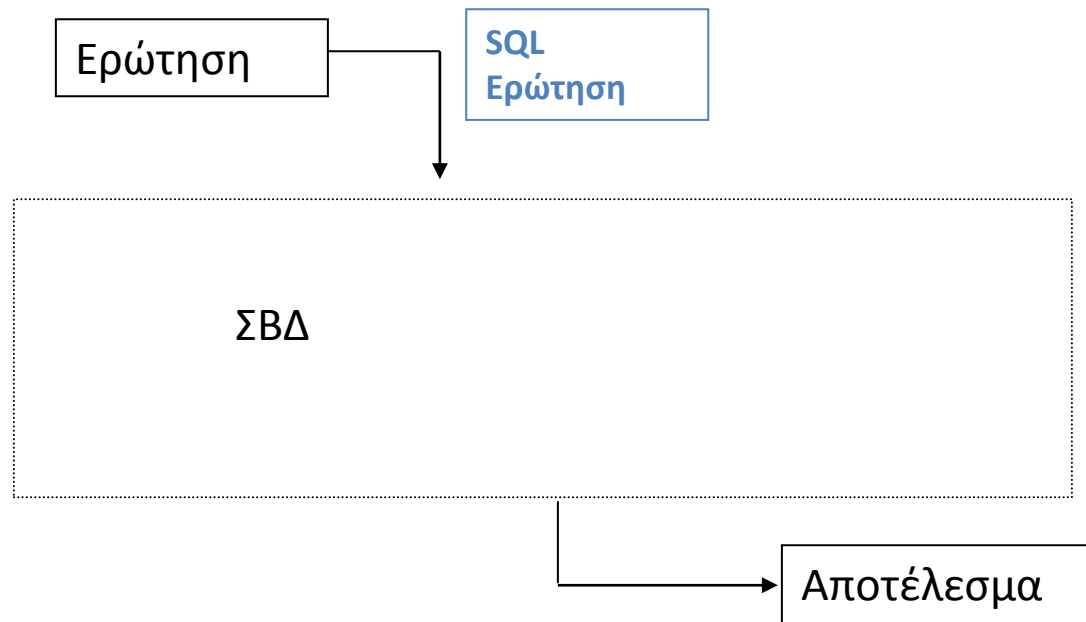


# Εισαγωγή στην Επεξεργασία Ερωτήσεων

# Επεξεργασία Ερωτήσεων

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



# Βήματα Επεξεργασίας

Τα βασικά βήματα στην επεξεργασία μιας ερώτησης είναι

1. Συντακτική Ανάλυση & Μετάφραση
2. Βελτιστοποίηση
3. Υπολογισμός (Εκτέλεση)

# Συντακτική Ανάλυση (parsing) και μετάφραση

Συντακτικός και σημασιολογικός έλεγχος (π.χ., τα ονόματα που αναφέρονται είναι ονόματα σχέσεων που υπάρχουν)

Αντικατάσταση των όψεων από τον ορισμό τους

Η SQL ερώτηση μεταφράζεται σε μια εσωτερική μορφή

Σε ποια εσωτερική μορφή; Ισοδύναμη έκφραση της σχεσιακής άλγεβρας

**select**  $A_1, A_2, \dots, A_n$

**from**  $R_1, R_2, \dots, R_m$

**where** P

$\pi_{A_1, A_2, \dots, A_n} (\sigma_P (R_1 \times R_2 \times \dots \times R_m))$

# Βελτιστοποίηση Ερωτήσεων

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

**select** balance

**from** account

**where** balance < 25000

•  $\pi_{\text{balance}} (\sigma_{\text{balance} < 2500} (\text{account}))$

•  $\sigma_{\text{balance} < 2500} (\pi_{\text{balance}} (\text{account}))$

*Με ποιο κριτήριο γίνεται η επιλογή της έκφρασης;*

- *Η βελτιστοποίηση είναι το πιο «δύσκολο» βήμα – θα δούμε κάποιους ευρυστικούς στη συνέχεια*

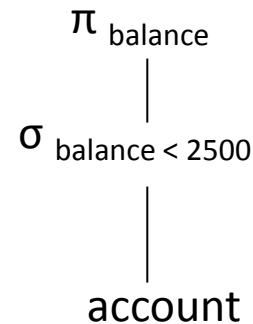
# Πλάνο Εκτέλεσης

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

Αναπαρίσταται με ένα δέντρο

Φύλλα: σχέσεις

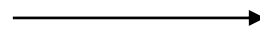
Εσωτερικοί κόμβοι: βασικές (primitive) πράξεις της σχεσιακής άλγεβρας



