# *Υπολογισμός Σχεσιακών Πράξεων*

## Κεφάλαιο 14 (μέρος Α - συζεύξεις)

# *Σχεσιακές Πράξεις*

- Θα δούμε πως γίνεται η υλοποίηση των:
  - *Επιλογή* (σ) επιλέγει υποσύνολο εγγραφών πίνακα.
  - *Προβολή* (π) διαγράφει ανεπιθύμητες στήλες πίνακα.
  - *Σύζευξη* (⋈) επιτρέπει συνδυασμό δύο πινάκων.
  - *Διαφορά* (-) εγγραφές που ανήκουν στον πίνακα 1 αλλά όχι στον πίνακα 2.
  - *Ένωση* (∪) εγγραφές που ανήκουν και στον πίνακα 1 και στον πίνακα 2.
  - *Συνάθροιση* (SUM, MIN, κλπ.) και GROUP BY
- Εφόσον κάθε πράξη επιστρέφει πίνακα, μπορεί να γίνει *σύνθεση* των πράξεων! Αφού περιγράψουμε την υλοποίηση των πράξεων, θα δούμε πώς μπορεί να *βελτιστοποιηθούν* αιτήματα που προκύπτουν από τη σύνθεσή τους.

# *Σχήμα για τα παραδείγματα*

Sailors (*sid*: integer, *sname*: string, *rating*: integer, *age*: real)
Reserves (*sid*: integer, *bid*: integer, *day*: date, *rname*: string)

- Όπως το παλιό σχήμα – προστέθηκε το *rname* για παραλλαγές στα αιτήματα.
- Reserves:
  - Κάθε εγγραφή είναι 40 bytes, 100 εγγραφές ανά σελίδα, 1000 σελίδες.
- Sailors:
  - Κάθε εγγραφή είναι 50 bytes, 80 εγγραφές ανά σελίδα, 500 σελίδες.

# *Συζεύξεις ισότητας με μια κοινή στήλη*

```
SELECT  *
FROM    Reserves R1, Sailors S1
WHERE   R1.sid=S1.sid
```

- Στην άλγεβρα: R⊠S. Πολύ συνηθισμένες! Πρέπει να βελτιστοποιηθούν με προσοχή. Το RXS είναι τεράστιος πίνακας, άρα RXS ακολουθούμενο από επιλογή δεν είναι αποτελεσματικό.
- Υπόθεση: M σελίδες στον R, $p_R$ εγγραφές ανά σελίδα, N σελίδες στον S, $p_S$ εγγραφές ανά σελίδα.
  - Στα παραδείγματά μας, R είναι ο Reserves και S ο Sailors.
- Αργότερα, θα δούμε πιο πολύπλοκες συνθήκες συζεύξεων.
- *Μετρική κόστους*: # I/O. Αγνοούμε κόστος εξόδου.

# *Σύζευξη με Απλή Εμφώλευση Βρόχων*

foreach tuple r in R do
  foreach tuple s in S do
    if $r_i$ == $s_j$ then add <r, s> to result

- Για κάθε εγγραφή του *εξωτερικού* πίνακα R, σαρώνουμε ολόκληρο τον *εσωτερικό* πίνακα S.
  - Κόστος: M + $p_R$ * M * N = 1000+100*1000*500 I/O.
- Σύζευξη με Εμφώλευση Βρόχων κατά Σελίδα: Για κάθε *σελίδα* του R, διάβασε κάθε *σελίδα* του S, και γράψε στην έξοδο ταιριαστά ζεύγη εγγραφών <r, s>, όπου r ανήκει στη σελίδα του R και s ανήκει στη σελίδα του S.
  - Κόστος: M + M*N = 1000 + 1000*500 I/O
  - Αν ο μικρότερος πίνακας (S) είναι εξωτερικός, Κόστος = 500 + 500*1000

