

Ευρετήρια

Ευρετήρια

Βάσεις Δεδομένων 2008-2009 Ευαγγελία Πλουρά 1

Ευρετήρια

- Ένα **ευρετήριο (index)** είναι μια **βοηθητική δομή αρχείου** που κάνει πιο αποδοτική την αναζήτηση μιας εγγραφής σε ένα αρχείο
- Το ευρετήριο καθορίζεται (συνήθως) σε **ένα γνώρισμα** του αρχείου που καλείται **πεδίο ευρετηριοποίησης (indexing field)**

Εγγραφή στο ευρετήριο:

Τιμή Πεδίου Ευρετηριοποίησης Δείκτης στο block της εγγραφής

Βάσεις Δεδομένων 2008-2009 Ευαγγελία Πλουρά 2

Ευρετήρια

Στόχος: αποδοτικές λειτουργίες αναζήτησης
Οι λειτουργίες ενημέρωσης γίνονται πιο αργές

Διαφορετικού τύπου εγγραφές ανάλογα με το πεδίο ευρετηριοποίησης:

(α) πεδίο διάταξης του αρχείου ή όχι
(β) κλειδί ή όχι
(πρωτεύον/δευτερεύον) - διαφορετικοί ορισμοί στα βιβλία

Βάσεις Δεδομένων 2008-2009 Ευαγγελία Πλουρά 3

Ευρετήρια

- **Πυκνό ευρετήριο:** μια καταχώρηση για κάθε εγγραφή του αρχείου
- **Μη πυκνό ευρετήριο**

Βάσεις Δεδομένων 2008-2009 Ευαγγελία Πλουρά 4

Πρωτεύον Ευρετήριο

Πρωτεύον ευρετήριο (primary index): ορισμένο στο **κλειδί διάταξης** του αρχείου

Για κάθε block του αρχείου (μη πυκνό ευρετήριο) η εγγραφή i του ευρετηρίου είναι της μορφής $\langle K(i), P(i) \rangle$ όπου:

- $K(i)$: η τιμή του πρωτεύοντος κλειδιού της πρώτης εγγραφής του block (*άγκυρα* του block)
- $P(i)$: δείκτης προς το block

• Το ευρετήριο στο πεδίο διάταξης (+ κλειδί) είναι ένα **μη πυκνό** ευρετήριο

Βάσεις Δεδομένων 2008-2009 Ευαγγελία Πλουρά 5

Πρωτεύον Ευρετήριο

Ποιο είναι το μέγεθος του ευρετηρίου (πόσα blocks);

Βάσεις Δεδομένων 2008-2009 Ευαγγελία Πλουρά 6



• Αναζήτηση

Διαδική αναζήτηση στο πρωτεύον ευρετήριο
Ανάγνωση του block από το αρχείο δεδομένων



• Εισαγωγή εγγραφής

αλλαγές και στο πρωτεύον ευρετήριο

μη διατεταγμένο αρχείο υπερχείλισης
συνδεδεμένη λίστα εγγραφών υπερχείλισης

• Διαγραφή εγγραφής

αλλαγές και στο πρωτεύον ευρετήριο
χρήση σημαδιών διαγραφής



Παράδειγμα (υπολογισμός μεγέθους αρχείου ευρετηρίου)

Έστω διατεταγμένο αρχείο με $r_A = 30.000$ εγγραφές, μέγεθος block $B = 1024$ bytes, σταθερού μεγέθους εγγραφές μεγέθους $R_A = 100$ bytes, όπου το πεδίο κλειδιού διάταξης έχει μέγεθος $V_A = 9$ bytes, μη εκτεινόμενη καταχώρηση.

Κατασκευάζουμε πρωτεύον ευρετήριο, μέγεθος δείκτη block $P = 6$ bytes

Μέγεθος αρχείου δεδομένων: 3.000 blocks

Μέγεθος αρχείου ευρετηρίου: 45 blocks



Παράδειγμα (υπολογισμός κόστους αναζήτησης)

Δεδομένα όπως πριν

(Έστω διατεταγμένο αρχείο με $r_A = 30.000$ εγγραφές, μέγεθος block $B = 1024$ bytes, σταθερού μεγέθους εγγραφές μεγέθους $R_A = 100$ bytes, όπου το πεδίο κλειδιού διάταξης έχει μέγεθος $V_A = 9$ bytes, μη εκτεινόμενη καταχώρηση. Κατασκευάζουμε πρωτεύον ευρετήριο, μέγεθος δείκτη block $P = 6$ bytes)

$$bfr_A = 10$$

$$bfr_E = 68$$

Μέγεθος αρχείου δεδομένων: 3.000 blocks

Μέγεθος αρχείου ευρετηρίου: 45 blocks

Αναζήτηση χωρίς ευρετήριο: $\lceil \log 3.000 \rceil = 12$ blocks

Αναζήτηση με ευρετήριο: $\lceil \log 45 \rceil + 1 = 7$ blocks

block ευρετηρίου block αρχείου

Διαδική γιατί το αρχείο ταξινομημένο



Access paths (μονοπάτια προσπέλασης)

- Το ευρετήριο αρχείου είναι ένα διατεταγμένο αρχείο με σταθερού μήκους εγγραφές
- Το αρχείο ευρετηρίου καταλαμβάνει μικρότερο χώρο από το ίδιο το αρχείο δεδομένων (οι καταχωρήσεις είναι μικρότερες και λιγότερες)
- Κάνοντας διαδική αναζήτηση στο ευρετήριο (γιατί το ευρετήριο είναι (πάντα) διατεταγμένο αρχείο) βρίσκουμε τον δείκτη στο block όπου αποθηκεύεται η εγγραφή που θέλουμε



Ευρετήριο συστάδων (clustering index): ορισμένο στο πεδίο διάταξης [το οποίο όμως δεν είναι κλειδί]

Υπάρχει μία εγγραφή για κάθε διακεκριμένη τιμή του πεδίου διάταξης (συστάδας) του αρχείου που περιέχει:

- την τιμή αυτή
- ένα δείκτη προς το πρώτο block του αρχείου δεδομένων που περιέχει μια εγγραφή με την τιμή αυτή στο πεδίο συστάδας
- Το ευρετήριο στο πεδίο διάταξης (+ όχι κλειδί) είναι ένα μη πυκνό ευρετήριο



Παράδειγμα (υπολογισμός μεγέθους ευρετηρίου)

Έστω διατεταγμένο αρχείο με $r_A = 30.000$ εγγραφές, μέγεθος block $B = 1024$ bytes, σταθερού μεγέθους εγγραφές μεγέθους $R_A = 100$ bytes, μη εκτεινόμενη καταχώρηση, όπου το πεδίο διάταξης έχει μέγεθος $V_A = 9$ bytes και υπάρχουν 1000 διαφορετικές τιμές και οι εγγραφές είναι ομοιόμορφα κατανομημένες ως προς τις τιμές αυτές. Υποθέτουμε ότι χρησιμοποιούνται άγκυρες block, κάθε νέα τιμή του πεδίου διάταξης αρχίζει στην αρχή ενός νέου block. Κατασκευάζουμε ευρετήριο συστάδων, μέγεθος δείκτη block $P = 6$ bytes

Μέγεθος αρχείου δεδομένων: 3.000 blocks
 Μέγεθος ευρετηρίου συστάδων: 15 blocks

$bfr_A = 10$
 $bfr_E = 68$



• Αναζήτηση (όπως πριν)

Διαδική αναζήτηση στο ευρετήριο
 Ανάγνωση blocks (τώρα μπορεί να είναι παραπάνω από ένα) από το αρχείο δεδομένων



Παράδειγμα (υπολογισμός κόστους αναζήτησης)

(στοιχεία όπως πριν) Έστω διατεταγμένο αρχείο με $r_A = 30.000$ εγγραφές, μέγεθος block $B = 1024$ bytes, σταθερού μεγέθους εγγραφές μεγέθους $R_A = 100$ bytes, μη εκτεινόμενη καταχώρηση, όπου το πεδίο διάταξης έχει μέγεθος $V_A = 9$ bytes και υπάρχουν 1000 διαφορετικές τιμές και οι εγγραφές είναι ομοιόμορφα κατανομημένες ως προς τις τιμές αυτές. Υποθέτουμε ότι χρησιμοποιούνται άγκυρες block, κάθε νέα τιμή του πεδίου διάταξης αρχίζει στην αρχή ενός νέου block. Κατασκευάζουμε ευρετήριο συστάδων, μέγεθος δείκτη block $P = 6$ bytes

Μέγεθος αρχείου δεδομένων: 3.000 blocks
 Μέγεθος αρχείου ευρετηρίου: 15 blocks

Αναζήτηση χωρίς ευρετήριο: $\lceil \log 3.000 \rceil + \text{ταιριάσματα} (= 3) \approx 15$ blocks
 Αναζήτηση με ευρετήριο: $\lceil \log 15 \rceil + 3 = 7$ blocks



Δευτερεύον ευρετήριο (secondary index): ορισμένο σε πεδίο διαφορετικό του πεδίου διάταξης



Περίπτωση 1: Το πεδίο ευρετηριοποίησης είναι κλειδί (καλείται και δευτερεύον κλειδί)

Υπάρχει μια εγγραφή για κάθε εγγραφή του αρχείου που περιέχει:

- την τιμή του κλειδιού για αυτήν την εγγραφή
- ένα δείκτη προς το block (ή την εγγραφή) του αρχείου δεδομένων που περιέχει την εγγραφή με την τιμή αυτή

- Το ευρετήριο σε πεδίο ΟΧΙ διάταξης (+ κλειδί) είναι ένα *πυκνό* ευρετήριο



Παράδειγμα (υπολογισμός μεγέθους ευρετηρίου)

