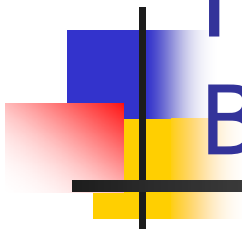
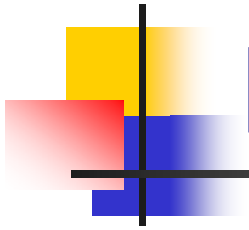


ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ - ΤΜΗΥΠ
ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ ΙΙ

Β. Μεγαλοοικονόμου

Επεξεργασία Ερωτημάτων/Βελτιστοποίηση

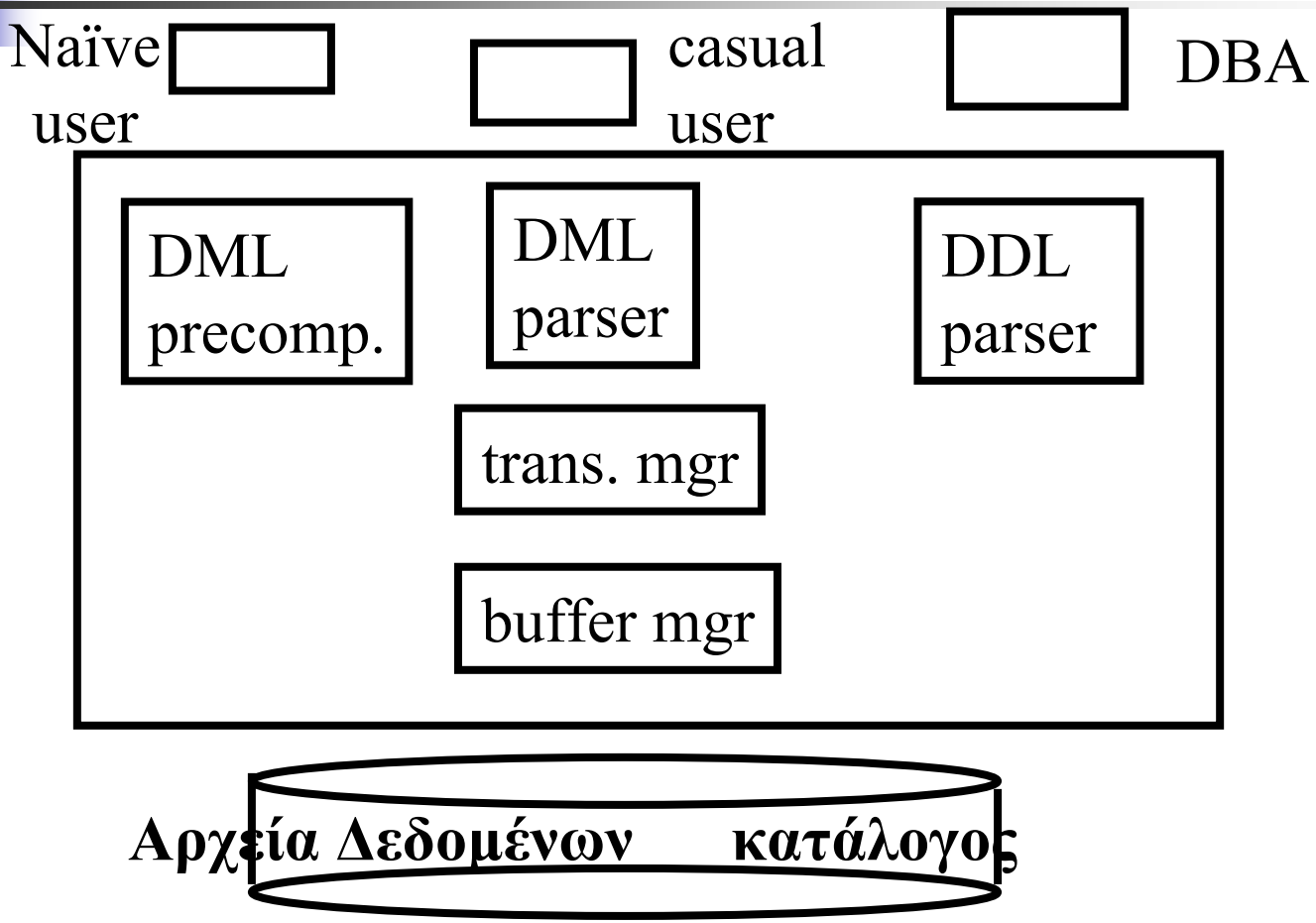
(παρουσίαση βασισμένη εν μέρη σε σημειώσεις των Silberchatz, Korth και Sudarshan και του C. Faloutsos)

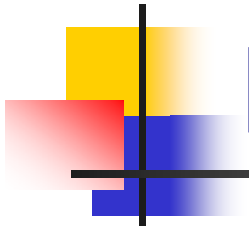


Γενικά – σχεσιακό μοντέλο

- Σχεσιακό μοντέλο - SQL
- Συναρτησιακές εξαρτήσεις & Κανονικοποίηση
- Φυσικός σχεδιασμός, Δεικτοδότηση
- Επεξεργασία ερωτημάτων/βελτιστοποίηση
- Επεξεργασία Δοσοληψιών
- Προχωρημένα ζητήματα
  - Κατανεμημένες Βάσεις Δεδομένων
  - OO- και OR-DBMSs

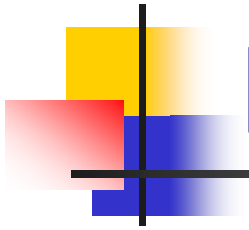
Επισκόπηση ενός ΣΔΒΔ





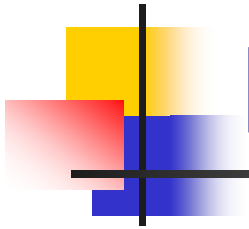
Επισκόπηση- λεπτομερώς

- Κίνητρο- Γιατί βελτιστ. Ερωτήματος;
- Ισοδυναμία εκφράσεων
- Εκτίμηση κόστους
- Κόστος ευρετηρίων
- Στρατηγικές συνένωσης (join)



Γιατί βελτιστ. Ερωτήματος;

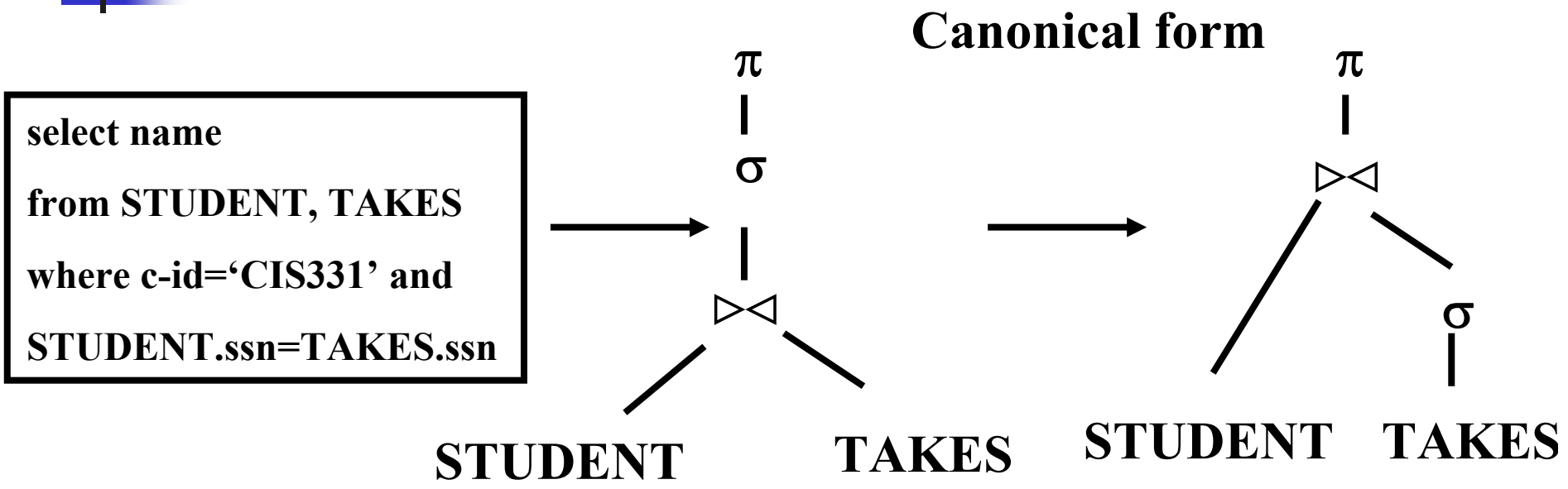
- SQL: ~δηλωτική
- Καλή Βελτ. Ερωτ. → Μεγάλη διαφορά
  - π.χ., ακολουθιακή σάρωση vs B-tree ευρετήριο, σε P=1,000 σελίδες



