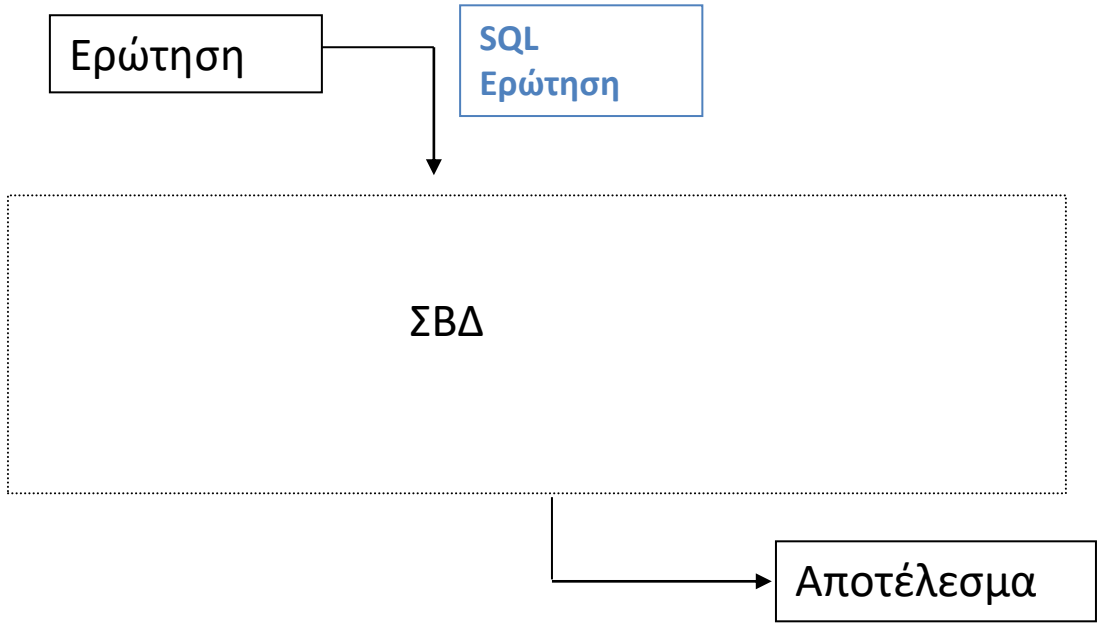
Εισαγωγή στην Επεξεργασία Ερωτήσεων

Επεξεργασία Ερωτήσεων

Η «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

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

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*

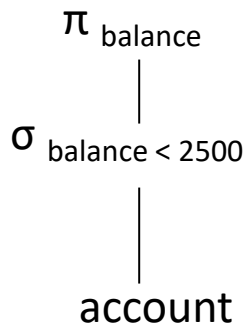
