

Επεξεργασία και Βελτιστοποίηση Ερωτήσεων



- ③ Ο βελτιστοποιητής ερωτήσεων (query optimizer)
- ③ Μετασχηματισμός εκφράσεων σχεσιακής άλγεβρας
- ③ Υπολογισμός μεγεθών πράξεων σχεσιακής άλγεβρας (επιλογή σύνδεση, άλλες πράξεις)
- ③ Τρόποι εκτέλεσης (και μέτρηση κόστους) πράξεων σχεσιακής άλγεβρας
- ③ Βελτιστοποίηση βάσει κόστους vs. Ευριστική βελτιστοποίηση

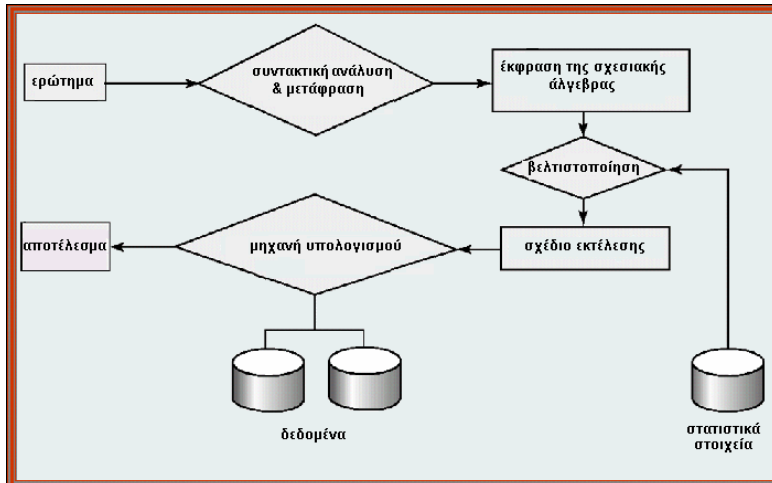
Βασική πηγή διαφανειών: Silberschatz et al., "Database System Concepts", 4/e

Επεξεργασία ερωτήσεων



- 🕒 Όταν ο client υποβάλει μια ερώτηση στο server (συνήθως σε δηλωτική γλώσσα, π.χ. SQL), ο server εκτελεί τα εξής βήματα:
 1. **Συντακτική ανάλυση** (parsing) και **μετάφραση** (translation)
 - 🕒 Μετατροπή σε έκφραση σχεσιακής άλγεβρας
 2. **Βελτιστοποίηση** (optimization)
 - 🕒 Σύνθεση εναλλακτικών πλάνων εκτέλεσης (ισοδύναμες εκφράσεις σχεσιακής άλγεβρας – χρήση ή μη ευρετηρίων για τις επιλογές – τρόποι υλοποίησης συνδέσεων)
 3. **Εκτέλεση πλάνου** (query plan execution) και **επιστροφή αποτελέσματος**

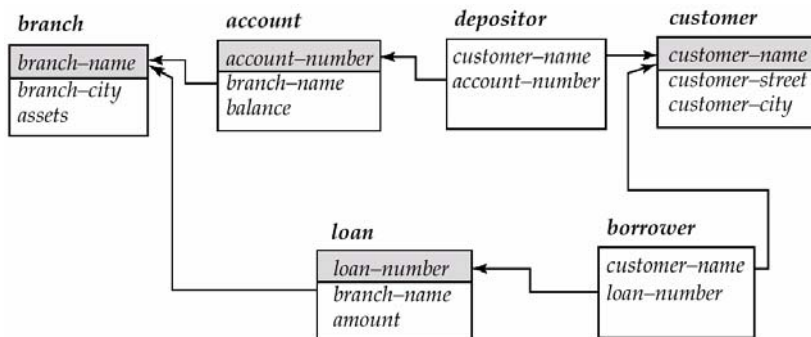
Η επεξεργασία ερωτήσεων σε μια εικόνα



Διάγραμμα Σχεσιακού σχήματος



🕒 Βάση δεδομένων τράπεζας
(πελάτες – υποκαταστήματα – καταθετικοί λογαριασμοί – δάνεια)



Επεξεργασία ερωτήσεων (Συν.)



③ Συντακτική ανάλυση και Μετάφραση

③ Η ερώτηση SQL

```
SELECT customer-name
FROM account JOIN depositor
WHERE balance < 2500
```

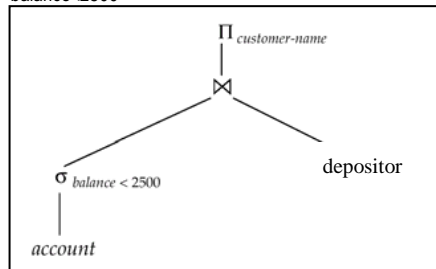
- ③ ελέγχεται για συντακτικά / σημασιολογικά σφάλματα και στη συνέχεια μεταφράζεται στην ισοδύναμη έκφραση σχεσιακής άλγεβρας:

$\Pi_{customer-name}(\sigma_{balance < 2500}(account \bowtie depositor))$

③ Βελτιστοποίηση .. $\Pi_{customer-name}(\sigma_{balance < 2500}(account) \bowtie depositor)$

③ Εκτέλεση πλάνου και επιστροφή αποτελέσματος

- ③ Η μηχανή υπολογισμού δέχεται ως είσοδο ένα **σχέδιο εκτέλεσης** (query execution plan - QEP), το εκτελεί και επιστρέφει τις απαντήσεις στον client.



5

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιούτας Σπύρος

Βελτιστοποίηση ερωτήσεων



- ③ Μια έκφραση της σχεσιακής άλγεβρας ενδέχεται να έχει πολλές ισοδύναμες εκφράσεις

③ Π.χ. η έκφραση $\sigma_{balance < 2500}(\Pi_{balance}(account))$ ισοδυναμεί με την έκφραση $\Pi_{balance}(\sigma_{balance < 2500}(account))$

- ③ Κάθε πράξη της σχεσιακής άλγεβρας μπορεί να υλοποιηθεί με χρήση κάποιου από πολλούς διαφορετικούς αλγορίθμους

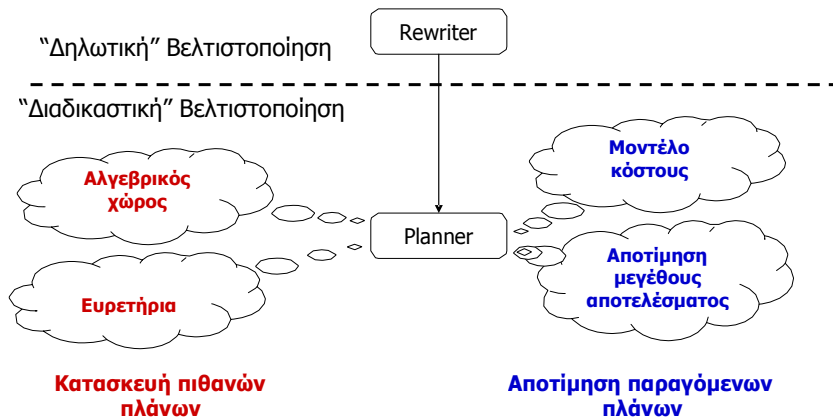
- ③ Άρα, δεν αρκεί μόνο ο προσδιορισμός της πράξης, αλλά χρειάζεται να προσδιοριστεί και ο αλγόριθμος που θα χρησιμοποιηθεί για την υλοποίησή της.

- ③ Ο καθορισμός μιας λεπτομερούς στρατηγικής για τον υπολογισμό μιας πράξης ονομάζεται **σχέδιο εκτέλεσης** (execution plan).

