

Επισκόπηση Αποθήκευσης και Ευρετηριοποίησης

Κεφάλαιο 8

*“How index-learning turns no student pale
Yet holds the eel of science by the tail.”
-- Alexander Pope (1688-1744)*

Δεδομένα σε Εξωτερική Μνήμη

- Δίσκοι: Ανακτούν τυχαία σελίδα με σταθερό κόστος
 - Αλλά η ανάγνωση πολλών συνεχόμενων σελίδων έχει χαμηλότερο κόστος από την ανάγνωσή τους με τυχαία σειρά
- Ταινίες: Διαβάζουν τις σελίδες μόνο με τη σειρά
 - Φθηνότερες από τους δίσκους - κατάλληλες για αρχειακή χρήση
- Οργάνωση Αρχείου: Μέθοδος τακτοποίησης ενός αρχείου με εγγραφές σε εξωτερική μνήμη.
 - **Record id (rid)**: αρκετό για τον φυσικό εντοπισμό μιας εγγραφής
 - **Ευρετήρια**: δομές δεδομένων που επιτρέπουν τον εντοπισμό των rids εγγραφών με δοσμένες τιμές στα πεδία κλειδιού αναζήτησης του ευρετηρίου
- Αρχιτεκτονική: ο Διαχειριστής Ενδιάμεσης Μνήμης μεταφέρει σελίδες από την εξωτερική μνήμη στην (κύρια) ενδιάμεση μνήμη. Τα επίπεδα αρχείων και ευρετηρίων καλούν το διαχειριστή ενδιάμεσης μνήμης.

Εναλλακτικές Οργανώσεις Αρχείων

- Πολλές εναλλακτικές οργανώσεις, κάθε μια κατάλληλη σε κάποια περίπτωση και όχι τόσο κατάλληλη σε άλλες:
 - Αρχεία Σωρού (Heap ή Random Order files): κατάλληλα όταν η τυπική προσπέλαση είναι η σάρωση για την ανάκτηση όλων των εγγραφών.
 - Ταξινομημένα Αρχεία (Sorted Files): προτιμότερα όταν πρέπει να ανακτηθούν οι εγγραφές με κάποια σειρά, ή όταν χρειάζεται μόνο ένα συγκεκριμένο υποσύνολο των εγγραφών.
 - Ευρετήρια: δομές δεδομένων που οργανώνουν τις εγγραφές μέσω δέντρων ή κατακερματισμού.

Όπως τα ταξινομημένα αρχεία, επιταχύνουν τις αναζητήσεις υποσυνόλων εγγραφών βασιζόμενα στις τιμές συγκεκριμένων πεδίων ('κλειδιού αναζήτησης')

Οι ενημερώσεις είναι ταχύτερες από ότι στα ταξινομημένα αρχεία

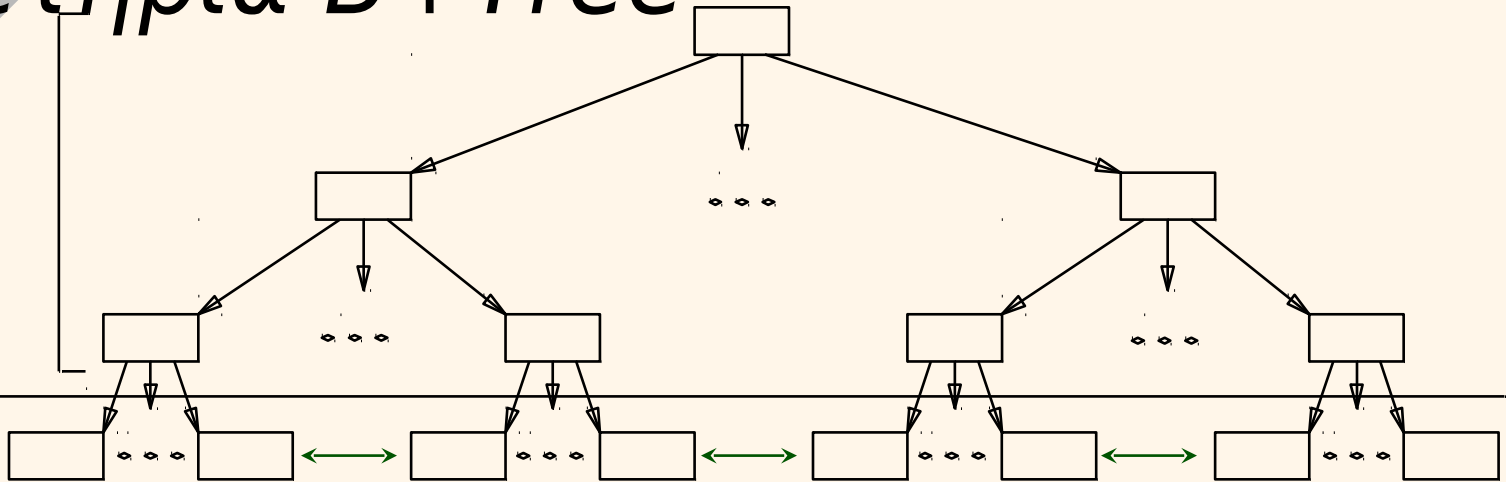
Ευρετήρια (Indexes)

