειδί ανά δεδομένο

- μια δοσοληψία πριν προσπελάσει ένα δεδομένο X ζητά ένα κλειδί -- αίτηση **lock(X)**
- μπορεί να προσπελάσει το δεδομένο, μόνο αφού της δοθεί το κλειδί -- *τότε παίρνει το κλειδί*;
- μια δοσοληψία μπορεί να άρει το κλειδί στο δεδομένο -- αίτηση **unlock(X)**

Τεχνικές Κλειδώματος

ένα απλό κλειδωμα:

στην πιο απλή περίπτωση, ένα **μόνο είδος κλειδιού**

• **lock(X)** :: πραγματοποιείται αν το δεδομένο δεν είναι ήδη κλειδωμένο, αλλιώς η δοσοληψία περιμένει μέχρι να ελευθερωθεί το δεδομένο

• μια δομή (πίνακας) (δεδομένο, μια ένδειξη (κλειδωμένο - μη-κλειδωμένο), ουρά με δοσοληψίες που περιμένουν)

• **unlock(X)**

Τεχνικές Κλειδώματος

Δυο προβλήματα:

1. Δεν επιτρέπει ταυτόχρονες αναγνώσεις.
2. Δε δουλεύει! (δηλαδή, δεν αρκεί για να δώσει σειριοποιησίμα χρονοπρογράμματα)
Χρειάζεται (όπως θα δούμε) διάταξη των πράξεων lock-unlock κάθε δοσοληψίας

Ας διορθώσουμε το πρώτο

Στόχος: πολλές δοσοληψίες να μπορούν να διαβάσουν ένα δεδομένο ταυτόχρονα

Δύο ειδών κλειδιά:

- διαμοιραζόμενο (shared) κλειδί ή κλειδί ανάγνωσης
- αποκλειστικό (exclusive) κλειδί ή κλειδί εγγραφής

• μια δοσοληψία πριν **διαβάσει** ένα δεδομένο X ζητά ένα διαμοιραζόμενο κλειδί -- αίτηση **S-lock(X)**

• μια δοσοληψία πριν **γράψει** ένα δεδομένο X ζητά ένα αποκλειστικό κλειδί -- αίτηση **X-lock(X)**

• η αίτηση για κλειδί δίνεται αν δεν υπάρχει «συγκρούμενο κλειδί»

• (πάλι) μια δοσοληψία μπορεί να άρει το κλειδί στο δεδομένο -- αίτηση **unlock (X)**

Πίνακας συμβατότητας κλειδιών

	S-Lock(X)	X-Lock(X)
S-Lock(X)	✓	
X-Lock(X)		

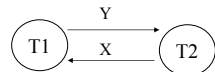
T1
S-Lock(Y)
R₁(Y)
Unlock(Y)

T2

S-Lock(X)
R₂(X)
Unlock(X)
X-Lock(Y)
W₂(Y)
Unlock(Y)
C₂

Δεν αρκεί για σεριοποιησιμότητα

S: R₁(Y) R₂(X) W₂(Y) C₂ W₁(X) C₁



Λύση: Κλειδωμα Δύο Φάσεων

S-Lock(X)
W₁(X)
Unlock(X)
C₁

- Πρωτόκολλο κλειδώματος δυο φάσεων (Two-Phase Locking 2PL)

Όλες οι πράξεις (αιτήσεις) κλειδώματος μιας δοσοληψίας προηγούνται της πρώτης πράξης (αίτησης) άρσης κλειδώματος της διαδικασίας

Δηλαδή, μόλις μια δοσοληψία αφήσει (unlock) ένα κλειδί δεν μπορεί να ζητήσει ξανά κλειδί

Κάθε δοσοληψία δυο φάσεις

- μια φάση επέκτασης ή εξάπλωσης
- μια φάση συρρίκνωσης

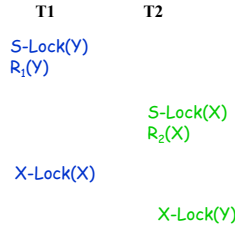
Αποδεικνύεται ότι είναι σωστό

Τροποποίηση ώστε να είναι αυστηρό;

