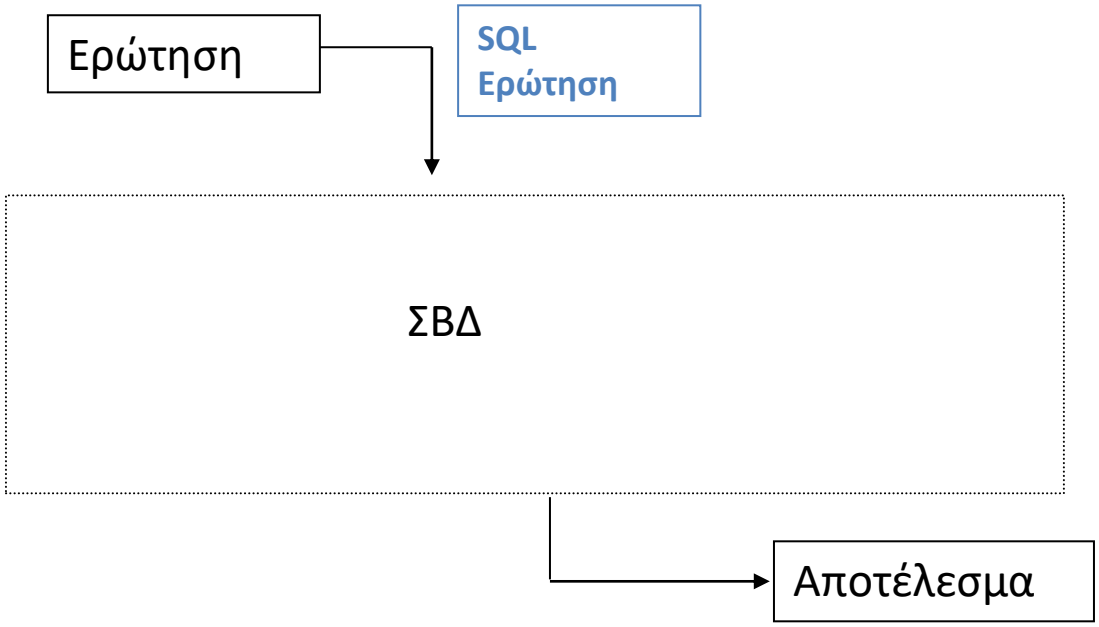
Εισαγωγή στην Επεξεργασία Ερωτήσεων

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 $A_1, A_2, \ldots, A_n$

FROM $R_1, R_2, \ldots, R_m$

WHERE P

$$\boldsymbol{\pi}_{A1, A2, \ldots, An} (\boldsymbol{\sigma}_{P} (R_1 \mathbf{x} R_2 \mathbf{x} \ldots \mathbf{x} R_m))$$