Πλάνο Εκτέλεσης

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

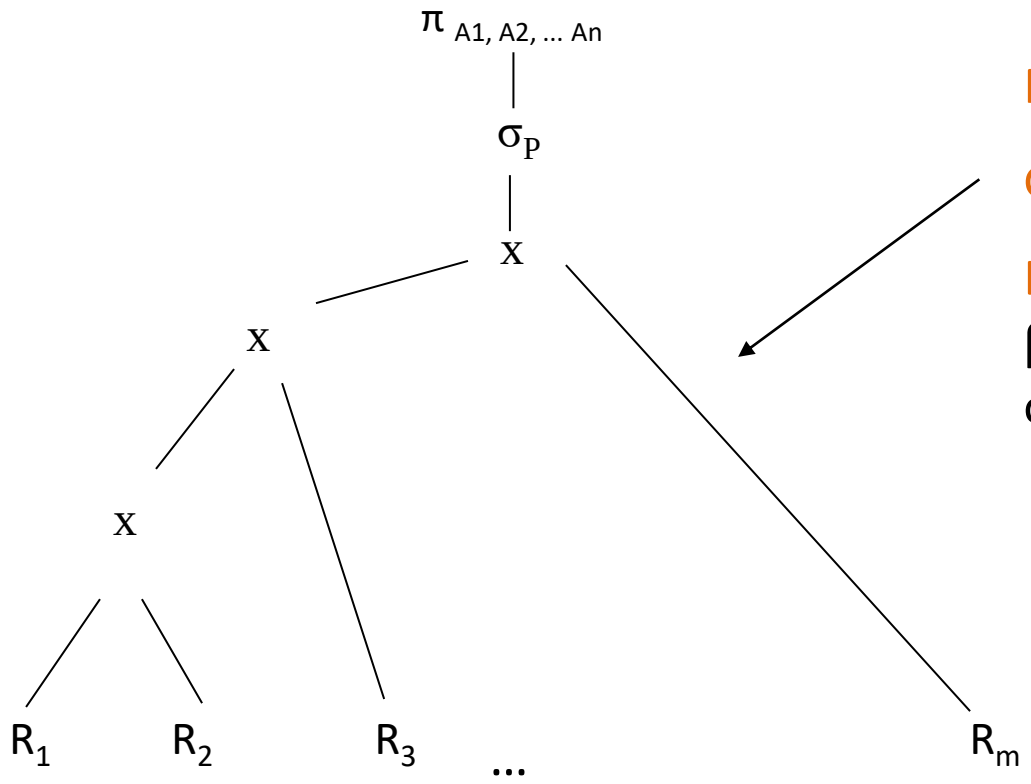
SELECT $A_1, A_2, \ldots, A_n$

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

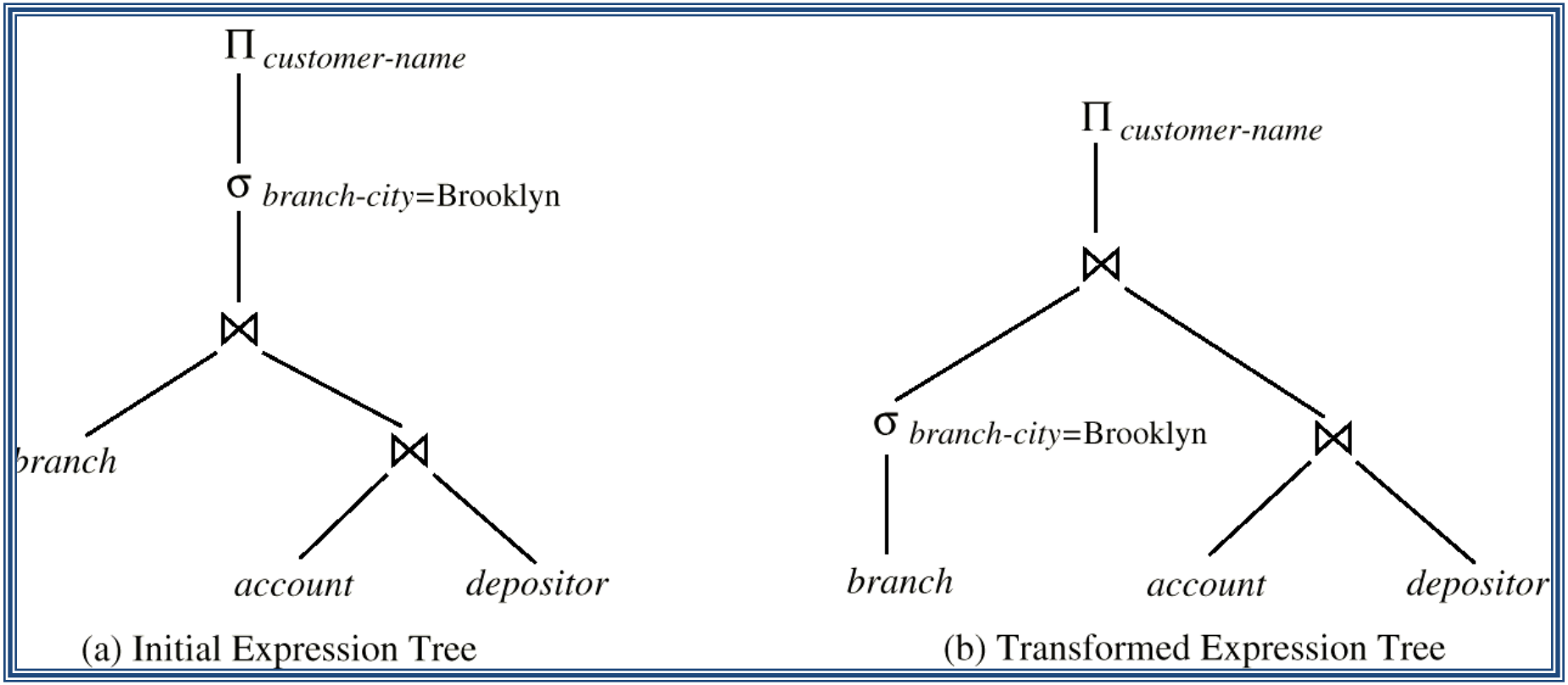
Ευριστικοί Κανόνες Βελτιστοποίησης Πλάνου Εκτέλεσης