6

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιούτας Σπύρος

Αφαιρετική δομή του βελτιστοποιητή ερωτήσεων



7

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιούτας Σπύρος

1^ο επίπεδο βελτιστοποίησης - Rewriter



- ③ Υπάρχει ένα επίπεδο «**δηλωτικής**» **βελτιστοποίησης**, ή **επανεγγραφής**, όπου παράγουμε λογικά ισοδύναμους τρόπους να εκφράσουμε μια ερώτηση μέσω του **rewriter**
- ③ Αυτό συμπεριλαμβάνει, συνήθως:
 - ③ Μετατροπή εκφράσεων σε «βολική» μορφή (χρήση conjunctive normal forms – CNF)
 - ③ Απλοποίηση εμφωλευμένων ερωτήσεων
 - ③ Σημασιολογικά έξυπνες μετατροπές
- ③ Παραδείγματα:
 - ③ ... WHERE NOT (Attr₁ = 10 OR Attr₂ > 3) ΓΙΝΕΤΑΙ
... WHERE Attr₁ <> 10 AND Attr₂ <= 3
 - ③ ... WHERE Attr = YEAR('1994-09-08') ΓΙΝΕΤΑΙ
... WHERE Attr = 1994
 - ③ Δοθέντος: R₁.A₁=R₂.A₂ AND R₂.A₂=R₃.A₃ AND R₁.A₁>5
προστίθενται... R₁.A₁=R₃.A₃ AND R₂.A₂>5 AND R₃.A₃>5

8

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιούτας Σπύρος

2^ο επίπεδο βελτιστοποίησης - Planner



- ③ Υπάρχει ένα επίπεδο «**διαδικαστικής**» **βελτιστοποίησης**, όπου παράγουμε (όλα;) τα διαφορετικά πλάνα εκτέλεσης, μέχρι να διαλέξουμε το πιο αποδοτικό. Η εργασία αυτή ανατίθεται στον **planner**.
- ③ Ο planner οφείλει:
 - ③ Να **κατασκευάσει ένα σύνολο πλάνων**, με βάση
 - ③ Ένα αλγεβρικό χώρο για τη σειρά εκτέλεσης των λειτουργιών (π.χ., να αποφασίσει με ποια σειρά θα κάνει το $R \times S \times T$)
 - ③ Ένα σύνολο από μεθόδους προσπέλασης στα δεδομένα (π.χ., full-index scan, full table scan, ...)
 - ③ Να **αποτιμά κάθε πλάνο** που παράγει, μέχρι στο τέλος να βρει το πιο αποδοτικό, με βάση
 - ③ Ένα μοντέλο κόστους που προβλέπει το κόστος εκτέλεσης κάθε πλάνου (σε χρόνο, disk I/O, ...)
 - ③ Ένα μοντέλο πρόβλεψης του μεγέθους, κυρίως των ενδιάμεσων αποτελεσμάτων
 - ③ Προσοχή: η αποτίμηση είναι πάντα προσέγγιση/πρόβλεψη και όχι ακριβής υπολογισμός...

Βελτιστοποίηση ερωτήσεων (συν.)



- ③ **Βελτιστοποίηση ερωτήσεων**: επιλέγουμε από τα διάφορα ισοδύναμα σχέδια εκτέλεσης εκείνο με το μικρότερο κόστος.
 - ③ Το κόστος υπολογίζεται χρησιμοποιώντας στατιστικές πληροφορίες που προέρχονται από τον κατάλογο της βάσης
 - ③ Π.χ. το πλήθος των πλειάδων κάθε σχέσης, το μέγεθος των πλειάδων, κλπ.
- ③ Στη συνέχεια θα μελετήσουμε:
 - ③ Αλγεβρικό χώρο (εναλλακτικές εκφράσεις σχεσιακής άλγεβρας)
 - ③ Αλγορίθμους για την εκτέλεση των πράξεων της σχεσιακής άλγεβρας
 - ③ Υπολογισμό κόστους βασικών πράξεων

Μετασχηματισμός εκφράσεων Σχεσιακής Άλγεβρας



- ③ Δύο εκφράσεις σχεσιακής άλγεβρας καλούνται **ισοδύναμες** αν σε οποιοδήποτε έγκυρο στιγμιότυπο της ΒΔ οι δύο εκφράσεις παράγουν το ίδιο σύνολο πλειάδων.
 - ③ Σημείωση: η διάταξη των πλειάδων δεν παίζει ρόλο.
- ③ Ένας **κανόνας ισοδυναμίας** καθορίζει αν μια έκφραση της πρώτης μορφής είναι ισοδύναμη με μια έκφραση της δεύτερης μορφής
 - ③ Εκφράσεις της πρώτης μορφής μπορούν να αντικατασταθούν με εκφράσεις της δεύτερης μορφής, και αντίστροφα.

Κανόνες Ισοδυναμίας



1. Οι συζευκτικές επιλογές μπορούν να αναλυθούν σε μια ακολουθία ξεχωριστών επιλογών.
$$\sigma_{\theta_1 \wedge \theta_2}(E) = \sigma_{\theta_1}(\sigma_{\theta_2}(E))$$
2. Οι επιλογές είναι μεταβατικές.
$$\sigma_{\theta_1}(\sigma_{\theta_2}(E)) = \sigma_{\theta_2}(\sigma_{\theta_1}(E))$$
3. Σε μια ακολουθία από προβολές μόνο η τελευταία προβολή χρειάζεται, οι υπόλοιπες μπορούν να παραληφθούν.
$$\Pi_{t_1}(\Pi_{t_2}(\dots(\Pi_{t_n}(E))\dots)) = \Pi_{t_1}(E)$$
4. Οι επιλογές μπορούν να συνδυαστούν με Καρτεσιανά Γινόμενα και Θ-συνδέσεις.

$$\begin{aligned}\sigma_{\theta}(E_1 \times E_2) &= E_1 \bowtie_{\theta} E_2 \\ \sigma_{\theta_1}(E_1 \bowtie_{\theta_2} E_2) &= E_1 \bowtie_{\theta_1 \wedge \theta_2} E_2\end{aligned}$$

Κανόνες Ισοδυναμίας (συν.)



5. Οι λειτουργίες Θ -σύνδεσης (και φυσικής σύνδεσης) είναι μεταβατικές.

$$E_1 \bowtie_{\theta} E_2 = E_2 \bowtie_{\theta} E_1$$

6. (a) Οι λειτουργίες φυσικής σύνδεσης είναι προσεταιριστικές:

$$(E_1 \bowtie E_2) \bowtie E_3 = E_1 \bowtie (E_2 \bowtie E_3)$$

- (b) Οι λειτουργίες Θ -σύνδεσης είναι προσεταιριστικές με την ακόλουθη έννοια:

$$(E_1 \bowtie_{\theta_1} E_2) \bowtie_{\theta_2 \wedge \theta_3} E_3 = E_1 \bowtie_{\theta_1 \wedge \theta_3} (E_2 \bowtie_{\theta_2} E_3)$$

όπου η συνθήκη θ_2 περιλαμβάνει μόνο γνωρίσματα των σχέσεων E_2 και E_3 .

Κανόνες Ισοδυναμίας (συν.)



7. Η επιλογή αντιμετατίθεται με Θ -σύνδεση κάτω από τις ακόλουθες δύο συνθήκες:

(α) όταν όλα τα γνωρίσματα της συνθήκης θ_1 προέρχονται μόνο από τα γνωρίσματα μιας εκ των σχέσεων που συμμετέχουν στη σύνδεση (έστω της E_1).

$$\sigma_{\theta_1}(E_1 \bowtie_{\theta} E_2) = (\sigma_{\theta_1}(E_1)) \bowtie_{\theta} E_2$$

(β) όταν όλα τα γνωρίσματα της συνθήκης θ_1 προέρχονται μόνο από τα γνωρίσματα της E_1 ενώ τα γνωρίσματα της θ_2 προέρχονται μόνο από τα γνωρίσματα της E_2 .

$$\sigma_{\theta_1 \wedge \theta_2}(E_1 \bowtie_{\theta} E_2) = (\sigma_{\theta_1}(E_1)) \bowtie_{\theta} (\sigma_{\theta_2}(E_2))$$

Κανόνες Ισοδυναμίας (συν.)



