Προηγμένα Θέματα Βάσεων Δεδομένων

Διδάσκων: Άγγελος Μιχάλας

Περιεχόμενα

- Συναλλαγές
- Έλεγχος ταυτοχρονισμού
- Επανάκτηση δεδομένων

Συναλλαγές

- Ένα **σύνολο λειτουργιών** το οποίο αποτελεί **μία λογική λειτουργική μονάδα** καλείται **συναλλαγή** (**transaction**).
- Μια συναλλαγή περιέχει:
  - εντολές ανάγνωσης,
  - εντολές εισαγωγής,
  - εντολές διαγραφής, ή
  - εντολές ενημέρωσης

  των δεδομένων της ΒΔ.

Συναλλαγές

- Η πολυπλοκότητα των σύγχρονων βάσεων δεδομένων οδηγεί στην ανάπτυξη αξιόπιστων μηχανισμών ελέγχου της ακεραιότητας των δεδομένων
  - *πολλές συναλλαγές μπορεί να βρίσκονται σε εξέλιξη, προσπελαύνοντας κοινά δεδομένα*
- Ο **έλεγχος ταυτοχρονισμού** (concurrency control) διαχειρίζεται τις συναλλαγές έτσι ώστε να αποφεύγονται οι παθολογικές καταστάσεις

Συναλλαγές

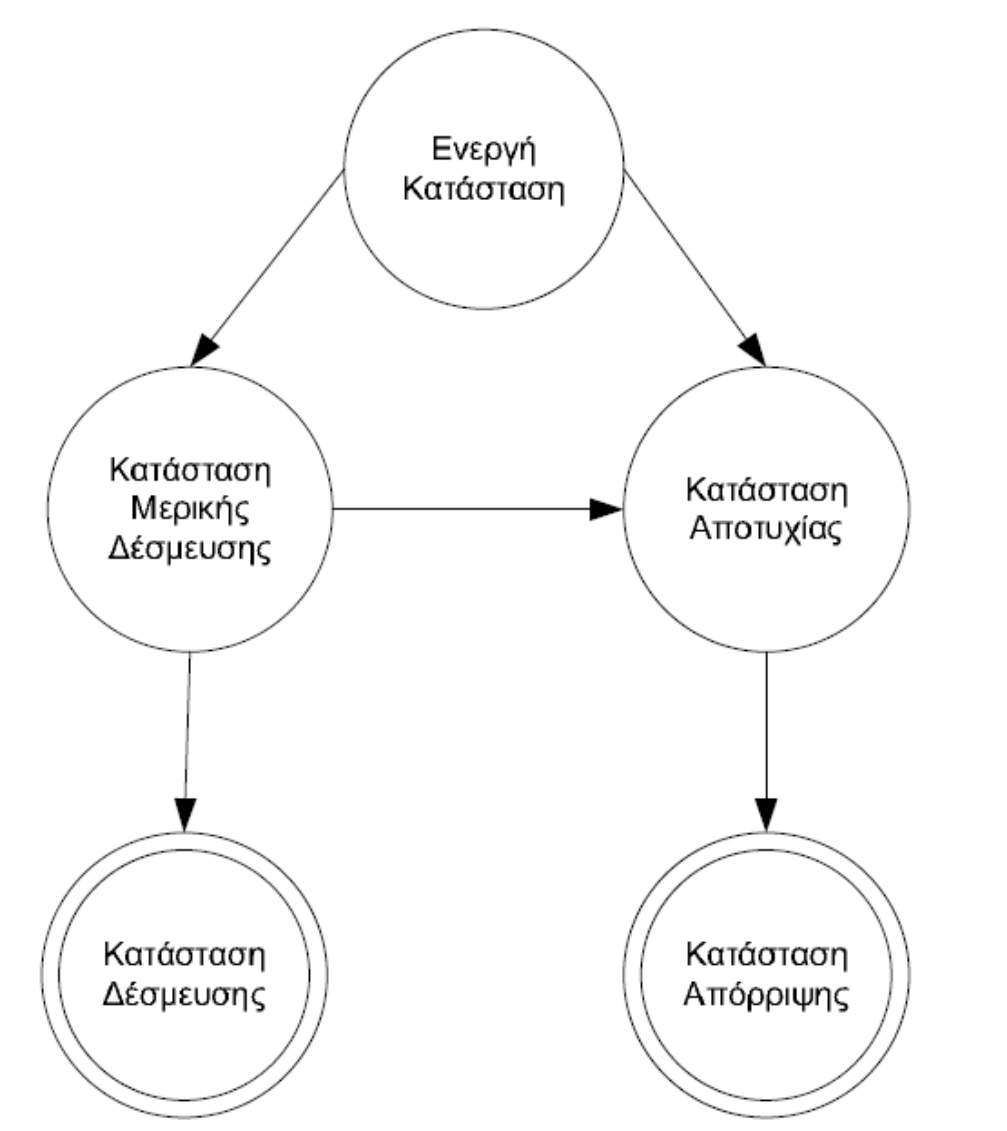
- Μια ΒΔ μπορεί να βρεθεί σε **ασταθή κατάσταση**:
  - από μία **βλάβη του συστήματος** (βλάβη του φυσικού μέσου αποθήκευσης) ή (άλλου μέρους του συστήματος),
  - μετά την αποκατάσταση της βλάβης πραγματοποιείται έλεγχος των δεδομένων.
- Υπάρχει ανάγκη ύπαρξης μηχανισμών **επανάκτησης** (recovery) δεδομένων, ώστε να επανέλθει η ΒΔ στην κανονική της κατάσταση όπου πληρούνται όλοι οι περιορισμοί ακεραιότητας.

