
Προηγμένα Θέματα Βάσεων Δεδομένων

Διδάσκων: Άγγελος Μιχάλας

Περιεχόμενα

- Συναλλαγές
- Έλεγχος ταυτοχρονισμού
- Επανάκτηση δεδομένων

Πρωτόκολλο κλειδώματος

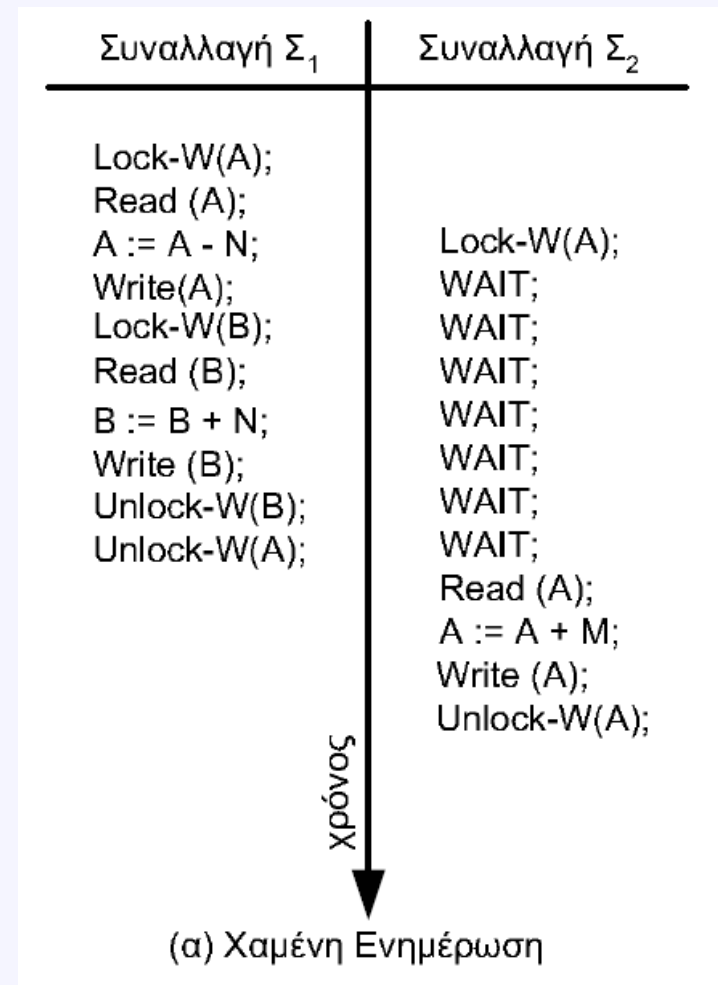
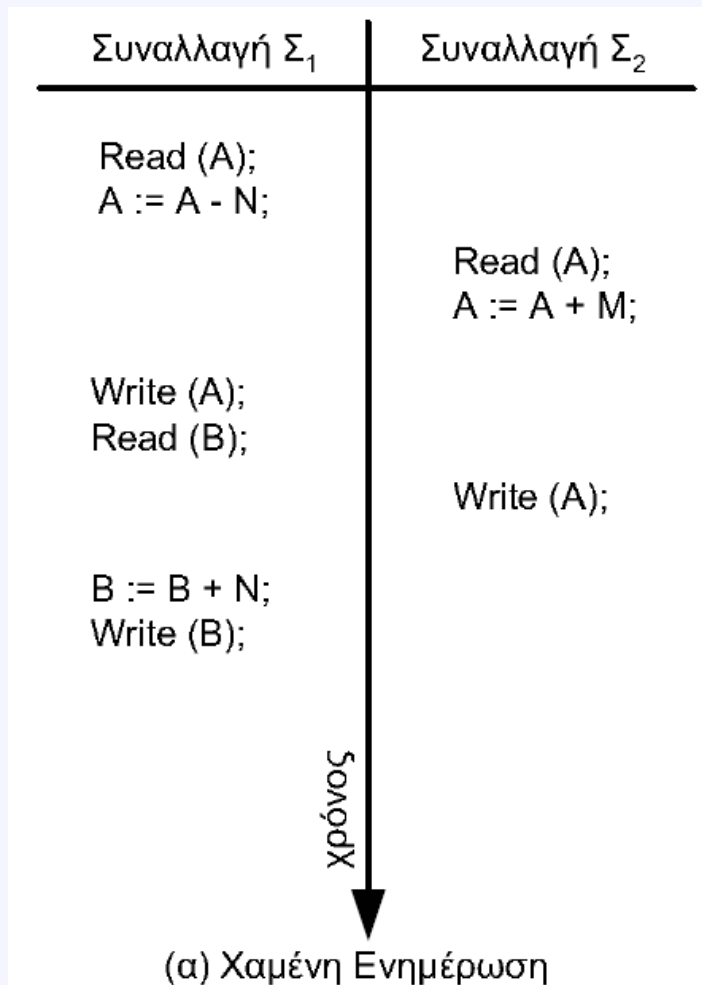
- Ένας μηχανισμός κλειδώματος πρέπει να ικανοποιεί κανόνες εγγύησης της συνέπειας των δεδομένων.
 - Η αυθαίρετη δέσμευση ή απελευθέρωση κλειδαριών μπορεί να δημιουργήσει σοβαρά προβλήματα στη συνέπεια των δεδομένων
- Οι κανόνες αυτοί ορίζουν ένα **πρωτόκολλο κλειδώματος** (locking protocol) και εγγυώνται ότι το χρονοδιάγραμμα που προκύπτει είναι σειριοποιήσιμο.