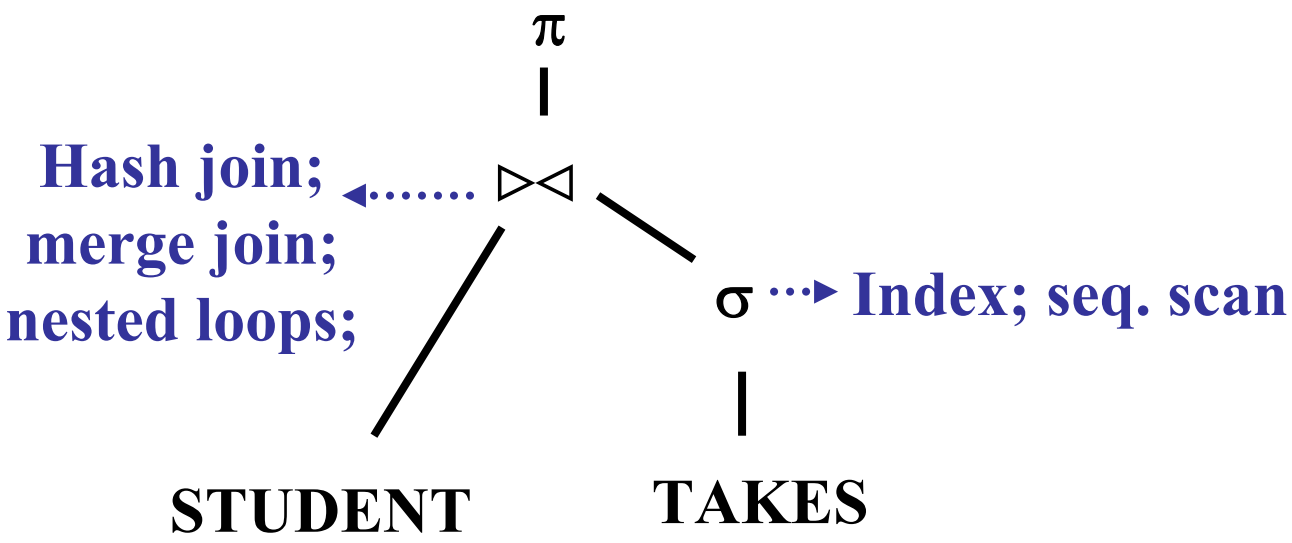
Βήματα Βελτ.Ερωτ.

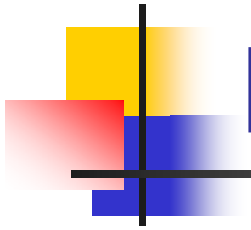
- φέρτε το ερώτημα στην εσωτερική μορφή (π.χ., parse tree)
- … στην ‘κανονική μορφή’ (συντακτική Βελτ. Ερωτ.)
- δημιουργήστε εναλλακτικά σχέδια
- εκτιμήστε το κόστος, επιλέξτε το καλύτερο

Βελτ.Ερωτ. - παράδειγμα



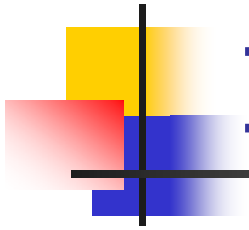
Βελτ.Ερωτ. - παράδειγμα





Επισκόπηση- λεπτομερώς

- Γιατί Βελτ.Ερωτ.;
- Ισοδυναμία εκφράσεων
- Εκτίμηση κόστους
- Κόστος ευρετηρίων
- Στρατηγικές συνένωσης (join)



Ισοδυναμία εκφράσεων

- ... ή συντακτική Βελτ. Ερωτ.
- Εν συντομία: εφάρμοσε επιλογές και προβολές νωρίς
- Περισσότερες λεπτομέρειες:

  δείτε κανόνες μετατροπής στο βιβλίο

Ισοδυναμία εκφράσεων

- Ε: Πώς να αποδείξετε έναν κανόνα μετασχηματισμού;

$$\sigma_P(R1 \bowtie R2) \stackrel{?}{=} \sigma_P(R1) \bowtie \sigma_P(R2)$$

- Α: Χρησιμοποιείστε το σχεσιακό λογισμό πλειάδων (TRC) για να δείξετε ότι LHS = RHS, π.χ.:

$$\sigma_P(R1 \cup R2) \stackrel{?}{=} \sigma_P(R1) \cup \sigma_P(R2)$$

Ισοδυναμία εκφράσεων

$$\sigma_P(R1 \cup R2) \stackrel{?}{=} \sigma_P(R1) \cup \sigma_P(R2)$$

$$t \in LHS \Leftrightarrow$$

$$t \in \quad (R1 \cup R2) \wedge P(t) \quad \Leftrightarrow$$

$$(t \in \quad R1 \quad \vee \quad t \in R2) \wedge P(t) \quad \Leftrightarrow$$

$$(t \in \quad R1 \wedge P(t)) \quad \vee \quad (t \in R2) \wedge P(t)) \quad \Leftrightarrow$$

Ισοδυναμία εκφράσεων

$$\sigma_P(R1 \cup R2) \stackrel{?}{=} \sigma_P(R1) \cup \sigma_P(R2)$$

...

$$(t \in \quad R1 \wedge P(t)) \vee (t \in R2) \wedge P(t)) \quad \Leftrightarrow$$

$$(t \in \quad \sigma_P(R1)) \vee (t \in \sigma_P(R2)) \quad \Leftrightarrow$$

$$t \in \quad \sigma_P(R1) \cup \sigma_P(R2) \quad \Leftrightarrow$$

$$t \in RHS$$

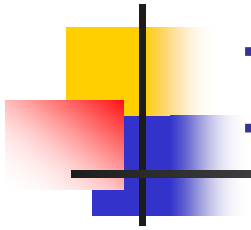
$$QED$$

Ισοδυναμία εκφράσεων

- Επιλογή (selection)
  - πραγματοποιήστε τις νωρίς
  - σπάστε ένα πολύπλοκο κατηγόρημα και πιέστε

$$\sigma_{p1\wedge p2\wedge \ldots pn}(R) = \sigma_{p1}(\sigma_{p2}(\ldots\sigma_{pn}(R))\ldots)$$

  - απλοποιήστε ένα πολύπλοκο κατηγόρημα
    - ('X=Y and Y=3') -> 'X=3 and Y=3'



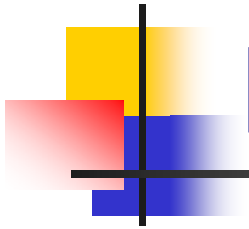
Ισοδυναμία εκφράσεων

- Προβολές (projections)
  - πραγματοποιήστε τις νωρίς (αλλά προσεκτικά…)
    - Μικρότερες πλειάδες
    - Λιγότερες πλειάδες (αν εξαλείφουν οι διπλότυπες)
  - Εξαιρέστε όλα τα γνωρίσματα εκτός από αυτά που ζητήθηκαν ή είναι απαραίτητα (π.χ., γνωρίσματα συνένωσης)

Ισοδυναμία εκφράσεων

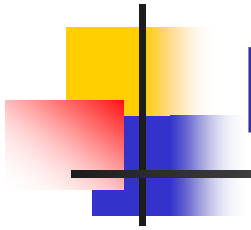
- Συνενώσεις
  - Αντιστροφή, συσχέτιση
    $R \bowtie S = S \bowtie R$

    $(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$
  - Ε: n-way συνένωση – Πόσες διαφορετικές ακολουθίες; ... Διεξοδική αρίθμηση είναι πολύ αργή...



Βήματα Βελτ.Ερωτ.

- φέρτε τα ερωτήματα στην εσωτερική μορφή (π.χ., parse tree)
- … στην ‘κανονική μορφή’ (συντακτική Βελτ.Ερωτ.)
- δημιουργήστε εναλλακτικά σχέδια
- εκτιμήστε το κόστος; επιλέξτε το καλύτερο

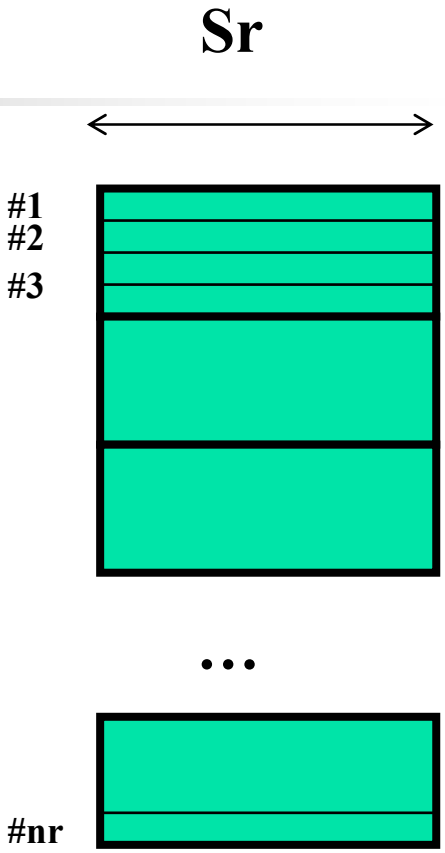


Εκτίμηση κόστους

- Π.χ., βρείτε το Α.Μ. των φοιτητών με βαθμό 10 στις «Βάσεις Δεδομένων» (χρησιμοποιώντας ακολουθιακή αναζήτηση)
- Πόση ώρα θα πάρει ένα ερώτημα?
  - CPU (αλλά: μικρό κόστος, μειώνεται, δύσκολο να εκτιμηθεί)
  - Δίσκος (κυρίως, # μεταφορών blocks)
- Πόσες πλειάδες θα χαρακτηριστούν;
- (Ποια στατιστικά χρειαζόμαστε να κρατάμε;)