- Ένα ευρετήριο πάνω σε ένα αρχείο επιταχύνει τις επιλογές πάνω στα **πεδία κλειδιού αναζήτησης** του ευρετηρίου.
 - Ένα οποιοδήποτε υποσύνολο πεδίων ενός πίνακα μπορεί να είναι το κλειδί αναζήτησης ενός ευρετηρίου πάνω στον πίνακα.
 - Το **κλειδί αναζήτησης δεν** είναι το ίδιο με το **κλειδί** (ελάχιστο υποσύνολο πεδίων που προσδιορίζουν μοναδικά μια εγγραφή).
- Ένα ευρετήριο περιέχει μια συλλογή **καταχωρίσεων δεδομένων**, και υποστηρίζει την αποτελεσματική ανάκτηση όλων των καταχωρίσεων δεδομένων k^* για μια δοσμένη τιμή k του κλειδιού αναζήτησης.
 - δοσμένης μιας καταχώρισης δεδομένων k^* , μπορούμε να βρούμε την εγγραφή με κλειδί k με \max ένα I/O (περισσότερα σε λίγο...)

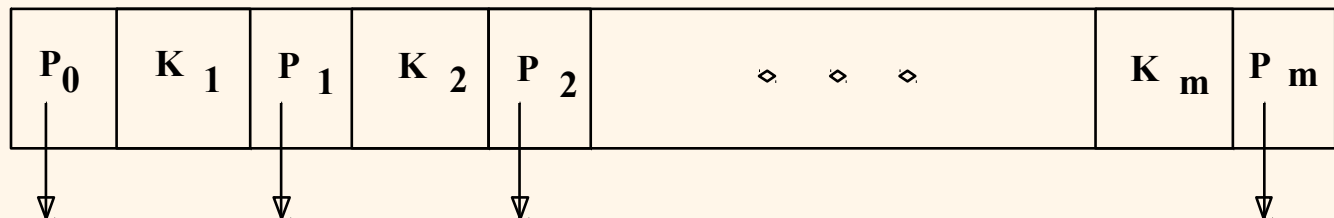
Ευρετήρια B+Tree

Σελίδες
Μη-φύλλα

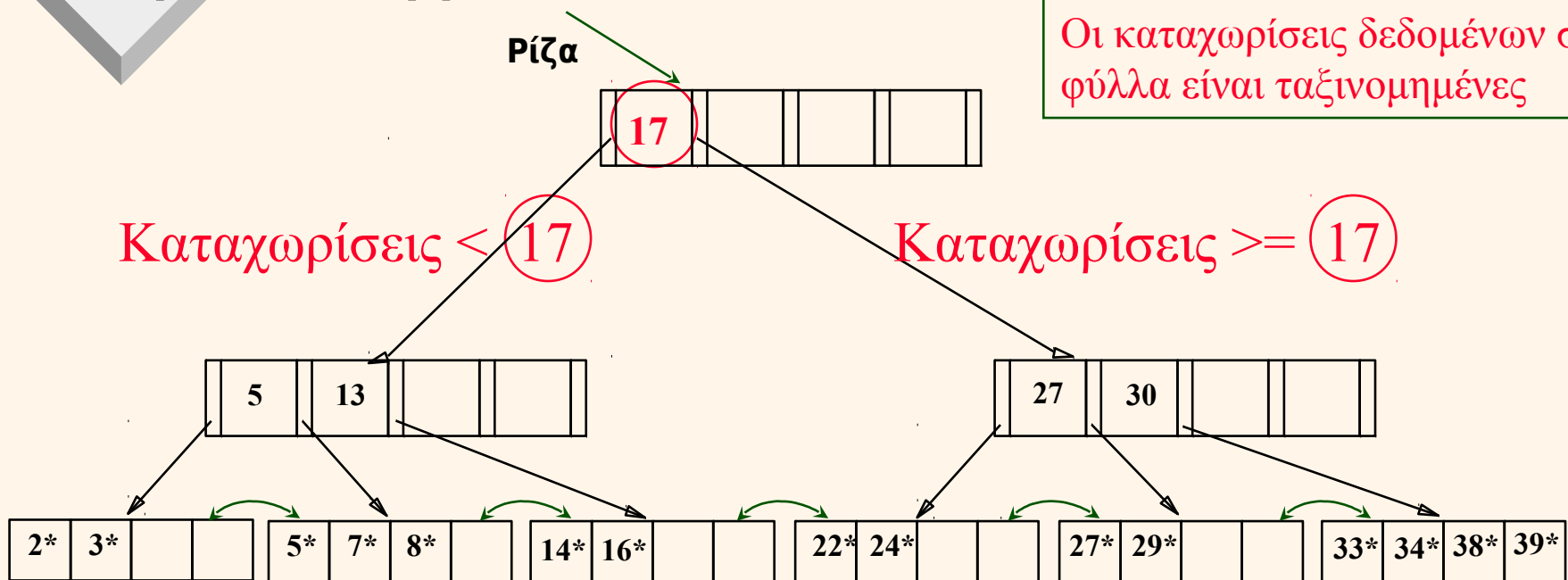
Σελίδες
Φύλλα
(Ταξινόμηση ως προς το κλειδί αναζήτησης)