# *Σύζευξη με Εμφ. Βρόχων και Ευρετήριο*

foreach tuple r in R do
  foreach tuple s in S where $r_i$ == $s_j$ do
    add <r, s> to result

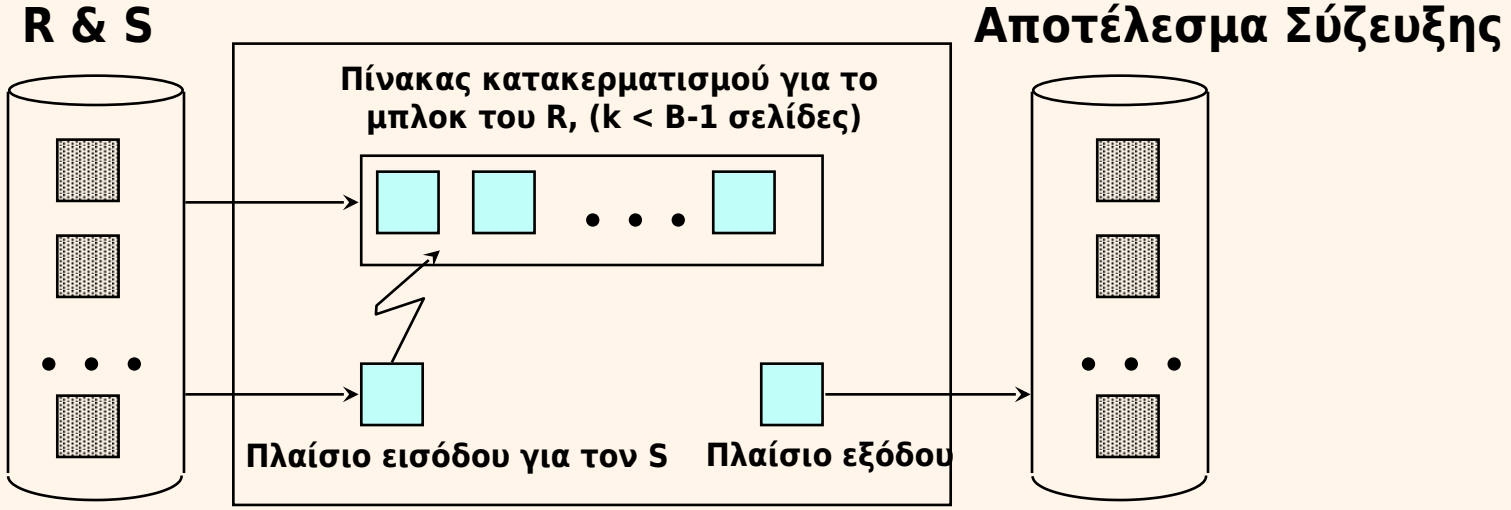
- Αν υπάρχει ευρετήριο στη στήλη σύζευξης στον ένα πίνακα (έστω στον S), μπορούμε να τον κάνουμε εσωτερικό για να το εκμεταλλευτούμε.
  - Κόστος: M + ( (M*$p_R$) * κόστος εύρεσης ταιριαστών εγγραφών του S)
- Για κάθε εγγραφή του R, το κόστος αναζήτησης του S μέσω ευρετηρίου είναι 1.2 για ευρετήριο κατακερματισμού, 2-4 για B+tree. Το κόστος εύρεσης των εγγραφών του S (υποθέτοντας Εναλλακτική (2) ή (3) για τις καταχωρίσεις δεδομένων) εξαρτάται από τη συγκρότηση.
  - Συγκροτημένο ευρετήριο: 1 I/O (συνήθως)
    Μη-συγκροτημένο: μέχρι 1 I/O ανά ταιριαστή εγγραφή του S.