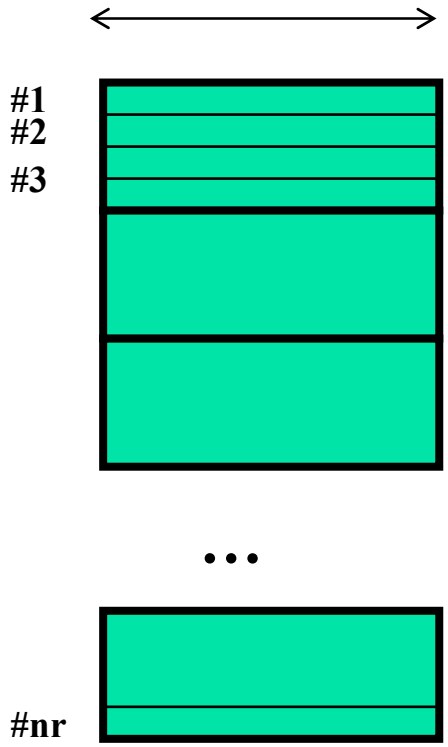
Εκτίμηση κόστους

- Στατιστικά: για κάθε σχέση ‘r’ κρατάμε
  - nr : # πλειάδων;
  - Sr : μέγεθος πλειάδων σε bytes



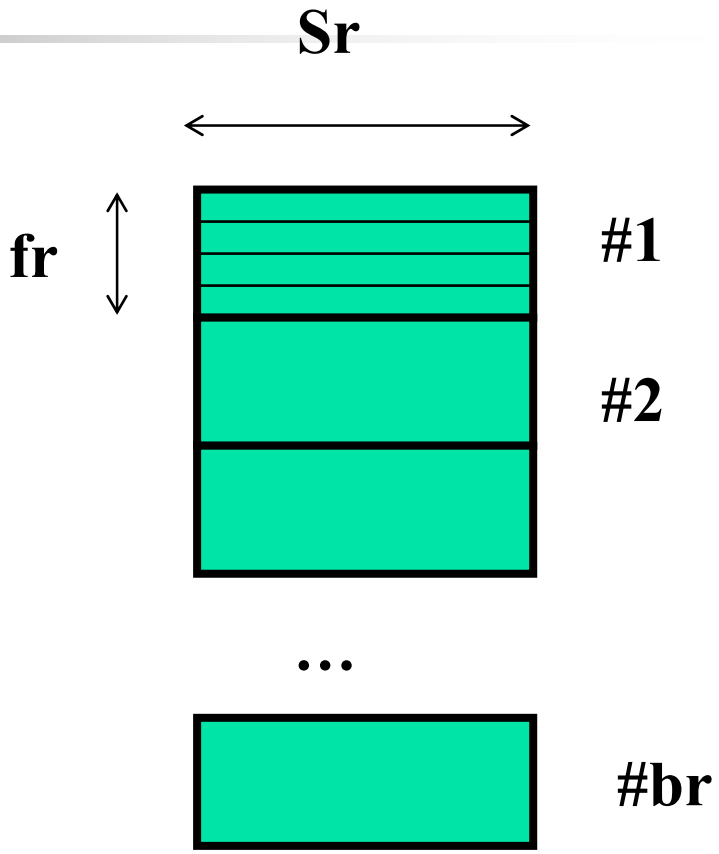
Εκτίμηση κόστους

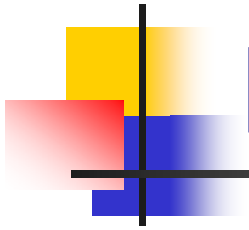
- Στατιστικά: για κάθε σχέση ‘r’ κρατάμε
  - …
  - V(A,r): αριθμό διακριτών τιμών του γνωρίσματος ‘A’
  - (πρόσφατα, και ιστογράμματα)



Παράγωγα στατιστικά

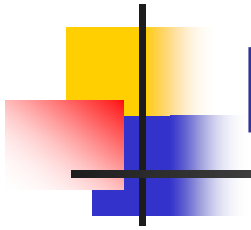
- fr: παράγοντας ομαδοποίησης (blocking factor) = max# εγγραφών/block (=??)
- br: # blocks (=??)
- SC(A,r) = επιλεκτικότητα= avg# εγγραφών με A=δοθέν (=??)





Παράγωγα στατιστικά

- fr: παράγοντας ομαδοποίησης = max# εγγραφών/block (= B/Sr ; B: μέγεθος block σε bytes)
- br: # blocks (= nr / fr)

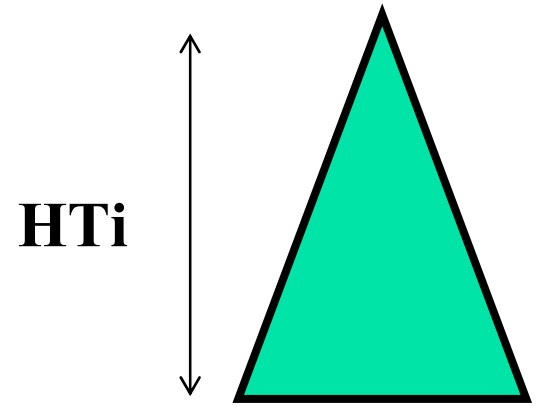


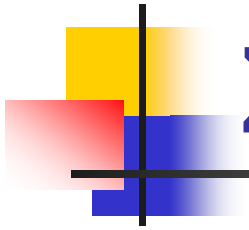
Παράγωγα στατιστικά

- SC(A,r) = επιλεκτικότητα = avg# εγγραφών με A=δοθέν (= nr / V(A,r)) (προϋποθέτει ομοιομορφία...) – πχ: 30,000 φοιτητές, 10 σχολές – πόσοι φοιτητές στην Πολυτεχνική Σχολή;

Επιπλέον μετρικές που χρειάζονται:

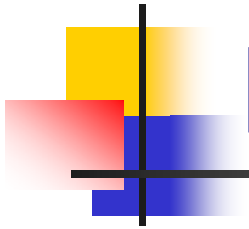
- Για ευρετήριο 'i':
  - fi: μέσος παράγοντας διακλάδωσης (fanout – degree) (~50-100)
  - HTi: # επίπεδα του ευρετηρίου 'i' (~2-3)
    - ~ log(#entries)/log(fi)
  - LBi: # μπλοκς στο επίπεδο των φύλλων





Στατιστικά

- Που τα αποθηκεύουμε;
- Πόσο συχνά τα ανανεώνουμε;



Βήματα Βελτ.Ερωτ.

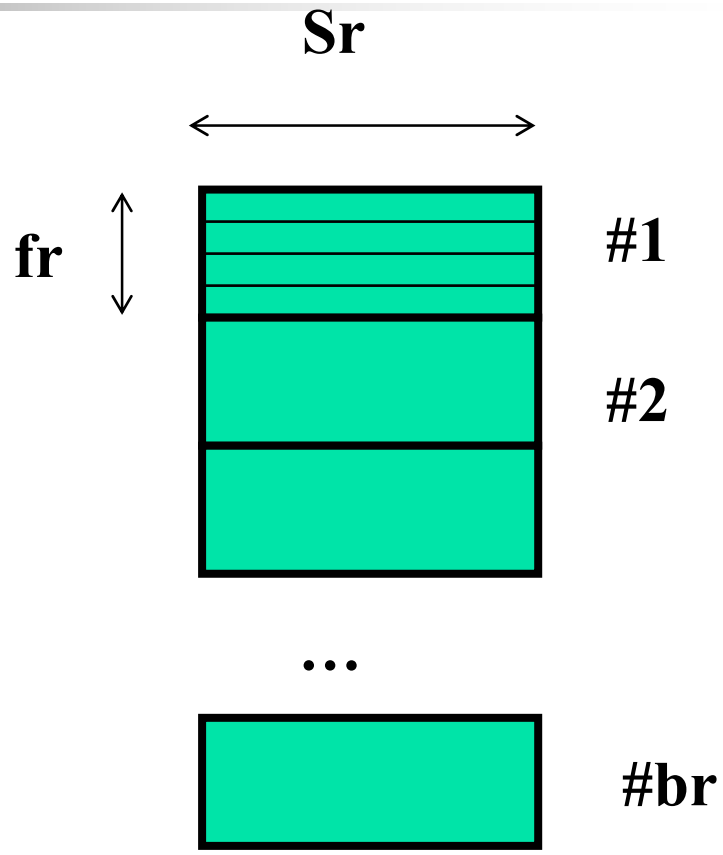
- Φέρτε το ερώτημα στην εσωτερική μορφή (π.χ., parse tree)
- … στην ‘κανονική μορφή’ (συντακτική Βελτ.Ερωτ.)
- δημιουργήστε εναλλακτικά πλάνα
  - επιλογές; ταξινόμηση; προβολές
  - συνενώσεις
- εκτιμήστε το κόστος, επιλέξτε το καλύτερο