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

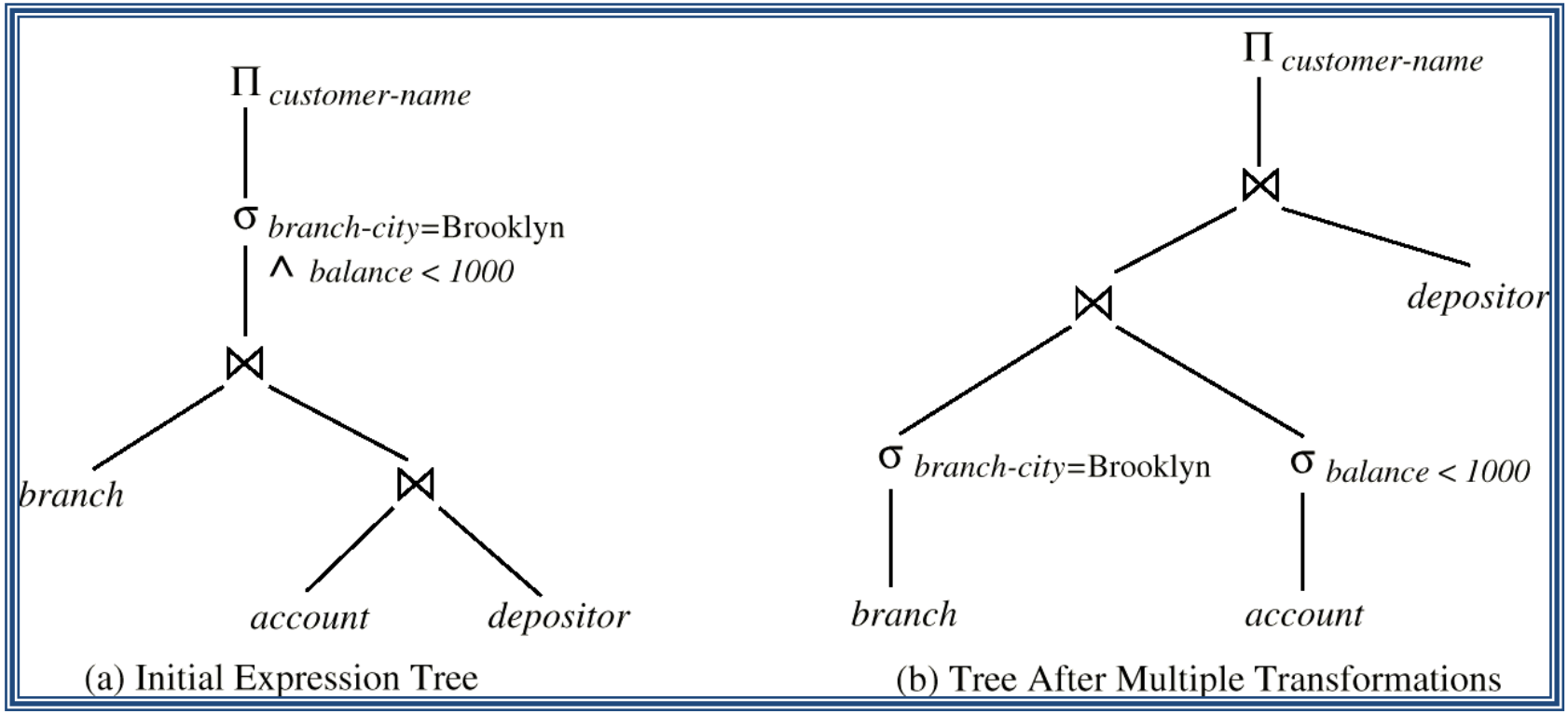
5. Διάσπαση και *μετακίνηση των λιστών προβολής όσο πιο κάτω* γίνεται στο δέντρο

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*:**

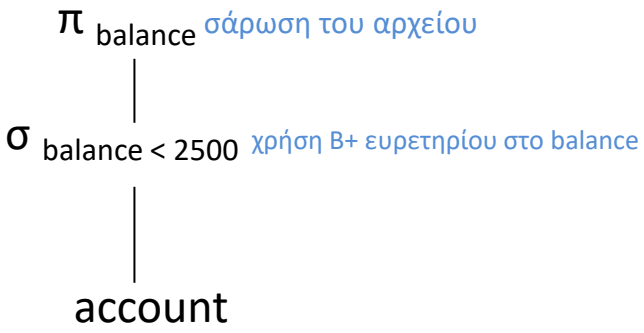
π.χ., για την υλοποίηση της επιλογής μπορεί για παράδειγμα:

- 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

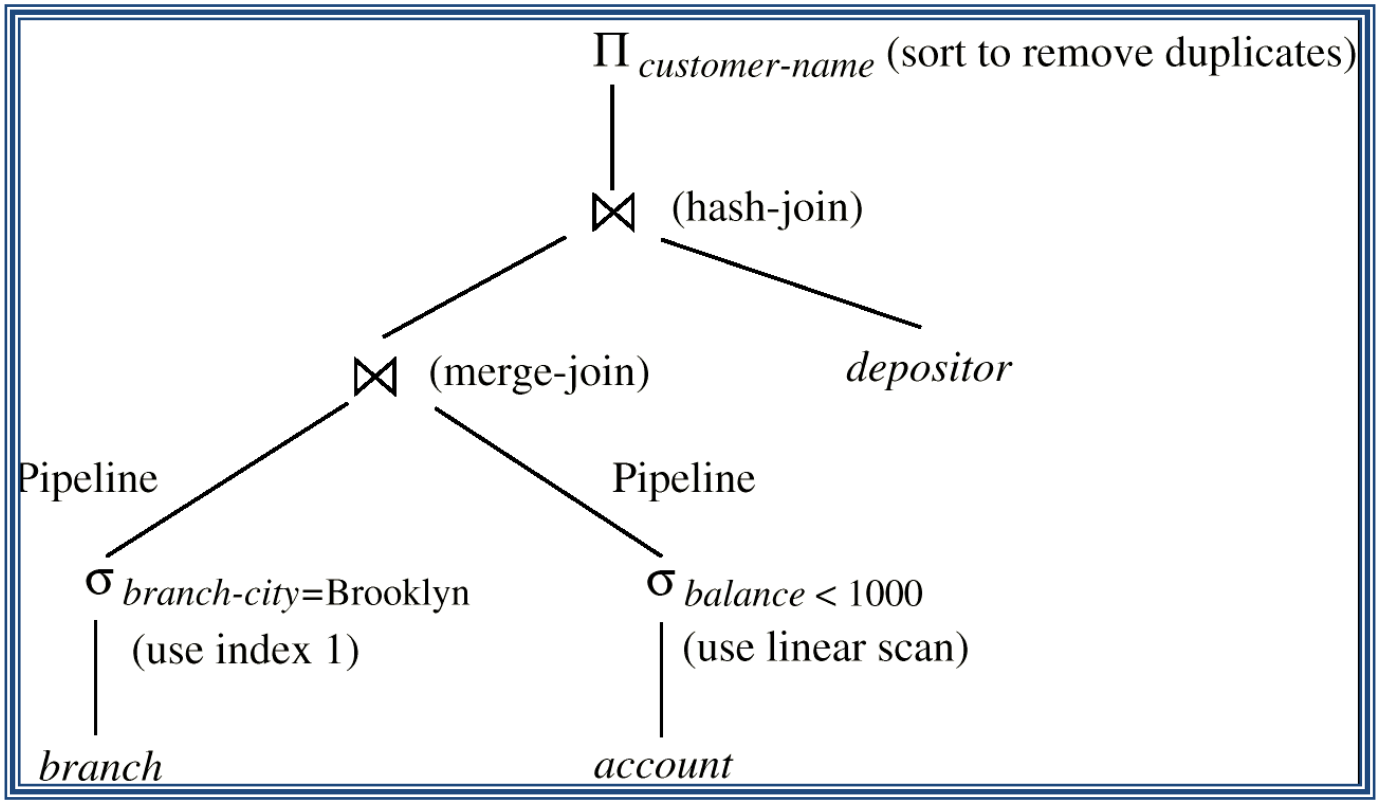
Άρα δεν αρκεί ο προσδιορισμός της πράξης - πρέπει να προσδιορίζεται **και** ο αλγόριθμος που θα χρησιμοποιηθεί για την υλοποίησή της