8. Η προβολή αντιμετατίθεται με Θ -σύνδεση ως εξής:
(α) αν η προβολή Π περιλαμβάνει μόνο γνωρίσματα από το σύνολο $L_1 \cup L_2$:

$$\Pi_{L_1 \cup L_2}(E_1 \bowtie_{\theta} E_2) = (\Pi_{L_1}(E_1)) \bowtie_{\theta} (\Pi_{L_2}(E_2))$$

(β) Ας θεωρήσουμε μια σύνδεση $E_1 \bowtie_{\theta} E_2$.

- ③ Έστω L_1 και L_2 τα σύνολα των γνωρισμάτων των σχέσεων E_1 και E_2 , αντίστοιχα.
- ③ Έστω L_3 το σύνολο των γνωρισμάτων της σχέσης E_1 που συμμετέχουν στη συνθήκη θ , αλλά δεν ανήκουν στο σύνολο $L_1 \cup L_2$
- ③ Έστω L_4 το σύνολο των γνωρισμάτων της σχέσης E_2 που συμμετέχουν στη συνθήκη θ , αλλά δεν ανήκουν στο σύνολο $L_1 \cup L_2$

$$\Pi_{L_1 \cup L_2}(E_1 \bowtie_{\theta} E_2) = \Pi_{L_1 \cup L_2}((\Pi_{L_1 \cup L_3}(E_1)) \bowtie_{\theta} (\Pi_{L_2 \cup L_4}(E_2)))$$

Κανόνες Ισοδυναμίας (συν.)



9. Η ένωση και η τομή είναι μεταβατικές

$$\begin{aligned} E_1 \cup E_2 &= E_2 \cup E_1 \\ E_1 \cap E_2 &= E_2 \cap E_1 \end{aligned}$$

- η διαφορά δεν είναι μεταβατική

10. Η ένωση και η τομή είναι προσεταιριστικές.

$$\begin{aligned} (E_1 \cup E_2) \cup E_3 &= E_1 \cup (E_2 \cup E_3) \\ (E_1 \cap E_2) \cap E_3 &= E_1 \cap (E_2 \cap E_3) \end{aligned}$$

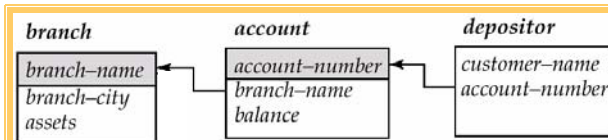
11. Η επιλογή αντιμετατίθεται με τις λειτουργίες \cup , \cap και $-$.

$$\sigma_{\theta}(E_1 - E_2) = \sigma_{\theta}(E_1) - \sigma_{\theta}(E_2)$$

ομοίως στην περίπτωση των τελεστών \cup και \cap

Επίσης: $\sigma_{\theta}(E_1 - E_2) = \sigma_{\theta}(E_1) - E_2$
ομοίως στην περίπτωση των τελεστών \cup και \cap

Παράδειγμα μετασχηματισμού



- ③ Ερώτηση: Βρες τα ονόματα όλων των πελατών που έχουν κάποιο λογαριασμό σε κάποιο υποκατάστημα που βρίσκεται στο Brooklyn.

$$\Pi_{\text{customer-name}} (\sigma_{\text{branch-city} = \text{"Brooklyn"}} (\text{branch} \bowtie (\text{account} \bowtie \text{depositor})))$$

- ③ Μετατροπή μέσω του κανόνα 7^ο:

$$\Pi_{\text{customer-name}} ((\sigma_{\text{branch-city} = \text{"Brooklyn"}} (\text{branch})) \bowtie (\text{account} \bowtie \text{depositor}))$$

- ③ Η όσο το δυνατόν ταχύτερη εκτέλεση της επιλογής μειώνει το μέγεθος της σχέσης της σύνδεσης

Παράδειγμα πολλαπλών μετασχηματισμών



- ③ Ερώτηση: Βρες τα ονόματα των πελατών που έχουν λογαριασμό σε κάποιο υποκατάστημα του Brooklyn και το υπόλοιπο του λογαριασμού > 1000.

$$\Pi_{\text{customer-name}} (\sigma_{\text{branch-city} = \text{"Brooklyn"} \wedge \text{balance} > 1000} (\text{branch} \bowtie (\text{account} \bowtie \text{depositor})))$$

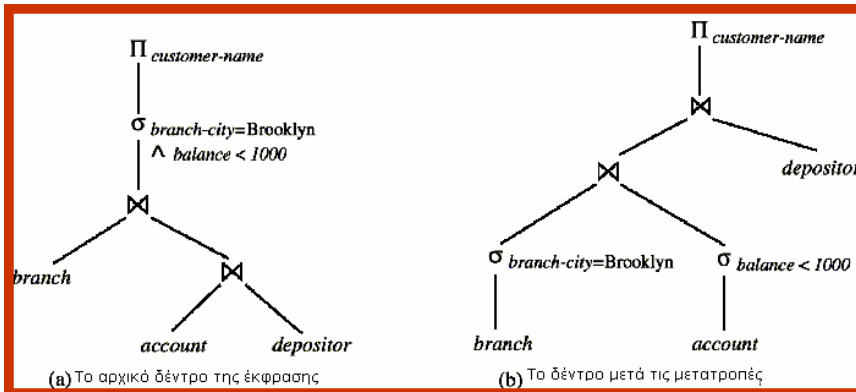
- ③ Μετατροπή μέσω του κανόνα 6^ο:

$$\Pi_{\text{customer-name}} ((\sigma_{\text{branch-city} = \text{"Brooklyn"} \wedge \text{balance} > 1000} (\text{branch} \bowtie (\text{account}))) \bowtie \text{depositor})$$

- ③ Μια δεύτερη μορφή προσφέρει τη δυνατότητα εκτέλεσης της επιλογής νωρίτερα, ως εξής

$$\Pi_{\text{customer-name}} ((\sigma_{\text{branch-city} = \text{"Brooklyn"}} (\text{branch}) \bowtie \sigma_{\text{balance} > 1000} (\text{account})) \bowtie \text{depositor})$$

Πολλαπλές μετατροπές (συν.)



19

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιούτας Σπύρος

Διάταξη συνδέσεων



- ③ Για όλες τις σχέσεις r_1, r_2 , και r_3 ισχύει:

$$(r_1 \bowtie r_2) \bowtie r_3 = r_1 \bowtie (r_2 \bowtie r_3)$$

- ③ Αν το αποτέλεσμα της σχέσης $r_2 \bowtie r_3$ είναι αρκετά μεγάλο ενώ το αποτέλεσμα της σχέσης $r_1 \bowtie r_2$ είναι αρκετά μικρό, επιλέγουμε τη διάταξη

$$(r_1 \bowtie r_2) \bowtie r_3$$

προκειμένου να υπολογίσουμε και να αποθηκεύσουμε μια μικρότερη ενδιάμεση σχέση.

20

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιούτας Σπύρος

Παράδειγμα διάταξης συνδέσεων



- ③ Έστω η έκφραση:

$\Pi_{\text{customer-name}} ((\sigma_{\text{branch-city} = \text{"Brooklyn"} (\text{branch})) \bowtie \text{account}} \bowtie \text{depositor})$

- ③ Μπορούμε να υπολογίσουμε την πράξη $\text{account} \bowtie \text{depositor}$ πρώτα, και στη συνέχεια να συνδέσουμε το αποτέλεσμα με το αποτέλεσμα της πράξης

$\sigma_{\text{branch-city} = \text{"Brooklyn"} (\text{branch})$

δυστυχώς το αποτέλεσμα της πράξης $\text{account} \bowtie \text{depositor}$ ενδέχεται να είναι πολύ μεγάλο.

