Προχωρημένα Θέματα Τεχνολογίας και Εφαρμογών Βάσεων Δεδομένων

Εσωτερική Αρχιτεκτονική Βάσεων Δεδομένων

Πάνος Βασιλειάδης

pvassil@cs.uoi.gr

Σεπτέμβρης 2009

www.cs.uoi.gr/~pvassil/courses/db_III/

Θεματολόγιο

- Γενική αρχιτεκτονική βάσεων δεδομένων & ΣΔΒΔ
- Εσωτερική αποθήκευση δεδομένων
- Δομές στη μνήμη
- Διεργασίες
- Παράρτημα: Αξιοπιστία και διαθεσιμότητα βάσεων δεδομένων

Οι εικόνες είναι από το Oracle 9i Database Concepts Manual, Release 1 (9.0.1), July 2001

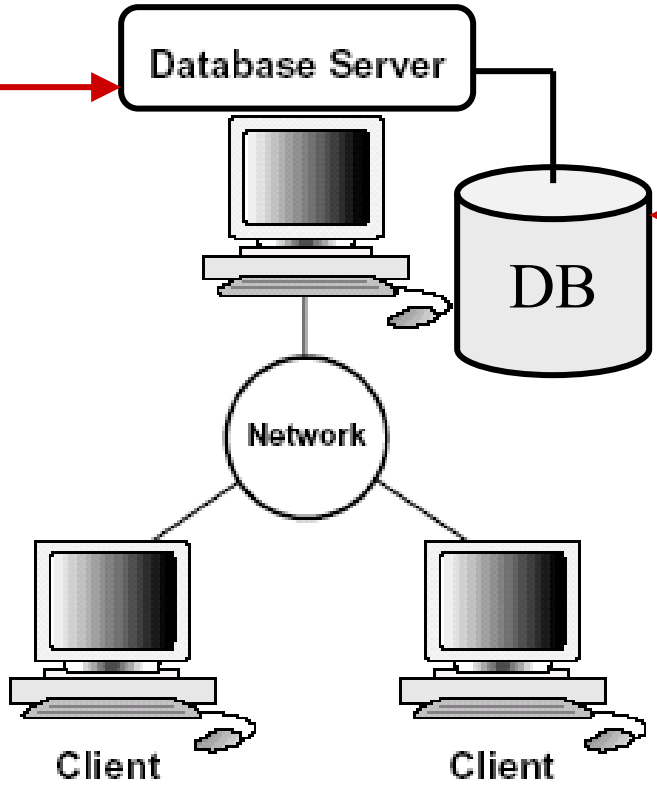
Θεματολόγιο

- Γενική αρχιτεκτονική βάσεων δεδομένων & ΣΔΒΔ
- Εσωτερική αποθήκευση δεδομένων
- Δομές στη μνήμη
- Διεργασίες
- Παράρτημα: Αξιοπιστία και διαθεσιμότητα βάσεων δεδομένων

Βάσεις Δεδομένων και ΣΔΒΔ

Σύστημα Διαχείρισης Βάσεων Δεδομένων

ΣΔΒΔ (DBMS)



Βάση Δεδομένων

Βάσεις Δεδομένων και ΣΔΒΔ

- Μια **Βάση Δεδομένων** είναι μια οργανωμένη συλλογή θεματικά συναφών δεδομένων (συνήθως σε μεγάλη κλίμακα)

- Ένα **Σύστημα Διαχείρισης Βάσεων Δεδομένων** (**DataBase Management System – DBMS**) είναι ένα πακέτο λογισμικού που σκοπό έχει τη διαχείριση βάσεων δεδομένων

Βάσεις Δεδομένων και ΣΔΒΔ

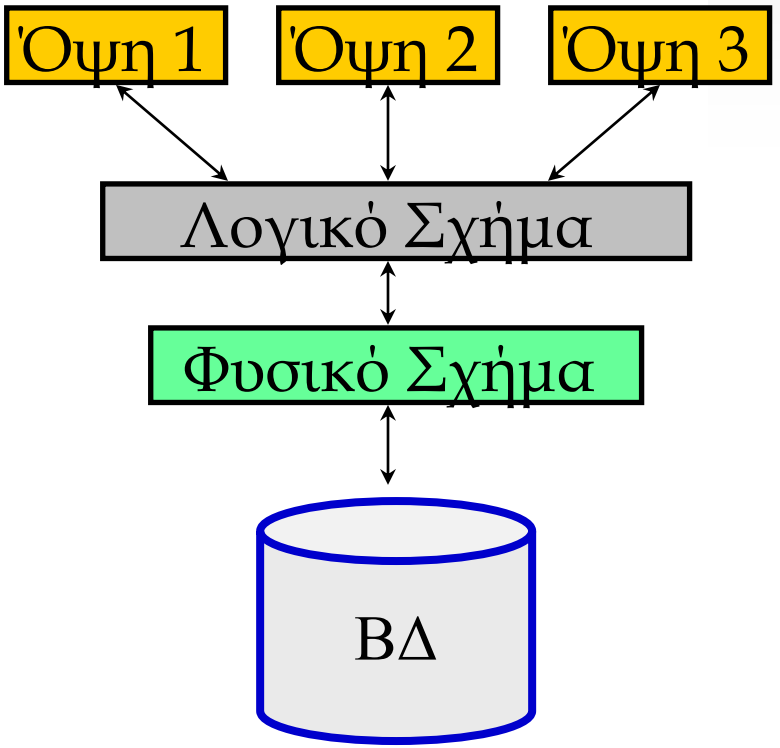
- Διαχείριση Βάσεων Δεδομένων =
  - αποθήκευση
  - επερώτηση
  - ανανέωση

  } των δεδομένων
  - εγγύηση της αξιοπιστίας
  - εγγύηση της διαθεσιμότητας

  } της ΒΔ
- Σχεσιακό ΣΔΒΔ (Relational DBMS – RDBMS): ΣΔΒΔ στο οποίο η οργάνωση των δεδομένων ακολουθεί το σχεσιακό μοντέλο

Σχήματα της Βάσης Δεδομένων

- Το **φυσικό σχήμα** αφορά την αποθήκευση των δεδομένων στη βάση δεδομένων
- Οι **όψεις** (των χρηστών) αφορούν το πώς βλέπουν οι χρήστες τη βάση δεδομένων
- Το **λογικό σχήμα** δρα ως γέφυρα ανάμεσα στα δύο άκρα



Σχήματα της Βάσης Δεδομένων

- **Λογικό σχήμα:**

Emp(emp_id:integer, name: varchar2[40], dept_id: integer)

Dept(dept_id: integer, manager-id: integer)

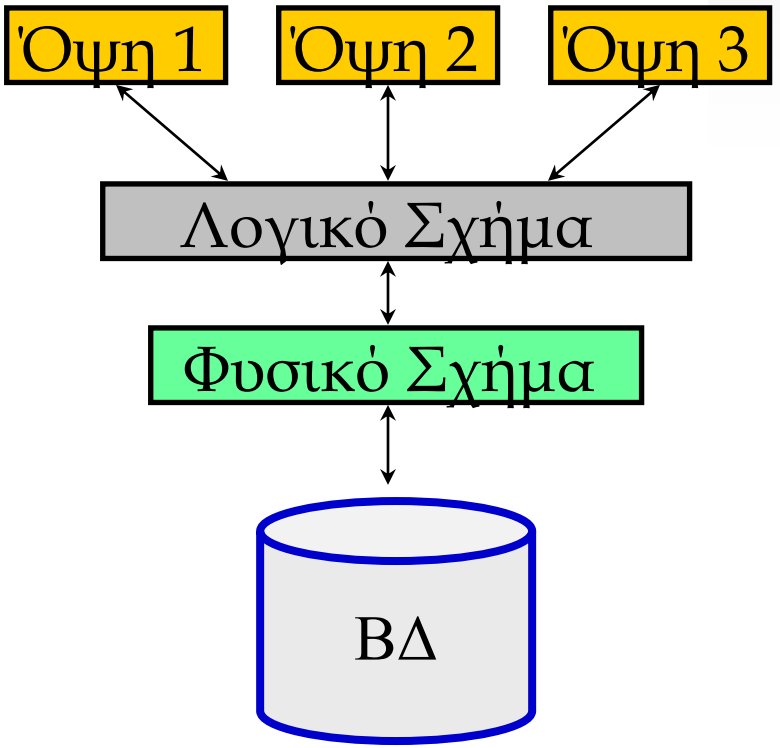
- **Φυσικό σχήμα**

ISAM files for relations in xxx.ora

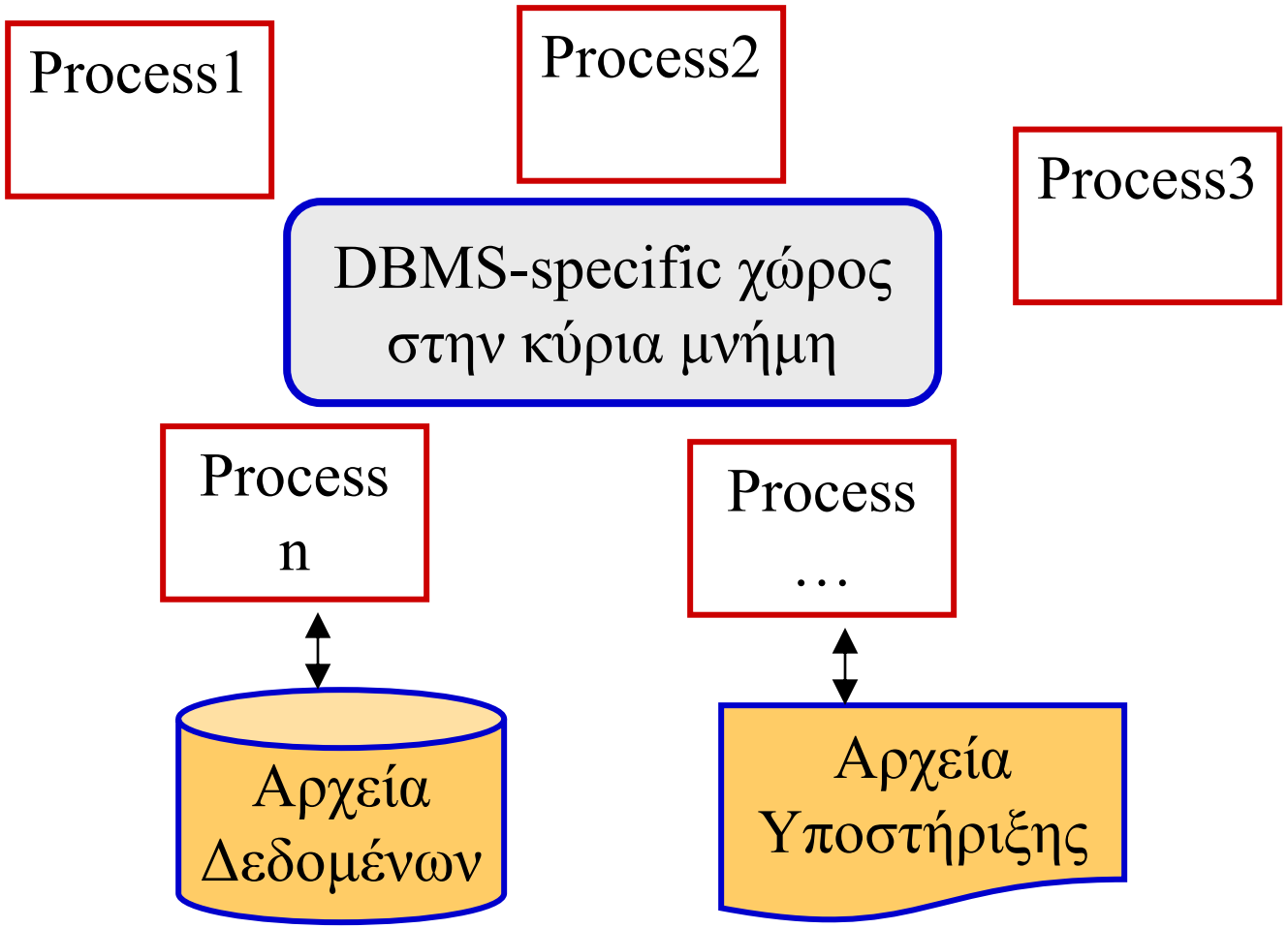
B+-tree index on emp_id

- **Όψεις**

Supervises(emp-id, manager-id)



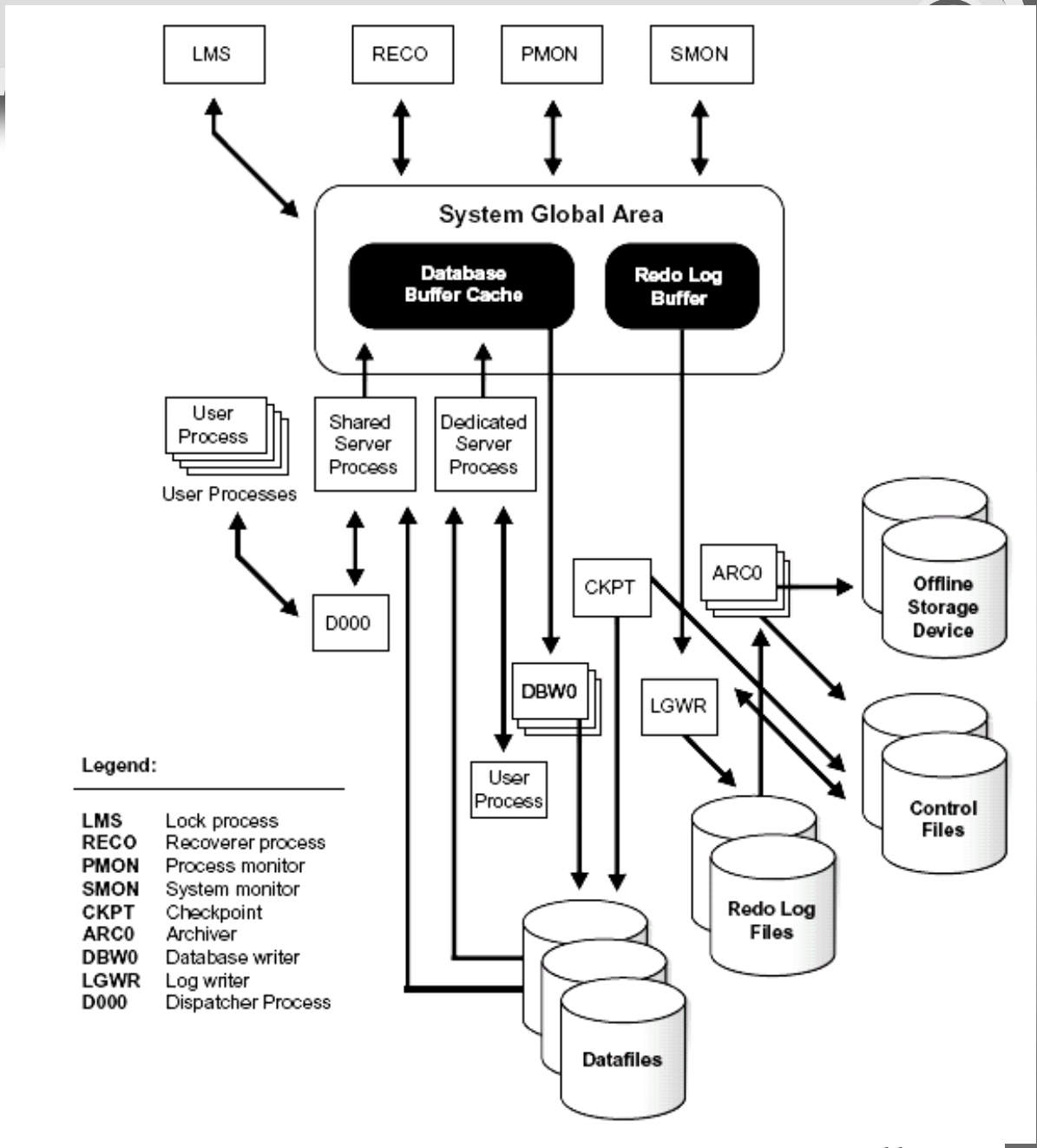
Γενική αρχιτεκτονική ενός DBMS στο runtime



DBMS στο runtime

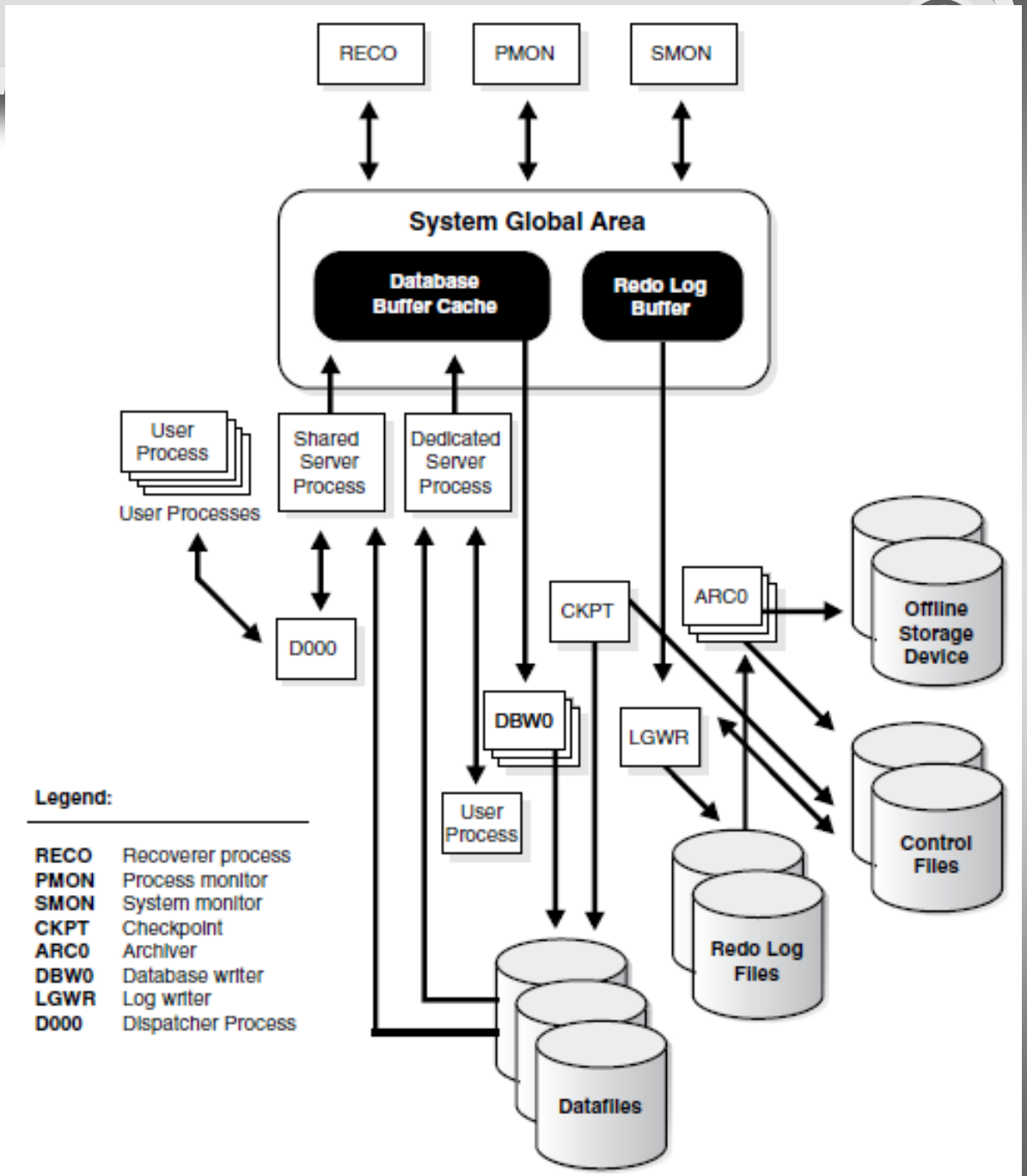
- Όταν ξεκινάμε το DBMS λέμε ότι ξεκινάμε ένα **στιγμιότυπο του DBMS** (**DBMS instance**).
- Ένα instance αποτελείται από
  - το χώρο που καταλαμβάνει στην κύρια μνήμη για να διεκπεραιώσει τη λειτουργία του DBMS
  - τις υποστηρικτικές διεργασίες που είτε συνδέουν το instance με τις εφαρμογές, είτε με τα δεδομένα, είτε κάνουν διάφορες διαχειριστικές λειτουργίες