Πρωτόκολλο κλειδώματος

- **Πρωτόκολλο Κλειδώματος Δύο Φάσεων** (two-phase locking) ορίζει 2 φάσεις λειτουργίας μιας συναλλαγής.
 - **φάση ανάπτυξης** (growing phase), όπου η συναλλαγή ζητά τη χορήγηση κλειδαριών και δεν επιτρέπεται η απελευθέρωση κλειδαριών,
 - **φάση συρρίκνωσης** (shrinking phase), όπου η συναλλαγή απελευθερώνει κλειδαριές και δεν επιτρέπεται να ζητήσει τη χορήγηση νέων κλειδαριών.

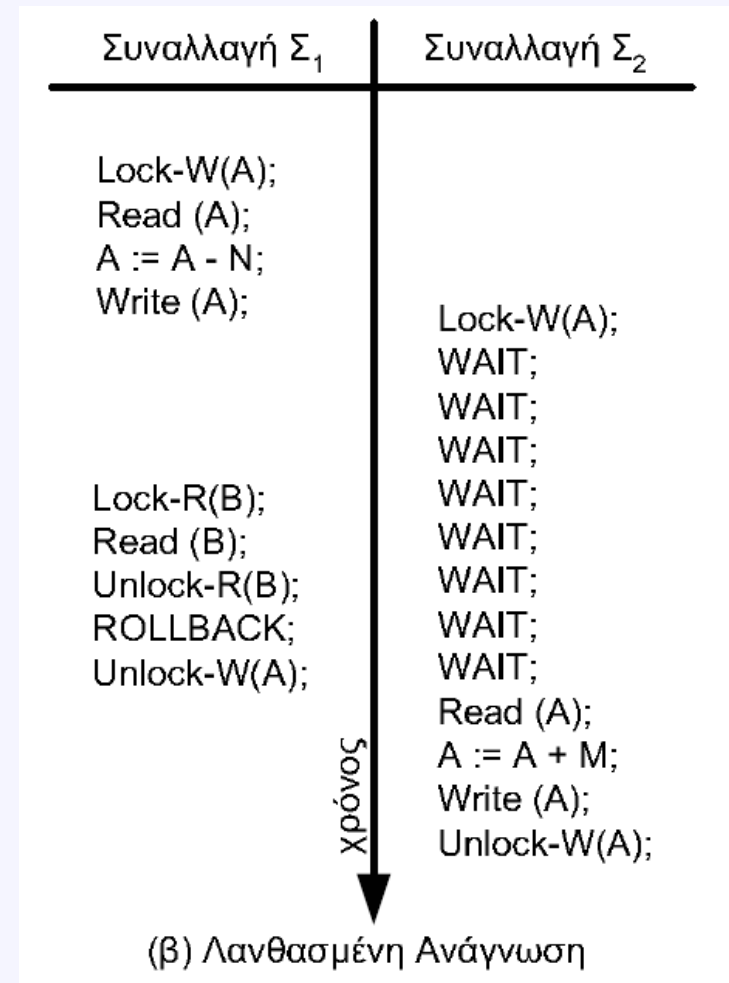
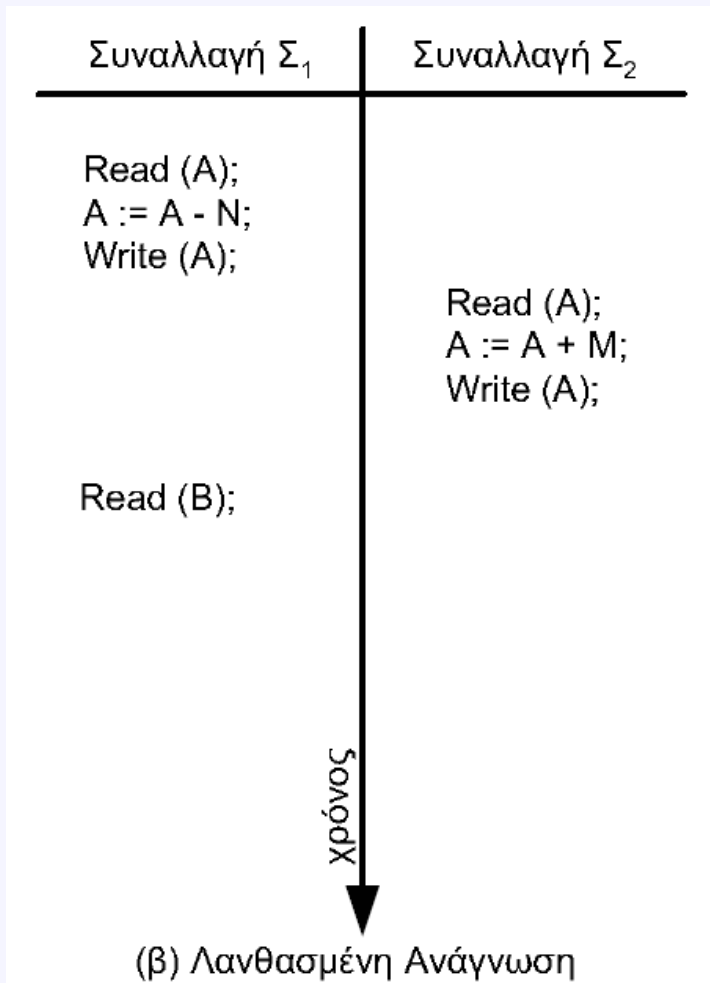
Πρωτόκολλο κλειδώματος δύο φάσεων

