

Επεξεργασία & Βελτιστοποίηση Ερωτημάτων



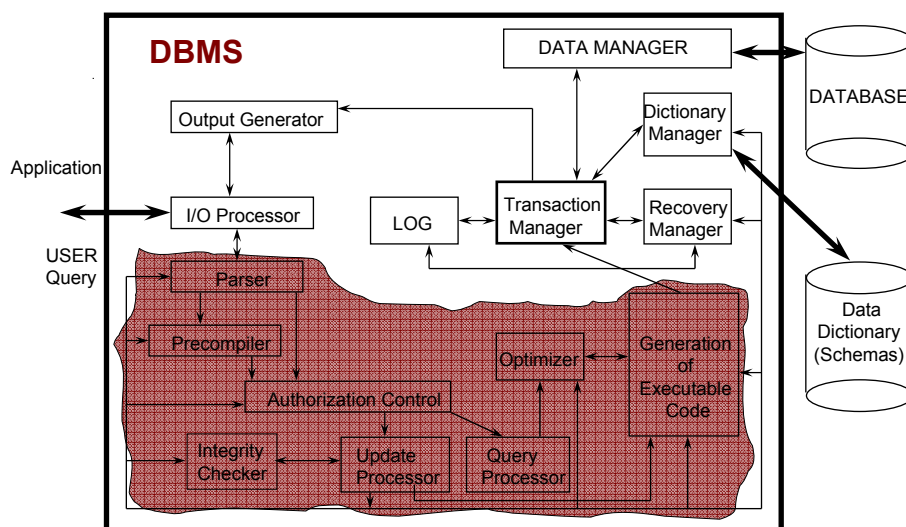
- Ο βελτιστοποιητής ερωτημάτων (query optimizer)
- Μετασχηματισμός εκφράσεων σχεσιακής άλγεβρας
- Υπολογισμός μεγεθών πράξεων σχεσιακής άλγεβρας
 - επιλογή, σύνδεση, άλλες πράξεις (προβολή κλπ.)
- Τρόποι εκτέλεσης (και κόστος) πράξεων σχεσιακής άλγεβρας
- Αποτίμηση σύνθετων εκφράσεων (materialization, pipelining)

Βασική πηγή διαφανειών: Silberschatz et al., "Database System Concepts", 4/e
Εργαστήριο Πληροφοριακών Συστημάτων, Πανεπιστήμιο Πειραιώς (<http://infolab.cs.unipi.gr/>)
έκδοση: Φεβ. 2010

ΒΔ: [6] Επεξεργασία & Βελτιστοποίηση Ερωτημάτων

ΠΑ.ΠΕΙ. – Γιάννης Θεοδωρίδης

Αρχιτεκτονική ενός ΣΔΒΔ



ΒΔ: [6] Επεξεργασία & Βελτιστοποίηση Ερωτημάτων

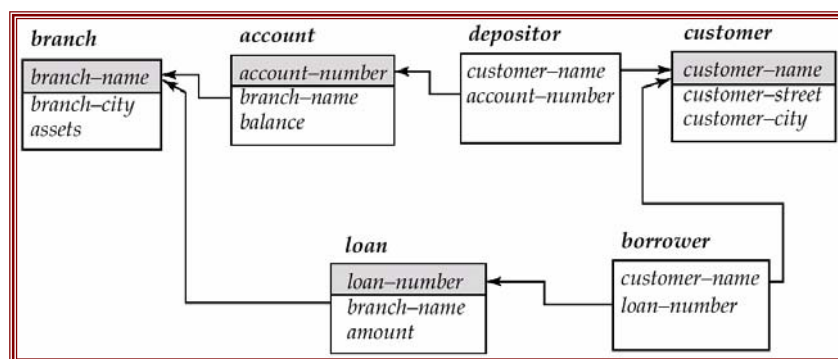
2

ΠΑ.ΠΕΙ. – Γιάννης Θεοδωρίδης

Διάγραμμα Σχεσιακού σχήματος



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



ΒΔ: [6] Επεξεργασία & Βελτιστοποίηση Ερωτημάτων

3

ΠΑ.ΠΕΙ. – Γιάννης Θεοδωρίδης



- ➔
- Ο βελτιστοποιητής ερωτημάτων (query optimizer)
 - Μετασχηματισμός εκφράσεων σχεσιακής άλγεβρας
 - Υπολογισμός μεγεθών πράξεων σχεσιακής άλγεβρας
 - επιλογή, σύνδεση, άλλες πράξεις (προβολή κλπ.)
 - Τρόποι εκτέλεσης (και κόστος) πράξεων σχεσιακής άλγεβρας
 - Αποτίμηση σύνθετων εκφράσεων (materialization, pipelining)

ΒΔ: [6] Επεξεργασία & Βελτιστοποίηση Ερωτημάτων

ΠΑ.ΠΕΙ. – Γιάννης Θεοδωρίδης

Επεξεργασία ερωτημάτων (1)

