Επεξεργασία Ερωτήσεων

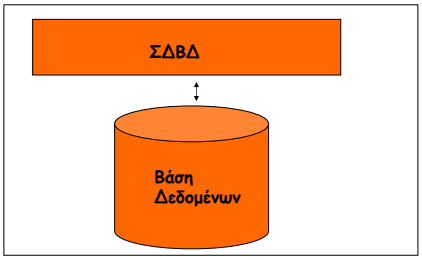
ΜΕΡΟΣ 1

Μοντελοποίηση (Μοντέλο Ο/Σ, Σχεσιακό, Λογικός Σχεδιασμός)
Προγραμματισμός (Σχεσιακή Άλγεβρα, SQL)
Δημιουργία/Κατασκευή
Εισαγωγή Δεδομένων
Επεξεργασία Δεδομένων

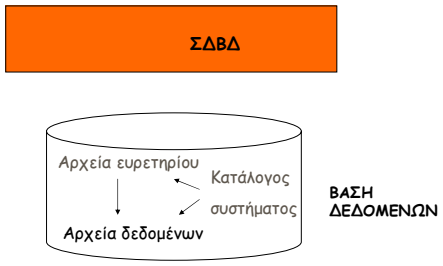
Με χρήση ΣΔΒΔ

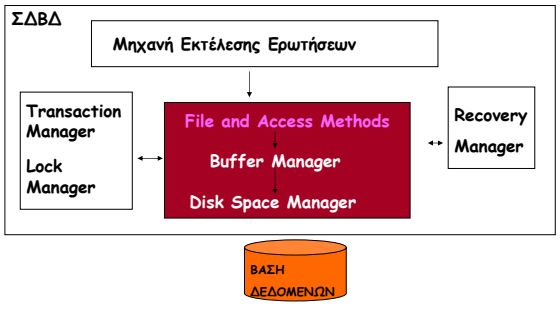
ΜΕΡΟΣ 2 *(Υλοποίηση ΣΔΒΔ)*
Αποθήκευση
Ευρετήρια
Εκτέλεση Ερωτήσεων

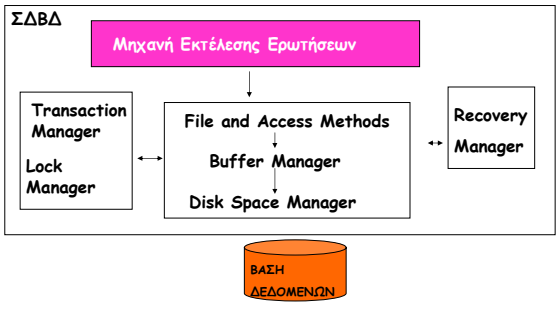
Το εσωτερικό ενός ΣΔΒΔ

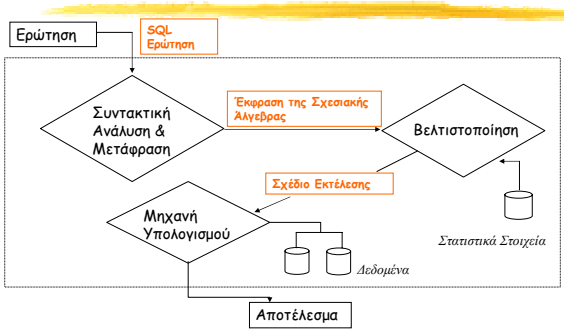


Η Δομή ενός ΣΔΒΔ









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 $A_1, A_2, ..., A_n$
from $R_1, R_2, ..., R_m$
where P

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

2. Βελτιστοποίηση

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

select balance
from account
where balance < 25000

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

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

$\pi_{balance}$
|
$\sigma_{balance < 2500,}$ χρησιμοποίησε το ευρετήριο 1
|
account

• 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

Τι θα καλύψουμε στο μάθημα:

1. Αλγόριθμους εκτέλεσης βασικών πράξεων
   - Επιλογή
   - Προβολή
   - Πράξεις συνόλων
   - Συνένωση

Εκτίμηση κόστους – μερικά στατιστικά σχετικά με το αρχείο δεδομένων

- 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 $b_R = \lceil n_R / f_R \rceil$
- V(A, R): αριθμός διαφορετικών τιμών του A

  $| \pi_A(R) |$ -- αν 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;

Εκτίμηση κόστους - αρχείο ευρετηρίου

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

Κόστος: Αριθμό blocks που μεταφέρονται

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

Θα εξετάσουμε:

- Επιλογή με συνθήκη ισότητας ($\sigma_{A = a}$ (R))
- Επιλογή με συνθήκη σύγκρισης – διαστήματος/περιοχής (range query) ($\sigma_{A \le u}$ (R)) ή ($\sigma_{A \ge u}$ (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))

Πιθανοί αλγόριθμοι εκτέλεσης:

E1 Σειριακή αναζήτηση

E2 Δυαδική αναζήτηση

E3 Χρήση πρωτεύοντος ευρετηρίου/κατακερματισμού

E4 Χρήση δευτερεύοντος ευρετηρίου/κατακερματισμού

Για την περίπτωση E3 και E4 λέμε ότι έχουμε μονοπάτι προσπέλασης (access path)

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

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 στα ίδια γειτονικά blocks

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

$HT_i + SC(A, R)$ ← 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)

Ε1 Σειριακή αναζήτηση

Ε2 Δυαδική αναζήτηση

Ε5 Χρήση πρωτεύοντος (πολυεπίπεδου) ευρετηρίου

Πρωτεύον, σημαίνει ταξινομημένο αρχείο, έστω σε αύξουσα διάταξη

A ≥ u

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

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

A ≤ u

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

Δε χρειάζεται ευρετήριο, γιατί;

Ε6 Χρήση δευτερεύοντος (πολυεπίπεδου) ευρετηρίου

- Εύρεση του πρώτου φύλλου του ευρετηρίου
- Για κάθε block (φύλλο) του ευρετηρίου διάβασε το αντίστοιχο block δεδομένων

(σημείωση, χρησιμοποιούμε το δείκτη ανάμεσα στα φύλλα)

Σάρωση των φύλλων του δέντρου

A ≤ u από την αρχή έως το u

A ≥ 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

Αν $c = n_R/2$, 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

Ε7 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

Ε8 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 κάποια από τις συνθήκες δεν έχει διαδρομή προσπέλασης
-> σάρωση όλου του αρχείου

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

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

$n_R * b_S + b_R$

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

$b_R * b_S + b_R$

Συμφέρει η τοποθέτηση της *μικρότερης* σχέσης στο 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

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

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]

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

Συνένωση

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

Συνένωση (ταξινόμηση-συγχώνευση)

Σ3 Ταξινόμηση – Συγχώνευση

Ταξινόμησε τις πλειάδες της R στο γνώρισμα A
Ταξινόμησε τις πλειάδες της S στο γνώρισμα B
i : = 1; j := 1;
while (i ≤ n_R and j ≤ 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
                k := k + 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

- 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

Ταξινόμησε τις πλειάδες της S στο ίδιο γνώρισμα

$i := 1;$ $j := 1;$

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

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

Τομή	Ένωση	Διαφορά
τίποτα	γράψε το S_j στο αποτέλεσμα	τίποτα

$j := j + 1$

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

Τομή	Ένωση	Διαφορά
τίποτα	γράψε το R_i στο αποτέλεσμα	γράψε το R_i στο αποτέλεσμα

$i := i + 1$

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

Τομή	Ένωση	Διαφορά
γράψε το R_i στο αποτέλεσμα $i := i + 1;$ $j := j + 1;$	$i := i + 1;$	$i := i + 1;$ $j := j + 1;$

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
  $i := i + 1;$