Συναλλαγές

- Μία συναλλαγή είναι συνήθως το αποτέλεσμα της εκτέλεσης ενός προγράμματος που είναι γραμμένο σε μία γλώσσα προγραμματισμού υψηλού επιπέδου.
- Οι εντολές που προσδιορίζουν μια συναλλαγή περικλείονται μεταξύ των εκφράσεων:
  - **BEGIN TRANSACTION** και
  - **END TRANSACTION**

Καταστάσεις Συναλλαγών

- Μια συναλλαγή μπορεί να έχει τις εξής καταστάσεις:



Καταστάσεις Συναλλαγών

- **Ενεργή** (active) Κατάσταση:
  - η συναλλαγή εισέρχεται στην ενεργή κατάσταση κατά την αρχή της επεξεργασίας της και παραμένει σε αυτή ενόσω εκτελείται.
- Κατάσταση **Μερικής Δέσμευσης** (partial commit):
  - η συναλλαγή θεωρείται μερικώς δεσμευμένη όταν έχει ολοκληρωθεί και η τελευταία εντολή της συναλλαγής.
- Κατάσταση **Αποτυχίας** (failed):
  - η συναλλαγή αποτυγχάνει όταν το ΣΔΒΔ αντιληφθεί ότι δεν μπορεί να συνεχίσει την ομαλή επεξεργασία της

Καταστάσεις Συναλλαγών

- Κατάσταση **Απόρριψης** (aborted):
  - η συναλλαγή βρίσκεται στην κατάσταση αυτή όταν τα δεδομένα έχουν επανέλθει στην προηγούμενη σταθερή κατάσταση, πριν την αρχή της εκτέλεσης της συναλλαγής.
  - στην κατάσταση απόρριψης το ΣΔΒΔ έχει δύο επιλογές: (α) την επανεκκίνηση της συναλλαγής από την αρχή ή (β) την καταστροφή της συναλλαγής.
- Κατάσταση **Δέσμευσης** (committed)
  - η συναλλαγή έχει ολοκληρώσει επιτυχώς την εκτέλεσή της.

Ιδιότητες Συναλλαγών (ACID)

- **Ατομικότητα** (atomicity):
  - αν υπάρχει έστω και μία εντολή της συναλλαγής, η οποία αποτυγχάνει να εκτελεσθεί, τότε ολόκληρη η συναλλαγή αποτυγχάνει επίσης.
- **Απομόνωση** (isolation):
  - κάθε συναλλαγή πρέπει να εκτελείται ανεξάρτητα από άλλες συναλλαγές
- **Μονιμότητα** (durability):
  - αν μία συναλλαγή ολοκληρωθεί με επιτυχία, τότε οι αλλαγές που έχει επιφέρει καταγράφονται μόνιμα στη ΒΔ και δεν μπορούν να ανακληθούν.

Ιδιότητες Συναλλαγών (ACID)

- **Συνέπεια** (consistency):
  - η συναλλαγή πρέπει να μετατρέπει τη βάση δεδομένων από μία συνεπή κατάσταση σε μία άλλη συνεπή κατάσταση (τα δεδομένα πρέπει να είναι ορθά).
  - οι μηχανισμοί ακεραιότητας δεδομένων δεν επαρκούν για την εγγύηση της συνέπειας.
- Θεωρείστε τη μεταφορά ενός ποσού από έναν τραπεζικό λογαριασμό σε άλλον. Η αφαίρεση του ποσού από τον πρώτο πρέπει να συνοδεύεται από την πρόσθεση του ποσού στο δεύτερο. Αν το αφαιρούμενο ποσό διαφέρει από το προστιθέμενο, τότε τα δεδομένα της βάσης δεν έχουν συνέπεια (δεν είναι ορθά).

Τρόποι Επεξεργασίας

- **Ακολουθιακή** ή **σειριακή** εκτέλεση
  - όπου **οι συναλλαγές εκτελούνται η μία μετά την άλλη** αλλά εμφανίζονται σημαντικές καθυστερήσεις με αποτέλεσμα να **μειώνεται η γενική απόδοση του συστήματος**.
- **Ταυτόχρονη** εκτέλεση πολλών συναλλαγών
  - όπου ο δίσκος ή η CPU μπορούν να απασχολούνται με άλλη συναλλαγή, οπότε αυξάνεται ο αριθμός των συναλλαγών που ολοκληρώνονται στη μονάδα του χρόνου (throughput).
  - Πλεονέκτημα: μειώνονται οι καθυστερήσεις (waiting time) και ο μέσος χρόνος εκτέλεσης των συναλλαγών (mean response time).
  - Μειονέκτημα: ενδέχεται να επιφέρει προβλήματα στην ακεραιότητα και συνέπεια των δεδομένων της ΒΔ.