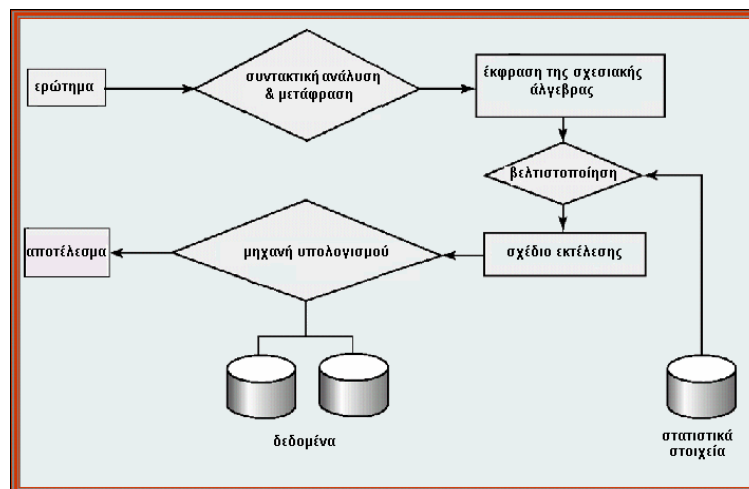


- Όταν ο client υποβάλει ένα ερώτημα στο server (συνήθως σε δηλωτική γλώσσα, π.χ. SQL), ο server εκτελεί τα εξής βήματα:
 1. **Συντακτική ανάλυση** (parsing) και **μετάφραση** (translation)
 - Μετατροπή σε έκφραση σχεσιακής άλγεβρας
 2. **Βελτιστοποίηση** (optimization)
 - Σύνθεση εναλλακτικών πλάνων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

Επεξεργασία ερωτημάτων (2)



- ή πιο απλά ...



Επεξεργασία ερωτημάτων (3)



■ Συντακτική ανάλυση και Μετάφραση

- Το ερώτημα SQL

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

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

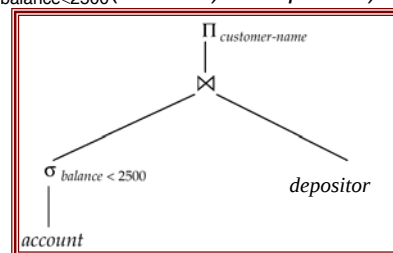
$$\Pi_{\text{customer-name}}(\sigma_{\text{balance} < 2500}(\text{account} \bowtie \text{depositor}))$$

■ Βελτιστοποίηση ...

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

■ Εκτέλεση πλάνου και επιστροφή αποτελέσματος

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



Βελτιστοποίηση ερωτημάτων

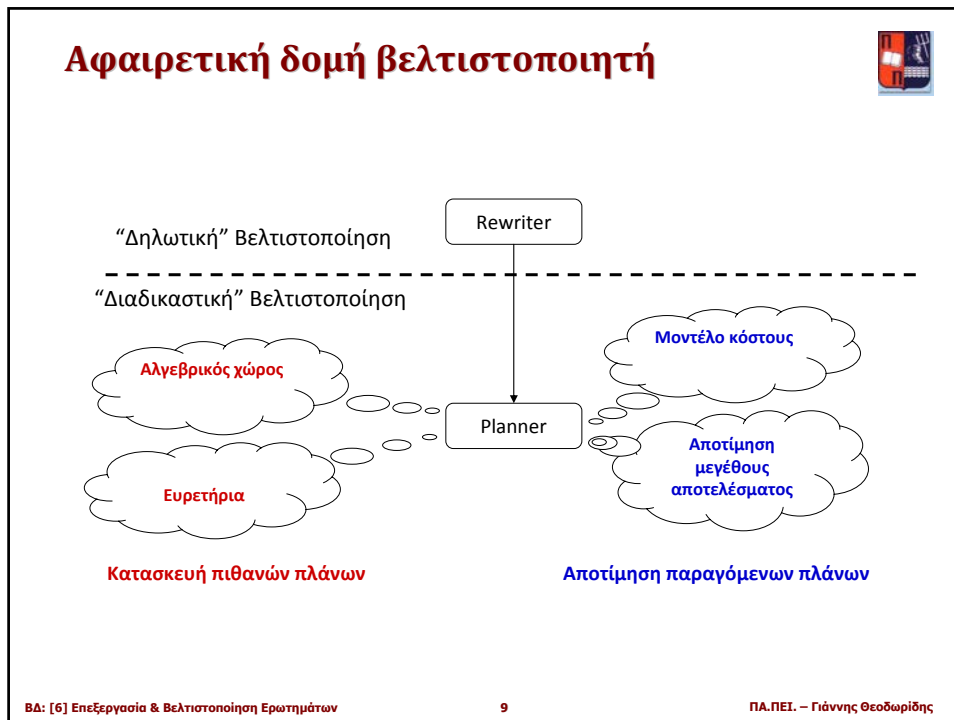


- Μια έκφραση της σχεσιακής άλγεβρας ενδέχεται να έχει πολλές ισοδύναμες εκφράσεις