Η αρχιτεκτονική της Oracle 9i



Η αρχιτεκτονική της Oracle 11g

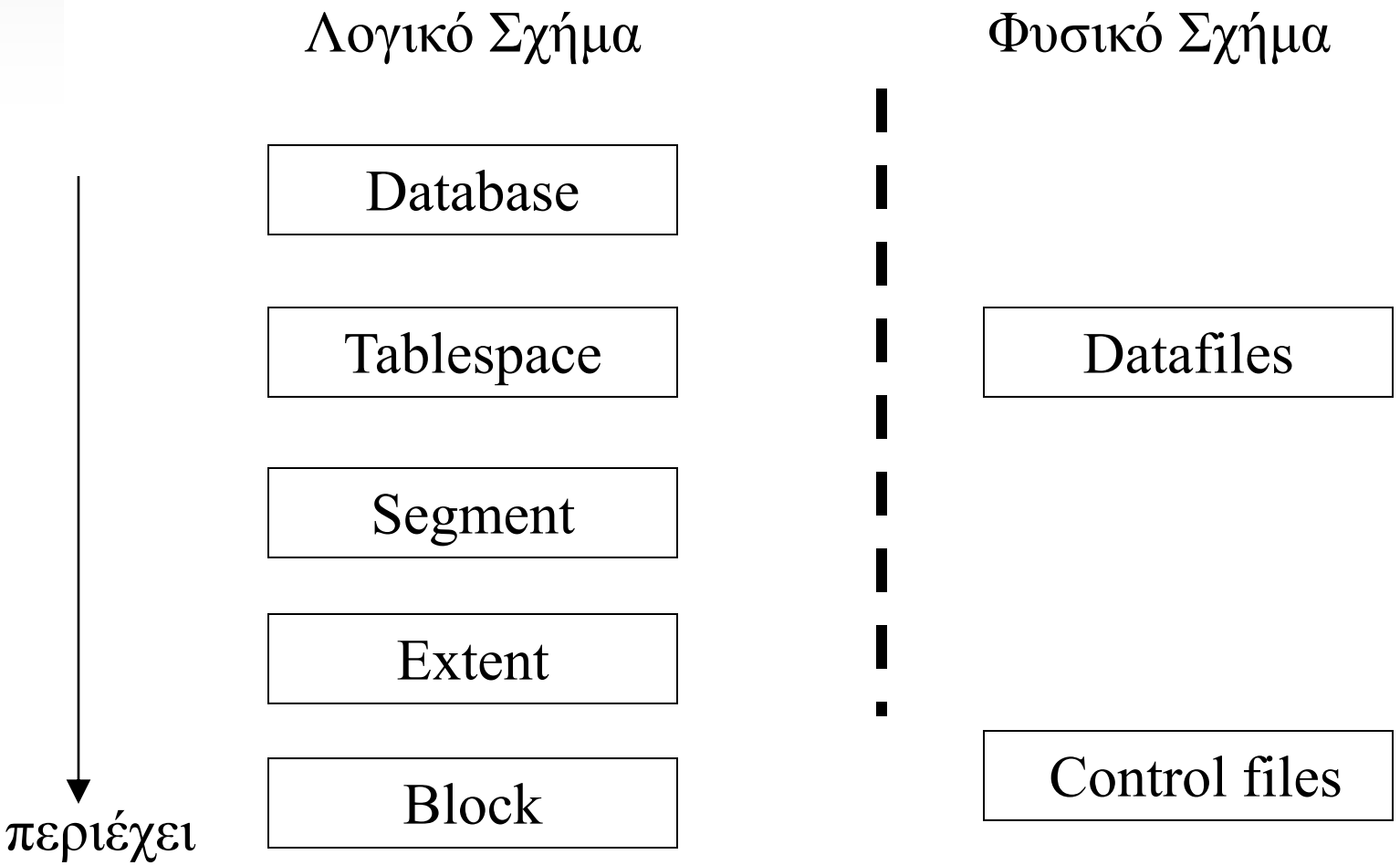
Τα DBMSs είναι πολύ ώριμο λογισμικό...



Θεματολόγιο

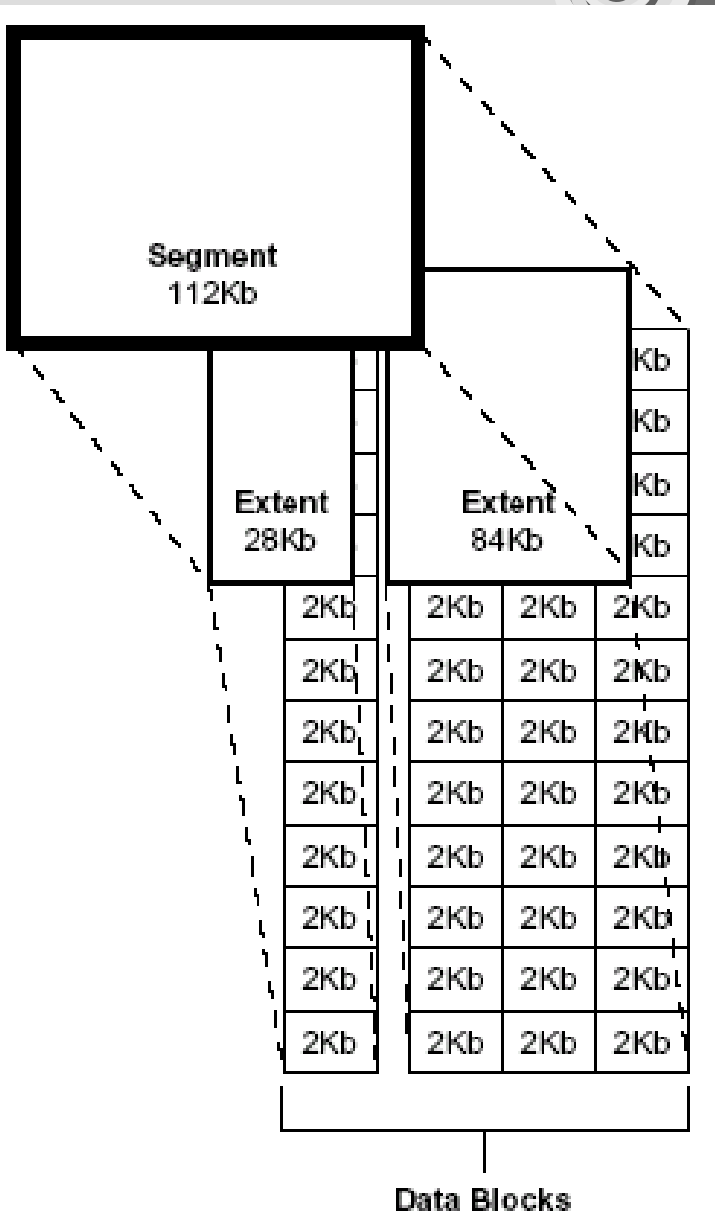
- Γενική αρχιτεκτονική βάσεων δεδομένων & ΣΔΒΔ
- Εσωτερική αποθήκευση δεδομένων
- Δομές στη μνήμη
- Διεργασίες
- Παράρτημα: Αξιοπιστία και διαθεσιμότητα βάσεων δεδομένων

Εσωτερική Οργάνωση των Δεδομένων στην Oracle

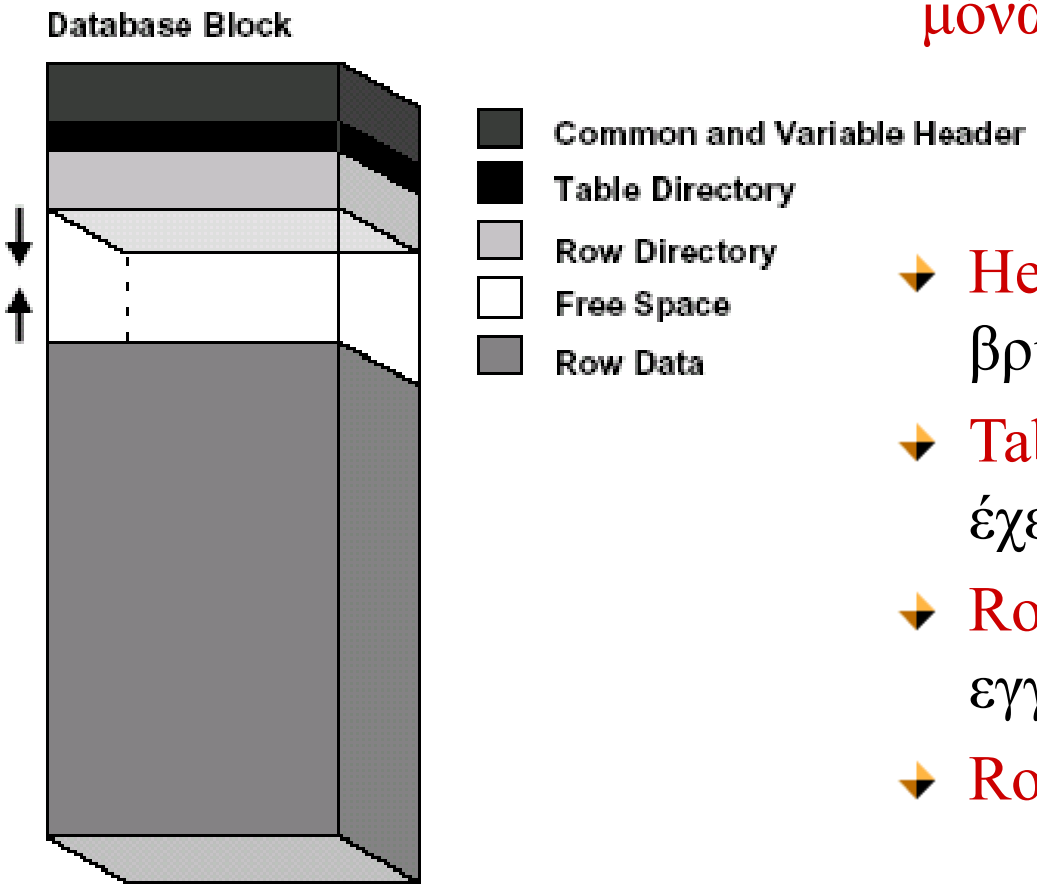


Εσωτερική Οργάνωση των Δεδομένων

- **Block** ή **page**: ένα σύνολο bytes στο σκληρό δίσκο
- **Extent**: ένα σύνολο από συνεχόμενα blocks
- **Segment**: ένα σύνολο από extent που αντιστοιχούν σε μια λογική οντότητα (π.χ., πίνακα ή ευρετήριο)

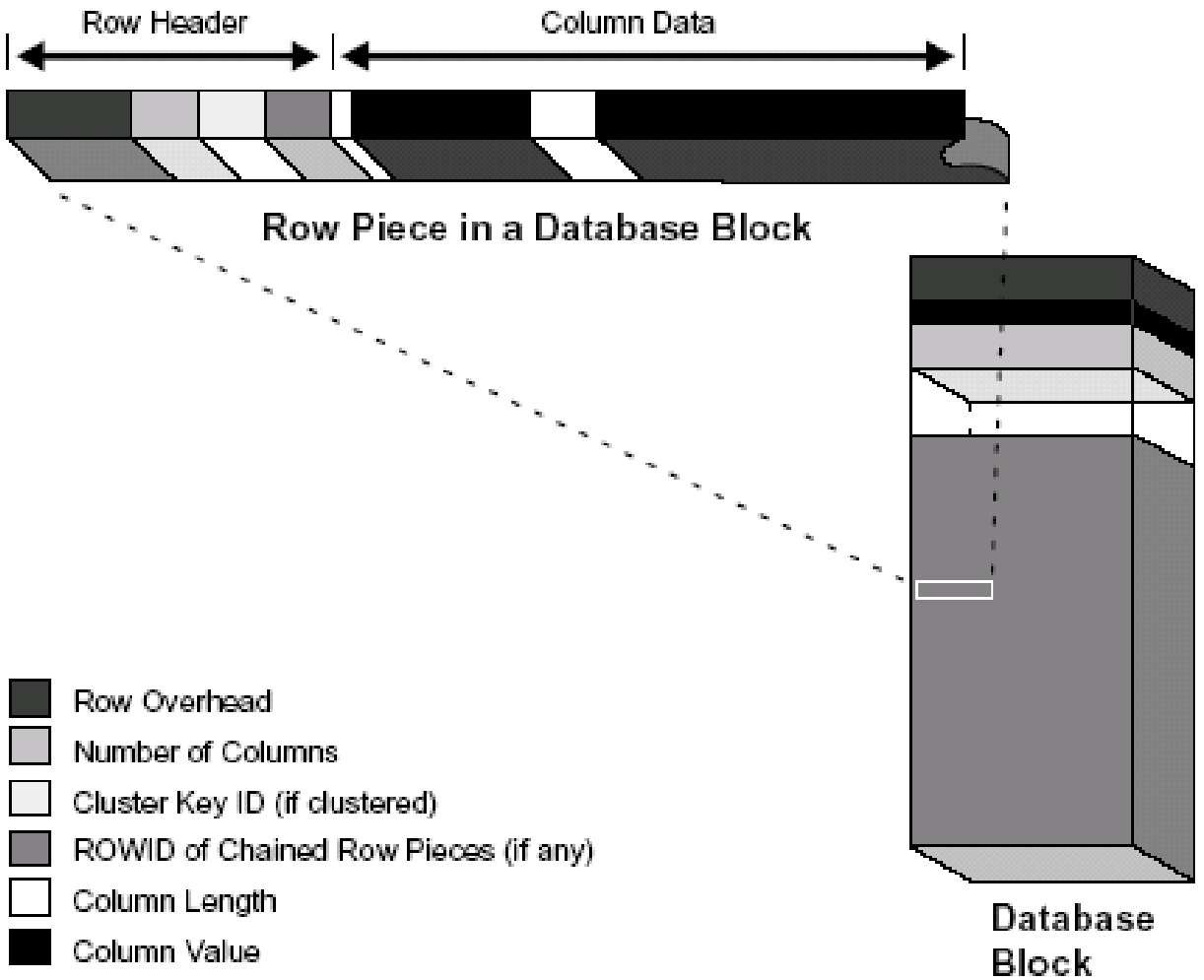


Εσωτερική Οργάνωση των Δεδομένων σε ένα block

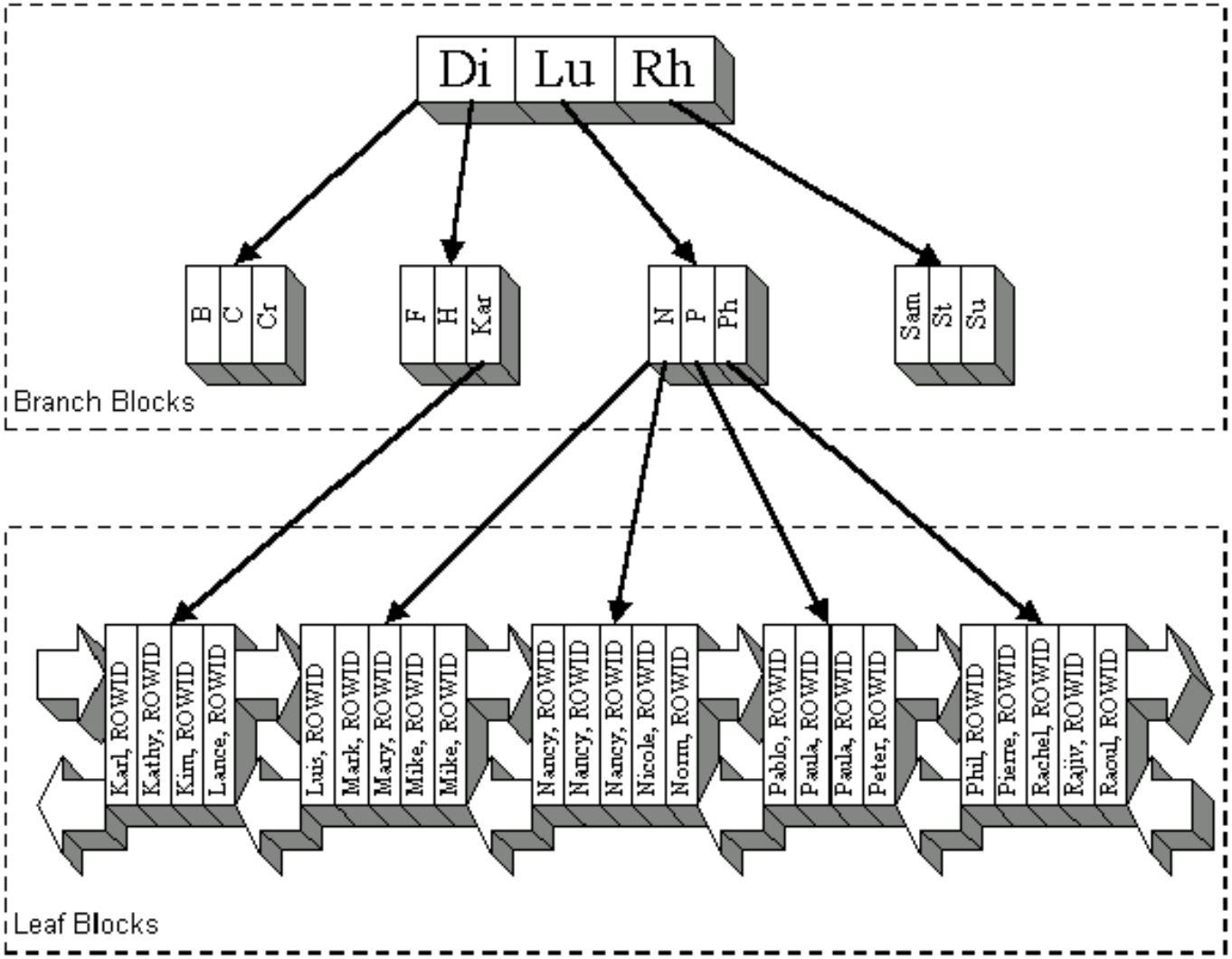


- Το block είναι η μικρότερη μονάδα I/O της ΒΔ
- Header: σε τι segment βρίσκεται το block
- Table Dir.: ποιος πίνακας έχει εγγραφές στο block
- Row Dir.: οι δ/σεις των εγγραφών στο block
- Row Data: οι εγγραφές

Εσωτερική Οργάνωση των Δεδομένων σε μια εγγραφή



Εσωτερική Οργάνωση των Δεδομένων σε ένα ευρετήριο



Εσωτερική Οργάνωση των Δεδομένων σε ένα extent

- Ένα **extent** είναι μια λογική μονάδα αποθήκευσης που περιλαμβάνει **συνεχόμενα** blocks.
- Το extent είναι η μονάδα βάση της οποίας το DBMS αποκτά **νέο (συνεχόμενο) χώρο** στο δίσκο όταν ένα segment γεμίσει

Εσωτερική Οργάνωση των Δεδομένων σε ένα segment

- Ένα **segment** είναι μια λογική μονάδα αποθήκευσης που αφορά τα δεδομένα μιας λογικής οντότητας
- Έχουμε 4 ειδών segment:
  - Table segments
  - Index segments
  - Temporary segments
  - Rollback segments

Table & Index Segments

- Όλα τα δεδομένα ενός πίνακα βρίσκονται σε ένα table segment
- Όλα τα δεδομένα ενός ευρετηρίου βρίσκονται σε ένα index segment