- Οι σελίδες φύλλα περιέχουν **καταχωρίσεις δεδομένων**, και είναι συνδεδεμένες (prev & next)
- Οι σελίδες μη-φύλλα έχουν **καταχωρίσεις ευρετηρίου** που χρησιμοποιούνται μόνο για να κατευθύνουν τις αναζητήσεις:
καταχώριση ευρετηρίου



Παράδειγμα B+Tree



- Βρες την 28^* , 29^* , όλες όσες είναι $> 15^*$ και $< 30^*$
- Εισαγωγή/Διαγραφή: Βρες την καταχώριση δεδομένων στο φύλλο και κάνε την αλλαγή. Μερικές φορές χρειάζεται να ενημερωθεί ο γονικός κόμβος.
 - και μερικές φορές οι αλλαγές προωθούνται προς τη ρίζα

Κατακερματισμένα Ευρετήρια

- Κατάλληλα για επιλογές με ισότητα.
- Ένα Αρχείο είναι μια συλλογή κάδων.
 - Κάδος = **πρωτεύουσα σελίδα** και μηδέν ή περισσότερες **σελίδες** υπερχείλισης.
 - Οι κάδοι περιέχουν καταχωρίσεις δεδομένων.
- **Συνάρτηση κατακερματισμού h** : $h(r)$ = κάδος στον οποίο ανήκει η (καταχώριση για την) εγγραφή r . Η h χρησιμοποιεί τα πεδία του **κλειδιού αναζήτησης** του r .
 - Δεν χρειάζονται 'καταχωρίσεις ευρετηρίου' στη δομή αυτή.

Εναλλακτικές για τις καταχωρίσεις δεδομένων k^* σε ένα ευρετήριο

- Μια καταχώριση δεδομένων k^* μπορεί να αποθηκεύσει:
 - (1) Εγγραφή δεδομένων με τιμή κλειδιού k , ή
 - (2) $\langle k, rid \text{ της εγγραφής με τιμή κλειδιού αναζήτησης } k \rangle$, ή
 - (3) $\langle k, \text{λίστα } rid \text{ των εγγραφών με τιμή κλειδιού αναζήτησης } k \rangle$
- Η επιλογή της εναλλακτικής για τις καταχωρίσεις δεδομένων είναι ανεξάρτητη από την τεχνική ευρετηριοποίησης που χρησιμοποιείται για την εύρεση των καταχωρίσεων δεδομένων για μια τιμή κλειδιού k .
 - Παραδείγματα τεχνικών ευρετηριοποίησης: B+trees, δομές βασισμένες στον κατακερματισμό
 - Συνήθως, ένα ευρετήριο περιέχει βοηθητική πληροφορία που κατευθύνει την αναζήτηση στις επιθυμητές καταχωρίσεις δεδομένων

Εναλλακτικές για τις καταχωρίσεις δεδομένων (συνέχεια)

- **Εναλλακτική 1:**

- Στην περίπτωση αυτή, η δομή ευρετηρίου αποτελεί και οργάνωση αρχείου για τις εγγραφές (όπως τα αρχεία σωρού ή τα ταξινομημένα αρχεία).
- Το πολύ ένα ευρετήριο πάνω σε ένα σύνολο εγγραφών μπορεί να χρησιμοποιεί την Εναλλακτική 1. (Αλλιώς, υπάρχει διπλή καταχώριση των εγγραφών, που οδηγεί σε σπατάλη χώρου και πιθανή ασυνέπεια των δεδομένων.)
- Αν οι εγγραφές δεδομένων είναι πολύ μεγάλες, το πλήθος των σελίδων που περιέχουν καταχωρίσεις δεδομένων είναι υψηλό. Αυτό συνεπάγεται ότι η βοηθητική πληροφορία του ευρετηρίου θα είναι επίσης μεγάλη.