Έλεγχος ταυτοχρονισμού

- Ο σωστός συντονισμός των συναλλαγών είναι ευθύνη του ΣΔΒΔ, το οποίο μέσω του **μηχανισμού ελέγχου ταυτοχρονισμού** εγγυάται την ομαλή εκτέλεση των συναλλαγών.
- *Ο τρόπος συντονισμού και εκτέλεσης ενός συνόλου συναλλαγών καλείται* ***χρονοδιάγραμμα*** *(schedule).*
- Το ΣΔΒΔ πρέπει να επιλέξει ένα χρονοδιάγραμμα που να ικανοποιεί τις απαιτήσεις συνέπειας των δεδομένων.

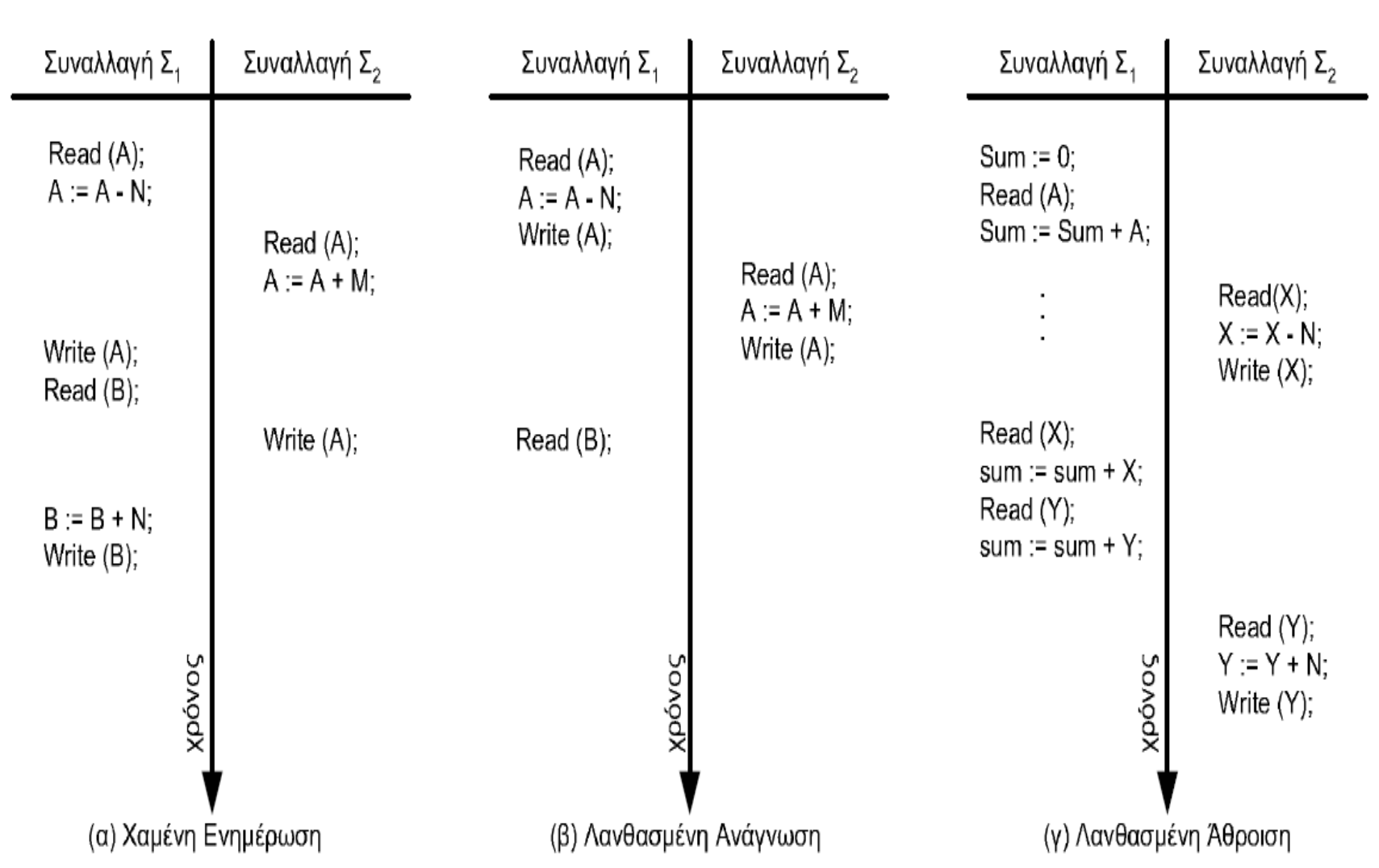
Έλεγχος ταυτοχρονισμού

- Ο συντονισμός των συναλλαγών μπορεί να επιλύσει προβλήματα που εμφανίζονται από την ταυτόχρονη εκτέλεση των συναλλαγών.
- **Πρόβλημα της χαμένης ενημέρωσης**: Εμφανίζεται όταν δύο συναλλαγές πραγματοποιούν ανάγνωση και αποθήκευση στα ίδια δεδομένα.