Παράδειγμα

```
SELECT R.A, T.D
FROM R, S, T
WHERE R.B = S.B
  AND S.C = T.C
  AND R.A < 10;
```

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 ευριστικούς στη συνέχεια*

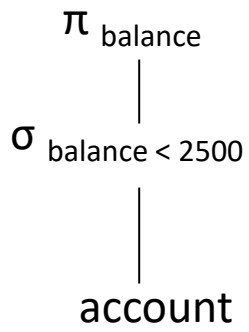
Πλάνο Εκτέλεσης

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

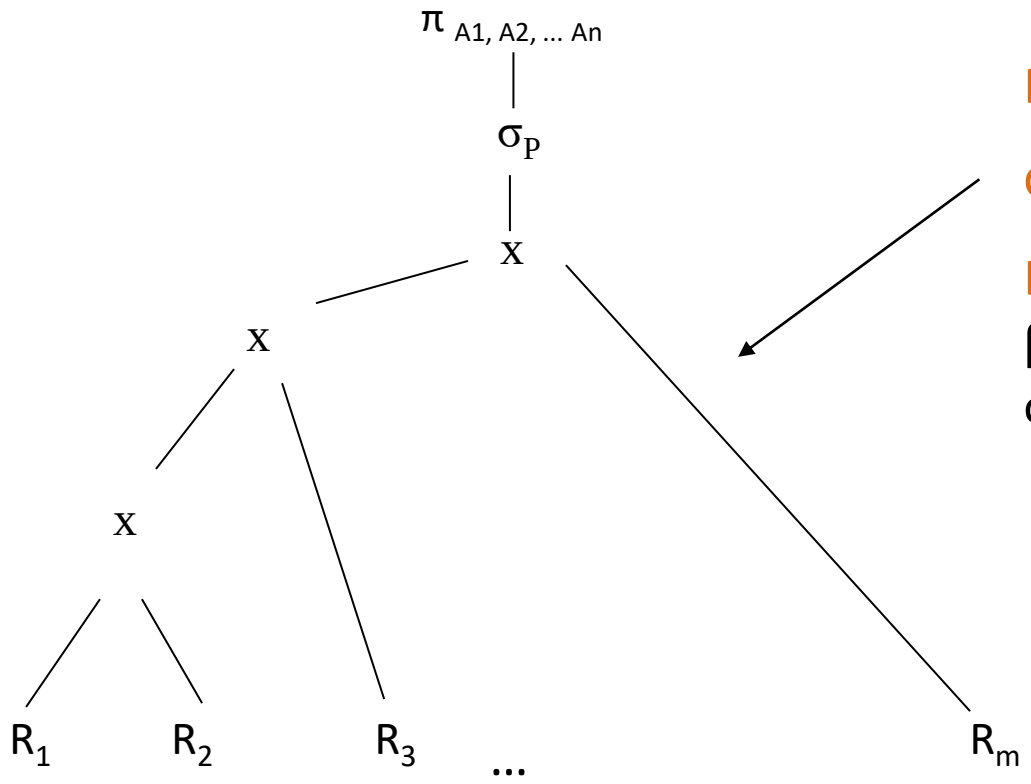
SELECT $A_1, A_2, \ldots, A_n$

FROM $R_1, R_2, \ldots, R_m$

WHERE P

Μετάφραση

$\boldsymbol{\pi}_{A1, A2, \ldots, An} (\boldsymbol{\sigma}_{P} (R_1 \mathbf{x} R_2 \mathbf{x} \ldots \mathbf{x} R_m))$



Πλάνο εκτέλεσης

Φύλλα: σχέσεις

Εσωτερικοί κόμβοι: βασικές πράξεις της σχεσιακής άλγεβρας

Βελτιστοποίηση του πλάνου

Βελτιστοποίηση

- Τα διαφορετικά πλάνα εκτέλεσης έχουν και διαφορικό 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

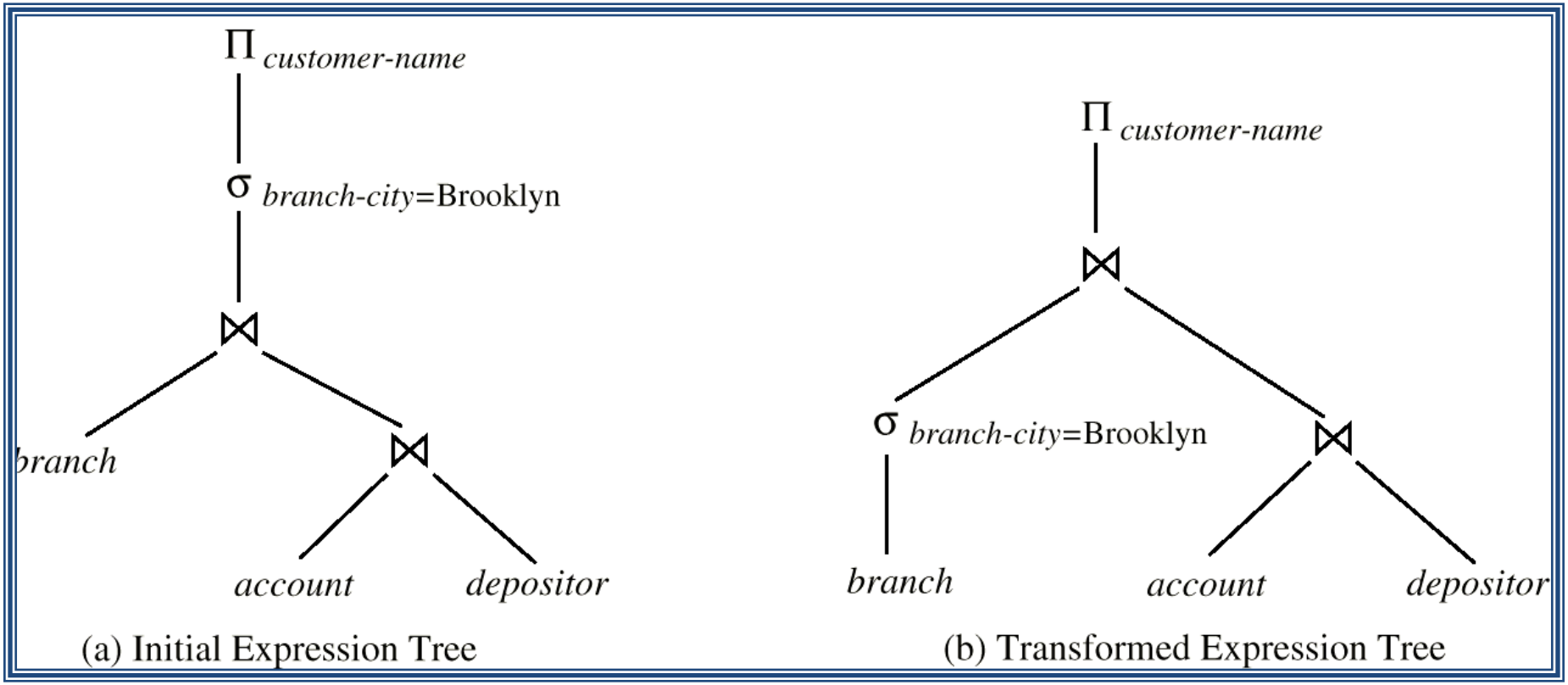
Ευριστικοί Κανόνες Βελτιστοποίησης Πλάνου Εκτέλεσης