- ③ Επειδή είναι πιο πιθανό ένα μικρό μέρος των πελατών της τράπεζας να έχουν λογαριασμούς σε υποκαταστήματα που βρίσκονται στο Brooklyn, είναι προτιμότερο να υπολογίσουμε πρώτα την πράξη

$\sigma_{\text{branch-city} = \text{"Brooklyn"} (\text{branch}) \bowtie \text{account}$

Μέτρηση κόστους ερωτήσεων



- ③ Το **κόστος μιας ερώτησης** ισούται με το συνολικό χρόνο που απαιτείται για την απάντησή της
- ③ Πολλοί παράγοντες επηρεάζουν το κόστος: οι προσπελάσεις στο δίσκο, η ισχύς της CPU, ακόμα και ο φόρτος του δικτύου
 - ③ Οι προσπελάσεις στο δίσκο που αποτελούν το βασικό κόστος υπολογισμού μιας ερώτησης είναι εύκολο να υπολογιστούν. Η μέτρηση γίνεται λαμβάνοντας υπόψη:
 - ③ Το πλήθος των αναζητήσεων
 - ③ Το πλήθος των αναγνώσεων σελίδων του δίσκου
 - ③ Το πλήθος των εγγραφών σελίδων στο δίσκο
 - ③ Το κόστος εγγραφής στο δίσκο είναι μεγαλύτερο από το κόστος ανάγνωσης από το δίσκο

Μέτρηση κόστους ερωτήσεων (συν.)

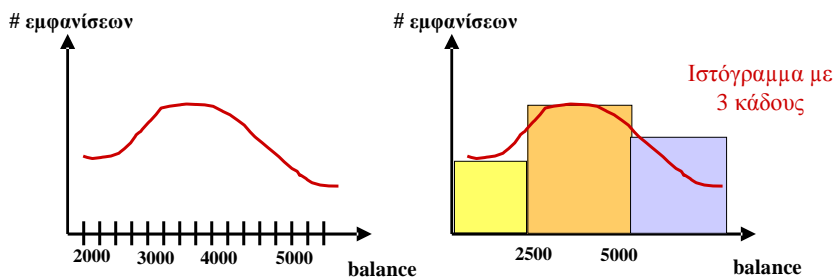


- ③ Για απλότητα χρησιμοποιούμε ως μέτρο του κόστους μιας ερώτησης **το πλήθος των σελίδων (blocks) που μεταφέρονται από το δίσκο** (κόστος ανάγνωσης δεδομένων)
 - ③ Για λόγους απλότητας δεν λαμβάνουμε υπόψη ούτε το κόστος της CPU
 - ③ Ούτε λαμβάνουμε υπόψη το κόστος εγγραφής των δεδομένων στο δίσκο
- ③ Το κόστος εξαρτάται από το μέγεθος του **buffer της κύριας μνήμης**
 - ③ Η ύπαρξη επιπλέον μνήμης μειώνει την ανάγκη για προσπελάσεις στο δίσκο
 - ③ Το ποσοστό της κύριας μνήμης που διατίθεται στον buffer εξαρτάται από τις άλλες διαδικασίες του λειτουργικού συστήματος που τρέχουν ταυτόχρονα και είναι δύσκολο να υπολογιστεί εκ των προτέρων

Εκτίμηση μεγέθους



- ③ Για να δουλέψουν οι συναρτήσεις κόστους, πρέπει να μπορούμε να αποτιμήσουμε το **μέγεθος των ενδιάμεσων αποτελεσμάτων**
- ③ Η πιο καλή τεχνική που έχουμε είναι τα **ιστογράμματα**
 - ③ Σ' ένα ιστογράμμα, διαιρούμε το εύρος των τιμών ενός πεδίου σε **κάδους** (buckets)
 - ③ Για κάθε τιμή που παίρνει το πεδίο, μετράμε τον αριθμό εμφανίσεων



Τρόποι εκτέλεσης και μέτρηση κόστους πράξεων σχεσιακής άλγεβρας



- ③ Επιλογή (με / χωρίς χρήση ευρετηρίου)
- ③ Σύνδεση
 - ③ Nested-loop join και οι παραλλαγές:
 - ③ Block-nested-loop join, Indexed-nested-loop join
 - ③ Sort-Merge join
 - ③ Hash join
- ③ Άλλες πράξεις (προβολή, συνάθροιση)
- ③ Σύνθετες εκφράσεις

Η πράξη Επιλογή



- ③ Χωρίς χρήση ευρετηρίου
 - ③ **Γραμμική αναζήτηση** (linear search). Σάρωση όλων των σελίδων του αρχείου και για κάθε εγγραφή μιας σελίδας έλεγχος αν ικανοποιεί τη συνθήκη της επιλογής.
 - ③ **Διαδική αναζήτηση** (binary search). Μπορεί να εφαρμοστεί μόνο αν το αρχείο είναι διατεταγμένο με βάση το γνώρισμα της επιλογής.
- ③ Με χρήση ευρετηρίου
 - ③ **Σάρωση ευρετηρίου** (Index scan) – αλγόριθμοι αναζήτησης που χρησιμοποιούν κάποιο ευρετήριο

Επιλογή: Γραμμική αναζήτηση



③ Σάρωση όλων των σελίδων του αρχείου και για κάθε εγγραφή μιας σελίδα έλεγχος αν ικανοποιεί τη συνθήκη της επιλογής.

- ③ Κόστος (# σελίδων δίσκου που σαρώνονται) = b_r
 - ③ Το b_r συμβολίζει το πλήθος των σελίδων που περιέχουν εγγραφές της σχέσης r
- ③ Αν η συνθήκη της επιλογής αναφέρεται σε κάποιο γνώρισμα - κλειδί, τότε (μέσο) κόστος = $b_r/2$
 - ③ η αναζήτηση σταματά όταν βρεθεί η εγγραφή

record 0	A-102	Perryridge	400
record 1	A-305	Round Hill	350
record 2	A-215	Mianus	700
record 3	A-101	Downtown	500
record 4	A-222	Redwood	700
record 5	A-201	Perryridge	900
record 6	A-217	Brighton	750
record 7	A-110	Downtown	600
record 8	A-218	Perryridge	700

27

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιούτας Σπύρος

Επιλογή: Δυαδική αναζήτηση



③ Μπορεί να εφαρμοστεί μόνο αν το αρχείο είναι διατεταγμένο με βάση το γνώρισμα της επιλογής.

- ③ Έστω όλες οι σελίδες μιας σχέσης αποθηκεύονται συνεχόμενες.
- ③ Το κόστος (# σελίδων που σαρώνονται) ισούται με:
 - ③ $\lceil \log_2(b_r) \rceil$
το κόστος εντοπισμού της πρώτης πλειάδας μέσω δυαδικής αναζήτησης στις σελίδες.
 - ③ + το πλήθος των σελίδων που περιέχουν εγγραφές που ικανοποιούν τη συνθήκη της επιλογής.

A-217	Brighton	750	
A-101	Downtown	500	
A-110	Downtown	600	
A-215	Mianus	700	
A-102	Perryridge	400	
A-201	Perryridge	900	
A-218	Perryridge	700	
A-222	Redwood	700	
A-305	Round Hill	350	

28

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιούτας Σπύρος

Επιλογή: Σάρωση ευρετηρίου

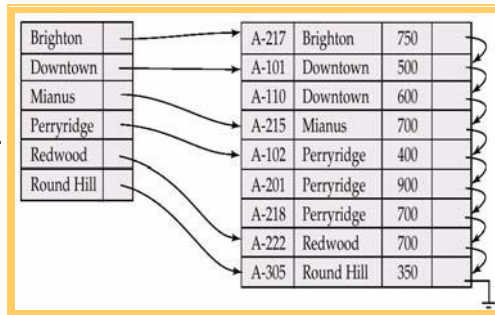


- ③ Η συνθήκη της επιλογής πρέπει να είναι πάνω στο κλειδί αναζήτησης του ευρετηρίου.
 - ③ C το κόστος εντοπισμού του κλειδικού αναζήτησης στο ευρετήριο
- ③ **Πρωτεύον ευρετήριο σε υποψήφιο κλειδί, ισότητα.** Ανάκτηση μια εγγραφής που ικανοποιεί τη συνθήκη ισότητας.