Εναλλακτικές για τις καταχωρίσεις δεδομένων (συνέχεια)

- **Εναλλακτικές 2 και 3:**

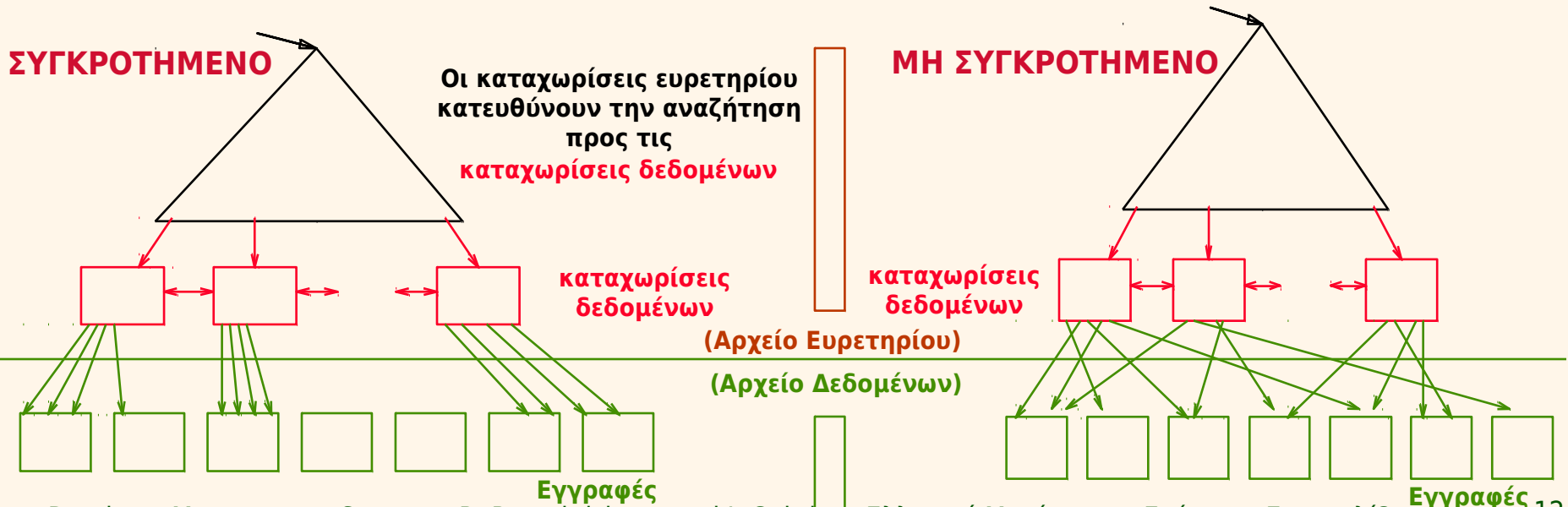
- Συνήθως, οι καταχωρίσεις δεδομένων είναι μικρότερες από τις εγγραφές δεδομένων. Έτσι, οι Εναλλακτικές αυτές είναι καλύτερες από την Εναλλακτική 1 για μεγάλα μεγέθη εγγραφών, ειδικά όταν τα κλειδιά αναζήτησης είναι μικρά. (Το ποσοστό της δομής του ευρετηρίου που χρησιμοποιείται για να κατευθύνει την αναζήτηση, και το οποίο εξαρτάται από το μέγεθος των καταχωρίσεων δεδομένων, είναι πολύ μικρότερο από ότι στην Εναλλακτική 1.)
- Η Εναλλακτική 3 είναι πιο συμπαγής από την Εναλλακτική 2, αλλά οδηγεί σε καταχωρίσεις δεδομένων μεταβλητού μήκους, ακόμα κι αν τα κλειδιά αναζήτησης είναι σταθερού μήκους.

Κατηγορίες Ευρετηρίων

- **Πρωτεύον (primary) vs. δευτερεύον (secondary):** Αν το κλειδί αναζήτησης περιέχει το πρωτεύον κλειδί, τότε το ευρετήριο ονομάζεται πρωτεύον.
 - **Μοναδικό (Unique)** ευρετήριο: Το κλειδί αναζήτησης περιέχει υποψήφιο κλειδί.
- **Συγκροτημένο (Clustered) vs. Μη-συγκροτημένο (Unclustered):** Αν η σειρά των εγγραφών είναι η ίδια, η σχεδόν η ίδια, με τη σειρά των καταχωρίσεων δεδομένων, τότε το ευρετήριο ονομάζεται συγκροτημένο.
 - Η Εναλλακτική 1 συνεπάγεται συγκροτημένο, αλλά όχι το αντίστροφο.
 - Ένα αρχείο μπορεί να είναι συγκροτημένο πάνω σε, το πολύ ένα, κλειδί αναζήτησης.
 - Το κόστος ανάκτησης εγγραφών μέσω ευρετηρίου ποικίλει *δραματικά* ανάλογα με το αν το ευρετήριο είναι συγκροτημένο ή όχι!