Έλεγχος ταυτοχρονισμού

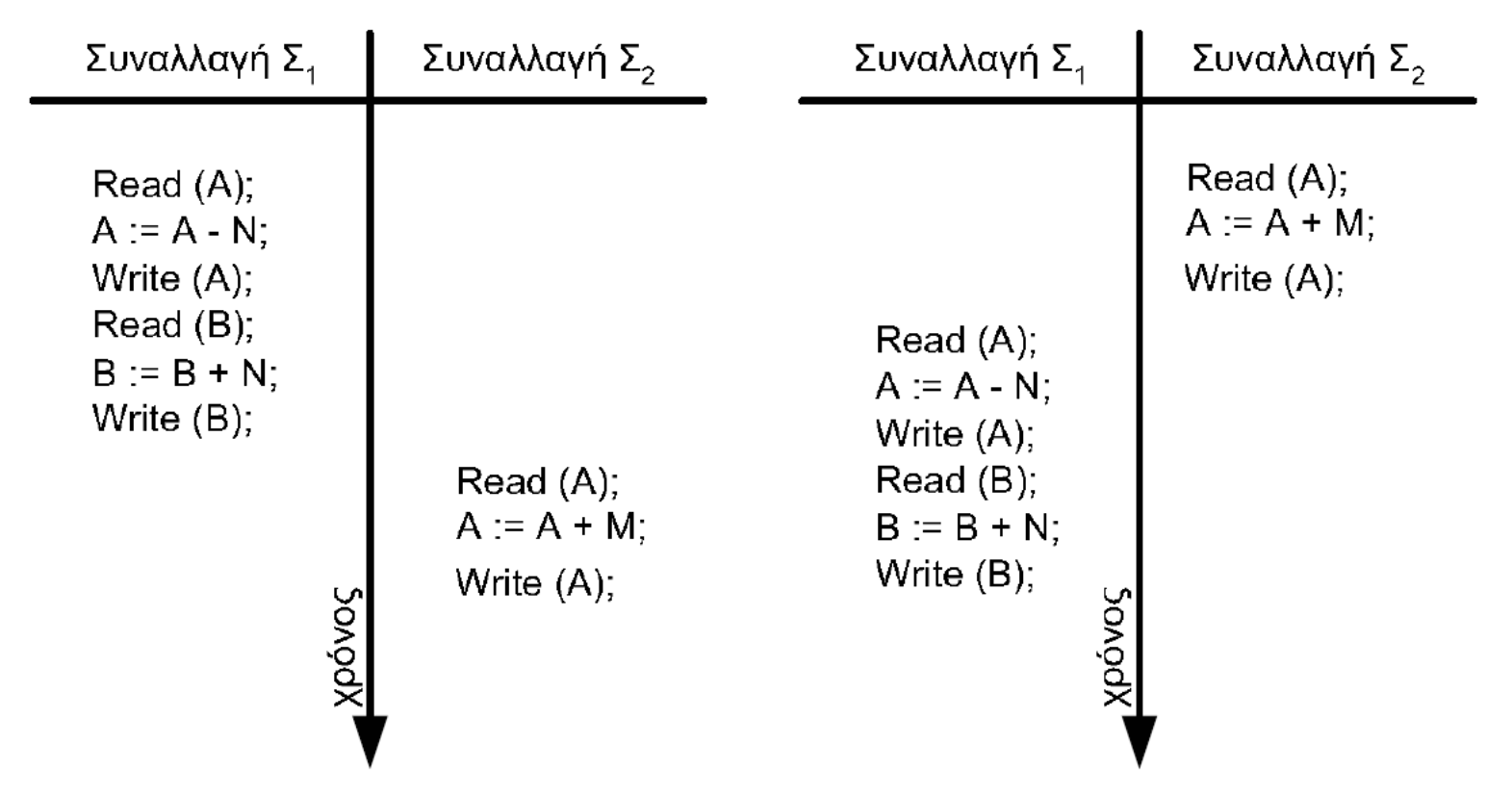
- **Πρόβλημα της λανθασμένης ανάγνωσης**: Εμφανίζεται όταν μία συναλλαγή πραγματοποιεί αλλαγές στα δεδομένα και για κάποιο λόγο αποτυγχάνει.
  - Πριν την επαναφορά των δεδομένων στην αρχική κατάσταση μία άλλη συναλλαγή διαβάζει τα δεδομένα με αποτέλεσμα να πραγματοποιεί λανθασμένη ανάγνωση.
- **Πρόβλημα της λανθασμένης άθροισης**: Κάποια δεδομένα ενημερώνονται από άλλη συναλλαγή πριν αθροιστούν ενώ άλλα αφού έχουν αθροιστεί.

Παραδείγματα Χρονοδιαγραμμάτων



Σειριακό χρονοδιάγραμμα

- Σε ένα σειριακό (serial) χρονοδιάγραμμα οι εντολές μίας συναλλαγής δεν περιπλέκονται χρονικά με εντολές των άλλων συναλλαγών.



Σειριακό χρονοδιάγραμμα

- Υπάρχουν **n!** δυνατά σειριακά **χρονοδιαγράμματα** εκτέλεσης **για n συναλλαγές**.
  - Αριθμός διατάξεων των n συναλλαγών
- Ο τρόπος εκτέλεσης των συναλλαγών ορίζεται από τη μέθοδο χρονοδρομολόγησης όπως αυτές που χρησιμοποιεί το λειτουργικό σύστημα (πως διαμοιράζει το ΛΣ τους πόρους όπως CPU, RAM, HD στις συναλλαγές) και επομένως δεν μπορούμε να γνωρίζουμε με ακρίβεια το χρόνο εκτέλεσης της κάθε εντολής μίας συναλλαγής.

Σειριοποίηση

- Για να διατηρεί τη συνέπεια των δεδομένων της ΒΔ θα πρέπει το **χρονοδιάγραμμα εκτέλεσης συναλλαγών** να είναι **ισοδύναμο** με κάποιο **σειριακό χρονοδιάγραμμα**.
- ***Δύο χρονοδιαγράμματα καλούνται ισοδύναμα αν μετά την εκτέλεσή τους επιφέρουν τις ίδιες αλλαγές στη ΒΔ***.