- Π.χ. η έκφραση $\sigma_{\text{balance} < 2500}(\Pi_{\text{balance}}(\text{account}))$

ισοδυναμεί με την έκφραση $\Pi_{\text{balance}}(\sigma_{\text{balance} < 2500}(\text{account}))$

- Κάθε πράξη της σχεσιακής άλγεβρας μπορεί να υλοποιηθεί με χρήση κάποιου από πολλούς διαφορετικούς αλγόριθμους
- Άρα, δεν αρκεί μόνο ο προσδιορισμός της πράξης, αλλά χρειάζεται να προσδιοριστεί και ο αλγόριθμος που θα χρησιμοποιηθεί για την υλοποίησή της.
- Ο καθορισμός μιας λεπτομερούς στρατηγικής για τον υπολογισμό μιας πράξης ονομάζεται **σχέδιο εκτέλεσης** (execution plan).



1^ο στάδιο - Rewriter

- Υπάρχει ένα επίπεδο «**δηλωτικής**» **βελτιστοποίησης**, ή **επανεγγραφής**, όπου παράγουμε λογικά ισοδύναμους τρόπους να εκφράσουμε ένα ερώτημα μέσω του **rewriter**
- Αυτό συμπεριλαμβάνει, συνήθως:
 - Μετατροπή εκφράσεων σε «βολική» μορφή (χρήση conjunctive normal forms – CNF)
 - Απλοποίηση εμφωλευμένων ερωτημάτων
 - Σημασιολογικά έξυπνες μετατροπές
- Παραδείγματα:
 - ... WHERE NOT(Attr₁ = 10 OR Attr₂ > 3) γίνεται
... WHERE Attr₁ <> 10 AND Attr₂ <= 3
 - ... WHERE Attr = YEAR('1994-09-08') γίνεται
... WHERE Attr = 1994
 - δοθέντος: $R_1.A_1=R_2.A_2$ AND $R_2.A_2=R_3.A_3$ AND $R_1.A_1>5$
προστίθενται... $R_1.A_1=R_3.A_3$ AND $R_2.A_2>5$ AND $R_3.A_3>5$

ΒΔ: [6] Επεξεργασία & Βελτιστοποίηση Ερωτημάτων

10

ΠΑ.ΠΕΙ. – Γιάννης Θεοδωρίδης

2^ο στάδιο – Planner (1)



- Υπάρχει ένα επίπεδο **«διαδικαστικής» βελτιστοποίησης**, όπου παράγουμε (όλα;) τα διαφορετικά πλάνα εκτέλεσης μέχρι να διαλέξουμε το πιο αποδοτικό. Η εργασία αυτή ανατίθεται στον **planner**.
- Ο planner οφείλει:
 - Να **κατασκευάσει ένα σύνολο πλάνων** και
 - Να **αποτιμήσει κάθε πλάνο** που παράγει, ώστε να βρει το πιο αποδοτικό
 - Προσοχή: η αποτίμηση είναι πάντα προσέγγιση / πρόβλεψη και όχι ακριβής υπολογισμός...

2^ο στάδιο – Planner (2)



- **Κατασκευή πλάνων**, με βάση
 - Ένα αλγεβρικό χώρο για τη σειρά εκτέλεσης των λειτουργιών (π.χ., να αποφασίσει με ποια σειρά θα κάνει το $R \bowtie S \bowtie T$)
 - Ένα σύνολο από μεθόδους προσπέλασης στα δεδομένα (π.χ., full-index scan, full table scan, ...)
- **Αποτίμηση κάθε πλάνου**, με βάση
 - Ένα μοντέλο κόστους που προβλέπει το κόστος εκτέλεσης κάθε πλάνου (σε χρόνο, disk I/O, ...)
 - Ένα μοντέλο πρόβλεψης του μεγέθους, κυρίως των ενδιάμεσων αποτελεσμάτων
 - Το κόστος υπολογίζεται με χρήση στατιστικών πληροφοριών που προέρχονται από τον κατάλογο της βάσης
 - πλήθος πλειάδων κάθε σχέσης, μέγεθος πλειάδων, κλπ.



- Ο βελτιστοποιητής ερωτημάτων (query optimizer)
- ➔ ▪ Μετασχηματισμός εκφράσεων σχεσιακής άλγεβρας
- Υπολογισμός μεγεθών πράξεων σχεσιακής άλγεβρας
 - επιλογή, σύνδεση, άλλες πράξεις (προβολή κλπ.)
- Τρόποι εκτέλεσης (και κόστος) πράξεων σχεσιακής άλγεβρας
- Αποτίμηση σύνθετων εκφράσεων (materialization, pipelining)

Μετασχηματισμός εκφράσεων ΣΑ



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

Κανόνες Ισοδυναμίας (1)



1. Οι συζευκτικές επιλογές μπορούν να αναλυθούν σε μια ακολουθία ξεχωριστών επιλογών.

$$\sigma_{\theta_1 \wedge \theta_2}(R) = \sigma_{\theta_1}(\sigma_{\theta_2}(R))$$

