Φυσική Αποθήκευση

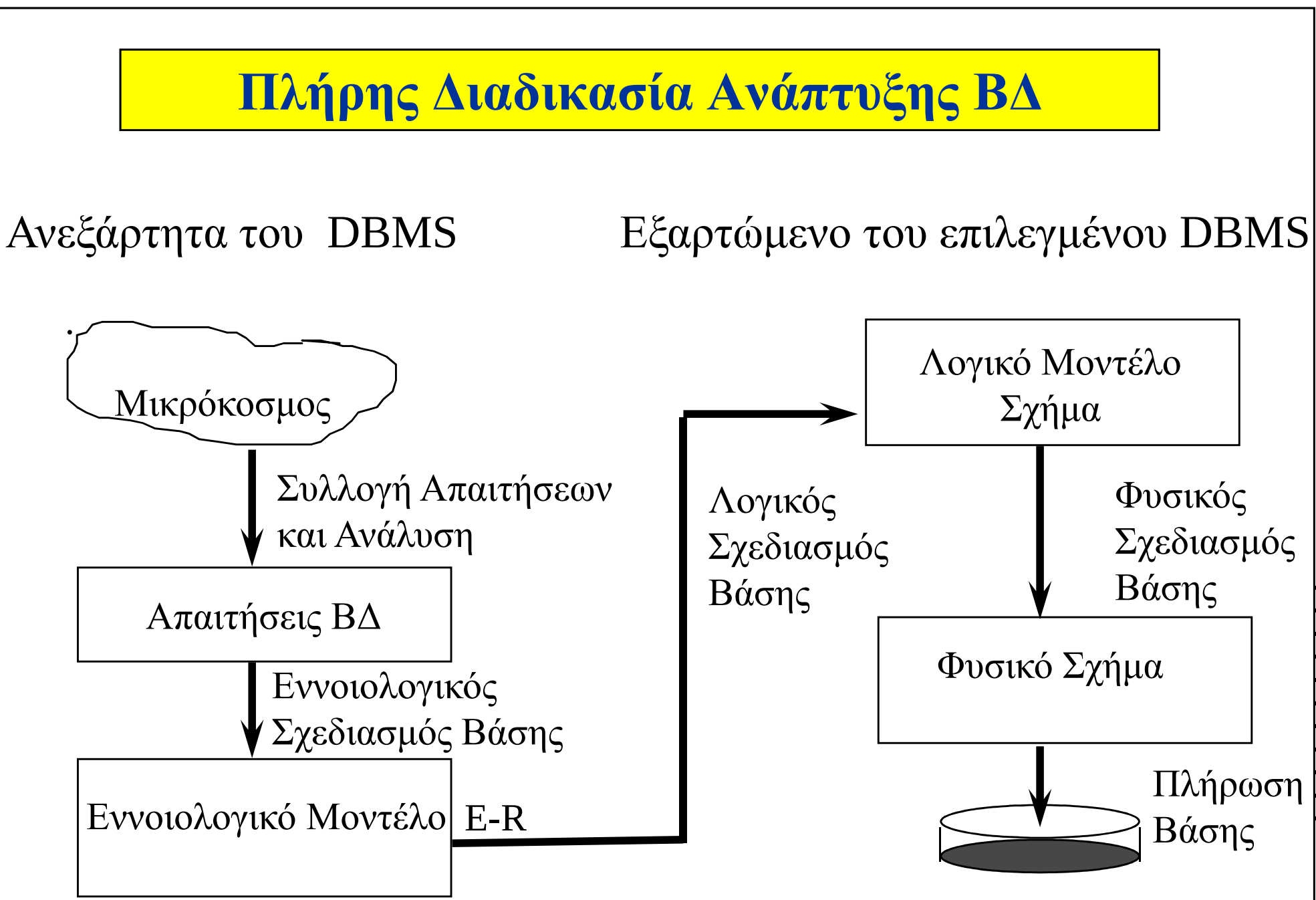
Οργανώσεις Αρχείων

Φυσικός Σχεδιασμός – Αποθήκευση Εγγραφών

ΣΥΝΟΨΗ ΕΝΟΤΗΤΑΣ

- Επισκόπηση των Μέσων Αποθήκευσης
- Μαγνητικοί Δίσκοι
- RAID – Συστοιχία Ανεξάρτητων Δίσκων
- Οργάνωση Αρχείων
- Οργάνωση Εγγραφών σε Αρχεία
- Πρόσβαση στη Μνήμη – Buffer Management
- Λεξικά Δεδομένων

Πλήρης Διαδικασία Ανάπτυξης ΒΔ



Φυσική Αποθήκευση

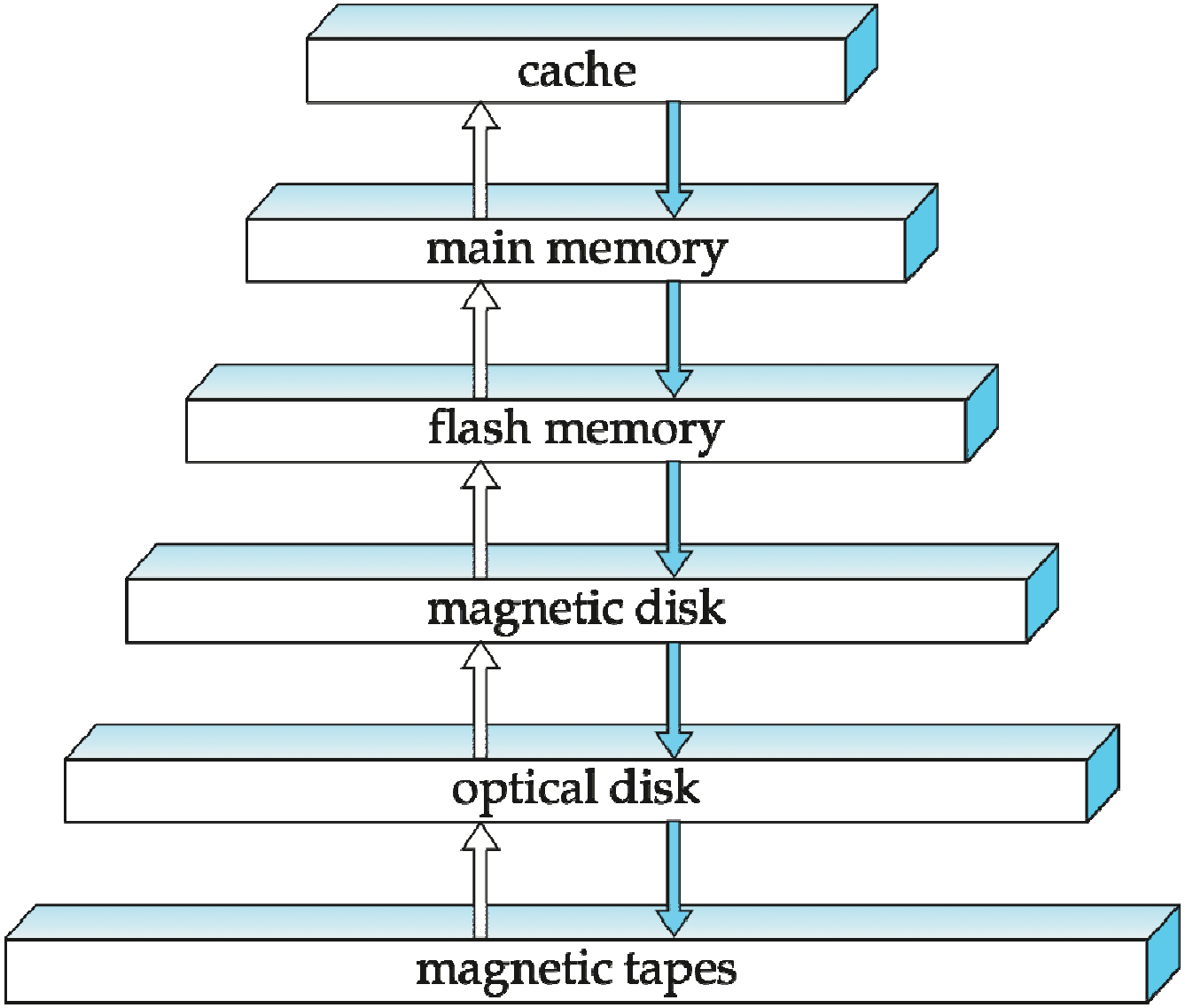
- Ο ***Διαχειριστής Δεδομένων (data manager)*** είναι το υπό-σύστημα του DBMS υπεύθυνο για τη ***φυσική βάση δεδομένων***
  Οι σημαντικές έννοιες είναι:
  - ***σύστημα αρχείων*** **(file system),**
  - ***διαχειριστής ενδιάμεσης μνήμης*** **(buffer manager),**
  - ***δομές ευρετηρίων*** **(access methods)**

- Κάθε DBMS έχει το δικό του ***Διαχειριστή Δεδομένων***, ο οποίος συχνά χρησιμοποιεί ένα κλασσικό σύστημα αρχείων όπως παρέχεται σε ένα Λειτουργικό Σύστημα ***ενισχυμένο*** με πρόσθετους μηχανισμούς

ΤΑΞΙΝΟΜΗΣΕΙΣ ΣΥΣΚΕΥΩΝ ΑΠΟΘΗΚΕΥΣΗΣ ΔΕΔΟΜΕΝΩΝ

- Ανάλογα με την **ταχύτητα** (Speed) πρόσβασης
- Ανάλογα με τη **χωρητικότητα**
- Ανάλογα με το **κόστος** πρόσβασης ανά μονάδα
- Ανάλογα με την **αξιοπιστία** (Reliability)
  - Πόσο συχνά χαλάει η συσκευή αποθήκευσης?
- Ανάλογα με τη **μονιμότητα**
  - Χάνονται τα δεδομένα όταν πέφτει το ρεύμα ή το σύστημα (crash)?
  - **volatile storage:** χάνεται η μνήμη όταν σβήνει ο ΗΥ
  - **non-volatile storage**: Παραμένουν τα δεδομένα.