- ③ Κόστος = $C + 1$

- ③ **Πρωτεύον ευρετήριο σε απλό γνώρισμα (όχι κλειδί), ισότητα.**

- ③ Ανάκτηση πολλαπλών εγγραφών.
- ③ Οι εγγραφές θα βρίσκονται σε συνεχόμενες σελίδες
- ③ Κόστος = $C +$ το πλήθος των σελίδων που περιέχουν τις εγγραφές που ικανοποιούν τη συνθήκη ισότητας



29

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιούτας Σπύρος

Επιλογή: Σάρωση ευρετηρίου (συν.)

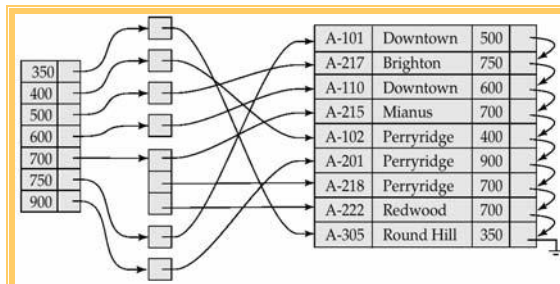


- ③ **Ισότητα στο κλειδί αναζήτησης του δευτερεύοντος ευρετηρίου.**

- ③ Ανάκτηση πολλαπλών εγγραφών
- ③ Κόστος = $C +$ το πλήθος των εγγραφών που ικανοποιούν τη συνθήκη αναζήτησης (μπορεί να είναι πολύ ακριβό!)

- ③ Κάθε εγγραφή μπορεί να βρίσκεται σε διαφορετική σελίδα

- ③ Για κάθε εγγραφή που ανακτάται απαιτείται μία πρόσβαση σε σελίδα



30

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιούτας Σπύρος

Επιλογή: συνθήκη με σύγκριση



- ③ Οι συνθήκες είναι της μορφής: $\sigma_{A \leq V}(r)$ ή $\sigma_{A \geq V}(r)$ και υλοποιούνται χρησιμοποιώντας:
 - ③ γραμμική ή δυαδική αναζήτηση στο αρχείο,
 - ③ ή ευρετήρια με τους ακόλουθους τρόπους:
- ③ **Πρωτεύον ευρετήριο, σύγκριση.** (Η σχέση είναι ταξινομημένη ως προς το γνώρισμα A)
 - ③ Για την επιλογή $\sigma_{A \geq V}(r)$
 - ③ χρήση του ευρετηρίου για την εύρεση της πρώτης πλειάδας με $A \geq V$
 - ③ σάρωση της σχέσης σειριακά ξεκινώντας από το σημείο αυτό
 - ③ Για την επιλογή $\sigma_{A \leq V}(r)$
 - ③ Σάρωση της σχέσης σειριακά μέχρι να βρεθεί η πρώτη πλειάδα με $A > V$. Δεν χρειάζεται να χρησιμοποιηθεί κάποιο ευρετήριο.

Επιλογή: συνθήκη με σύγκριση (συν.)



- ③ **Δευτερεύον ευρετήριο, σύγκριση.**
 - ③ Για την επιλογή $\sigma_{A \geq V}(r)$
 - ③ χρήση του ευρετηρίου για την εύρεση της πρώτης εγγραφής του ευρετηρίου με $A \geq V$
 - ③ σάρωση του ευρετηρίου σειριακά ξεκινώντας από το σημείο αυτό για να βρεθούν οι δείκτες στις εγγραφές
 - ③ Για την επιλογή $\sigma_{A \leq V}(r)$
 - ③ Σάρωση των φύλλων του ευρετηρίου για να βρεθούν οι δείκτες στις εγγραφές μέχρι να βρεθεί η πρώτη εγγραφή του ευρετηρίου με $A > V$.

Υλοποίηση σύνθετων επιλογών



- ③ **Σύζευξη (Conjunction):** $\sigma_{\theta_1} \wedge \theta_2 \wedge \dots \wedge \theta_n(r)$
- ③ **Συζευκτική επιλογή με χρήση απλού ευρετηρίου.**
 - ③ Επιλογή μιας διάταξης των θ_i και ενός από τους προηγούμενους αλγορίθμους που "προκαλούν" το μικρότερο κόστος για την επιλογή $\sigma_{\theta_i}(r)$.
 - ③ Για κάθε μία από τις πλειάδες του προηγούμενου βήματος, έλεγχος εάν ισχύουν οι υπόλοιπες συνθήκες.
- ③ **Συζευκτική επιλογή με χρήση σύνθετου ευρετηρίου.**
 - ③ Χρήση κατάλληλου σύνθετου ευρετηρίου, αν υπάρχει (συνδυασμός δύο ή περισσότερων γνωρισμάτων).

Υλοποίηση σύνθετων επιλογών (συν.)



- ③ **Συζευκτική επιλογή με τομή δεικτών.**
 - ③ Απαιτείται η ύπαρξη ευρετηρίων με δείκτες.
 - ③ Για κάθε συνθήκη χρήση του αντίστοιχου ευρετηρίου, και εύρεση της τομής των συνόλων των εγγραφών δεικτών.
 - ③ Στη συνέχεια ανάκτηση των εγγραφών από το αρχείο.
 - ③ Αν κάποιες συνθήκες δεν έχουν κατάλληλα ευρετήρια, εκ των υστέρων έλεγχος στη μνήμη.

Η λειτουργία της σύνδεσης



- ③ Υπάρχουν πολλοί διαφορετικοί αλγόριθμοι για την υλοποίηση της λειτουργίας της σύνδεσης
 1. **Σύνδεση με εμφώλευση βρόχων** (Nested-loop join). Παραλλαγές:
 - a. **Σύνδεση με εμφώλευση βρόχων κατά block** (Block-nested-loop join)
 - b. **Σύνδεση με εμφώλευση βρόχων μέσω ευρετηρίου** (Indexed-nested-loop join)
 2. **Σύνδεση με ταξινόμηση και συγχώνευση** (Sort-Merge join)
 3. **Σύνδεση με κατακερματισμό** (Hash join)
- ③ Η επιλογή του αλγορίθμου που θα χρησιμοποιήσουμε εξαρτάται από το κόστος του.

Nested-Loop Join



Ψευδοκώδικας για τον υπολογισμό του theta join $r \bowtie_{\theta} s$:

```
για κάθε πλειάδα  $t_x$  της  $r$  {  
  για κάθε πλειάδα  $t_s$  της  $s$  {  
    εξέτασε αν ο συνδυασμός  $(t_x, t_s)$  ικανοποιεί τη  
    συνθήκη  $\theta$   
    αν ναι, πρόσθεσε το  $t_x \cdot t_s$  στο αποτέλεσμα  
  }  
}
```

- ③ Η r ονομάζεται **εξωτερική σχέση** (outer relation) και η s **εσωτερική σχέση** (inner relation) της συνένωσης.
- ③ Δεν απαιτείται η ύπαρξη κάποιου ευρετηρίου.
- ③ Μπορεί να χρησιμοποιηθεί για οποιαδήποτε συνθήκη θ .
- ③ Είναι ακριβός, αφού εξετάζει κάθε ζεύγος πλειάδων των δύο σχέσεων.

Nested-Loop Join (συν.)



- ③ Στη χειρότερη περίπτωση, όταν η διαθέσιμη μνήμη αρκεί μόνο για ένα block από κάθε σχέση, το κόστος ισούται με:

$$n_r * b_s + b_r \text{ προσβάσεις στο δίσκο}$$

- ③ Αν η μικρότερη σχέση χωράει εξ' ολοκλήρου στη μνήμη, θα πρέπει να χρησιμοποιείται αυτή ως εσωτερική σχέση. Σε τέτοια περίπτωση το κόστος μειώνεται σε:

$$b_r + b_s \text{ προσβάσεις στο δίσκο.}$$

- ③ Ο block nested-loop αλγόριθμος (βλ. επόμενη διαφάνεια) είναι προτιμότερος από τον nested-loop join αλγόριθμο.