2. Οι επιλογές είναι αντιμεταθετικές.

$$\sigma_{\theta_1}(\sigma_{\theta_2}(R)) = \sigma_{\theta_2}(\sigma_{\theta_1}(R))$$

3. Σε μια ακολουθία από προβολές μόνο η τελευταία προβολή χρειάζεται, οι υπόλοιπες μπορούν να παραληφθούν.

$$\Pi_{t_1}(\Pi_{t_2}(\dots(\Pi_{t_m}(R))\dots)) = \Pi_{t_1}(R)$$

4. Οι επιλογές μπορούν να συνδυαστούν με Καρτεσιανά Γινόμενα και Θ-συνδέσεις.

$$\sigma_{\theta}(R \bowtie S) = R \bowtie_{\theta} S$$

$$\sigma_{\theta_1}(R \bowtie_{\theta_2} S) = R \bowtie_{\theta_1 \wedge \theta_2} S$$

Κανόνες Ισοδυναμίας (2)



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

$$R \bowtie_{\theta} S = S \bowtie_{\theta} R$$

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

$$(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$$

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

$$(R \bowtie_{\theta_1} S) \bowtie_{\theta_2 \wedge \theta_3} T = R \bowtie_{\theta_1 \wedge \theta_3} (S \bowtie_{\theta_2} T)$$

όπου η συνθήκη θ_2 περιλαμβάνει μόνο χαρακτηριστικά των σχέσεων S και T .

Κανόνες Ισοδυναμίας (3)



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

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

$$\sigma_{\theta_1}(R \bowtie_{\theta} S) = (\sigma_{\theta_1}(R)) \bowtie_{\theta} S$$

(β) όταν όλα τα χαρακτηριστικά της συνθήκης θ_1 προέρχονται μόνο από τα χαρακτηριστικά της R ενώ τα χαρακτηριστικά της θ_2 προέρχονται μόνο από τα χαρακτηριστικά της S .

$$\sigma_{\theta_1 \wedge \theta_2}(R \bowtie_{\theta} S) = (\sigma_{\theta_1}(R)) \bowtie_{\theta} (\sigma_{\theta_2}(S))$$

Κανόνες Ισοδυναμίας (4)



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

$$\Pi_{L_1 \cup L_2}(R_1 \bowtie_{\theta} R_2) = (\Pi_{L_1}(R_1)) \bowtie_{\theta} (\Pi_{L_2}(R_2))$$

(β) Ας θεωρήσουμε μια σύνδεση $R_1 \bowtie_{\theta} R_2$.

- Έστω L_1 και L_2 τα σύνολα των χαρακτηριστικών των σχέσεων R_1 και R_2 , αντίστοιχα.
- Έστω L_3 το σύνολο των χαρακτηριστικών της σχέσης R_1 που συμμετέχουν στη συνθήκη θ , αλλά δεν ανήκουν στο σύνολο $L_1 \cup L_2$.
- Έστω L_4 το σύνολο των χαρακτηριστικών της σχέσης R_2 που συμμετέχουν στη συνθήκη θ , αλλά δεν ανήκουν στο σύνολο $L_1 \cup L_2$.

$$\Pi_{L_1 \cup L_2}(R_1 \bowtie_{\theta} R_2) = \Pi_{L_1 \cup L_2}((\Pi_{L_1 \cup L_3}(R_1)) \bowtie_{\theta} (\Pi_{L_2 \cup L_4}(R_2)))$$

Κανόνες Ισοδυναμίας (5)



9. Η ένωση και η τομή είναι αντιμεταθετικές

$$R \cup S = S \cup R$$

$$R \cap S = S \cap R$$

Όμως: η διαφορά δεν είναι αντιμεταθετική

10. Η ένωση και η τομή είναι προσεταιριστικές.

$$(R \cup S) \cup T = R \cup (S \cup T)$$

$$(R \cap S) \cap T = R \cap (S \cap T)$$

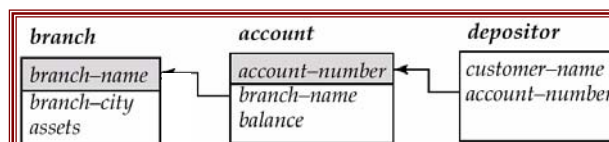
11. Η επιλογή επιμερίζεται με τις λειτουργίες $\cup$, $\cap$ και $-$.

$$\sigma_\theta (R - S) = \sigma_\theta (R) - \sigma_\theta (S)$$

ομοίως στην περίπτωση των τελεστών $\cup$ και $\cap$

Επίσης: $\sigma_\theta (R - S) = \sigma_\theta (R) - S$,
αν η συνθήκη θ αφορά μόνο σε χαρακτηριστικά της R
ομοίως στην περίπτωση των τελεστών $\cup$ και $\cap$

Παράδειγμα μετασχηματισμού



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

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

- Μετατροπή μέσω του κανόνα 6^α:

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

...