# *Παραδείγματα: Σύζευξη με Εμφ. Βρόχων*

- Ευρετήριο Κατακερματισμού (Εναλλακτική 2) στο *sid* του Sailors (ως εσωτερικός πίνακας):
  - Σάρωσε τον Reserves: 1000 I/O, 100*1000 εγγραφές.
  - Για κάθε εγγραφή του Reserves: 1.2 I/O για την εύρεση της καταχώρισης δεδομένων στο ευρετήριο, συν 1 I/O για την εύρεση της (ακριβώς μίας) ταιριαστής εγγραφής του Sailors. Σύνολο: 220,000 I/O.
- Ευρετήριο Κατακερματισμού (Εναλλακτική 2) στο *sid* του Reserves (ως εσωτερικός πίνακας):
  - Σάρωσε τον Sailors: 500 I/O, 80*500 εγγραφές.
  - Για κάθε εγγραφή του Sailors: 1.2 I/O για την εύρεση της σελίδας με τις καταχωρίσεις δεδομένων στο ευρετήριο, συν το κόστος για την ανάκτηση των ταιριαστών εγγραφών του Reserves. Υποθέτοντας ομοιόμορφη κατανομή, 2.5 κρατήσεις ανά ναύτη (100,000 / 40,000). Κόστος ανάκτησης τους 1 ή 2.5 I/O ανάλογα με το αν το ευρετήριο είναι συγκροτημένο.

# *Σύζευξη με Εμφ. Βρόχων κατά Μπλοκ*

- Χρησιμοποίησε ένα πλαίσιο ως είσοδο για τη σάρωση του εσωτερικού πίνακα S, ένα πλαίσιο ως έξοδο, και όλα τα υπόλοιπα πλαίσια για την υποδοχή "μπλοκ" σελίδων του R.
  - Για κάθε ταίριασμα εγγραφής r του R-μπλοκ, με s του πλαισίου του S, πρόσθεσε το <r, s> στο αποτέλεσμα. Μετά διάβασε το επόμενο μπλοκ του R, σάρωσε τον S, κλπ.



# *Παραδείγματα: Σύζευξη με Εμφ. Βρόχων κατά Μπλοκ*

- Κόστος: Σάρωση εξωτερικού + #εξωτερικών μπλοκ * σάρωση εσωτερικού
  - #εξωτερικά μπλοκ = $\lceil$# σελίδων εξωτερικού / μέγεθος μπλοκ$\rceil$
- Με Reserves (R) ως εξωτερικό, και 100 σελίδες του R:
  - Κόστος σάρωσης R = 1000 I/O, για συνολικά 10 *μπλοκ*.
  - Ανά μπλοκ του R, σάρωση του Sailors (S): 10*500 I/O.
  - Αν υπάρχει χώρος για μόνο 90 σελίδες του R, θα σαρώναμε τον S 12 φορές.
- Με μπλοκ των 100 σελίδων του Sailors ως εξωτερικός:
  - Κόστος σάρωσης του S = 500 I/O, για συνολικά 5 μπλοκ.
  - Ανά μπλοκ του S, σάρωση του Reserves (R): 5*1000 I/O.
- Με *σειριακές αναγνώσεις*, η ανάλυση αλλάζει: ίσως συμφέρει να μοιράσουμε τα πλαίσια εξίσου στους R και S.

