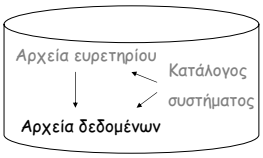
Επεξεργασία Ερωτήσεων

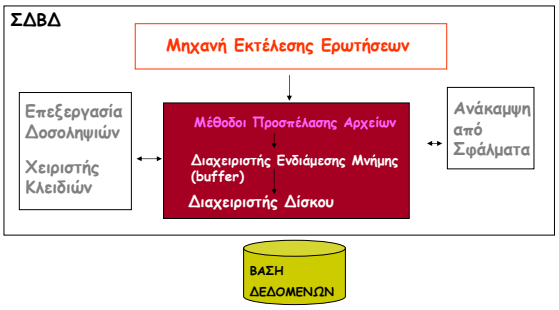
ΣΔΒΔ

Σύνολο από προγράμματα για τη διαχείριση της ΒΔ

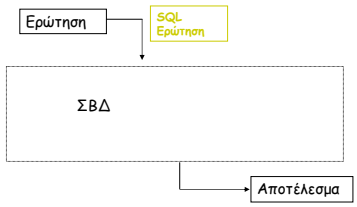


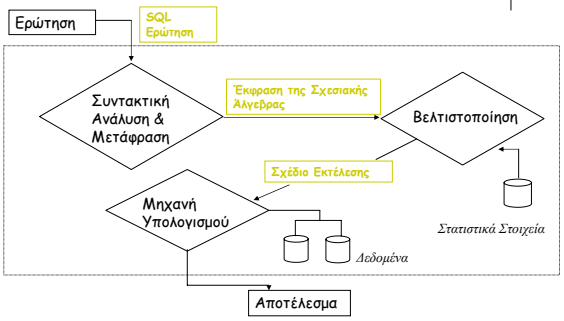
ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Σύστημα Βάσεων Δεδομένων (ΣΒΔ)



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 $A_1, A_2, \ldots, A_n$
from $R_1, R_2, \ldots, R_m$
where P

$\pi_{A1, A2, \ldots, An} (\sigma_P (R_1 \times R_2 \times \ldots \times R_m))$

2. Βελτιστοποίηση

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

select balance
from account
where balance < 25000

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

Με ποιο κριτήριο γίνεται η επιλογή της έκφρασης;

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

π.χ., για την υλοποίηση της επιλογής μπορεί

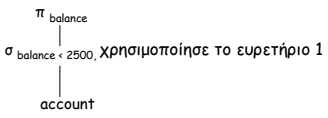
είτε να σαρώσουμε (scan) όλο το αρχείο ελέγχοντας κάθε εγγραφή αν ικανοποιεί τη συνθήκη

είτε αν υπάρχει π.χ., ένα B^+ 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



- Τα διαφορετικά σχέδια εκτέλεσης έχουν και διαφορικό κόστος
- Βελτιστοποίηση: η διαδικασία επιλογής του σχεδίου εκτέλεσης που έχει το μικρότερο κόστος
- Εκτίμηση του κόστους (συνήθως χρήση στατιστικών στοιχείων)

3. Εκτέλεση

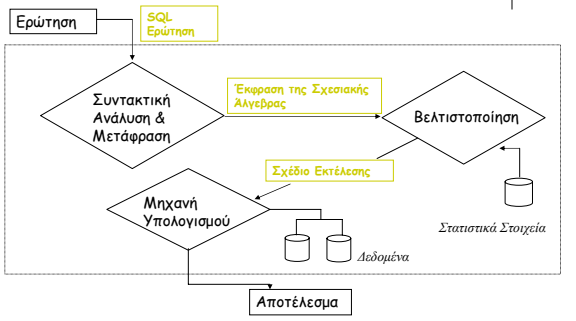
Μηχανή εκτέλεσης που εκτελεί τις βασικές πράξεις

Εκτέλεση

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

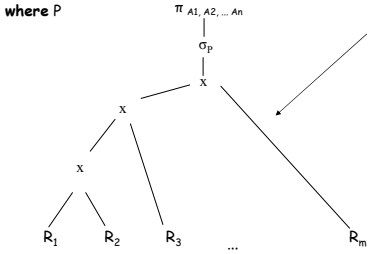
Επεξεργασία Ερωτήσεων (ανακεφαλαίωση)



Επεξεργασία Ερωτήσεων (ανακεφαλαίωση)

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

Μετάφραση → $\pi_{A1, A2, \ldots, An}(\sigma_P(R_1 \times R_2 \times \ldots \times R_m))$



Πλάνο εκτέλεσης (ποιες πράξεις και με ποιον αλγόριθμο)