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

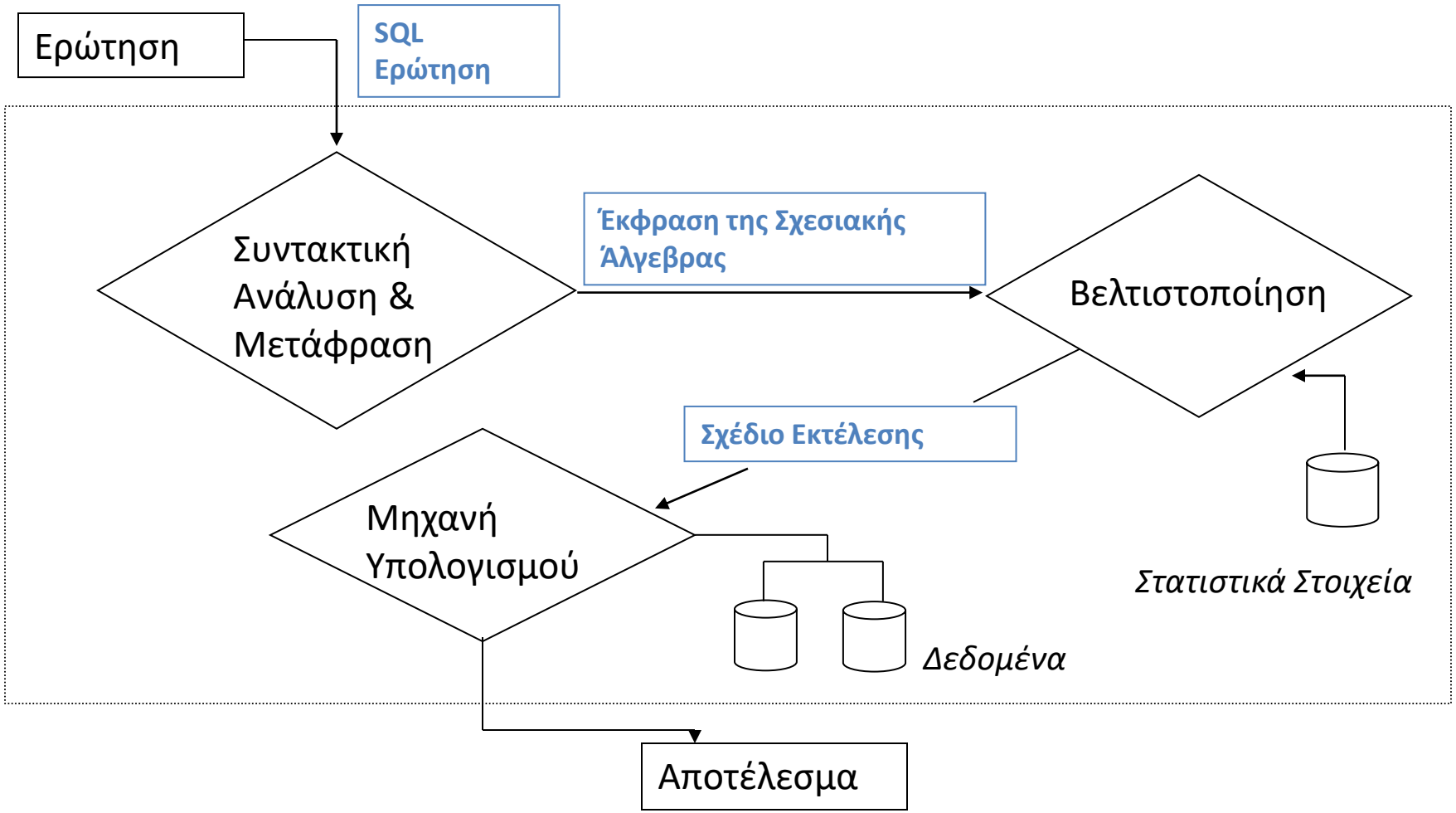
- Υπάρχουν υλοποιημένοι μια σειρά αλγορίθμων για κάθε βασική πράξη (π.χ., που χρησιμοποιούν ή όχι ευρετήρια κλπ)
- Το ΣΔΒΔ κάνει μια *εκτίμηση του κόστους* και *επιλέγει* για κάθε πράξη *τον αλγόριθμο* με τον μικρότερο (με βάση την εκτίμηση) κόστος

- Η εκτίμηση του κόστους γίνεται χρησιμοποιώντας στατιστικά στοιχεία που αποθηκεύονται στη βάση δεδομένων για αυτό το σκοπό

Τι θα δούμε στο τελευταίο μάθημα

- Αλγορίθμους υπολογισμού των βασικών πράξεων της σχεσιακής άλγεβρας χωρίς και με χρήση ευρετηρίων

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) – selectivity (επιλεκτικότητα)

  1 αν κλειδί, αν ομοιόμορφη;

- Με βάση τα στατιστικά υπολογίζεται το I/O κόστος (σε αριθμό blocks) και επιλέγεται ο αλγόριθμος με το μικρότερο κόστος
- Επιβάρυνση για την ενημέρωση των στατιστικών