Ιεραρχία Μνήμης



Συσκευές Αποθήκευσης (α)

- **Cache** – Η ταχύτερη, πιο μικρή και πιο ακριβή (ΔΕΝ ΜΑΣ ΑΠΑΧΟΛΕΙ ΣΤΙΣ ΒΔ)
- **Κύρια Μνήμη (Main memory)**:
  - Γρήγορη Πρόσβαση (10s έως 100s των nanoseconds; 1 nanosecond = 10^{-9} seconds)
  - Γενικά πολύ μικρή ή πολύ ακριβή για ΟΛΗ τη ΒΔ
    - » Φτάνει μέχρι κάποιες δεκάδες Gigabytes σήμερα
    - » Κάθε χρόνο, η χωρητικότητα αυξάνει και το κόστος χαμηλώνει (περίπου δύο (2) φορές κάθε 2 με 3 χρόνια)
  - **Volatile**

Συσκευές Αποθήκευσης (β)

- **Flash memory** (EEPROM - Electrically Erasable Programmable Read-Only Memory)
  - Δε χάνεται όταν πέφτει το ρεύμα (non-volatile)
  - Περιορισμοί στις εγγραφές δεδομένων (σχετικά μικρός αριθμός write/erase cycles)
  - Σχεδόν το ίδιο γρήγορα reads με την κύρια μνήμη
  - Λίγο πιο αργά writes
  - Χωρητικότητες μέχρι 256 GB
  - Πολύ υψηλό κόστος/GB μνήμης
  - Memory cards, usb flash drives, solid state drives (SSD)

Συσκευές Αποθήκευσης (γ)

- **Μαγνητικός Δίσκος (Magnetic-disk)**
  - Τα δεδομένα σε περιστρεφόμενο δίσκο και τα read/write γίνονται με μαγνητικά μέσα
  - Ο ΤΥΠΙΚΟΣ ΤΡΟΠΟΣ ΑΠΟΘΗΚΕΥΣΗΣ ΔΕΔΟΜΕΝΩΝ ΣΕ Συστήματα Διαχείρισης ΒΔ
  - Τα δεδομένα πρέπει να μεταφερθούν από τον δίσκο στην κύρια μνήμη για επεξεργασία και μετά να γραφτούν πάλι στον δίσκο
  - Πολύ πιο αργή από την Κύρια Μνήμη
  - Σημερινή χωρητικότητα έως 10TB
    - » 2-3X αύξηση κάθε 2 χρόνια
  - Τα δεδομένα σώζονται μετά από διακοπές λειτουργίας
    - » Σφάλματα του δίσκου μπορεί να καταστρέψουν δεδομένα - σπάνιο

Συσκευές Αποθήκευσης (δ)

- **Οπτική Μνήμη (Optical storage)**
  - Σαν το μαγνητικό δίσκο, αλλά τα READ/WRITE γίνονται με ΟΠΤΙΚΟ τρόπο
  - CD-ROM (700 MB) και DVD (4.7 to 17 GB)
  - Blu-ray disks: 27 GB to 54 GB
  - Write-one, read-many (WORM) optical disks χρησιμοποιούνται για μόνιμη αποθήκευση (CD-R and DVD-R)
  - Υπάρχουν και εκδόσεις για πολλαπλά WRITE (CD-RW, DVD-RW, και DVD-RAM)
  - Πιο αργά από Μαγνητικούς Δίσκους
  - **Juke-box** συστήματα, με μεγάλο αριθμό δίσκων και ένα μηχανισμό για αυτόματη φόρτωση / εκφόρτωση των δίσκων (για πολύ μεγάλες ΒΔ)

Συσκευές Αποθήκευσης (ε)

- **Ταινία Αποθήκευσης (Tape storage)**
  - Βασικά για backup
  - **sequential-access** – πολύ αργή
  - Μεγάλη χωρητικότητα (40 με 300 GB)
  - Ακριβές Συσκευές για READ / WRITE
  - Tape jukeboxes
    - » Για εκατοντάδες terabytes (1 terabyte = 10^9 bytes) μέχρι και για petabyte (**1 petabyte = 10^{12} bytes**)

Γιατί δεν αποθηκεύονται τα πάντα σε Κύρια Μνήμη ?

- *Κοστίζει ακριβά.* $300 αγοράζουν 128MB RAM **ή** 7.5GB δίσκο.
- *Η κύρια μνήμη είναι ευμετάβλητη / ασταθής .*

Θέλουμε να σώζουμε τα δεδομένα μεταξύ χρήσεων. (Προφανώς!)

- Τυπική Ιεραρχία:
  - Κύρια Μνήμη (RAM) για δεδομένα επίκαιρης χρήσης.
  - Δίσκοι για την Βάση Δεδομένων (δευτερεύουσα μνήμη).
  - Ταινίες για την αποθήκευση παλαιοτέρων εκδόσεων της Βάσης Δεδομένων (μαζική αποθήκευση).
- Ένα DBMS έχει την παρακάτω **Ιεραρχία Μνήμης:**

  Ταινία $\rightarrow$ ***Δίσκος*** $\rightarrow$ ***Κύρια Μνήμη*** $\rightarrow$ ***Cache***

  (σειριακή) (άμεση)
  - *Οι ταινίες είναι για* **μαζική αποθήκευση**, *οι ταινίες για την* ***μόνιμη*** (**persistent**) αποθήκευση ΒΔ, ενώ η κύρια μνήμη και η *cache* για επεξεργασία των **δοσοληψιών και άλλων** DBMS πράξεων