4. Συνδυασμός μιας πράξης καρτεσιανού γινομένου με μια πράξη επιλογής που ακολουθεί

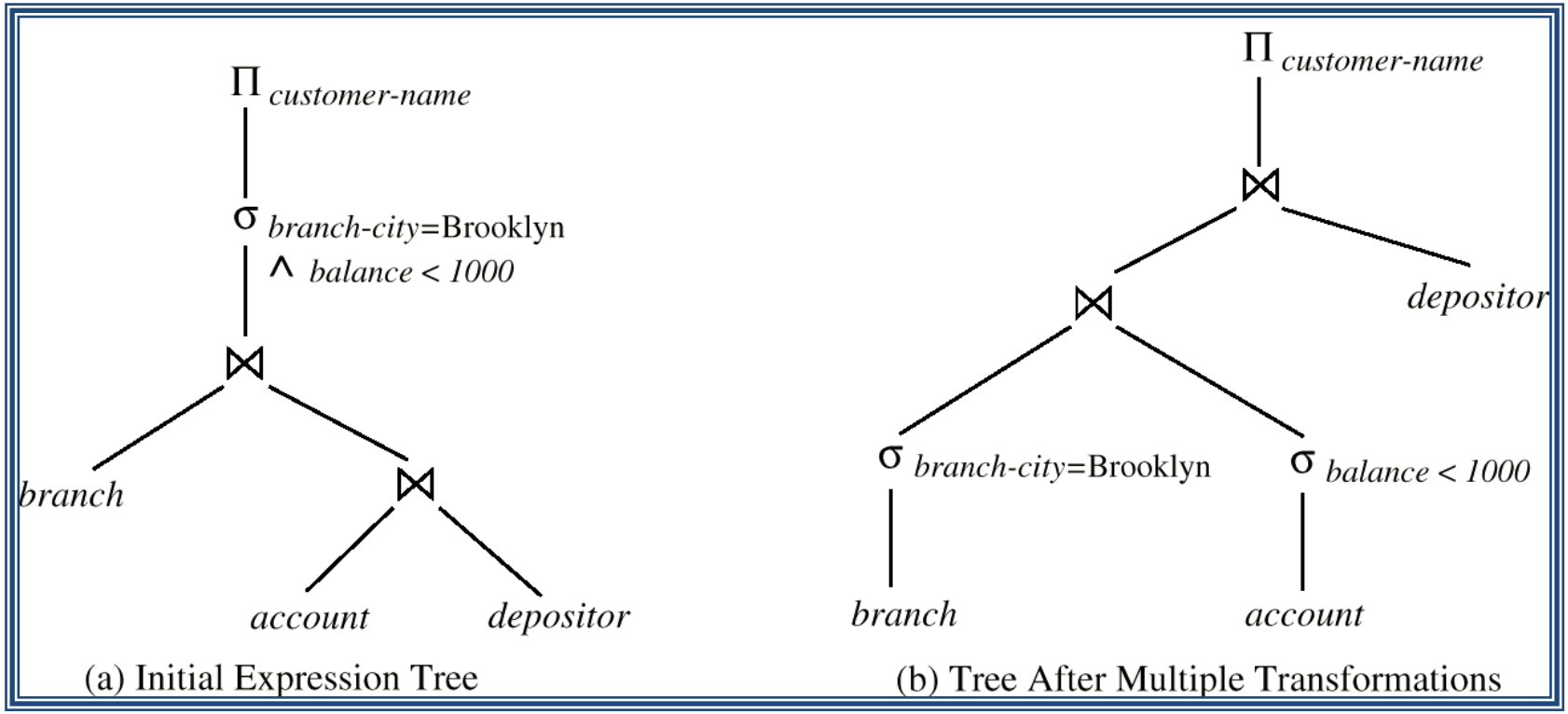
5. Διάσπαση και *μετακίνηση των λιστών προβολής όσο πιο κάτω* γίνεται στο δέντρο

