

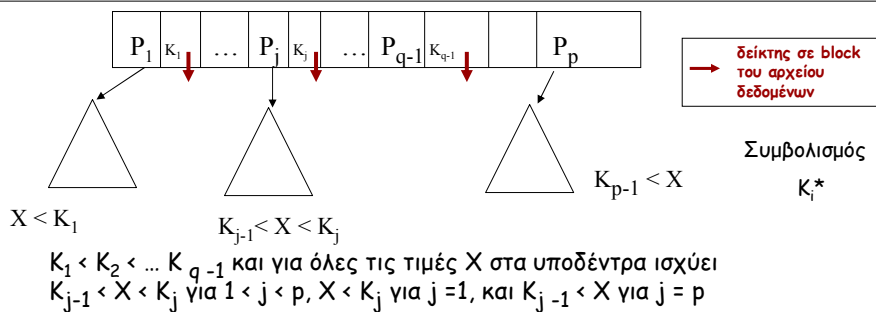


# Δεντρικά Ευρετήρια



## 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





Κάθε κόμβος του δέντρου είναι ένα block στο δίσκο

Ισοζυγισμένο: όλοι οι κόμβοι-φύλλα στο ίδιο επίπεδο

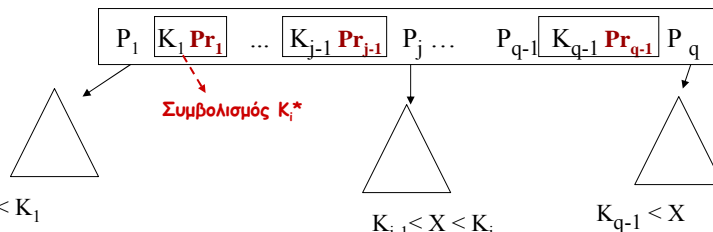
B-δέντρο: ένα δέντρο αναζήτησης που παραμένει ισοζυγισμένο και χωρίς «πολύ αδειανούς» κόμβους



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

1. 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



2. Σε κάθε κόμβο  $K_1 < K_2 < \dots < K_{q-1}$

3. Για όλες τις τιμές  $X$  στο υποδέντρο που δείχνει το  $P_j$  ισχύει  $K_{j-1} < X < K_j$  για  $1 < j < q$ ,  $X < K_j$  για  $j=1$ , και  $K_{j-1} < X$  για  $j=q$

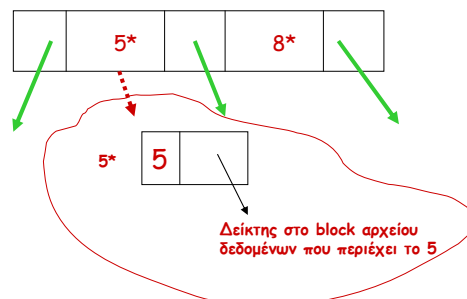


4. Κάθε κόμβος έχει το πολύ p δείκτες δέντρου
5. Κάθε κόμβος εκτός της ρίζας και των φύλλων έχει τουλάχιστον  $\lceil p/2 \rceil$  δείκτες. Η ρίζα έχει τουλάχιστον 2 εκτός αν είναι ο μόνος κόμβος του δέντρου.
6. Ένας κόμβος με q δείκτες δέντρου περιέχει q - 1 τιμές πεδίου αναζήτησης (και άρα και q - 1 δείκτες δεδομένων).
7. 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 Επιπέδου 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

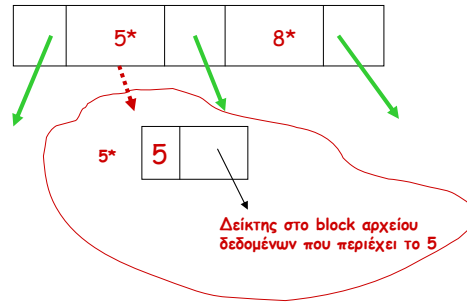
**ΠΡΟΣΟΧΗ:** η εισαγωγή της μεσαίας τιμής στο γονέα αν ο γονέας είναι γεμάτος μπορεί να οδηγήσει σε διάσπαση του γονέα. Η διάσπαση μπορεί να οδηγήσει ως τη ρίζα, οπότε δημιουργείται και νέο επίπεδο.