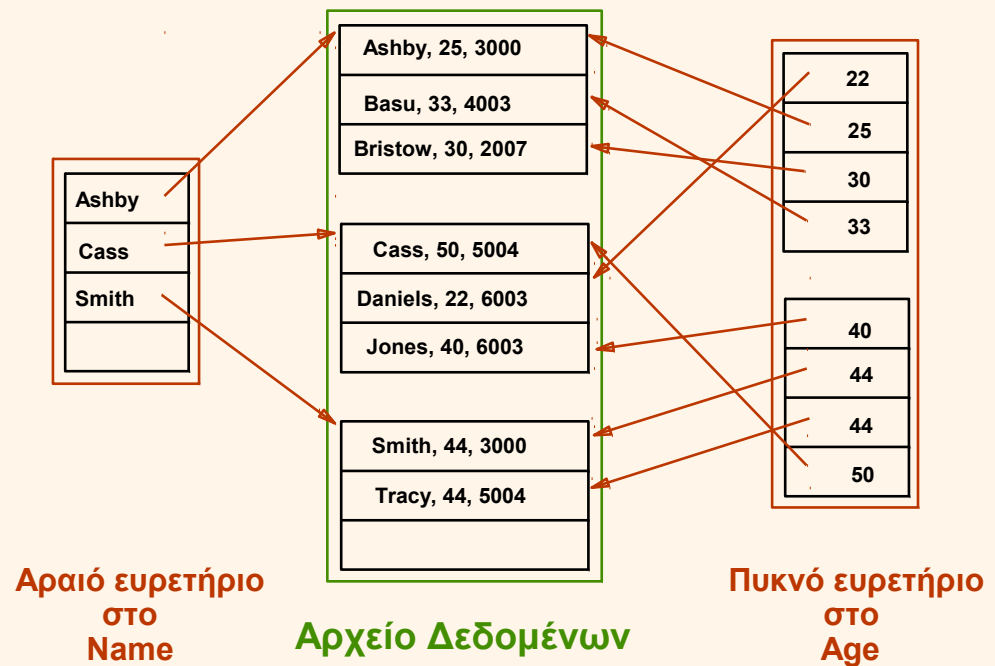
Συγκροτημένο vs. Μη-συγκροτημένο

- Υποθέστε ότι για τις καταχωρίσεις δεδομένων χρησιμοποιείται η Εναλλακτική 2, και ότι οι εγγραφές είναι αποθηκευμένες σε αρχείο Σωρού.
 - Για την κατασκευή του συγκροτημένου ευρετηρίου, ταξινομούμε το αρχείο Σωρού (με ελεύθερο χώρο για μελλοντικές εισαγωγές).
 - Κατά τις εισαγωγές μπορεί να χρειαστούν σελίδες Υπερχείλισης. (Άρα η σειρά των εγγραφών θα είναι παρόμοια, αλλά όχι ίδια, με τη σειρά ταξινόμησης.)



Πυκνό vs. Αραιό

- Αν υπάρχει έστω και μια καταχώριση δεδομένων ανά τιμή κλειδιού αναζήτησης (για κάποια εγγραφή), τότε το ευρετήριο είναι **ΠΥΚΝΟ**.
 - Η Εναλλακτική 1 οδηγεί σε πυκνά ευρετήρια.
 - Κάθε αραιό ευρετήριο είναι συγκροτημένο!
 - Τα αραιά ευρετήρια είναι μικρότερα. Όμως τα πυκνά ευρετήρια δίνουν τη δυνατότητα για διάφορες βελτιστοποιήσεις.



Μοντέλο Κόστους για την Ανάλυση

- Για απλοποίηση, αγνοούμε το κόστος CPU:
 - **B:** το πλήθος των σελίδων δεδομένων
 - **R:** το πλήθος των εγγραφών ανά σελίδα
 - **D:** (μέσος) χρόνος για την ανάγνωση ή εγγραφή σελίδας
- Η μέτρηση του αριθμού των λειτουργιών I/O αγνοεί το κέρδος της προφόρτωσης μπλοκ σελίδων – συνεπώς, ακόμη και το κόστος I/O είναι κατά προσέγγιση.
- Ανάλυση της μέσης περίπτωσης – βασίζεται σε πολλές απλοποιητικές υποθέσεις.
- Είναι όμως επαρκής για να δούμε τη γενική εικόνα!

Σύγκριση Οργανώσεων Αρχείου

- Αρχεία Σωρού (τυχαία σειρά, εισαγωγή στο EOF)
- Ταξινομημένα αρχεία, ταξινομημένα στο $\langle \text{age}, \text{sal} \rangle$
- Συγκροτημένο αρχείο B+tree, Εναλλακτική 1, κλειδί αναζήτησης $\langle \text{age}, \text{sal} \rangle$
- Αρχείο Σωρού με μη-συγκροτημένο ευρετήριο B+tree στο κλειδί αναζήτησης $\langle \text{age}, \text{sal} \rangle$
- Αρχείο Σωρού με μη-συγκροτημένο ευρετήριο κατακερματισμού στο κλειδί αναζήτησης $\langle \text{age}, \text{sal} \rangle$