6. Εντοπισμός υποδέντρων με ομάδες πράξεων που μπορεί να εκτελεστούν με κοινό αλγόριθμο

Παράδειγμα


Π customer-name
σ branch-city=Brooklyn
⋈
branch
⋈
account
depositor
(a) Initial Expression Tree
Π customer-name
⋈
σ branch-city=Brooklyn
⋈
branch
account
depositor
(b) Transformed Expression Tree

Παράδειγμα


Π customer-name
σ branch-city=Brooklyn
∧ balance < 1000
⋈
branch
⋈
account
depositor
(a) Initial Expression Tree
Π customer-name
⋈
⋈
depositor
σ branch-city=Brooklyn
σ balance < 1000
branch
account
(b) Tree After Multiple Transformations

Συνενώσεις

Σειρά εκτέλεσης συνένωσης με χρήση της commutativity (αντιμεταθετικής) και associativity (προσεταιριστικής) ιδιότητας

Για n σχέσεις -> 2^n επιλογές

Με βάση την επιλεκτικότητα: πρώτα η συνένωση που δίνει το μικρότερο αποτέλεσμα
Σύμβαση: Η σχέση στα αριστερά αντιστοιχεί στην εξωτερική σχέση της συνένωσης

Ειδικές διατάξεις
Left-deep join tree (η δεξιά είναι πάντα σχέση (όχι ενδιάμεσο αποτέλεσμα))
Right-deep join tree
Bushy

Παράδειγμα

R(A,B) S(B,C) T(C,D)

```
SELECT R.A, T.D
FROM R, S, T
WHERE R.B = S.B
  AND S.C = T.C
  AND R.A < 10;
```

Φυσικό Πλάνο Εκτέλεσης

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

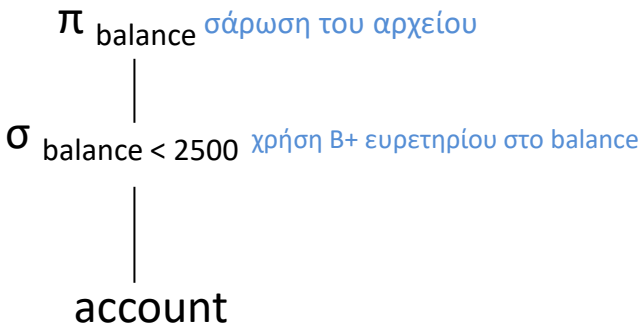
π.χ., για την υλοποίηση της επιλογής μπορεί για παράδειγμα:

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

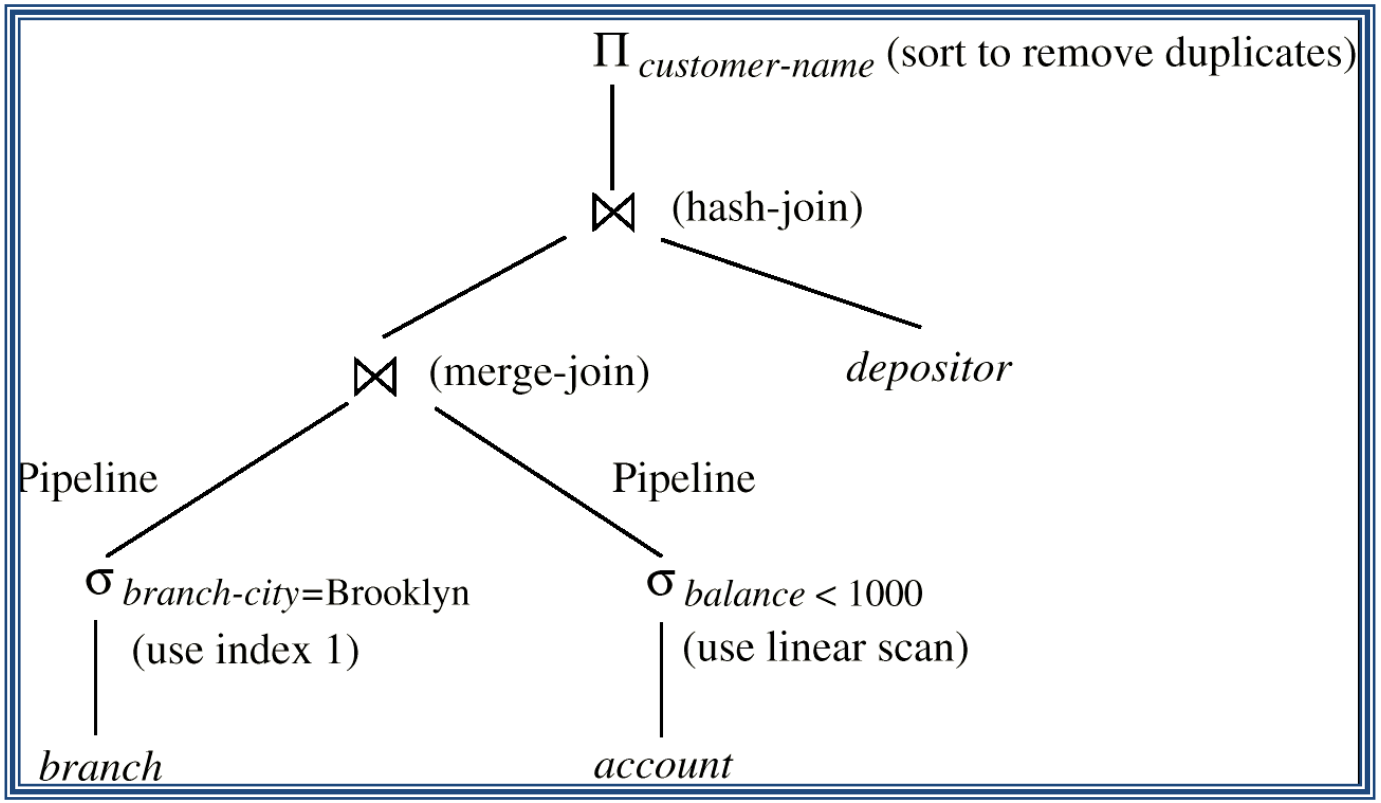
Άρα δεν αρκεί ο προσδιορισμός της πράξης - πρέπει να προσδιορίζεται **και** ο αλγόριθμος που θα χρησιμοποιηθεί για την υλοποίησή της

Φυσικό Πλάνο Εκτέλεσης

Λογικό πλάνο εκτέλεσης – μόνο τις πράξεις
Φυσικό πλάνο εκτέλεσης – περιλαμβάνει και τον αλγόριθμο που θα χρησιμοποιηθεί



Παράδειγμα


Π customer-name (sort to remove duplicates)
⋈ (hash-join)
⋈ (merge-join)
depositor
Pipeline
Pipeline
σ branch-city=Brooklyn
(use index 1)
σ balance < 1000
(use linear scan)
branch
account