## 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)



Δείκτης στο block αρχείου  
δεδομένων που περιέχει το 5

## B-δέντρα



### Διαγραφή τιμής

Τιμή προς διαγραφή ανήκει σε φύλλο  $\rightarrow$  OK

Τιμή προς διαγραφή ανήκει σε εσωτερικό κόμβο  $\rightarrow$

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)



### Διαγραφή τιμής

Αν υποχείλιση

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 υποχείλιση και στο  
 γονέα)

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



- 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



## 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 ευρετηριοποίησης έχει μέγεθος  $V_A = 8$  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);



### Υπολογισμός επιπέδων

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

Πόσα επίπεδα χρειαζόμαστε για να ευρετηριοποιήσουμε 65.000 τιμές;

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

	#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	

$$\text{Σύνολο: } 61.440 + 3.840 + 240 + 15 \text{ (65.535)}$$



**Διαφορά B<sup>+</sup> 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*

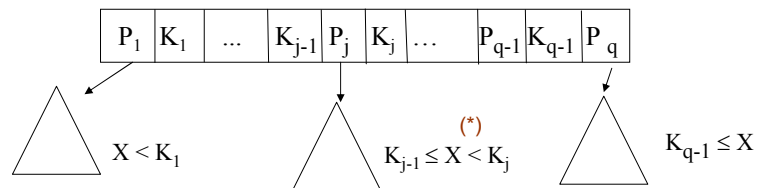




Ένα **B<sup>+</sup>-δέντρο** τάξεως (order)  $p$  ορίζεται ως εξής:

1. Κάθε **εσωτερικός κόμβος** είναι της μορφής

$\langle P_1, K_1, P_2, K_2, \dots, K_{q-1}, P_{q-1}, P_q \rangle$   $q \leq p$ , 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  $K_j < X < K_{j+1}$  για  $1 < j < q$ ,  $X < K_1$  για  $j=1$ , και  $K_{q-1} <= X$  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

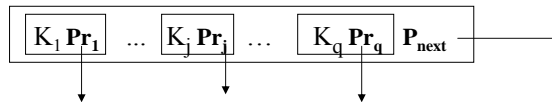
$\langle \langle K_1, Pr_1 \rangle, \langle K_2, Pr_2 \rangle, \dots, \langle K_q, Pr_q \rangle, P_{next} \rangle$ ,  $q \leq p_{leaf}$ , όπου

$p_{leaf}$  είναι η τάξη των κόμβων-φύλλων

$K_i$  τιμή αναζήτησης,

$Pr_i$  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

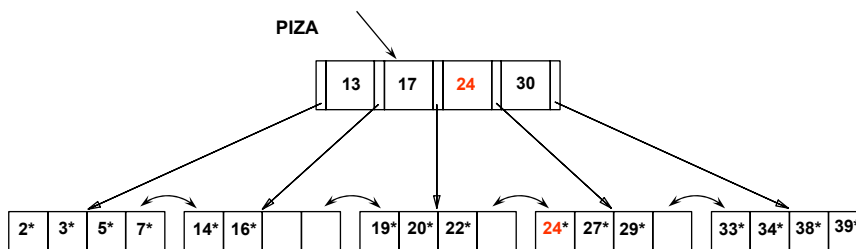
4. Κάθε κόμβος-φύλλο έχει **τουλάχιστον**  $\lceil (p_{leaf} / 2) \rceil$  τιμές.

5. Όλοι οι κόμβοι-φύλλα βρίσκονται στο ίδιο επίπεδο.