Block-Nested-Loop Join



- ③ Παραλλαγή του nested-loop join όπου κάθε block της εσωτερικής σχέσης συνδυάζεται με όλα τα blocks της εξωτερικής σχέσης.
- ③ Ψευδοκώδικας:

```
Για κάθε block  $B_r$  της  $r$  {
  Για κάθε block  $B_s$  της  $s$  {
    Για κάθε πλειάδα  $t_r$  του  $B_r$  {
      Για κάθε πλειάδα  $t_s$  του  $B_s$  {
        εξέτασε αν το ζεύγος  $(t_r, t_s)$  ικανοποιεί τη
        συνθήκη  $\theta$ 
        αν ναι, πρόσθεσε το  $t_r \cdot t_s$  στο
        αποτέλεσμα
      }
    }
  }
}
```

Block-Nested-Loop Join (συν.)



- ③ Κόστος χειρότερης περίπτωσης:

$$b_r * b_s + b_r \text{ προσβάσεις στο δίσκο.}$$

- ③ Κάθε block της εσωτερικής σχέσης διαβάζεται από μία φορά για κάθε block της εξωτερικής σχέσης (αντί να διαβάζεται από μια φορά για κάθε πλειάδα της εξωτερικής σχέσης όπως στον nested-loop join αλγόριθμο)

- ③ Κόστος καλύτερης περίπτωσης:

$$b_r + b_s \text{ προσβάσεις στο δίσκο.}$$

Indexed-Nested-Loop Join



- ③ Αντί να κάνουμε αναζήτηση στο αρχείο μπορούμε να κάνουμε αναζήτηση στο ευρετήριο αν
 - ③ Η σύνδεση είναι ισότητας (equi-join) ή φυσική (natural join)
 - ③ Υπάρχει ευρετήριο στο γνώρισμα σύνδεσης της εσωτερικής σχέσης
- ③ Για κάθε πλειάδα t_r της εξωτερικής σχέσης r , χρησιμοποιούμε το ευρετήριο (που είναι χτισμένο πάνω στην εσωτερική σχέση s) για να βρούμε τις πλειάδες της s που ικανοποιούν τη συνθήκη της σύνδεσης.

Indexed-Nested-Loop Join (συν.)

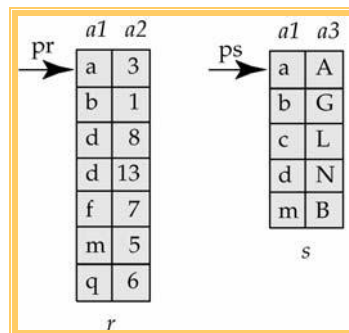


- ③ Κόστος σύνδεσης: $b_r + n_r * c$ προσβάσεις στο δίσκο
 - ③ c : το κόστος για την αναζήτηση στο ευρετήριο και την προσκόμιση όλων των πλειάδων της εσωτερικής σχέσης s που ικανοποιούν τη συνθήκη.
 - ③ Το c μπορεί να υπολογιστεί ως το κόστος μιας απλής πράξης επιλογής στη σχέση s , όπου η επιλογή χρησιμοποιεί ως συνθήκη τη συνθήκη της σύνδεσης.
- ③ Αν υπάρχουν ευρετήρια πάνω στη συνθήκη της σύνδεσης και για τις δύο σχέσεις r, s , ως εξωτερική σχέση χρησιμοποιούμε τη σχέση με τις λιγότερες πλειάδες. (γιατί;)

Sort-Merge Join



- 1. Ταξινομούμε και τις δύο σχέσεις ως προς το γνώρισμα της σύνδεσης (αν δεν είναι ήδη ταξινομημένες ως προς αυτό).
- 2. Συγχωνεύουμε τις ταξινομημένες σχέσεις
 - ④ Το βήμα της συγχώνευσης μοιάζει με το βήμα της συγχώνευσης του αλγορίθμου sort-merge.
 - ④ Η κύρια διαφορά έγκειται στον διαφορετικό τρόπο αντιμετώπισης των διπλότυπων τιμών που εμφανίζονται στο γνώρισμα της σύνδεσης — πρέπει να ταιριάξει κάθε ζεύγος που έχει την ίδια τιμή στο γνώρισμα της σύνδεσης



Sort-Merge Join (συν.)



- ③ Μπορεί να χρησιμοποιηθεί μόνο στην περίπτωση της σύνδεσης ισότητας (equi-join) ή της φυσικής σύνδεσης (natural join).
- ③ Κάθε σελίδα χρειάζεται να διαβαστεί μία μόνο φορά (αν υποθέσουμε πως όλες οι πλειάδες για μια δοθείσα τιμή των γνωρισμάτων της σύνδεσης χωράνε στη μνήμη)
- ③ Έτσι το πλήθος των προσβάσεων στο δίσκο για τον αλγόριθμο merge-join είναι:

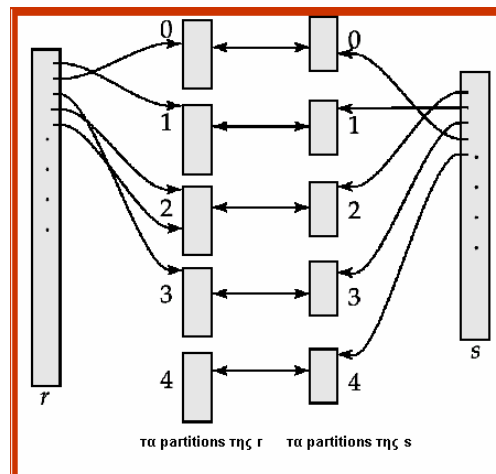
$$b_r + b_s + \text{το κόστος ταξινόμησης των σχέσεων}$$

(ο τελευταίος παράγοντας προκύπτει μόνο αν οι σχέσεις δεν είναι ήδη ταξινομημένες).

Hash Join



- ③ Εφαρμόζεται στην περίπτωση σύνδεσης ισότητας (equi-join) ή φυσικής σύνδεσης (natural join).
- ③ Μια συνάρτηση κατακερματισμού h χρησιμοποιείται για το διαμερισμό των πλειάδων των δύο σχέσεων σε τμήματα (partitions).



Hash Join (συν.)



- ③ Οι r πλειάδες του τμήματος r_i της σχέσης r χρειάζεται να συγκριθούν μόνο με τις s πλειάδες του τμήματος s_i της σχέσης s . Δεν χρειάζεται να γίνει σύγκριση με τις πλειάδες κάποιου άλλου τμήματος της σχέσης s επειδή:
 - ③ Μια πλειάδα της σχέσης r και μια πλειάδα της σχέσης s που ικανοποιούν τη συνθήκη της σύνδεσης θα έχουν την ίδια τιμή για τα γνωρίσματα της σύνδεσης.
 - ③ Αν η τιμή αυτή καταρκεματίζεται σε κάποια τιμή i , η πλειάδα της σχέσης r θα πρέπει να ανήκει στο τμήμα r_i και η πλειάδα της σχέσης s θα πρέπει να ανήκει στο τμήμα s_i .

Ο αλγόριθμος Hash Join



- Η σχέση s ονομάζεται **είσοδος κατασκευής των τμημάτων** (build input) ενώ η σχέση r ονομάζεται **είσοδος διερεύνησης της σύνδεσης** (probe input).
- Η σύνδεση μέσω κατακερματισμού των σχέσεων r και s υλοποιείται ως εξής:

1. Διαμοίρασε τη σχέση s χρησιμοποιώντας τη συνάρτηση κατακερματισμού h . Κατά τη διαμοίραση, μια σελίδα της μνήμης διατίθεται ως **buffer εξόδου για κάθε τμήμα της σχέσης s** .
2. Κάνε το ίδιο και για τη σχέση r .
3. ...

Ο αλγόριθμος Hash Join (συν.)



...