Σύγκριση ως προς τις Λειτουργίες

- Σάρωση (Scan): Φέρει όλες τις εγγραφές από το δίσκο
- Αναζήτηση με ισότητα (Equality search)
- Αναζήτηση με διάστημα (Range selection)
- Εισαγωγή μιας εγγραφής (Insert a record)
- Διαγραφή μιας εγγραφής (Delete a record)

Υποθέσεις της Ανάλυσης

- Αρχεία Σωρού:
 - Αναζήτηση με ισότητα στο κλειδί – ακριβώς μια απάντηση.
- Ταξινομημένα Αρχεία:
 - Τα αρχεία συμπύσσονται μετά από διαγραφές.
- Ευρετήρια:
 - Εναλλακτικές 2 και 3: μέγεθος καταχώρισης δεδομένων = 10% του μεγέθους της εγγραφής
 - Κατακερματισμένα Αρχεία: Δεν υπάρχουν κάδοι υπερχείλισης
σελίδες 80% γεμάτες => Μέγεθος αρχείου = 1,25 μεγέθους δεδομένων
 - Δέντρα: σελίδες 67% γεμάτες (τυπικό για B+trees)
Συνεπάγεται μέγεθος αρχείου = 1,5 μεγέθους δεδομένων

Υποθέσεις (συνέχεια)

- Σαρώσεις:
 - Οι σελίδες του επιπέδου φύλλων στα δενδρικά ευρετήρια είναι συνδεδεμένες.
 - Για τα μη-συγκροτημένα ευρετήρια έχουμε σάρωση των καταχωρίσεων δεδομένων και του αρχείου με τις εγγραφές.
- Αναζητήσεις με διάστημα:
 - Για να περιορίσουμε το σύνολο των εγγραφών που πρέπει να ανακτήσουμε, χρησιμοποιούμε δενδρικά ευρετήρια, αλλά αγνοούμε τα ευρετήρια κατακερματισμού.

Κόστος λειτουργιών

	(α) Σάρωση	(β) Με ισότητα	(γ) Με διάστημα	(δ) Εισαγωγή	(ε) Διαγραφή
(1) Σωρού	BD	0.5BD	BD	2D	Search + D
(2) Ταξινομημένο	BD	D log₂ B	D(log₂ B + #pgs with matching recs)	Search + BD	Search + BD
(3) Συγκροτημένο	1.5BD	D log_F 1.5B	D(log_F 1.5B + #pgs with matching recs)	Search + D	Search + D
(4) Μη Συγκροτημένο	BD(R+0.15)	D(1 + log_F 0.15B)	D(log_F 0.15B + #pgs with matching recs)	Search + 2D	Search + 2D
(5) Μη Συγκροτημένο Κατακερματισμού	BD(R+0.125)	2D	BD	Search + 2D	Search + 2D

- Πίσω από αυτές τις (χονδρικές) εκτιμήσεις υπάρχουν πολλές υποθέσεις!



Κατανοώντας το φόρτο

- Για κάθε αίτημα αναζήτησης:
 - Ποιους πίνακες προσπελάζει;
 - Ποια πεδία ανακτώνται;
 - Ποια πεδία εμπλέκονται σε συνθήκες επιλογής/σύζευξης; Πόσο επιλεκτικές είναι πιθανόν να είναι οι συνθήκες αυτές;
- Για κάθε αίτημα ενημέρωσης:
 - Ποια πεδία εμπλέκονται σε συνθήκες επιλογής/σύζευξης; Πόσο επιλεκτικές είναι πιθανόν να είναι οι συνθήκες αυτές;
 - Ο τύπος της ενημέρωσης (INSERT/DELETE/UPDATE) και τα πεδία που εμπλέκονται.



Επιλογή Ευρετηρίων

- Ποια ευρετήρια θα πρέπει να δημιουργήσουμε;
 - Ποιοι πίνακες θα πρέπει να έχουν ευρετήρια; Ποιο(α) πεδίο(α) πρέπει να είναι το κλειδί αναζήτησης; Μήπως πρέπει να κατασκευάσουμε πολλά ευρετήρια;
- Για κάθε ευρετήριο, τί είδους ευρετήριο θα πρέπει να είναι;
 - Συγκροτημένο; Κατακερματισμού/Δενδρικό;

Επιλογή Ευρετηρίων (συνέχεια)

- **Μια προσέγγιση:** Θεωρείστε τα πιο σημαντικά αιτήματα κατά σειρά. Θεωρείστε το καλύτερο πλάνο εκτέλεσης χρησιμοποιώντας τα υπάρχοντα ευρετήρια, και δείτε αν ένα επιπλέον νέο ευρετήριο κάνει δυνατό ένα καλύτερο πλάνο. Αν ναι, δημιουργήστε το νέο ευρετήριο.
 - Προφανώς, τα παραπάνω συνεπάγονται ότι πρέπει να κατανοούμε πώς το DBMS εκτιμά το κόστος των αιτημάτων και δημιουργεί **πλάνα εκτίμησης αιτημάτων!**
 - Προς το παρόν, ασχολούμαστε με απλά αιτήματα ενός πίνακα.
- Πριν τη δημιουργία ενός ευρετηρίου, πρέπει να υπολογίσουμε τις επιπτώσεις των ενημερώσεων στο φόρτο του συστήματος!
 - **Θέμα ρύθμισης:** Τα ευρετήρια κάνουν τα αιτήματα ταχύτερα και τις ενημερώσεις πιο αργές. Επίσης, απαιτούν χώρο στο δίσκο.