Παρατηρήσεις

- Οι αιτήσεις lock και unlock πρέπει να είναι ατομικές πράξεις
- **Αναβάθμιση κλειδιού:** μια δοσοληψία που κατέχει ένα διαμοιραζόμενο κλειδί μπορεί να αναβαθμιστεί ώστε να κατέχει ένα αποκλειστικό κλειδί

Οι τεχνικές κλειδώματος μπορεί να προκαλέσουν αδιέξοδα (deadlocks)



Η T1 περιμένει την T2 να ελευθερώσει το X, και η T2 περιμένει την T1 να ελευθερώσει το Y

Δυο τεχνικές:

- **Πρωτόκολλα Πρόληψης Αδιεξόδων (Deadlock Prevention):** Αποφυγή δημιουργίας αδιεξόδου
- **Πρωτόκολλα Ανίχνευσης Αδιεξόδου (Deadlock Detection):** Ελέγχουμε περιοδικά αν το σύστημα βρίσκεται σε κατάσταση αδιεξόδου

- Κάθε δοσοληψία πρέπει να κλειδώνει *όλα* τα δεδομένα που χρειάζεται πριν ξεκινήσει
- Αν δε μπορεί να κλειδώσει έστω και ένα, δε κλειδώνει κανένα και προσπαθεί ξανά

«σπάσιμο» του κύκλου ⇒ κάποια διάταξη μεταξύ των δοσοληψιών

Κάθε δοσοληψία T έχει ένα χρονόσημο TS(T):
 Μια διαδικασία παίρνει ένα χρονόσημο κατά την εκκίνησή της.
 TS(T1) < TS(T2), σημαίνει ότι η T1 ξεκίνησε πριν την T2

ιδέα: μια δοσοληψία **περιμένει** μόνο αν το κλειδί το έχει μια δοσοληψία με **μικρότερο** (μεγαλύτερο) χρονόσημο, αλλιώς **ακυρώνεται**

Δύο σχήματα

- αναμονής-θανάτωσης
- τραυματισμού-αναμονής

Και στα δύο σχήματα οι παλιές δοσοληψίες «εκτοπίζουν» τις νεώτερες

Πρόληψη Αδιεξόδων

Έστω ότι η T_i ζητά να κλειδώσει το X που είναι κλειδωμένο από την T_j

- αναμονή-θανάτωση

Αν $TS(T_i) < TS(T_j)$
 T_i περιμένει (* περιμένουμε αν το κλειδί το έχει νεώτερη *)
 Αλλιώς (* το κλειδί το έχει παλαιότερη *)
 ακυρώνεται (πεθαίνει!) η T_i και επανεκκινείται με το ίδιο (γιατί;) χρονόσημα

- τραυματισμός -αναμονή

Αν $TS(T_i) > TS(T_j)$
 T_i περιμένει (* περιμένουμε αν το κλειδί το έχει παλαιότερη *)
 Αλλιώς (* το κλειδί το έχει νεώτερη *)
 ακυρώνεται (τραυματίζεται) η T_j και επανεκκινείται με το ίδιο χρονόσημα

Ανίχνευση Αδιεξόδων

Κατασκευή γράφου αναμονής (wait-for graph)

κόμβοι : δοσοληψίες

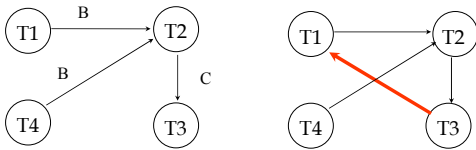
ακμή από τον κόμβο T_i στον T_j , αν η T_i περιμένει την T_j να αφήσει ένα κλειδί

- Περιοδικά έλεγχος για κύκλους στο γράφο αναμονής

Ανίχνευση Αδιεξόδων

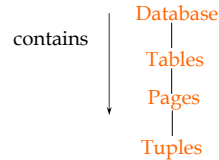
Παράδειγμα:

T1: S-Lock(A), R(A), X-Lock(B), W(B)
 T2: S-Lock(B), X-Lock(C)
 T3: S-Lock(C), R(C)
 T4: X-Lock(A), X-Lock(B)