Σειριοποίηση

- Η διαδιακασία παραγωγής ενός χρονοδιαγράμματος ισοδύναμου με ένα σειριακό χρονοδιάγραμμα καλείται **σειριοποίηση** (serializability) και έχει δύο μορφές:
  - σειριοποίηση **σύγκρουσης** (conflict), και
  - σειριοποίηση **όψης** (view).

Σειριοποίηση Σύγκρουσης

- Έστω χρονοδιάγραμμα S και δύο χρονικά συνεχόμενες εντολές E_i, E_j (που αναφέρονται στα ίδια δεδομένα D) των συναλλαγών Σ1, Σ2, αντιστοίχως.
- Διακρίνουμε τις εξής περιπτώσεις:
  - E_i = read(D) και E_j = read(D)
  - E_i = read(D) και E_j = write(D)
  - E_i = write(D) και E_j = read(D)
  - E_i = write(D) και E_j = write(D)
- Στην 1η περίπτωση η σειρά εκτέλεσης δεν έχει σημασία
- Στις άλλες (έχει σημασία η σειρά) έχουμε σύγκρουση

Σειριοποίηση Σύγκρουσης

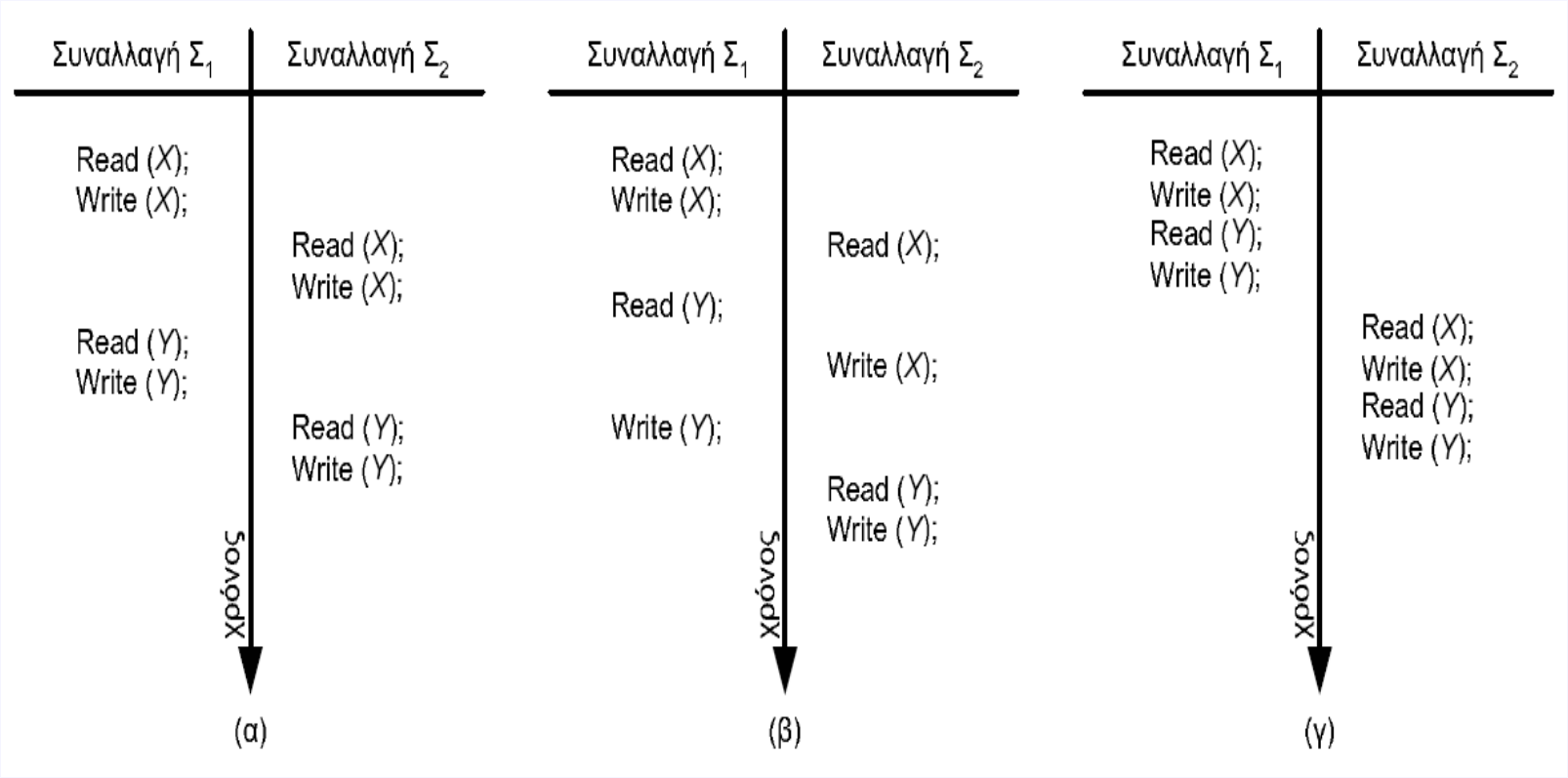
- Αν τουλάχιστον μία εντολή είναι εντολή αποθήκευσης, τότε οι εντολές E_i και E_j βρίσκονται σε σύγκρουση
- Αν οι εντολές E_i και E_j δεν βρίσκονται σε σύγκρουση, τότε μπορούμε να αλλάξουμε τη σειρά εκτέλεσης και να δημιουργήσουμε ένα νέο χρονοδιάγραμμα.