- Πως το πρωτόκολλο 2 φάσεων δίνει λύση στο πρόβλημα των χαμένων ενημερώσεων



Πρωτόκολλο κλειδώματος δύο φάσεων

- Πως το πρωτόκολλο 2 φάσεων δίνει λύση στο πρόβλημα της λανθασμένης ανάγνωσης



Πρωτόκολλο κλειδώματος δύο φάσεων

- Αποδεικνύεται ότι αν κάθε συναλλαγή ικανοποιεί τους κανόνες του πρωτοκόλλου κλειδώματος δύο φάσεων τότε **το χρονοδιάγραμμα** που προκύπτει **είναι πάντοτε σειριοποιήσιμο ως προς τις συγκρούσεις**
[Eswaran 1976]
- Χρησιμοποιείται όταν δεν υπάρχουν διαθέσιμες πληροφορίες σχετικά με τη σειρά των προσπελάσεων των δεδομένων από κάθε συναλλαγή

Πρωτόκολλο κλειδώματος βασισμένο σε γράφο

- Όταν είναι γνωστή η σειρά προσπελάσεων των δεδομένων από τη συναλλαγή χρησιμοποιείται το **πρωτόκολλο κλειδώματος βασισμένο σε γράφο**.
- Έστω $D=\{d_1, d_2, \dots, d_n\}$ ένα σύνολο n δεδομένων που προσπελούνται από τις συναλλαγές είτε για ανάγνωση είτε για αποθήκευση.
- Ορίζουμε μία **μερική διάταξη** στο σύνολο D ως εξής:
 - τα δεδομένα d_i και d_j συνδέονται με τη σχέση \Rightarrow αν κάθε συναλλαγή που προσπελάει τα d_i και d_j προσπελάει το d_i πριν από το d_j .

Πρωτόκολλο κλειδώματος βασισμένο σε γράφο

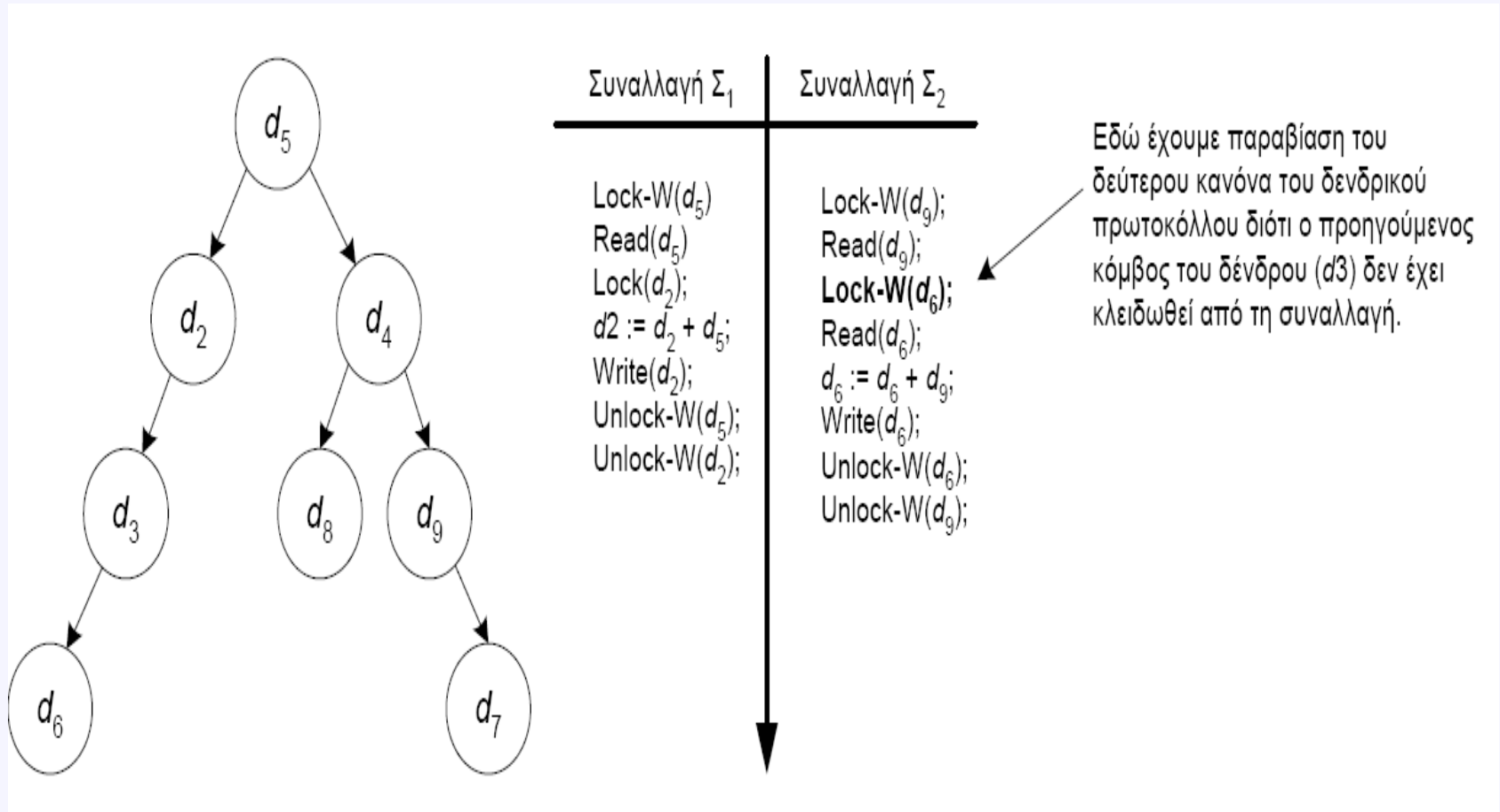
- Η μερική διάταξη επιτρέπει την αναπαράσταση του συνόλου ως κατευθυνόμενο άκυκλο γράφο.
- **Ειδική περίπτωση του κατευθυνόμενου άκυκλου γράφου είναι ο δενδρικός γράφος, όπου εφαρμόζεται το δενδρικό πρωτόκολλο (tree protocol).**