Πράξη επιλογής

$$\sigma_{<\text{συνθήκη}>}(R)$$

Υπόθεση: ο πίνακας (σχέση) είναι αποθηκευμένος σε αρχείο
Θέλουμε να βρούμε τις εγγραφές (πλειάδες) που ικανοποιούν την συνθήκη

Αλγόριθμοι για την πράξη της **επιλογής**

Πιθανοί *αλγόριθμοι εκτέλεσης* για την *επιλογή*:

Χωρίς ευρετήριο (ανάλογα με την οργάνωση του αρχείου)

Ε1: Σειριακή αναζήτηση (σάρωση, scan) όλου του αρχείου

Ε2: Δυαδική αναζήτηση (αν το αρχείο είναι ταξινομημένο)

Ε3: Απλό look-up – εφαρμογή συνάρτησης κατακερματισμού (αν οργάνωση κατακερματισμού)

Με ευρετήριο στο Α αν υπάρχει (ανάλογα με το είδος του ευρετηρίου)

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

Επιλογή – συνθήκη ισότητας

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

Εκτίμηση του κόστους των διαφορετικών πιθανών αλγορίθμων και επιλογή του καλύτερου

1. Βασικό στην εκτίμηση κόστους η επιλεκτικότητα

- Είναι ή όχι το Α κλειδί;

Αν ναι, ένα ταίριασμα αλλιώς «εκτίμηση» του πλήθους των ταιριασμάτων

2. Αν το Α είναι πεδίο διάταξης όλα τα ταιριάσματα σε γειτονικά blocks, αν δεν είναι πεδίο διάταξης, στη χειρότερη περίπτωση κάθε ταίριασμα σε διαφορετικά block

Επιλογή – συνθήκη ισότητας
$\sigma_{A = \alpha}(R)$

Ε1 Σειριακή αναζήτηση (σάρωση)

Διάβασμα (scan) όλου του αρχείου

Μπορεί να χρησιμοποιηθεί σε οποιοδήποτε αρχείο

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

Επιλογή – συνθήκη ισότητας

Ε2 Δυαδική αναζήτηση

b_R: αριθμός blocks της σχέσης R
SC(A, R): μέσος αριθμός πλειάδων που ικανοποιεί μια συνθήκη («ταιριάσματα»), 1 αν κλειδί
f_R : παράγοντας ομαδοποίησης

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

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

\+

$\lceil SC(A, R)/f_R \rceil - 1$ ← Εύρεση των υπόλοιπων

Παράδειγμα: Αρχείο με $\mathbf{r_A}$ = 30.000 εγγραφές, μέγεθος block **B** = 1024 bytes, σταθερού μεγέθους εγγραφές μεγέθους $\mathbf{R_A}$ = 100 bytes, μη εκτεινόμενη καταχώρηση, όπου το πεδίο Α έχει μέγεθος $\mathbf{V_A}$ = 9 bytes.

σειριακή αναζήτηση

Μέγεθος αρχείου δεδομένων: 3.000 blocks

Διαβάζουμε ένα-ένα τα blocks του αρχείου

Περίπτωση Α κλειδί

χαρ. περίπτωση 3000 blocks

μέσο όρο 3000/2 = 1500 blocks

Περίπτωση Α όχι κλειδί, *υπόθεση 1000 διαφορετικές τιμές και ομοιόμορφη κατανομή*

3.000

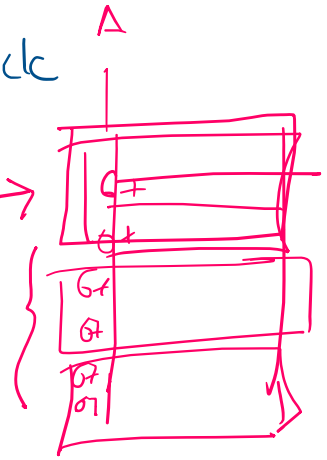
Παράδειγμα: Αρχείο με $\mathbf{r_A}$ = 30.000 εγγραφές, μέγεθος block **B** = 1024 bytes, σταθερού μεγέθους εγγραφές μεγέθους $\mathbf{R_A}$ = 100 bytes, μη εκτεινόμενη καταχώρηση, όπου το πεδίο Α έχει μέγεθος $\mathbf{V_A}$ = 9 bytes. Το Α είναι πεδίο διάταξης.

