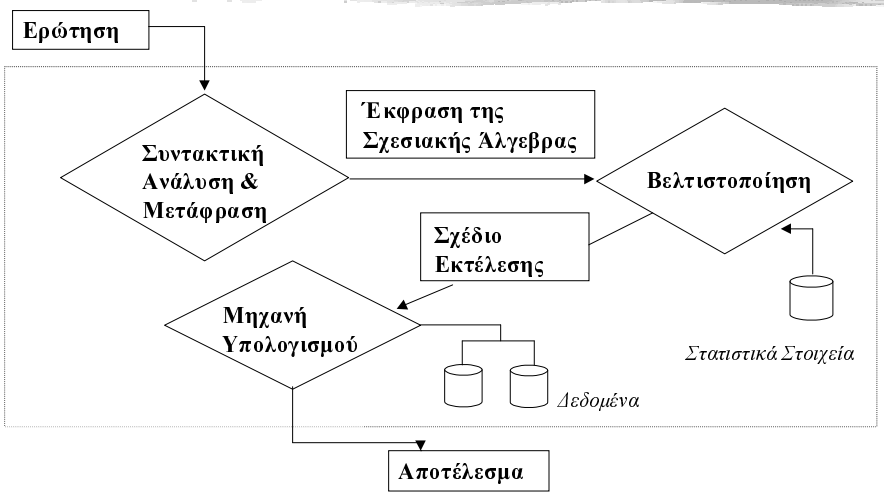
Επεξεργασία Ερωτήσεων

Επεξεργασία Ερωτήσεων



Επεξεργασία Ερωτήσεων

Τα βασικά βήματα στην επεξεργασία μιας ερώτησης είναι

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

Συντακτική Ανάλυση & Μετάφραση

1. Συντακτική Ανάλυση (Parsing) & Μετάφραση

Η 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

Εσωτερική μορφή; Έκφραση της σχεσιακής άλγεβρας

2. Βελτιστοποίηση

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

select balance
from account
where balance < 25000

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

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

π.χ., για την υλοποίηση της επιλογής μπορεί

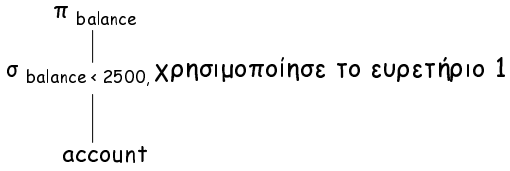
είτε να σαρώσουμε (scan) όλο το αρχείο ελέγχοντας κάθε εγγραφή

είτε αν υπάρχει π.χ., ένα B^+ ευρετήριο στο γνώρισμα balance να χρησιμοποιήσουμε το ευρετήριο

Βελτιστοποίηση

Κάθε πράξη της σχεσιακής άλγεβρας μπορεί να υλοποιηθεί με διαφορετικούς αλγορίθμους: βασικές (primitive) πράξεις (πράξη + αλγόριθμος)

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



Βελτιστοποίηση

- 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

1. Αλγόριθμους εκτέλεσης βασικών πράξεων

2. Ισοδυναμία σχεσιακών εκφράσεων - ευριστική επιλογή

Αλγόριθμοι Εκτέλεσης Βασικών Πράξεων

Εκτίμηση κόστους - 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

  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;

Αλγόριθμοι Εκτέλεσης Βασικών Πράξεων

Εκτίμηση κόστους - αρχείο ευρετηρίου

- f_i : παράγοντας διακλάδωσης,

  πολυεπίπεδο f_0, B^+ δέντρο ~ βαθμό
- H_i: αριθμός επιπέδων
- LB_i: αριθμός block φύλλων

Αριθμό blocks που μεταφέρονται

Επιλογή

Ε1 Σειριακή αναζήτηση

Ε2 Δυαδική αναζήτηση

Ε3 Χρήση πρωτεύοντος ευρετηρίου/ κατακερματισμού

Ε4 Χρήση δευτερεύοντος ευρετηρίου/ κατακερματισμού

Μονοπάτι προσπέλασης

Επιλογή - συνθήκη ισότητας

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

Ε1 Σειριακή αναζήτηση

b_R

b_R / 2 αν το Α υποψήφιο κλειδί

Μπορεί να χρησιμοποιηθεί σε οποιοδήποτε αρχείο

Ε2 Δυαδική αναζήτηση

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 Α υποψήφιο κλειδί;

Ε3 Χρήση πρωτεύοντος ευρετηρίου

Μπορεί να χρησιμοποιηθεί μόνο αν υπάρχει πρωτεύον ευρετήριο στο Α

$HT_i + 1$ ← Εύρεση και μεταφορά της πρώτης

Αν το Α δεν είναι υποψήφιο κλειδί -- ευρετήριο συστάδων

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

Ε4 Χρήση δευτερεύοντος ευρετηρίου

Μπορεί να χρησιμοποιηθεί μόνο αν υπάρχει δευτερεύον ευρετήριο στο A

Αν το A είναι υποψήφιο κλειδί

$HT_i + 1$

Αν το A δεν είναι υποψήφιο κλειδί

$HT_i + SC(A, R)$ Στη χειρότερη περίπτωση κάθε εγγραφή που ικανοποιεί τη συνθήκη σε διαφορετικό block