Δενδρικό πρωτόκολλο

- Το δενδρικό πρωτόκολλο έχει τους εξής κανόνες:
 1. η κάθε συναλλαγή μπορεί να ζητήσει χορήγηση της πρώτης κλειδαριάς σε οποιοδήποτε δεδομένο,
 2. μία συναλλαγή μπορεί να ζητήσει τη χορήγηση κλειδαριάς στο δεδομένο d_j μόνο αν έχει ήδη κλειδώσει το δεδομένο d_i και ισχύει $d_i \Rightarrow d_j$
 3. μία συναλλαγή μπορεί να απελευθερώσει μία κλειδαριά οποιαδήποτε χρονική στιγμή,
 4. ένα δεδομένο που έχει κλειδωθεί από μία συναλλαγή και στη συνέχεια η κλειδαριά έχει απελευθερωθεί, δεν επιτρέπεται να κλειδωθεί για δεύτερη φορά από τη ίδια συναλλαγή.

Δενδρικό πρωτόκολλο

- Παράδειγμα παραβίασης του 2ου κανόνα



Δενδρικό πρωτόκολλο

- Αποδεικνύεται ότι κάθε χρονοδιάγραμμα που ικανοποιεί τους κανόνες του δενδρικού πρωτοκόλλου είναι σειριοποιήσιμο ως προς τις συγκρούσεις

Διαχείριση αδιεξόδου

- Οι μηχανισμοί ελέγχου ταυτόχρονων προσπελάσεων που βασίζονται σε πρωτόκολλα κλειδώματος μπορούν να φέρουν το σύνολο των συναλλαγών σε αδιέξοδο
- Ένα σύνολο συναλλαγών βρίσκεται σε **αδιέξοδο** (deadlock), όταν καμία συναλλαγή δεν είναι εφικτό να συνεχίσει την εκτέλεσή της, λόγω αναμονής για την απελευθέρωση κλειδαριών.
- Το αδιέξοδο αποτελεί παθολογική κατάσταση και πρέπει να αποφεύγεται.

Διαχείριση αδιεξόδου

- Η διαχείριση του αδιεξόδου πραγματοποιείται γενικά με δύο μεθόδους:
 - **αποφυγή αδιεξόδου** (deadlock prevention), όπου το σύστημα ελέγχει την πιθανότητα να συμβεί αδιέξοδος και προσπαθεί να το αποφύγει (**προτείνεται όταν συμβαίνουν πολλά αδιέξοδα**),
 - **αναγνώριση αδιεξόδου και επανάκτηση** (deadlock detection and recovery) όπου το σύστημα είναι σε θέση να αναγνωρίσει ένα αδιέξοδος και προβαίνει σε ενέργειες για την κατάργησή του (**προτείνεται όταν δεν συμβαίνουν πολλά αδιέξοδα**).

Αποφυγή αδιεξόδου

- Μία μέθοδος αποφυγής του αδιεξόδου είναι ο ορισμός ενός **μέγιστου χρονικού διαστήματος** για την αναμονή χορήγησης μίας κλειδαριάς.
- **Αν εξαντληθεί το χρονικό διάστημα** και η κλειδαριά δεν έχει ακόμη χορηγηθεί στη συναλλαγή, **το σύστημα τερματίζει τη διαδικασία και την επαναδημιουργεί.**
- Η μέθοδος αυτή αποφυγής αδιεξόδου χρησιμοποιείται από πολλά ΣΔΒΔ λόγω της απλότητάς της.

Αποφυγή αδιεξόδου

- Μια άλλη μέθοδος είναι αυτή των **χρονικών σφραγίδων** (timestamps) και προτείνονται δύο αλγόριθμοι:
 - **Wait-Die**, που επιτρέπει σε μία **παλαιότερη συναλλαγή** (μικρή χρονοσφραγίδα) να **περιμένει** την απελευθέρωση μίας κλειδαριάς από μία **νεότερη** (μεγαλύτερη χρονοσφραγίδα). Διαφορετικά, η συναλλαγή τερματίζεται και το σύστημα την επανεκκινεί.

Αποφυγή αδιεξόδου

- Μια άλλη μέθοδος είναι αυτή των **χρονικών σφραγίδων** (timestamps) και προτείνονται δύο αλγόριθμοι:
 - **Wound-Wait**, όπου επιτρέπεται σε **μία νεότερη συναλλαγή** να **περιμένει** την εκτέλεση **μίας παλαιότερης**. Αν η παλαιότερη συναλλαγή ζητά τη χορήγηση **μίας κλειδαριάς** που είναι δεσμευμένη από **μία νεότερη συναλλαγή**, τότε η νεότερη συναλλαγή **τερματίζεται**.