Εκτίμηση κόστους & δημιουργία πλάνου

- Επιλογές – π.χ.,

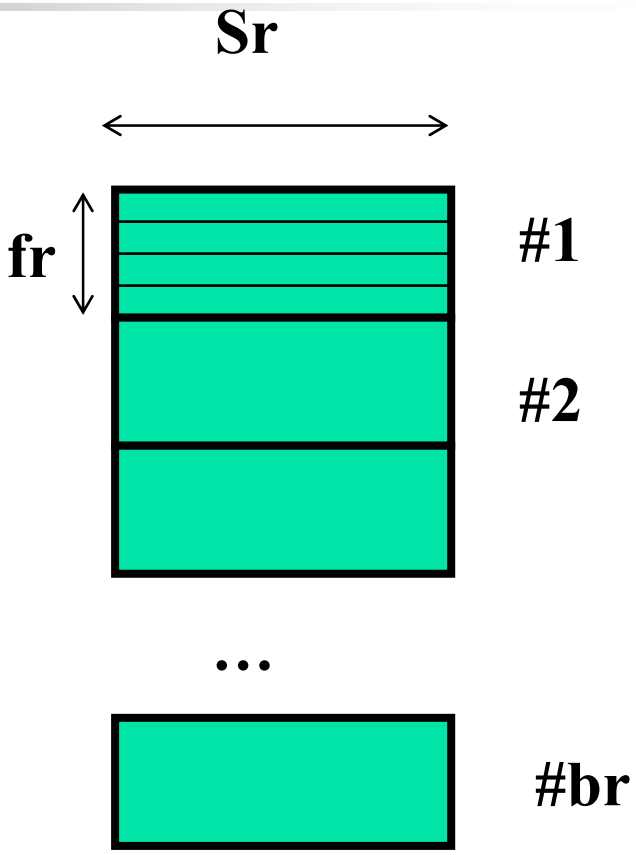
  select *
  from TAKES
  where grade = 'A'

- Πλάνα;



Εκτίμηση κόστους & δημιουργία πλάνου

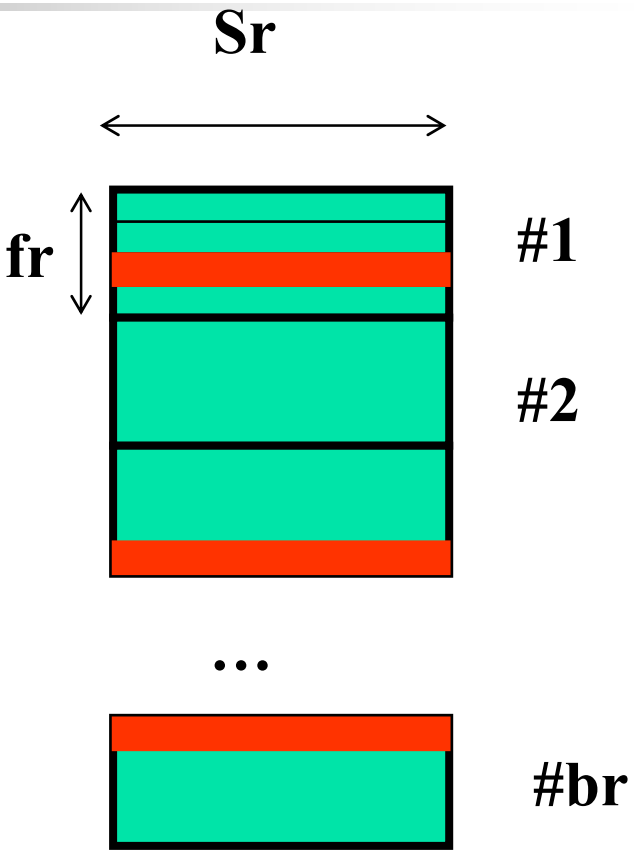
- Πλάνα;
  - Σειριακή αναζήτηση
  - Δυαδική αναζήτηση
    - (αν είναι ταξινομημένο κατά αύξουσα σειρά)
  - Αναζήτηση ευρετηρίου
    - αν υπάρχει ένα ευρετήριο



Εκτίμηση κόστους & δημιουργία πλάνου

Διαδοχ. Αναζ.– κόστος;

- br (χειρότερη περιπτωση)
- br/2 (μέση περίπτωση, αν αναζητάμε με βάση το πρωτεύων κλειδί)

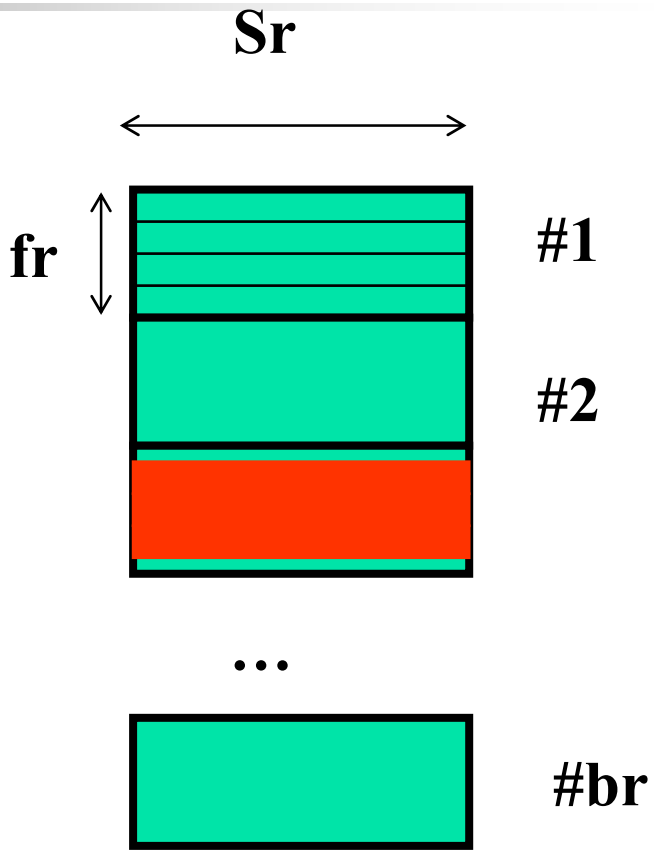


Εκτίμηση κόστους & δημιουργία πλάνου

δυαδική αναζ.– κόστος;

αν είναι ταξινομημένο κατά αύξουσα σειρά:

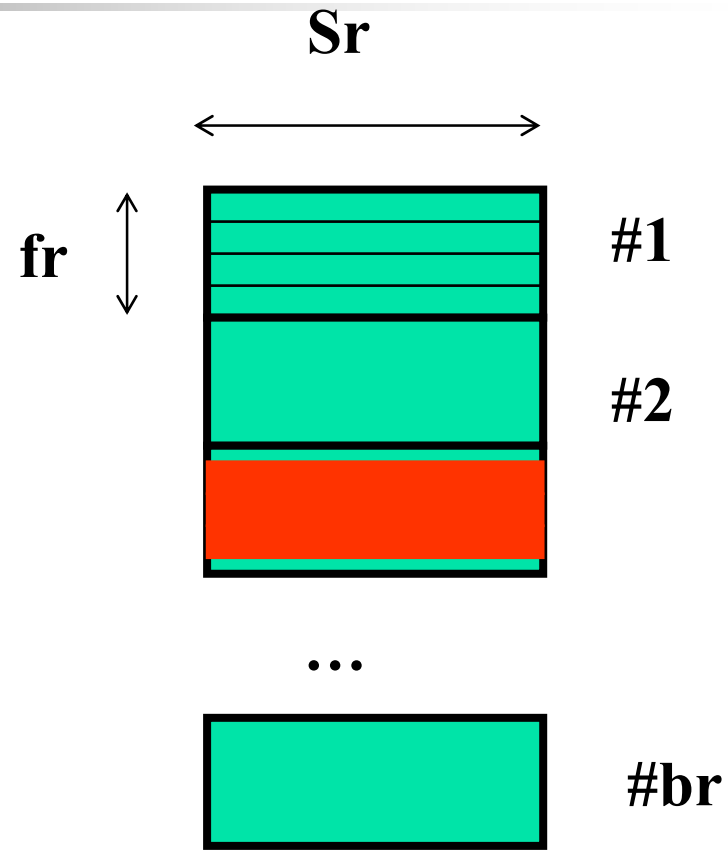
- ~log(br) **+** SC(A,r)/fr (=#μπλοκς καταλυμένων από αρμόδιες πλειάδες) -1