Αρχές Επιλογής Ευρετηρίων

- Πεδία στο WHERE είναι υποψήφια κλειδιά αναζήτησης.
 - Συνθήκες με ισότητα συνιστούν ευρετήριο κατακερματισμού.
 - Αίτημα με διάστημα τιμών συνιστά δενδρικό ευρετήριο.

Η Συγκρότηση είναι ιδιαίτερα χρήσιμη για αιτήματα με διάστημα - αλλά και για αιτήματα με ισότητα όταν υπάρχουν πολλά διπλότυπα.
- Όταν η WHERE περιέχει πολλές συνθήκες, μπορούμε να θεωρήσουμε σύνθετο κλειδί αναζήτησης.
 - Η σειρά των πεδίων είναι σημαντική για τις αναζητήσεις με διάστημα.
 - Τέτοια ευρετήρια μπορεί να επιτρέψουν στρατηγικές υπολογισμού **με ευρετήριο-μόνο (index-only)** για σημαντικά αιτήματα.

Σε τέτοιες περιπτώσεις, η συγκρότηση δεν είναι σημαντική!
- Επίλεξε ευρετήρια που ωφελούν πολλά αιτήματα. Εφόσον υπάρχει ένα το πολύ συγκροτημένο ευρετήριο ανά πίνακα, διάλεξε το κλειδί αναζήτησής του έτσι ώστε να βασίζεται σε σημαντικά αιτήματα που θα επωφεληθούν από τη συγκρότηση.

Παραδείγματα Συγκροτημένων Ευρετηρίων

- Ένα ευρετήριο B+tree στο *E.age* μπορεί να βρει τις εγγραφές που ικανοποιούν τη συνθήκη.
 - Πόσο επιλεκτική είναι η συνθήκη;
 - Είναι το ευρετήριο συγκροτημένο;
- Ας δούμε το αίτημα GROUP BY.
 - Αν υπάρχουν πολλές εγγραφές με *E.age > 10*, χρήση του ευρετηρίου στο *E.age* και ταξινόμηση των ανακτηθέντων εγγραφών μπορεί να έχει υψηλό κόστος.
 - Ένα συγκροτημένο ευρετήριο στο *E.dno* μπορεί να είναι καλύτερο!
- Αιτήματα με ισότητα και διπλοεγγραφές:
 - Συγκροτημένο στο *E.hobby* βοηθά!

```
SELECT E.dno
FROM Emp E
WHERE E.age>40
```

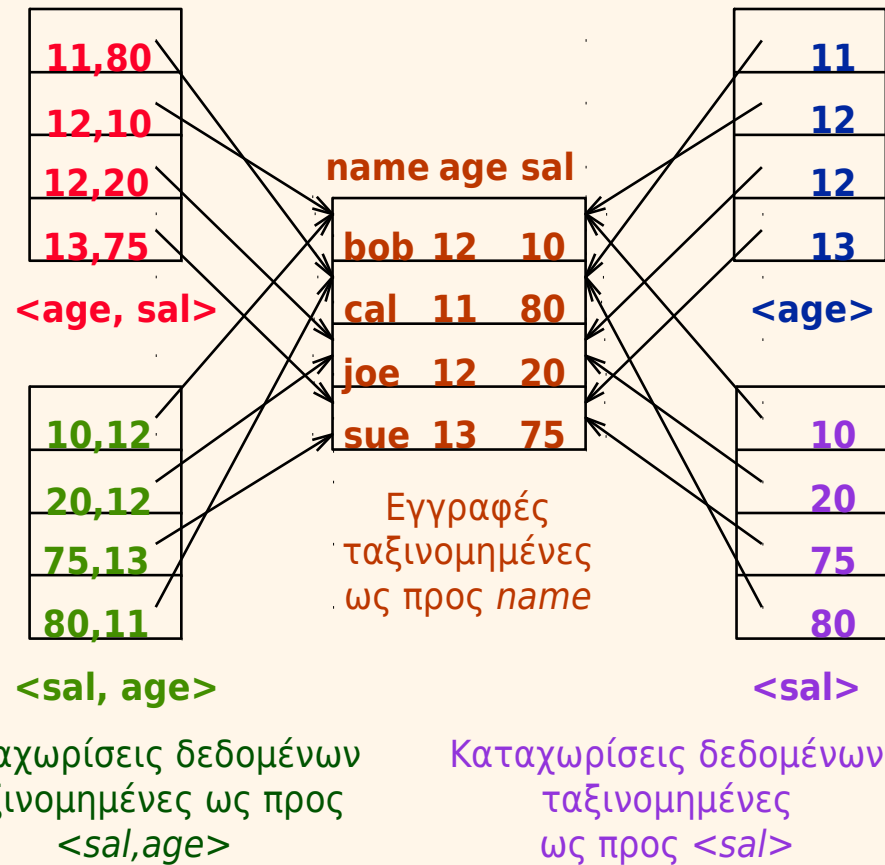
```
SELECT E.dno, COUNT (*)
FROM Emp E
WHERE E.age>10
GROUP BY E.dno
```

```
SELECT E.dno
FROM Emp E
WHERE E.hobby=Stamps
```