Διακριτότητα

⌘ Δύσκολο να αποφασίσουμε τη διακριτότητα (granularity) για το κλειδωμα



Τεχνικές Ελέγχου Συνδρομικότητας

Τεχνικές

✓ 1. Κλειδώματος (locking)

→ 2. Διάταξης χρονοσημάτων (timestamps)

3. Πιστοποίησης (validation)

Διάταξη Χρονοσημάτων

Το χρονόσημα δημιουργείται από το ΣΔΒΔ και προσδιορίζει μοναδικά μια δοσοληψία

Ιδέα: διάταξη των δοσοληψιών με βάση το χρονόσημα τους (δηλαδή, χρονοπρόγραμμα ισοδύναμο με σειριακό στο οποίο οι δοσοληψίες εμφανίζονται διατεταγμένες με βάση τις τιμές των χρονοσημάτων)

⇒ άρα η σειρά προσπέλασης στα δεδομένα πρέπει να μη παραβιάζει τη σειριοποιησιμότητα

Δηλαδή: αν μια πράξη a_i μιας δοσοληψίας T_i συγκρούεται με μια πράξη a_j μιας δοσοληψίας T_j και $TS(T_i) < TS(T_j)$, τότε η a_i πρέπει να προηγείται της a_j . Αλλιώς, restart τη δοσοληψία.

Κάθε δεδομένο X έχει δύο τιμές χρονοσημάτων:

$ΧΣΑ(X)$ (χρονόσημα ανάγνωσης) το μεγαλύτερο μεταξύ όλων των χρονοσημάτων των δοσοληψιών που διάβασαν το X (διαισθητικά, η πιο πρόσφατη που το διάβασε)

$ΧΣΕ(X)$ (χρονόσημα εγγραφής) το μεγαλύτερο μεταξύ όλων των χρονοσημάτων των δοσοληψιών που έγραψαν το X

Η δοσοληψία T με $ΧΣ(T)$ εκτελεί μια πράξη **ανάγνωσης $R(X)$**

Αν $ΧΣ(T) < ΧΣΕ(X)$ (αυτό παραβιάζει τη διάταξη)

η T ακυρώνεται,

μπορεί να ξαναρχίσει αλλά με μεγαλύτερο χρονόσημα (γιατί:)

Αν $ΧΣ(T) > ΧΣΕ(X)$

η ανάγνωση είναι επιτρεπτή

θέσε το $ΧΣΑ(X) = \max\{ΧΣΑ(T), ΧΣ(TA)\}$

• Οι αλλαγές στο $ΧΣΑ(X)$ πρέπει να γράφονται στο δίσκο! Αυτό και το ότι η δοσοληψίες ξαναρχίζουν προκαλεί overhead

Η δοσοληψία T με $ΧΣ(T)$ εκτελεί μια **πράξη εγγραφής $W(X)$**

Αν $ΧΣΑ(X) > ΧΣ(T)$ ή $ΧΣΕ(X) > ΧΣ(T)$

η T ακυρώνεται (γιατί:)

Βελτιστοποίηση: τι σημαίνει $ΧΣΕ(X) > ΧΣ(T)$

Ο **κανόνας του Thomas για εγγραφές** (Thomas Write Rule) Μπορούμε να αγνοήσουμε μερικές «ξεπερασμένες» ή **τυφλές** εγγραφές, δε χρειάζεται επανεκκίνηση της T (η εγγραφή της T ακολουθείται από άλλη εγγραφή, χωρίς ενδιάμεση ανάγνωση)

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

Επιτρέπει σειριοποίηση - αλλά όχι σειριοποίηση βάσει συγκρούσεων

• Δυστυχώς, παράγει και χρονοπρογράμματα χωρίς δυνατότητα ανάκαμψης

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

⊗ Τροποποίηση

☑ Buffer all writes μέχρι την επικύρωση του writer (αλλά το $ΧΣΕ(X)$ τροποποιείται κανονικά)

☑ Block readers T (όπου $ΧΣ(T) > ΧΣΕ(X)$) μέχρι να επικυρωθεί ο writer του X

⊗ Όμοια με το να κρατούν οι writers X-Locks μέχρι την επικύρωση τους αλλά όχι ακριβώς 2PL