Εκτίμηση κόστους & δημιουργία πλάνου

εκτίμηση της επιλογής πρωτευόντων SC(A,r):

μη τετριμμένη



Εκτίμηση κόστους & δημιουργία πλάνου

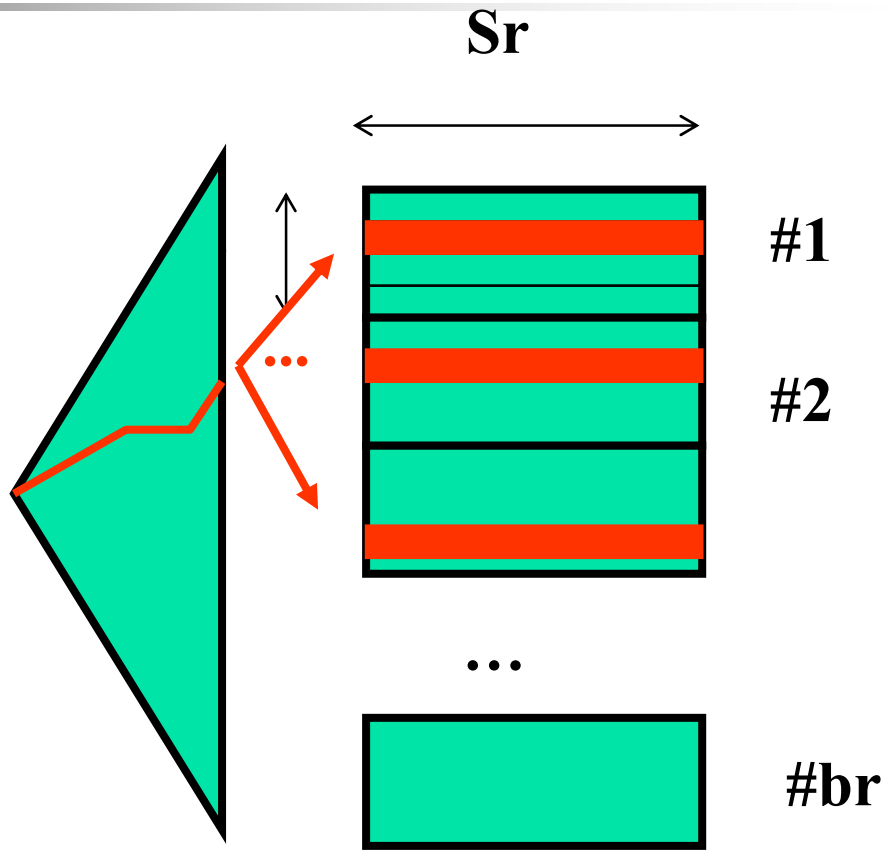
μέθοδος#3:
ευρετήριο – κόστος?

- επίπεδα ευρετηρίου +
- μπλοκς με αρμόδιες πλειάδες

περίπτωση#1: πρωτεύων κλειδί

περίπτωση#2: δευτ. κλειδί – συσταδοποιημένο ευρετήριο

περίπτωση#3: δευτ. κλειδί – μη συσταδοποιημένο ευρετήριο



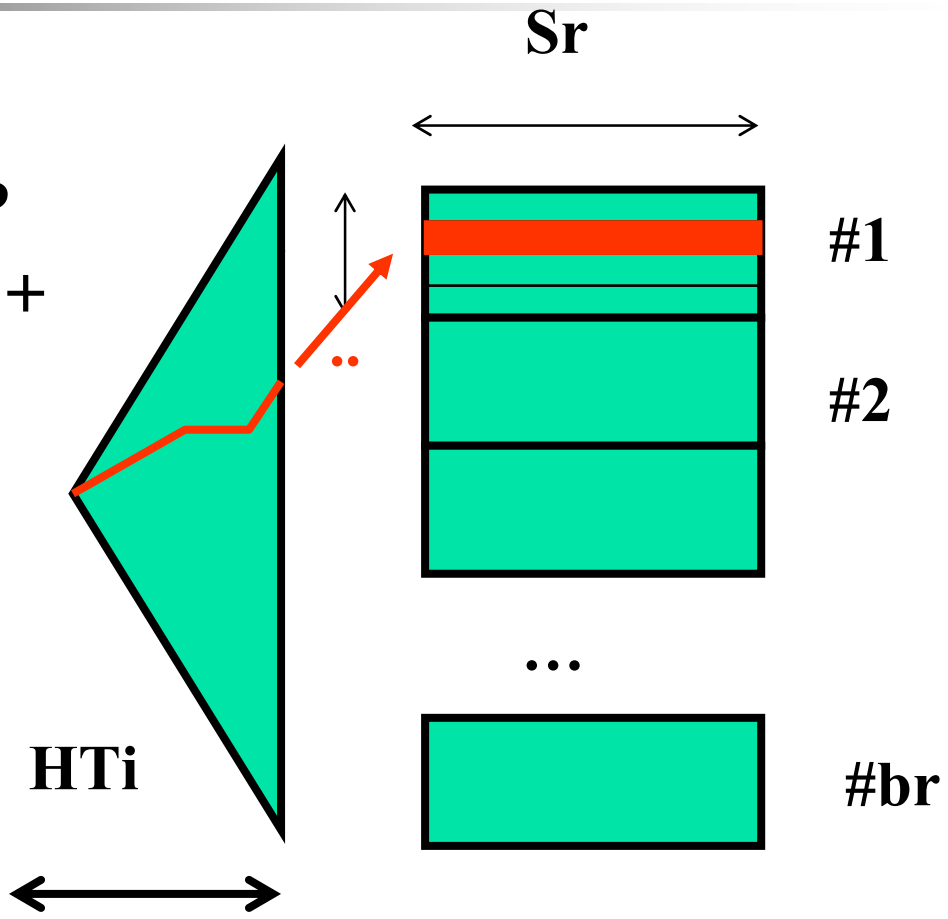
Εκτίμηση κόστους & δημιουργία πλάνου

μέθοδος#3:
ευρετήριο – κόστος?

- επίπεδα ευρετηρίου +
- μπλοκς με αρμόδιες πλειάδες

περίπτωση#1: πρωτεύων κλειδί – κόστος:

HTi + 1



Εκτίμηση κόστους & δημιουργία πλάνου

μέθοδος#3:
ευρετήριο – κόστος?

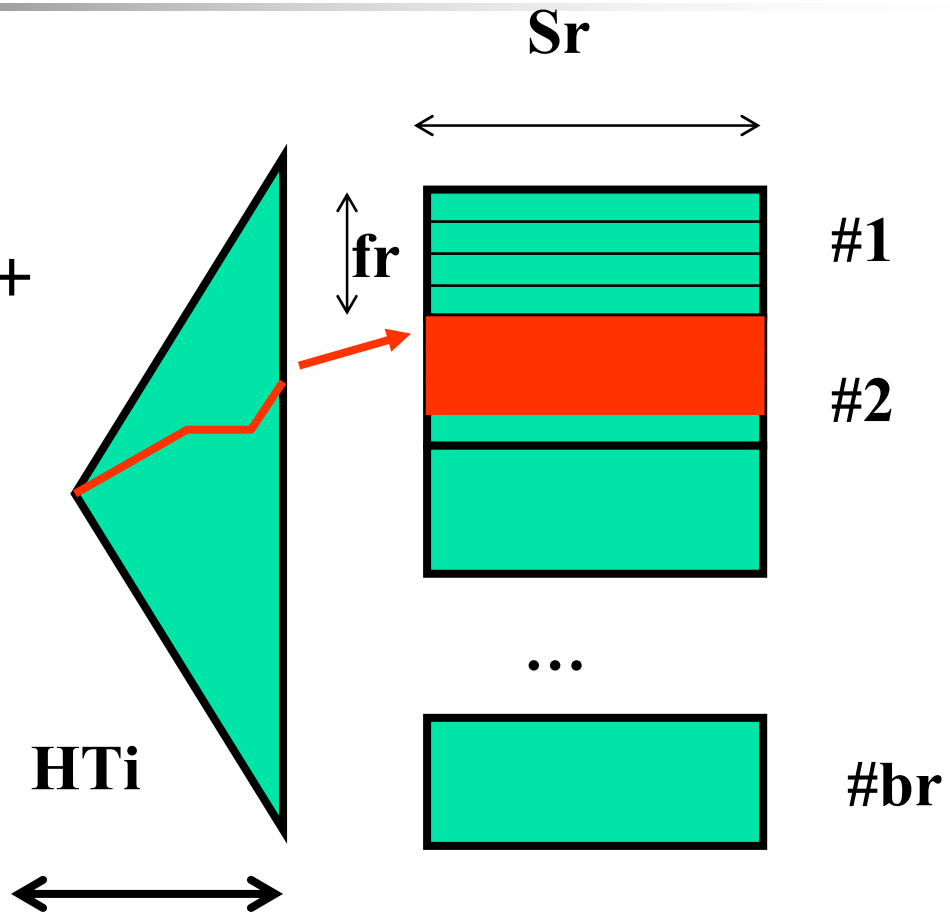
- επίπεδα ευρετηρίου +
- μπλοκς με αρμόδιες πλειάδες