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

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



```
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
```



### Αναζήτηση (αναδρομική εκδοχή)

```
nodepointer find(keyvalue K):  
    return tree_search(root, K);  
end;
```



### 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



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



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

## B+-δέντρα: Εισαγωγή



Οι διασπάσεις κόμβων (εκτός ρίζας) "μεγαλώνουν" το δέντρο

Η διάσπαση της ρίζας "υψώνει" το δέντρο

## B+-δέντρα (παράδειγμα)

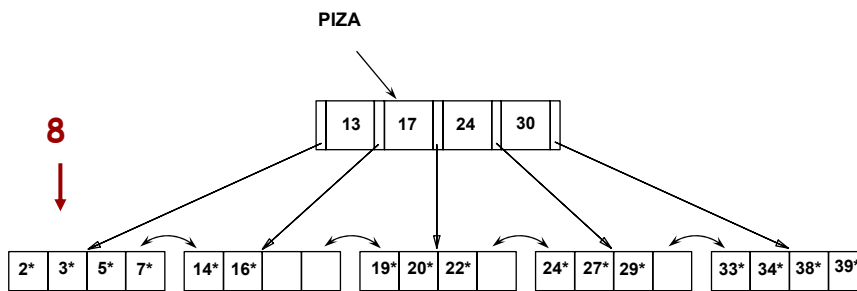


5, 9, 7, 14, 6, 19, 10 και τάξη  $p = 3$  (2 τιμές ανά κόμβο, 3 δείκτες block ευρετηρίου)  
και  $p_{leaf} = 2$

## B+-δέντρα: Εισαγωγή



### Εισαγωγή της καταχώρησης 8\*



Βάσεις Δεδομένων 2009-2010

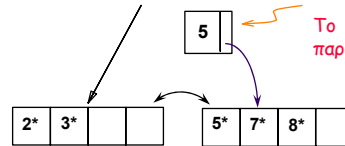
Ευαγγελία Πιπουρά

29

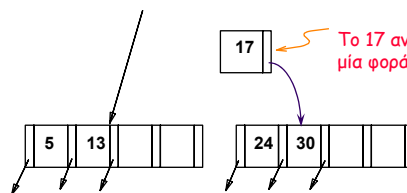
## B+-δέντρα: Εισαγωγή



Καταχώρηση στον κόμβο γονέα (αντιγραφή)



Καταχώρηση στον κόμβο γονέα (μεταφορά)

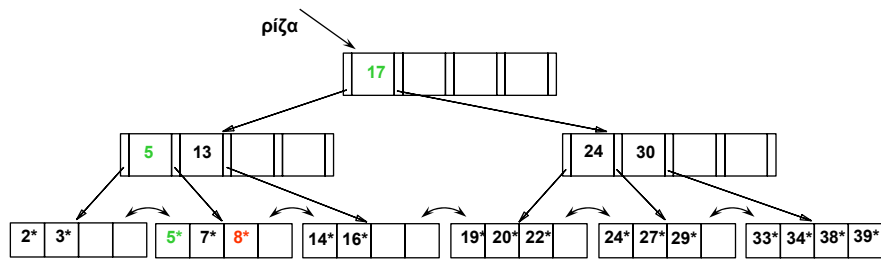


Βάσεις Δεδομένων 2009-2010

Ευαγγελία Πιπουρά

30

## B+-δέντρα: Εισαγωγή



Η ρίζα διασπάστηκε οδηγώντας σε αύξηση του ύψους.

## B+-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 ( $\succ \lceil (n/2) \rceil$ )

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 ( $\succ \lceil (n/2) \rceil$ )

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, **σβήνεται** η εγγραφή που δείχνει στον κόμβο (πιθανότητα νέας υποχείλισης)