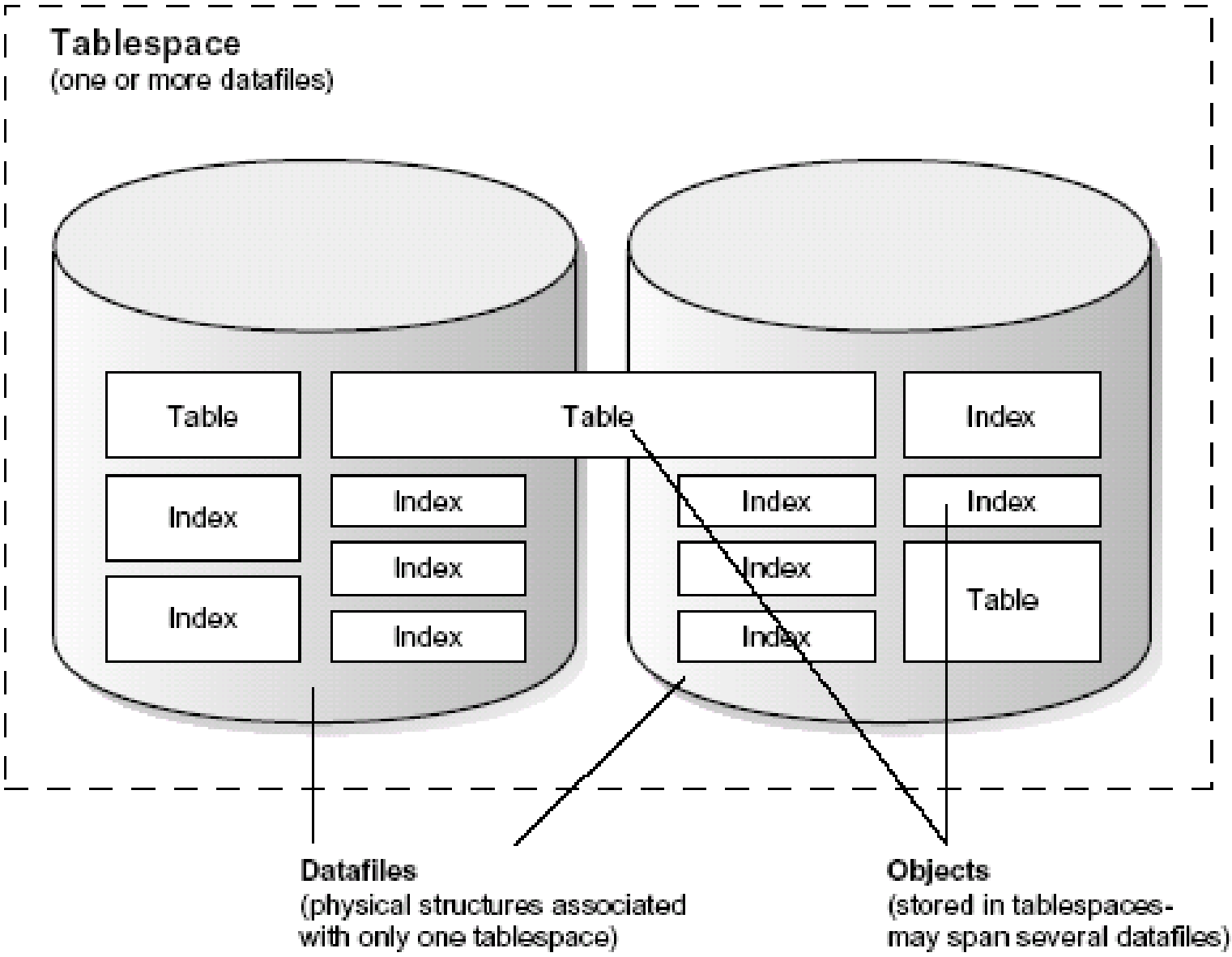
Για την ακρίβεια, υπάρχουν και άλλες περιπτώσεις που θα μάθετε όταν εισάγουμε την έννοια του partition…

Temporary & Rollback Segments

- Όταν μια SQL εντολή χρειάζεται βοηθητικό χώρο στο δίσκο (διότι η κύρια μνήμη δεν φτάνει), καταφεύγει σε ένα **temporary segment**
- Συνήθως αφορά ταξινόμηση εγγραφών (sorting), σε πράξεις ORDER BY, GROUP BY, DISTINCT, …ή και CREATE INDEX

- Όταν μια συναλλαγή μεταβάλει δεδομένα, η προηγούμενη κατάστασή τους αποθηκεύεται σε ένα **rollback segment**
- Αν ο χρήστης ακυρώσει τη συναλλαγή ή το σύστημα αποτύχει, μπορούμε να βρούμε την προηγούμενη τιμή των δεδομένων από το rollback segment

Datafiles & Tablespaces



Control files

- Αρχεία που βοηθούν στο να καταγράφουν κρίσιμες πληροφορίες για μια ΒΔ
  - Όνομα ΒΔ
  - Datafiles της ΒΔ
  - Log files
  - Κρίσιμες ημερομηνίες για τη λειτουργία του συστήματος (π.χ., checkpoints)
  - ...

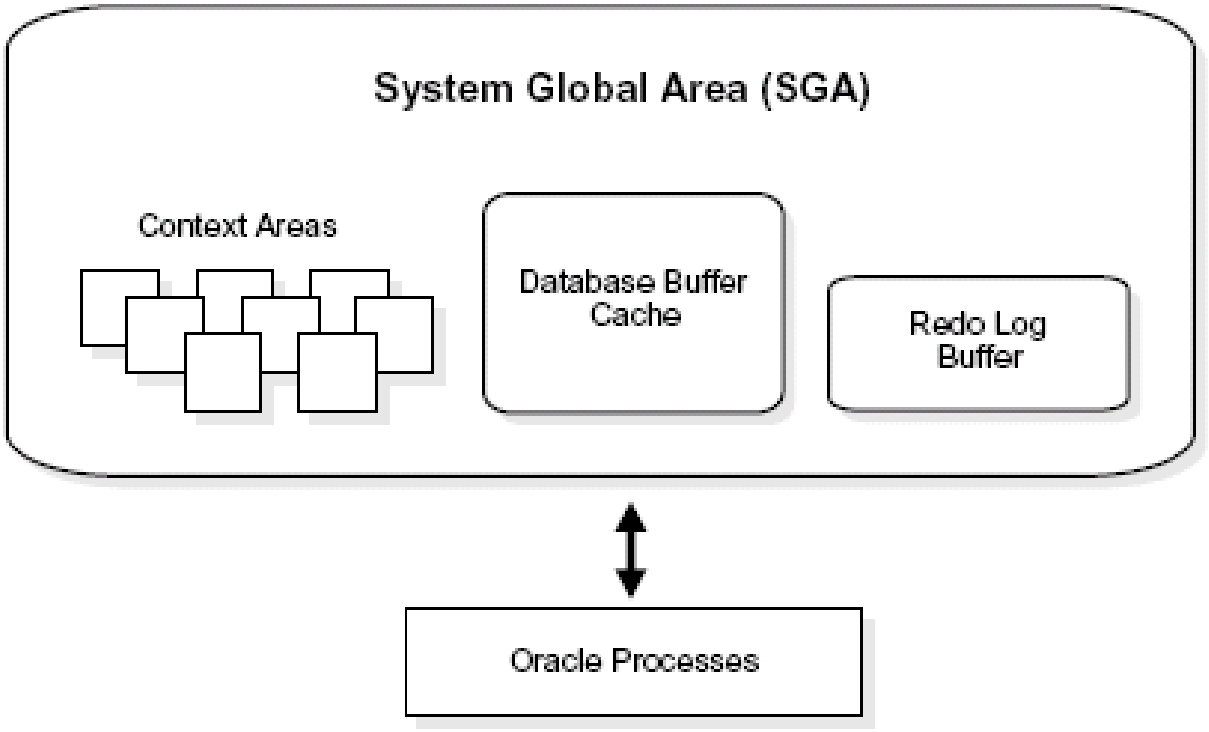
Θεματολόγιο

- Γενική αρχιτεκτονική βάσεων δεδομένων & ΣΔΒΔ
- Εσωτερική αποθήκευση δεδομένων
- Δομές στη μνήμη
- Διεργασίες
- Παράρτημα: Αξιοπιστία και διαθεσιμότητα βάσεων δεδομένων

DBMS στο runtime

- Όταν ξεκινάμε το DBMS λέμε ότι ξεκινάμε ένα **στιγμιότυπο του DBMS** (**DBMS instance**).
- Ένα instance αποτελείται από
  - το χώρο που καταλαμβάνει στην κύρια μνήμη για να διεκπεραιώσει τη λειτουργία του DBMS . Στην Oracle ο χώρος αυτός ονομάζεται System Global Area (SGA)
  - τις υποστηρικτικές διεργασίες που είτε συνδέουν το instance με τις εφαρμογές, είτε με τα δεδομένα, είτε κάνουν διάφορες διαχειριστικές λειτουργίες

Oracle Instance



Διαδικασία εκκίνησης/τερματισμού

- Η διαδικασία έναρξης του instance είναι:

1. Εκκίνηση του instance, που αντιστοιχεί πρακτικά στην δέσμευση του σχετικού χώρου στη μνήμη και εκκίνηση των σχετικών διεργασιών
2. Mounting μια βάσης δεδομένων στο instance, που σημαίνει τη συσχέτιση της ΒΔ με το instance αυτό
3. Άνοιγμα (opening) της βάσης δεδομένων, που σημαίνει ότι η ΒΔ γίνεται διαθέσιμη στους χρήστες προς επερώτηση

- Προφανώς, η διαδικασία τερματισμού έχει την αντίστροφη σειρά

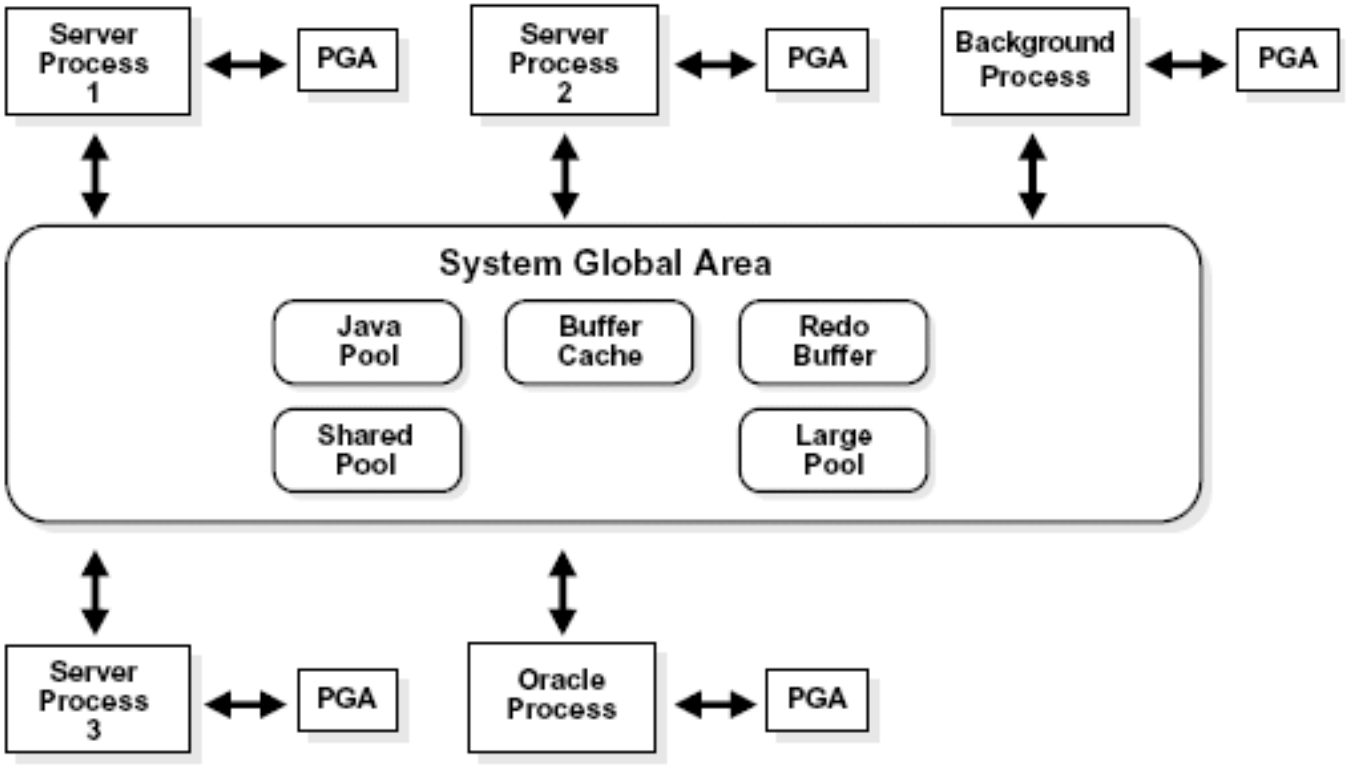
Διαδικασία εκκίνησης/τερματισμού

- Εάν δεν κάνουμε open τη ΒΔ, ο DBA μπορεί να επιτελέσει διαχειριστικές λειτουργίες
- Εναλλακτικά, μπορεί να ανοίξει τη ΒΔ σε read-only mode
- Εάν κάποιο instance τερματιστεί με ανώμαλο τρόπο (π.χ., αποτυχία της μνήμης, ή του δίσκου) τότε οι διαχειριστικές λειτουργίες που το DBMS κάνει αυτόματα γίνονται στο άνοιγμα της ΒΔ

Δομές στη μνήμη

- Δύο είναι οι βασικές δομές στη μνήμη:
- System Global Area (SGA) και περιλαμβάνει κυρίως δομές για την επικοινωνία δίσκου και μνήμης:
  - data buffers
  - log buffers
- Program Global Area (PGA) που ουσιαστικά κρατά τις λεπτομέρειες για κάθε διαδικασία του συστήματος και κυρίως για κάθε σύνδεση χρήστη

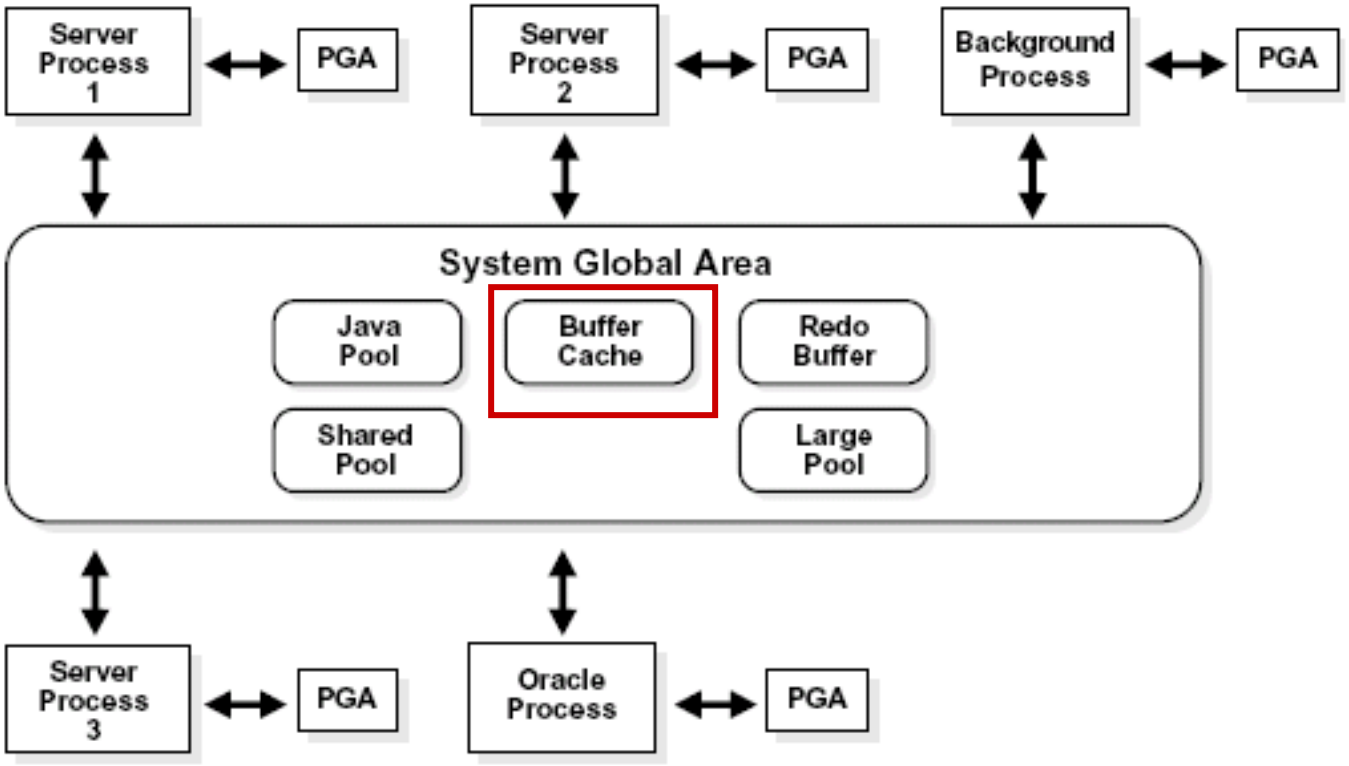
Δομές στην μνήμη



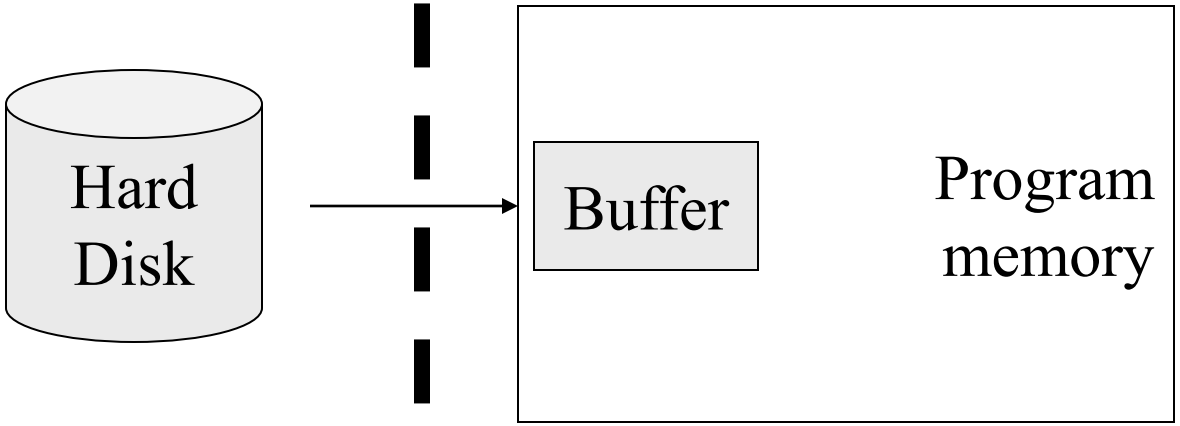
System Global Area

- Η SGA περιέχει δομές δεδομένων κοινές σε όλους τους χρήστες
  - Data Buffer Cache
  - Log Buffer
  - Shared Pool
  - …