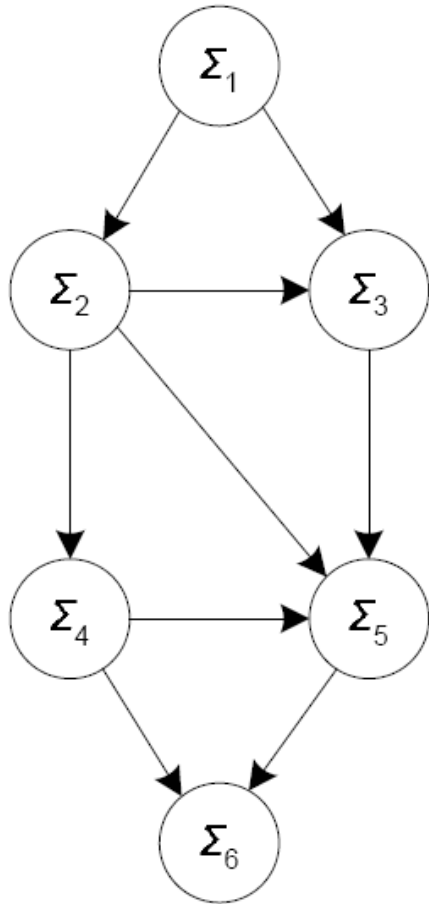
Αναγνώριση και κατάργηση αδιεξόδου

- Η αναγνώριση αδιεξόδου σε ένα σύνολο συναλλαγών διαπιστώνεται με τη βοήθεια ενός κατευθυνόμενου γράφου που καλείται **γράφος αναμονής** (wait graph)
- Ο γράφος αυτός αποτελείται από ένα σύνολο κορυφών V και ένα σύνολο ακμών E . **Για κάθε συναλλαγή υπάρχει και μία αντίστοιχη κορυφή στο γράφο.**
 - Αν υπάρχει ακμή από την κορυφή Σ_i προς την κορυφή Σ_j , τότε αυτό δηλώνει ότι η συναλλαγή Σ_i αναμένει μέχρι η συναλλαγή Σ_j να απελευθερώσει την κλειδαριά σε ένα τμήμα δεδομένων που πρέπει να προσπελάσει η Σ_i .

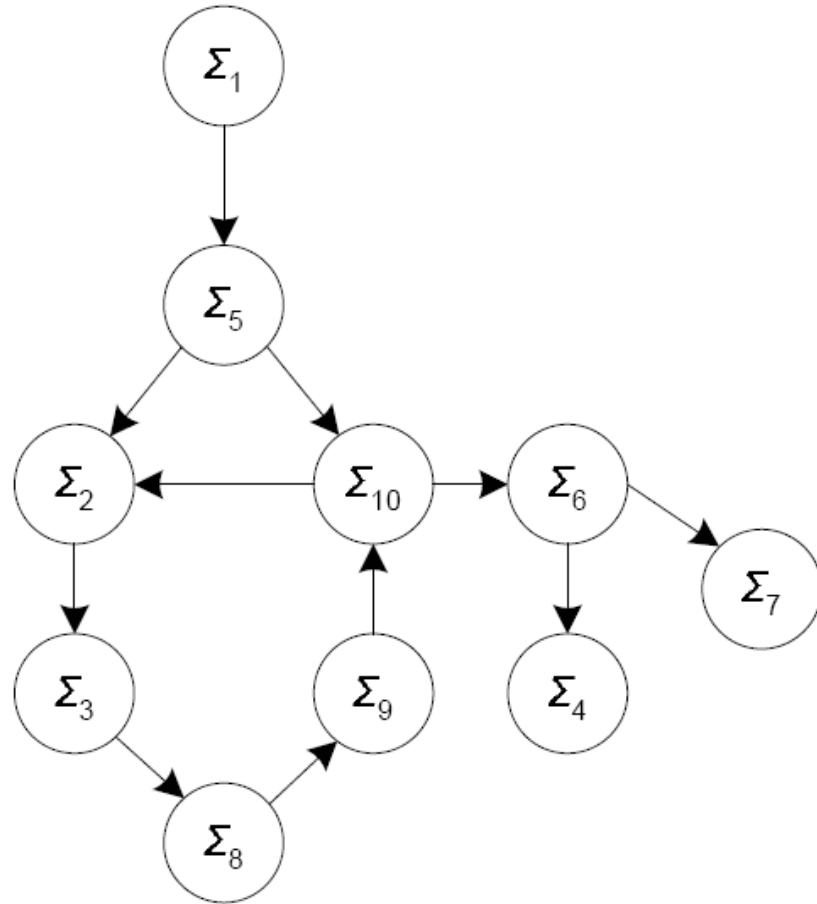
Αναγνώριση και κατάργηση αδιεξόδου

- **Αν στο γράφο αναμονής δεν υπάρχει κύκλος, τότε δεν υπάρχει αδιέξοδο.** Αντίθετα, η ύπαρξη έστω και ενός κύκλου σημαίνει ότι ένα υποσύνολο των συναλλαγών βρίσκεται σε αδιέξοδο.
- *Αν συμβαίνουν συχνά αδιέξοδα πρέπει να είναι συχνός και ο έλεγχος του γράφου αναμονής.*
- Το ΣΔΒΔ μπορεί να διατηρεί στατιστικά στοιχεία για τον αριθμό των αδιεξόδων και τη συχνότητα εμφάνισής τους και ανάλογα να προσδιορίζεται και η συχνότητα του ελέγχου για αδιέξοδο.