Έστω αρχείο με $r_A = 30.000$ εγγραφές, μέγεθος block $B = 1024$ bytes, σταθερού μεγέθους εγγραφές μεγέθους $R_A = 100$ bytes, μη εκτεινόμενη καταχώρηση, όπου το πεδίο κλειδιού έχει μέγεθος $V_A = 9$ bytes αλλά δεν είναι πεδίο διάταξης. Κατασκευάζουμε δευτερεύον ευρετήριο, μέγεθος δείκτη block $P = 6$ bytes

Μέγεθος αρχείου δεδομένων: 3.000 blocks

Μέγεθος αρχείου ευρετηρίου: 442 blocks

45 για πρωτεύον

Παράδειγμα (υπολογισμός κόστους αναζήτησης)

Στοιχεία όπως πριν

(Έστω αρχείο με $r_A = 30.000$ εγγραφές, μέγεθος block $B = 1024$ bytes, σταθερού μεγέθους εγγραφές μεγέθους $R_A = 100$ bytes, μη εκτεινόμενη καταχώρηση, όπου το πεδίο κλειδιού έχει μέγεθος $V_A = 9$ bytes αλλά δεν είναι πεδίο διάταξης. Κατασκευάζουμε δευτερεύον ευρετήριο, μέγεθος δείκτη block $P = 6$ bytes)

Μέγεθος αρχείου δεδομένων: 3.000 blocks
 Μέγεθος αρχείου ευρετηρίου: 442 blocks

$$bfr_A = 10$$

$$bfr_E = 68$$

Αναζήτηση χωρίς ευρετήριο (σειριακή αναζήτηση, γιατί το αρχείο δεδομένων δεν είναι ταξινομημένο): $3.000/2 = 1500$ blocks

Αναζήτηση με ευρετήριο: $\lceil \log 442 \rceil + 1 = 10$ blocks

Για πρωτεύον ήταν 45 και 7 blocks αντίστοιχα

Περίπτωση 2: Το πεδίο ευρετηριοποίησης δεν είναι κλειδί

1. Πυκνό ευρετήριο: μία καταχώρηση για κάθε εγγραφή
2. Μεταβλητού μήκους εγγραφές με ένα επαναλαμβανόμενο πεδίο για το δείκτη
3. Μία εγγραφή ευρετηρίου για κάθε τιμή του πεδίου ευρετηριοποίησης + ένα ενδιάμεσο επίπεδο για την διαχείριση των πολλαπλών δεικτών

Αναζήτηση

Διαδική αναζήτηση στο δευτερεύον ευρετήριο
 Ανάγνωση του block (ή των blocks) από το ενδιάμεσο επίπεδο
 Ανάγνωση των blocks (συνήθως τόσα όσες οι εγγραφές που ταιριάζουν) από το αρχείο δεδομένων

Εισαγωγή

Απλή αν δεν αφορά εισαγωγή νέας τιμής στο ευρετήριο

Παράδειγμα (υπολογισμός μεγέθους ευρετηρίου)

Έστω μη διατεταγμένο αρχείο (αρχείο σωρού) με $r_A = 30.000$ εγγραφές, μέγεθος block $B = 1024$ bytes, σταθερού μεγέθους εγγραφές μεγέθους $R_A = 100$ bytes, μη εκτεινόμενη καταχώρηση, όπου το πεδίο ευρετηριοποίησης (δηλαδή, το πεδίο στο οποίο θα κατασκευάσουμε το ευρετήριο) έχει μέγεθος $V_A = 9$ bytes. Υπάρχουν 100 διαφορετικές τιμές και οι εγγραφές είναι ομοιόμορφα κατανεμημένες ως προς τις τιμές αυτές. Κατασκευάζουμε ευρετήριο συστάδων χρησιμοποιώντας την επιλογή (3), μέγεθος δείκτη block $P = 6$ bytes

Ευρετήριο $bfr_E = 68$ $b_E = 2$ κόστος αναζήτησης;
 Ενδιάμεσο επίπεδο -- Ποια είναι η οργάνωσή του;
 $bfr_{EE} = 170$ $b_{EE} = 177$ blocks

• Επιπρόσθετες δομές για την πιο αποδοτική εκτέλεση ερωτήσεων/αναζητήσεων - προκαλούν όμως επιβάρυνση στις τροποποιήσεις

• Εύκολη η λογική διάταξη των εγγραφών με βάση το πεδίο ευρετηριοποίησης

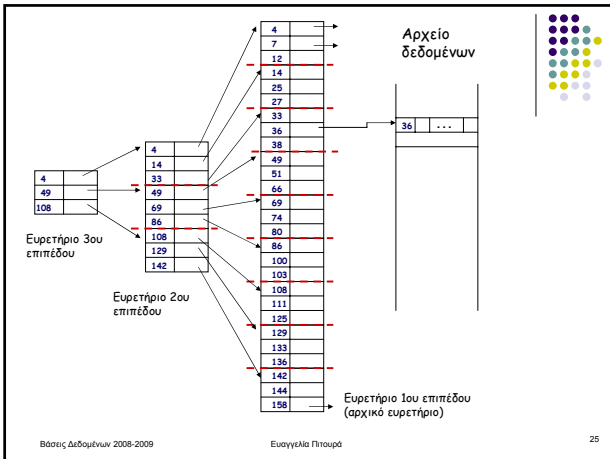
• Ανακτήσεις με σύνθετες συνθήκες, μπορεί να γίνουν χρησιμοποιώντας τα blocks του ευρετηρίου

Ιδέα:

Τα ευρετήρια είναι αρχεία - χτίζουμε ευρετήρια πάνω στα αρχεία ευρετηρίου

Το αρχείο είναι διατεταγμένο και το πεδίο διάταξης είναι και κλειδί (δρα πρωτεύον ευρετήριο!)

Υπενθύμιση (παράγοντας ομαδοποίησης: αριθμός εγγραφών ανά block)
 Παράγοντας ομαδοποίησης (blocking factor), όταν $B \geq R$
 $bfr = \lfloor (B / R) \rfloor$, όπου B μέγεθος block σε byte και R μέγεθος εγγραφής σε bytes



Ευρετήριο Πολλών Επιπέδων

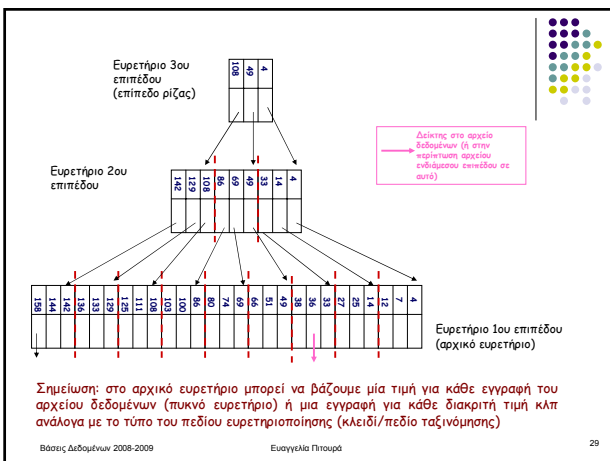
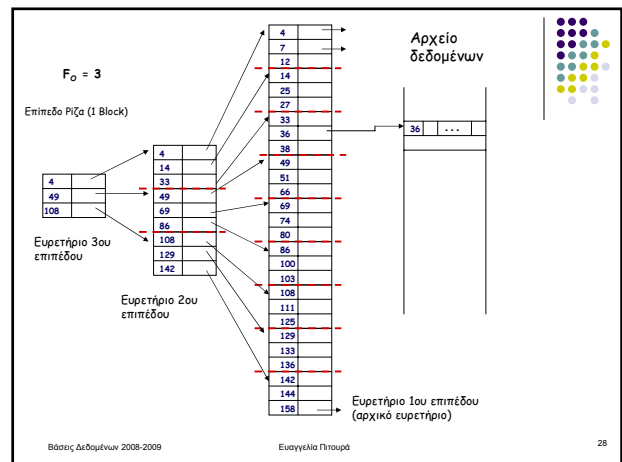
- Έστω ότι το αρχείο ευρετηρίου είναι το **πρώτο ή βασικό επίπεδο**. Έστω ότι ο παράγοντας ομαδοποίησης είναι f_0 και ότι έχει r_1 blocks. Το αρχείο είναι διατεταγμένο και το πεδίο διάταξης είναι και κλειδί.
- Δημιουργούμε ένα πρωτεύον ευρετήριο για το ευρετήριο πρώτου επιπέδου - **δεύτερο επίπεδο**.
Παράγοντας ομαδοποίησης: f_0 Αριθμός block $\lceil (r_1/f_0) \rceil$
- Δημιουργούμε ένα πρωτεύον ευρετήριο για το ευρετήριο δεύτερου επιπέδου - **τρίτο επίπεδο**.
Παράγοντας ομαδοποίησης: f_0 Αριθμός block $\lceil (r_1/(f_0)^2) \rceil$

Βάσεις Δεδομένων 2008-2009 Ευαγγελία Πιτουρά 26

Ευρετήριο Πολλών Επιπέδων

- Μέχρι πόσα επίπεδα:
Μέχρι όλες οι εγγραφές του ευρετηρίου να χωρούν σε ένα block.
Έστω t κορυφαίο επίπεδο $\lceil (r_1/(f_0)^t) \rceil = 1$ (top level)
- Το f_0 ονομάζεται και **παράγοντας διακλάδωσης** του ευρετηρίου.