Παράδειγμα μετασχηματισμού (2)



■ ...

■ Μετατροπή μέσω του κανόνα 7^α:

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

■ Μετατροπή μέσω του κανόνα 7^β:

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


■ Ο βελτιστοποιητής ερωτημάτων (query optimizer)

■ Μετασχηματισμός εκφράσεων σχεσιακής άλγεβρας



■ Υπολογισμός μεγεθών πράξεων σχεσιακής άλγεβρας

■ επιλογή, σύνδεση, άλλες πράξεις (προβολή κλπ.)

■ Τρόποι εκτέλεσης (και κόστος) πράξεων σχεσιακής άλγεβρας

■ Αποτίμηση σύνθετων εκφράσεων (materialization, pipelining)

Μέτρηση κόστους ερωτημάτων (1)



- Το **κόστος ενός ερωτήματος** ισούται με το συνολικό χρόνο που απαιτείται για την απάντησή του
 - Πολλοί παράγοντες επηρεάζουν το κόστος: οι προσπελάσεις στο δίσκο, η ισχύς της CPU, ακόμα και ο φόρτος του δικτύου
- Οι προσπελάσεις στο δίσκο, που αποτελούν το βασικό κόστος υπολογισμού ενός ερωτήματος, είναι εύκολο να υπολογιστούν. Η εκτίμηση γίνεται λαμβάνοντας υπόψη:
 - Το μέγεθος (σε εγγραφές) της απάντησης ενός ερωτήματος
 - Το πλήθος των σελίδων δίσκου που διαβάζονται / γράφονται για να απαντηθεί ένα ερώτημα

Μέτρηση κόστους ερωτημάτων (2)



- Για απλότητα χρησιμοποιούμε ως μέτρο του κόστους ενός ερωτήματος **το πλήθος των σελίδων (blocks) που μεταφέρονται από το δίσκο** (κόστος ανάγνωσης δεδομένων)
 - Για λόγους απλότητας δεν λαμβάνουμε υπόψη το κόστος της CPU
 - Ούτε λαμβάνουμε υπόψη το κόστος εγγραφής των δεδομένων στο δίσκο
- Το κόστος εξαρτάται από το μέγεθος του **buffer της κύριας μνήμης** (της τάξης κάποιων Mb)
 - Η ύπαρξη επιπλέον μνήμης μειώνει την ανάγκη για προσπελάσεις στο δίσκο
 - 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

Παραδοχές (από Οργάνωση Αρχείων & Ευρετήρια)



- Η μονάδα μεταφοράς μεταξύ δίσκου και μνήμης είναι ένα block δίσκου (με μέγεθος b bytes)
- Κάθε αρχείο έχει εγγραφές ενός τύπου μόνο
- Το μέγεθος των εγγραφών είναι σταθερό (με μέγεθος r bytes)
- Οι εγγραφές δεν επιτρέπεται να διασχίζουν τα όρια ενός block

Παράγοντας σελιδοποίησης (blocking factor):

$$\text{bfr} = \lfloor (b / r) \rfloor$$

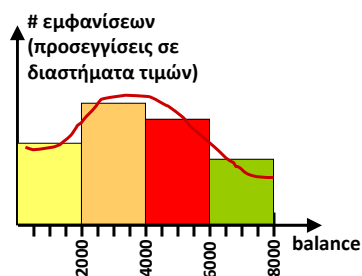
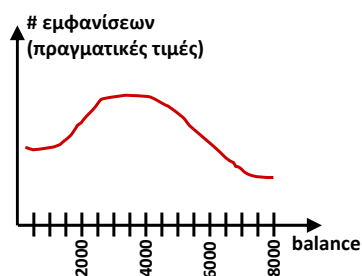
Αριθμός blocks για την αποθήκευση ενός αρχείου n εγγραφών:

$$B = \lceil (n / \text{bfr}) \rceil$$

Εκτίμηση μεγέθους - Ιστογράμματα (1)



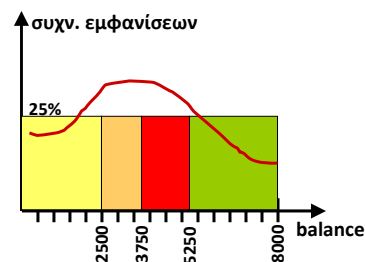
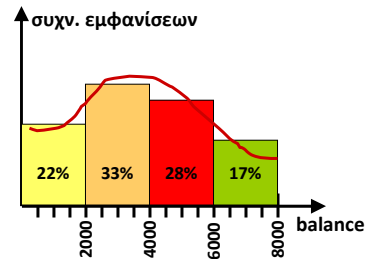
- Για να δουλέψουν οι συναρτήσεις κόστους, πρέπει να μπορούμε να αποτιμήσουμε το **μέγεθος των ενδιάμεσων αποτελεσμάτων**
- Η πιο καλή τεχνική που έχουμε είναι τα **ιστογράμματα**
 - Σ' ένα ιστόγραμμα, διαιρούμε το εύρος των τιμών ενός πεδίου σε **κάδους** (buckets)
 - Για κάθε τιμή που παίρνει το πεδίο,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