Τεχνικές

- ✓ 1. Κλειδώματος (locking)
- ✓ 2. Διάταξης χρονοσημάτων (timestamps)
- 3. Πιστοποίησης (validation)

Οι τεχνικές κλειδώματος είναι συντηρητικές (αποφεύγονται οι συγκρούσεις)

Μειονεκτήματα

- επιβάρυνση (overhead) χειρισμού κλειδώματος
- αποφυγή/ανίχνευση αδιεξόδων
- lock contention για τα δεδομένα που χρησιμοποιούνται συχνά

Αν οι συγκρούσεις είναι σπάνιες, μεγαλύτερη συγχρονικότητα, αν αντί για κλειδωμα, έλεγχος για συγκρούσεις όταν μια δοσοληψία επικυρώνεται (commits)

Κάθε δοσοληψία έχει τρεις φάσεις

- **ΑΝΑΓΝΩΣΗ**: η δοσοληψία διαβάζει από τη βδ, αλλά τροποποιεί προσωπικά αντίγραφα των δεδομένων
- **ΠΙΣΤΟΠΟΙΗΣΗ**: έλεγχος για συγκρούσεις
- **ΕΓΓΡΑΦΗ**: γράφει τα τοπικά αντίγραφα στη βδ

Έλεγχος συνθηκών που είναι ικανές για να εξασφαλίσουν ότι δεν υπήρχαν συγκρούσεις

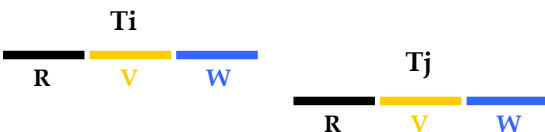
- Κάθε δοσοληψία T έχει ένα μοναδικό αριθμό TID (χρονόσημα)
- Το TID ανατίθεται στο τέλος της φάσης ΑΝΑΓΝΩΣΗΣ (ακριβώς πριν αρχίσει η πιστοποίηση)
- Με κάθε δοσοληψία

ReadSet(T): το σύνολο των δεδομένων που διάβασε η T

WriteSet(T): το σύνολο των δεδομένων που διάβασε η T

ΕΛΕΓΧΟΣ Περίπτωση 1

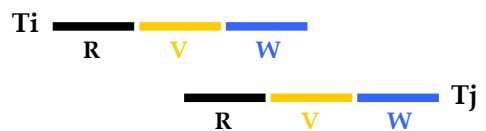
⌘ Για όλα τα i και j τέτοια ώστε $T_i < T_j$, η T_i τελειώνει πριν αρχίσει η T_j .



ΕΛΕΓΧΟΣ Περίπτωση 2

⌘ Για όλα τα i και j τέτοια ώστε $T_i < T_j$:

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



ΕΛΕΓΧΟΣ Περίπτωση 3

- ⌘ Για όλα τα i και j τέτοια ώστε $T_i < T_j$:
 - η T_i τελειώνει τη φάση ανάγνωσης πριν αρχίσει η φάση ανάγνωσης της T_j
 - $WriteSet(T_i) \cap ReadSet(T_j) = \emptyset$
 - $WriteSet(T_i) \cap WriteSet(T_j) = \emptyset$



- ⌘ Πιστοποίηση της T (έλεγχος 1 & 2):

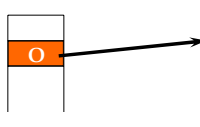
```

valid = true;
// S = set of Xacts that committed after Begin(T)
< foreach Ts in S do {
  if ReadSet(Ts) does intersect WriteSet(T)
    then valid = false;
}
if valid then { install updates; // Write phase
                Commit T }
else Restart T
    
```

τέλος κρίσιμης περιοχής

Κάθε εγγραφή παράγει ένα καινούργιο αντίγραφο ενώ οι αναγνώστες διαβάζουν ένα κατάλληλο παλιό

ΚΥΡΙΟ ΤΜΗΜΑ (Τρέχουσες εκδόσεις των αντικειμένων της ΒΔ)