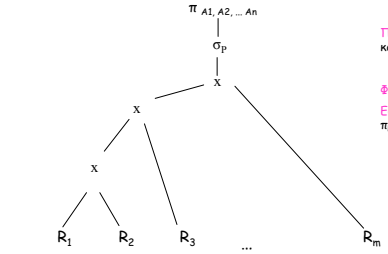
Βελτιστοποίηση

Εκτέλεση

Επιλογή κατάλληλου αλγορίθμου για κάθε βασική πράξη της σχεσιακής άλγεβρας

Επεξεργασία Ερωτήσεων

Λίγα λόγια για τη βελτιστοποίηση



Πλάνο εκτέλεσης (ποιες πράξεις και με ποιον αλγόριθμο)

Φύλλα: σχέσεις

Εσωτερικοί κόμβοι: βασικές πράξεις της σχεσιακής άλγεβρας

Βελτιστοποίηση του πλάνου

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

Βελτιστοποίηση Ερωτήσεων

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

  $|\pi_A(R)|$ -- 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;

Στατιστικά στοιχεία επίσης για το *αρχείο ευρετηρίου* (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 ($\sigma_{\Theta 1 \text{ AND } \Theta 2 \ldots \text{ AND } \Theta n}$ (R))
- Επιλογή με διάζευξη ($\sigma_{\Theta 1 \text{ OR } \Theta 2 \ldots \text{ OR } \Theta n}$ (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 Θi συνθήκη ισότητας σε ένα γνώρισμα υποψήφιο κλειδί $s_i = 1 / n_R$

Αν Θi 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

$\lceil \log (b_R) \rceil$ ← Εύρεση της πρώτης

\+

$\lceil SC(A, r)/f_R \rceil - 1$ ← 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

$HT_i + 1$ ← 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

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

$HT_i + 1$ ← Εύρεση και μεταφορά της πρώτης

Αν το A δεν είναι υποψήφιο κλειδί + 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

$\sigma_{A \le u} (R)$ ή $\sigma_{A \ge u} (R)$

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

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

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

Θα δούμε δύο αλγορίθμους:

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

$A \ge u$

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

$HT_i + \lceil c / f_R \rceil$

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

$A \le u$

Δε χρειάζεται ευρετήριο, γιατί;

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

$A \le u$ από την αρχή έως το u

$A \ge u$ από το u έως το τέλος

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

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

$HT_i + LB_i/2 + n_R/2$

Επιλογή - συνθήκη σύζευξης

$\sigma_{\Theta 1 \text{ AND } \Theta 2 \ldots \text{ AND } \Theta n}(R)$

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:

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

s_i επιλεκτικότητα της Θi

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 περισσοτέρων γνωρισμάτων που εμφανίζονται σε οποιαδήποτε απλές συνθήκες

Ε9 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

$\sigma_{\Theta 1 \text{ OR } \Theta 2 \ldots \text{OR } \Theta n}(R)$

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

Αλγόριθμους εκτέλεσης βασικών πράξεων

- Επιλογή
- Συνένωση
- Πράξεις συνόλων

Συνένωση

$R \bowtie_{R.A \text{ op } S.B} S$

Σ1 Εμφωλευμένος (εσωτερικός - εξωτερικός) βρόγχος

Σ2 Χρήση μιας δομής προσπέλασης

Σ3 Ταξινόμηση-Συγχώνευση

Σ4 Συνένωση με κατακερματισμό

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

Επιλεκτικότητα συνένωσης μιας σχέσης:

$$\frac{\text{το πλήθος των εγγραφών (πλειάδων) που επιλέγονται}}{\text{το πλήθος των εγγραφών (πλειάδων) του αρχείου (σχέσης)}}$$

• Σε ορισμένες περιπτώσεις μπορεί να δημιουργηθεί ένα ευρετήριο ειδικά για τη συνένωση

Σ1 Εμφωλευμένος (εσωτερικός-εξωτερικός) βρόγχος

Για κάθε εγγραφή t της R
  Για κάθε εγγραφή s της S
    Αν t[A] op s[B] πρόσθεσε το t s στο αποτέλεσμα

Αγνοώντας την εγγραφή των blocks του αποτελέσματος

$b_r + n_R * b_S$

Συνένωση (εμφωλευμένος βρόγχος)

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] op 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 (εμφωλευμένος βρόγχος)

Πριν θεωρήσαμε ότι έχουμε 2 block στη μνήμη (buffers) διαθέσιμους)

Αν υπάρχουν $n_B > 2$ **blocks στη μνήμη** που μπορεί να χρησιμοποιηθούν για τον υπολογισμό της συνένωσης συμφέρει να διαβάζουμε τα blocks της σχέσης του εξωτερικού βρόγχου ανά $n_B - 1$