Εκτίμηση μεγέθους – Ιστογράμματα (2)



- Παραλλαγές:
 - **Ιστογράμματα ίσου πλάτους:** κάθε κάδος καλύπτει το ίδιο διάστημα τιμών
 - **Ιστογράμματα ίσου ύψους:** κάθε κάδος καλύπτει ίσο αριθμό (ή συχνότητα) εμφανίσεων
- (Λανθασμένες) Υποθέσεις εργασίας:
 - οι τιμές των πεδίων είναι ισοπίθανα μοιρασμένες
 - τα πεδία είναι ανεξάρτητα μεταξύ τους
- Ανάγκη περιοδικής ενημέρωσης των ιστογραμμάτων



ΒΔ: [6] Επεξεργασία & Βελτιστοποίηση Ερωτημάτων

27

ΠΑ.ΠΕΙ. – Γιάννης Θεοδωρίδης



- Ο βελτιστοποιητής ερωτημάτων (query optimizer)
- Μετασχηματισμός εκφράσεων σχεσιακής άλγεβρας
- Υπολογισμός μεγεθών πράξεων σχεσιακής άλγεβρας
 - επιλογή, σύνδεση, άλλες πράξεις (προβολή κλπ.)
- ➔ ■ Τρόποι εκτέλεσης (και κόστος) πράξεων σχεσιακής άλγεβρας
- Αποτίμηση σύνθετων εκφράσεων (materialization, pipelining)

ΒΔ: [6] Επεξεργασία & Βελτιστοποίηση Ερωτημάτων

ΠΑ.ΠΕΙ. – Γιάννης Θεοδωρίδης

Τρόποι εκτέλεσης πράξεων ΣΑ



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

Η πράξη Επιλογή



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

Επιλογή - Γραμμική αναζήτηση



- Σάρωση όλων των σελίδων του αρχείου και για κάθε εγγραφή μιας σελίδας έλεγχος αν ικανοποιεί τη συνθήκη της επιλογής.
 - Κόστος (# σελίδων δίσκου που σαρώνονται) = B_R
 - Αν η συνθήκη της επιλογής αναφέρεται σε κάποιο χαρακτηριστικό - κλειδί, τότε (μέσο) κόστος = $B_R/2$
 - 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

ΒΔ: [6] Επεξεργασία & Βελτιστοποίηση Ερωτημάτων

31

ΠΑ.ΠΕΙ. – Γιάννης Θεοδωρίδης

Επιλογή - Δυαδική αναζήτηση



- Μπορεί να εφαρμοστεί μόνο σε διατεταγμένα αρχεία.
 - Έστω όλες οι σελίδες μιας σχέσης αποθηκεύονται συνεχόμενες.
 - Το κόστος (# σελίδων που σαρώνονται) ισούται με:
 - $\lceil \log_2(B_R) \rceil$
το κόστος εντοπισμού της πρώτης πλειάδας μέσω δυαδικής αναζήτησης στις σελίδες.
 - + 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	

ΒΔ: [6] Επεξεργασία & Βελτιστοποίηση Ερωτημάτων

32

ΠΑ.ΠΕΙ. – Γιάννης Θεοδωρίδης

Επιλογή - Απευθείας αναζήτηση



- Μπορεί να εφαρμοστεί μόνο σε κατακερματισμένα αρχεία.
 - Το κόστος (# σελίδων που σαρώνονται) ισούται με:
 - 1 (δηλ. κόστος εντοπισμού μέσω συνάρτησης κατακερματισμού)
 - + το πλήθος των σελίδων που ενδεχομένως πρέπει να σαρωθούν λόγω συγκρούσεων.

bucket 0				bucket 5	A-102	Perryridge	400
					A-201	Perryridge	900
					A-218	Perryridge	700
bucket 1				bucket 6			
bucket 2				bucket 7	A-215	Mianus	700
bucket 3	A-217	Brighton	750	bucket 8	A-101	Downtown	500
	A-305	Round Hill	350		A-110	Downtown	600
bucket 4	A-222	Redwood	700	bucket 9			

ΒΔ: [6] Επεξεργασία & Βελτιστοποίηση Ερωτημάτων

33

ΠΑ.ΠΕΙ. – Γιάννης Θεοδωρίδης

Επιλογή: Σάρωση ευρετηρίου (1)



- Η συνθήκη της επιλογής πρέπει να είναι πάνω στο κλειδί διάταξης του ευρετηρίου.
 - C το κόστος εντοπισμού του κλειδιού διάταξης στο ευρετήριο (εντοπισμός με δυαδική αναζήτηση)
- Πρωτεύουν ευρετήριο σε κλειδί, συνθήκη ισότητας.** Ανάκτηση μιας εγγραφής που ικανοποιεί τη συνθήκη ισότητας.
 - $\text{Κόστος} = C + 1$
- Πρωτεύουν ευρετήριο σε απλό χαρακτηριστικό (όχι κλειδί), συνθήκη ισότητας.**
 - Ανάκτηση πολλαπλών εγγραφών που όμως βρίσκονται σε συνεχόμενες σελίδες
 - $\text{Κόστος} = C + \text{πλήθος σελίδων που περιέχουν τις εγγραφές που ικανοποιούν τη συνθήκη ισότητας}$