# *Σύζευξη με Ταξινόμηση και Συγχώνευση*

- Ταξινόμησε τους R και S στην κοινή στήλη σύζευξης, μετά σάρωσέ τους για να τους συγχωνεύσεις (ως προς τη στήλη σύζευξης) και γράψε τις εγγραφές που προκύπτουν.
  - Σάρωσε τον R μέχρι *τρέχουσα εγγραφή του R >= τρέχουσα εγγραφή του S*, μετά σάρωσε τον S μέχρι *τρέχουσα εγγραφή του S >= τρέχουσα εγγραφή του R*. Συνέχισε μέχρι *τρέχουσα εγγραφή του R = τρέχουσα εγγραφή του S*.
  - Στο σημείο αυτό, όλες οι εγγραφές του R με την ίδια τιμή στο Ri (*τρέχουσα ομάδα του R*) και όλες οι εγγραφές του S με ίδια τιμή στο Sj (τρέχουσα ομάδα του *S*) *ταιριάζουν*. Γράψε στην έξοδο το <r, s> για όλα αυτά τα ζεύγη τέτοιων εγγραφών.
  - Μετά, συνέχισε τη σάρωση των R και S.
- Ο R σαρώνεται μια φορά. Κάθε ομάδα του S σαρώνεται μια φορά για κάθε ταιριαστή εγγραφή του R. (Είναι πιθανό πολλαπλές σαρώσεις μιας ομάδας του S να βρίσκουν τις απαραίτητες σελίδες στην ενδιάμεση μνήμη.)

# *Παράδειγμα: Σύζευξη με Ταξινόμηση και Συγχώνευση*

| sid | sname | rating | age |
|---|---|---|---|
| 22 | dustin | 7 | 45.0 |
| 28 | yuppy | 9 | 35.0 |
| 31 | lubber | 8 | 55.5 |
| 44 | guppy | 5 | 35.0 |
| 58 | rusty | 10 | 35.0 |

| sid | bid | day | rname |
|---|---|---|---|
| 28 | 103 | 12/4/96 | guppy |
| 28 | 103 | 11/3/96 | yuppy |
| 31 | 101 | 10/10/96 | dustin |
| 31 | 102 | 10/12/96 | lubber |
| 31 | 101 | 10/11/96 | lubber |
| 58 | 103 | 11/12/96 | dustin |

- Κόστος: ταξινόμηση M + ταξινόμηση N + (M+N)
  - Το κόστος σάρωσης, M+N, θα μπορούσε να είναι M*N (πολύ απίθανο!)
- Με 35, 100 ή 300 σελίδες ενδιάμεσης μνήμης, και ο Reserves και ο Sailors μπορούν να ταξινομηθούν σε 2 περάσματα. Συνολικό κόστος σύζευξης: 7500.