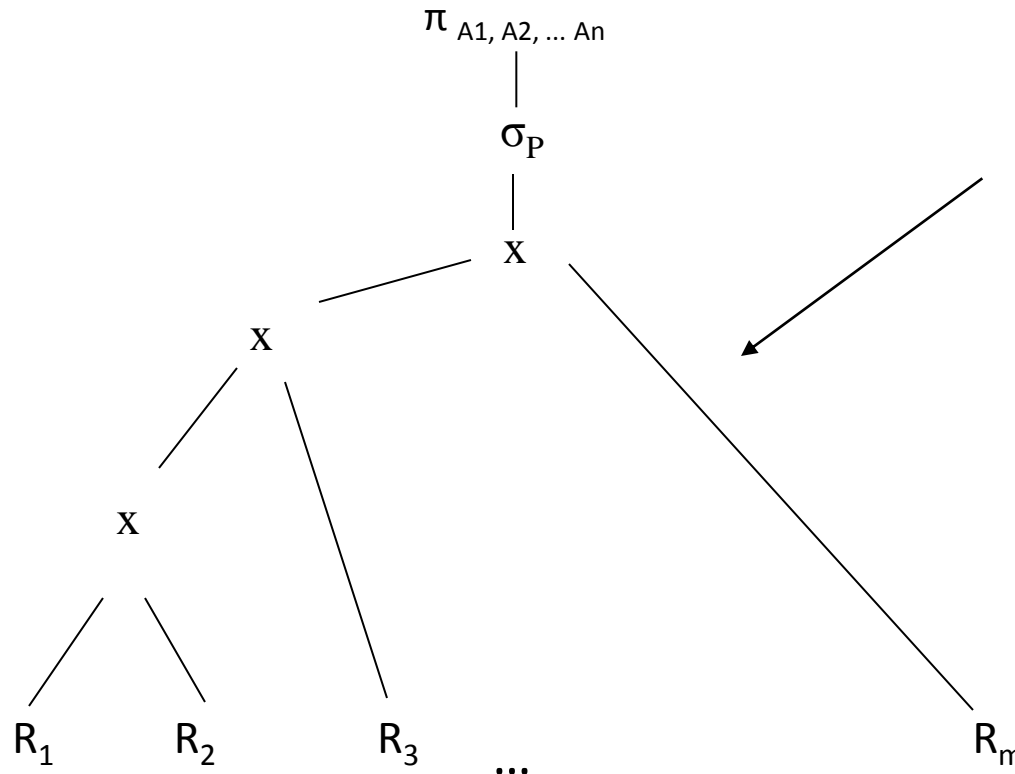
# Πλάνο Εκτέλεσης

**select**  $A_1, A_2, \dots, A_n$   
**from**  $R_1, R_2, \dots, R_m$   
**where**  $P$

Μετάφραση



$\pi_{A_1, A_2, \dots, A_n} (\sigma_P (R_1 \times R_2 \times \dots \times R_m))$



Πλάνο εκτέλεσης

Φύλλα: σχέσεις

Εσωτερικοί κόμβοι:  
βασικές πράξεις της  
σχεσιακής άλγεβρας

Βελτιστοποίηση του  
πλάνου

# Βελτιστοποίηση

- Τα διαφορετικά πλάνα εκτέλεσης έχουν και διαφορεικό κόστος
- **Βελτιστοποίηση**: η διαδικασία επιλογής του σχεδίου εκτέλεσης που έχει το μικρότερο κόστος
- **Εκτίμηση του κόστους** (συνήθως χρήση στατιστικών στοιχείων)
  - επιλεξιμότητα (selectivity): ποσοστό πλειάδων εισόδου που εμφανίζονται στο αποτέλεσμα



# Ευριστικοί Κανόνες Βελτιστοποίησης Πλάνου Εκτέλεσης

*Γενική ιδέα: εκτέλεση πρώτα των πράξεων με μικρή επιλεξιμότητα 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. Επανα-διευθέτηση των φύλλων ώστε να εκτελούνται πρώτα οι σχέσεις που έχουν τις πιο περιοριστικές πράξεις επιλογής

# Ευριστικοί Κανόνες Βελτιστοποίησης Πλάνου Εκτέλεσης

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

# Φυσικό Πλάνο Εκτέλεσης

Κάθε πράξη της σχεσιακής άλγεβρας μπορεί να υλοποιηθεί με *διαφορετικούς αλγορίθμους*:

π.χ., για την υλοποίηση της επιλογής μπορεί για παράδειγμα:

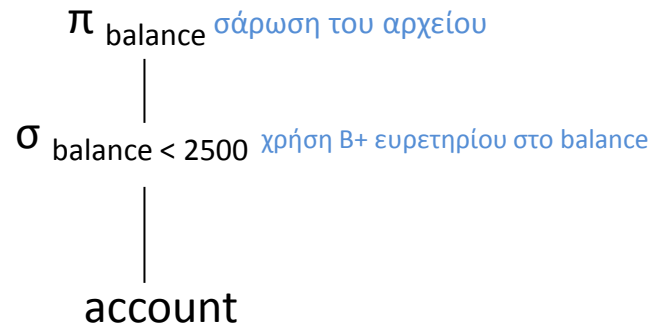
- να σαρώσουμε (scan – σειριακή αναζήτηση) όλο το αρχείο ελέγχοντας κάθε εγγραφή αν ικανοποιεί τη συνθήκη
- αν υπάρχει π.χ., ένα B<sup>+</sup> ευρετήριο στο γνώρισμα balance να χρησιμοποιήσουμε το ευρετήριο

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

# Φυσικό Πλάνο Εκτέλεσης

Λογικό πλάνο εκτέλεσης – μόνο τις πράξεις

Φυσικό πλάνο εκτέλεσης – περιλαμβάνει και τον αλγόριθμο που θα χρησιμοποιηθεί

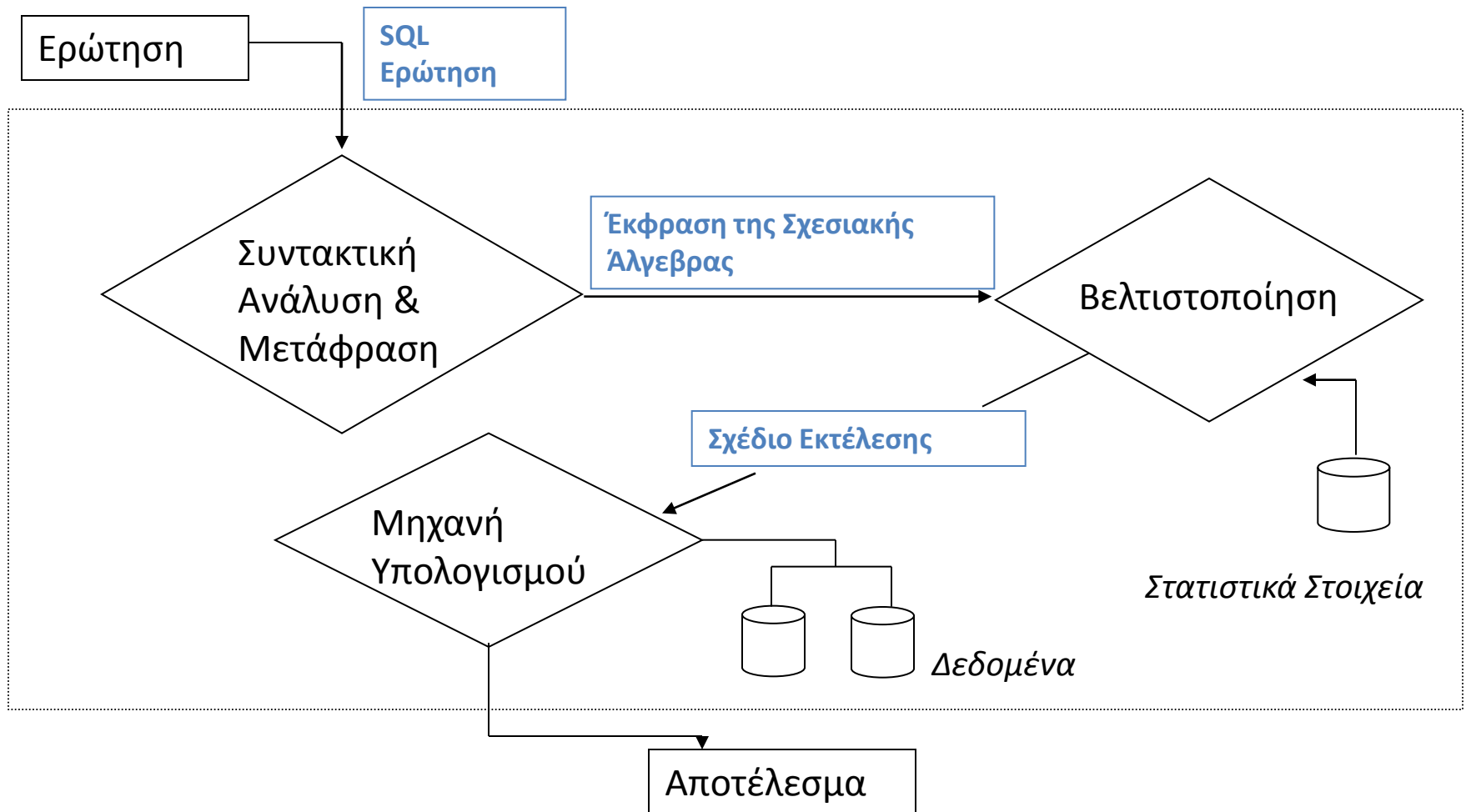


# Εκτέλεση Ερωτήσεων

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 κάνει μια *εκτίμηση του κόστους* και *επιλέγει τον αλγόριθμο* για κάθε πράξη με τον μικρότερο (με βάση την εκτίμηση) κόστος
- Η εκτίμηση του κόστους γίνεται με βάση στατιστικά στοιχεία που αποθηκεύονται στη βάση δεδομένων για αυτό το σκοπό

# Επεξεργασία Ερωτήσεων



# Αλγόριθμοι για βασικές πράξεις

- ✓ Στη συνέχεια, θα δούμε κάποιους αλγορίθμους για την εκτέλεση βασικών πράξεων (επιλογής, συνένωσης και συνόλων) της σχεσιακής άλγεβρας και κάποια εκτίμηση του κόστους τους

*Διαφορετικοί αλγόριθμοι ανάλογα με το αν το αρχείο είναι ή όχι διατεταγμένο, αν υπάρχει ή όχι ευρετήριο και από το είδος του ευρετηρίου*

# Αλγόριθμοι για βασικές πράξεις: στατιστικά στοιχεία

Για να επιλέξουμε ποιόν αλγόριθμο, διατηρούμε στατιστικά στοιχεία

Παράδειγμα

Για ένα *αρχείο δεδομένων* μιας σχέσης R, μπορεί να διατηρούμε στοιχεία όπως:

- $n_R$ : 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  $b_R = \lceil n_R / f_R \rceil$

Στατιστικά στοιχεία επίσης για το *αρχείο ευρετηρίου* (αν υπάρχει)

