

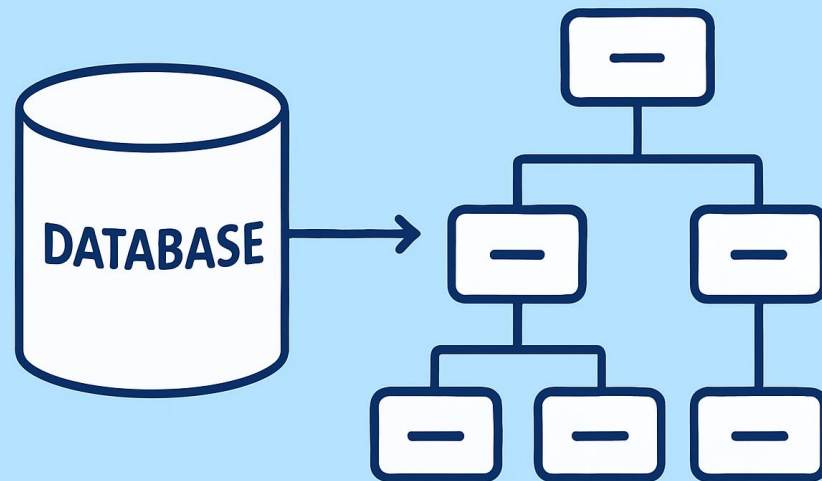
Φυσική Σχεδίαση Βάσεων Δεδομένων (Μέρος 1ο)

Αρχεία, Ευρετήρια και B-Trees

Φροντιστήριο

Βοηθός Μαθήματος: Μιχαήλ Γαλλιάρης
(email: csdp1413@csd.uoc.gr)

ΗΥ-360 Αρχεία και Βάσεις Δεδομένων
05/12/2025

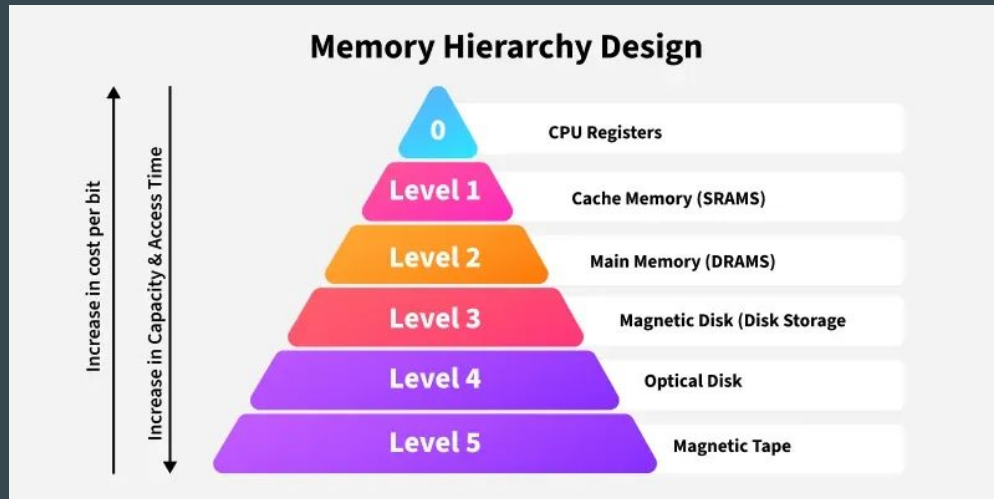


Τι θα δούμε σήμερα

- Φυσική Αποθήκευση: Γιατί ο Δίσκος είναι το Bottleneck;
- Οργάνωση Αρχείων: Heap vs. Sorted Files.
- Εισαγωγή στα Ευρετήρια (Indexes).
- Η Δομή B-Tree: Θεωρία και Χαρακτηριστικά.
- B-Tree Visual Tool
- Άσκηση: Υπολογισμός των παραμέτρων d και e .