Αναγνώριση και κατάργηση αδιεξόδου



(α) άκυκλος γράφος αναμονής
(χωρίς αδιέξοδο)



(β) γράφος αναμονής με κύκλο
(ύπαρξη αδιεξόδου)

Κατάργηση αδιεξόδου

- Η συχνότερα χρησιμοποιούμενη μέθοδος για την κατάργηση του αδιεξόδου είναι ο **τερματισμός μίας η περισσότερων συναλλαγών** και η **επαναφορά των δεδομένων** στην κατάσταση που βρισκόταν πριν από την εκτέλεση των συναλλαγών αυτών (ROLLBACK).

Κατάργηση αδιεξόδου

Η μέθοδος ROLLBACK εκτελείται ως εξής:

- **Βήμα 1. Επιλογή συναλλαγών για ROLLBACK.**

Η επιλογή των συναλλαγών πραγματοποιείται με βάση το ελάχιστο κόστος. Στο προσδιορισμός του κόστους συμμετέχουν:

- ο μέχρι στιγμής χρόνος εκτέλεσης της συναλλαγής και ο υπολειπόμενος χρόνος για την ολοκλήρωσή της,
- ο αριθμός των δεδομένων που έχουν ενημερωθεί από τη συναλλαγή,
- ο αριθμός των δεδομένων που πρέπει να προσπελασθούν από τη συναλλαγή,
- Ο αριθμός των συναλλαγών όπου θα εφαρμοσθεί η λειτουργία ROLLBACK.

Κατάργηση αδιεξόδου

Η μέθοδος ROLLBACK εκτελείται ως εξής:

- **Βήμα 2. Επιλογή σημείου ROLLBACK.**

- η απλούστερη λύση είναι να γίνει **επαναφορά της συναλλαγής στην αρχή**, καταργώντας όλες τις αλλαγές που έχει επιφέρει στα δεδομένα,
 - η λύση αυτή έχει μεγάλο κόστος, διότι η συναλλαγή μπορεί να έχει ενημερώσει ένα μεγάλο αριθμό δεδομένων,
- συνήθως, το αδιέξοδο καταργείται εφαρμόζοντας **ROLLBACK στις τελευταίες εντολές της συναλλαγής**.

Κατάργηση αδιεξόδου

Η μέθοδος ROLLBACK εκτελείται ως εξής:

- **Βήμα 3. Αποφυγή ατέρμονης αναμονής.**

- η **ατέρμονη αναμονή** (starvation) είναι ένα φαινόμενο που μπορεί να εμφανισθεί **αν επιλέγεται συνεχώς η ίδια συναλλαγή για την εφαρμογή του ROLLBACK**,
 - αυτό σημαίνει ότι συνεχώς η ίδια συναλλαγή εμποδίζεται να ολοκληρώσει την εκτέλεσή της,
- το ΣΔΒΔ μπορεί να αποτρέψει την επιλογή της ίδιας συναλλαγής υιοθετώντας ένα **threshold** που **καθορίζει πόσες φορές μπορεί να επιλεγεί μια συναλλαγή**.

Μηχανισμός χρονικών σφραγίδων

- Ένας **εναλλακτικός μηχανισμός ελέγχου ταυτόχρονων συναλλαγών** είναι ο **μηχανισμός χρονικών σφραγίδων**
- Η **χρονική σφραγίδα** είναι μία **μεταβλητή** που **προσδιορίζει τη χρονική στιγμή** της αρχής της εκτέλεσης μίας συναλλαγής.
- Ο μηχανισμός χρονικών σφραγίδων **δεν χρησιμοποιεί κλειδαριές** για την προστασία μίας περιοχής δεδομένων, άρα **δεν υπάρχει περίπτωση να δημιουργηθεί αδιέξοδο.**

Μηχανισμός χρονικών σφραγίδων

- Μία χρονική σφραγίδα παράγεται είτε από το ρολόι του συστήματος είτε ενημερώνοντας την τιμή μίας μεταβλητής κάθε φορά που δημιουργείται μία νέα συναλλαγή.
- Η χρονική σφραγίδα της συναλλαγής Σ_i συμβολίζεται με $TS(\Sigma_i)$.

Μηχανισμός χρονικών σφραγίδων

- Η προσπέλαση επιτρέπεται μόνο αν η τελευταία ενημέρωση πραγματοποιήθηκε από παλαιότερη συναλλαγή (που θα έχει μικρότερη χρονική σφραγίδα).
- Αλλιώς πραγματοποιείται τερματισμός και επανεκκίνηση της συναλλαγής με νέα χρονική σφραγίδα.

Μηχανισμός χρονικών σφραγίδων

- Χρονικές σφραγίδες ορίζονται όχι μόνο για τις συναλλαγές αλλά και για τα δεδομένα:
 - σε κάθε σελίδα δεδομένων αντιστοιχεί μία χρονική σφραγίδα ανάγνωσης και μία χρονική σφραγίδα αποθήκευσης,
 - οι δύο χρονικές σφραγίδες συμβολίζονται με **timestamp-W** και **timestamp-R** αντίστοιχα.