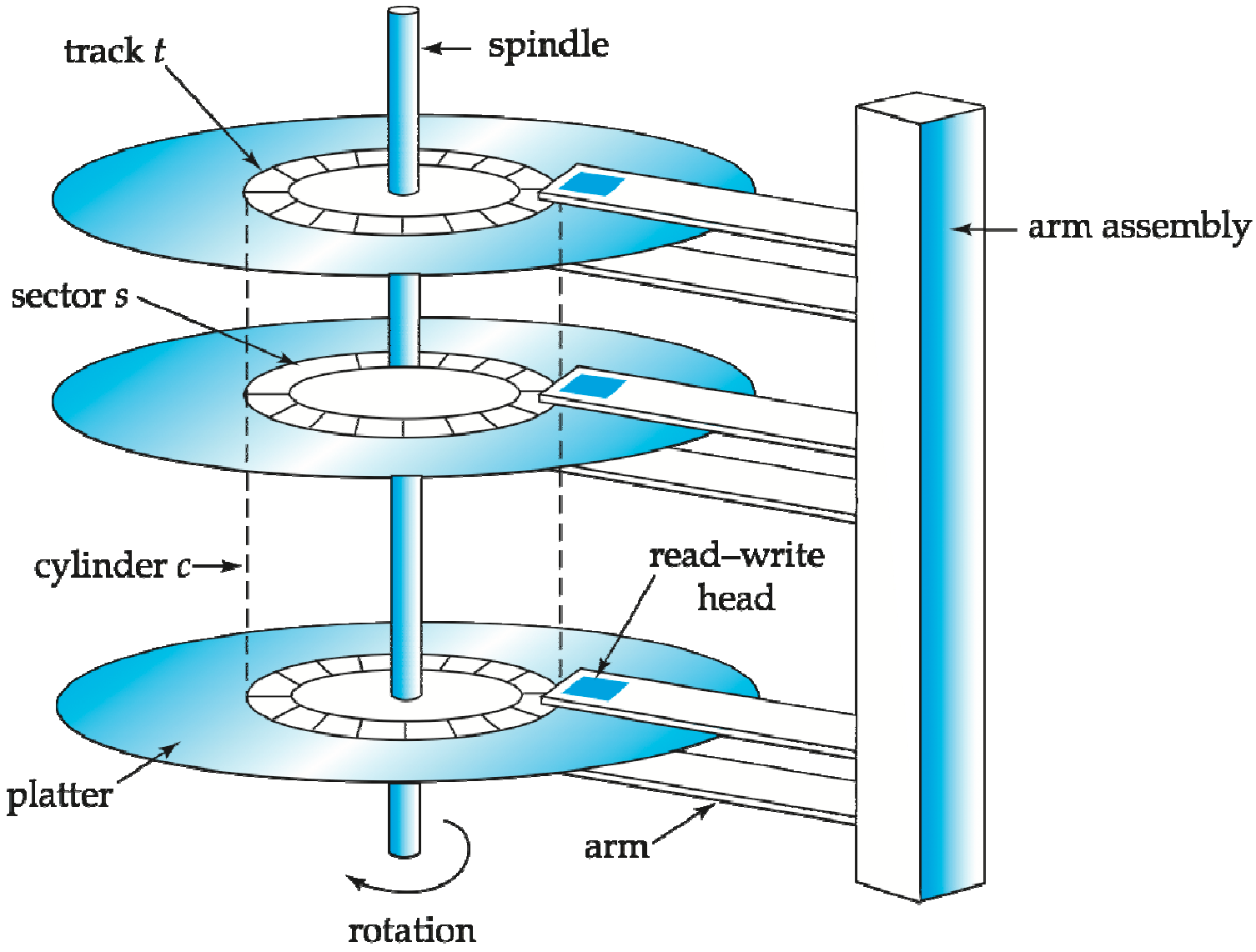
Δίσκοι και Αρχεία

- Το DBMS αποθηκεύει πληροφορίες σε (σκληρούς) δίσκους
- Αυτό έχει σημαντικές επιπτώσεις για το Σχεδιασμό των DBMS!
- Διακρίνονται 2 πολύ σημαντικές πράξεις
  - **READ**: μεταφέρει δεδομένα από το Δίσκο στην Κύρια Μνήμη (RAM).
  - **WRITE**: μεταφέρει δεδομένα από τη RAM στο Δίσκο.
  - Αυτές οι δύο πράξεις είναι υψηλού κόστους (χρονικά), σε σχέση με πράξεις που γίνονται εντός της Κύριας Μνήμης , **άρα θα πρέπει να μελετώνται και να σχεδιάζονται πολύ προσεκτικά!**

Μαγνητικοί Δίσκοι (1)

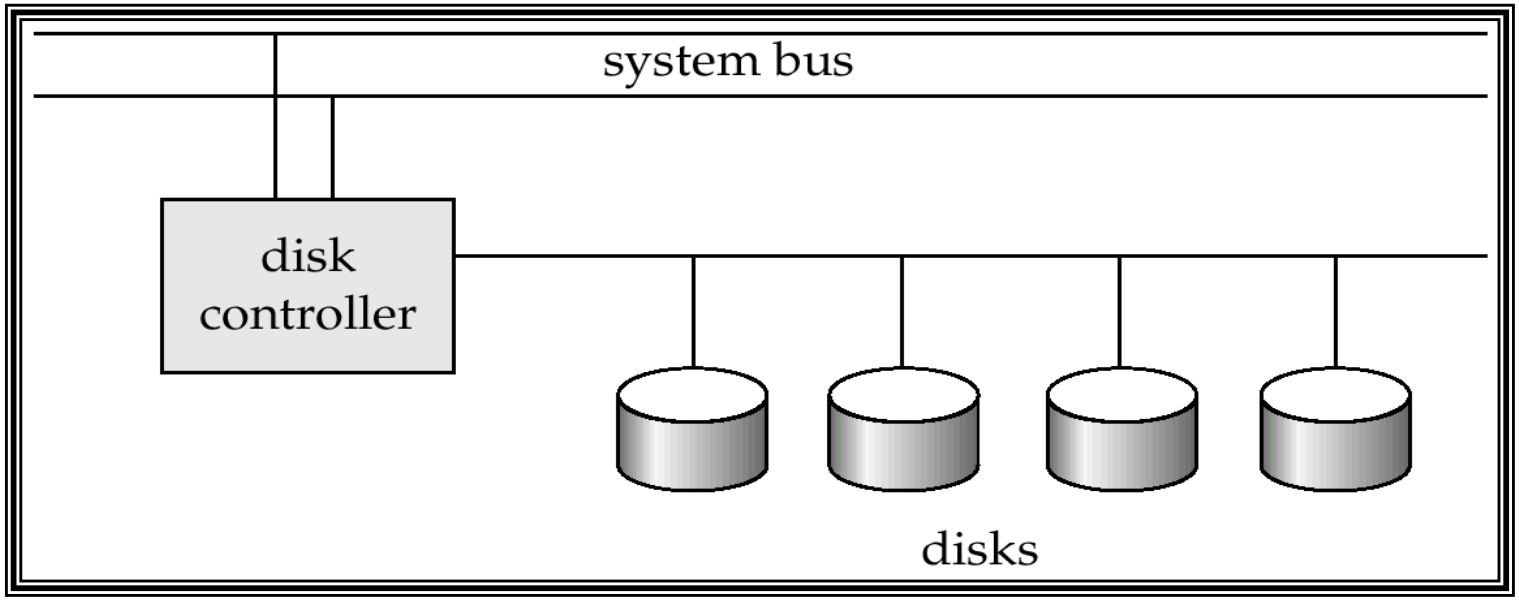
- Τα δεδομένα αποθηκεύονται ως ***μαγνητικές περιοχές*** σε μαγνητικούς δίσκους
- Ένας HDD έχει πολλούς δίσκους συνδεδεμένους σε ένα κύλινδρο (1-5)
- Οι δίσκοι χωρίζονται σε ομόκεντρους κύκλους, που ονομάζονται αυλάκια (***tracks),*** μέχρι 100000 σε κάθε δίσκο
- Τα αυλάκια χωρίζονται σε ***τομείς*** (***sectors)***, μεγέθους 512 bytes
- Πλήρεις τομείς μεταφέρονται μεταξύ μνήμης και δίσκου
- Για read/write ενός sector
  - » Ο βραχίονας κινείται για να βρει η κεφαλή το σωστό track
  - » Οι δίσκοι περιστρέφονται συνεχώς και τα δεδομένα γράφονται/διαβάζονται όταν το sector περάσει κάτω από την κεφαλή
- **Reads και writes έχουν κόστος** λόγω των καθυστερήσεων που εισάγουν τα *seek time (τοποθέτηση βραχίονα)* και *rotational latency (εύρεση τομέα*)
- Ένα **ΜΠΛΟΚ / ΣΕΛΙΔΑ** (**Block / Page**) είναι μια συνεχής σειρά από τομείς (στο ίδιο αυλάκι) που για πρακτικούς λόγους αποτελούν την «ιδανικότερη» μονάδα μεταφοράς μεταξύ Κυρίας Μνήμης και Δίσκου. Το μέγεθος κυμαίνεται από 512 Byte έως μερικά Kbyte (τυπικά μεγέθη 4096 ή 8192 ή 16384 bytes)
- Μια φυσική διεύθυνση στο Δίσκο αποτελείται από: ***αριθμό επιφανείας***, ***αριθμό ατράκτου*** (στην ίδια επιφάνεια) & ***αριθμό block*** (στην ίδια άτρακτο)

Τμήματα Δίσκου



Μαγνητικοί Δίσκοι - Ελεγκτές

- Πολλοί δίσκοι συνδέονται σε ένα υπολογιστικό σύστημα μέσω ελεγκτή
- Δέχεται υψηλού επιπέδου εντολές για read/write ενός τομέα
- Τις εκτελεί κινώντας τον βραχίονα
- Υπολογίζει και κρατά **checksums** για κάθε sector ώστε να ελέγχει αν τα δεδομένα διαβάστηκαν σωστά
- Τυποποιήσεις για Συστήματα Δίσκων
  - ATA, SATA, SCSI και πολλές άλλες παραλλαγές



Disk Subsystems

- Disks usually connected directly to computer system
- In **Storage Area Networks (SAN)**, a large number of disks are connected by a high-speed network to a number of servers
- In **Network Attached Storage (NAS)** networked storage provides a file system interface using networked file system protocol, instead of providing a disk system interface