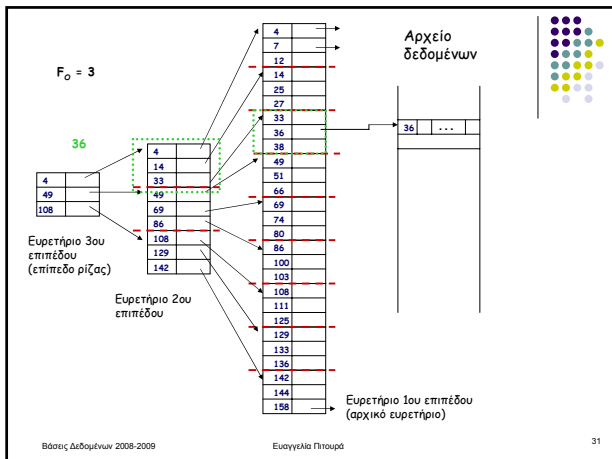
Βάσεις Δεδομένων 2008-2009 Ευαγγελία Πιτουρά 27



Ευρετήριο Πολλών Επιπέδων

- Αναζήτηση**
 $p :=$ διεύθυνση του block του κορυφαίου επιπέδου του ευρετηρίου
 $t :=$ αριθμός επιπέδων του ευρετηρίου
 for $j = t$ to 1 step -1 do
 read block με διεύθυνση p του ευρετηρίου στο επίπεδο j
 αναζήτηση στο block p της εγγραφής i με τιμή $K_j(i) \leq K < K_j(i+1)$
 read το block του αρχείου δεδομένων με διεύθυνση p
 Αναζήτηση στο block p της εγγραφής i με τιμή $K_j(i) \leq K < K_j(i+1)$

Βάσεις Δεδομένων 2008-2009 Ευαγγελία Πιτουρά 30



Ευρετήριο Πολλών Επιπέδων

- Εισαγωγή/διαγραφή

τροποποιήσεις πολλαπλών ευρετηρίων

Δυναμικό πολυεπίπεδο ευρετήριο: Β-δέντρα και Β+δέντρα

Ευρετήριο Πολλών Επιπέδων

Παράδειγμα (υπολογισμός μεγέθους ευρετηρίου)

Έστω αρχείο με $r_A = 30.000$ εγγραφές, μέγεθος block $B = 1024$ bytes, σταθερού μεγέθους εγγραφές μεγέθους $R_A = 100$ bytes, μη εκτεινόμενη καταχώρηση, όπου το πεδίο κλειδιού έχει μέγεθος $V_A = 9$ bytes αλλά δεν είναι πεδίο διάταξης. Κατασκευάζουμε δευτερεύον ευρετήριο, μέγεθος δείκτη block $P = 6$ bytes

$$f_0 = \lfloor (1024 / (9 + 6)) \rfloor = 68$$

Μέγεθος αρχείου δεδομένων: 3.000 blocks
 Μέγεθος αρχείου ευρετηρίου *πρώτου* επιπέδου: 442 blocks
 Μέγεθος αρχείου ευρετηρίου *δευτέρου* επιπέδου: $\lceil (442 / 68) \rceil = 7$ blocks
 Μέγεθος αρχείου ευρετηρίου *τρίτου* επιπέδου: $\lceil (7 / 68) \rceil = 1$ block

Άρα $t = 3$

Ευρετήριο Πολλών Επιπέδων

Παράδειγμα (υπολογισμός κόστους αναζήτησης)

Έστω αρχείο με $r_A = 30.000$ εγγραφές, μέγεθος block $B = 1024$ bytes, σταθερού μεγέθους εγγραφές μεγέθους $R_A = 100$ bytes, μη εκτεινόμενη καταχώρηση, όπου το πεδίο κλειδιού έχει μέγεθος $V_A = 9$ bytes αλλά δεν είναι πεδίο διάταξης. Κατασκευάζουμε δευτερεύον ευρετήριο, μέγεθος δείκτη block $P = 6$ bytes

Άρα $t = 3$

Παράδειγμα
 $t + 1 = 4$ προσπελάσεις
 Για το δευτερεύον ήταν 10 και χωρίς ευρετήριο 1500

Δεντρικά Ευρετήρια

- Τα αρχεία ευρετηρίων είναι απλά αρχεία, άρα και σε αυτά μπορούν να οριστούν ευρετήρια
- Καταλήγουμε λοιπόν σε μια ιεραρχία δομών ευρετηρίων (πρώτο επίπεδο, δεύτερο επίπεδο, κλπ.)
- Κάθε επίπεδο του ευρετηρίου είναι ένα *διατεταγμένο* αρχείο, συνεπώς, εισαγωγές/διαγραφές εγγραφών απαιτούν επιπλέον δουλειά
- Ένα πολύ-επίπεδο ευρετήριο αποτελεί ένα *Δέντρο Αναζήτησης*
 Όπου κάθε κόμβος (block) έχει f_0 δείκτες και f_0 τιμές κλειδιού

Δέντρα Αναζήτησης

Ένα *δέντρο αναζήτησης* (search tree) τάξεως p είναι ένα δέντρο τέτοιο ώστε κάθε κόμβος του περιέχει το πολύ $p - 1$ τιμές αναζήτησης και p δείκτες ως εξής

Συμβολισμός K_j^*

$K_1 < K_2 < \dots < K_{q-1}$ και για όλες τις τιμές X στα υποδέντρα ισχύει $K_{j-1} < X < K_j$ για $1 < j < p$, $X < K_j$ για $j = 1$, και $K_{p-1} < X$ για $j = p$



Κάθε κόμβος του δέντρου είναι ένα block στο δίσκο

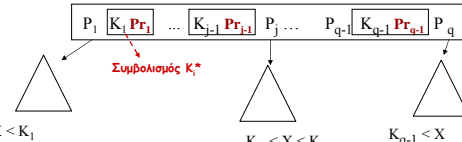
Ισοζυγισμένο: όλοι οι κόμβοι-φύλλα στο ίδιο επίπεδο

Β-δέντρο: ένα δέντρο αναζήτησης που παραμένει ισοζυγισμένο και χωρίς «πολύ αδειανούς» κόμβους



Ένα Β-δέντρο τάξεως (order) p ορίζεται ως εξής:

- Κάθε εσωτερικός κόμβος είναι της μορφής $\langle P_1, \langle K_1, Pr_1 \rangle, P_2, \langle K_2, Pr_2 \rangle, \dots, \langle K_{q-1}, Pr_{q-1} \rangle, P_q \rangle$, $q < p$, όπου P_i δείκτης δέντρου, K_i τιμή αναζήτησης, Pr_i δείκτης δεδομένων



- Σε κάθε κόμβο $K_1 < K_2 < \dots < K_{q-1}$
- Για όλες τις τιμές X στο υποδέντρο που δείχνει το P_j ισχύει $K_{j-1} < X < K_j$ για $1 < j < q$, $X < K_j$ για $j=1$, και $K_{j-1} < X$ για $j=q$



- Κάθε κόμβος έχει το πολύ p δείκτες δέντρου

- Κάθε κόμβος εκτός της ρίζας και των φύλλων έχει τουλάχιστον $\lceil p/2 \rceil$ δείκτες. Η ρίζα έχει τουλάχιστον 2 εκτός αν είναι ο μόνος κόμβος του δέντρου.

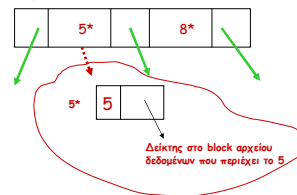
- Ένας κόμβος με q δείκτες δέντρου περιέχει $q - 1$ τιμές πεδίου αναζήτησης (και άρα και $q - 1$ δείκτες δεδομένων).

- Όλα τα φύλλα βρίσκονται στο ίδιο επίπεδο. Τα φύλλα έχουν την ίδια δομή εκτός του ότι οι δείκτες δέντρου είναι null.



τάξη $p = 3$ (2 τιμές ανά κόμβο, 3 δείκτες block ευρετηρίου)

Δείκτης σε block ευρετηρίου (null για κόμβους φύλλα)



Αναζήτηση

Διαβάζουμε το block της ρίζας

Αν η εγγραφή δεν υπάρχει στο κόμβο διαβάζουμε το αντίστοιχο block στο επόμενο πεδίο



Εισαγωγή τιμής

Αρχικά ένας μόνο κόμβος (ρίζα) στο Επίπεδο 0

Όταν ο κόμβος ρίζα γεμίσει ($p - 1$ τιμές κλειδιού), νέα εισαγωγή οδηγεί στην διάσπαση του κόμβου σε δύο κόμβους στο Επίπεδο 1: η μεσαία τιμή μένει στη ρίζα, οι υπόλοιπες μοιράζονται εξίσου σε δύο κόμβους του Επίπεδου 1

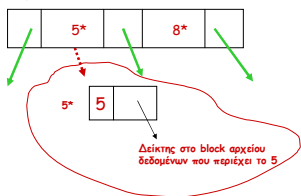
Όταν ένας κόμβος εκτός της ρίζας γεμίσει, νέα εισαγωγή οδηγεί σε διάσπαση του κόμβου σε δύο κόμβους στο ίδιο επίπεδο και μεταφορά της μεσαίας τιμής στον γονέα του κόμβου

ΠΡΟΣΟΧΗ: η εισαγωγή της μεσαίας τιμής στο γονέα αν ο γονέας είναι γεμάτος μπορεί να οδηγήσει σε διάσπαση του γονέα. Η διάσπαση μπορεί να οδηγήσει ως τη ρίζα, οπότε δημιουργείται και νέο επίπεδο.