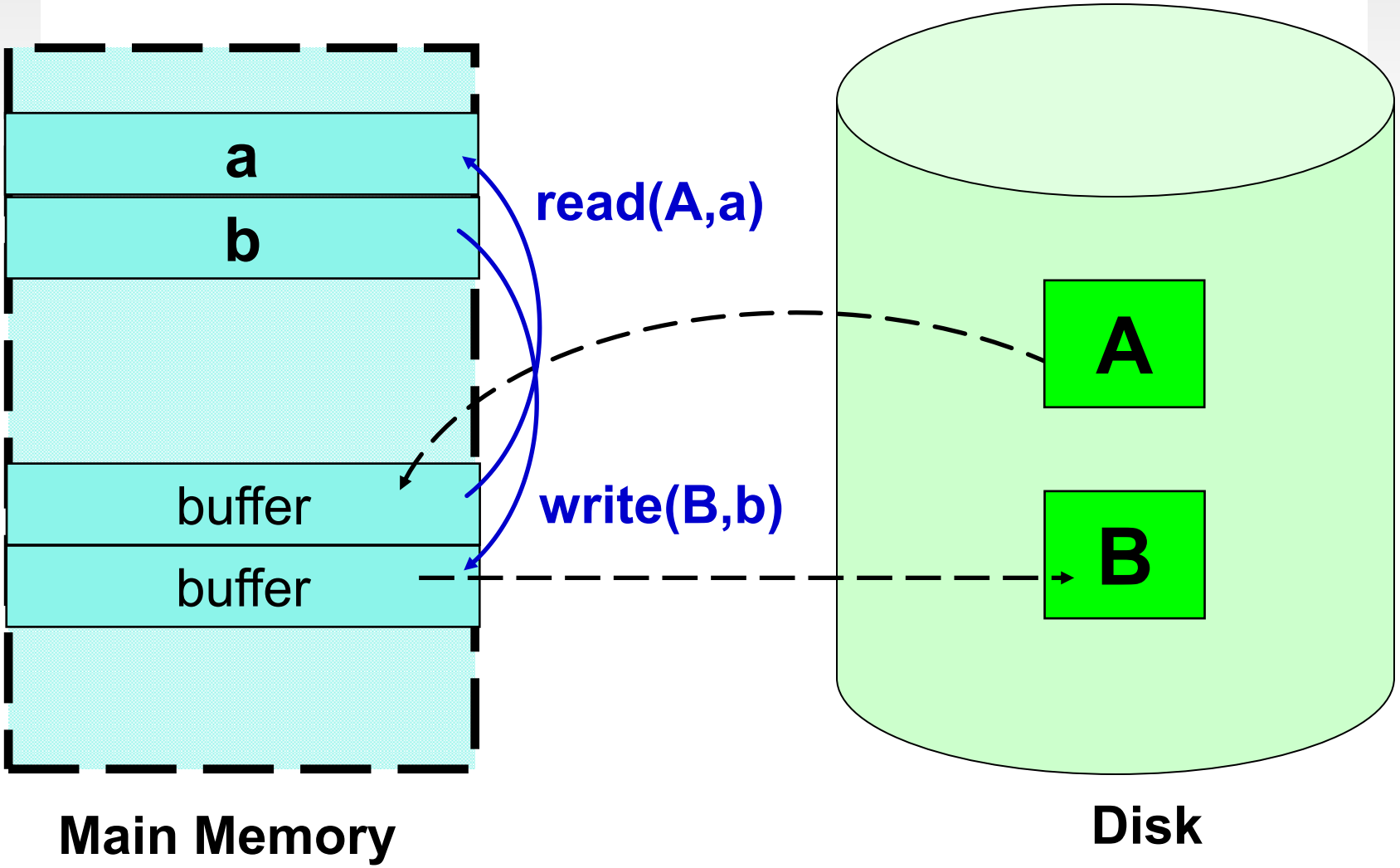
Δομές στην μνήμη



Data Buffer Cache



Data Buffer Cache



Data Buffer Cache

- Η Data Cache είναι μια ειδική περιοχή της μνήμης από (προς) την οποία μεταφέρουμε σελίδες δεδομένων προς (από) το σκληρό δίσκο

- Ξανά: Τα δεδομένα είναι οργανωμένα σε blocks (pages) και θεωρήστε αντιστοιχία από ένα block προς ένα buffer χάριν απλότητας

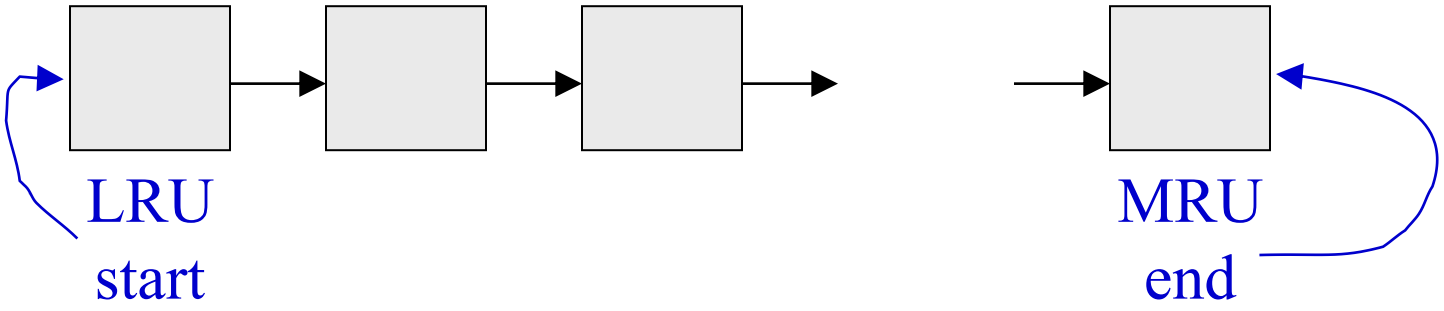
Data Buffer Cache

- Είδη buffers
  - Free buffer: δεν έχει χρησιμοποιηθεί ακόμα
  - Pinned buffer: χρησιμοποιείται από μια διεργασία
  - Dirty buffer: αν μεταφέρουμε κάποια σελίδα στη μνήμη και μεταβάλλουμε τα δεδομένα της, ο buffer ονομάζεται βρώμικος(dirty)

Data Buffer Cache

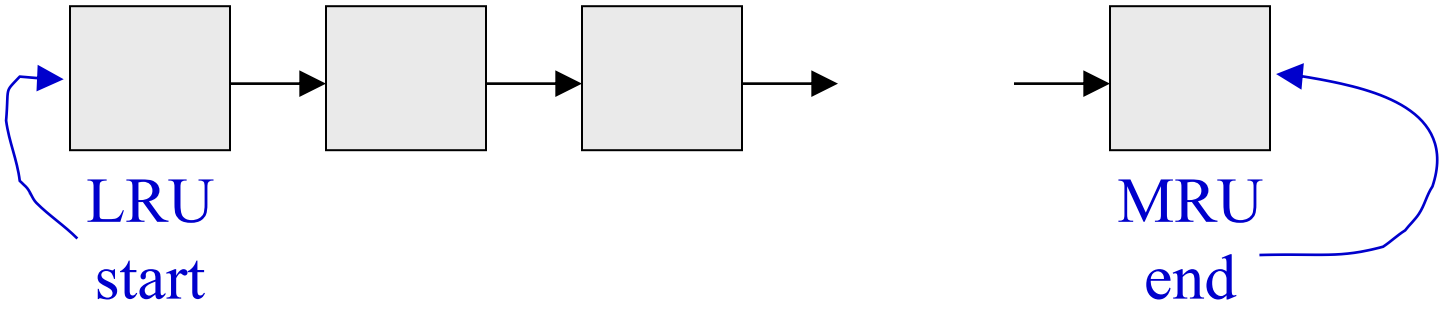
- Χρησιμοποιούμε δύο λίστες, την Write List και την LRU list για την οργάνωση των buffers στη μνήμη
- Write list: κάθε dirty buffer μπαίνει στη write list μέχρι να τον αντιγράψουμε στο δίσκο
- Least Recently Used -- LRU list: (i) ελεύθερους, (ii) pinned και (iii) dirty buffers που έχουν αντιγραφεί στο σκληρό δίσκο

Data Buffer Cache



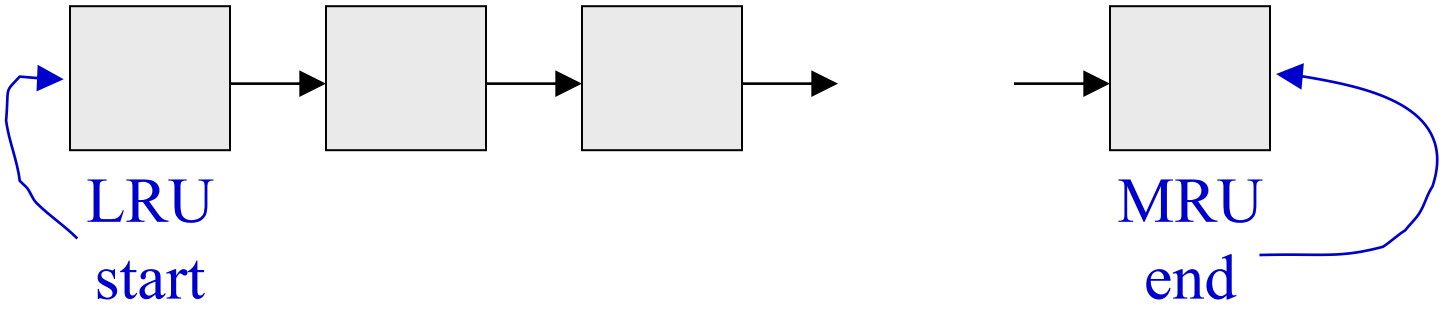
- Κάθε φορά που για κάποιο λόγο προσπελαύνουμε ένα buffer από την LRU αρχή της LRU λίστας, τον τοποθετούμε στο MRU τέλος της λίστας (είναι όντως ο most recently used)
- Κάποια στιγμή, κάποιοι dirty buffers είναι πιθανόν να εμφανιστούν στην αρχή της λίστας...

Data Buffer Cache



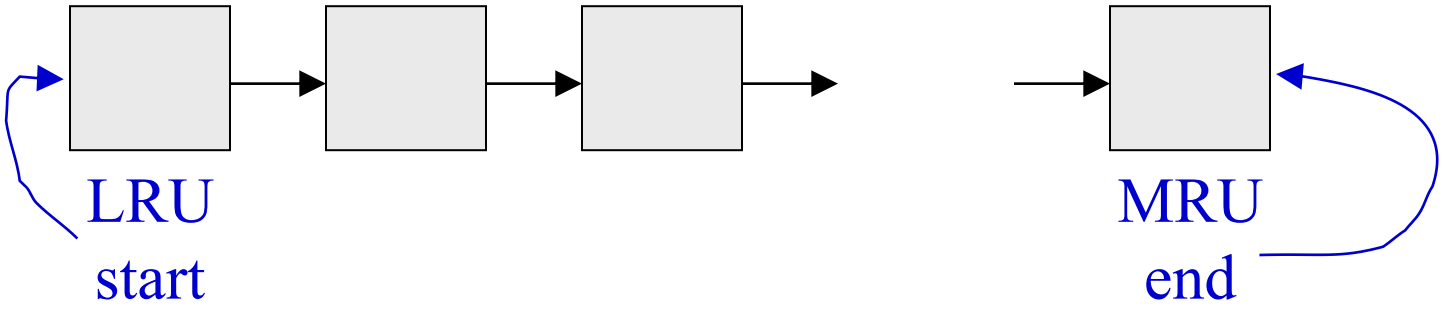
- Όταν κάποια ερώτηση ψάχνει κάποιο record, ψάχνουμε όλη την cache
- Αν το βρούμε (cache hit), επιστρέφουμε το record
- Αλλιώς, (cache miss) το αναζητούμε στο δίσκο. Σε αυτή την περίπτωση, πρέπει να βρούμε ένα buffer για να βάζουμε τις σελίδες που θα χρησιμοποιούμε

Data Buffer Cache



- Αναζήτηση ελεύθερου buffer:
  - Το σύστημα ψάχνει την LRU list μέχρι να βρει ελεύθερο buffer.
  - Αν ξεπεράσει κάποιο όριο (threshold) buffers χωρίς να βρει, τότε γράφει buffers στο δίσκο ώστε να μπορεί να τους χρησιμοποιήσει στη συνέχεια.
  - Ποιους? Τους dirty buffers που είναι οι least recently used => στην αρχή της λίστας

Data Buffer Cache

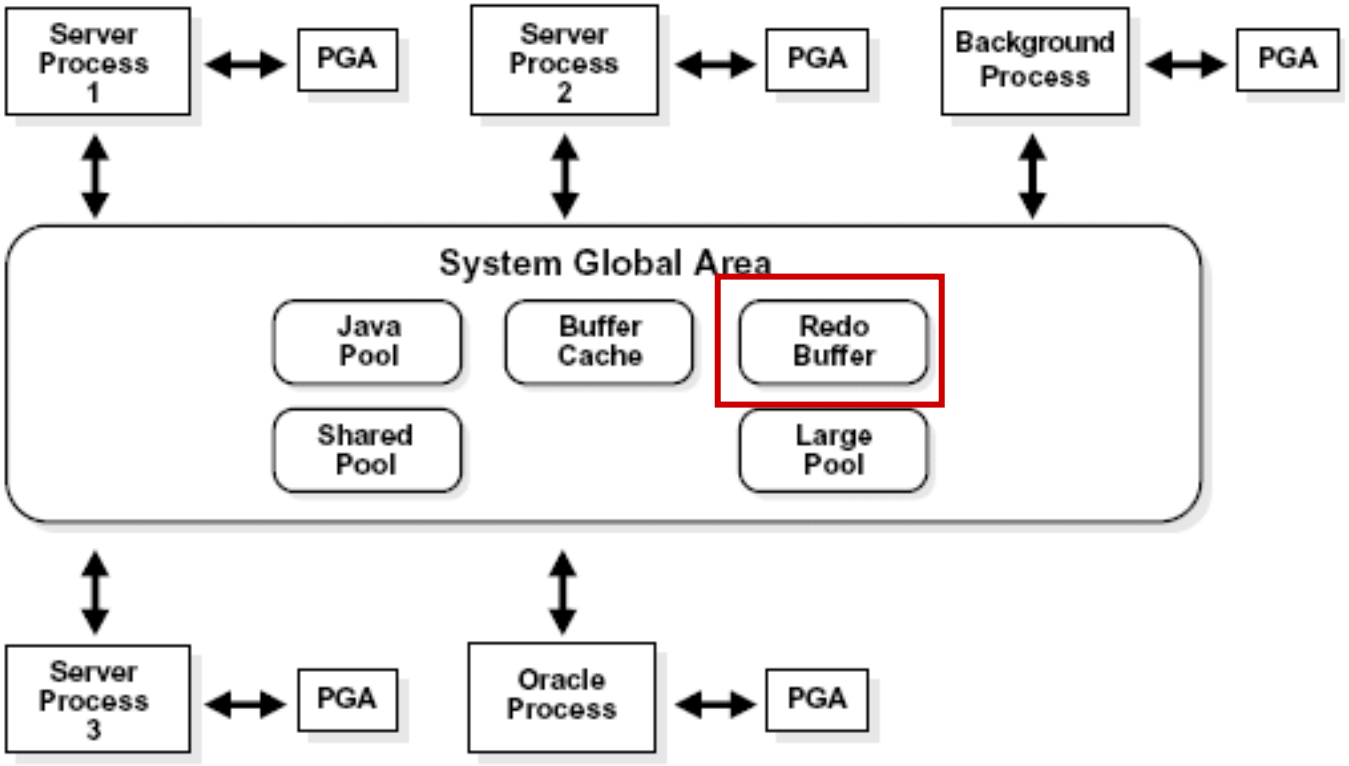


- Εν γένει, αν γεμίσει η cache, και έχει συνεχόμενες cache misses, το σύστημα γράφει στο δίσκο dirty buffers ώστε να κάνει χώρο.
- Επίσης κάθε 20-30 min, στα checkpoints κάνει το ίδιο...
- Πάλι, οι dirty buffers στην αρχή της LRU είναι εκείνοι που μεταφέρονται...

Συχνότητα Checkpoint

- ORACLE:
  - LOG_CHECKPOINT_TIMEOUT: 1800 sec (30')
  - Την ίδια στιγμή, η συχνότητα μεταξύ checkpoints για τα log records (βλ. παρακάτω) είναι 3 sec
- DB2:
  - ~10 – 15', εκπεφρασμένη ως ο αριθμός των log records που δημιουργήθηκαν ανάμεσα σε δύο checkpoints (π.χ., 150000 log records)
- MS SQL server:
  - Automatic configuration (λέει)
  - Δέστε http://msdn.microsoft.com/en-us/library/ms189573.aspx

Δομές στην μνήμη



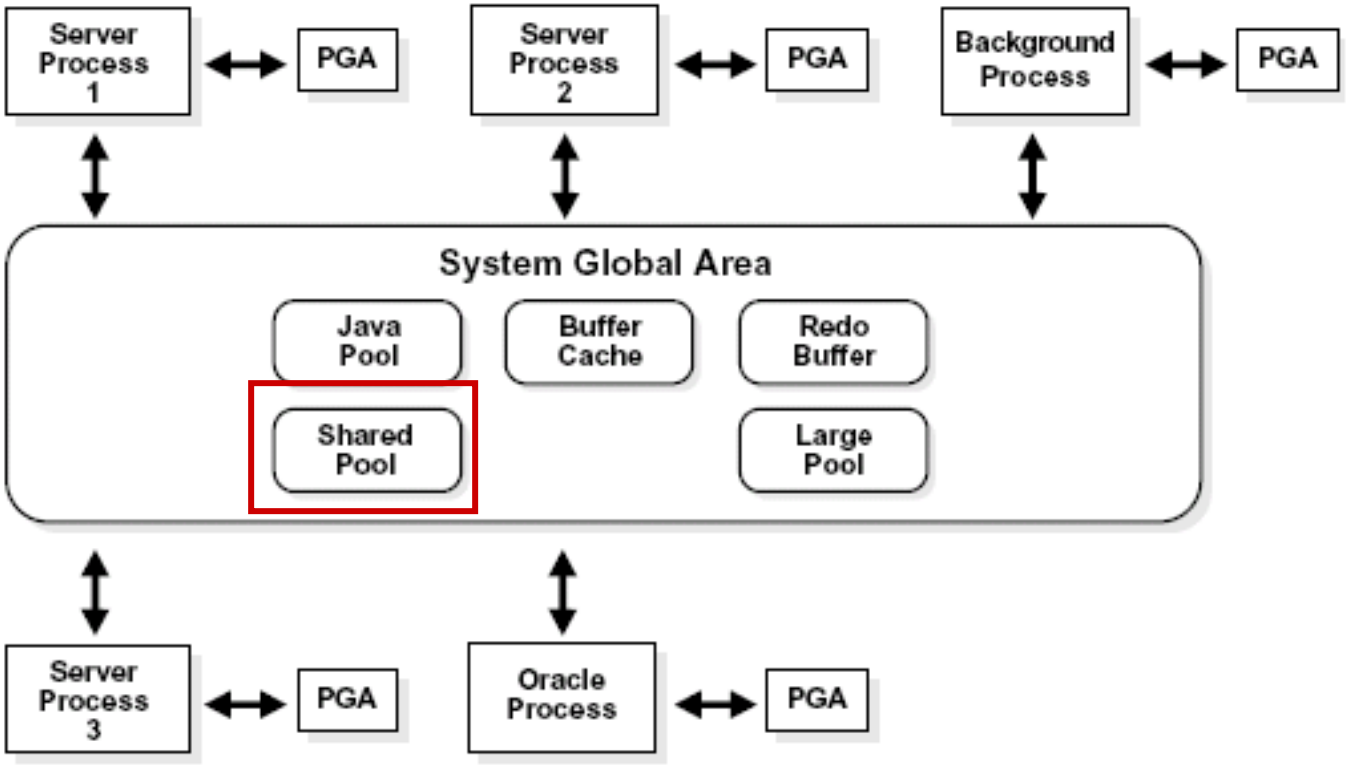
Log buffer

- Οι εντολές τύπου `INSERT`, `DELETE`, `UPDATE` μεταβάλλουν το περιεχόμενο της ΒΔ
- Για να μπορεί να κρατήσει την παλαιά κατάσταση της ΒΔ, το DBMS χρησιμοποιεί ένα ειδικού τύπου αρχείο, το **log file**, στο οποίο καταγράφει τις ενέργειές του
- Αν το σύστημα αποτύχει για κάποιο λόγο, με βάση τις εγγραφές του log file μπορούμε να επανακτήσουμε την παλαιά κατάσταση της βάσης.