Μέγεθος αρχείου δεδομένων: 3.000 blocks

$bfr_A = 10$ εγ/block

Περίπτωση Α κλειδί

$\lceil \log_2 3000 \rceil = 12$ blocks



Περίπτωση Α όχι κλειδί, *υπόθεση 1000 διαφορετικές τιμές και ομοιόμορφη κατανομή*

εκτίμηση του # ταιριασμάτων

έχουμε 30.000/1000 εγγραφές ανά τιμή του Α

= 30 αριθμοτέλεσμα 30 πλειάδες

$\lceil \frac{30}{10} \rceil = 3$ blocks

12 + 2 blocks

Στην περίπτωση ευρετηρίου:

Κόστος αναζήτησης =

κόστος αναζήτησης στο ευρετήριο +

κόστος ανάγνωσης των ταιριασμάτων από το αρχείο

Επιλογή – συνθήκη ισότητας

Ε4 Χρήση ευρετηρίου σε πεδίο που είναι πεδίο διάταξης του αρχείου

- Οι υπόλοιπες εγγραφές που ικανοποιούν τη συνθήκη (εγγραφές με την ίδια τιμή) αν υπάρχουν βρίσκονται στα επόμενα *blocks* του αρχείου δεδομένων

Επιλογή – συνθήκη ισότητας

E4

Πεδίο ευρετηριοποίησης το Α που είναι και πεδίο διάταξης

b_R: αριθμός blocks της σχέσης R
SC(A, R): μέσος αριθμός πλειάδων που ικανοποιεί μια συνθήκη, 1 αν κλειδί
f_R: παράγοντας ομαδοποίησης
HT_i: κόστος αναζήτησης στο ευρετήριο, αν δέντρο, τότε ο αριθμός επιπέδων του

$HT_i + 1$ ⟵ 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

↑ #blocks που καταλαμβάνουν τα τυριάσματα

Επιλογή – συνθήκη ισότητας

E5 Χρήση ευρετηρίου σε πεδίο που ***δεν*** είναι πεδίο διάταξης του αρχείου

- Δεν υπάρχει διάταξη, οπότε στη χειρότερη περίπτωση κάθε εγγραφή που ικανοποιεί τη συνθήκη σε ***διαφορετικά*** *blocks*

Επιλογή – συνθήκη ισότητας

E5

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

↑ τουλάχιστον όσα τα χαρακτήρα

Επιλογή – συνθήκη με σύγκριση

$\sigma_{A \leq \alpha}(R)$ ή $\sigma_{A \geq \alpha}(R)$ και διάστημα $\sigma_{\alpha 1 \leq A \leq \alpha 2}(R)$

Αλγόριθμοι όπως πριν

- Χωρίς ευρετήριο

  Ανάλογα με την οργάνωση του αρχείου: σειριακή αναζήτηση, δυαδική αναζήτηση, αλλά *δεν είναι δυνατή η εφαρμογή συνάρτησης κατακερματισμού*

- Με ευρετήριο

  Πρωτεύον/δευτερεύον, είδος ευρετηρίου

Χρειάζεται εκτίμηση του αριθμού των ταιριασμάτων

Επιλογή – συνθήκη με σύγκριση

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

Αρχείο σωρού, σειριακή αναζήτηση

Επιλογή – συνθήκη με σύγκριση

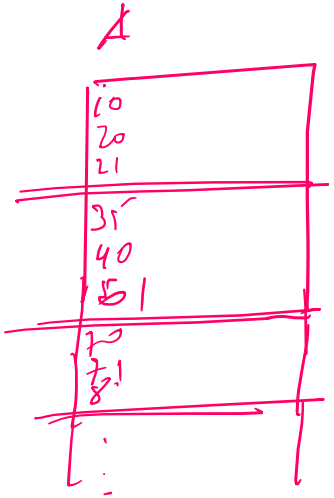
$\sigma_{A \leq \alpha}(R)$

Αρχείο με πεδίο διάταξης το Α

Έστω αύξουσα διάταξη, χρειάζεται δυαδική αναζήτηση;

δεχρειάζεται διαβάζουμε τα blocks των αρχείων σειριακά μέχρι να βρούμε μια εγγραφή με Α>α

σΑ≤4



Παράδειγμα. Αρχείο με $\mathbf{r_A}$ = 30.000 εγγραφές ενός πίνακα R, μέγεθος block **B** = 1024 bytes, σταθερού μεγέθους εγγραφές μεγέθους $\mathbf{R_A}$ = 100 bytes, μη εκτεινόμενη καταχώρηση, όπου το πεδίο A έχει μέγεθος $\mathbf{V_A}$ = 9 bytes. *Το Α παίρνει τιμές από το 1 έως το 1000, οι οποίες είναι ομοιόμορφα κατανεμημένες.* Το Α είναι πεδίο διάταξης. Έστω αύξουσα διάταξη. Κόστος $\sigma_{A<90}(R)$;