Μηχανισμός χρονικών σφραγίδων

- Η τιμή της χρονικής σφραγίδας ανάγνωσης είναι η τιμή της χρονικής σφραγίδας της συναλλαγής που πραγματοποίησε τελευταία ανάγνωση στη σελίδα δεδομένων.
- Η τιμή της χρονικής σφραγίδας αποθήκευσης είναι η χρονική σφραγίδα της συναλλαγής που πραγματοποίησε τελευταία ενημέρωση στη σελίδα.
- Κάθε φορά που πραγματοποιείται ανάγνωση ή αποθήκευση της σελίδας δεδομένων D , ενημερώνονται οι τιμές $\text{timestamp-W}(D)$ και $\text{timestamp-R}(D)$, αντιστοίχως.

Πρωτόκολλο διάταξης χρονικών σφραγίδων

- Έστω ότι η συναλλαγή Σ_i εκτελεί λειτουργία ανάγνωσης στη σελίδα δεδομένων D . Διακρίνουμε τις εξής περιπτώσεις:
 1. Αν ισχύει $TS(\Sigma_i) < \text{timestamp-W}(D)$, τότε η συναλλαγή Σ_i προσπαθεί να διαβάσει μία τιμή του D η οποία έχει ήδη αλλάξει από άλλη συναλλαγή. Στην περίπτωση αυτή η λειτουργία ανάγνωσης διακόπτεται.
 2. Αν $TS(\Sigma_i) \geq \text{timestamp-W}(D)$, τότε η λειτουργία ανάγνωσης επιτρέπεται να ολοκληρωθεί και ενημερώνεται η τιμή $\text{timestamp-R}(D)$ ως εξής: $\text{timestamp-R}(D) = \max\{\text{timestamp-R}(D), TS(\Sigma_i)\}$.