B-δέντρα (παράδειγμα)

τάξη $p = 3$ (2 τιμές ανά κόμβο, 3 δείκτες block ευρετηρίου) - Εισαγωγή 5, 8, 7, 14, 19, 6, 10

Δείκτης σε block ευρετηρίου (null για κόμβους φύλλα)



B-δέντρα

Διαγραφή τιμής

Τιμή προς διαγραφή ανήκει σε *φύλλο* → ok

Τιμή προς διαγραφή ανήκει σε *εσωτερικό κόμβο* →

Αν σβήσουμε το K_i , τότε το **μικρότερο κλειδί** του υποδέντρου P_{i+1} πρέπει να το **αντικαταστήσει** (δηλαδή το μικρότερο κλειδί του κόμβου στα δεξιά του κλειδιού που διαγράφεται)

B-δέντρα

Διαγραφή τιμής

Αν υποχρέωση

αν είναι δυνατόν **ανακατανομή** με τον αριστερό αδελφό
 αν όχι, προσπάθεια ανακατανομής με το δεξιό αδελφό
 αν όχι ανακατανομή, **συγχώνευση** και των τριών κόμβων σε δύο κόμβους
 σε περίπτωση συγχώνευσης: διαγράφουμε και την αντίστοιχη εγγραφή στον γονέα (πιθανή υποχρέωση και στο γονέα)

▪ Σε κάθε περίπτωση (ανακατανομή και συγχώνευση) κατεβάζουμε και την τιμή του γονέα - και στο γονέα ανεβαίνει η νέα μεσαία τιμή

B-δέντρα

▪ Κάθε κόμβος του B-δέντρου καταλαμβάνει μια σελίδα (block)

Υπολογισμός Τάξης p (ώστε κάθε κόμβος να χωρά σε ένα block)

Έστω B μέγεθος block, V μέγεθος πεδίου αναζήτησης (δηλαδή του πεδίου ευρετηριοποίησης), Pr μέγεθος δείκτη δεδομένων (εγγραφής) και P μέγεθος δείκτη δέντρου (block)

$$p * P + (p - 1) * (Pr + V) \leq B$$

$$p * (P + Pr + V) \leq B + V + Pr$$

$$p \leq (B + V + Pr) / (P + Pr + V)$$

Παράδειγμα, $V = 9$ bytes, $B = 512$ bytes, $Pr = 7$ bytes, $P = 6$ bytes, τότε $p = 23$

B-δέντρα

Υπολογισμός επιπέδων

Έστω όπως πριν, $p = 23$. Έστω ότι κάθε κόμβος είναι γεμάτος κατά 69%.

Πόσα επίπεδα χρειαζόμαστε για να ευρετηριοποιήσουμε 65.000 τιμές;

$(p - 1) * 0,69 = 22 * 0,69 = 15$ κλειδιά και $15 + 1 = 16$ δείκτες ανά κόμβο

	#κόμβων	#τιμές	#δείκτες
Ρίζα	1 κόμβος	15 ($22 * 0,69$) καταχωρήσεις	16 δείκτες
Επίπεδο 1:	16 κόμβοι	240 ($16 * 15$) καταχωρήσεις	256 δείκτες
Επίπεδο 2:	256 κόμβοι	3.840 ($256 * 15$) καταχωρήσεις	4.096 δείκτες
Επίπεδο 3:	4.096 κόμβοι	61.440	

Σύνολο: $61.440 + 3.840 + 240 + 15$ (65.535)

B-δέντρα

Ποιες εγγραφές βάζουμε στο B-δέντρο

(πεδίο διάταξης - κλειδί, πεδίο διάταξης - όχι κλειδί, όχι πεδίο διάταξης - κλειδί, όχι πεδίο διάταξης - όχι κλειδί)

Αναζήτηση διαστήματος τιμών



Παράδειγμα (υπολογισμός μεγέθους ευρετηρίου)

Έστω διατεταγμένο αρχείο με $r_A = 300.000$ εγγραφές, μέγεθος block $B = 1024$ bytes, σταθερού μεγέθους εγγραφές μεγέθους $R_A = 100$ bytes, μη εκτεινόμενη καταχώρηση. Κατασκευάζουμε ευρετήριο B-δέντρο πάνω στο πεδίο διάταξης που είναι και κλειδί. Το πεδίο διάταξης έχει μέγεθος $V_A = 9$ bytes.

Μέγεθος δείκτη block (και για το αρχείο δεδομένων και για τους κόμβους του ευρετηρίου είναι) $P = 6$ bytes

Ποιο είναι ο (ελάχιστος) αριθμός επιπέδων για ένα B-δέντρο
Τι μπορείτε να πείτε για το μέγεθος του (σε αριθμό block);



Διαφορά B+ από B-δέντρο: Αποθηκεύουμε δείκτες δεδομένων (στο αρχείο δεδομένων) μόνο στα φύλλα

Δύο τύποι κόμβων:

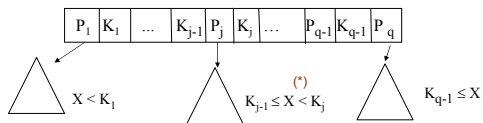
- εσωτερικοί κόμβοι
- φύλλα

Όλες οι τιμές του πεδίου αναζήτησης εμφανίζονται στα φύλλα.
Κάποιες τιμές μπορεί να εμφανίζονται *παραπάνω από μια φορά*



Ένα B+ -δέντρο τάξεως (order) p ορίζεται ως εξής:

1. Κάθε **εσωτερικός κόμβος** είναι της μορφής $\langle P_1, K_1, P_2, K_2, \dots, K_{q-1}, P_q, K_q \rangle$ $q \leq p$, όπου P_i δείκτης δέντρου, K_i τιμή αναζήτησης



2. Σε κάθε εσωτερικό κόμβο $K_1 < K_2 < \dots < K_{q-1}$
3. Για όλες τις τιμές X στο υποδέντρο που δείχνει το P_j ισχύει $K_{j-1} \leq X < K_j$ για $1 < j < q$, $X < K_j$ για $j=1$, και $K_{j-1} \leq X$ για $j=q$

(*) σύμβαση, θα μπορούσε και $K_{j-1} < X < K_j$



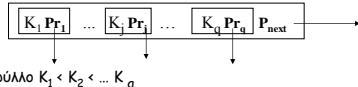
4. Κάθε εσωτερικός κόμβος έχει το **πολύ p** δείκτες δέντρου
5. Κάθε εσωτερικός κόμβος **εκτός της ρίζας** έχει **τουλάχιστον $\lceil p/2 \rceil$** . Η ρίζα έχει τουλάχιστον 2 εκτός αν είναι ο μόνος κόμβος του δέντρου.
6. Ένας κόμβος με q δείκτες δέντρου περιέχει q - 1 τιμές πεδίου αναζήτησης



1. Κάθε **κόμβος-φύλλο** είναι της μορφής $\langle \langle K_1, P_{r_1} \rangle, \langle K_2, P_{r_2} \rangle, \dots, \langle K_q, P_{r_q} \rangle, P_{next} \rangle$, $q \leq p_{leaf}$, όπου p_{leaf} είναι η τάξη των κόμβων-φύλλων K_i τιμή αναζήτησης,

P_{r_i} δείκτης δεδομένων που δείχνει στο block (ή στην εγγραφή) με τιμή στο πεδίο αναζήτησης K_i (ή σε ένα block ενδιάμεσου επιπέδου αν το πεδίο αναζήτησης δεν είναι κλειδί),

P_{next} δείχνει στο επόμενο φύλλο και χρησιμοποιείται για τη γρήγορη ανάγνωση του αρχείου σε διάταξη



2. Σε κάθε κόμβο-φύλλο $K_1 < K_2 < \dots < K_q$

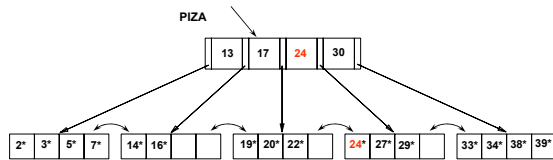


3. Κάθε κόμβος-φύλλο έχει το **πολύ p_{leaf}** τιμές
4. Κάθε κόμβος-φύλλο έχει **τουλάχιστον $\lceil p_{leaf}/2 \rceil$** τιμές.
5. Όλοι οι κόμβοι-φύλλα βρίσκονται στο ίδιο επίπεδο.

Β+-δέντρα

Η αναζήτηση ξεκινά από τη ρίζα, και οι συγκρίσεις των κλειδιών μας οδηγούν στα φύλλα

Αναζήτηση για τα 5*, 15*, όλες οι καταχωρήσεις $\geq 24^*$...



Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

55

Β+-δέντρα: Αναζήτηση

Nodepointer tree_search(nodepointer P, keyvalue K)

if P is a leaf return(P);

else

if $K < K_1$

tree_search(P_1 , K)

else

find i such that $K_i \leq K < K_{i+1}$

return tree_search(P_i , K)

end

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

56

Β+-δέντρα: Αναζήτηση

Αναζήτηση (αναδρομική εκδοχή)

nodepointer find(keyvalue K):

return tree_search(root, K);

end;

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

57

Β+-δέντρα: Εισαγωγή

Εισαγωγή

1. Αναζήτηση του φύλλου για εισαγωγή: έστω φύλλο P

2. Εισαγωγή τιμής K στο κόμβο P

Αν ο κόμβος-φύλλο δεν είναι γεμάτος
εισαγωγή της τιμής

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

58

Β+-δέντρα: Εισαγωγή

Αν ο κόμβος-φύλλο είναι γεμάτος (έχει p_{leaf} εγγραφές)

διάσπαση του κόμβου:

- οι πρώτες $k = \lfloor (p_{leaf} + 1) / 2 \rfloor$ παραμένουν στον κόμβο
- οι υπόλοιπες σε καινούργιο κόμβο
- εισαγωγή (αντιγραφή) της $k+1$ -οστής τιμής (K_{k+1}) στο γονέα

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

59

Β+-δέντρα: Εισαγωγή

Αν ένας εσωτερικός κόμβος είναι γεμάτος (έχει p εγγραφές)

διάσπαση του κόμβου: έστω $k = \lfloor ((p+1)/2) \rfloor$

- οι εγγραφές μέχρι το P_k (μετά την εισαγωγή) παραμένουν στον κόμβο
- η $k+1$ -οστή K_{k+1} τιμή **μεταφέρεται (δεν αντιγράφεται)** στον πατέρα
- οι υπόλοιπες σε καινούργιο κόμβο