Διαβάζουμε σειριακά ένα-ένα τα blocks έχει A>,90

Εκτίμηση αριθμού πλειάδων που ικανοποιούν τη συνθήκη με βάση τα στοιχεία της εκφώνησης

→ #πλειάδων με A < 90

30 * 89 πλειάδες που ικανοποιούν την συνθήκη

30 * 89 = 2670

⌈2670⌉ = 267 blocks

Επιλογή – συνθήκη με σύγκριση

$\sigma_{\alpha1 \leq A \leq \alpha2}(R)$

Με ευρετήριο

Όπως πάντα
Κόστος αναζήτησης = κόστος αναζήτησης στο ευρετήριο + κόστος ανάγνωσης των ταιριασμάτων από το αρχείο

- Αν το Α είναι **πεδίο διάταξης** για το αρχείο αρκεί να βρούμε τη θέση του α1, διαβάζουμε τα επόμενα blocks από το αρχείο δεδομένων έως να βρούμε εγγραφή με A > α2

- Αν το Α **δεν είναι πεδίο διάταξης**, θα πρέπει να βρούμε όλες τις τιμές στο ευρετήριο που ανήκουν σε αυτό το διάστημα

Παράδειγμα: Έστω ένας πίνακας (σχέση) CITY(Name, Population, Country) ο οποίος έχει πληροφορία για 150.000 πόλεις που είναι ομοιόμορφα κατανεμημένες σε 1.000 χώρες (δηλαδή 100 πόλεις ανά χώρα). Το Name είναι *κλειδί*. Ο πίνακας είναι αποθηκευμένος σε *ένα διατεταγμένο αρχείο* ως προς το *γνώρισμα* Name. Τα γνωρίσματα Name και Country έχουν μέγεθος 16 bytes, το γνώρισμα Population 32 bytes και ένα block (σελίδα) 2048 bytes. Υποθέστε ότι όλοι οι δείκτες έχουν μέγεθος 32 bytes.

Το μικρότερο B+-δέντρο στο γνώρισμα Population. Εκτίμηση του κόστους της

$\sigma_{50000 < \text{Population} < 100000}(\text{CITY})$

Υποθέστε ότι το Population παίρνει διακριτές τιμές και ότι υπάρχουν 300 πόλεις που ικανοποιούν τη συνθήκη.

Τάξη εσωτερικών κόμβων, p = 32

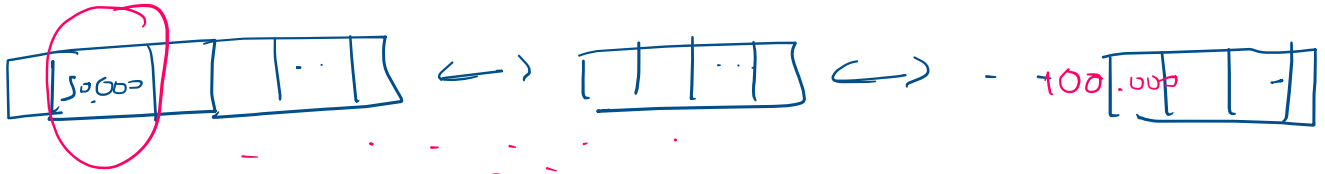
Τάξη φύλλων p_l= 31

4 επίπεδα

Παράγοντας ομαδοποίησης για το αρχείο δεδομένων 32 εγγραφές ανά block

1. Αναζήτηση του 50000 στο ευρετήριο → φύλλο που εμφανίζεται το 50000

φύλλα στο B+ δέντρο



Βήμα 2

2. Διαβάζουμε τα φύλλα στα οποία μπορεί να βρίσκεται τιμή > 100.000. Πόσα είναι τα φύλλα?

$\lceil \frac{300}{31^*} \rceil = 10$, 10 φύλλα

(* τάξη φύλλων, σχετικό δέντρο)

3. Διαβάζουμε τις εγγραφές από το αρχείο

300

Βήμα 1: ~~[illegible]~~ #επιπέδων 4

Βήμα 2: 10 - 1 = 9 ~~φύλλα~~

Βήμα 3: 300

Επιλογή με σύζευξη

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

$\sigma_{County = Greece \text{ and } Population > 100000}(City)$

$P1$ $P2$

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

- Επιλογή του γνωρίσματος συνθήκη με τη *μικρότερη* επιλεκτικότητα (γιατί;)
- Χρήση μίας 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

Ερωτήσεις;

- Εξετάσεις με ανοικτά βιβλία/σημειώσεις.

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

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