Για κάθε $n_B - 1$ 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] op s[B] πρόσθεσε το t s στο αποτέλεσμα

$b_R + \lceil (b_R / (n_B - 1)) \rceil * b_S$

Συνένωση (χρήση ευρετηρίου)

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 (ταξινόμηση-συγχώνευση)

Σ3 Ταξινόμηση – Συγχώνευση

Έστω συνθήκη ισότητας

Ταξινόμησε τις πλειάδες της R στο γνώρισμα A
Ταξινόμησε τις πλειάδες της S στο γνώρισμα B
i : = 1; j := 1;
while ($i \le n_R$ and $j \le n_S$)
  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 (ταξινόμηση-συγχώνευση)

```
else  (*  R_i[A] = S_j[B]  *)
   πρόσθεσε το R_i . S_j στο αποτέλεσμα
   k := j + 1;  (* γράψε και τις άλλες πλειάδες της S που ταιριάζουν, αν υπάρχουν *)
        while ((k ≤ n_S) and (R_i[A] = S_k[B]))
              πρόσθεσε το R_i . S_k στο αποτέλεσμα
               k := k + 1;
   m := i + 1;  (* γράψε και τις άλλες πλειάδες της R που ταιριάζουν,
                   αν υπάρχουν *)
        while ((m ≤ n_R)  and (R_m[A] = S_j[B]))
              πρόσθεσε το R_m . S_j στο αποτέλεσμα
              m := m + 1;
   i := m; j := k;
```


Συνένωση (ταξινόμηση-συγχώνευση)

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

$b_R + b_S$

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

Σ4 Συνένωση με κατακερματισμό

(συνθήκη ισότητας)

- χωρίζουμε με βάση μια συνάρτηση κατακερματισμού h τις πλειάδες της S και της R σε *κάδους* -- στον ίδιο κάδο αν $h(t_R[A]) = h(t_S[B])$
- δηλαδή οι πλειάδες με $t_R[A] = t_S[B]$ πέφτουν στον ίδιο κάδο άρα αρκεί να ελέγξουμε μεταξύ τους τις πλειάδες που πέφτουν στον ίδιο κάδο

Κατακερμάτισε τις εγγραφές της R χρησιμοποιώντας την $h(t_R[A])$

Για κάθε εγγραφή t_S της S

$k := h(t_S[B])$

σύγκρινε το $t_S[B]$ με $t_{Ri}[A]$ για όλες τις εγγραφές t_{Ri} της R στον κάδο k

- Χρησιμοποιούμε την μικρότερη σχέση για το πρώτο πέρασμα.
- Αν όλοι οι κάδοι που προκύπτουν χωράνε στη μνήμη, κόστος $b_R + b_S$

Αν δεν χωρούν όλοι οι κάδοι – τροποποίηση

Ταξινόμησε τις πλειάδες της R σε ένα γνώρισμα (έστω A)

Ταξινόμησε τις πλειάδες της S στο ίδιο γνώρισμα

i : = 1; j := 1;

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

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

Τομή τίποτα

Ένωση γράψε το S_j στο αποτέλεσμα

Διαφορά τίποτα

j := j + 1

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;

Αν υπάρχουν ακόμα εγγραφές για κάποιο αρχείο:

Ένωση

while ($i \le n_R$)

γράψε το R_i στο αποτέλεσμα

i: = i + 1;

while ($j \le n_S$)

γράψε το S_j στο αποτέλεσμα

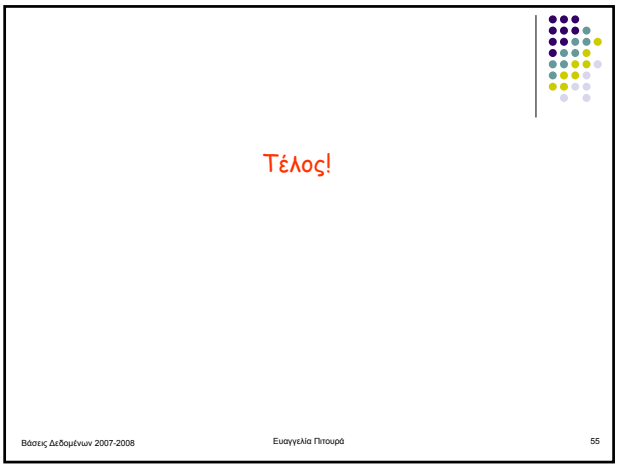
j: = j + 1;

Διαφορά

while ($i \le n_R$)

γράψε το R_i στο αποτέλεσμα

i: = i + 1;


Τέλος!
Βάσεις Δεδομένων 2007-2008
Ευαγγελία Πιτουρά
55