Μετρικές απόδοσης

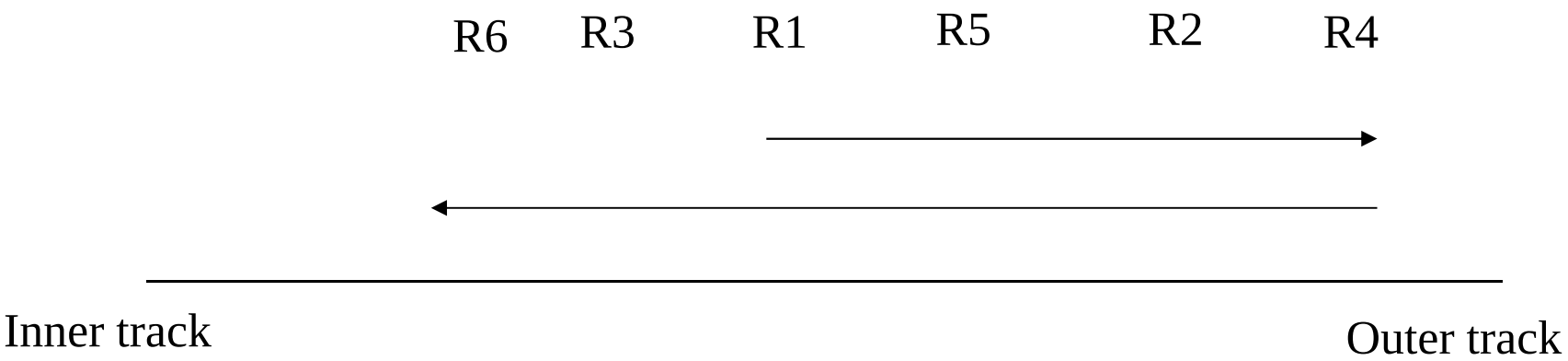
- Χρόνος Προσπέλασης (ανάγνωση / εγγραφή read/write) :
  - *Χρόνος αναζήτησης (seek time)* - κίνηση κεφαλής (track)
  - *Καθυστέρηση Περιστροφής (rotational latency)* - αναμονή για να φτάσει το μπλοκ κάτω από τη κεφαλή
  - *Χρόνος Μεταφοράς (block transfer time)* – *η* ουσιαστική κίνηση δεδομένων από / προς το Δίσκο)
- Seek time και rotational delay είναι οι μεγαλύτερες καθυστερήσεις.
  - Seek time μεταξύ 4 και 10msec
  - Rotational delay μεταξύ 4 και 11msec
  - Transfer rate περίπου 1msec για μια σελίδα 4KB
  - Το κλειδί για μικρότερες καθυστερήσεις είναι: **μικρότερες seek / rotation delays!**
  - Χρησιμοποιούνται λύσεις Υλικού ή / και Λογισμικού για να επιτευχθεί αυτό

Μετρικές απόδοσης

- **Mean time to failure (MTTF)** – Ο μέσος χρόνος που αναμένεται ο Δίσκος να λειτουργεί συνεχώς χωρίς πρόβλημα
- Τυπικά, 3 με 5 χρόνια
  - Η πιθανότητα αστοχίας ενός νέου δίσκου είναι πολύ μικρή και αντιστοιχεί σε ένα «θεωρητικό» MTTF των 30,000 με 1,200,000 ώρες για ένα καινούργιο Δίσκο
    - » Το MTTF των 1,200,000 ωρών για ένα νέο δίσκο σημαίνει ότι για κάθε 1000 νέους δίσκους, ένας από αυτούς θα αστοχήσει σε 1,200,000 ώρες
  - MTTF ελαττώνεται καθώς ο Δίσκος γηράσκει

Optimization of Disk-Block Access

- **Block** – συνεχόμενη σειρά τομέων στο ίδιο track
  - Τα δεδομένα μεταφέρονται από τον δίσκο στη μνήμη σε blocks
  - Μέγεθος 512 bytes - several kilobytes
    - » Smaller blocks: more transfers from disk
    - » Larger blocks: more space wasted due to partially filled blocks
    - » Typical block sizes today range from 4 to 16 kilobytes
- **Disk-arm-scheduling** algorithms order pending accesses to tracks so that disk arm movement is minimized
  - **elevator algorithm**:



Βελτιστοποίηση για Block Access

- Στην **Οργάνωση Αρχείου (File organization),** που θα εξετάσουμε αργότερα, βελτιστοποιούμε το χρόνο πρόσβασης με την κατάλληλη οργάνωση των blocks ώστε να αντιστοιχεί με το **πως** θα γίνει η πρόσβαση
  - Π.Χ., Αποθήκευε σχετιζόμενες πληροφορίες στον ίδιο ή σε κοντινό κύλινδρο
  - Files may get **fragmented** over time
    - » E.g. if data is inserted to/deleted from the file
    - » Or free blocks on disk are scattered, and newly created file has its blocks scattered over the disk
    - » Sequential access to a fragmented file results in increased disk arm movement
  - Some systems have utilities to defragment the file system, in order to speed up file access

RAID

- **RAID: Redundant Arrays of Independent Disks**
  - Οργανωτικές τεχνικές διάταξης δίσκων που δίνουν την αίσθηση / όψη ενός και μόνο δίσκου με
    - » Μεγάλη χωρητικότητα και υψηλή ταχύτητα (πολλοί δίσκοι σε Παράλληλη Χρήση)
    - » Μεγάλη αξιοπιστία (αποθήκευση των δεδομένων με επαναληπτικότητα έτσι ώστε να γίνεται εύκολα η ανάκαμψη των δεδομένων)
- **Η πιθανότητα να αστοχήσει ένας δίσκος σε ένα σύνολο πολλών δίσκων είναι πολύ μεγαλύτερη από το να αστοχήσει ένας συγκεκριμένος.**
  - Π.Χ, σε ένα σύστημα με 100 δίσκους, αν ο κάθε ένας έχει MTTF των 100,000 ωρών (περίπου 11 ετών), το σύστημα θα έχει MTTF των 1000 ωρών (περίπου 41 ημέρες)
- **Αλλαγή Ονομασίας**
  - Αρχικά, το I στο RAID διαβαζόταν «inexpensive»
  - Σήμερα το I διαβάζεται «independent»

Μεγαλύτερη αξιοπιστία με επαναληπτικότητα

- **Redundancy** (Πλεονασμός) store extra information that can be used to rebuild information lost in a disk failure

- **Mirroring** (**shadowing**)
  - Ένας λογικός Δίσκος αποτελείται από 2 φυσικούς Δίσκους.
  - Κάθε write γίνεται και στους 2 Δίσκους
    - » Reads μπορεί να γίνουν από οποιονδήποτε
  - Αν ο ένας δίσκος αστοχεί, τα δεδομένα είναι ακόμη διαθέσιμα
    - » Μικρή πιθανότητα να χαλάσουν και οι δύο Δίσκοι ταυτόχρονα

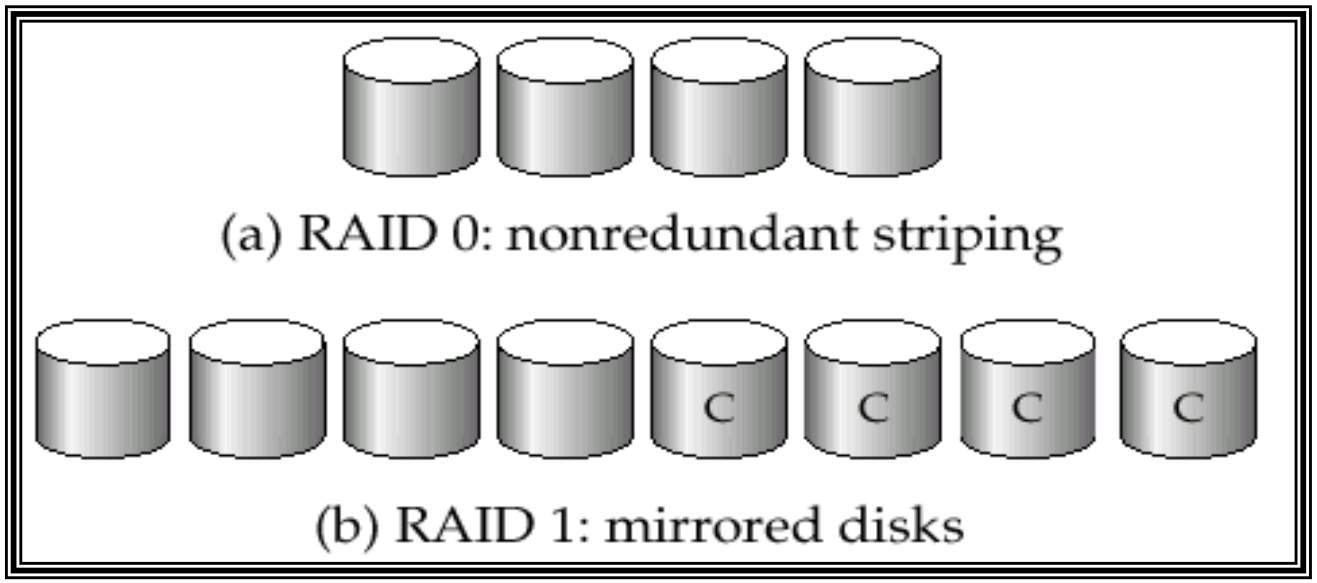
Μεγαλύτερη απόδοση με Παραλληλισμό

- Δύο βασικοί στόχοι του Παραλληλισμού σε ένα σύστημα δίσκων:
  1. Κατανομή φόρτου για πολλές μικρές αιτήσεις έτσι ώστε να αυξηθεί ο ρυθμός μεταφοράς (throughput)
  2. Παραλληλισμός μεγάλων αιτήσεων για να μειωθεί ο χρόνος απόκρισης (response time)
- Τα δεδομένα διαχωρίζονται (striping data) σε πολλαπλούς δίσκους.
  - **Bit-level striping** – Διαχωρισμός των bits του κάθε byte σε πολλαπλούς δίσκους (δε χρησιμοποιείται συχνά)
  - **Block-level striping** – με *n* Δίσκους, το block *i* του αρχείου πηγαίνει στο Δίσκο (*i* mod *n*) + 1

ΕΠΙΠΕΔΑ RAID

■**RAID Level 0**: Block striping; non-redundant.