Δευτερεύουσα Μνήμη & Κόστος Προσπέλασης

- Οι Βάσεις Δεδομένων είναι πολύ μεγάλες για να χωρέσουν στη RAM.
- Η μεταφορά από τον δίσκο (I/O) είναι η πιο "ακριβή" διαδικασία.
- Μονάδα Μέτρησης: Δεν μετράμε χρόνο (sec), μετράμε Block Accesses (πόσα blocks διαβάζουμε).
- Block (ή Page): Η μικρότερη μονάδα μεταφοράς δεδομένων (π.χ. 4KB, 8KB).



Από τις Εγγραφές στα Blocks

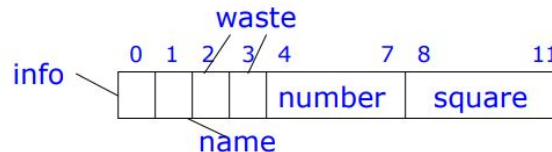
- Record (Εγγραφή): Η φυσική αποθήκευση μιας πλειάδας (tuple).
- Blocking Factor (R ή bfr): Πόσες εγγραφές χωράνε σε ένα Block;

- Ο Τύπος:

$$R = \lfloor \frac{B - H}{S} \rfloor$$

- B: Μέγεθος Block (bytes)
H: Header Block (bytes - συχνά αμελητέο ή 0 στις ασκήσεις)
S: Μέγεθος Εγγραφής (bytes)

■ **Example:** Assume blocks of length 64. We wish to store records of fixed length representing numbers.



Βασικές Μέθοδοι: Heap, Sorted & Hash

Μέθοδος	Περιγραφή	Πλεονεκτήματα	Μειονεκτήματα
Σωρός (Heap File)	Τυχαία σειρά εγγραφών (χρονολογική).	Ταχύτατη Εισαγωγή (Append στο τέλος).	Αργή Αναζήτηση (Linear Scan).
Ταξινομημένο (Sorted File)	Ταξινόμηση βάσει κλειδιού.	Γρήγορη Αναζήτηση (Binary Search).	Αργή Εισαγωγή (Απαιτεί αναδιάταξη).
Κατακερματισμός (Hashed File)	Οργάνωση σε Buckets μέσω συνάρτησης Hash.	Άμεση Πρόσβαση $O(1)$ για ισοτιμίες (=).	Αδυναμία Range Search (Range Queries $> <$).

Τι είναι το Index;

- Μια βοηθητική δομή που περιέχει ζεύγη (Key, Pointer).
- Επιτρέπει την άμεση εύρεση του Block χωρίς να σαρώσουμε (scan) όλο το αρχείο.
- Στόχος: Συνδυασμός ταχύτητας Sorted (στο Search) και Heap (στο Insert).
- Η πιο διαδεδομένη δομή στις Βάσεις Δεδομένων είναι τα B-Trees.

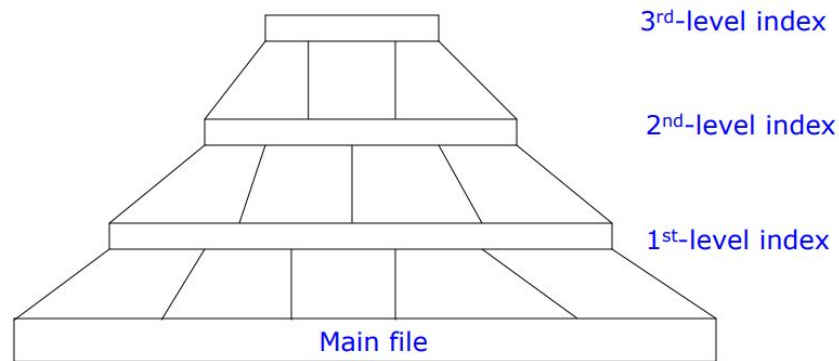
■ For every block b of the file, there is a record (v,b) in the index; v is a key value that is at least as low as any key value on b , but higher than any key value on any block preceding b . If b is the first block, $v=-\infty$ is used.