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

60

Β+-δέντρα: Εισαγωγή

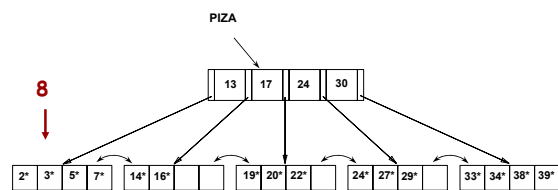
Οι διασπάσεις κόμβων (εκτός ρίζας) "μεγαλώνουν" το δέντρο
 Η διάσπαση της ρίζας "υψώνει" το δέντρο

Β+-δέντρα (παράδειγμα)

5, 8, 7, 14, 19, 6, 10 και τάξη $p = 3$ (2 τιμές ανά κόμβο, 3 δείκτες block ευρητηρίου)
 και $p_{leaf} = 2$

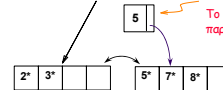
Β+-δέντρα: Εισαγωγή

Εισαγωγή της καταχώρησης 8*



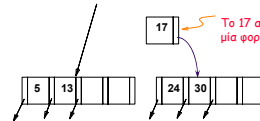
Β+-δέντρα: Εισαγωγή

Καταχώρηση στον κόμβο γονέα (αντιγραφή)



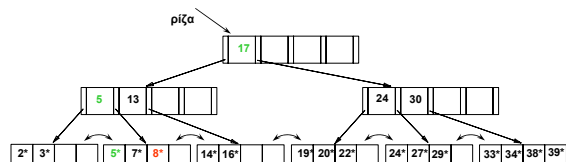
Το 5 ανεβαίνει επάνω, αλλά παραμένει και στο φύλλο

Καταχώρηση στον κόμβο γονέα (μεταφορά)



Το 17 ανεβαίνει επάνω και παρουσιάζεται μόνο μία φορά στο ευρητήριο (σε αντίθεση με τα φύλλα)

Β+-δέντρα: Εισαγωγή



Η ρίζα διασπάστηκε οδηγώντας σε αύξηση του ύψους.

Β+-δέντρα

Όλες οι τιμές εμφανίζονται στα φύλλα και κάποιες επαναλαμβάνονται και σε εσωτερικούς κόμβους (η τιμή K σε ένα εσωτερικό κόμβο μπορεί επίσης να εμφανίζεται ως η πιο αριστερή τιμή στο φύλλο του υποδέντρου με ρίζα το δείκτη στα δεξιά του K)



Διαγραφή

1. Αναζήτηση του φύλλου που περιέχει το K: έστω φύλλο P
2. Αν υποχείλιση
 - αν είναι δυνατόν ανακατανομή με τον αριστερό αδελφό ($\lceil n/2 \rceil$)
 - αν όχι, προσπάθεια ανακατανομής με το δεξιό αδελφό
 - αν όχι, συγχώνευση και των τριών κόμβων σε δύο κόμβους



2. Αν υποχείλιση (αναλυτικά)

<ανακατανομή εγγραφών>

Αν είναι δυνατόν ανακατανομή με τον αριστερό αδελφό ($\lceil n/2 \rceil$)
αν όχι, προσπάθεια ανακατανομής με το δεξιό αδελφό

ανακατανομή εγγραφών σε κάθε κόμβο

βρείτε την εγγραφή στο γονέα του δεξιού κόμβου N

αντικατάσταση της τιμής κλειδιού στο γονέα τους με τη μικρότερη τιμή του κόμβου N

<συγχώνευση κόμβων>

Αν δεν είναι δυνατή η ανακατανομή

συγχώνευση κόμβων

οδηγεί σε διαγραφή στο παραπάνω επίπεδο, **σφηνεται** η εγγραφή που δείχνει στον κόμβο (πιθανότητα νέας υποχείλισης)



Εσωτερικοί κόμβοι

Ειδική περίπτωση στη συγχώνευση εσωτερικών κόμβων, όταν συγχωνεύεται ο ακραίος αριστερός δείκτης ενός εσωτερικού κόμβου (ο οποίος δεν έχει τιμή)

Τότε, πρέπει να συμβουλευτούμε τον γονέα των δύο κόμβων που συγχωνεύονται -> χρήση της τιμής του δείκτη που δείχνει σε αυτόν τον κόμβο

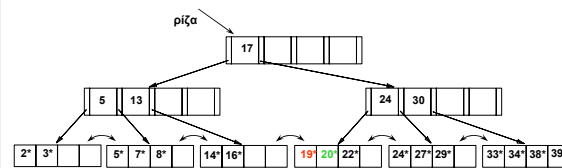
«Κατεβάζουμε» την τιμή από τον γονέα ως πιο αριστερή τιμή στον προς συγχώνευση κόμβο

Ειδικά για την ανακατανομή εσωτερικών κόμβων

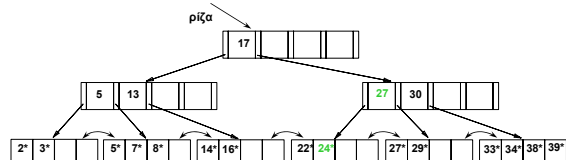
Πάει μέσω του γονέα τους

Δηλαδή θεωρούμε και την τιμή του γονέα στην ανακατανομή

Η τιμή αυτή αλλάζει στο γονέα



Διαγραφή 19, 20

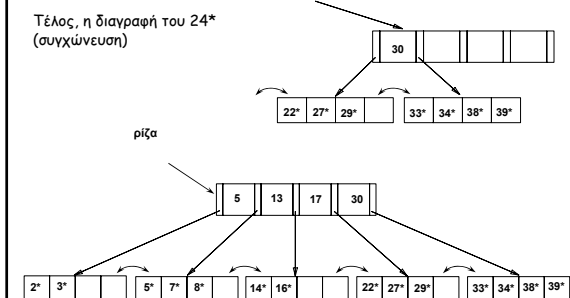


Το παράδειγμα μετά τη διαγραφή του 19* και του 20* (ανακατανομή με δεξιό αδελφό και αντικατάσταση του 24 με 27)

Διαγραφή του 24 ->



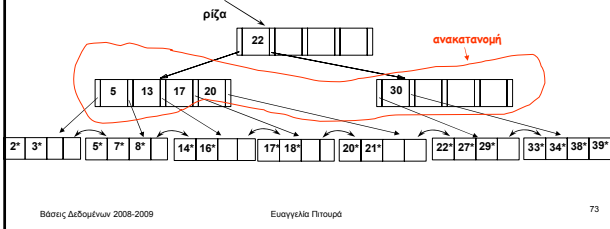
Τέλος, η διαγραφή του 24* (συγχώνευση)



Β+-δέντρα: Διαγραφή

Παράδειγμα ανακατανομής

Έστω στο παρακάτω δέντρο μετά από συγχώνευση φύλλων

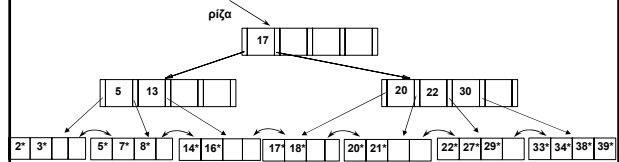


Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

73

Β+-δέντρα: Διαγραφή



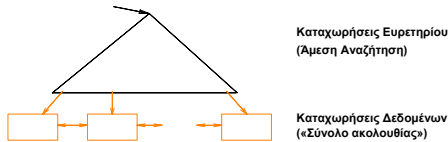
Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

74

Β+-δέντρα γενικά

- Εισαγωγή/Διαγραφή με κόστος $\log_p N$ --- κρατούν το δέντρο σε ισορροπημένη μορφή. ($F =$ διακλάδωση, $N =$ αριθμός των φύλλων)
- Ελάχιστη πληρότητα 50% (εκτός της ρίζας).
- Εξαιρετική δομή ΚΑΙ για ερωτήσεις ισότητας ΚΑΙ για ερωτήσεις διαστήματος (range queries).
- Το αρχείο δεδομένων μπορεί να είναι ή όχι ταξινομημένο



Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

75

Β+-δέντρα

- Κάθε κόμβος του Β+-δέντρου καταλαμβάνει μια σελίδα (block)

Τάξη p ώστε κάθε εσωτερικός-κόμβος να χωρά σε ένα block

Έστω B μέγεθος block, V μέγεθος πεδίου αναζήτησης, P_r μέγεθος δείκτη (εγγραφής) και P μέγεθος δείκτη δέντρου (block)

$$p * P + (p - 1) * V \leq B$$

$$p * (P + V) \leq B + V$$

$$p \leq (B + V) / (P + V)$$

Παράδειγμα, $V = 9$ bytes, $B = 512$, $P_r = 7$ bytes, $P = 6$ bytes, τότε $p = 34$

Για Β-δέντρο, $p = 23$

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

76

Β+-δέντρα

Τάξη p_{leaf} ώστε κάθε φύλλο να χωρά σε ένα block

Έστω B μέγεθος block, V μέγεθος πεδίου αναζήτησης, P_r μέγεθος δείκτη δεδομένων (εγγραφής) και P μέγεθος δείκτη δέντρου (block)

$$p_{leaf} * (P_r + V) + P \leq B$$

$$p_{leaf} * (P_r + V) \leq B - P$$

$$p_{leaf} \leq (B - P) / (P_r + V)$$

Παράδειγμα, $V = 9$ bytes, $B = 512$, $P_r = 7$ bytes, $P = 6$ bytes, τότε $p_{leaf} = 31$