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

ΒΔ: [6] Επεξεργασία & Βελτιστοποίηση Ερωτημάτων

34

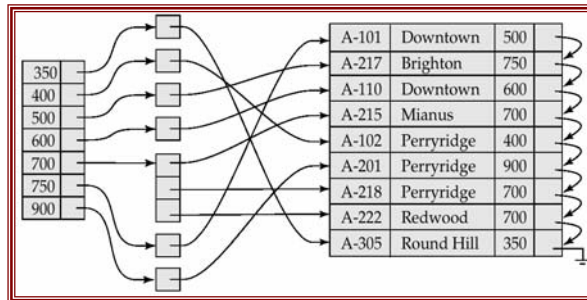
ΠΑ.ΠΕΙ. – Γιάννης Θεοδωρίδης

Επιλογή: Σάρωση ευρετηρίου (2)



- Δευτερεύον ευρετήριο σε απλό χαρακτηριστικό (όχι κλειδί), συνθήκη ισότητας.

- Ανάκτηση πολλαπλών εγγραφών
- Κόστος = C + το πλήθος των εγγραφών που ικανοποιούν τη συνθήκη αναζήτησης (μπορεί να είναι πολύ ακριβό!)
 - Κάθε εγγραφή μπορεί να βρίσκεται σε διαφορετική σελίδα
 - Για κάθε εγγραφή που ανακτάται απαιτείται μία πρόσβαση σε σελίδα



ΒΔ: [6] Επεξεργασία & Βελτιστοποίηση Ερωτημάτων

35

ΠΑ.ΠΕΙ. – Γιάννης Θεοδωρίδης

Επιλογή: συνθήκη με σύγκριση (1)



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

ΒΔ: [6] Επεξεργασία & Βελτιστοποίηση Ερωτημάτων

36

ΠΑ.ΠΕΙ. – Γιάννης Θεοδωρίδης

Επιλογή: συνθήκη με σύγκριση (2)



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

Υλοποίηση σύνθετων επιλογών (1)



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

Υλοποίηση σύνθετων επιλογών (2)



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

Η λειτουργία της σύνδεσης



- 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** (*Index-nested-loop join*)
 2. **Σύνδεση με ταξινόμηση και συγχώνευση** (*Sort-Merge join*)
 3. **Σύνδεση με κατακερματισμό** (*Hash join*)

Nested-Loop Join



Ψευδοκώδικας για τον υπολογισμό της Θ -σύνδεσης $R \bowtie S$:

Nested-Loop Join

```

1 FOR each tuple r in R
2   FOR each tuple s in S
3     IF r[A]=s[B] THEN output (r,s)
  
```

- Η R ονομάζεται **εξωτερική σχέση** (*outer relation*) και η S **εσωτερική σχέση** (*inner relation*) της συνένωσης.
- Δεν απαιτείται η ύπαρξη ευρετηρίου.
- Κοστίζει αρκετά, αφού εξετάζει όλα τα ζεύγη πλειάδων μεταξύ των δύο σχέσεων.

Block-Nested-Loop Join (1)



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

Block-Nested-Loop Join

```

1 FOR each block br in R
2   FOR each block bs in S
3     FOR each tuple r in br
4       FOR each tuple s in bs
5         IF r[A]=s[B] THEN output (r,s)
  
```

Block-Nested-Loop Join (2)



- Κόστος σύνδεσης:

$$B_R * B_S + B_R \text{ προσβάσεις στο δίσκο.}$$

- 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)

Index-Nested-Loop Join (1)



- Αντί να κάνουμε αναζήτηση στο αρχείο μπορούμε να κάνουμε αναζήτηση στο ευρετήριο αν
 - Η σύνδεση είναι ισοσύνδεση (*equi-join*) ή φυσική σύνδεση (*natural join*)
 - Υπάρχει ευρετήριο στο χαρακτηριστικό σύνδεσης της εσωτερικής σχέσης
- Για κάθε πλειάδα 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.