Χαρακτηριστικά B-Tree (και B+ Tree)

- Balanced (Ισορροπημένο): Όλα τα φύλλα βρίσκονται στο ίδιο επίπεδο.
- Δυναμικό: Το ύψος προσαρμόζεται αυτόματα.
- Δομή:
 - Εσωτερικοί Κόμβοι (Index Nodes): Περιέχουν μόνο οδηγούς (pointers & keys).
 - Φύλλα (Data Nodes): Περιέχουν τα πραγματικά δεδομένα (Records).
- Αυτό που περιγράφουμε είναι τεχνικά B+ Tree, το οποίο χρησιμοποιείται συνήθως στις ΒΔ.

- B-Trees

- Index files can become quite large for large main files
- Indices on index files are possible



Δυναμική Προσαρμογή & Ισορροπία

- Δυναμική Δομή: Το B-Tree προσαρμόζεται αυτόματα καθώς εισάγουμε ή διαγράφουμε δεδομένα, διατηρώντας το ύψος του ελάχιστο.
- Αναζήτηση (Search):
 - Ξεκινάμε από τη Ρίζα.
 - Οι δείκτες κατευθύνουν την αναζήτηση στο σωστό υποδέντρο (π.χ. "τιμές < 50 αριστερά").
- Εισαγωγή (Insertion) & Split:
 - Τα δεδομένα εισάγονται πάντα στα Φύλλα.
 - Overflow: Αν ένα φύλλο γεμίσει, διασπάται (Split) στα δύο.
 - Το μεσαίο κλειδί "προάγεται" (ανεβαίνει) στον πατέρα.
- Αποτέλεσμα: Το δέντρο ψηλώνει μόνο όταν διασπαστεί η Ρίζα. Έτσι παραμένει πάντα ισορροπημένο.

Οπτικοποίηση Λειτουργίας (Visual Tool)

- Εργαλείο: USFCA B+ Tree Visualization
- Σενάρια που εξετάζουμε:
 1. Sequential Insert: Εισαγωγή αριθμών με τη σειρά (παρατήρηση του Split στα δεξιά).
 2. Random Insert: Τυχαία εισαγωγή και αναδιάταξη του δέντρου.
 3. Παρατήρηση: Πώς το δέντρο διατηρείται ισορροπημένο (ίδιο βάθος σε όλα τα φύλλα).
- <https://www.cs.usfca.edu/~galles/visualization/BPlusTree.html>

Χωρητικότητα Κόμβων (Ο Συμβολισμός μας)

d (Για Index Nodes):

- Max Κλειδιά: $2d - 1$
- Max Δείκτες: $2d$
- Min Δείκτες: d (εκτός ρίζας)

e (Για Data Nodes / Φύλλα):

- Max Εγγραφές: $2e - 1$
- Min Εγγραφές: e

- A B-tree is characterized by two parameters d, e :
 - d and e are integers such that, the number of index records a block can hold is $2*d-1$ and the number of records of the main file a block can hold is $2*e-1$.

Υπολογισμός Παραμέτρων Δέντρου

- Δεδομένα: Block = 4096 bytes, Record = 200 bytes, Key = 20 bytes, Pointer = 4 bytes.
- Ερώτημα A: Πόσο είναι το e (για τα φύλλα);
- Ερώτημα B: Πόσο είναι το d (για τους κόμβους ευρετηρίου);

Λύση Υπολογισμού

Υπολογισμός e (Data Nodes):

- $R = \lfloor 4096 / 200 \rfloor = 20$.
- Άρα Max Records = 20.
- $2e - 1 \leq 20 \rightarrow 2e \leq 21 \rightarrow e \leq 10.5 \rightarrow \mathbf{e = 10}$.

Υπολογισμός d (Index Nodes):

- Index Entry Size = Key + Pointer = $20 + 4 = 24$ bytes.
- Entries ανά Block = $\lfloor 4096 / 24 \rfloor = 170$.
- Max Keys = $2d - 1 \leq 170 \rightarrow 2d \leq 171 \rightarrow d \leq 85.5 \rightarrow \mathbf{d = 85}$.

Τέλος φροντιστηρίου

Ερωτήσεις;

ΗΥ - 360 Αρχεία και Βάσεις Δεδομένων
Φυσική Σχεδίαση Βάσεων Δεδομένων (Μέρος 1ο)
Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης
(05/12/25)