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

77

Β+-δέντρα

Υπολογισμός επιπέδων

Παράδειγμα, $V = 9$ bytes, $B = 512$, $P_r = 7$ bytes, $P = 6$ bytes, τότε $p = 34$. Έστω ότι κάθε κόμβος είναι γεμάτος κατά 69%. Πόσες καταχωρήσεις (τιμές) χωρά αν έχει 3 επίπεδα

Ρίζα	1 κόμβος	22 (33*0,69) καταχωρήσεις	23 δείκτες
Επίπεδο 1:	23 κόμβοι	506 (23*22) καταχωρήσεις	529 δείκτες
Επίπεδο 2:	529 κόμβοι	11.638 (529*22) καταχωρήσεις	12.167 δείκτες
Επίπεδο φύλλων:	12.167 κόμβοι	255.507 (12.167 * 31 * 0.69) δείκτες	δεδομένων

Σε 3 επίπεδα 255.507 εγγραφές έναντι 65.535 για το Β-δέντρο

Σημείωση: εγγραφές μόνο στα φύλλα

Βάσεις Δεδομένων 2008-2009

Ευαγγελία Πιτουρά

78

Παρατηρήσεις

- Τυπική Τάξη: 100. Τυπικός Παράγοντας Πληρότητας: 67%.
- Μέση τιμή διακλάδωσης (fan out) = 133
- Τυπικές Δυνατότητες:
 - Ύψος 4: $133^4 = 312,900,700$ εγγραφές
 - Ύψος 3: $133^3 = 2,352,637$ εγγραφές
- Μπορεί να κρατά τα υψηλότερα επίπεδα στη μνήμη (buffer):
 - Επίπεδο 1 = 1 block = 8 Kbytes
 - Επίπεδο 2 = 133 blocks = 1 Mbyte
 - Επίπεδο 3 = 17,689 blocks = 133 Mbytes

Είδη Ευρετηρίων

- Ευρετήριο ενός επιπέδου ένα διατεταγμένο αρχείο με εγγραφές $\langle K(i), P(i) \rangle$
- Ευρετήριο πολλών επιπέδων
- Ευρετήρια δομής δέντρου (B-δέντρα, B+-δέντρα)
- Ευρετήρια κατακερματισμού

h(τιμή) → στο κάδο οι εγγραφές είναι εγγραφές ευρετηρίου, δηλαδή ζεύγη (τιμή, δείκτης-στο-block(ς)-του-αρχείου-δεδομένων που-είναι-η-εγγραφή-με-αυτήν-την-τιμή)

Ορισμοί

Πρωτεύον: όταν το πεδίο ευρετηριοποίησης είναι πρωτεύον κλειδί και πεδίο διάταξης του αρχείου

Δευτερεύον: αλλιώς

Συστάδων (clustered index) αν η διάταξη των εγγραφών στο ευρετήριο όμοια ή παρόμοια αυτής των εγγραφών στο αρχείο δεδομένων (συμβαίνει, πχ όταν το ευρετήριο κτίζεται στο πεδίο ταξινόμησης του αρχείου δεδομένων)

Το πολύ ένα ευρετήριο συστάδων - δηλαδή ένα ευρετήριο στο πεδίο διάταξης του αρχείου

Range scan (αναζήτηση περιοχής)

- Συστάδων: #σελίδων στο αρχείο που ταιριάζουν
- Μη συστάδων: αριθμός εγγραφών στο ευρετήριο που ταιριάζουν - για κάθε τέτοια εγγραφή → μια σελίδα αρχείου

Οι εγγραφές στο ευρετήριο K*

1. Η πραγματική εγγραφή (πλειάδα) με τιμή K στο κλειδί οργάνωση αρχείου
2. $\langle K, rid \rangle$
3. $\langle K, list\ of\ rids \rangle$

Ανεξάρτητα του ΣΔΒΔ

Μικρόκοσμος

Συλλογή Απαιτήσεων και Ανάλυση

Απαιτήσεις ΒΔ

Εννοιολογικός Σχεδιασμός Βάσης

Εννοιολογικό Μοντέλο (Σχήμα)

Εξαρτώμενο του επιλεγμένου ΣΔΒΔ

(π.χ., με Σχεσιακό Μοντέλο)

Λογικό Μοντέλο -- Σχήματα/Όψεις

Λογικός Σχεδιασμός Βάσης

Φυσικός Σχεδιασμός Βάσης

Φυσικό Μοντέλο Εσωτερικό Σχήμα

Πλήρωση Βάσης

Βάση Δεδομένων

Διάγραμμα Ο/Σ



- Μετά τον σχεδιασμό Ο/Σ και το λογικό σχεδιασμό (σχεσιακό μοντέλο), έχουμε τα εννοιολογικά και λογικά (με τις όψεις) σχήματα για τη Βάση Δεδομένων.
- Το επόμενο βήμα είναι ο **Φυσικός Σχεδιασμός**, δηλαδή η επιλογή των δομών αποθήκευσης των σχέσεων, η επιλογή των ευρετηρίων, οι αποφάσεις για συστάδες - γενικά ότι είναι απαραίτητο για να επιτευχθούν οι προσδοκώμενες επιδόσεις χρήσης της ΒΔ.
- Η υλοποίηση μιας (φυσικής) Σχεσιακής Βάσης Δεδομένων περιλαμβάνει τη δημιουργία ΚΑΤΑΛΟΓΩΝ ΣΥΣΤΗΜΑΤΟΣ (directory system tables)



Η SQL-92 δεν περιλαμβάνει εντολές για τη δημιουργία ευρετηρίων. Τα περισσότερα εμπορικά ΣΔΒΔ το υποστηρίζουν

```
create [unique] index <index_name>
on <table_name> (<attr_list>);
```

- Η <attr_list> μπορεί να περιέχει παραπάνω από ένα γνωρίσματα.
- Προαιρετικό UNIQUE σημαίνει ότι το <attr_list> είναι κλειδί του <table_name>.



```
drop index <index_name>
```

- Η Oracle δημιουργεί αυτόματα ευρετήρια για κάθε UNIQUE ή PRIMARY KEY ορισμό.

```
select <index_name> from user_indexes
```



Για να κάνουμε όσο το δυνατόν καλύτερο τον Φυσικό Σχεδιασμό πρέπει να :

Κατανοήσουμε το **Φόρτο Εργασίας (workload)**

Ποιες είναι οι σημαντικές ερωτήσεις και πόσο συχνά εμφανίζονται.

Ποιες είναι οι πιο σημαντικές τροποποιήσεις και πόσο συχνά εμφανίζονται.

Ποια είναι η επιθυμητή επίδοση για την εκτέλεση αυτών των ερωτήσεων και τροποποιήσεων.



Πριν δημιουργήσουμε ένα ευρετήριο, πρέπει να συνηγορήσουμε και την επίδρασή του σε ενημερώσεις του φορτίου εργασίας!

Ένα ευρετήριο κάνει τις ερωτήσεις ΠΙΟ ΓΡΗΓΟΡΕΣ και τις ενημερώσεις ΠΙΟ ΑΡΓΕΣ

Επιπλέον, απαιτεί και χώρο στον δίσκο



Για κάθε ερώτηση (query) το φόρτο εργασίας:

Σε ποιες σχέσεις έχει πρόσβαση?

Ποια γνωρίσματα ανακαλεί?

Ποια γνωρίσματα υπεισέρχονται στις συνθήκες για selection/join? Πόσο επιλεκτικές είναι αυτές οι συνθήκες?

Για κάθε ενημέρωση (insert/delete/update):

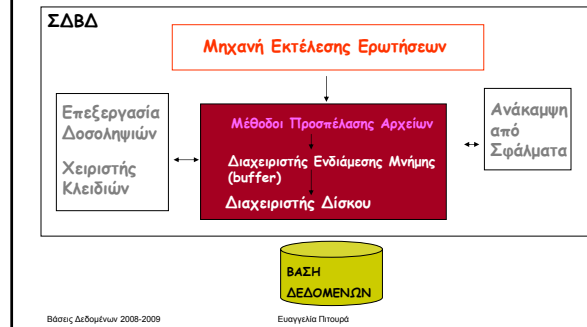
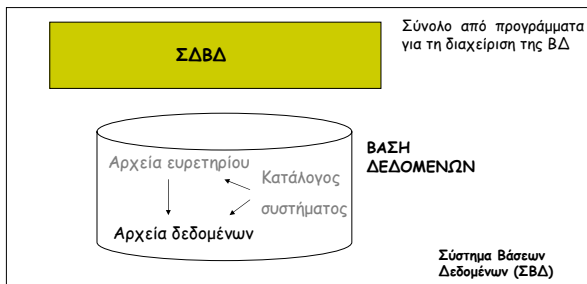
Ποια γνωρίσματα υπεισέρχονται στις συνθήκες για selection/join? Πόσο επιλεκτικές είναι αυτές οι συνθήκες?

Ο τύπος της ενημέρωσης (INSERT/DELETE/UPDATE), και τα γνωρίσματα που θα επηρεασθούν

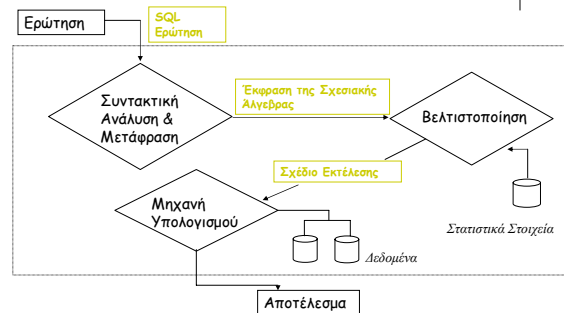
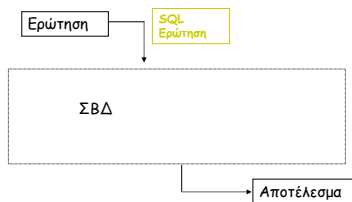
Αποφάσεις που Απαιτούνται

- Τι ευρετήρια πρέπει να δημιουργηθούν;
- Ποιες σχέσεις πρέπει να έχουν ευρετήρια; Ποια γνωρίσματα χρησιμοποιούνται για αναζήτηση; Πρέπει να ορίσουμε πολλαπλά ευρετήρια;
- Για κάθε ευρετήριο, τι είδους ευρετήριο πρέπει να είναι;
 - Συστάδες; Δέντρο/Κατακερματισμός; Δυναμικό/Στατικό; Πυκνό/Μη-πυκνό;
- Χρειάζονται αλλαγές και στο εννοιολογικό/λογικό Σχήμα;
 - Διαφορετικό κανονικοποιημένο σχήμα;
 - Denormalization (μήπως χρειάζεται από-κανονικοποίηση);
 - Όψεις, Επανάληψη Δεδομένων (replication) ...