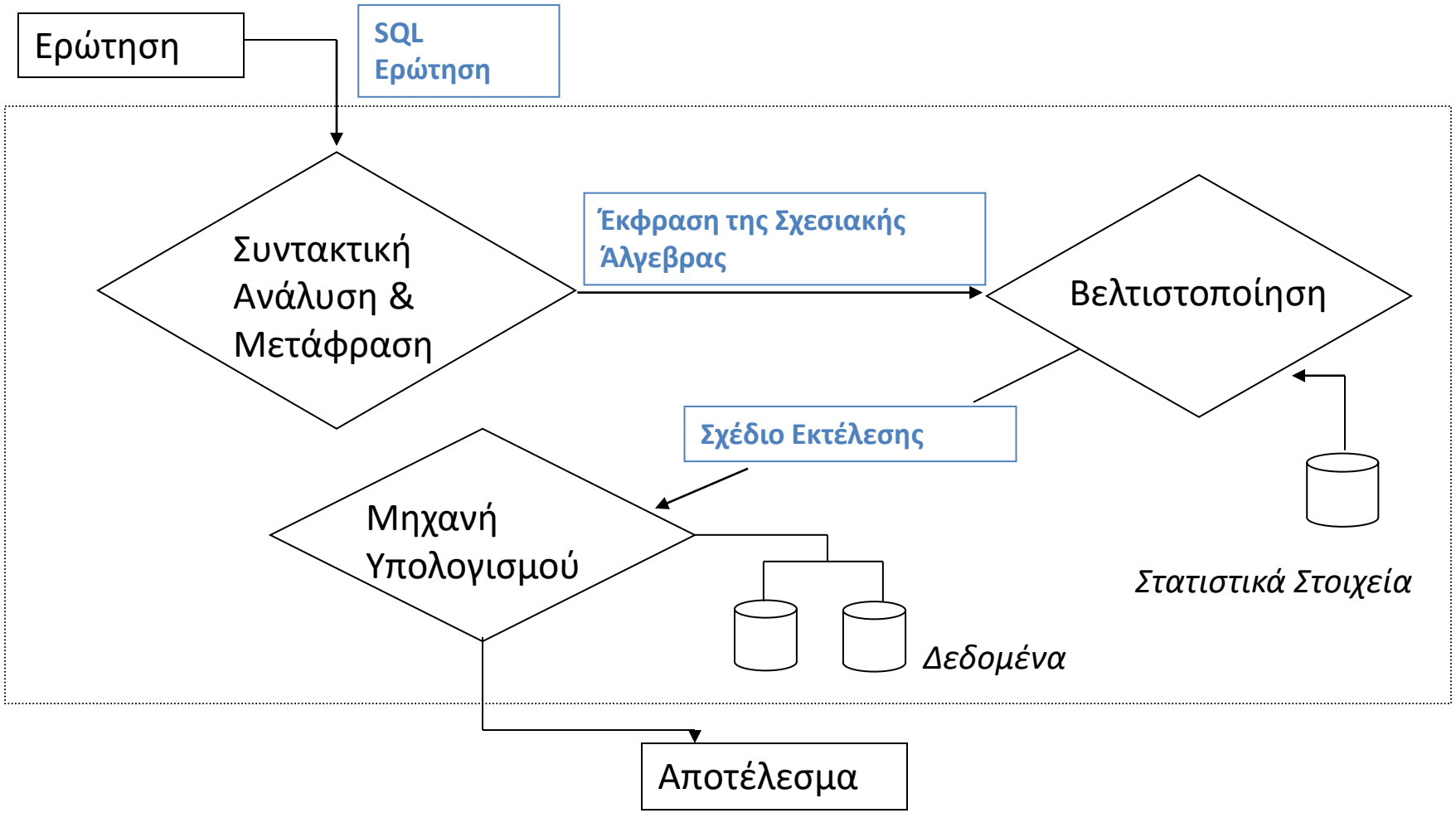
Εκτέλεση Ερωτήσεων

Μηχανή εκτέλεσης που εκτελεί τις βασικές πράξεις

- Υπάρχουν υλοποιημένοι μια σειρά αλγορίθμων για κάθε βασική πράξη (π.χ., που χρησιμοποιούν ή όχι ευρετήρια κλπ)
- Το ΣΔΒΔ κάνει μια *εκτίμηση του κόστους* και *επιλέγει* για κάθε πράξη *τον αλγόριθμο* με τον μικρότερο (με βάση την εκτίμηση) κόστος

- Η εκτίμηση του κόστους γίνεται χρησιμοποιώντας 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 πράξεων 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,
  - Πολυεπίπεδο f_0, B^+ δέντρο ~ 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*:

Ε1: Σειριακή αναζήτηση (σάρωση, scan)

Ε2: Δυαδική αναζήτηση (αν το αρχείο είναι ταξινομημένο)

Ε3: Χρήση πρωτεύοντος ευρετηρίου/κατακερματισμού (αν υπάρχει)

Ε4: 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 (σάρωση)