- Όταν μας νοιάζει η ταχύτητα και δεν ενδιαφερόμαστε αν χάσουμε δεδομένα.



(a) RAID 0: nonredundant striping

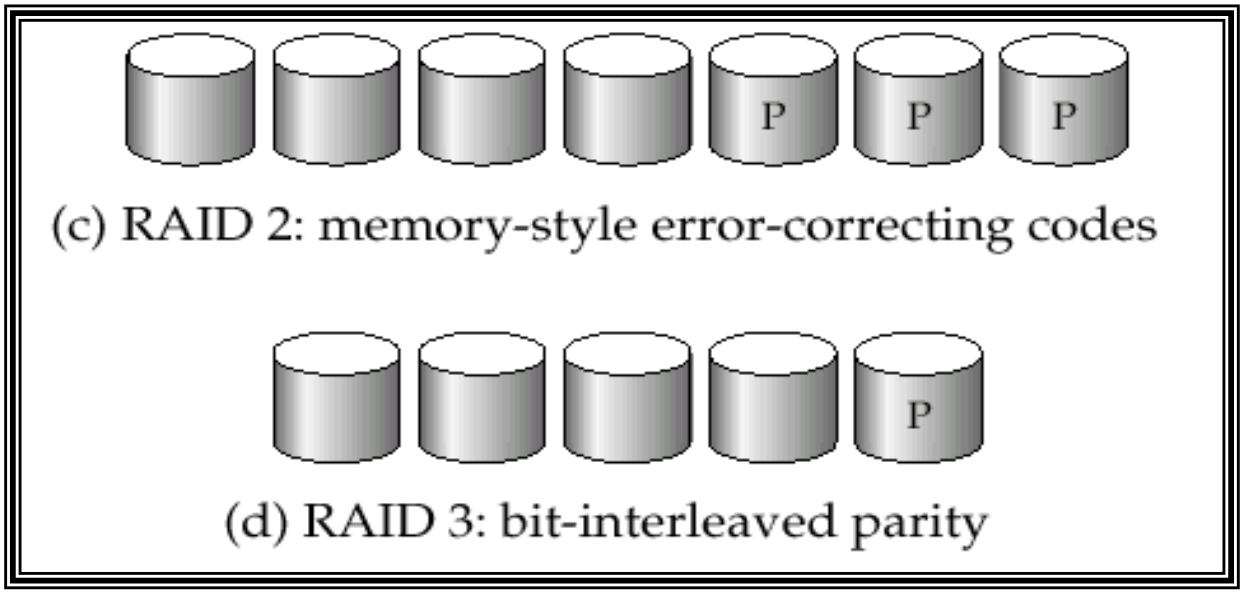
(b) RAID 1: mirrored disks

■**RAID Level 1**: Mirrored disks with block striping

- Η καλύτερη οργάνωση για LOG αρχεία και για ταχύτητα στα WRITE

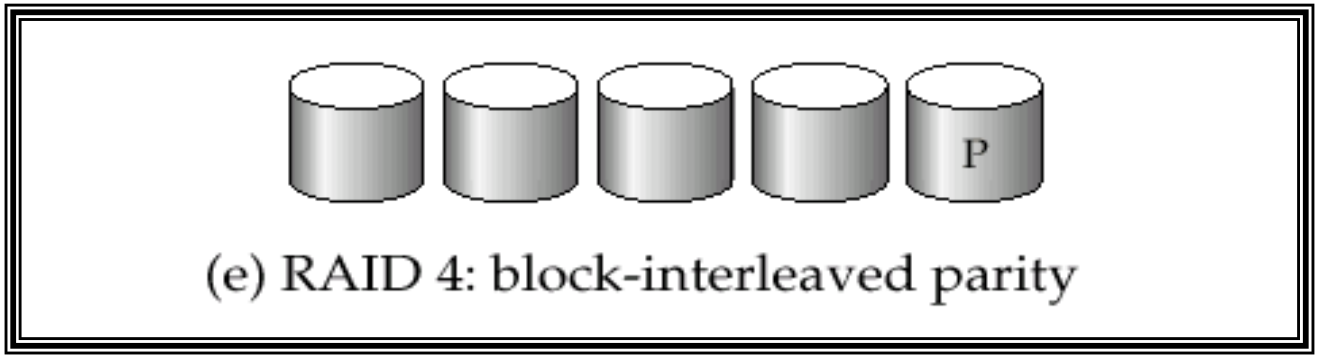
ΕΠΙΠΕΔΑ RAID

- **RAID Level 2**: Memory-Style Error-Correcting-Codes (bit striping)
- **RAID Level 3**: Bit-Interleaved Parity
  - Καλό για έλεγχο λαθών σε περίπτωση αστοχίας
  - Κάνει ότι κάνει και το Level 2, αλλά με χαμηλότερο κόστος



ΕΠΙΠΕΔΑ RAID

- **RAID Level 4:** Block-Interleaved Parity;
  - Χρησιμοποιεί block-level striping, και κρατά parity block σε ξεχωριστό δίσκο
  - Σαφώς καλύτερο από το Level 3 (σε I/O και ρυθμό μεταφοράς)
  - Μερικά προβλήματα με bottlenecks (κάθε write πρέπει να υπολογίσει parity block)



(e) RAID 4: block-interleaved parity

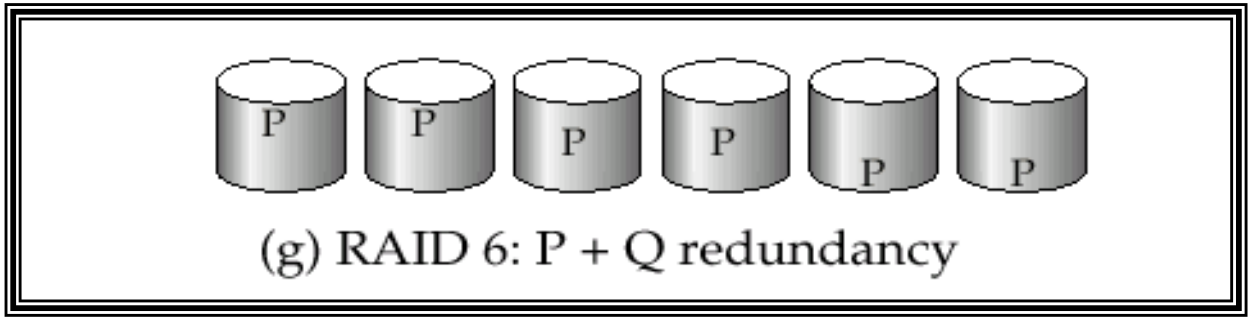
ΕΠΙΠΕΔΑ RAID

- **RAID Level 5:** Block-Interleaved Distributed Parity
  - partitions data and parity among all *N* + 1 disks, rather than storing data in *N* disks and parity in 1 disk.
  - Η καλύτερη διάταξη για ΒΔ
  - Higher I/O rates than Level 4.
    - » Block writes occur in parallel if the blocks and their parity blocks are on different disks.
  - Subsumes Level 4: provides same benefits, but avoids bottleneck of parity disk.