## B+-δέντρα: Διαγραφή

### 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

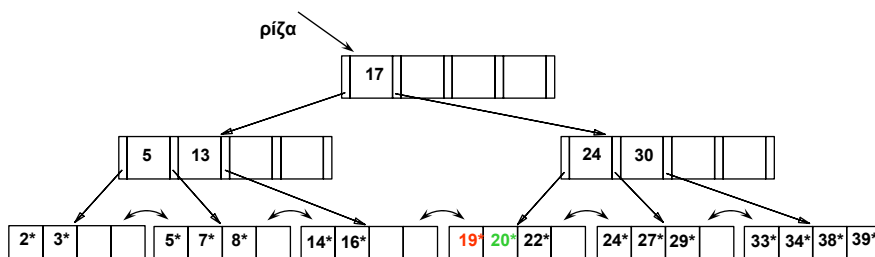
Πάλι μέσω του γονέα τους

Δηλαδή θεωρούμε και την τιμή του γονέα στην ανακατανομή

Η τιμή αυτή αλλάζει στο γονέα

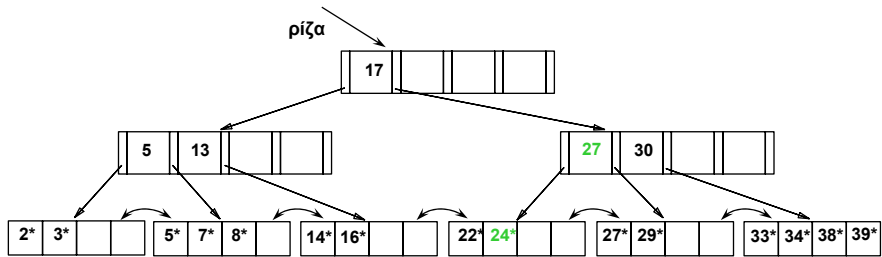


## B+-δέντρα: Παράδειγμα



### Διαγραφή 19, 20

### B+-δέντρα: Διαγραφή



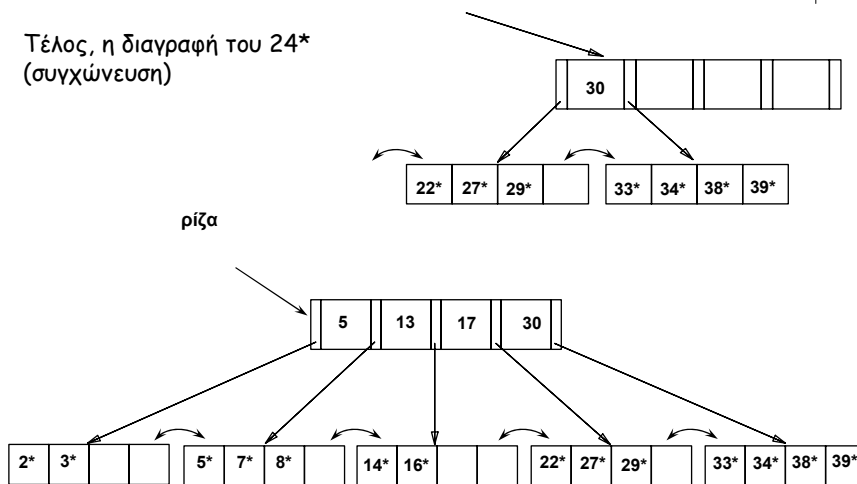
Το παράδειγμα μετά τη διαγραφή του 19\* και του 20\* (ανακατανομή με δεξί αδελφό και αντικατάσταση του 24 με 27)