- $f_i$ : παράγοντας διακλάδωσης,
  - Π ολυεπίπεδο  $f_0$ ,  $B^+$  δέντρο  $\sim$  τάξη
- $H_i$ : αριθμός επιπέδων
- $LB_i$ : αριθμός block φύλλων



# Αλγόριθμοι για βασικές πράξεις: στατιστικά στοιχεία

Άλλα στατιστικά στοιχεία;

- $V(A, R)$ : πλήθος των διαφορετικών τιμών που παίρνει το γνώρισμα  $A$   
 $|\pi_A(R)|$  -- 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;
- Με βάση τα στατιστικά επιλέγεται ο αλγόριθμος με το μικρότερο κόστος
- Υπολογίζεται το I/O κόστος (Αριθμό blocks που μεταφέρονται)
- Επιβάρυνση για την ενημέρωση των στατιστικών

# Αλγόριθμοι για την πράξη της επιλογής

Πιθανοί αλγόριθμοι εκτέλεσης για την *επιλογή*:

E1: Σειριακή αναζήτηση

E2: Δυαδική αναζήτηση (αν το αρχείο είναι ταξινομημένο)

E3: Χρήση πρωτεύοντος ευρετηρίου/κατακερματισμού (αν υπάρχει)

E4: Χρήση δευτερεύοντος ευρετηρίου/κατακερματισμού (αν υπάρχει)

Αν υπάρχει κάποιο ευρετήριο, λέμε ότι έχουμε *μονοπάτι προσπέλασης* (access path)

# Επιλογή – συνθήκη ισότητας

$$\sigma_{A=\alpha}(R)$$

## E1 Σειριακή αναζήτηση

Διάβασμα (scan) όλου του αρχείου

$b_R$ : αριθμός blocks της σχέσης  $R$

$b_R$

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

Μπορεί να χρησιμοποιηθεί σε οποιοδήποτε αρχείο

# Επιλογή – συνθήκη ισότητας

## E2 Δυαδική αναζήτηση

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

Μπορεί να χρησιμοποιηθεί μόνο αν το αρχείο είναι *διατεταγμένο* με βάση το A (δηλαδή, το γνώρισμα της επιλογής)

$$\begin{array}{r} \lceil \log ( b_R ) \rceil \\ + \\ \lceil SC(A, R)/f_R \rceil - 1 \end{array} \quad \begin{array}{l} \longleftarrow \text{Εύρεση της πρώτης} \\ \longleftarrow \text{Εύρεση των υπόλοιπων} \end{array}$$