(f) RAID 5: block-interleaved distributed parity

ΕΠΙΠΕΔΑ RAID

- **RAID Level 6**: P+Q Redundancy scheme
  - Αποθηκεύει extra πληροφορία για πολλαπλές αστοχίες δίσκου
  - Καλύτερη αξιοπιστία από το προηγούμενο αλλά με μεγάλο επιπλέον κόστος (σπάνια χρησιμοποιείται)



(g) RAID 6: P + Q redundancy

Επιλογή του RAID Level

- Σήμερα γίνεται μεταξύ του 1 και 5 επιπέδου μόνο (τα άλλα είναι σαφώς υποδεέστερα ή υπερ-καλύπτονται από τα 1 και 5)
- Level 1 καλύτερο για WRITEs
- Level 1 χειρότερο σε κόστος αποθήκευσης

- Το Level 5 προτιμάται για εφαρμογές με μικρό αριθμό ενημερώσεων και μεγάλο αριθμό δεδομένων
- Το Level 1 προτιμάται για όλες τις άλλες εφαρμογές

Οργάνωση αρχείων

- Η βάση αποθηκεύεται ως συλλογή από αρχεία (*files*). Κάθε file είναι μια σειρά από εγγραφές (*records*). Μια εγγραφή είναι μια σειρά από πεδία
- Μια προσέγγιση:
  - –Σταθερό μήκος εγγραφών
  - –Κάθε αρχείο έχει εγγραφές ενός συγκεκριμένου τύπου μόνο
  - –Διαφορετικά αρχεία για διαφορετικές σχέσεις

  Εύκολο στην υλοποίηση

Εγγραφές σταθερού μήκους

- Απλή προσέγγιση:
  - Κάθε εγγραφή *i* ξεκινά από το byte $n * (i - 1)$, όπου *n* το μήκος της κάθε εγγραφής
  - Έύκολη προσπέλαση αλλά η τελευταία εγγραφή μπορεί να μη χωράει ολόκληρη
    - » Αλλαγή: μην επιτρέπεις σε εγγραφές να μοιράζονται σε 2 blocks

- Διαγραφή εγγραφής *i*: εναλλακτικές:
  - Μετακίνηση εγγραφών $i + 1, \ldots, n$ σε $i, \ldots, n - 1$
  - Μετακίνηση εγγραφής n σε *i*
  - Μη μετακινείς εγγραφές αλλά κράτα links σε όλα τα free records (*free list*)

record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

Deleting record 3 and compacting

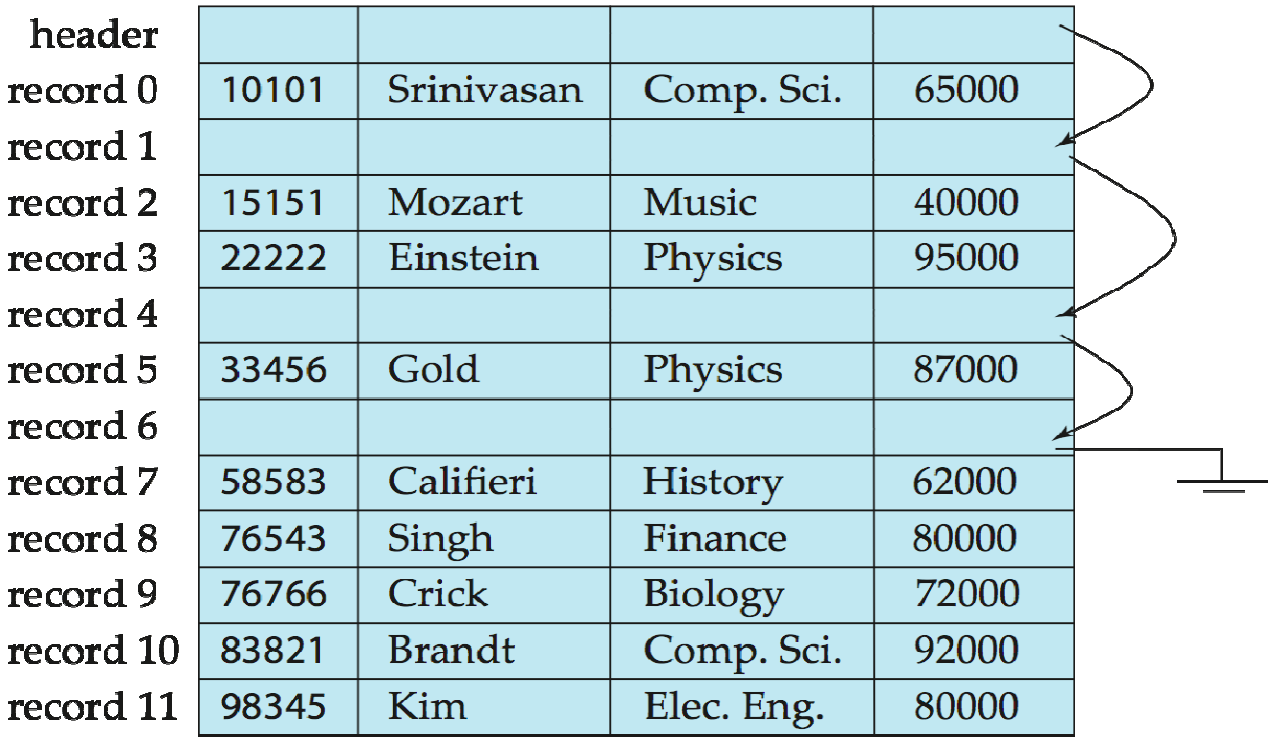
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

Deleting record 3 and moving last record

record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 11	98345	Kim	Elec. Eng.	80000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000

Free Lists

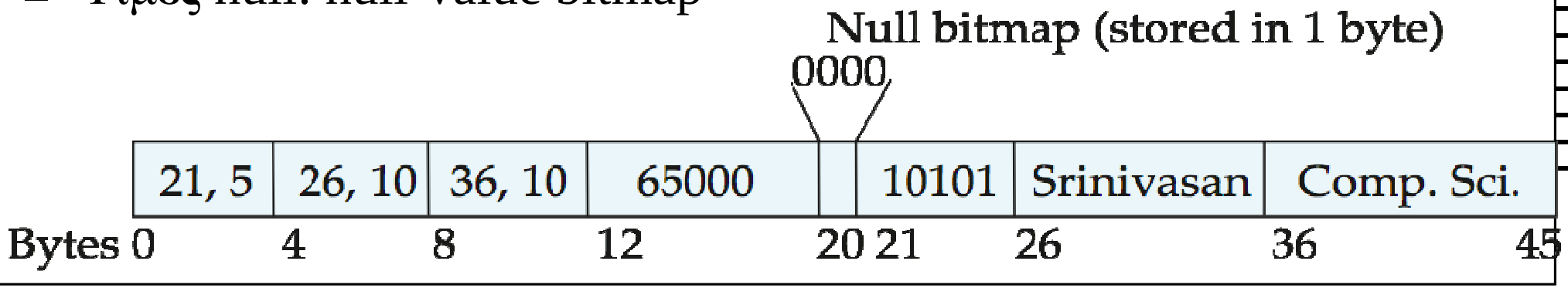
- Διεύθυνση της πρώτης σβησμένης εγγραφής σε header.
- Διεύθυνση της δεύτερης σβησμένης εγγραφής στη θέση της πρώτης, κλπ.
- Διευθύνεις = pointers που δείχνουν στη θέση μιας εγγραφής
- Αποδοτικό σε χώρο: reuse space for normal attributes of free records to store pointers



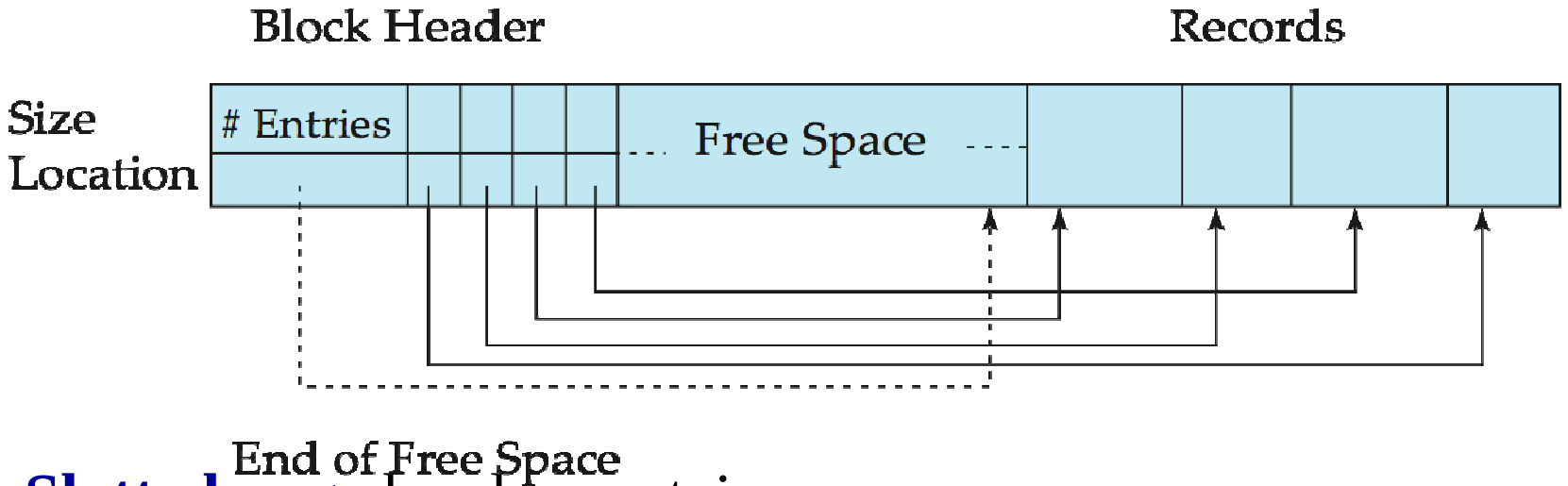
header				
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1				
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4				
record 5	33456	Gold	Physics	87000
record 6				
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