Σειριοποίηση Σύγκρουσης

- Αν ένα χρονοδιάγραμμα εκτέλεσης S μπορεί να μετασχηματισθεί σε ένα άλλο χρονοδιάγραμμα S' αλλάζοντας τη σειρά εκτέλεσης των εντολών, τότε τα S και S' καλούνται **ισοδύναμα ως προς τις συγκρούσεις**.
- Το χρονοδιάγραμμα S καλείται σειριοποιήσιμο ως προς τις συγκρούσεις (conflict serializable) αν είναι ισοδύναμο ως προς τις συγκρούσεις με ένα σειριακό χρονοδιάγραμμα.

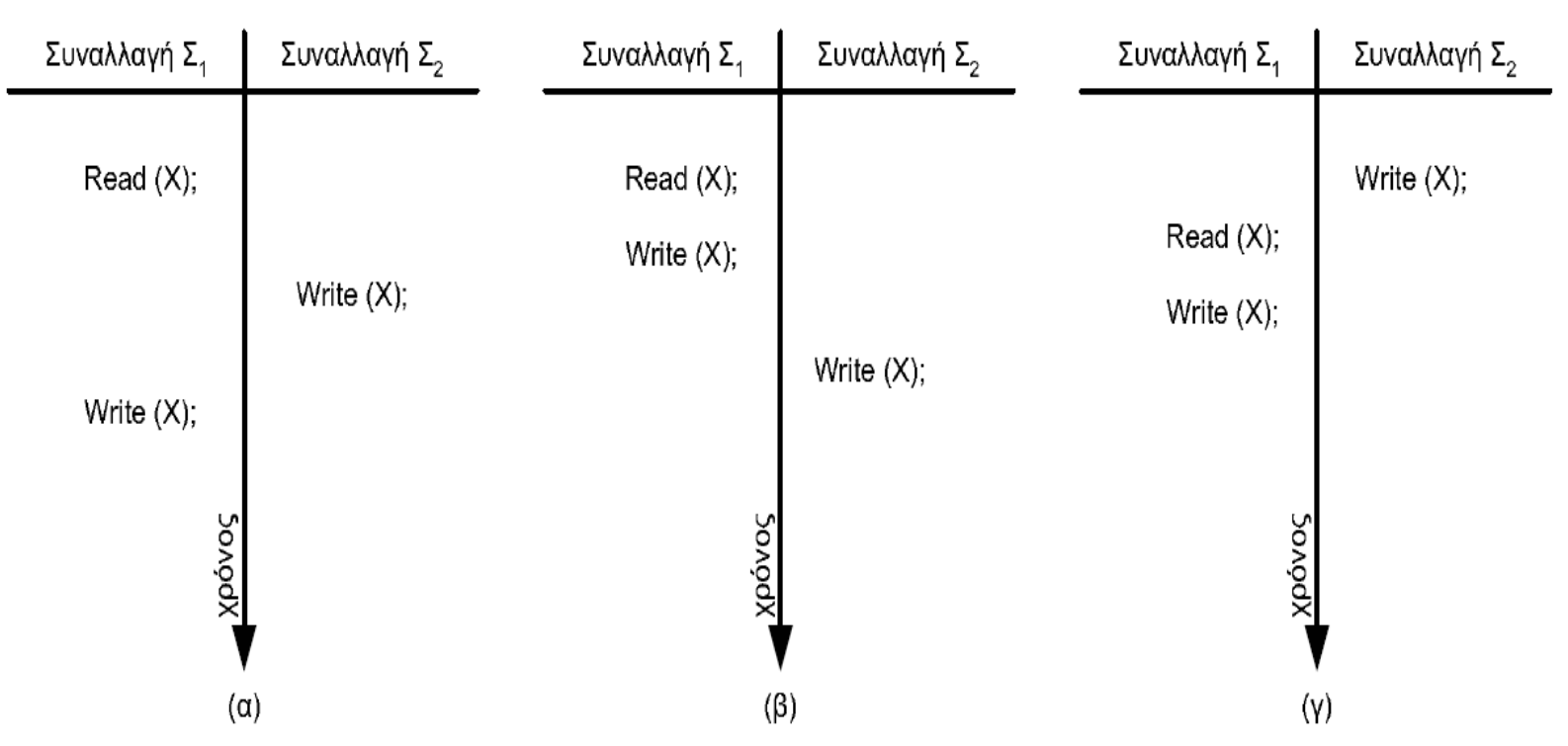
Σειριοποίηση Σύγκρουσης - Παράδειγμα

- Η εντολή Write(X), Read(Y) δεν συγκρούονται ➔ μπορούμε να αλλάξουμε τη σειρά εκτέλεσης.
- Η εντολή Read(X), Read(Y) δεν συγκρούονται ➔ μπορούμε να αλλάξουμε τη σειρά εκτέλεσης.



Σειριοποίηση Σύγκρουσης - Παράδειγμα

- **Είναι όλα τα χρονοδιαγράμματα ισοδύναμα με ένα σειριακό; ΟΧΙ**
- **Παράδειγμα**: Το (α) δεν είναι ισοδύναμο ούτε με το (β) ούτε με το (γ).