Διάβασμα (scan) όλου του αρχείου

b_R: αριθμός blocks της σχέσης R

b_R

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

Αν όχι κλειδί, πρέπει να βρούμε όλες τις πλειάδες με τιμή α

Μπορεί να χρησιμοποιηθεί σε οποιοδήποτε αρχείο

Επιλογή – συνθήκη ισότητας

Ε2 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 Α (δηλαδή, 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;

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 Α

$HT_i + 1$ ⟵ Εύρεση και μεταφορά της πρώτης

Αν το Α δεν είναι υποψήφιο κλειδί -- ευρετήριο συστάδων

$HT_i + \lceil SC(A, R)/f_R \rceil$ ⟵ Εύρεση και των υπόλοιπων

ΣΗΜΕΙΩΣΗ: Πρωτεύον ευρετήριο στο Α, σημαίνει ότι οι εγγραφές του αρχείου δεδομένων είναι ταξινομημένες (διατεταγμένες) ως προς Α άρα οι υπόλοιπες εγγραφές με την ίδια τιμή (αν υπάρχουν) βρίσκονται σε γειτονικά blocks του αρχείου δεδομένων

Επιλογή – συνθήκη ισότητας

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

Ε4 Χρήση δευτερεύοντος δεντρικού ευρετηρίου

Μπορεί να χρησιμοποιηθεί μόνο αν υπάρχει τέτοιο ευρετήριο στο Α

Αν το Α είναι υποψήφιο κλειδί

$HT_i + 1$ ⟵ Εύρεση και μεταφορά της πρώτης

Αν το Α 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

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

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

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

Έστω *αύξουσα διάταξη*

Σειριακή ανάγνωση

Από το 1ο block του αρχείου έως την πρώτη εγγραφή με A > u

Κόστος?

Επιλογή – συνθήκη με σύγκριση

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

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

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

$$\sigma_{P1 \textbf{ AND } P2 \textbf{ ... AND } Pn}(R)$$

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

Αν υπάρχουν *παραπάνω από ένα* ευρετήρια μπορούμε επίσης να υπολογίσουμε πρώτα την τομή των blocks που επιστρέφουν ως ταίριασμα

Επιλογή με διάζευξη

$$\sigma_{P1 \textbf{ OR } P2 \textbf{ ... OR } Pn}(R)$$

Αν έστω και μία 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

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] op 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] 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

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 Διάταξη - Συγχώνευση

Έστω συνθήκη ισότητας $R \bowtie_{R.A = S.B} S$

Διάταξε τις πλειάδες της R στο γνώρισμα A (έστω αύξουσα)

Διάταξε τις πλειάδες της S στο γνώρισμα B (έστω αύξουσα)

i : = 1; 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 . 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 . 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 . S_j στο αποτέλεσμα

m := m + 1;

i := m; j := k;

Συνένωση

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

$b_R + b_S$

Κόστος Διάταξης: $b_R * \log(b_R) + b_S * \log(b_S)$

Πράξεις συνόλων

- R ∪ S (ένωση)
- R ∩ S (τομή)
- R – S (διαφορά)

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

Πράξεις συνόλων

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

Διάταξε 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[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;

Ασκήσεις

Άσκηση 1

Θεωρείστε ότι τον πίνακα BOOK
BOOK(ISBN, TITLE, PUB-YEAR)
που έχει πληροφορία για 1.000.000 βιβλία και είναι αποθηκευμένος σε ένα αρχείο στο δίσκο το οποίο είναι διατεταγμένο ως προς το γνώρισμα TITLE και καταλαμβάνει 20.000 blocks.

Επίσης, έχουμε ένα B+-δέντρο ως ευρετήριο στο γνώρισμα ISBN που έχει τάξη 55 για τους εσωτερικούς κόμβους και 65 για τα φύλλα. Θεωρείστε ότι μπορείτε να χρησιμοποιείστε κάποια blocks στη μνήμη για την αποθήκευση του ευρετηρίου.

(i) Πόσα blocks στην μνήμη επαρκούν για την αποθήκευση των δύο πρώτων επιπέδων; Απαντήστε τα επόμενα ερωτήματα υποθέτοντας ότι τα δύο πρώτα επίπεδα του B+-δέντρου είναι στη μνήμη.