Log buffer

- Ακριβώς επειδή το log file βρίσκεται και αυτό στο δίσκο, χρειαζόμαστε μια περιοχή στην κύρια μνήμη, αφιερωμένη στην διαχείριση του log file
- Στον log buffer καταγράφουμε τις εγγραφές που πρόκειται να τοποθετήσουμε στο log file
- Σε τακτά χρονικά διαστήματα (πχ., 1 sec), ή ΠΡΙΝ γράψουμε dirty buffers με δεδομένα στο δίσκο, ή αν γεμίσει παραπάνω από το 1/3 των log buffers γράφουμε (flush) τις σχετικές εγγραφές από το log buffer στο log file

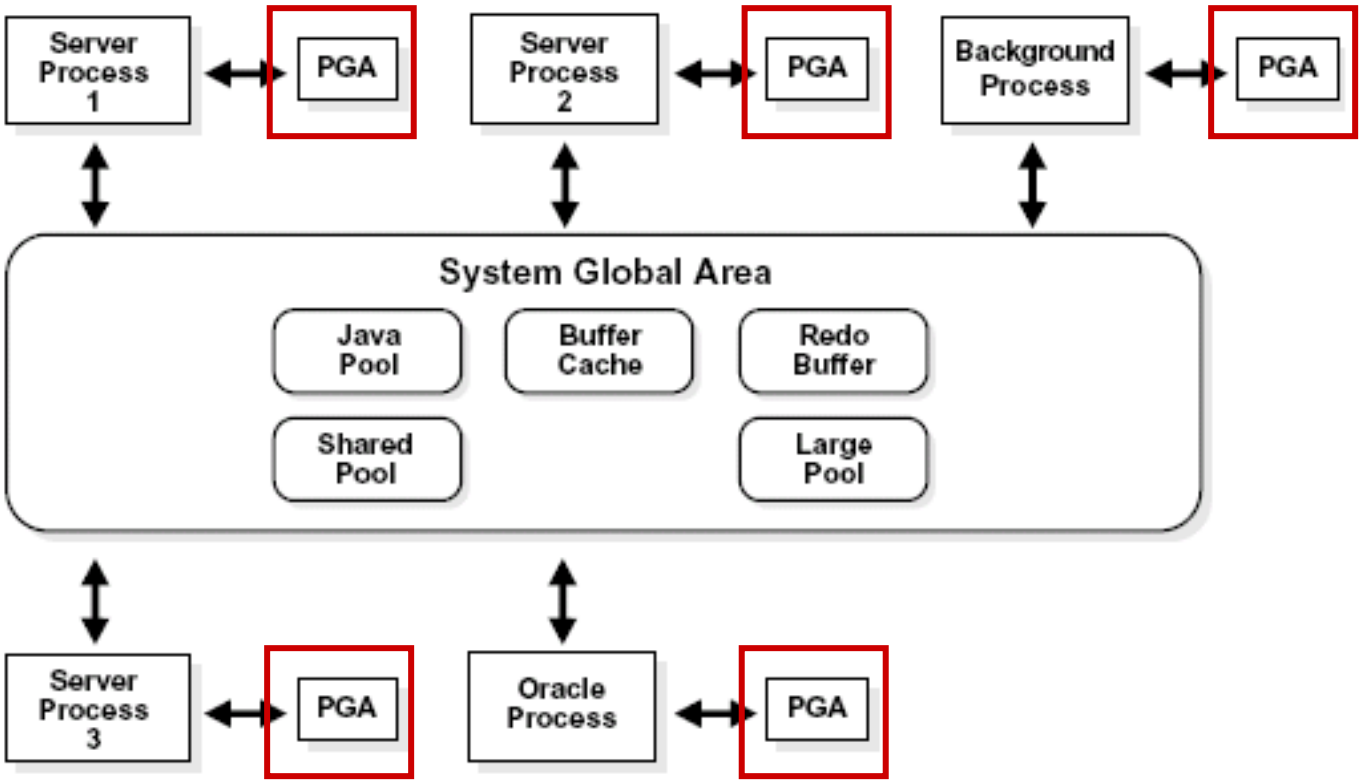
Δομές στην μνήμη



Shared Pool

- Η **κοινή περιοχή (shared pool)** χρησιμοποιείται για δύο λόγους:
- Data dictionary: λίστα με περιγραφές του σχήματος των πινάκων, των δικαιωμάτων πρόσβασης των χρηστών, κλπ.
- SQL area: τα parse trees & τα πλάνα εκτέλεσης των SQL εντολών μπορούν να επαναχρησιμοποιηθούν από πολλούς χρήστες, γι' αυτό και παραμένουν στην SQL area

Δομές στην μνήμη



Program Global Areas

- Ας υποθέσουμε ότι ένας χρήστης θέλει να συνδεθεί στη ΒΔ για να κάνει μια σειρά από ερωτήσεις
- Το DBMS κάνει allocate κάποιο τμήμα της κύριας μνήμης για την εξυπηρέτηση του process του εν λόγω χρήστη. Θα ονομάζουμε το process αυτό, server process.
- Η PGA για το εν λόγω process, θα έχει πληροφορία που αφορά τη σύνδεση του χρήστη

Program Global Areas

- Σε κάθε PGA, τριών ειδών πληροφορίες έχουμε:
  - Σταθερή (persistent) πληροφορία: ποιος είναι ο χρήστης, από πού συνδέεται και πώς, ...
  - Run-time πληροφορία: τα δεδομένα που σχετίζονται με μια SQL εντολή. Π.χ.,
    - Αν κάνουμε INSERT, οι νέες τιμές που θα εισαχθούν στη βάση
    - Αν κάνουμε SELECT, οι εγγραφές που ανασύραμε από τη βάση

Program Global Areas

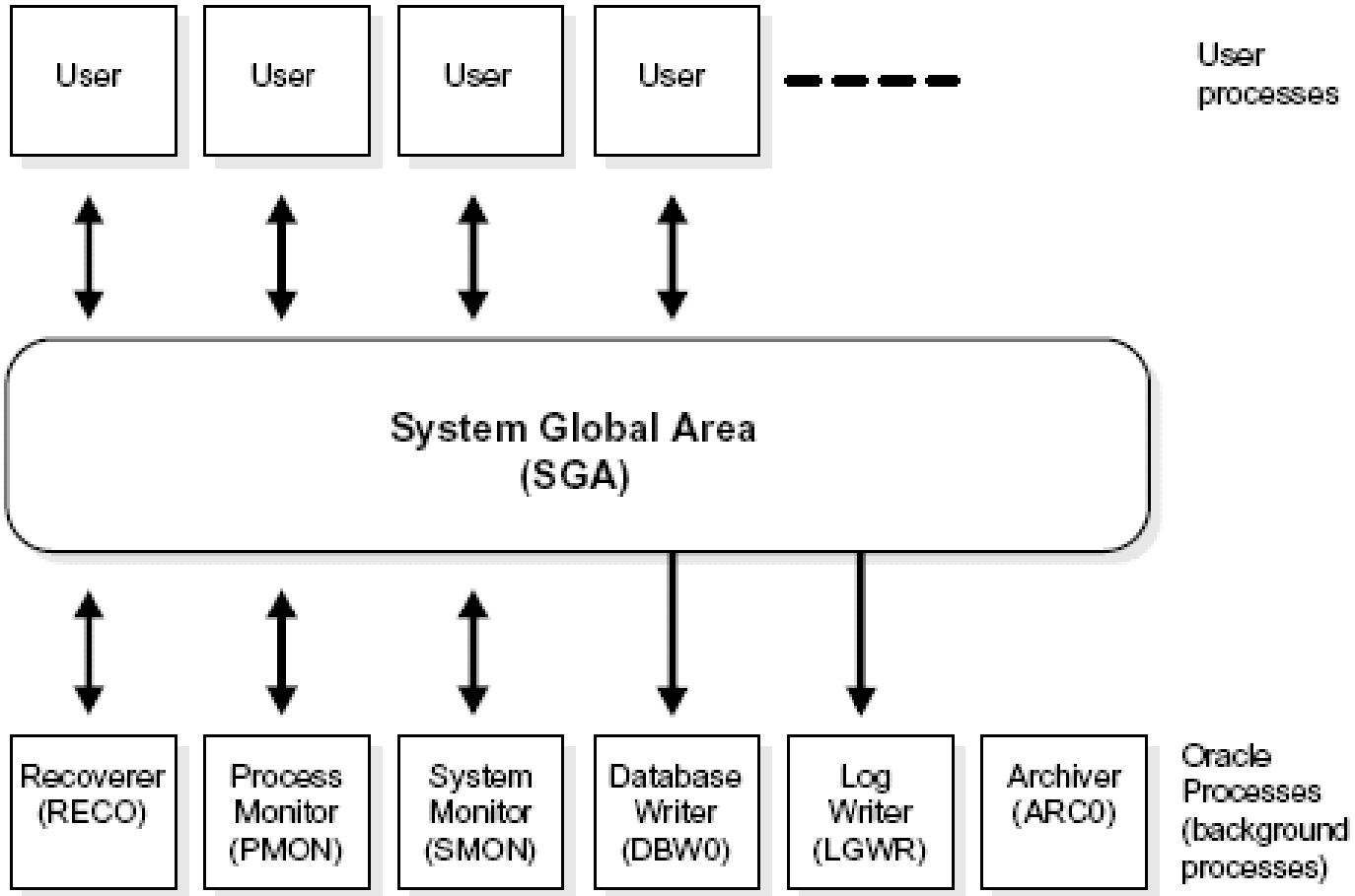
- Επίσης, υπάρχει χώρος για ενδιάμεσα αποτελέσματα των ερωτήσεων
- Π.χ., σε ένα `ORDER BY` query, χρειαζόμαστε επιπλέον χώρο για να ταξινομήσουμε τα δεδομένα

- **Προσοχή**: *όλες οι δύσκολες δουλειές στις ΒΔ παραδοσιακά λύνονται με ταξινόμηση*

Θεματολόγιο

- Γενική αρχιτεκτονική βάσεων δεδομένων & ΣΔΒΔ
- Εσωτερική αποθήκευση δεδομένων
- Δομές στη μνήμη
- Διεργασίες
- Παράρτημα: Αξιοπιστία και διαθεσιμότητα βάσεων δεδομένων

Διεργασίες


User
User
User
User
User processes
System Global Area
(SGA)
Recoverer
(RECO)
Process
Monitor
(PMON)
System
Monitor
(SMON)
Database
Writer
(DBW0)
Log
Writer
(LGWR)
Archiver
(ARC0)
Oracle
Processes
(background
processes)

Διεργασίες

- Υπάρχουν δύο ειδών διεργασίες:
- Οι διεργασίες των χρηστών (user processes) που πρακτικά είναι οι εφαρμογές ή κάποιο query prompt (τύπου sqlplus)
- Οι διεργασίες του DBMS που χωρίζονται:
  - Server processes
  - Background processes

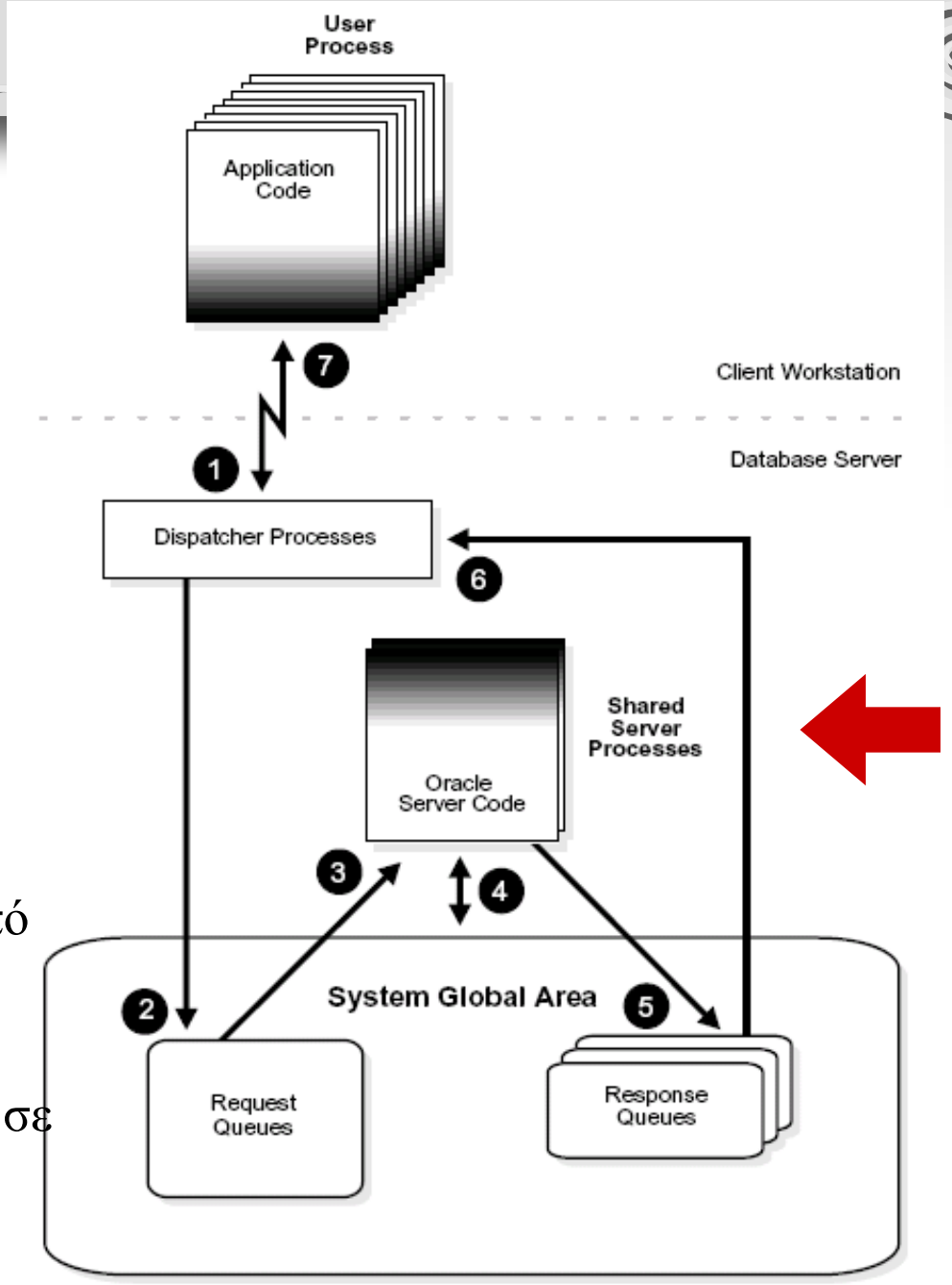
Διεργασίες χρηστών

- Κάποιες εφαρμογές στον client
- Ο χρήστης συνδέεται (`CONNECT`) με τη βάση δεδομένων με username & password
- Μια session είναι μια σειρά πράξεις που κάνει ο χρήστης μεταξύ connect & disconnect

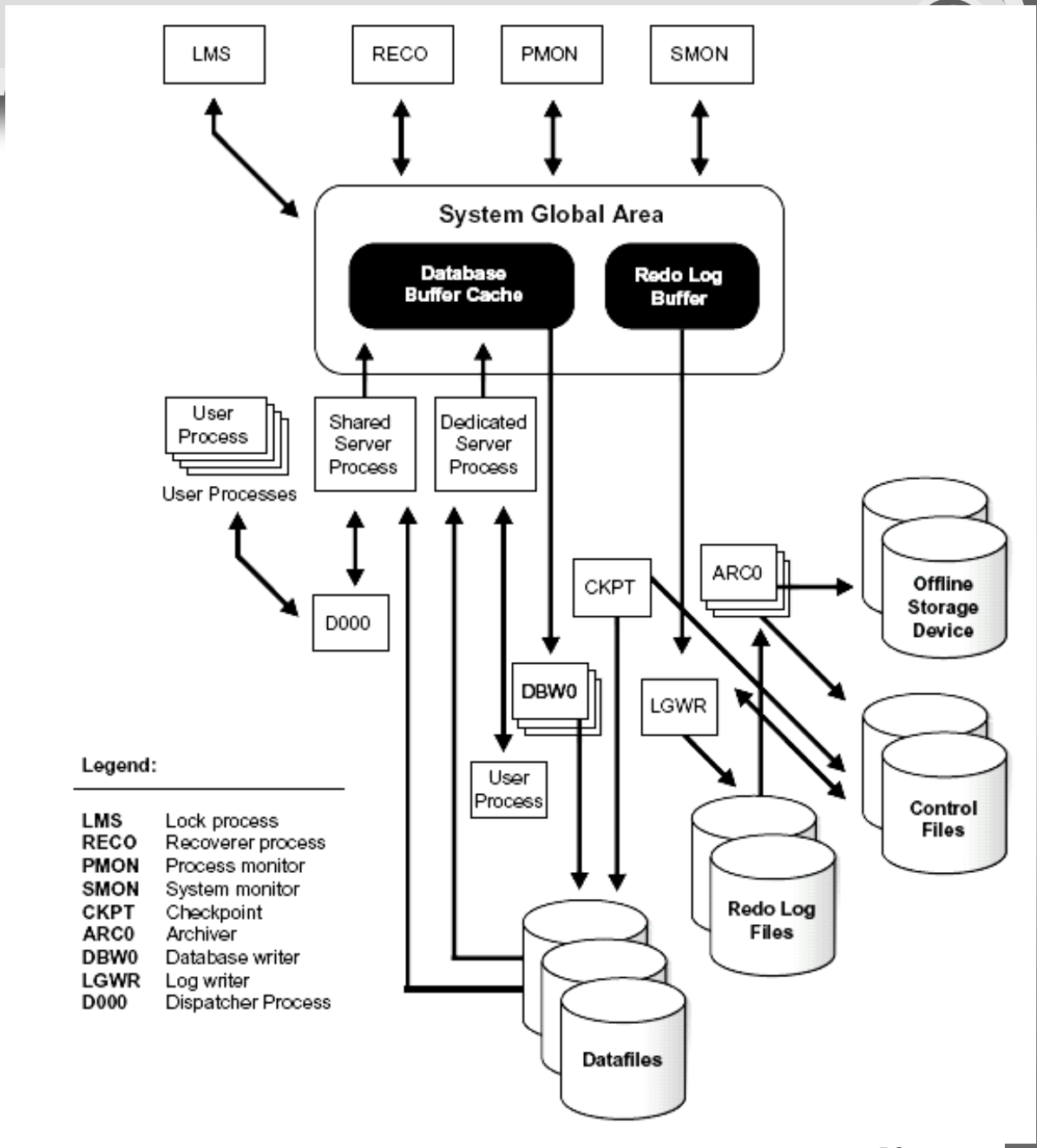
Server Processes

Οι **server processes** λαμβάνουν το αίτημα ενός χρήστη και

- Κάνουν parse και βελτιστοποιούν την SQL εντολή
- Εκτελούν την εντολή: μεταφέρουν τα δεδομένα από το δίσκο και κάνουν τις σχετικές πράξεις
- Τοποθετούν το αποτέλεσμα σε κατάλληλες δομές