περίπτωση#2: δευτ. κλειδί – συσταδοποιημένο ευρετήριο

Ή πρωτ. ευρετ.σε μη-κλειδί

…ανάκληση πολλαπλών εγγραφών

HTi + SC(A,r)/fr



Εκτίμηση κόστους & δημιουργία πλάνου

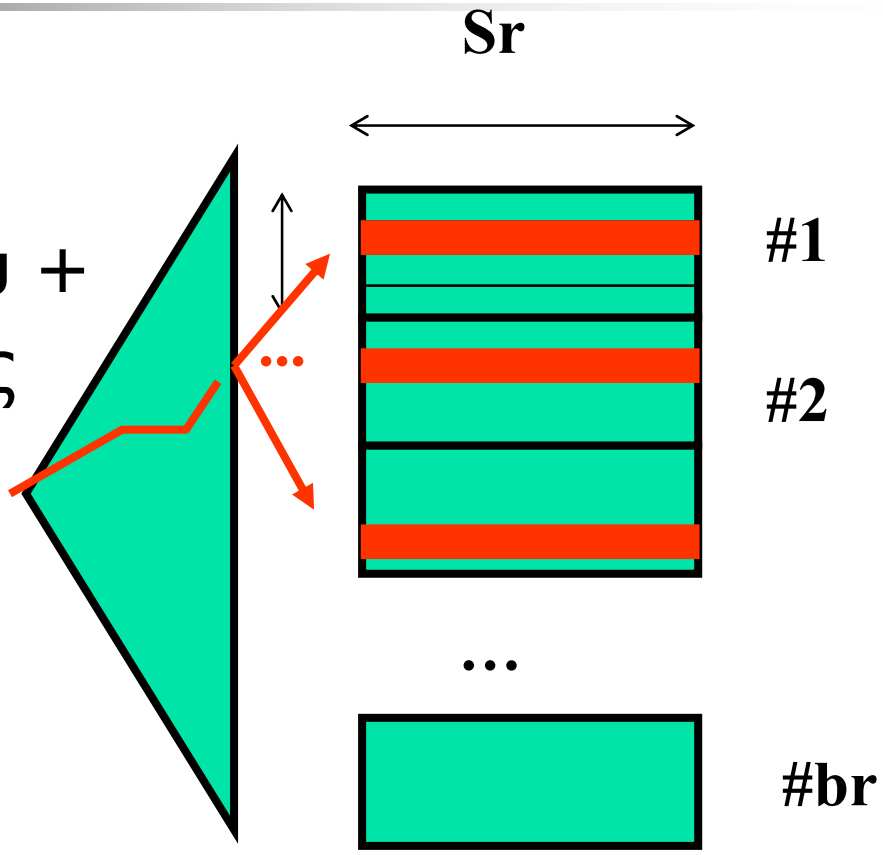
μέθοδος#3:
ευρετήριο – κόστος?

- επίπεδα ευρετηρίου +
- μπλοκς με αρμόδιες πλειάδες

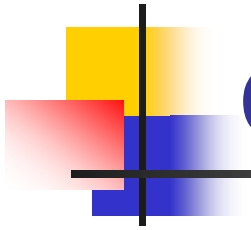
περίπτωση#3: δευτ. κλειδί – μη συσταδοποιημένο ευρετήριο

HTi + SC(A,r)

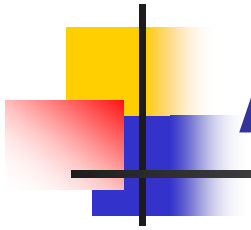
(actually, pessimistic...)



Εκτίμηση κόστους – αριθμητικό παράδειγμα

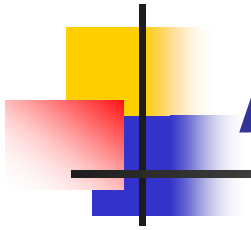


- βρείτε τους λογαριασμούς με *όνομα-υποκαταστήματος* = 'Πάτρα'
- Λογαριασμός (*όνομα-υποκαταστήματος*, υπόλοιπο, ...)



Αριθμητικό παράδειγμα (συνεχ.)

- n-account = 10,000 πλειάδες
- f-account = 20 πλειάδες/μπλοκ
- V(υπόλοιπο, λογαριασμός) = 500 διακριτές τιμές
- V(όνομα-υποκαταστήματος, λογαριασμός) = 50 διακριτές τιμές
- για branch-index: παράγοντας διακλάδωσης (fanout) fi = 20



Αριθμητικό παράδειγμα

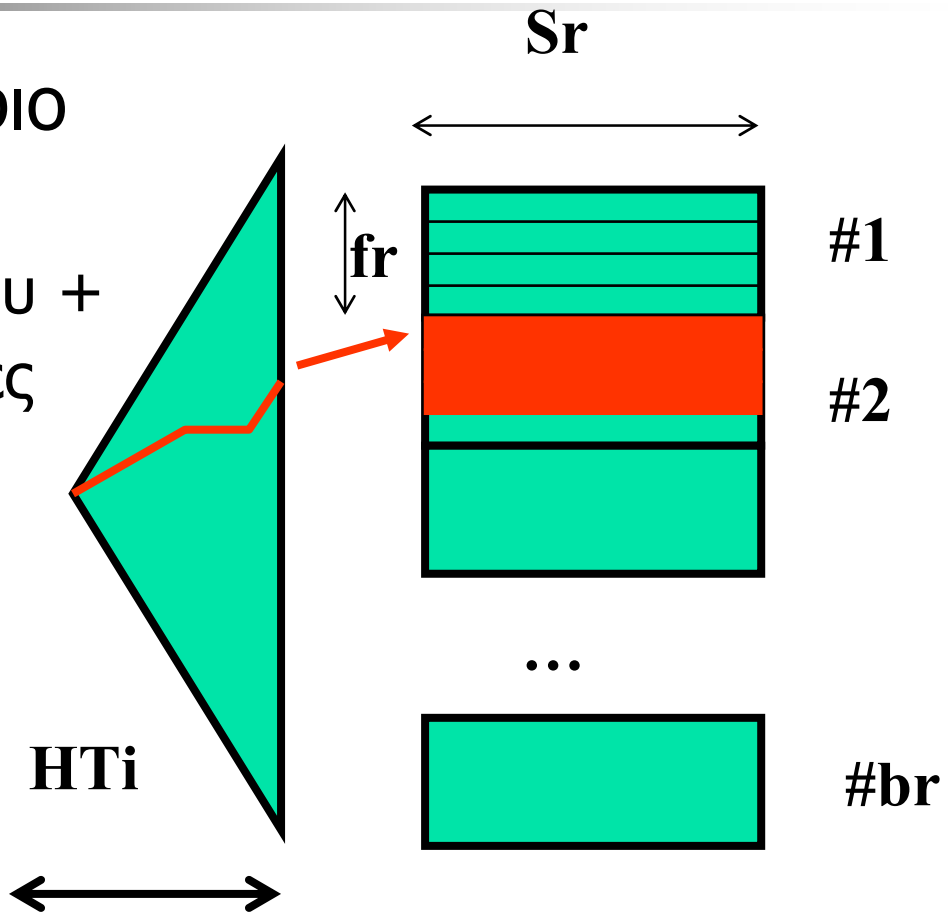
- E1: κόστος σειριακής αναζήτησης;
- A1: 500 προσπελάσεις δίσκου
- E2: υποθέστε ένα ευρετήριο συστάδων στο όνομα-υποκαταστήματος – κόστος;

Εκτίμηση κόστους & δημιουργία πλάνου

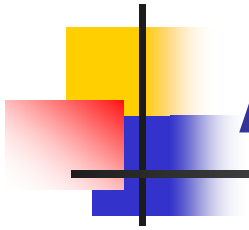
μέθοδος#3: ευρετήριο
- κόστος;
  - Επίπεδα ευρετηρίου +
  - μπλοκς με αρμόδιες πλειάδες

περίπτωση#2: δευτ. κλειδί – συσταδοποιημένο ευρετήριο

HTi + SC(A,r)/fr



Αριθμητικά παραδείγματα



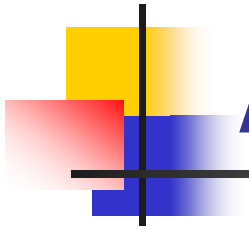
- A2:

  HTi +

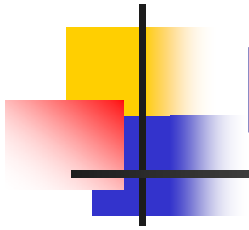
  SC(όνομα-υποκαταστήματος, λογαριασμός)/f-account

- HTi: 50 τιμές, με παράγοντας διακλάδωσης 20 -> HT=2 επίπεδα (log(50)/log(20) = 1+)
- SC(..)= # αρμόδιες εγγραφές =
  - nr/V(A,r) = 10,000/50 = 200 πλειάδες
  - SC/f: έκταση 200/20 μπλοκς= 10 μπλοκς

Αριθμητικά παραδείγματα



- A2 τελική απάντηση:
  2+10= 12 προσπελάσεις μπλοκς
- (αντί 500 προσπελάσεις μπλοκς σειριακής αναζήτησης)
- σημείωση: in all fairness
  - σειριακές αναζητήσεις δίσκου: ~2msec (ή και λιγότερο)
  - Τυχαίες προσπελάσεις μπλοκς: ~10msec

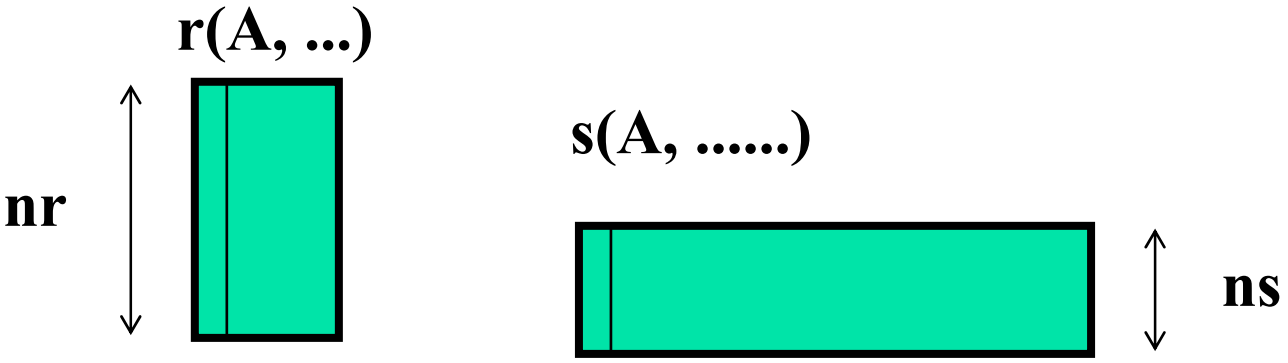


Επισκόπηση- λεπτομερώς

- Κίνητρο- Γιατί βελτιστ. Ερωτήματος;
- Ισοδυναμία εκφράσεων
- Εκτίμηση κόστους
- Κόστος ευρετηρίων
- Στρατηγικές συνένωσης (join)

2-way joins

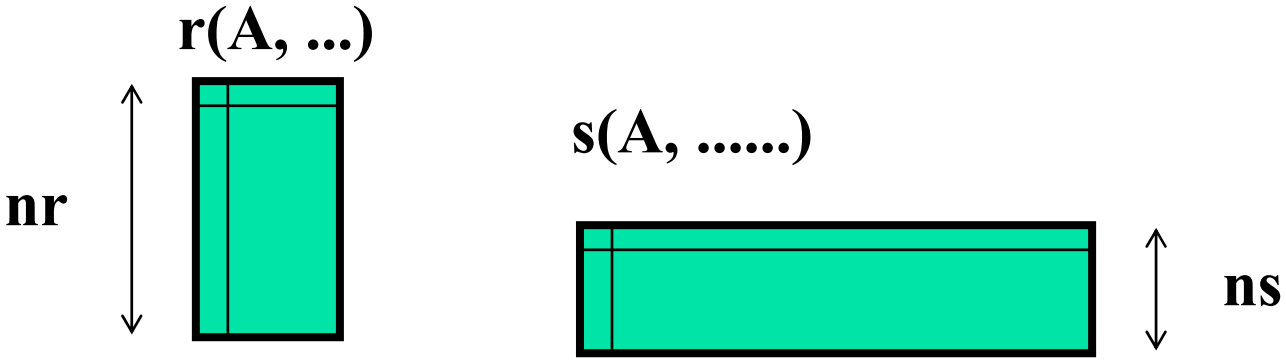
- αλγόριθμος(-οι) για r JOIN s?
- nr, ns πλειάδες κάθε σχεση



2-way joins

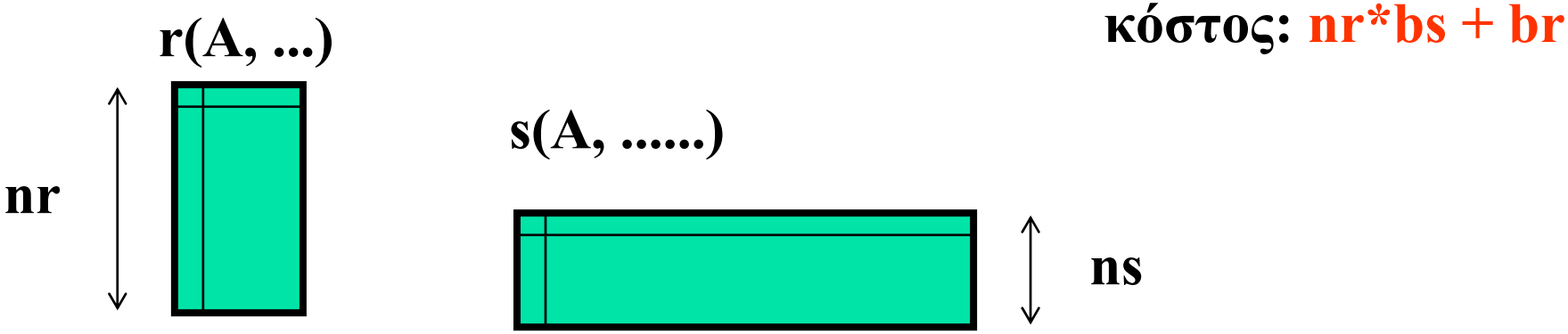
- Αλγόριθμος #0: (αφελής) εμφωλευμένες επαναλήψεις (**ΑΡΓΟ**!)

```
for each tuple tr of r
    for each tuple ts of s
        print, if they match
```



2-way joins

- Αλγόριθμος #0: γιατί είναι κακή?
- πόσες προσπελάσεις δίσκου ('br' και 'bs' είναι ο αριθμός των μπλοκς για 'r' και 's')?

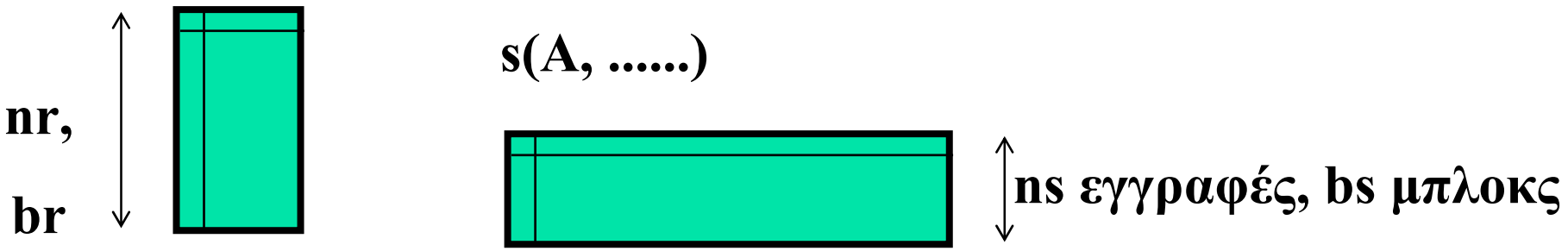


κόστος: nr*bs + br

2-way joins

- Αλγόριθμος #1: Blocked εμφωλευμένων επαναλήψεων συνένωση
  - read in a block of r
    - read in a block of s
      - print matching tuples