Ευρετήρια με σύνθετα κλειδιά αναζήτησης

- **Σύνθετα κλειδιά αναζήτησης:** Αναζήτηση με συνδυασμό πεδίων.
 - **Αίτημα με ισότητα:** Κάθε τιμή πεδίου ισούται με σταθερά. Π.χ., σχετικά με το ευρετήριο πάνω στο <sal,age>:
age=20 and sal =75
 - **Αίτημα με διάστημα:** Κάποια τιμή πεδίου δεν είναι σταθερά. Π.χ.:
 - age =20; or age=20 and sal > 10
- Οι καταχωρίσεις δεδομένων στο ευρετήριο είναι ταξινομημένες ως προς το κλειδί αναζήτησης ώστε να υποστηρίζουν αιτήματα με διάστημα.
 - **Λεξικογραφική ταξινόμηση, ή**
 - **Χωρική ταξινόμηση.**

Παραδείγματα ευρετηρίων σύνθετου κλειδιού με λεξικογραφική ταξινόμηση.



Σύνθετα Κλειδιά Αναζήτησης

- Για ανάκτηση των εγγραφών του Emp με $age=30$ AND $sal=4000$, είναι προτιμότερο ένα ευρετήριο στο $\langle age, sal \rangle$ από ένα ευρετήριο στο age ή το sal .
 - Η επιλογή του κλειδιού του ευρετηρίου είναι ανεξάρτητη της συγκρότησης.
- Αν η συνθήκη είναι: $20 < age < 30$ AND $3000 < sal < 5000$:
 - Το καλύτερο είναι συγκροτημένο ευρετήριο στο $\langle age, sal \rangle$ ή στο $\langle sal, age \rangle$.
- Αν η συνθήκη είναι: $age=30$ AND $3000 < sal < 5000$:
 - Ένα συγκροτημένο ευρετήριο στο $\langle age, sal \rangle$ είναι πολύ καλύτερο από ένα ευρετήριο στο $\langle sal, age \rangle$!
- Τα σύνθετα ευρετήρια είναι μεγαλύτερα και ενημερώνονται πιο συχνά.

Πλάνα με Ευρετήριο-μόνο

- Πολλά αιτήματα μπορούν να απαντηθούν χωρίς καθόλου ανάκτηση των εγγραφών ενός ή περισσότερων πινάκων αν υπάρχει ένα κατάλληλο ευρετήριο.

<E.dno>

```
SELECT E.dno, COUNT(*)  
FROM Emp E  
GROUP BY E.dno
```

<E.dno, E.sal>

Tree index!

```
SELECT E.dno, MIN(E.sal)  
FROM Emp E  
GROUP BY E.dno
```

<E.age, E.sal>

or

<E.sal, E.age>

Tree index!

```
SELECT AVG(E.sal)  
FROM Emp E  
WHERE E.age=25 AND  
E.sal BETWEEN 3000 AND 5000
```

Πλάνα με Ευρετήριο-μόνο (2)

- Πλάνα με ευρετήριο-μόνο είναι εφικτά αν το κλειδί είναι το $\langle \text{dno}, \text{age} \rangle$ ή έχουμε ένα δενδρικό ευρετήριο με κλειδί το $\langle \text{age}, \text{dno} \rangle$
 - Ποιο είναι καλύτερο;
 - Τί συμβαίνει αν θεωρήσουμε το δεύτερο αίτημα;

```
SELECT E.dno, COUNT (*)  
FROM Emp E  
WHERE E.age=30  
GROUP BY E.dno
```

```
SELECT E.dno, COUNT (*)  
FROM Emp E  
WHERE E.age>30  
GROUP BY E.dno
```

Πλάνα με Ευρετήριο-μόνο (3)

- Πλάνα με ευρετήριο-μόνο υπάρχουν και για αιτήματα που εμπλέκουν παραπάνω από έναν πίνακα – περισσότερα γι' αυτό αργότερα.

<E.dno>

```
SELECT D.mgr
FROM Dept D, Emp E
WHERE D.dno=E.dno
```

<E.dno,E.eid>

```
SELECT D.mgr, E.eid
FROM Dept D, Emp E
WHERE D.dno=E.dno
```