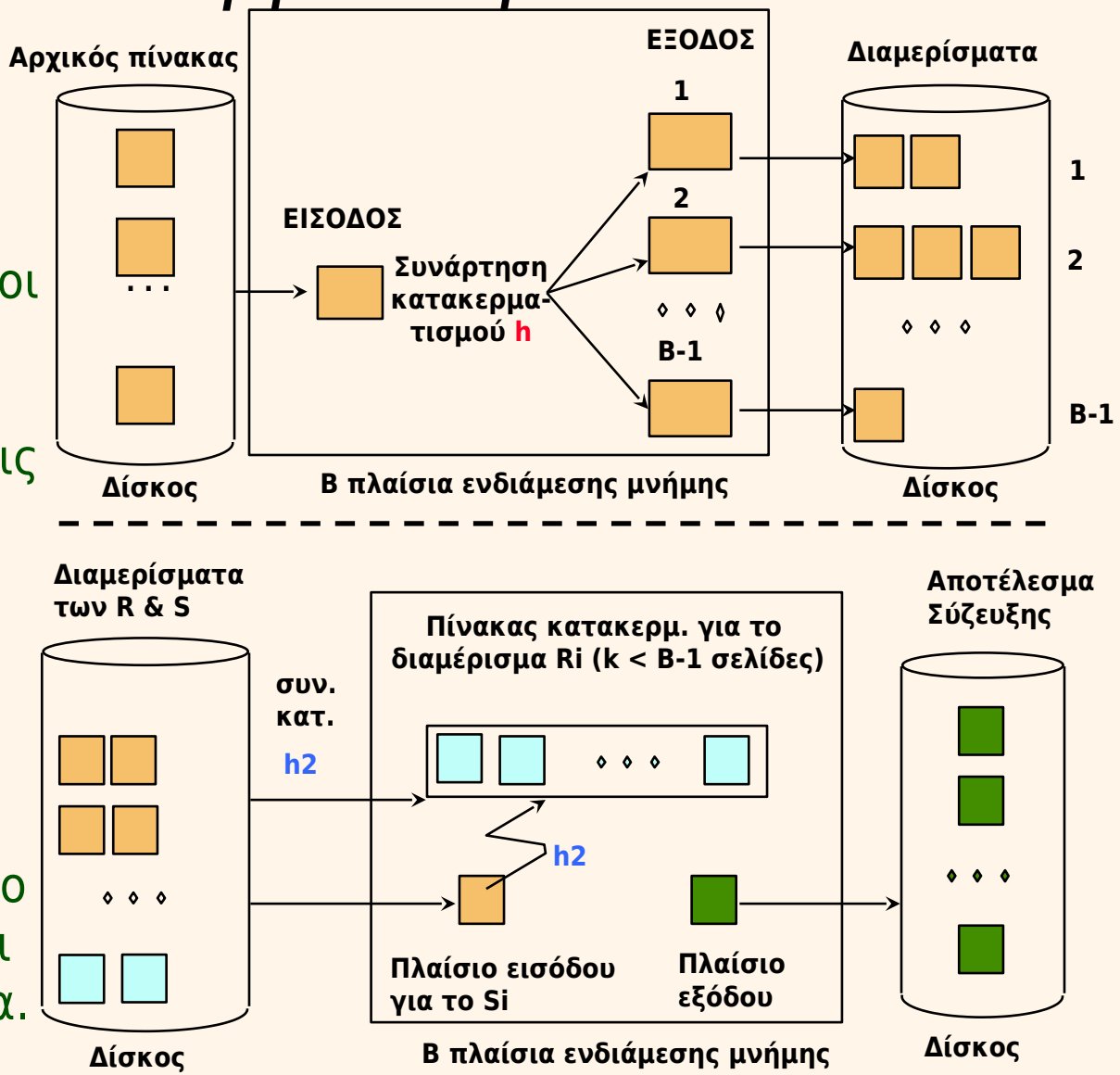
(Κόστος σύζευξης με εμφ. βρόχων ανά μπλοκ*: 2500 ως 15000 I/O*)

# *Βελτίωση της Σύζευξης με Ταξ+Συγχ*

- Μπορούμε να ενώσουμε τις φάσεις ταξινόμησης των R και S με τη συγχώνευση που απαιτείται για τη σύζευξη.
  - Με B > $\sqrt{L}$, όπου *L* είναι το μέγεθος του μεγαλύτερου πίνακα, και χρησιμοποιώντας τη βελτίωση στην ταξινόμηση που παράγει συρμούς μήκους 2B στο Πέρασμα 0, το #συρμών του κάθε πίνακα είναι < B/2.
  - Δεσμεύουμε 1 σελίδα ανά συρμό για κάθε πίνακα, και "συγχωνεύουμε" ενώ ελέγχουμε τη συνθήκη σύζευξης.
  - Κόστος: διάβασε+γράψε κάθε πίνακα στο Πέρασμα 0 + διάβασε κάθε πίνακα στο (μοναδικό) πέρασμα της συγχώνευσης (+ γράψε τις εγγραφές που ανήκουν στο αποτέλεσμα).
  - Στο παράδειγμα, το κόστος πέφτει από 7500 σε 4500 I/O.
- Στην πράξη, το κόστος της σύζευξης με ταξ+συγχ, όπως και το κόστος της εξωτ. ταξινόμ. είναι *γραμμικό*.

# Σύζευξη με Κατακερματισμό

- Διαμέρισε και τους δυο πίνακες με τη συνάρτηση κατακερματισμού **h**: οι εγγραφές του R στο διαμέρισμα i θα ταιριάζουν μόνο με τις εγγραφές του S στο διαμέρισμα i.
- Διάβασε ένα διαμέρισμα του R, κατακερμάτισέ το με την **h2 (<> h!)**. Σάρωσε το αντίστοιχο διαμέρισμα του S, και ψάξε για ταιριάσματα.