Σειριοποίηση Όψης

- Έστω δύο χρονοδιαγράμματα S και S΄ με τα ίδια σύνολα συναλλαγών. Τα S και S΄ ονομάζονται ισοδύναμα αν για κάθε στοιχείο δεδομένων D:
  1. αν η συναλλαγή Σ_i διαβάζει την αρχική τιμή του D στο χρονοδιάγραμμα S, τότε πρέπει η συναλλαγή Σ_i επίσης να διαβάζει την αρχική τιμή του D στο χρονοδιάγραμμα S΄,

Σειριοποίηση Όψης

- Έστω δύο χρονοδιαγράμματα S και S′ με τα ίδια σύνολα συναλλαγών. Τα S και S′ ονομάζονται ισοδύναμα αν για κάθε στοιχείο δεδομένων D:
  2. αν η συναλλαγή Σ_i εκτελεί την εντολή Read(D) στο χρονοδιάγραμμα S και αυτή η τιμή έχει παραχθεί από τη συναλλαγή Σ_j, τότε η Σ_i πρέπει επίσης να διαβάσει την τιμή που παράγει η Σ_j στο χρονοδιάγραμμα S′,

Σειριοποίηση Όψης

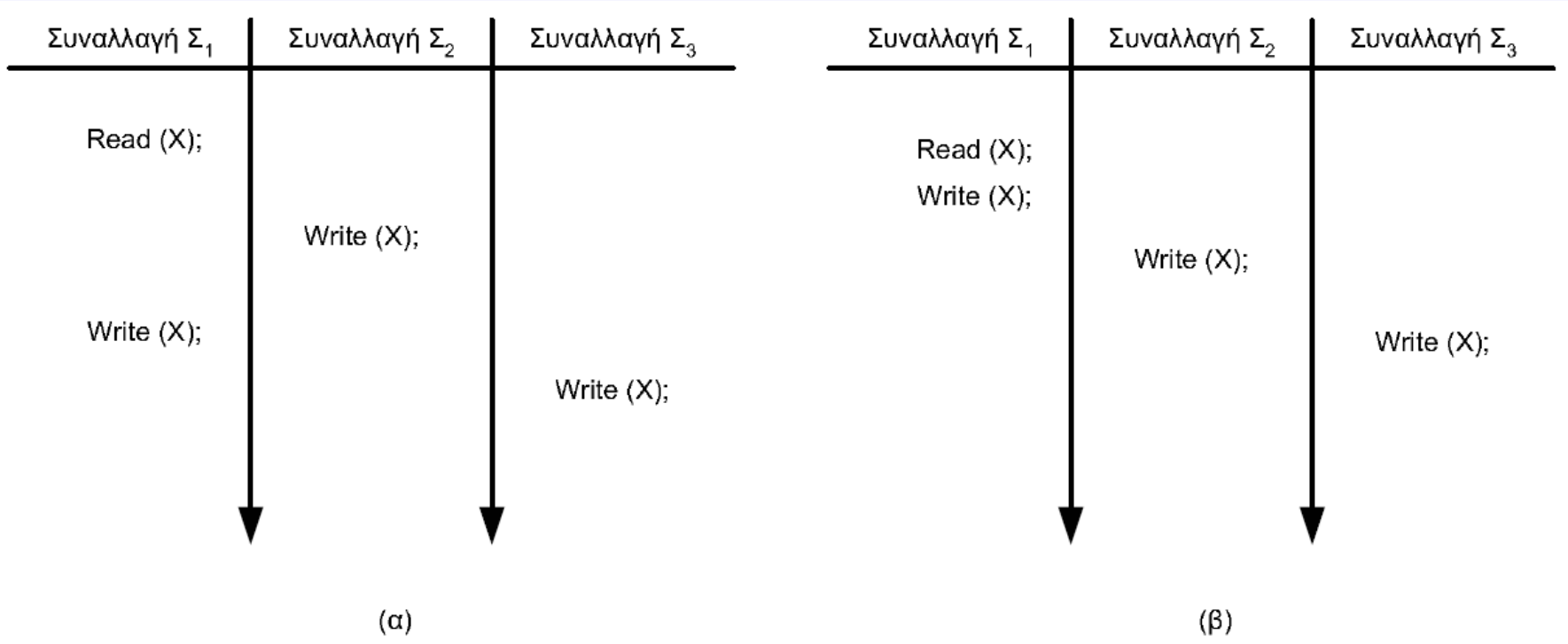
- Έστω δύο χρονοδιαγράμματα S και S΄ με τα ίδια σύνολα συναλλαγών. Τα S και S΄ ονομάζονται ισοδύναμα αν για κάθε στοιχείο δεδομένων D:
  3. η συναλλαγή που εκτελεί την τελευταία εντολή Write(D) στο χρονοδιάγραμμα S, πρέπει και στο χρονοδιάγραμμα S' να εκτελεί την τελευταία εντολή Write(D).