Αν το A υποψήφιο κλειδί;

# Επιλογή – συνθήκη ισότητας

## Ε3 Χρήση πρωτεύοντος δεντρικού 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 (ύψος)

Μπορεί να χρησιμοποιηθεί μόνο αν υπάρχει τέτοιο ευρετήριο στο A

$HT_i + 1$  ← Εύρεση και μεταφορά της πρώτης

Αν το A δεν είναι υποψήφιο κλειδί -- ευρετήριο συστάδων

$HT_i + \lceil SC(A, R)/f_R \rceil$  ← 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*

# Επιλογή – συνθήκη ισότητας

## E4 Χρήση δευτερεύοντος δεντρικού 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

$HT_i + 1$  ← Εύρεση και μεταφορά της πρώτης

Αν το A δεν είναι υποψήφιο κλειδί  $\pm$  κόστος για την εύρεση των υπολοίπων

$HT_i +$  *ενδιάμεσο επίπεδο*

$+SC(A, R)$  ← Εύρεση και των υπόλοιπων

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

# Επιλογή – συνθήκη με σύγκριση

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

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

Έστω αύξουσα διάταξη

Σειριακή ανάγνωση

Από το 1<sup>ο</sup> block του αρχείου έως την πρώτη εγγραφή με  $A > u$

Κόστος?

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

# Επιλογή – συνθήκη με σύγκριση

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

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

Έστω *αρχείου σωρού* (δεν υπάρχει διάταξη) και B+ δέντρο

Εύρεση στο B+ δέντρο της τιμής u

Χρήση εγγραφών στο φύλλο για τις υπόλοιπες τιμές

Κόστος?



# Επιλογή με σύζευξη

$$\sigma_{P_1 \text{ AND } P_2 \dots \text{ AND } P_n} (R)$$

Υπάρχει διαδρομή προσπέλασης (ευρετήριο) για ένα από τα γνωρίσματα που εμφανίζονται σε οποιαδήποτε συνθήκη

Επιλογή του γνωρίσματος 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

# Επιλογή με διάζευξη

$$\sigma_{P_1 \text{ OR } P_2 \dots \text{ OR } P_n} (R)$$

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

# Συνένωση



Σ1 Εμφωλευμένος (εσωτερικός - εξωτερικός) βρόγχος

Σ2 Χρήση μιας δομής προσπέλασης

Σ3 Ταξινόμηση-Συγχώνευση

Έχει σημασία πόσο χώρο μνήμης κάθε χρονική στιγμή (buffers) μπορούμε να χρησιμοποιήσουμε για τις σχέσεις – δηλαδή, πόσα blocks στην μνήμη

Αρχικά, ας υποθέσουμε ότι έχουμε μόνο 2 blocks

# Συνένωση

## Σ1 Εμφωλευμένος (εσωτερικός-εξωτερικός) βρόγχος

Για κάθε εγγραφή  $t$  της  $R$

Για κάθε εγγραφή  $s$  της  $S$

Αν  $t[A]$  ορ  $s[B]$  πρόσθεσε το  $t$   $s$  στο αποτέλεσμα

Αγνοώντας το κόστος για την εγγραφή των *blocks* του αποτελέσματος

$$b_r + n_R * b_s$$

# Συνένωση

Για κάθε block  $B_r$  της  $R$

Για κάθε block  $B_s$  της  $S$

Για κάθε εγγραφή  $t$  του  $B_r$

Για κάθε εγγραφή  $s$  του  $B_s$

Αν  $t[A]$  ορ  $s[B]$  πρόσθεσε το  $t$   $s$  στο αποτέλεσμα

*Αγνοώντας την εγγραφή των blocks του αποτελέσματος*

$$b_R + b_R * b_S$$

Συμφέρει η τοποθέτηση της μικρότερης σχέσης στον εξωτερικό βρόγχο

# Συνένωση

## Σ2 Χρήση μιας δομής προσπέλασης

Η σχέση για την οποία υπάρχει ευρετήριο τοποθετείται στον εσωτερικό βρόγχο.  
Έστω ότι υπάρχει ευρετήριο για το γνώρισμα  $B$  της σχέσης  $S$