Διαγραφή του 24 ->

### B+-δέντρα: Διαγραφή



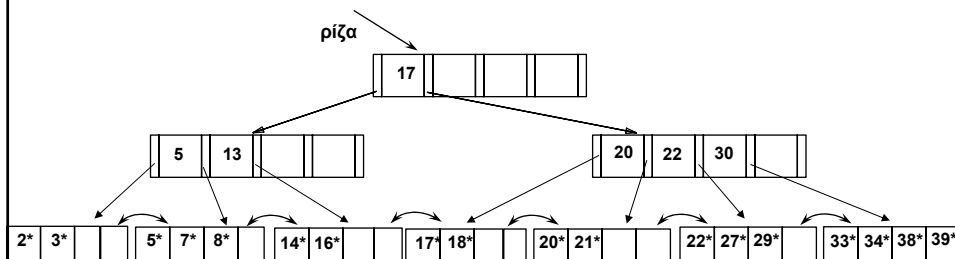
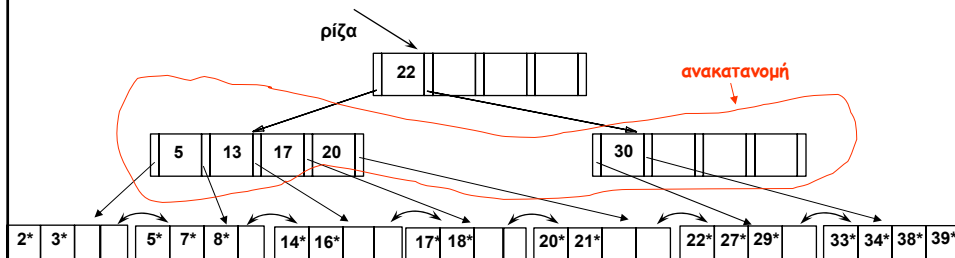
Τέλος, η διαγραφή του 24\* (συγχώνευση)





Παράδειγμα ανακατανομής

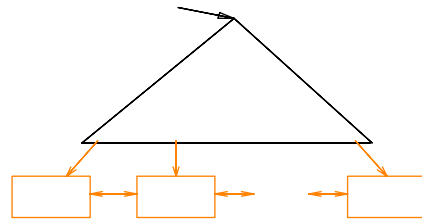
Έστω στο παρακάτω δέντρο μετά από συγχώνευση φύλλων





## B+-δέντρα γενικά

- Εισαγωγή/Διαγραφή με κόστος  $\log_F N$  --- 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



Καταχωρήσεις Ευρετηρίου  
(Άμεση Αναζήτηση)

Καταχωρήσεις Δεδομένων  
(«Σύνολο ακολουθίας»)



## B+-δέντρα

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

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

Έστω B μέγεθος block, V μέγεθος πεδίου αναζήτησης, Pr μέγεθος δείκτη δεδομένων (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, Pr = 7 bytes, P = 6 bytes, τότε p = 34

Για B-δέντρο, p = 23



### 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  $p_{leaf} = 31$



### 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 ( $12.167 * 31 * 0,69$ ) καταχωρήσεις	12.167 δείκτες

Σε 3 επίπεδα **255.507** εγγραφές έναντι 65.535 για το B-δέντρο

→ Σημείωση: εγγραφές μόνο στα φύλλα



### 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(s)-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





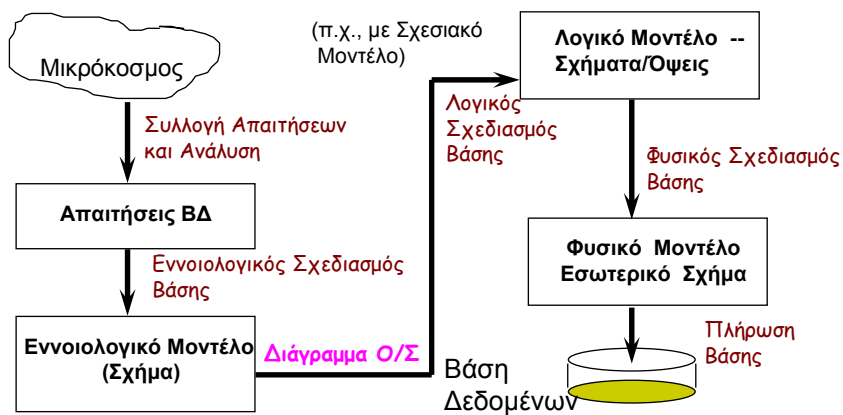
Οι εγγραφές στο ευρετήριο  $K^*$

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

Εξαρτώμενο του επιλεγμένου ΣΔΒΔ



## Φυσικός Σχεδιασμός



- 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)

## Ευρετήρια



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 συνυπολογίσουμε 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) ...