ΣΥΛΛΟΓΗ ΕΚΔΟΣΕΩΝ VERSION POOL (Παλιές εκδόσεις που μπορεί να είναι χρήσιμες σε κάποιους ενεργούς αναγνώστες)

Οι αναγνώστες μπορούν πάντα να προχωρήσουν -- αλλά μπορεί να χρειαστεί να περιμένουν μέχρι την επικύρωση των εγγραφών

Διατήρηση παλαιότερων εκδοχών, χρήσιμη ως ιστορικό εξέλιξης των τιμών (χρονικές (temporal) βάσεις δεδομένων)

Βασική ιδέα:

οι αναγνώστες διαβάζουν κατάλληλες εκδόσεις ώστε να διατηρήσουν τη σειριοποιησιμότητα

Παραλλαγές των τεχνικών ελέγχου συνδρομικότητας

Θα δούμε με χρονοσχήματα

- Κάθε έκδοση ενός αντικειμένου έχει ως XSE το $XΣ$ της T που την έγραψε και ως XSA το $XΣ$ της T που διάβασε αυτήν την έκδοση πιο πρόσφατα

Μια πράξη εγγραφής με δσοληψία T δημιουργεί μια καινούργια έκδοση του δεδομένου, με XSA και XSE ίσο με $XΣ(T)$

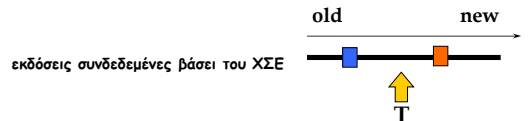
- Οι εκδόσεις συνδέονται (chained backward) μπορούμε να σήσουμε εκδόσεις που είναι πολύ «παλιές για να έχουν ενδιαφέρον»

- Κάθε δσοληψία: **Reader** ή **Writer**.
 - Writer μπορεί να γράψει κάποιο αντικείμενο; Reader μόνο διαβάζει.
 - Κάθε δσοληψία δηλώνει αν είναι Reader όταν ξεκινά.

Στόχος είναι η Readers να μην αποτυγχάνουν

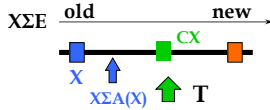
Reader T

- Για την ανάγνωση ενός αντικειμένου:
 - Βρες την πιο νεώτερη έκδοση με $XSE < XΣ(T)$ (Η αναζήτηση ξεκινά από την πιο πρόσφατη - κοίτα σχήμα)
 - Αν υποθέσουμε ότι όλες οι εκδόσεις υπάρχουν, μια Reader T δε χρειάζεται να επανηγηθεί.



Writer T

- Για την ανάγνωση, ακολούθησε το πρωτόκολλο του Reader
- Για να γράφεις ένα αντικείμενο X:
 - Βρες τη νεότερη έκδοση με $XΣΕ(X) < XΣΕ(T)$
 - Αν $XΣA(X) < XΣ(T)$, η T δημιουργεί ένα αντίγραφο CX του X, με ένα δείκτη στο X, και $XΣΕ(CX) = XΣ(T)$, $XΣA(CX) = XΣ(T)$. (Οι εγγραφές γίνονται buffered μέχρι την επικύρωση της T, οι άλλες δοσοληψίες μπορούν να δουν τις τιμές των XΣ αλλά δεν μπορούν να διαβάσουν την έκδοση CX μέχρι η T να επικυρωθεί.)
 - Αλλιώς, απόρριψη.



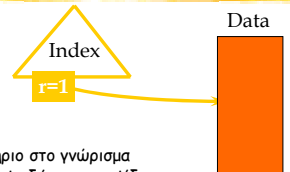
Αν επιτρέψουμε εγγραφές και διαγραφές στοιχείων, τα πρωτόκολλα δε δουλεύουν σωστά (ούτε το αυστηρό 2PL)

```
Sailors(sid, sname, rating, age)
Boats(bid, bname, color)
Reserves(sid, bid, day)
```