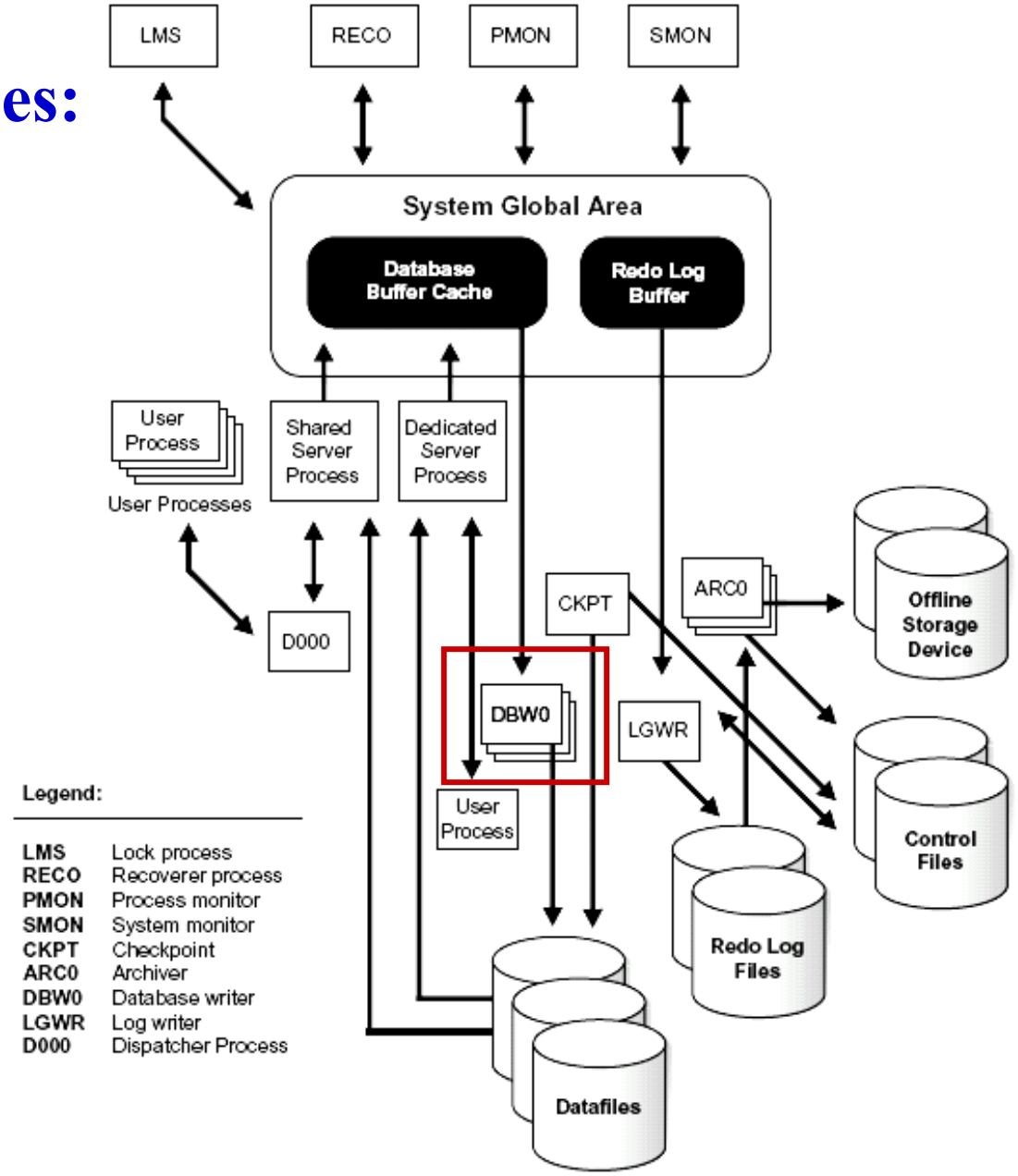


Background processes



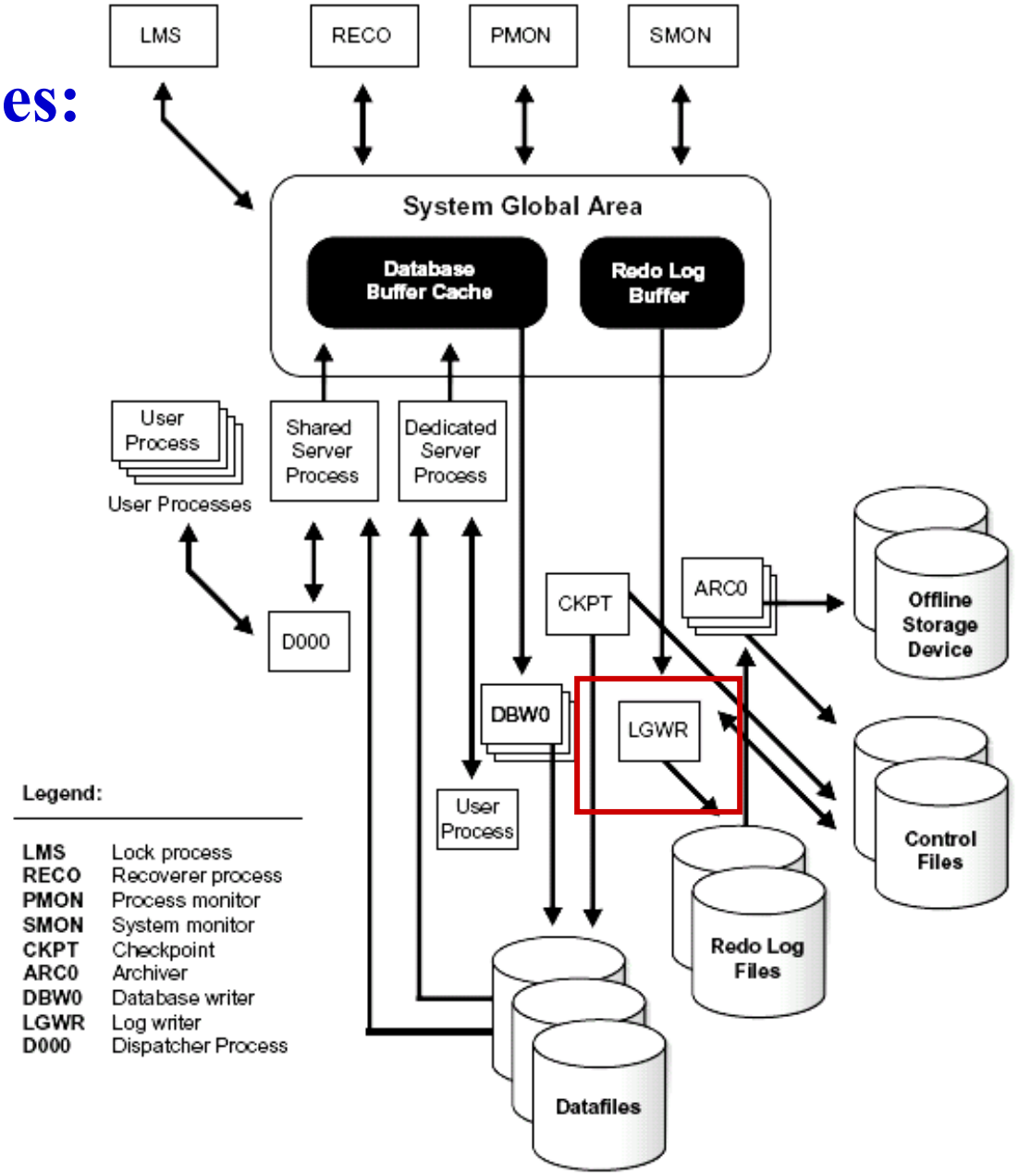
Background processes: Database Writer (DBW0)

Υπεύθυνος για να μεταφέρει σελίδες από την LRU list της SGA στο δίσκο



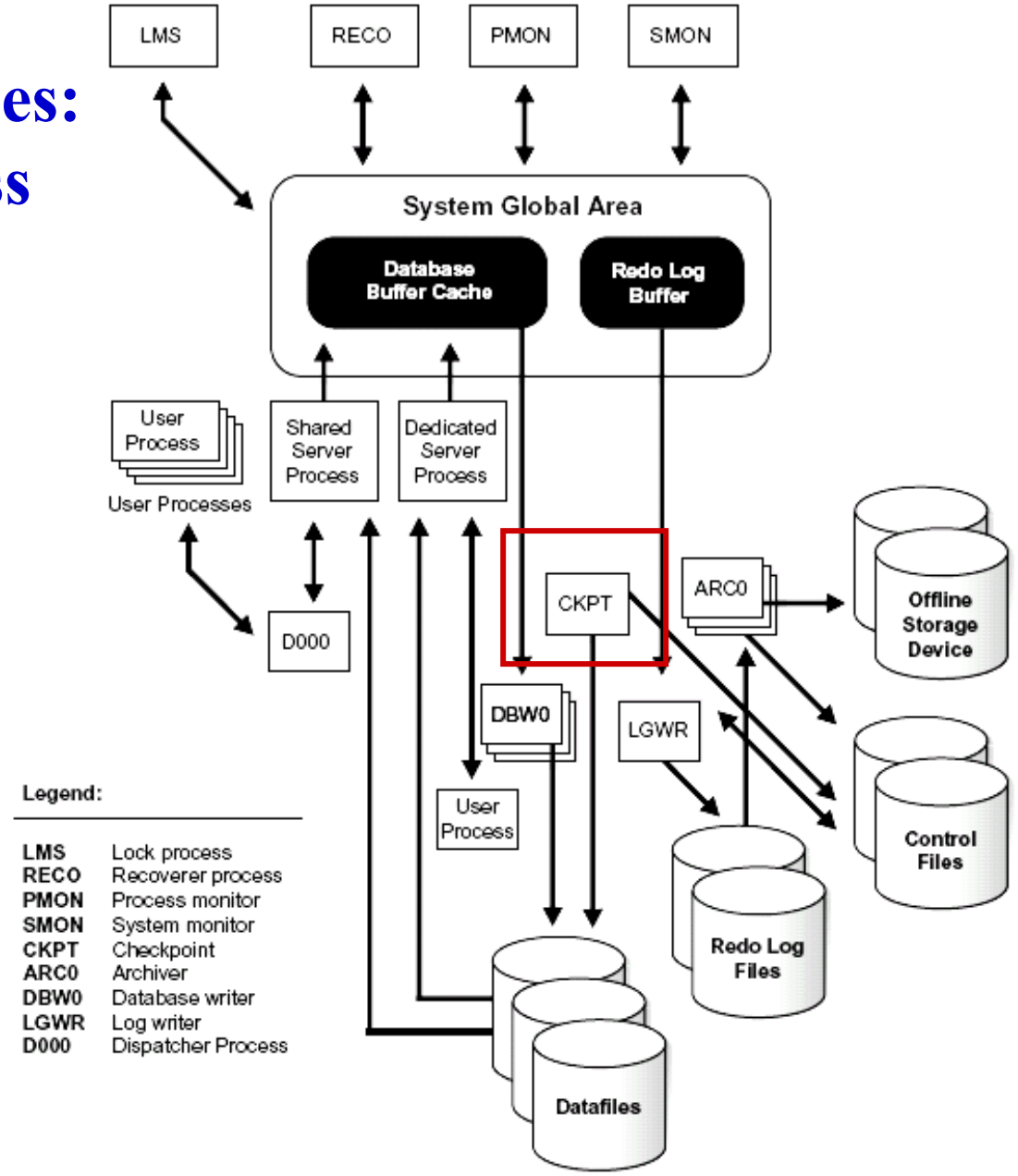
Background processes: Log Writer (LGW0)

Υπεύθυνος για να μεταφέρει σελίδες από τους log buffers στο log file



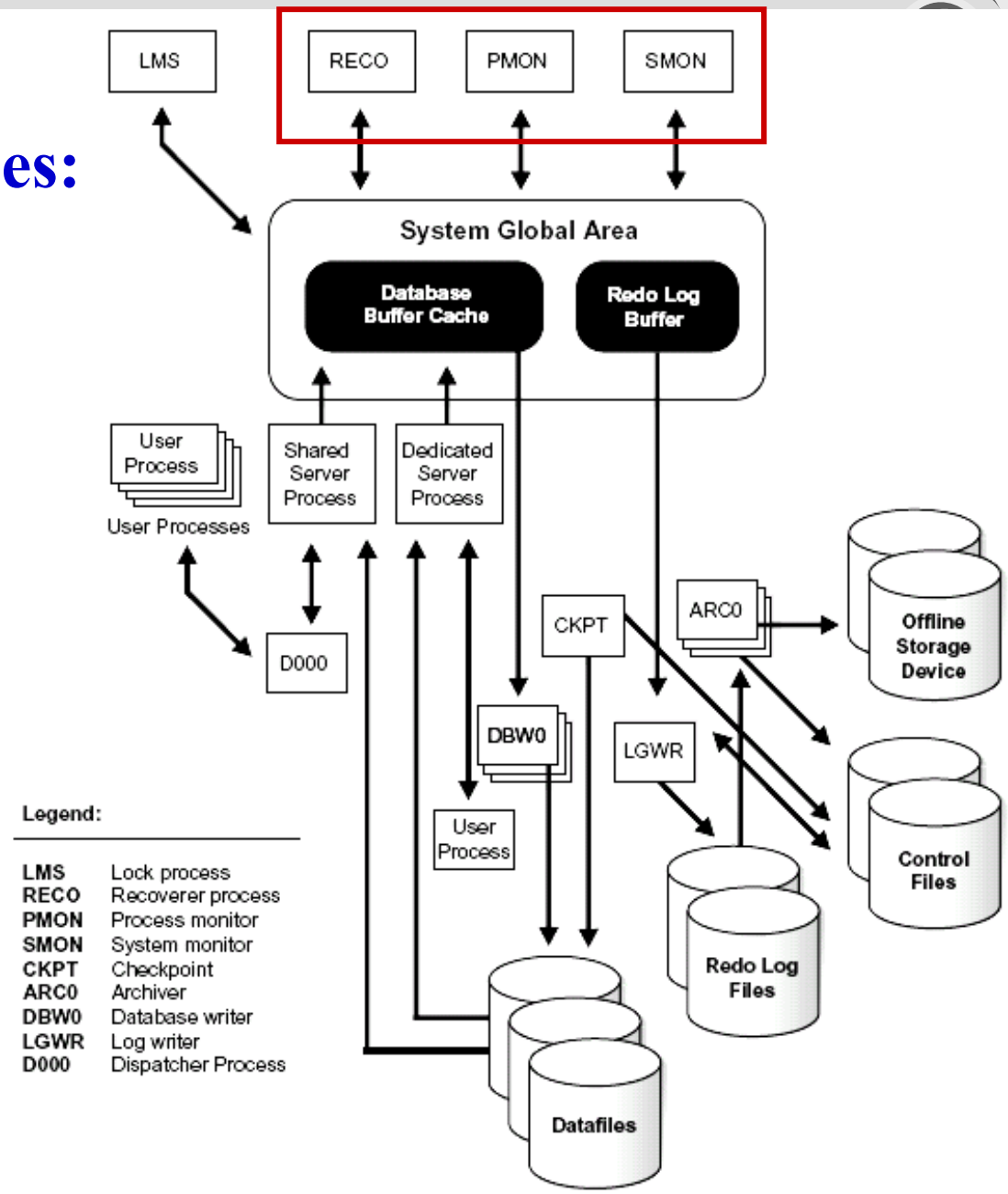
Background processes: Checkpoint Process (CKPT)

Υπεύθυνος για να κάνει checkpoint σε τακτά διαστήματα



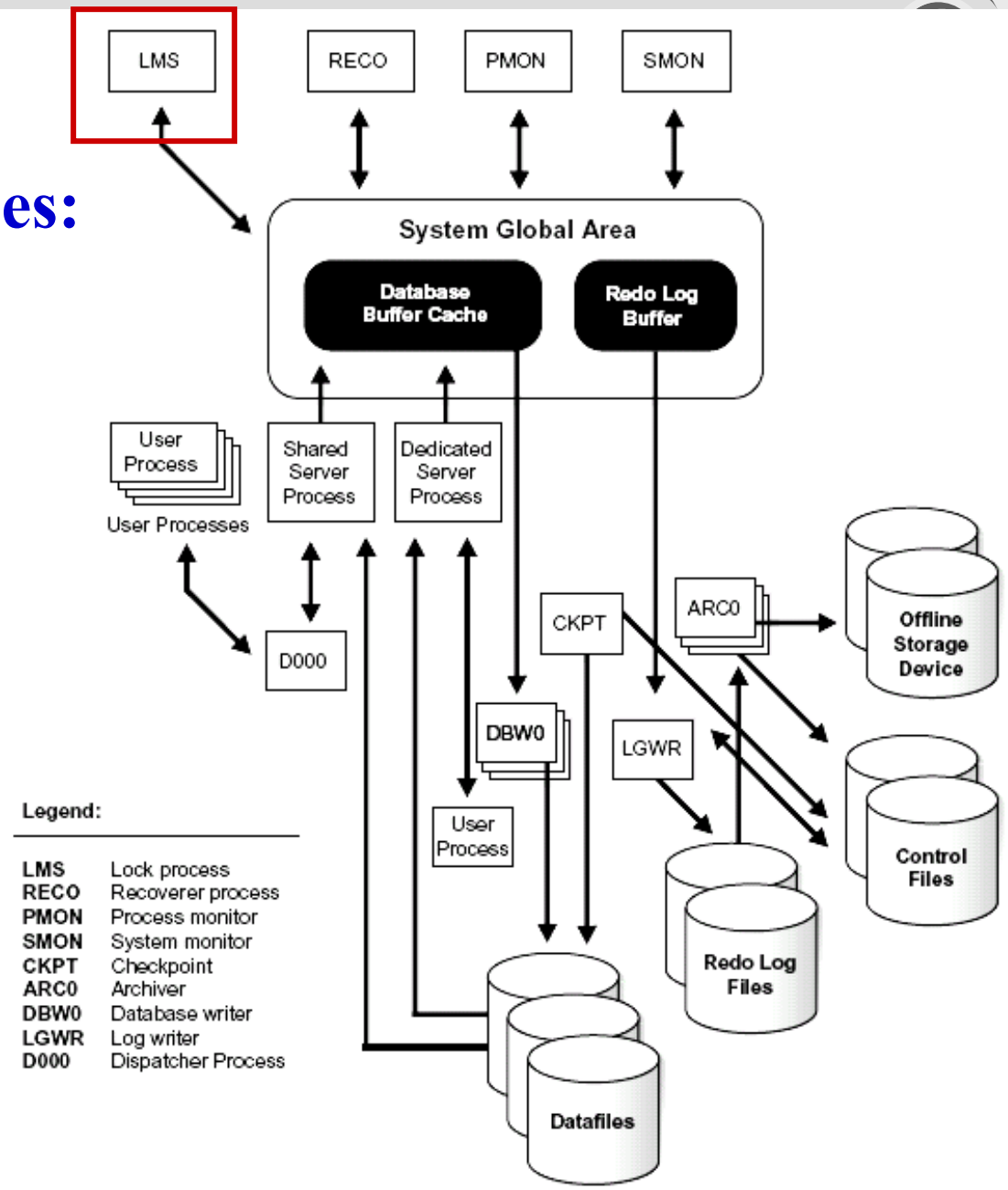
Background processes: Monitors

- Recoverer (RECO): 2PC για κατανεμημένες ΒΔ
- Process Monitor (PMON): process recovery & memory cleanup
- System Monitor (SMON): crash recovery & temporary segment cleaning