Επεξεργασία Ερωτήσεων



Θα δούμε την «πορεία» μιας SQL ερώτησης (πως εκτελείται)





Τα βασικά βήματα στην επεξεργασία μιας ερώτησης είναι

1. Συντακτική Ανάλυση & Μετάφραση
2. Βελτιστοποίηση
3. Υπολογισμός



1. Συντακτική Ανάλυση (Parsing) & Μετάφραση

Η SQL ερώτηση μεταφράζεται σε μια εσωτερική μορφή αφού γίνει ο απαραίτητος συντακτικός και σημασιολογικός έλεγχος (π.χ., τα ονόματα που αναφέρονται είναι ονόματα σχέσεων που υπάρχουν)

Αντικατάσταση των όψεων από τον ορισμό τους

Σε ποια εσωτερική μορφή: Έκφραση της σχεσιακής άλγεβρας

```
select A1, A2, ..., An
from R1, R2, ..., Rm      πA1, A2, ..., An (σP (R1 × R2 × ... × Rm))
where P
```



2. Βελτιστοποίηση

Μια SQL ερώτηση μπορεί να μεταφραστεί σε διαφορετικές (ισοδύναμες) εκφράσεις της σχεσιακής άλγεβρας

```
select balance      • πbalance (σbalance < 2500 (account))
from account
where balance < 25000 • σbalance < 2500 (πbalance(account))
```

Με ποιο κριτήριο γίνεται η επιλογή της έκφρασης;



Κάθε πράξη της σχεσιακής άλγεβρας μπορεί να υλοποιηθεί με διαφορετικούς αλγόριθμους:

π.χ., για την υλοποίηση της επιλογής μπορεί

είτε να σαρώσουμε (scan) όλο το αρχείο ελέγχοντας κάθε εγγραφή αν ικανοποιεί τη συνθήκη

είτε αν υπάρχει π.χ., ένα Β* ευρετήριο στο γνώρισμα balance να χρησιμοποιήσουμε το ευρετήριο

Άρα δεν αρκεί ο προσδιορισμός της πράξης - πρέπει να προσδιορίζεται και ο αλγόριθμος που θα χρησιμοποιηθεί για την υλοποίησή της



βασικές (primitive) πράξεις: πράξη + αλγόριθμος

Σχέδιο εκτέλεσης (execution plan): μια ακολουθία από βασικές πράξεις

```
πbalance
|
σbalance < 2500 χρησιμοποιήσει το ευρετήριο 1
|
account
```



- Τα διαφορετικά σχέδια εκτέλεσης έχουν και διαφορεικό κόστος

- **Βελτιστοποίηση:** η διαδικασία επιλογής του σχεδίου εκτέλεσης που έχει το μικρότερο κόστος

- Εκτίμηση του κόστους (συνήθως χρήση στατιστικών στοιχείων)



3. Εκτέλεση

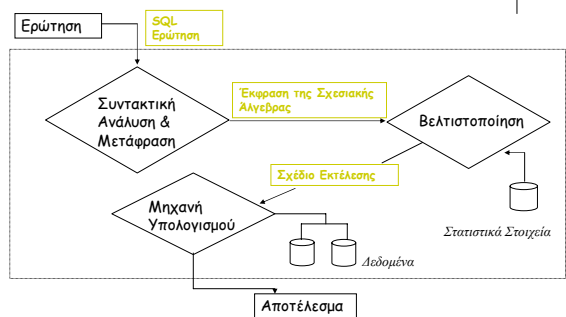
Μηχανή εκτέλεσης που εκτελεί τις βασικές πράξεις



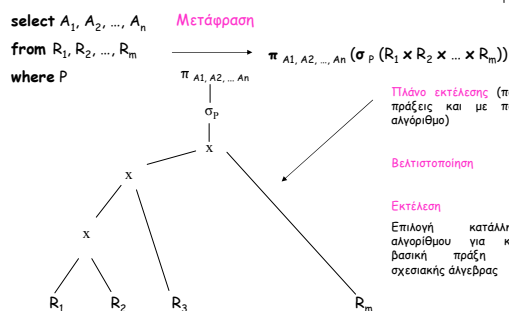
Υπάρχουν υλοποιημένοι μια σειρά από αλγόριθμοι για κάθε βασική πράξη (π.χ., που χρησιμοποιούν ή όχι ευρετήρια κλπ)

Γενικά, το ΣΔΒΔ με βάση κάποια *στατιστικά στοιχεία* κάνει μια *εκτίμηση του κόστους* και *επιλέγει τον αλγόριθμο* για κάθε πράξη με τον μικρότερο (με βάση την εκτίμηση) κόστος

Επεξεργασία Ερωτήσεων (ανακεφαλαίωση)

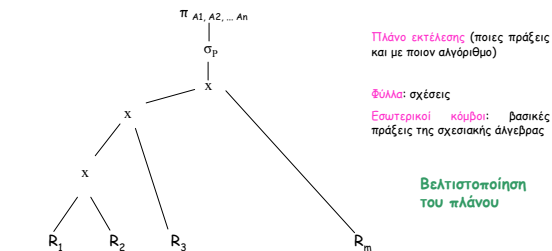


Επεξεργασία Ερωτήσεων (ανακεφαλαίωση)



Επεξεργασία Ερωτήσεων

Λίγα λόγια για τη βελτιστοποίηση



Βελτιστοποίηση Ερωτήσεων

Μερικοί ευριστικοί κανόνες

Γενική ιδέα: εκτέλεση πρώτα των πράξεων με μικρή επιλεκτικότητα ώστε να περιοριστεί το μέγεθος των ενδιάμεσων αποτελεσμάτων

1. Διάσπαση των πράξεων επιλογής με συζευκτικές συνθήκες σε ακολουθίες πράξεων επιλογής
2. Μετατοπίζουμε την *πράξη επιλογής όσο πιο κάτω* επιτρέπεται από τα γνωρίσματα που περιλαμβάνονται στη συνθήκη
3. Επαναδιευθέτηση των φύλλων ώστε να εκτελούνται πρώτα οι σχέσεις που έχουν τις πιο περιοριστικές πράξεις επιλογής



4. Συνδυασμός μιας πράξης καρτεσιανού γινομένου με μια πράξη επιλογής που ακολουθεί

5. Διάσπαση και μετακίνηση των λιστών προβολής όσο πιο κάτω γίνεται στο δέντρο

6. Εντοπισμός υποδέντρων με ομάδες πράξεων που μπορεί να εκτελεστούν με κοινό αλγόριθμο



Αλγόριθμους εκτέλεσης βασικών πράξεων

- Επιλογή
- Προβολή
- Πράξεις συνόλων
- Συνένωση



Για να επιλέξουμε ποιόν αλγόριθμο θα χρησιμοποιήσουμε, διατηρούμε στατιστικά στοιχεία

Για ένα αρχείο δεδομένων μιας σχέσης R:

- n_R : αριθμός πλειάδων της σχέσης R
- b_R : αριθμός blocks της σχέσης R
- s_R : μέγεθος σε bytes κάθε πλειάδας της σχέσης R
- f_R : παράγοντας ομαδοποίησης (αριθμός εγγραφών ανά block) αν μη εκτεινόμενη, $f_R = \lfloor B / s_R \rfloor$ και $b_R = \lceil n_R / f_R \rceil$

Ενημέρωση στατιστικών στοιχείων;



Άλλα στατιστικά στοιχεία;

Π.χ., για μια πράξη επιλογής στο γνώρισμα A

- $V(A, R)$: αριθμός διαφορετικών τιμών του A
 $|π_A(R)|$ -- αν το A κλειδί;
- $SC(A, R)$: μέσος αριθμός πλειάδων που ικανοποιεί μια συνθήκη (δεδομένου ότι υπάρχει μια τουλάχιστον που την ικανοποιεί)
1 αν κλειδί, αν ομοιόμορφη;



Στατιστικά στοιχεία επίσης για το αρχείο ερευτηρίου (αν υπάρχει)

- f_i : παράγοντας διακλάδωσης, πολυεπίπεδο f_0 , B* δέντρο ~ τάξη
- H_i : αριθμός επιπέδων
- LB_i : αριθμός block φύλλων

Με βάση τα στατιστικά επιλέγεται ο αλγόριθμος με το μικρότερο κόστος
Κόστος: Αριθμό blocks που μεταφέρονται



Επιλογή

Θα εξετάσουμε:

- Επιλογή με συνθήκη ισότητας ($\sigma_{A = a}(R)$)
- Επιλογή με συνθήκη σύγκρισης - διαστήματος/περιοχής (range query) ($\sigma_{A \leq u}(R)$) ή ($\sigma_{A \geq u}(R)$)
- Επιλογή με σύζευξη ($\sigma_{\theta_1 \text{ AND } \theta_2 \dots \text{ AND } \theta_n}(R)$)
- Επιλογή με διάζευξη ($\sigma_{\theta_1 \text{ OR } \theta_2 \dots \text{ OR } \theta_n}(R)$)