Σειριοποίηση Όψης

- Οι **δύο πρώτοι κανόνες εγγυώνται** ότι **οι συναλλαγές** και στα δύο χρονοδιαγράμματα **θα διαβάσουν τα ίδια δεδομένα**
  - Επομένως θα εκτελέσουν τις ίδιες λειτουργίες στα δεδομένα
- Ο **τρίτος κανόνας** σε συνδιασμό με τους άλλους δύο **εγγυάται** ότι **το αποτέλεσμα** της συνολικής εκτέλεσης των συναλλαγών **θα είναι το ίδιο και στα δύο χρονοδιαγράμματα**
  - Επομένως θα επιφέρουν το ίδιο αποτέλεσμα στη ΒΔ

Σειριοποίηση Όψης

- Ένα χρονοδιάγραμμα S καλείται **σειριοποιήσιμο ως προς την όψη** (view serializable) **αν είναι ισοδύναμο ως προς την όψη με κάποιο σειριακό χρονοδιάγραμμα**.



Μηχανισμοί Κλειδώματος

- **Βασικός μηχανισμός συντονισμού της εκτέλεσης των συναλλαγών**
- Η **κλειδαριά** (lock) αντιστοιχεί σε ένα τμήμα δεδομένων και **είναι μεταβλητή που περιγράφει την κατάσταση του τμήματος των δεδομένων** σε σχέση με τις λειτουργίες που ενεργούν στα συγκεκριμένα δεδομένα.

Μηχανισμοί Κλειδώματος

- Δύο είναι οι βασικοί τύποι κλειδαριών:
  - **κλειδαριά ανάγνωσης** (read) ή κλειδαριά διαμοιρασμού (share), που επιτρέπει σε μία συναλλαγή να διαβάσει τα δεδομένα αλλά όχι να τα ενημερώσει ή να τα διαγράψει,
  - **κλειδαριά αποθήκευσης** (write) ή αποκλειστική κλειδαριά (exclusive), που επιτρέπει σε μία συναλλαγή να διαβάσει ή να ενημερώσει τα δεδομένα.
- Μπορούν να χορηγηθούν **πολλές κλειδαριές ανάγνωσης** αλλά **μόνο μία αποθήκευσης**.

Μηχανισμοί Κλειδώματος

- Ο μηχανισμός κλειδαριών λειτουργεί ως εξής:
  1. μία συναλλαγή που αφορά ανάγνωση δεδομένων, ζητά να της χορηγηθεί μία κλειδαριά ανάγνωσης,
     - αν απαιτείται ενημέρωση δεδομένων, τότε ζητά μία κλειδαριά αποθήκευσης,
  2. αν δεν υπάρχει άλλη κλειδαριά για τη συγκεκριμένη σελίδα δεδομένων, τότε η κλειδαριά χορηγείται στη συναλλαγή,
  3. αν υπάρχει ήδη άλλη κλειδαριά που έχει χορηγηθεί σε άλλη συναλλαγή, τότε πραγματοποιείται έλεγχος για το αν θα χορηγηθεί η κλειδαριά στη νέα συναλλαγή ή όχι.

Μηχανισμοί Κλειδώματος

4. αν η σελίδα δεδομένων έχει κλειδωθεί με χρήση κλειδαριάς ανάγνωσης και η νέα συναλλαγή ζητά επίσης κλειδαριά ανάγνωσης, τότε το σύστημα επιτρέπει τη χορήγηση της κλειδαριάς.
5. Σε διαφορετική περίπτωση η νέα συναλλαγή πρέπει να περιμένει μέχρι η προηγούμενη συναλλαγή να ολοκληρώσει την εργασία με τη σελίδα δεδομένων και να απελευθερώσει την κλειδαριά.
6. μία συναλλαγή απελευθερώνει την κλειδαριά όταν η εκτέλεσή της ολοκληρωθεί είτε με επιτυχία είτε με αποτυχία.
7. οι μόνιμες αλλαγές στα δεδομένα είναι ορατές από τις υπόλοιπες συναλλαγές μόνο μετά την απελευθέρωση της κλειδαριάς αποθήκευσης από τη συναλλαγή.

Μηχανισμοί Κλειδώματος

- Μερικά συστήματα επιτρέπουν σε μια συναλλαγή να αλλάξει τον τύπο της κλειδαριάς.
- Μια συναλλαγή που πρέπει να ενημερώσει κάποια δεδομένα μπορεί να ζητήσει **αναβάθμιση** (upgrade) της κλειδαριάς από κλειδαριά ανάγνωσης σε κλειδαριά αποθήκευσης.
- Αν η εγγραφή των δεδομένων έχει ολοκληρωθεί αλλά η συναλλαγή δεν έχει ολοκληρώσει την ανάγνωση των δεδομένων, τότε μπορεί να ζητήσει **υποβάθμιση** (downgrade) της κλειδαριάς από αποθήκευσης σε ανάγνωσης.

Μηχανισμοί Κλειδώματος

- Ο μηχανισμός κλειδώματος δεν δίνει πάντοτε λύση στο πρόβλημα της συνέπειας

Μηχανισμοί Κλειδώματος - Αντιπαράδειγμα

- Θεωρούμε X=100, Y=400. Εφαρμόζουμε τις κατάλληλες εντολές Lock, Unlock στο σχήμα α και παίρνουμε το σχήμα β
- Αν εκτελεστεί πρώτα η Σ1 και μετά η Σ2 θα έχουμε X=220, Y=330. Αν εκτελεστεί πρώτα η Σ2 και μετά η Σ1 θα έχουμε X=210, Y=340. ➔ το χρονοδιάγραμμα δεν είναι σειριοποιήσιμο