# *Παρατηρήσεις για τη Σύζευξη με Κατακερματισμό*

- #διαμερισμάτων k < B-1 (γιατί;), και B-2 > μέγεθος μεγαλύτερου διαμερίσματος που χωρά στη μνήμη. Υποθέτοντας ομοιόμορφη κατανομή στα μεγέθη των διαμερισμάτων και μεγιστοποιώντας το k:
  - k= B-1, και M/(B-1) < B-2, δηλαδή, B πρέπει να είναι > $\sqrt{M}$
- Απαιτείται ακόμη λίγη μνήμη αν κατασκευάσουμε έναν πίνακα κατακερματισμού για να επιταχύνουμε το ταίριασμα των εγγραφών.
- Αν η συνάρτηση κατακερματισμού δε διαμερίζει ομοιόμορφα, ένα ή παραπάνω διαμερίσματα του R μπορεί να μη χωράνε στη μνήμη. Μπορούμε να εφαρμόσουμε την τεχνική της σύζευξης με κατακερματισμό αναδρομικά για να συζεύξουμε ένα τέτοιο διαμέρισμα του R με το αντίστοιχο διαμέρισμα του S.

# *Κόστος Σύζευξης με Κατακερματισμό*

- Στη φάση διαμερισμού, διαβάζουμε+γράφουμε και τους δυο πίνακες: 2(M+N). Στη φάση ταιριά-σματος, διαβάζουμε και τους δυο πίνακες: M+N.
- Στο παράδειγμά μας, έχουμε συνολικά 4500 I/O.
- Σύζευξη με Ταξ+Συγχ vs. με Κατακερματισμό:
  - Αν έχουμε την ελάχιστη μνήμη (*ποιά είναι αυτή για τη κάθε μέθοδο;*) και οι δυο έχουν κόστος 3(M+N) I/O. Η Σύζευξη με Κατακερματισμό υπερέχει αν οι πίνακες διαφέρουν σημαντικά ως προς το μέγεθος. Επίσης, μπορεί να παραλληλοποιηθεί.
  - Η Σύζευξη με Ταξ+Συγχ δεν επηρεάζεται τόσο από την κατανομή των δεδομένων – το αποτέλεσμα είναι ταξινομημένο.

# *Γενικές Συνθήκες Σύζευξης*

- Ισότητες με πολλά πεδία (π.χ., *R.sid=S.sid* AND *R.rname=S.sname*):
  - Για τη Σύζευξη με Εμφ. Βρόχων και Ευρετήριο, κατασκεύασε ευρετήριο στα *<sid, sname>* (αν ο S είναι εσωτερικός) ή χρησιμοποίησε υπάρχοντα ευρετήρια στο *sid* ή στο *sname*.
  - Για τη Σύζευξη με Ταξ+Συγχ και με Κατακερματισμό, ταξινόμησε/διαμέρισε σε συνδυασμό των δυο πεδίων σύζευξης.
- Συνθήκες ανισότητας (π.χ., *R.rname < S.sname*):
  - Για τη Σύζευξη με Εμφ. Βρόχων και Ευρετήριο, χρειαζόμαστε (συγκροτημένο!) ευρετήριο B+tree.
    - Αναζητήσεις με διάστημα πάνω στον εσωτερικό πίνακα, # ταιριασμάτων πιθανόν πολύ μεγαλύτερο από τις συζεύξεις με ισότητα.
  - Συζεύξεις με Ταξ+Συγχ ή με Κατακερματισμό ανεφάρμοστες.
  - Σε αυτό το σενάριο, η Σύζευξη με Εμφ. Βρόχους κατά μπλοκ μπορεί να αποτελεί την καλύτερη μέθοδο σύζευξης.