Θα δούμε ποιος είναι ο καλύτερος (με το μικρότερο κόστος σε blocks) αλγόριθμος για την εκτέλεση της πράξης



Πιθανοί αλγόριθμοι εκτέλεσης για την επιλογή:

- E1: Σειριακή αναζήτηση
- E2: Δυαδική αναζήτηση
- E3: Χρήση πρωτεύοντος ευρετηρίου/κατακερματισμού
- E4: Χρήση δευτερεύοντος ευρετηρίου/κατακερματισμού

Για τον E2 πρέπει το αρχείο να είναι ταξινομημένο
 Για τους E3 και E4 λέμε ότι έχουμε μονοπάτι προσπέλασης (access path)



Επιλεκτικότητα επιλογής:

το πλήθος των εγγραφών (πλειάδων) που επιλέγονται (δηλ. ικανοποιούν την συνθήκη) / το πλήθος των εγγραφών (πλειάδων) του αρχείου (σχέσης)

• Έστω $s_i = | \sigma_{\theta_i}(R) |$

επιλεκτικότητα: s_i / n_R

Αν θ_i συνθήκη ισότητας σε ένα γνώρισμα υποψήφιο κλειδί $s_i = 1 / n_R$

Αν θ_i συνθήκη ισότητας σε ένα γνώρισμα, ομοιόμορφη κατανομή, k διακριτές τιμές, $s_i = k / n_R$



Επιλογή - συνθήκη ισότητας $\sigma_A = a (R)$

E1 Σειριακή αναζήτηση

Διάβασμα (scan) όλου του αρχείου

b_R : αριθμός blocks της σχέσης R

b_R

$b_R/2$ αν το A υποψήφιο κλειδί (οπότε το αποτέλεσμα έχει μόνο μία πλειάδα, σταματάμε την αναζήτηση μόλις τη βρούμε)

Μπορεί να χρησιμοποιηθεί σε οποιοδήποτε αρχείο



E2 Δυαδική αναζήτηση

b_R : αριθμός blocks της σχέσης R
 $SC(A, R)$: μέσος αριθμός πλειάδων που ικανοποιεί μια συνθήκη
 f_R : παράγοντας ομαδοποίησης

Μπορεί να χρησιμοποιηθεί μόνο αν το αρχείο είναι διατεταγμένο με βάση το γνώρισμα της επιλογής

$$\lceil \log (b_R) \rceil \leftarrow \text{Εύρεση της πρώτης}$$

$$+ \lceil SC(A, r)/f_R \rceil - 1 \leftarrow \text{Εύρεση των υπόλοιπων}$$

Αν το A υποψήφιο κλειδί;

E3 Χρήση πρωτεύοντος (πολυεπίπεδου) ευρετηρίου



Πρωτεύον ευρετήριο σημαίνει ταξινομημένο αρχείο

b_R : αριθμός blocks της σχέσης R
 $SC(A, R)$: μέσος αριθμός πλειάδων που ικανοποιεί μια συνθήκη
 f_R : παράγοντας ομαδοποίησης
 HT_i : αριθμός επιπέδων

Μπορεί να χρησιμοποιηθεί μόνο αν υπάρχει τέτοιο ευρετήριο στο A

$HT_i + 1 \leftarrow$ Εύρεση και μεταφορά της πρώτης

Αν το A δεν είναι υποψήφιο κλειδί -- ευρετήριο συστάδων

$HT_i + \lceil SC(A, R)/f_R \rceil \leftarrow$ Εύρεση και των υπόλοιπων

ΣΗΜΕΙΩΣΗ: Πρωτεύον ευρετήριο στο A, σημαίνει ότι οι εγγραφές του αρχείου δεδομένων είναι ταξινομημένες (διατεταγμένες) ως προς A άρα οι υπόλοιπες εγγραφές με την ίδια τιμή (αν υπάρχουν) βρίσκονται σε γειτονικά blocks του αρχείου δεδομένων



E4 Χρήση δευτερεύοντος (πολυεπίπεδου) ευρετηρίου

b_R : αριθμός blocks της σχέσης R
 $SC(A, R)$: μέσος αριθμός πλειάδων που ικανοποιεί μια συνθήκη
 f_R : παράγοντας ομαδοποίησης
 HT_i : αριθμός επιπέδων

Μπορεί να χρησιμοποιηθεί μόνο αν υπάρχει τέτοιο ευρετήριο στο A
 Αν το A είναι υποψήφιο κλειδί

$HT_i + 1 \leftarrow$ Εύρεση και μεταφορά της πρώτης

Αν το A δεν είναι υποψήφιο κλειδί ± κόστος για την εύρεση των υπολοίπων

$HT_i + \text{ενδιάμεσο επίπεδο} + SC(A, R) \leftarrow$ Εύρεση και των υπόλοιπων

Στη χειρότερη περίπτωση κάθε εγγραφή που ικανοποιεί τη συνθήκη σε διαφορετικά block

Επιλογή - συνθήκη με σύγκριση

$$\sigma_{A \leq u} (R) \text{ ή } \sigma_{A \geq u} (R)$$

Έστω ότι c πλειάδες ικανοποιούν τη συνθήκη

Γενικά $c = n_R/2$ (δηλαδή, οι μισές)

Έστω min, max (μικρότερη, μεγαλύτερη τιμή του A), αν ομοιόμορφη κατανομή και $\sigma_{A \leq u} (R)$

$$c = \begin{cases} 0 & \text{αν } u < \min \\ n_R & \text{αν } u \geq \max \\ n_R * [(u - \min) / (\max - \min)] & \end{cases}$$

$$\sigma_{A \leq u} (R)$$

Θα δούμε δύο αλγόριθμους:

E5 Χρήση πρωτεύοντος πολυ-επίπεδου ευρετηρίου

E6 Χρήση δευτερεύοντος πολυ-επίπεδου ευρετηρίου

E5 Χρήση πρωτεύοντος (πολυεπίπεδου) ευρετηρίου

Πρωτεύον, σημαίνει ταξινομημένο αρχείο, έστω σε αύξουσα διάταξη

$$A \geq u$$

1. Χρήση ευρετηρίου για την εύρεση της πρώτης εγγραφής $A \geq u$
2. Σάρωση όλου του αρχείου ξεκινώντας από αυτήν την εγγραφή

$$HT_i + \lceil c / f_R \rceil$$

$$A \leq u$$

Δε χρειάζεται ευρετήριο, γιατί:

*c: επιλεξιμότητα (πλειάδες που ικανοποιούν την συνθήκη)
f_R: παράγοντας ομαδοποίησης
HT_i: αριθμός επιπέδων*

E6 Χρήση δευτερεύοντος (πολυεπίπεδου) ευρετηρίου

- Εύρεση του πρώτου φύλλου του ευρετηρίου *Μη ταξινομημένο αρχείο*
- Για κάθε block (φύλλο) του ευρετηρίου διάβασε το αντίστοιχο block δεδομένων *(σημείωση, αν B+ δέντρο χρησιμοποιούμε το δείκτη ανάμεσα στα φύλλα)*

Σάρωση των φύλλων του δέντρου

$$A \leq u \text{ από την αρχή έως το } u$$

$$A \geq u \text{ από το } u \text{ έως το τέλος}$$

Π.χ., αν $c = n_R/2$, και A κλειδί τότε (αν κάθε εγγραφή σε διαφορετικό block)

$$HT_i + LB_i/2 + n_R/2$$

*c: επιλεξιμότητα (πλειάδες που ικανοποιούν την συνθήκη)
n_R: αριθμός εγγραφών
LB_i: αριθμός block φύλλων
HT_i: αριθμός επιπέδων*

Επιλογή - συνθήκη σύζευξης

$$\sigma_{\theta_1 \text{ AND } \theta_2 \dots \text{ AND } \theta_n} (R)$$

Επιλεκτικότητα μιας συνθήκης:

το πλήθος των εγγραφών (πλειάδων) που την ικανοποιούν
το πλήθος των εγγραφών (πλειάδων) του αρχείου (σχέσης)

• Αν οι συνθήκες είναι ανεξάρτητες, το μέγεθος του αποτελέσματος:

$$\frac{n_R * s_1 * s_2 * \dots * s_n}{n_R^n}$$

s_i επιλεκτικότητα της θ_i

E7 Συζευκτική επιλογή με χρήση ενός απλού ευρετηρίου

Υπάρχει διαδρομή προσπέλασης για ένα από τα γνωρίσματα που εμφανίζονται σε οποιαδήποτε απλή συνθήκη

Επιλογή του γνωρίσματος στην απλή συνθήκη με τη **μικρότερη** επιλεκτικότητα (γιατί:)

Χρήση μιας από τις προηγούμενες μεθόδους για την ανάκτηση των εγγραφών που ικανοποιούν αυτήν την συνθήκη και

έλεγχος για κάθε επιλεγμένη εγγραφή αν ικανοποιεί και τις υπόλοιπες συνθήκες



E8 Συζευκτική επιλογή με χρήση σύνθετου ευρετηρίου

Αν υπάρχει ευρετήριο στο συνδυασμό δύο ή περισσότερων γνωρισμάτων που εμφανίζονται σε οποιαδήποτε απλές συνθήκες



E9 Συζευκτική επιλογή με τομή δεικτών

Αν υπάρχουν ευρετήρια σε περισσότερα από ένα από τα γνωρίσματα

Τότε διαβάζουμε τα blocks του αρχείου δεδομένων που δίνονται από όλα τα ευρετήρια



Επιλογή - συνθήκη διάζευξης

$$\sigma_{\theta_1 \text{ OR } \theta_2 \dots \text{ OR } \theta_n (R)}$$

Αν έστω και μία από τις συνθήκες δεν έχει διαδρομή προσπέλασης -> σάρωση όλου του αρχείου



ΤΕΛΟΣ