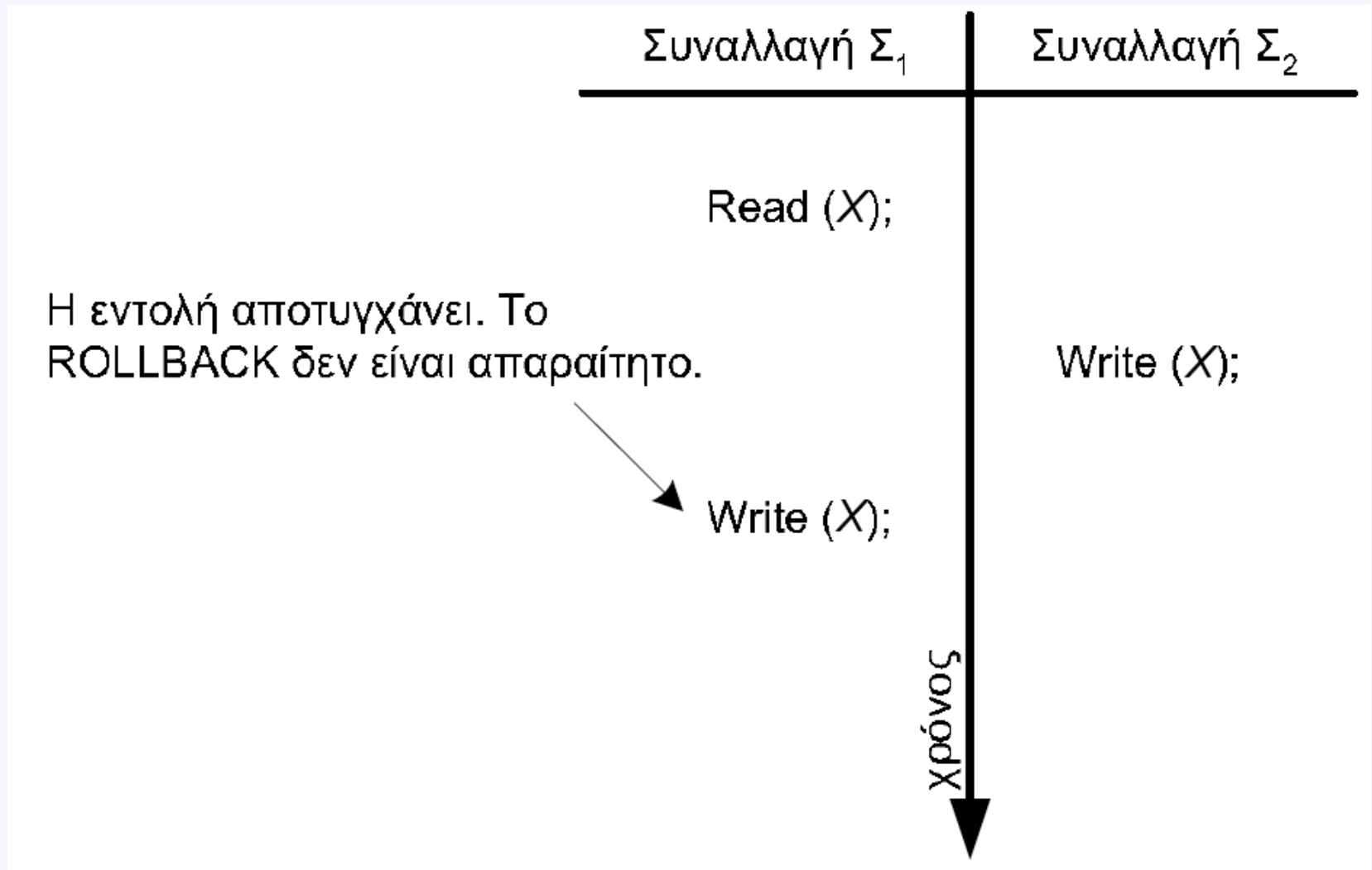
Πρωτόκολλο διάταξης χρονικών σφραγίδων

- Έστω τώρα ότι η συναλλαγή Σ_i εκτελεί λειτουργία αποθήκευσης στη σελίδα δεδομένων D . Διακρίνουμε τις εξής περιπτώσεις:
 1. Αν $TS(\Sigma_i) < \text{timestamp-W}(D)$, τότε η συναλλαγή προσπαθεί να ενημερώσει μία τιμή που έχει ήδη αλλάξει από άλλη συναλλαγή. Στην περίπτωση αυτή η διαδικασία εγγραφής τερματίζεται και μπαίνει σε λειτουργία ο μηχανισμός ROLLBACK για τη συναλλαγή Σ_i .
 2. Αν $TS(\Sigma_i) < \text{timestamp-R}(D)$, τότε η τρέχουσα τιμή του D διαβάσθηκε προηγουμένως από άλλη συναλλαγή. Επομένως, και σε αυτήν την περίπτωση η αποθήκευση διακόπτεται και πραγματοποιείται ROLLBACK.
 3. Αν δεν ισχύει καμία από τις δύο συνθήκες, τότε η λειτουργία εγγραφής μπορεί να ολοκληρωθεί χωρίς πρόβλημα και η χρονική σφραγίδα εγγραφής της Σ_i ενημερώνεται ως εξής: $\text{timestamp-W}(D) = TS(\Sigma_i)$.

Κανόνας αποθήκευσης του Thomas

- Με τη χρήση της μεθόδου των χρονικών σφραγίδων αν μία συναλλαγή προσπαθήσει να αλλάξει την τιμή μίας σελίδας δεδομένων, η οποία **έχει ήδη ενημερωθεί από άλλη συναλλαγή (μεταγενέστερη από την 1^η)**, τότε η εκτέλεση της συναλλαγής διακόπτεται και πραγματοποιείται **ROLLBACK**.
 - **$TS(\Sigma_i) < \text{timestamp}-W(D)$**
- Σύμφωνα με τον κανόνα αποθήκευσης του Thomas **αρκεί να αγνοήσουμε τη λειτουργία αποθήκευσης**, οπότε:
 - μειώνεται το κόστος λειτουργίας, και
 - αυξάνεται η απόδοση του συστήματος.

Κανόνας αποθήκευσης του Thomas



Κανόνας αποθήκευσης του Thomas

- Το νέο πρωτόκολλο χρονικών σφραγίδων διαφέρει μόνο ως προς τη διαχείριση των εντολών αποθήκευσης:
 - αν $TS(\Sigma_i) < \text{timestamp-W}(D)$,
τότε η συναλλαγή προσπαθεί να ενημερώσει μία τιμή που έχει ήδη αλλάξει από άλλη συναλλαγή. Στην περίπτωση αυτή η εντολή αποθήκευσης μπορεί να αγνοηθεί με ασφάλεια.
 - αν $TS(\Sigma_i) < \text{timestamp-R}(D)$,
τότε η τρέχουσα τιμή του D διαβάσθηκε προηγουμένως από άλλη συναλλαγή. Στην περίπτωση αυτή **διακόπτεται η εντολή αποθήκευσης** και πραγματοποιείται **ROLLBACK**.
 - αν **δεν ισχύει καμία από τις δύο συνθήκες**,
τότε η λειτουργία αποθήκευσης μπορεί να ολοκληρωθεί χωρίς πρόβλημα και $\text{timestamp-W}(D) = TS(\Sigma_i)$.

Επανάκτηση δεδομένων

- Οι συχνότερες αιτίες που οδηγούν σε διακοπή της λειτουργίας του συστήματος είναι:
 - **πτώση συστήματος**, λόγω εσφαλμένης λειτουργίας του υλικού ή του λογισμικού συστήματος,
 - **πρόβλημα στο μέσο αποθήκευσης**, το οποίο μπορεί να οφείλεται σε καταστροφή της μαγνητικής επιφάνειας του δίσκου, του μηχανικού ή του ηλεκτρονικού υλικού του δίσκου,
 - **σφάλμα λογισμικού εφαρμογής**, το οποίο οφείλεται συνήθως σε σφάλμα εκτέλεσης (run-time error) κατά τη διάρκεια ενημέρωσης των δεδομένων της ΒΔ,
 - φυσικά αίτια, όπως πυρκαγιές, σεισμοί κλπ.

Επανάκτηση δεδομένων

- Για να είναι εφικτή η λειτουργία της επανάκτησης δεδομένων, το ΣΔΒΔ πρέπει να διαθέτει:
 - μηχανισμό δημιουργίας **αντιγράφων ασφαλείας** (backup) για τα δεδομένα της ΒΔ,
 - καταγραφή **ημερολογίου** (log) όπου να αποθηκεύεται η κατάσταση της κάθε συναλλαγής και οι αλλαγές που έχουν πραγματοποιηθεί στη ΒΔ,
 - μηχανισμό **χρονικών σημείων ελέγχου** (checkpoints) που σηματοδοτούν τις στιγμές που οι αλλαγές που έχουν επιφέρει οι συναλλαγές έχουν γίνει μόνιμες,
 - μέθοδο επανάκτησης, η οποία λαμβάνοντας υπόψη τα αντίγραφα ασφαλείας, τις πληροφορίες ημερολογίου και τα χρονικά σημεία ελέγχου, επαναφέρει τη ΒΔ σε κατάσταση συνέπειας.