3. Για κάθε i :

- (α) Φόρτωσε το s_i στη μνήμη και "χτίσε" ένα ευρετήριο κατακερματισμού στη μνήμη χρησιμοποιώντας το γνώρισμα της σύνδεσης. Η συνάρτηση κατακερματισμού για το ευρετήριο κατακερματισμού στη μνήμη είναι διαφορετική από την συνάρτηση κατακερματισμού h που αναφέραμε παραπάνω.
- (β) Διάβασε από το δίσκο τις πλειάδες του τμήματος r_i μία προς μία. Για κάθε πλειάδα t_r βρες όλες τις ταιριαστές πλειάδες t_s της σχέσης s_i χρησιμοποιώντας το ευρετήριο κατακερματισμού της μνήμης. Η σύνδεση των γνωρισμάτων τους αποτελεί την έξοδο.

Το κόστος του αλγορίθμου Hash Join



- ③ Το κόστος του hash-join είναι:

$$3(b_r + b_s)$$

- ③ Το πλήθος των τμημάτων της σχέσης r ισούται με το πλήθος των τμημάτων της σχέσης s .
- ③ Το ίδιο ισχύει και για τον αριθμό των περασμάτων που απαιτούνται για τη διαμοίραση των σχέσεων r και s .

Τεχνικές σύνδεσης στα εμπορικά ΣΔΒΔ



	IBM DB2	IBM Informix	Microsoft SQL Server	Oracle	Sybase
Block-nested-loop join	√	√	√	√	√
Indexed-nested-loop join	√	√	√		√
Sort-Merge join			√	√	√
Hash join	√	√	√	√	√

Άλλες λειτουργίες: Προβολή, Συνάθροιση



- ③ Η **προβολή** υλοποιείται εφαρμόζοντας προβολή σε κάθε πλειάδα και απαλείφοντας στη συνέχεια τις διπλότυπες.
- ③ Η **απαλειφή των διπλότυπων τιμών** μπορεί να γίνει είτε μέσω κατακερματισμού είτε μέσω ταξινόμησης.
 - ③ Στην περίπτωση της **ταξινόμησης** οι διπλότυπες εγγραφές γειτνιάζουν, κα έτσι μπορούν να σβηστούν όλες πλην μίας.
 - ③ Η περίπτωση του **κατακερματισμού** είναι παρόμοια – οι διπλότυπες εγγραφές θα “μπουν” στο ίδιο τμήμα.
- ③ Η **συνάθροιση** μπορεί να υλοποιηθεί με τρόπο παρόμοιο με αυτόν της απαλειφής των διπλότυπων πλειάδων.

	IBM DB2	IBM Informix	Microsoft SQL Server	Oracle	Sybase
Ταξινόμηση	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Κατακερματισμός		<input type="checkbox"/>	<input type="checkbox"/>		

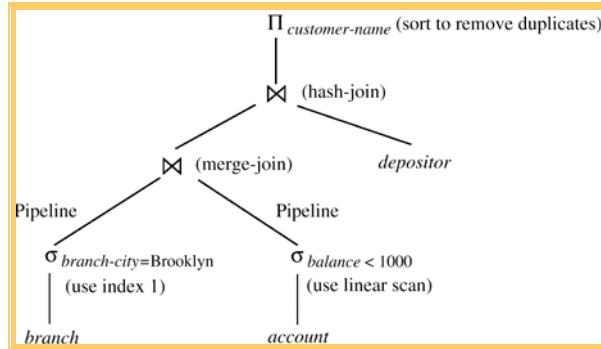
Σχέδιο εκτέλεσης



③ Ένα **σχέδιο εκτέλεσης ερώτησης** (query execution plan – QEP) καθορίζει:

(α) ποιος αλγόριθμος θα χρησιμοποιηθεί για κάθε επιμέρους λειτουργία και

(β) πώς θα συντονιστεί η εκτέλεση των λειτουργιών.



51

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιούτας Σπύρος

Επιλογή σχεδίου εκτέλεσης



- ③ Όταν επιλέγουμε ένα σχέδιο εκτέλεσης πρέπει να λαμβάνουμε υπόψη και την αλληλεπίδραση των επιμέρους λειτουργιών: η επιλογή του φθηνότερου αλγορίθμου για κάθε επιμέρους λειτουργία δεν οδηγεί κατ' ανάγκη και στο μικρότερο συνολικό κόστος. π.χ.
- ③ Ο αλγόριθμος Sort-Merge join μπορεί να είναι πιο ακριβός από τον αλγόριθμο Hash join, ωστόσο η έξοδος του είναι ταξινομημένη, γεγονός που μειώνει το κόστος μιας συνάθροισης σε επόμενη φάση.
- ③ Στην πράξη οι βελτιστοποιητές ερωτήσεων ενσωματώνουν στοιχεία των δύο ακόλουθων γενικών προσεγγίσεων:
1. **Βελτιστοποίηση βάσει κόστους** (cost-based optimization): Αναζήτηση όλων των σχεδίων εκτέλεσης και επιλογή του βέλτιστου με βάση το κόστος.
 2. **Ευριστική βελτιστοποίηση** (heuristic optimization): Χρήση ευριστικών αλγορίθμων για την επιλογή ενός σχεδίου εκτέλεσης.

52

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιούτας Σπύρος

Βελτιστοποίηση βάσει κόστους



- ③ Οι βελτιστοποιητές ερωτήσεων (query optimizers) χρησιμοποιούν συστηματικά τους κανόνες ισοδυναμίας προκειμένου να παράγουν τις ισοδύναμες εκφράσεις μιας δοθείσας έκφρασης.
- ③ Διαισθητικά, **παράγουν όλες τις ισοδύναμες εκφράσεις** εκτελώντας επαναληπτικά το ακόλουθο βήμα μέχρι να μη μπορούν να βρεθούν άλλες τέτοιες εκφράσεις:
 - ③ Για κάθε νέα έκφραση που εντοπίζεται, χρησιμοποιούνται όλοι οι δυνατοί κανόνες ισοδυναμίας και εντοπίζεται ένα σύνολο εκφράσεων το οποίο και προστίθεται στο αρχικό σύνολο εκφράσεων.
- ③ Η παραπάνω προσέγγιση είναι πολύ ακριβή τόσο ως προς το χώρο κα ως προς το χρόνο
- ③ **Οι απαιτήσεις χρόνου μπορούν να μειωθούν αν δεν δημιουργηθούν όλες οι εκφράσεις.**

53

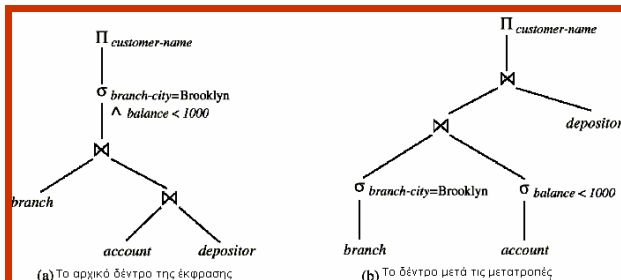
ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιούτας Σπύρος

Βελτιστοποίηση βάσει κόστους (συν.)



- ③ Έστω ότι θέλουμε να βρούμε τη βέλτιστη διάταξη για την έκφραση $r_1 \bowtie r_2 \dots \bowtie r_n$. Υπάρχουν $\frac{(2(n-1))!}{(n-1)!}$ διαφορετικές διατάξεις συνδέσεων.
- ③ Για $n = 7 \rightarrow 665280$, για $n = 10 \rightarrow 176$ δισεκατομμύρια!
- ③ **Δεν χρειάζεται να δημιουργήσουμε όλες τις διατάξεις.**

- ③ Με **δυναμικό προγραμματισμό**, υπολογίζεται μόνο μία φορά η πιο συμφέρουσα διάταξη για κάθε υποσύνολο των $\{r_1, r_2, \dots, r_n\}$ και αποθηκεύεται για μελλοντική χρήση.