Εγγραφές μεταβλητού μήκους

- Εγγραφές μεταβλητού μήκους προκύπτουν στις βάσεις
  - Αποθήκευση διάφορων τύπων εγγραφών στο ίδιο αρχείο
  - Τύποι που επιτρέπουν μεταβλητό μήκος (π.χ. **varchar**)
- Τα γνωρίσματα αποθηκεύονται με σειρά
- Πρώτα τα σταθερού μήκου
- Τα γνωρίσματα μεταβλητού μήκους αναπαριστώνται από πληροφορία σταθερού μήκους (offset, length)
- Οι πραγματικές τιμές αποθηκεύονται μετά τις σταθερού μήκους
- Τιμές null: null-value bitmap



Δομή σελίδας με χρήση θέσεων



- **Slotted page** header contains:
  - Αριθμό εγγραφών
  - Τέλος του ελεύθερου χώρου
  - Θέση και μέγεθος κάθε εγγραφής
- Οι εγγραφές μετακινούνται μέσα στο block για να μη μένει κενός χώρος μεταξύ τους – απαιτείται ενημέρωση του header
- Οι δείκτες δεν δείχνουν απευθείας στις εγγραφές αλλά στο αντίστοιχο entry του header

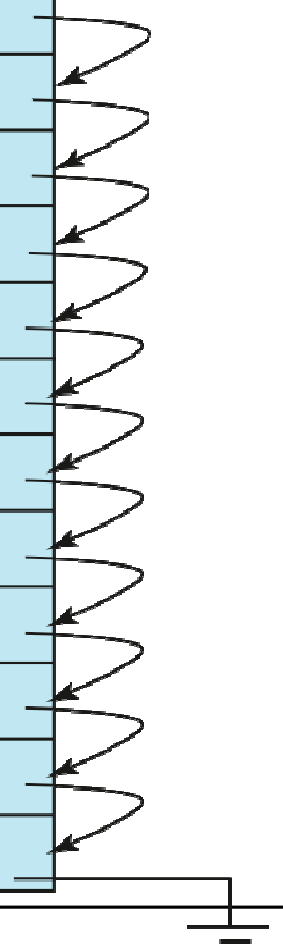
Οργάνωση εγγραφών σε αρχεία

- **Σε σωρό - Heap** – μια εγγραφή τοποθετείται οπουδήποτε υπάρχει χώρος στο αρχείο
- **Σειριακή - Sequential** – αποθηκεύει τις εγγραφές σειριακά με βάση την τιμή ενός κλειδιού αναζήτησης
- **Hashing** – υπολογίζεται μια συνάρτηση hash σε κάποιο attribute κάθε εγγραφής. Το αποτέλεσμα καθορίζει σε ποιο block θα τοποθετηθεί η εγγραφή
- Εγγραφές διαφορετικών σχέσεων γράφονται σε διαφορετικά αρχεία. Σε **multitable clustering file organization** αποθηκεύονται εγγραφές διαφορετικών σχέσεων στο ίδιο αρχείο
  - Κίνητρο: σχετιζόμενες εγγραφές στο ίδιο block για ελαχιστοποίηση I/O

Σειριακή οργάνωση αρχείου

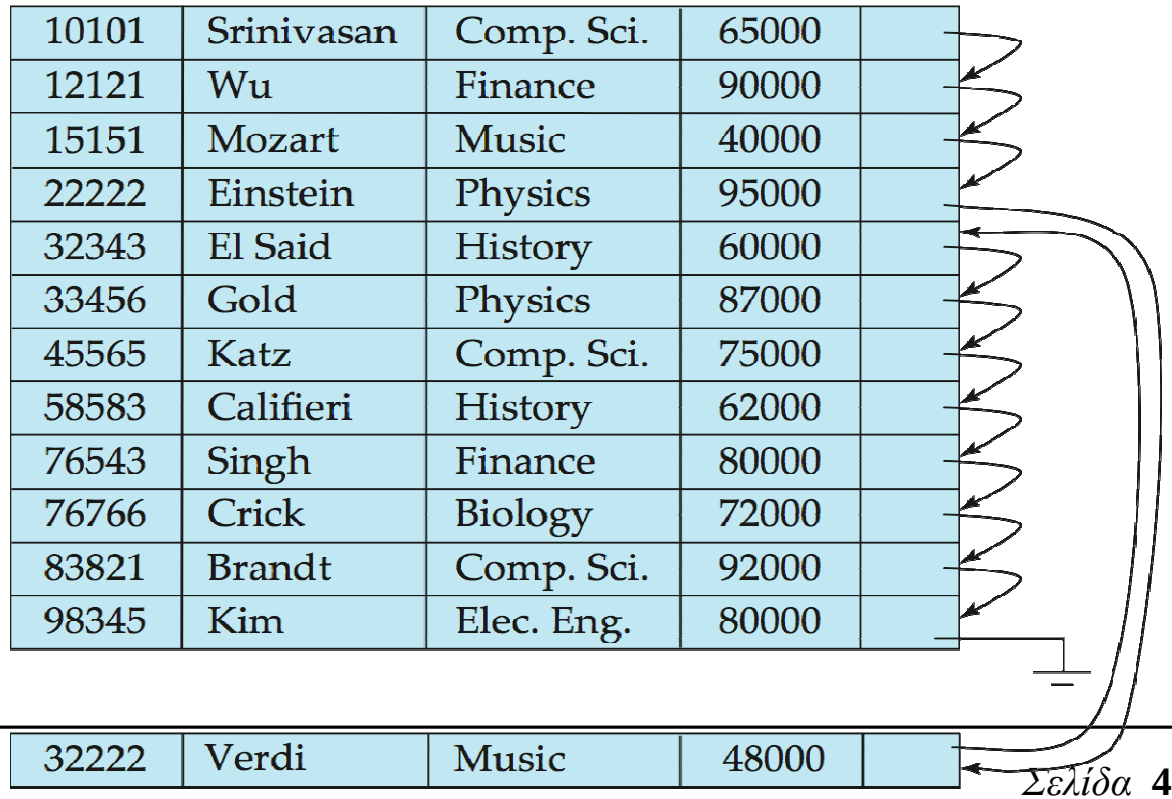
- Κατάλληλο για εφαρμογές που απαιτούν σειριακή επεξεργασία του αρχείου
- Οι εγγραφές ταξινομούνται με βάση κάποιο κλειδί αναζήτησης

10101	Srinivasan	Comp. Sci.	65000	
12121	Wu	Finance	90000	
15151	Mozart	Music	40000	
22222	Einstein	Physics	95000	
32343	El Said	History	60000	
33456	Gold	Physics	87000	
45565	Katz	Comp. Sci.	75000	
58583	Califieri	History	62000	
76543	Singh	Finance	80000	
76766	Crick	Biology	72000	
83821	Brandt	Comp. Sci.	92000	
98345	Kim	Elec. Eng.	80000	



Σειριακή οργάνωση αρχείου

- Διαγραφή – χρήση αλυσίδων δεικτών
- Εισαγωγή – βρες τη θέση που πρέπει να εισαχθεί η εγγραφή
  - Αν υπάρχει ελεύθερος χώρος τότε γράψε
  - Αν δεν υπάρχει γράψε σε overflow block
  - Σε κάθε περίπτωση, ανανέωσε pointer
- Χρειάζεται αναδιοργάνωση για να διατηρούμε τη σειρά



10101	Srinivasan	Comp. Sci.	65000	
12121	Wu	Finance	90000	
15151	Mozart	Music	40000	
22222	Einstein	Physics	95000	
32343	El Said	History	60000	
33456	Gold	Physics	87000	
45565	Katz	Comp. Sci.	75000	
58583	Califieri	History	62000	
76543	Singh	Finance	80000	
76766	Crick	Biology	72000	
83821	Brandt	Comp. Sci.	92000	
98345	Kim	Elec. Eng.	80000	

32222	Verdi	Music	48000	

Multitable Clustering

Πολλές σχέσεις σε ένα αρχείο

department

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Physics	Watson	70000

instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
83821	Brandt	Comp. Sci.	92000

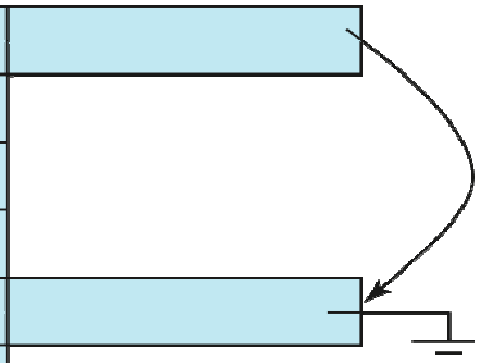
multitable clustering of *department* and *instructor*

Comp. Sci.	Taylor	100000
45564	Katz	75000
10101	Srinivasan	65000
83821	Brandt	92000
Physics	Watson	70000
33456	Gold	87000