Επιλογή - συνθήκη με σύγκριση

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

Έστω ότι c πλειάδες ικανοποιούν τη συνθήκη

Γενικά $c = n_r / 2$

Έστω min, max, ομοιόμορφη κατανομή και $\sigma_{A \leq u}(R)$

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

Επιλογή: Συνθήκη με Σύγκριση

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

Δ1 Σειριακή αναζήτηση; Δ2 Δυαδική αναζήτηση;

Επιλογή: Συνθήκη με Σύγκριση

Ε5 Χρήση πρωτεύοντος ευρετηρίου

$A \geq u$

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;

Ε6 Χρήση δευτερεύοντος ευρετηρίου

Σάρωση των φύλλων του δέντρου

A ≤ **u** από την αρχή έως το u

A ≥ **u** από το u έως το τέλος

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:

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

Επιλογή: Συνθήκη Σύζευξης

- 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, $s_i = k / n_R$

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

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

Επιλογή: Συνθήκη Σύζευξης

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

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

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

$$R \triangleright\triangleleft_{R.A \text{ op } S.B} S$$

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 Br της R
    Για κάθε block Bs της S
        Για κάθε εγγραφή t του Br
            Για κάθε εγγραφή s του Bs
                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

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

Σ2 Χρήση μιας δομής προσπέλασης

Η σχέση για την οποία υπάρχει ευρετήριο τοποθετείται στο 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
            Χρησιμοποιείσαι 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

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

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;
        if (R_i[A] > S_j[B])
                j := j + 1;
```

Συνένωση

```
else   (* Ri[A] = Sj[B] *)
    πρόσθεσε το Ri . Sj στο αποτέλεσμα
    k := j + 1;  (* γράψε και τις άλλες πλειάδες της S που ταιριάζουν, αν υπάρχουν *)
        while ((k ≤ nS) and (Ri[A] = Sk[B]))
                πρόσθεσε το Ri . Sk στο αποτέλεσμα
                 k := k + 1;
     m := i + 1;  (* γράψε και τις άλλες πλειάδες της R που ταιριάζουν,
                          αν υπάρχουν *)
        while ((m ≤ nR)  and (Rm[A] = Sj[B]))
                πρόσθεσε το Rm . Sj στο αποτέλεσμα
                k := k + 1;
  i := m; j := k;
```


Συνένωση

Αν αγνοήσουμε τη ταξινόμηση για τη σάρωση:

$b_R + b_S$

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 $b_R + b_S$

Συνένωση

Έστω ότι δε χωρούν όλοι οι κάδοι.

Έστω M καταχωρητές, N κάδοι, και η συνάρτηση h(k) = k mod N

(* Κατακερμάτισε τις εγγραφές της R χρησιμοποιώντας την $h(t_R[A])$ - στη μνήμη όλοι οι κάδοι του πρώτου κάδου και ένα block για τους υπόλοιπους *)

Για κάθε t_R της R

k := $h(t_R[A])$

if (k > 1 και το block του k-οστού κάδου είναι γεμάτο)

μετέφερε το block στη δίσκο

Συνένωση

(* στη μνήμη όλα τα blocks του 1ου κάδου - γράψε τα blocks των άλλων κάδων στο δίσκο *)

Για κάθε εγγραφή t_S της S

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

αν k = 1 σύγκρινε το $t_S[B]$ με όλες τις εγγραφές t_{Ri} στον κάδο 1

αν k > 1, γράψε την εγγραφή στο δίσκο

Συνένωση

(* μετά το πρώτο βήμα - έχει γίνει συνένωση όλων των εγγραφών του πρώτου κάδου και στο δίσκο υπάρχουν N - 1 κάδοι της R και N - 1 κάδοι της S *)

i := 2; j := 1;

while (i ≤ N)

φέρε στη μνήμη όλα τα blocks του i-οστού κάδου

while (j ≤ # blocks του i-οστού κάδου του αρχείου S)

φέρε στη μνήμη το j-οστό block B_j του i-οστού κάδου του αρχείου S

Για κάθε εγγραφή t_S του B_j

σύγκρινε το $t_S[B]$ με όλες τις εγγραφές t_{Ri} στον κάδο i

Πράξεις Συνόλων

Ταξινόμησε τις πλειάδες της R σε ένα γνώρισμα A

Ταξινόμησε τις πλειάδες της S στο ίδιο γνώρισμα

i : = 1; j := 1;

while (i ≤ n_R and j ≤ n_S)

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

Τομή	**Ένωση**	**Διαφορά**
τίποτα	γράψε το S_j στο αποτέλεσμα	τίποτα

j := j + 1

Πράξεις Συνόλων

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

Τομή
τίποτα

Ένωση
γράψε το R_i στο αποτέλεσμα

Διαφορά
γράψε το R_i στο αποτέλεσμα

j := j + 1

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

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

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;

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

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

Δέντρο ερώτησης

Φύλλα: σχέσεις εισόδου
Εσωτερικοί κόμβοι: πράξεις της σχεσιακής άλγεβρας

Εκτέλεση δέντρου ερώτησης

Βελτιστοποίηση Ερωτήσεων

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