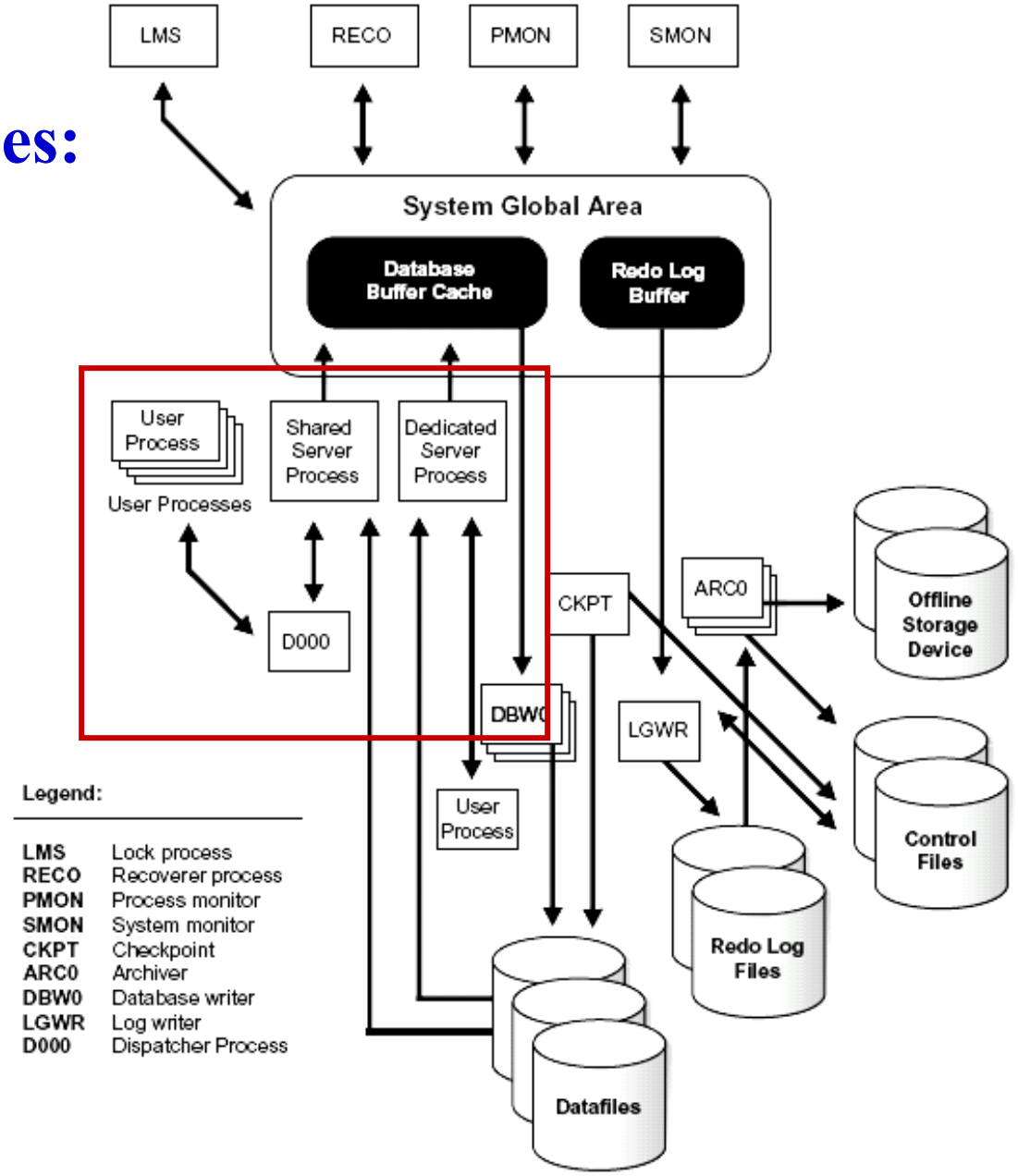
Background processes: Listener

- Η διαδικασία που καθιστά εφικτή τη σύνδεση ενός client στον server
- Ο Listener καθορίζει επίσης και τον τρόπο και τις λεπτομέρειες της σύνδεσης



Background processes: Server processes (ξανά)

- Δύο τρόποι:
  - Shared
  - Dedicated

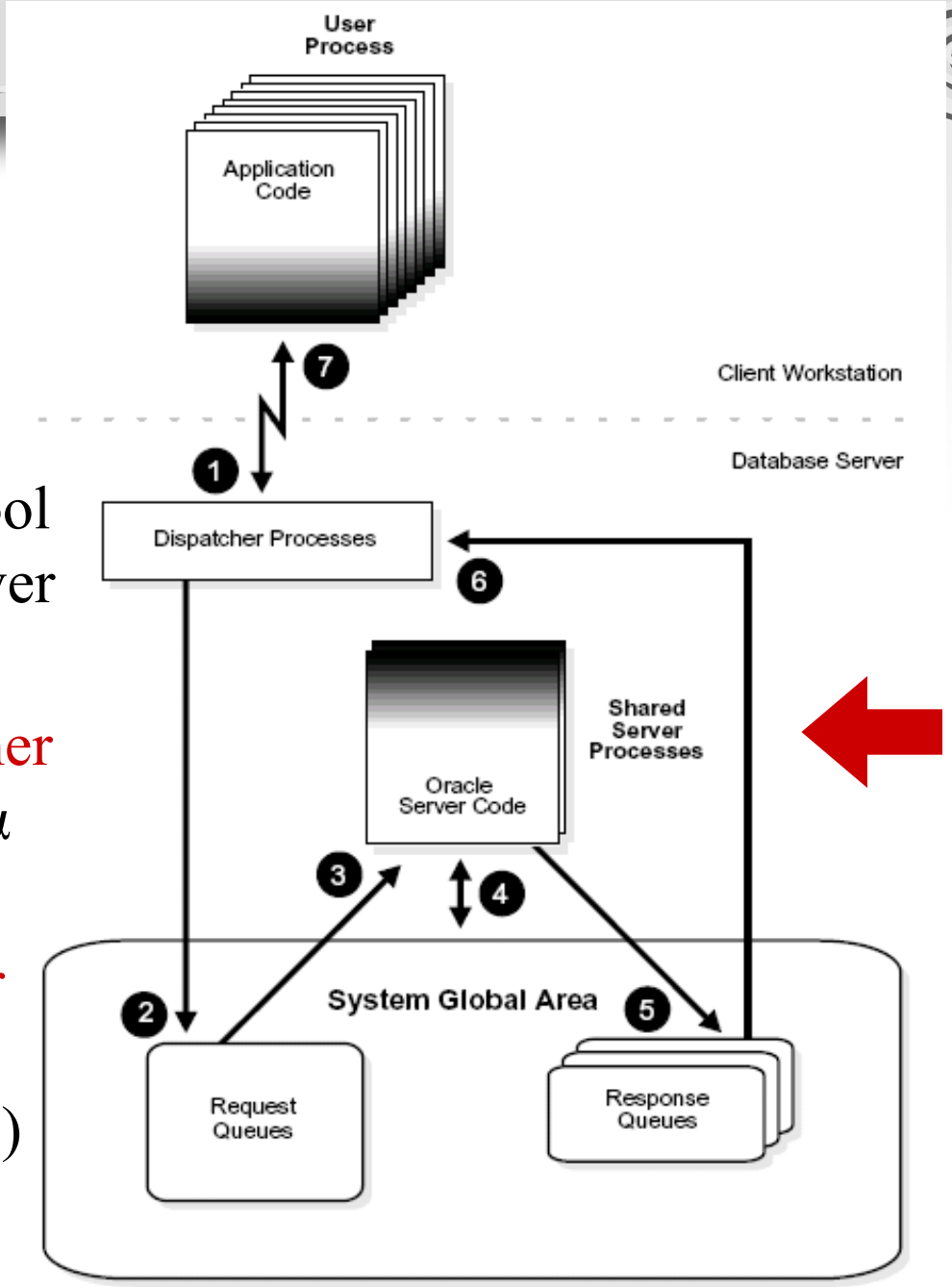


Shared Server Processes

(1) Οι **shared server processes** έχουν ένα pool από connections & server processes

(2) Κάθε φορά ο dispatcher βάζει μια αίτηση σε μια ουρά αιτήσεων

(3) Κάποια από τις server processes αναλαμβάνει να την εξυπηρετήσει (4)

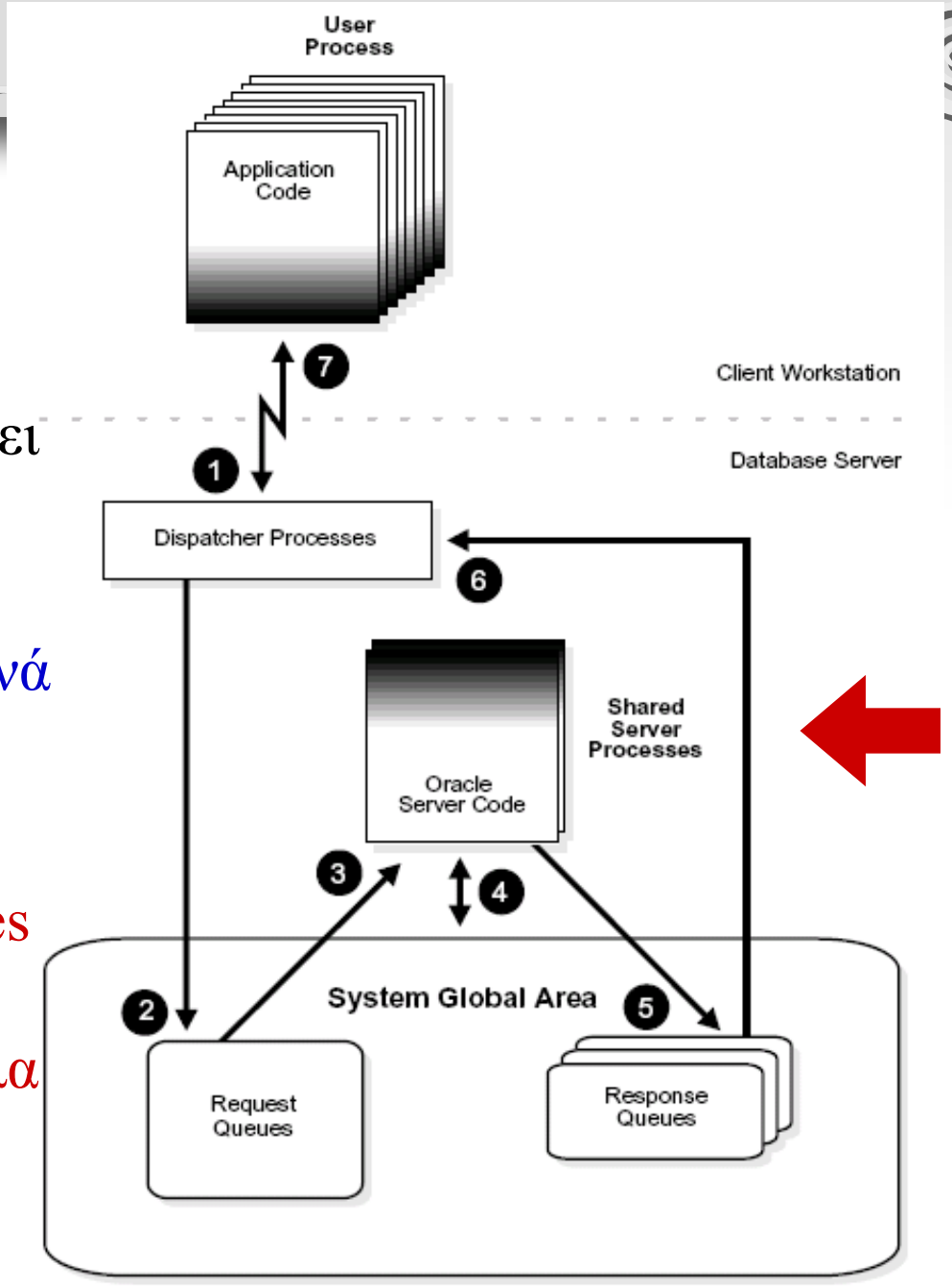


Shared Server Processes

(5) Το αποτέλεσμα μπαίνει σε μια ουρά αποτελεσμάτων

(6,7) Το αποτέλεσμα γυρνά πίσω στον client

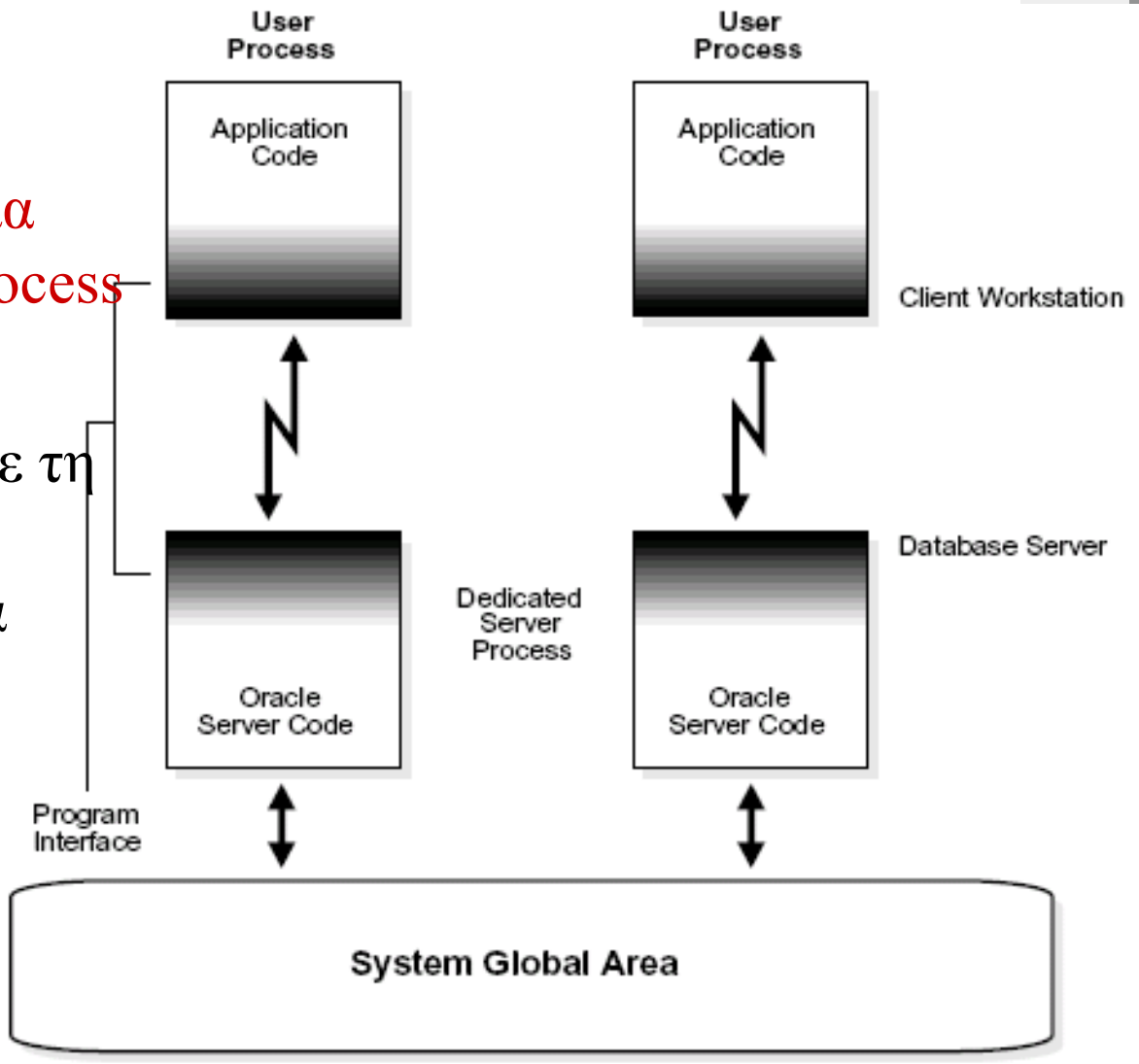
Οι shared server processes ουσιαστικά δεν έχουν τίποτε στην PGA => όλα τα δεδομένα τους είναι στην SGA



Dedicated Server Processes

Για κάθε client και μια dedicated server process

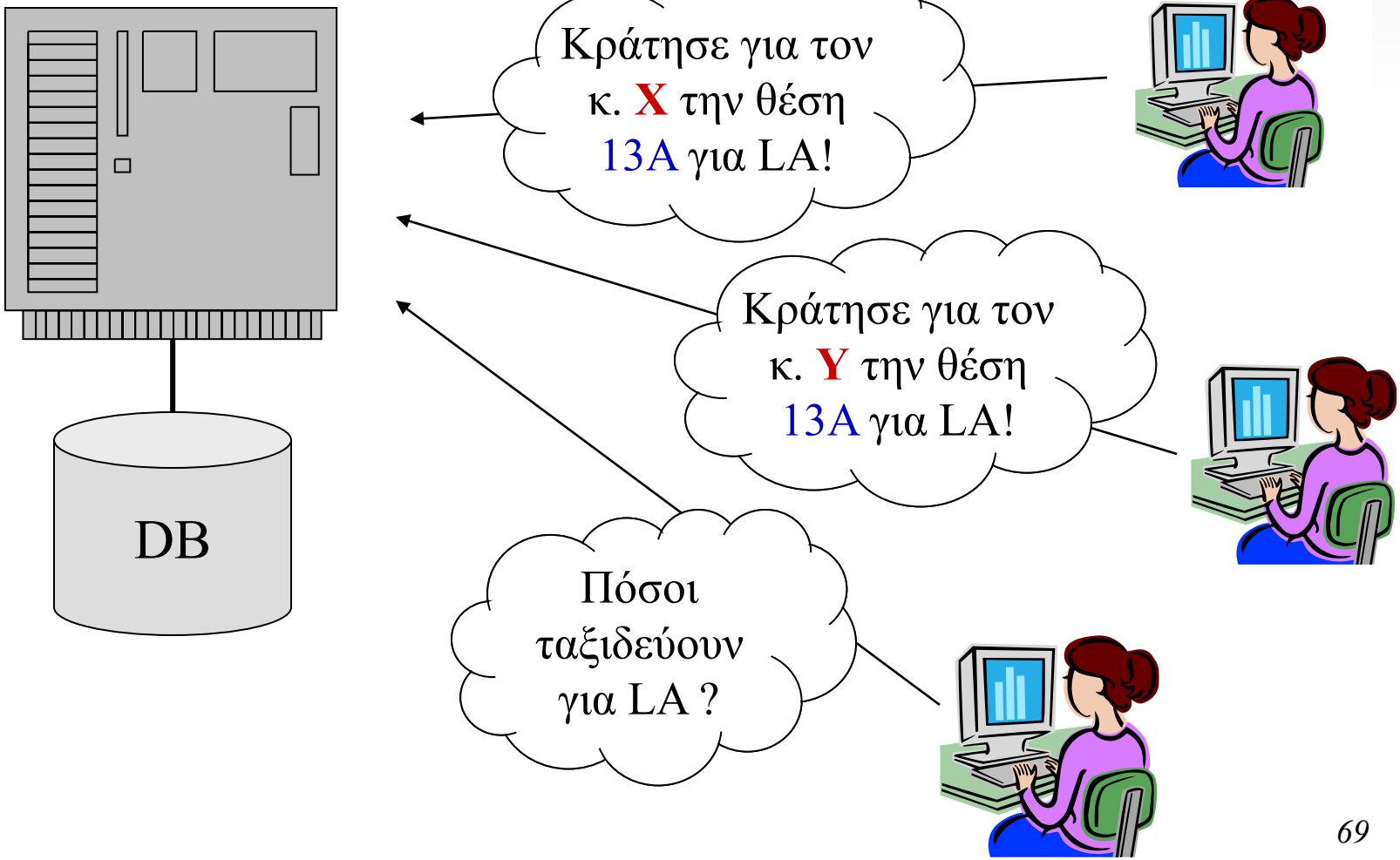
Πιο βαρύ σε σχέση με τη μνήμη που καταναλώνεται ανά σύνδεση



Θεματολόγιο

- Γενική αρχιτεκτονική βάσεων δεδομένων & ΣΔΒΔ
- Εσωτερική αποθήκευση δεδομένων
- Δομές στη μνήμη
- Διεργασίες
- Παράρτημα: Αξιοπιστία και διαθεσιμότητα βάσεων δεδομένων

Συναλλαγές & Ταυτοχρονισμός



Συναλλαγή

- **Συναλλαγή** είναι
  - μια σειρά από ενέργειες, οι οποίες
  - διαβάζουν ή γράφουν
  - αντικείμενα της βάσης
- στα αγγλικά “transaction”
- σειρά: διατεταγμένο σύνολο, λίστα

Για προχωρημένους: στην θεωρία [της πληροφορικής γενικά, και των ΒΔ ειδικά], οι λεξούλες τύπου «σειρά» έχουν σημασία ... (π.χ., σειρά ≠ σύνολο)

Παράδειγμα συναλλαγής

T_0: μεταφορά 50€ από το λογαριασμό Α στο λογαριασμό Β

```
read(A);
A := A - 50;
write(A);
read(B);
B := B + 50;
write(B).
```