Multitable clustering

- Καλό για queries που περιέχουν *department* $\bowtie$ *instructor*, και για queries που αφορούν τους instructors ενός department
- Κακό για τα queries που αφορούν μόνο το *department*
- Αλυσίδες δεικτών για εγγραφές μιας σχέσης

Comp. Sci.	Taylor	100000	
45564	Katz	75000	
10101	Srinivasan	65000	
83821	Brandt	92000	
Physics	Watson	70000	
33456	Gold	87000	



Αποθήκευση του Data Dictionary (Λεξικό)

Data dictionary (system catalog) κρατά τα μετά-δεδομένα: δηλαδή, δεδομένα για τα δεδομένα, όπως

- Πληροφορίες για Σχέσεις
  - Ονόματα των σχέσεων
  - Ονόματα και τύπους των γνωρισμάτων σε κάθε σχέση
  - Ονόματα και ορισμούς των όψεων - views
  - Περιορισμούς Ακεραιότητας
- User and accounting information, including passwords
- Statistical and descriptive data
  - number of tuples in each relation
- Physical file organization information
  - How relation is stored (sequential/hash/…)
  - Physical location of relation
    - » operating system file name or
    - » disk addresses of blocks containing records of the relation
- Information about indices

Αποθήκευση του Data Dictionary (συνέχεια)

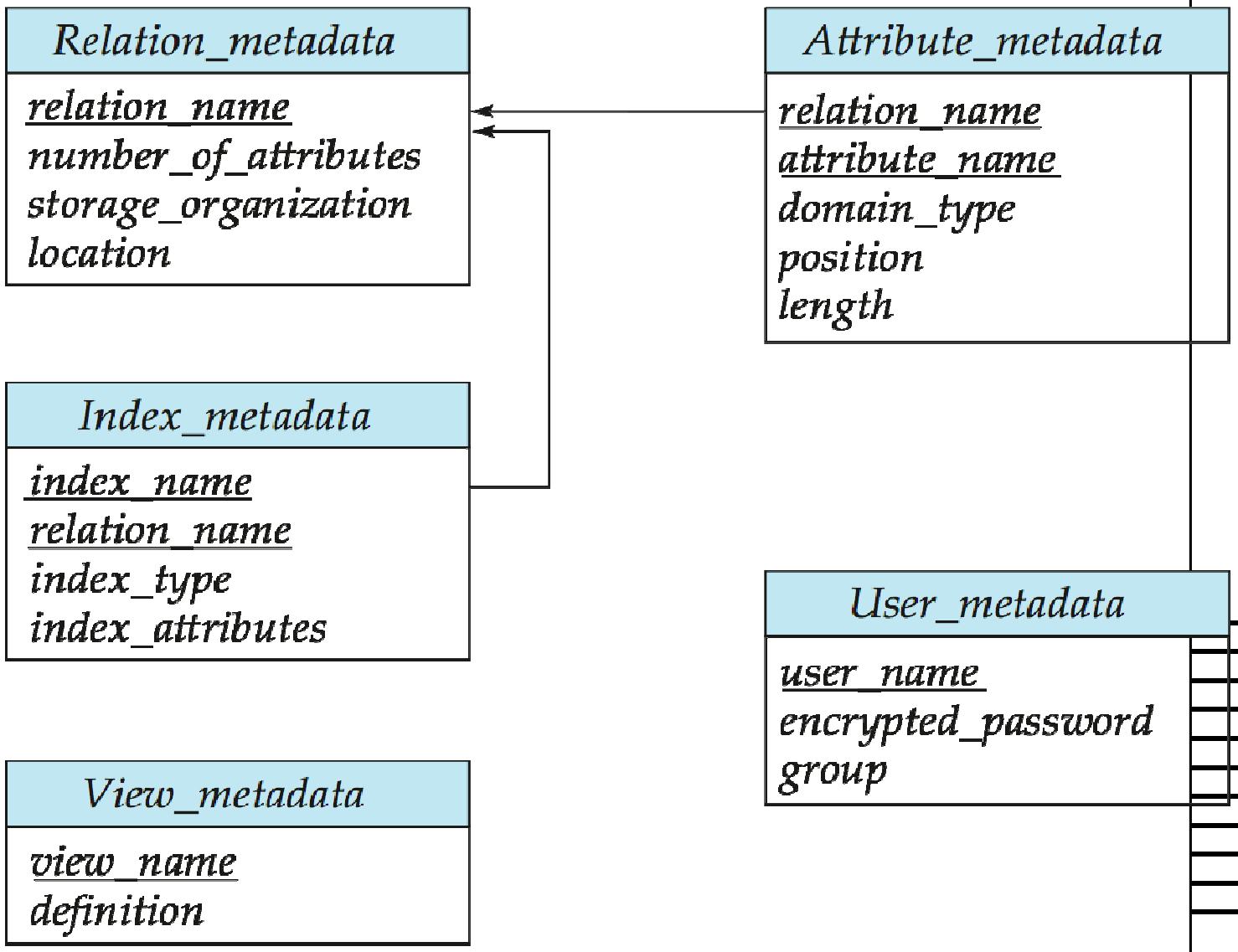
- Δομή του Λεξικού: Χρησιμοποιεί εναλλακτικά:
  - Ειδικές Δομές Δεδομένων προσανατολισμένες για αποδοτική πρόσβαση
  - Ένα σύνολο σχέσεων, με προσανατολισμό πάλι την απόδοση

  (συνήθως αποτελεί την προτιμητέα λύση)
- Ενδεικτικά, Το Λεξικό για μια Βάση Δεδομένων:

Relation-metadata = (relation-name, number-of-attributes, storage-organization, location)
Attribute-metadata = (attribute-name, relation-name, domain-type, position, length)
User-metadata = (user-name, encrypted-password, group)
Index-metadata = (index-name, relation-name, index-type, index-attributes)
View-metadata = (view-name, definition)

Data Dictionary

- Ειδικές Δομές Δεδομένων για αποδοτική πρόσβαση
- Ένα σύνολο σχέσεων
  (συνήθως αποτελεί την προτιμητέα λύση)



Storage Access

- Ένα αρχείο βάσης χωρίζεται σε σταθερού μήκους **blocks**. Τα blocks είναι μονάδες δέσμευσης χώρου και μεταφοράς δεδομένων
- Ένα σύστημα βάσης προσπαθεί να ελαχιστοποιήσει τη μεταφορά από blocks ανάμεσα στον δίσκο και στη μνήμη – κρατάμε όσο περισσότερα blocks μπορούμε στη μνήμη
- **Buffer** – κομμάτι της μνήμης που είναι διαθέσιμο για την αποθήκευση αντιγράφων block του δίσκου
- **Buffer manager** – υποσύστημα υπεύθυνο για την δέσμευση χώρου buffer

Buffer Manager

- Ένα πρόγραμμα καλεί τον buffer manager όταν χρειάζεται κάποιο block από τον δίσκο
  1. Αν το block είναι ήδη στον buffer, ο buffer manager επιστρέφει την διεύθυνση του block στην κύρια μνήμη
  2. Διαφορετικά, ο buffer manager
     1. Δεσμεύει χώρο στον buffer για το block
        1. Αντικαθιστώντας κάποιο παλιότερο αν χρειάζεται
        2. Το block αυτό γράφεται πάλι στον δίσκο (αν έχει αλλαγές)
     2. Διαβάζει το block από τον δίσκο στον buffer, και επιστρέφει την διεύθυνση του block στη μνήμη

Στρατηγική αντικατάστασης buffer

- Τα περισσότερα λειτουργικά αντικαθιστούν το ελάχιστα χρησιμοποιούμενο block **least recently used** (LRU strategy)
- Ιδέα – χρησιμοποίησε τα μοτίβα χρήσης του παρελθόντος για να προβλέψεις μελλοντική χρήση
- Queries have well-defined access patterns (such as sequential scans), and a database system can use the information in a user's query to predict future references
  - LRU can be a bad strategy for certain access patterns involving repeated scans of data
    - » For example: when computing the join of 2 relations r and s by a nested loops

      for each tuple *tr* of *r* do

        for each tuple *ts* of *s* do

          if the tuples *tr* and *ts* match …
  - Mixed strategy with hints on replacement strategy provided by the query optimizer is preferable

Στρατηγική αντικατάστασης buffer

- **Pinned block** – memory blocks που δεν επιτρέπεται να γραφτούν πάλι στον δίσκο
- **Toss-immediate** (αναγκαστική έξοδος) – απελευθερώνει τη μνήμη ενός block μόλις υποστεί επεξεργασία το τελευταίο tuple του
- **Most recently used (MRU) strategy** – υποψήφιο για αποχώρηση αυτό που χρησιμοποιήθηκε τελευτάιο
- Ο Buffer manager μπορεί να χρησιμοποιεί στατιστικά σχετικά με την πιθανότητα ένα request να αναφέρεται σε συγκεκριμένη σχέση
  - E.g., the data dictionary is frequently accessed. Heuristic: keep data-dictionary blocks in main memory buffer
- Buffer managers also support **forced output** of blocks for the purpose of recovery