Για κάθε block  $B_r$  της  $R$

Για κάθε εγγραφή  $t$  του  $B_r$

Χρησιμοποίησε το ευρετήριο στο  $B$  για να βρεις τις εγγραφές  $s$  της  $S$   
τέτοιες ώστε  $t[A]$  op  $s[B]$

$b_R + n_R * C$  όπου  $C$  το κόστος μιας επιλογής στο  $S$  (δηλαδή της εύρεσης της εγγραφής (εγγραφών) του  $S$  που ικανοποιούν τη συνθήκη)

# Συνένωση

## Σ3 Ταξινόμηση - Συγχώνευση

Έστω συνθήκη ισότητας

Ταξινόμηση τις πλειάδες της R στο γνώρισμα A

Ταξινόμηση τις πλειάδες της S στο γνώρισμα B

$i := 1; \quad j := 1;$

while ( $i \leq n_R$  and  $j \leq n_S$ )

if ( $R_i[A] < S_j[B]$ )

$i := i + 1;$  (\*προχώρησε το δείκτη στην R \*)

if ( $R_i[A] > S_j[B]$ )

$j := j + 1;$  (\* προχώρησε το δείκτη στην S\*)

# Συνένωση

else (\*  $R_i[A] = S_j[B]$  \*)

πρόσθεσε το  $R_i \cdot S_j$  στο αποτέλεσμα

$k := j + 1$ ; (\* γράψε και τις άλλες πλειάδες της  $S$  που ταιριάζουν, αν υπάρχουν \*)

while (( $k \leq n_S$ ) and ( $R_i[A] = S_k[B]$ ))

πρόσθεσε το  $R_i \cdot S_k$  στο αποτέλεσμα

$k := k + 1$ ;

$m := i + 1$ ; (\* γράψε και τις άλλες πλειάδες της  $R$  που ταιριάζουν, αν υπάρχουν \*)

while (( $m \leq n_R$ ) and ( $R_m[A] = S_j[B]$ ))

πρόσθεσε το  $R_m \cdot S_j$  στο αποτέλεσμα

$m := m + 1$ ;

$i := m$ ;  $j := k$ ;



# Συνένωση

Αν αγνοήσουμε τη ταξινόμηση για τη συγχώνευση (merge) απλή σάρωση των δύο αρχείων:

$$b_R + b_S$$

Ταξινόμηση:  $b_R * \log(b_R) + b_S * \log(b_S)$

# Πράξεις συνόλων

- $R \cup S$  (ένωση)
- $R \cap S$  (τομή)
- $R - S$  (διαφορά)

Θα δούμε έναν αλγόριθμο βασισμένο σε merge-sort (ταξινόμηση-συγχώνευση)

# Πράξεις συνόλων

Ταξινομήσε τις πλειάδες της  $R$  σε ένα γνώρισμα (έστω  $A$ )

Ταξινομήσε τις πλειάδες της  $S$  στο ίδιο γνώρισμα

$i := 1; \quad j := 1;$

while ( $i \leq n_R$  and  $j \leq n_S$ )

if ( $R_i[A] > S_j[A]$ )

Τομή

τίποτα

Ένωση

γράψε το  $S_j$  στο  
αποτέλεσμα

Διαφορά

τίποτα

$j := j + 1$

# Πράξεις συνόλων

else if ( $R_i[A] < S_j[A]$ )

Τομή

τίποτα

Ένωση

γράψε το  $R_i$  στο αποτέλεσμα

Διαφορά

γράψε το  $R_i$  στο αποτέλεσμα

$i := i + 1$

else (\*  $R_i[A] = S_j[A]$  \*)

Τομή

γράψε το  $R_i$  στο αποτέλεσμα

$i := i + 1;$

$j := j + 1;$

Ένωση

$i := i + 1;$

Διαφορά

$i := i + 1;$

$j := j + 1;$

# Πράξεις συνόλων

Αν υπάρχουν ακόμα εγγραφές για κάποιο αρχείο:

## Ένωση

while ( $i \leq n_R$ )

    γράψε το  $R_i$  στο αποτέλεσμα

$i := i + 1$ ;

while ( $j \leq n_S$ )

    γράψε το  $S_j$  στο αποτέλεσμα

$j := j + 1$ ;

## Διαφορά

while ( $i \leq n_R$ )

    γράψε το  $R_i$  στο αποτέλεσμα

$i := i + 1$ ;

# Ερωτήσεις;