κόστος: br + br * bs

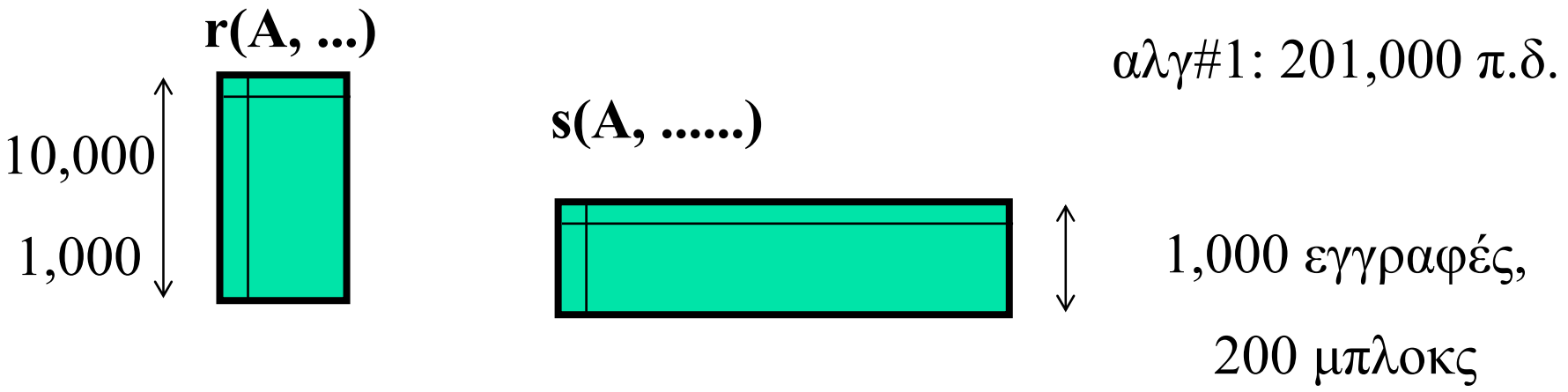


2-way joins

- Αριθμητικό παράδειγμα:
  - nr = 10,000 πλειάδες, br = 1,000 μπλοκς
  - ns = 1,000 πλειάδες, bs = 200 μπλοκς

αλγ#0: 2,001,000 π.δ.

αλγ#1: 201,000 π.δ.



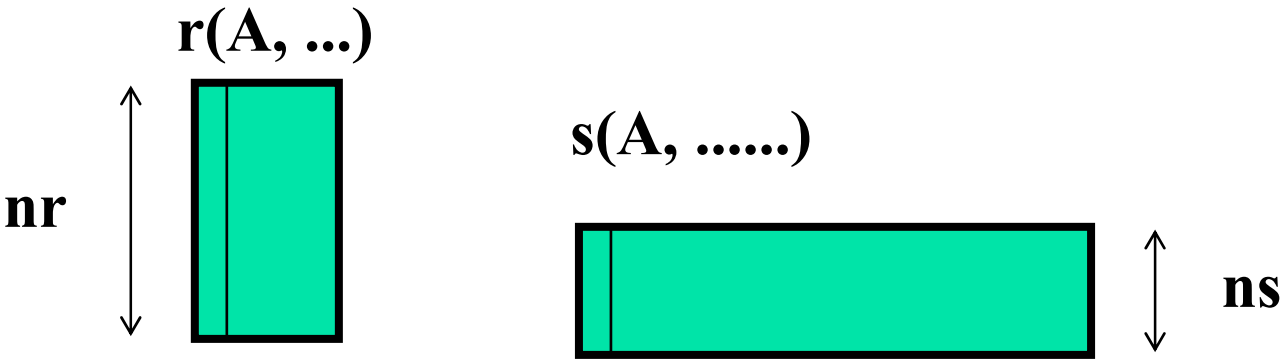
2-way joins

- Παρατήρηση 1: Αλγ#1: ασύμετρος:
  - κόστος: br + br * bs – ανάστροφοι ρόλοι:
  - κόστος= bs + bs*br
- Καλύτερη επιλογή?

Μικρότερη σχέση στη εξωτερική επανάληψη

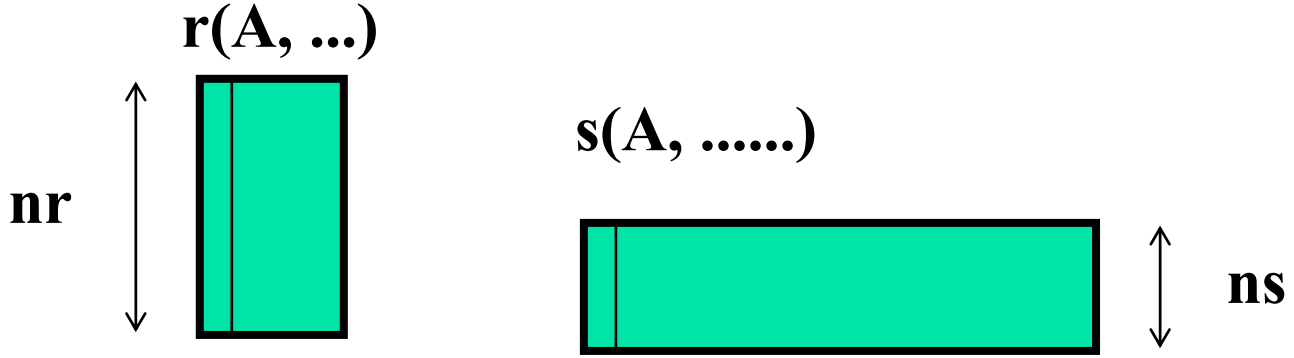
2-way joins

- Άλλοι αλγόριθμοι για r JOIN s;
- nr, ns πλειάδες το καθένα



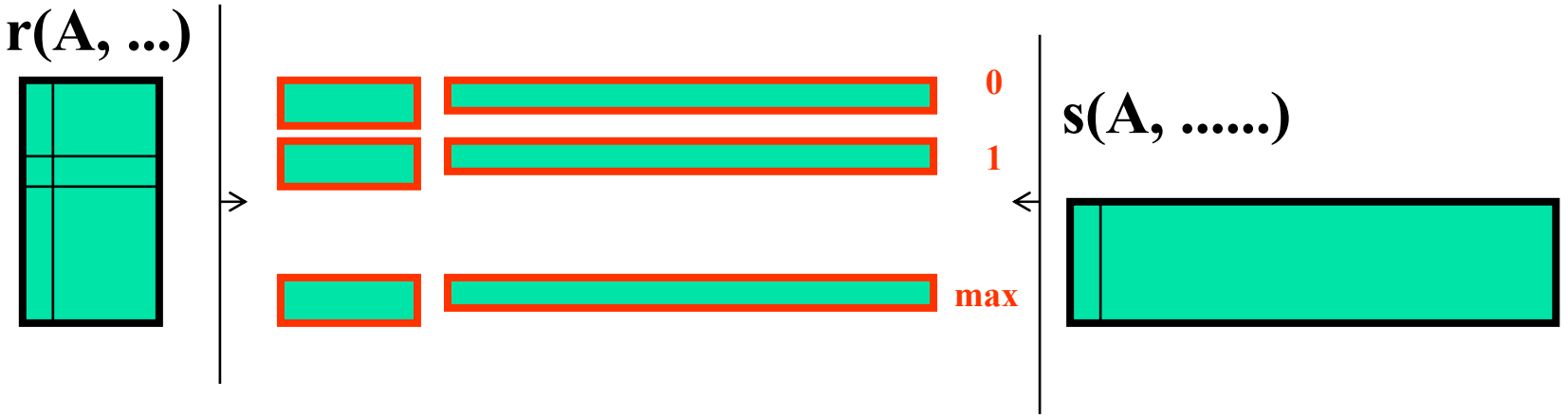
2-way joins - other algo's

- **sort-merge**
  - Ταξινόμησε 'r'; Ταξινόμησε 's'; Merge ταξινομημένες εκδόσεις των r και s.
  - Καλό, αν η μια ή και οι δύο σχέσεις είναι
  - ηδη ταξινομημένες

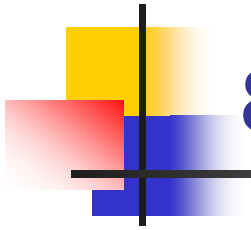


2-way joins - other algo's

- hash join:
  - hash 'r' into (0, 1, ..., 'max') buckets
  - hash 's' into buckets (same hash function)
  - join each pair of matching buckets



Δομή των βελτιστοποιητών επερωτήσεων:

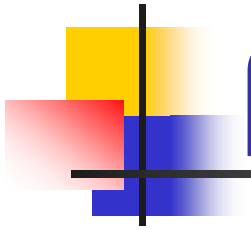


Περισσότερα heuristics από Oracle, Sybase and Starburst (-> DB2) : στο βιβλίο

Γενικά: Βελτιστοποίηση Επερωτήσεων: Είναι πολύ σημαντική για μεγάλες βάσεις δεδομένων

('**explain select** <sql-statement>' δείχνει το πλάνο)

Συμπεράσματα – βήματα βελτ.Επερ.



- φέρτε το ερώτημα στην εσωτερική μορφή (πχ., parse tree)
- … στην ‘κανονική μορφή’ (συντακτική Βελτ.Ερωτ.)
- δημιουργήστε εναλλακτικά σχέδια
  - επιλογές (απλές; σύνθετα κατηγορήματα)
  - ταξινόμηση; προβολές
  - συνενώσεις
- εκτιμήστε το κόστος, επιλέξτε το καλύτερο