Βελτιστοποίηση με ευριστικούς αλγορίθμους



- ③ Η βελτιστοποίηση βάσει κόστους είναι ακριβή, ακόμα και όταν υλοποιείται μέσω δυναμικού προγραμματισμού. Για το λόγο αυτό, πολλά συστήματα χρησιμοποιούν ευριστικούς αλγορίθμους.
- ③ Στη βελτιστοποίηση με ευριστικούς αλγορίθμους το δέντρο της ερώτησης μετασχηματίζεται μέσω ενός συνόλου κανόνων που τυπικά βελτιώνει την απόδοση της εκτέλεσης:
 - ③ **Εκτέλεση των επιλογών νωρίς** (έτσι μειώνεται το πλήθος των πλειάδων)
 - ③ **Εκτέλεση των προβολών νωρίς** (έτσι μειώνεται το πλήθος των γνωρισμάτων)
 - ③ **Εκτέλεση των πιο περιοριστικών επιλογών και συνδέσεων πριν από άλλες παρόμοιες λειτουργίες**
- ③ Άλλα συστήματα χρησιμοποιούν μόνο ευριστικούς αλγορίθμους, ενώ άλλα συνδυάζουν ευριστικούς με εν μέρει βελτιστοποίηση βάσει κόστους.

Ένας ευριστικός βελτιστοποιητής ερωτήσεων



- ③ Κανόνες
 - ③ Κανόνας 1: «εκτέλεσε τις επιλογές όσο το δυνατό νωρίτερα»
 - ③ Κανόνας 2: «αντικατέστησε τα καρτεσιανά γινόμενα με συνδέσεις όπου είναι δυνατό»
 - ③ Κανόνας 3: «αν υπάρχουν αρκετές συνδέσεις, εκτέλεσε πρώτα τις πιο περιοριστικές»
 - ③ Κανόνας 4: «με προβολή, απομάκρυνε τα άχρηστα γνωρίσματα όσο το δυνατό νωρίτερα»
- ③ Όλοι οι κανόνες έχουν στόχο την δημιουργία μικρών ενδιάμεσων σχέσεων

Ένας ευριστικός βελτιστοποιητής ερωτήσεων (συν.)



③ Μεθοδολογία

- ③ Μετασχηματισμός SQL ερώτησης στην ("τετριμμένη") ισοδύναμη έκφραση Σχεσιακής Άλγεβρας
- ③ Εφαρμογή αλγεβρικών μετασχηματισμών (Κανόνες 1, 2, 3, 4) → εναλλακτικά πλάνα εκτέλεσης
- ③ Επιλογή του 'βέλτιστου πλάνου' και εκτέλεσή του

③ Π.χ. δίδεται η SQL:

```
SELECT DISTINCT D.customer-name
FROM   account A, depositor D
WHERE  A.account-number = D.account-number
AND    balance < 2500
```

... που ισοδυναμεί με την ακόλουθη έκφραση σχεσιακής άλγεβρας:

$$\Pi_{\text{customer-name}}(\sigma_{\text{balance} < 2500 \wedge \text{A.account-number} = \text{D.account-number}} (A \times D))$$

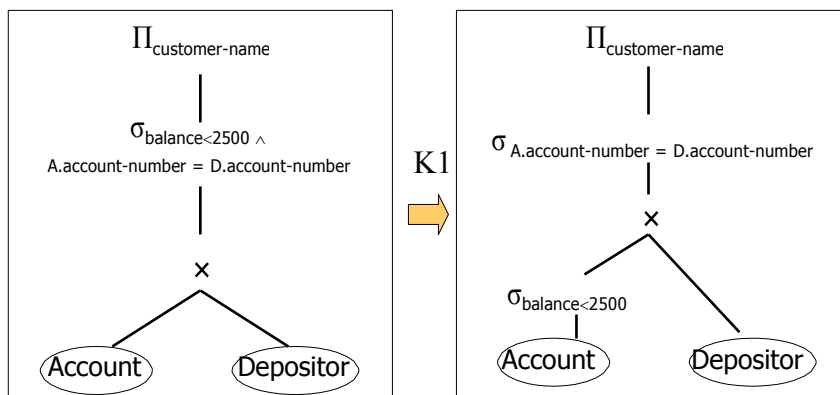
57

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιώτας Σπύρος

Παράδειγμα ευριστικής βελτιστοποίησης



«εκτέλεσε τις επιλογές όσο το δυνατό νωρίτερα»



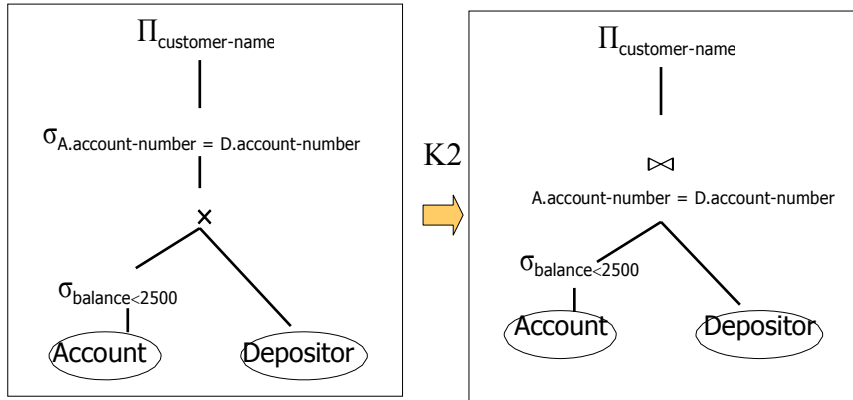
58

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιώτας Σπύρος

Παράδειγμα ευριστικής βελτιστοποίησης (συν.)



«αντικατέστησε τα καρτεσιανά γινόμενα με συνδέσεις, όπου είναι δυνατό»



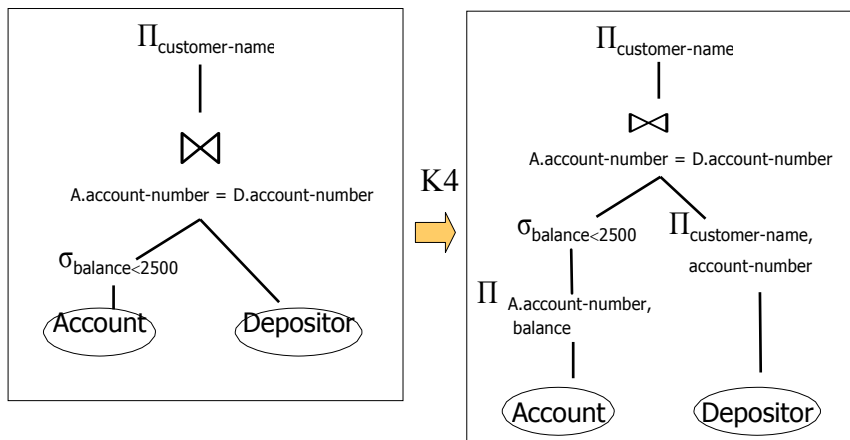
59

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιούρας Σπύρος

Παράδειγμα ευριστικής βελτιστοποίησης (συν.)



«με προβολή, απομάκρυνε τα άχρηστα γνωρίσματα όσο το δυνατό νωρίτερα»



60

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ. – Σιούρας Σπύρος

Η βελτιστοποίηση στα εμπορικά ΣΔΒΔ



	IBM DB2	IBM Informix	Microsoft SQL Server	Oracle	Sybase
Τεχνικές Βελτιστοποίησης	?	?	?	?	?

- ③ Όλα τα εμπορικά ΣΔΒΔ κρατάνε ως «επτασφράγιστα μυστικά» τις λεπτομέρειες κατασκευής του βελτιστοποιητή ερωτήσεων που έχουν υλοποιήσει. Πάντως:
 - ③ Όλα χρησιμοποιούν ιστογράμματα για την εκτίμηση του μεγέθους του αποτελέσματος
 - ③ Όλα ευνοούν τη χρήση ευρετηρίων σε πλάνα εκτέλεσης
 - ③ Όλα αναζητούν δέντρα αριστερού βάθους χρησιμοποιώντας δυναμικό προγραμματισμό
- ③ Η ανάπτυξη ενός τέτοιου λογισμικού αντιστοιχεί σε εργασία δεκάδων ανθρωποετών.



Τέλος ενότητας!