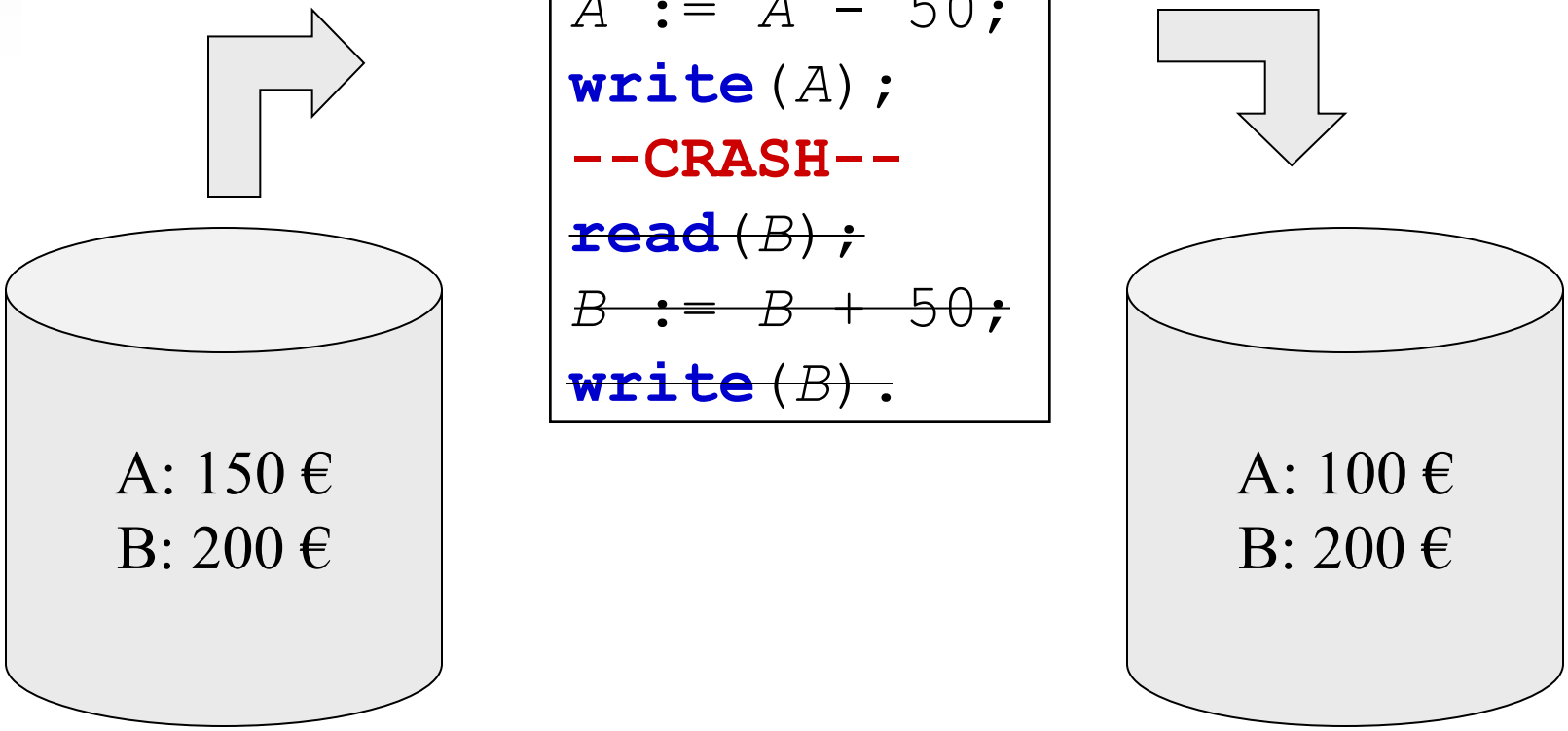
Προβληματισμός

- Δύο είναι τα βασικά προβλήματα με τις συναλλαγές:
  - Τι θα γίνει αν κατά τη διάρκεια της εκτέλεσης, πέσει το σύστημα?
  - Τι θα γίνει αν δύο συναλλαγές επιχειρούν να μεταβάλλουν το ίδιο αντικείμενο ταυτοχρόνως?

Ορολογία: Στο εξής το σύστημα δεν «πέφτει», αλλά «αποτυγχάνει» ☺

Αποτυχία

```
read(A);
A := A - 50;
write(A);
--CRASH--
read(B);
B := B + 50;
write(B).
```



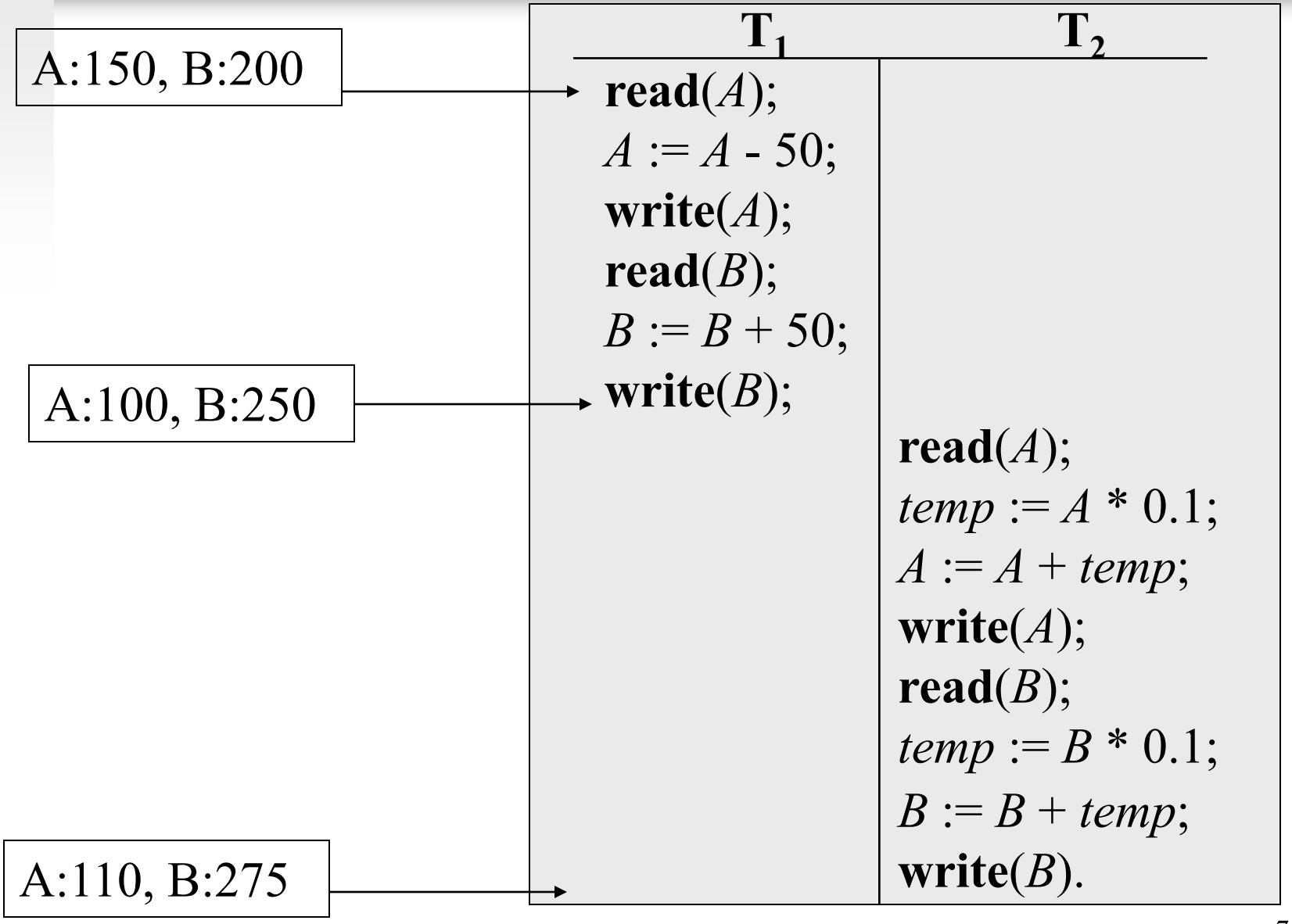
A+B=350 ⟷ A+B=300

Ταυτοχρονισμός

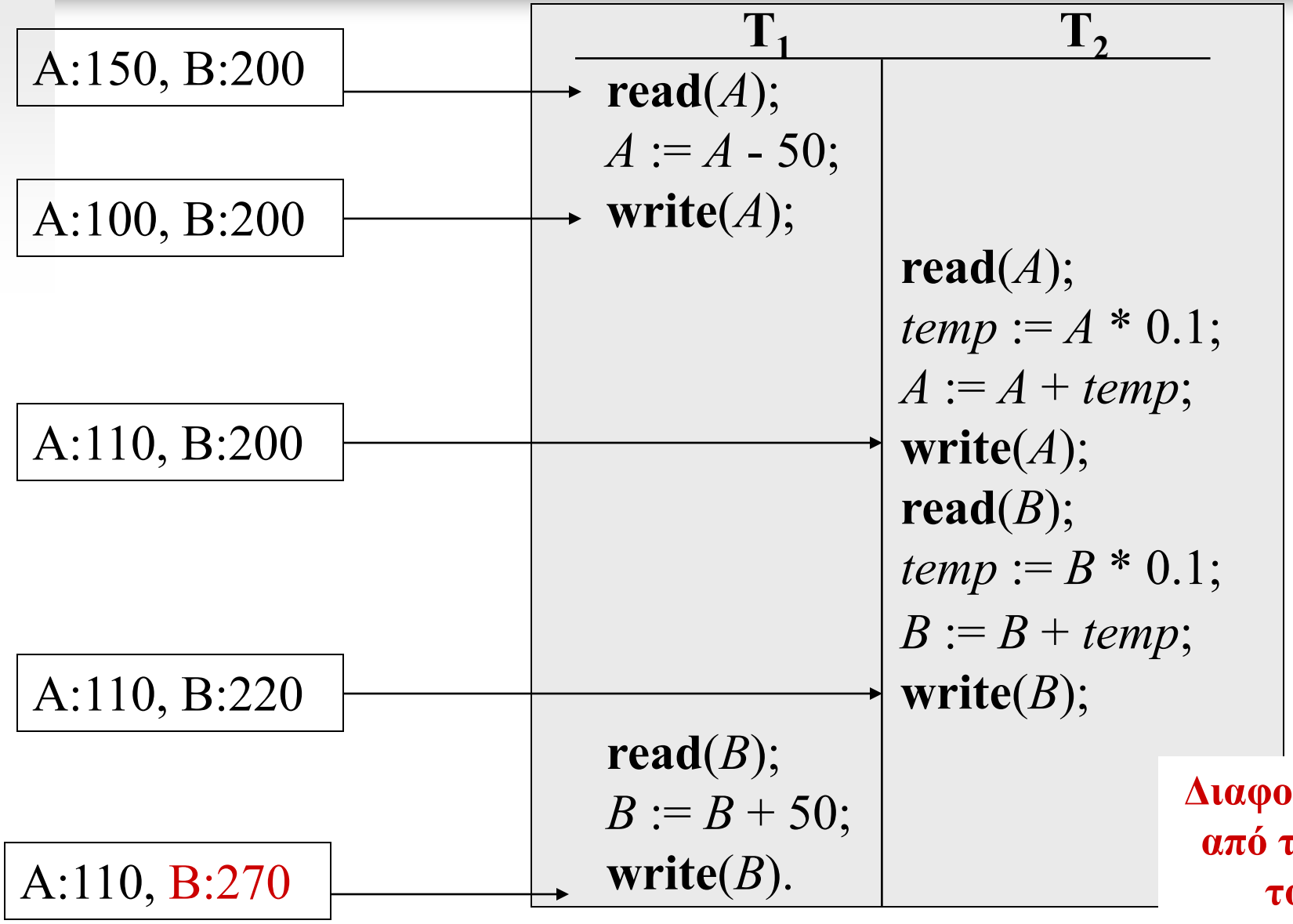
Έστω δύο συναλλαγές που τρέχουν σειριακά:

- T1: μεταφέρει 50 € από τον Α στον Β
- T2: κάνει αύξηση στον Α και το Β κατά 10%

Ταυτοχρονισμός σειριακά



Ταυτοχρονισμός με πολυπλεξία συναλλαγών



Διαφορετικό από το 275 του σειριακού!!!

Καταστάσεις μιας συναλλαγής

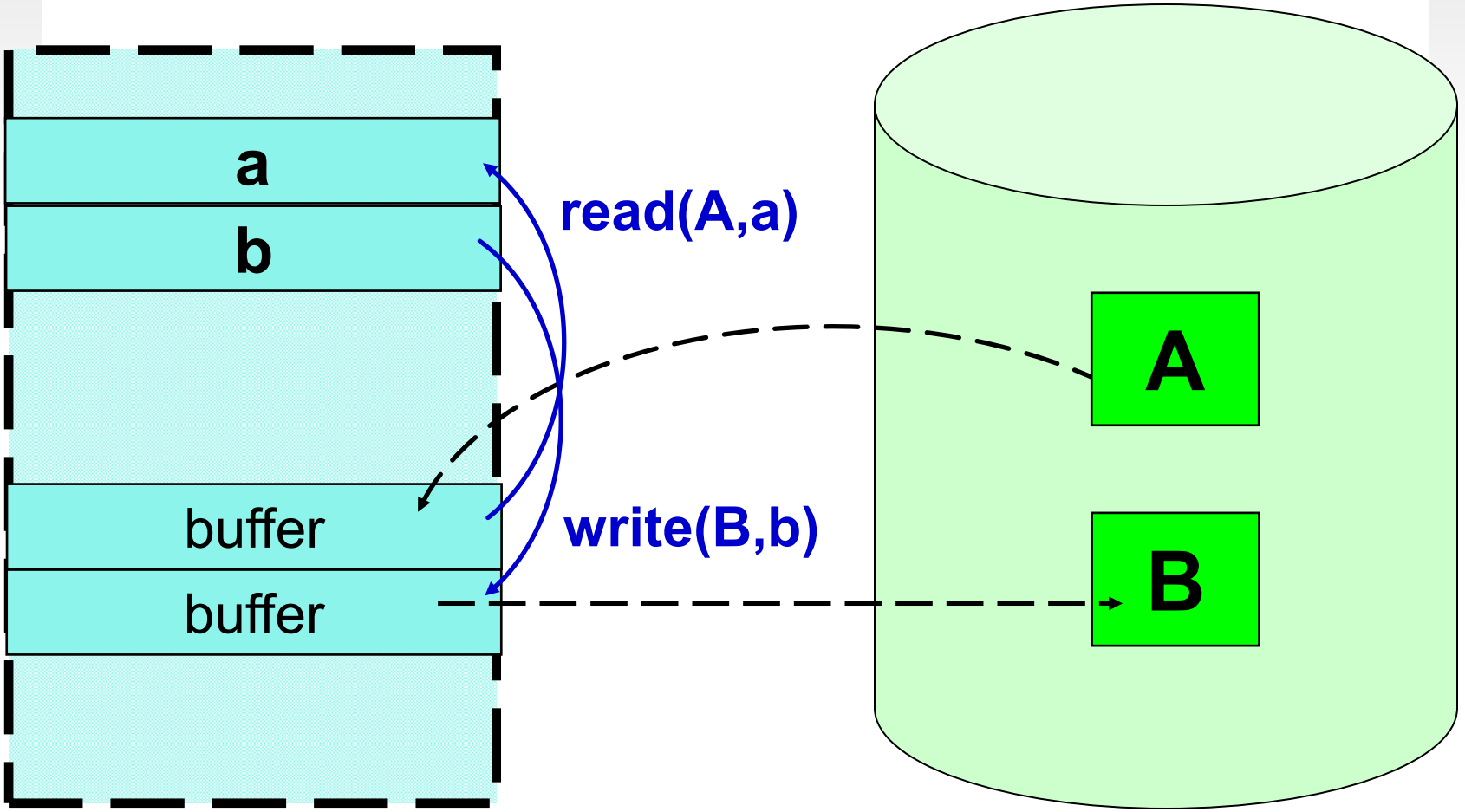
- **Active**: στο ξεκίνημα και κατά τη διάρκειά της
- **Failed**: όταν το DBMS αντιληφθεί ότι η συναλλαγή δεν μπορεί να συνεχίσει
- **Aborted**: όταν η αποτυχημένη συναλλαγή έχει αναιρεθεί από το σύστημα και η ΒΔ είναι σε συνεπή μορφή
- **Committed**: όταν η συναλλαγή επιτύχει και η ΒΔ είναι σε συνεπή μορφή.
- Συμβολισμός:
  - **COMMIT**$_X$: η συναλλαγή X τερματίζει επιτυχώς
  - **ABORT**$_x$: η συναλλαγή X αποτυγχάνει

Αστοχίες του συστήματος

- Κακό πρόγραμμα (με λάθη, δηλ.)
- Αστοχία συναλλαγής (αδιέξοδο, abort από τον χρήστη, κλπ)
- Αστοχία του συστήματος (πτώση ρεύματος, αδιέξοδο λειτουργικού συστήματος)
- Αστοχία υλικού (καταστροφή σκληρού δίσκου)

Failure: αστοχία ή αποτυχία

Μοντέλο Read & Write



Οι διακεκομμένες αναπαριστούν λειτουργίες που μπορεί να καθυστερήσουν/ αγνοηθούν

Σε περίπτωση αποτυχίας ...

- Σκοπός είναι να διατηρήσουμε τη συνέπεια του συστήματος.
- Πρέπει να επαναλάβουμε (**REDO**) όλες τις συναλλαγές που έκαναν commit
- Πρέπει να αναιρέσουμε (**UNDO**) όσες παρενέργειες επέφεραν οι συναλλαγές που δεν πρόλαβαν να κάνουν commit

Log (Ιστορικό)

- **Log** (Ιστορικό/Ιχνος/Ημερολόγιο): ένα αρχείο στο σκληρό δίσκο που καταγράφει όλη την ιστορία των ενεργειών που εκτελέστηκαν από το DBMS
- Ενέργειες:
  - BOT/EOT (Begin/End Of Transaction)
  - INS/UPD/DEL (ήτοι, write) ένα record
  - COMMIT/ABORT μια συναλλαγή
  - UNDO/REDO μια ενέργεια εγγραφής (write)

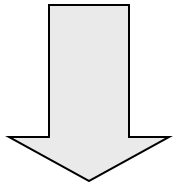
Log Record τύπου “Write”

- LSN
- Transaction ID (TID)
- Σελίδα που κάνουμε update
- Offset στην εν λόγω σελίδα
- Μήκος σε bytes που αλλάζουμε
- Παλιά τιμή
- Νέα τιμή

T1: UPDATE EMP

SET ID = 30

WHERE ID = 3



Log Entry: LSN12,T1,PAGE32,0xFFF32,8,3,30

Βασικές έννοιες για το Log

- Το log ως αρχείο είναι ένα σύνολο από εγγραφές (**log records**)
- Κάθε log record χαρακτηρίζεται μονοσήμαντα από ένα **Log Sequence Number** (**LSN**).
- Το σύστημα εκδίδει αυτόματα το max(LSN)+1 για κάθε νέα log record

Write Ahead Log

Προτού γράψεις οτιδήποτε στη ΒΔ, καταχώρησε την αντίστοιχη εγγραφή στο log

- Για να γράψεις μια updated σελίδα από το buffer πίσω στο δίσκο, πρέπει στο log (στο δίσκο) να έχουν περαστεί οι παλιές τιμές για τα records της
- Για να κάνεις commit μια συναλλαγή πρέπει στο log (στο δίσκο) να έχουν γραφτεί όλες οι σχετικές log records

Checkpoints – Σημεία ελέγχου

- Περιοδικά, το σύστημα κάνει τις εξής ενέργειες:
  - Σταματά κάθε άλλη ενέργεια
  - Καταγράφει το σύνολο των ενεργών συναλλαγών
  - Γράφει (**flush**) όλους τους buffers με log records, στο δίσκο
  - Γράφει (**flush**) όλους τους buffers με records της ΒΔ, στο δίσκο
  - Γράφει στο log μια εγγραφή CHK (checkpoint)

Ανάνηψη αν έχουμε WAL

- Έστω ότι το σύστημα αποτυγχάνει και πρέπει να το επαναφέρουμε (ήτοι, να επαναφέρουμε τη ΒΔ σε συνεπή μορφή).
- Η διαδικασία αυτή ονομάζεται **ανάνηψη** (**recovery**) ή ανάκαμψη ή επαναφορά
- Αν έχουμε χρησιμοποιήσει WAL κατά την κανονική λειτουργία του συστήματος, η ανάνηψη έχει **3 φάσεις:**
  1. Ανάλυση
  2. UNDO
  3. REDO