- T1 κλειδώνει όλες τις σελίδες που περιέχουν εγγραφές sailor με *rating* = 1, και βρίσκει τον πιο ηλικιωμένο (έστω, *age* = 71).
- Μετά, η T2 εισάγει ένα νέο sailor: *rating* = 1, *age* = 96.
- T2 επίσης διαγράφει τον πιο ηλικιωμένο ναυτικό με *rating* = 2 (έστω, *age* = 80), και επικυρώνεται.
- T1 τώρα κλειδώνει όλες τις σελίδες που περιέχουν εγγραφές sailor με *rating* = 2, βρίσκει τον πιο ηλικιωμένο (έστω, *age* = 63).
- Δεν υπάρχει ισοδύναμη με την παραπάνω σειριακή εκτέλεση

- Η T1 έμμεσα υποθέτει ότι έχει κλειδώσει το σύνολο όλων των εγγραφών με *rating* = 1.
- Η υπόθεση ισχύει μόνο αν δεν προστεθούν εγγραφές ενώ εκτελείται η T1
- Χρειάζεται κάποιος μηχανισμός για να το επιβάλει: **κλειδωμα ευρετηρίου** (index locking) και **κλειδωμα συνθηκών** (predicate locking).

Τα παράδειγμα δείχνει ότι η σειριοποιησιμότητα βάσει συγκρούσεων δίνει σειριοποιησιμότητα μόνο όταν τα αντικείμενα στη βδ είναι σταθερά



Κλειδωμα Ευρετηρίου

- Αν υπάρχει ένα πυκνό ευρετήριο στο γνώρισμα *rating*, η T1 θα μπορούσε να κλειδώσει τη σελίδα του ευρετηρίου που περιέχει τις εγγραφές με *rating* = 1.
- Αν δεν υπάρχει καμία εγγραφή με *rating* = 1

Κλειδωμα Συνθήκης

- Κλειδώσε όλες τις εγγραφές που ικανοποιούν κάποια συνθήκη λογικής, π.χ., *age* > 2**salary*.
- Το κλειδωμα ευρετηρίου είναι μια ειδική μορφή κλειδώματος συνθήκης που το ευρετήριο επιτρέπει αποδοτική υλοποίηση του κλειδώματος συνθήκης

Ιεραρχική δομή

Νέα είδη κλειδίων IS (πρόθεση διαμοιραζόμενου) IX (πρόθεση αποκλειστικό)

Για να κλειδώσει με S (X) πρέπει να κλειδώσει όλους τους προγόνους με IS (IX) αρχίζοντας από τη ρίζα

Συχνά IX και S (διάγραμμα όλου, τροποποίηση ενός) είδος κλειδιού SIX X

	IS	IX	S	SIX	X
IS	T	T	T	T	
IX	T	T			
S	T		T		
SIX	T				
X					

Κλειδωμα Πολλαπλής Κλιμάκωσης

Δύο φάσεις
Σειρά αποδέσμευσης κλειδίων:

Υποστήριξη δοσοληπιών στην SQL-92

Δοσοληψία αυτόματα με προτάσεις που τροποποιούν είτε τη βάση δεδομένων είτε το σχήμα

Μέθοδος πρόσβασης: READ ONLY ή READ WRITE

Επίπεδο απομόνωσης: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE (αυστηρό 2PL)

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE READ ONLY
default: SERIALIZABLE, READ WRITE

Υποστήριξη δοσοληπιών στην SQL-92

Isolation Level	Dirty Read	Unrepeatable Read	Phantom Problem
Read Uncommitted	Maybe	Maybe	Maybe
Read Committed	No	Maybe	Maybe
Repeatable Reads	No	No	Maybe
Serializable	No	No	No

Αναλυτικά στο βιβλίο

Υποστήριξη δοσοληπιών

IBM DB2, Informix, Microsoft SQL server, Sybase Αυστηρό 2PL (με παραλλαγές για επίπεδα απομόνωσης λιγότερα αυστηρά του SERIALIZABLE)

Microsoft SQL server: χρονόσημα εγγραφής

Oracle 8: πολλαπλές εκδόσεις, ώστε οι αναγνώσεις δεν περιμένουν ποτέ

Κλειδωμα πολλαπλής κλιμάκωσης σε επίπεδο πίνακα, σελίδας και εγγραφής