(ii) Εκτιμήστε το κόστος της ερώτησης:

```
SELECT * FROM BOOK WHERE ISBN = 2101010;
```

(iii) Θεωρείστε την ερώτηση

```
SELECT * FROM BOOK
WHERE ISBN > 1451010 AND ISBN < 8899000 and TITLE = 'SteppenWolf';
```

και ότι υπάρχουν 100 βιβλία με ISBN μεταξύ 1451010 και 8899000 και 2 βιβλία με τίτλο SteppenWolf.
Συμφέρει να χρησιμοποιήσουμε το ευρετήριο για αυτήν την ερώτηση ή όχι και γιατί.

Άσκηση 2

(α) Έστω ένα ευρετήριο επεκτατού κατακερματισμού, όπου κάθε κάδος (bucket/block) μπορεί να χωρέσει έως 2 εγγραφές.
Εισάγετε τις τιμές 7, 8, 15, 14, 23, 2, 10.
Δώστε το ευρετήριο που προκύπτει μετά από κάθε διάσπαση κάδου καθώς και το αντίστοιχο ολικό βάθος του καταλόγου και το τοπικό βάθος κάθε θέσης.
Χρησιμοποιήστε τα τελευταία ψηφία της δυαδικής αναπαράστασης των τιμών

(β) Έστω ένα ευρετήριο γραμμικού κατακερματισμού όπου κάθε κάδος (bucket/block) μπορεί να χωρέσει έως 2 εγγραφές και η αρχική συνάρτηση κατακερματισμού είναι η συνάρτηση $h(k) = k \bmod 2$
(i) Εισάγετε τις τιμές 7, 8, 15, 14, 23, 2, 10. Δώστε το ευρετήριο (τουλάχιστον) κάθε φορά που γίνεται διάσπαση κάποιου κάδου.
(ii) Ποιο είναι το ζεύγος συναρτήσεων που χρησιμοποιείτε όταν έχουμε 20 κάδους (χωρίς να μετράμε πιθανούς κάδους υπερχείλισης).

Άσκηση 3

Έστω μια σχέση R(A, B, C) με κλειδί το γνώρισμα A, η οποία είναι αποθηκευμένη σε ένα διατεταγμένο αρχείο με πεδίο διάταξης το γνώρισμα B. Υπάρχει ένα B+-δέντρο ευρετήριο στο γνώρισμα A. Το B+-δέντρο είναι τάξης 45 για τους εσωτερικούς κόμβους και 50 για τα φύλλα και έχει συνολικά 4 επίπεδα (συμπεριλαμβανομένου του επιπέδου της ρίζας και των φύλλων). Ο παράγοντας ομαδοποίησης (blocking factor) για το αρχείο δεδομένων είναι 100 εγγραφές ανά σελίδα. Θεωρήστε ότι το B+-δέντρο είναι όσο το δυνατόν πιο *γεμάτο,* δηλαδή, έχει το μεγαλύτερο επιτρεπτό αριθμό τιμών σε κάθε κόμβο του.

(α) Δώστε μια εκτίμηση του κόστους για την εισαγωγή μιας τιμής σε αυτό το δέντρο (δηλαδή, πόσα μπλοκ θα χρειαστεί να διαβάσουμε/γράψουμε) και μια εκτίμηση του μεγέθους του δέντρου που προκύπτει.

(β) Δώστε μια εκτίμηση του κόστους για τη διαγραφή μιας τιμής σε αυτό το δέντρο (δηλαδή, πόσα μπλοκ θα χρειαστεί να διαβάσουμε/γράψουμε) και μια εκτίμηση του μεγέθους του δέντρου που προκύπτει.

(γ) Αντί για το B+-δέντρο, κατασκευάζουμε ένα ευρετήριο επεκτατού κατακερματισμού. Υποθέστε ότι σε κάθε κάδο (bucket) χωρούν 60 τιμές. Ποιο θα είναι το μικρότερο ολικό βάθος καταλόγου για ένα τέτοιο ευρετήριο;

(δ) Δώστε για καθένα από το (i)-(iv) παρακάτω παράδειγμα μιας SQL ερώτησης για την οποία ο πιο αποδοτικός τρόπος για την εκτέλεση της είναι πάντα

(i) χρήση του B+-δέντρου

(ii) χρήση κατακερματισμού

(iii) δυαδική αναζήτηση στο αρχείο δεδομένων

(iv) σειριακή ανάγνωση (scan) του αρχείο δεδομένων

Ερωτήσεις;