Index-Nested-Loop Join (2)

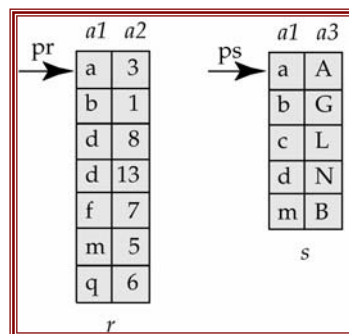


- Κόστος σύνδεσης: $B_R + N_R * C$ προσβάσεις στο δίσκο
 - C : το κόστος για την αναζήτηση στο ευρετήριο και την προσκόμιση όλων των πλειάδων της εσωτερικής σχέσης S που ικανοποιούν τη συνθήκη.
 - Το C μπορεί να υπολογιστεί ως το κόστος μιας απλής πράξης επιλογής στη σχέση S , όπου η επιλογή χρησιμοποιεί ως συνθήκη τη συνθήκη της σύνδεσης.
- 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)



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



Sort-Merge Join (2)



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

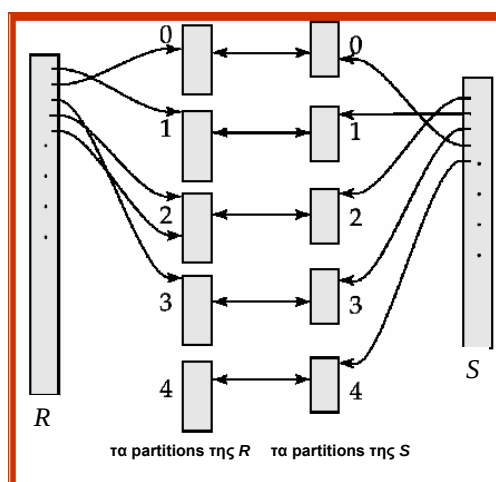
$$B_R + B_S + \text{το κόστος ταξινόμησης των σχέσεων}$$

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

Hash Join (1)



- Εφαρμόζεται μόνο στην περίπτωση ισοσύνδεσης ή φυσικής σύνδεσης.
- Μια συνάρτηση κατακερματισμού h χρησιμοποιείται για το διαμερισμό των πλειάδων των δύο σχέσεων σε τμήματα (*partitions*).



Hash Join (2)



- 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:
 - Μια πλειάδα της σχέσης R και μια πλειάδα της σχέσης S που ικανοποιούν τη συνθήκη της σύνδεσης θα έχουν την ίδια τιμή για τα χαρακτηριστικά της σύνδεσης.
 - Αν η τιμή αυτή κατακερματίζεται 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 .

Hash Join (3)



- Η σχέση S ονομάζεται **είσοδος κατασκευής των τμημάτων** (*build input*) ενώ η σχέση R ονομάζεται **είσοδος διερεύνησης της σύνδεσης** (*probe input*).
- Ψευδοκώδικας (μια απλή εκδοχή):

```

1 Build hash file on S(A)
2 FOR each tuple r in R
3   Access relevant bucket b on A
4   FOR each tuple s in S stored in b
5     IF r[A]=s[B] THEN output (r,s)
  
```

Άλλες λειτουργίες: Προβολή, Συνάθροιση



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



- Ο βελτιστοποιητής ερωτημάτων (query optimizer)
- Μετασχηματισμός εκφράσεων σχεσιακής άλγεβρας
- Υπολογισμός μεγεθών πράξεων σχεσιακής άλγεβρας
 - επιλογή, σύνδεση, άλλες πράξεις (προβολή κλπ.)
- Τρόποι εκτέλεσης (και κόστος) πράξεων σχεσιακής άλγεβρας
- ➡ ■ Αποτίμηση σύνθετων εκφράσεων (materialization, pipelining)

Αποτίμηση σύνθετων εκφράσεων



- Μέχρι τώρα είδαμε αλγορίθμους για την αποτίμηση απλών λειτουργιών
- Για την αποτίμηση ενός ολόκληρου **δέντρου εκφράσεων** (*expression tree*) υπάρχουν δύο εναλλακτικές μέθοδοι:
 - **Materialization**: “Παραγωγή” των αποτελεσμάτων μιας έκφρασης της οποίας οι είσοδοι είναι είτε απλές σχέσεις είτε έχουν ήδη υπολογιστεί και αποθηκευτεί (*materialize*) στο δίσκο.
 - **Pipelining**: “Μεταφορά” των πλειάδων στις λειτουργίες ανώτερων επιπέδων (*parent operations*).

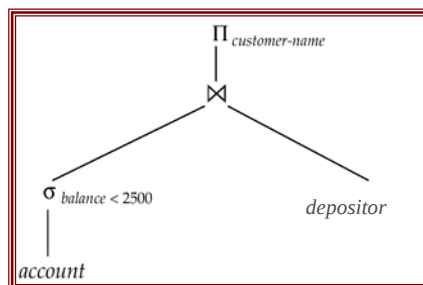
Materialization



- Υπολογισμός μίας λειτουργίας τη φορά, ξεκινώντας από το χαμηλότερο επίπεδο. Χρήση των ενδιάμεσων αποτελεσμάτων (που είναι αποθηκευμένα σε προσωρινές σχέσεις) για τον υπολογισμό των λειτουργιών του επόμενου επιπέδου.
- Π.χ., υπολογισμός και αποθήκευση της λειτουργίας

$\sigma_{balance < 2500}(account)$

- στη συνέχεια υπολογισμός και αποθήκευση της σύνδεσής της με τη σχέση *depositor*, τέλος υπολογισμός της προβολής της ως προς το χαρακτηριστικό *customer-name*.



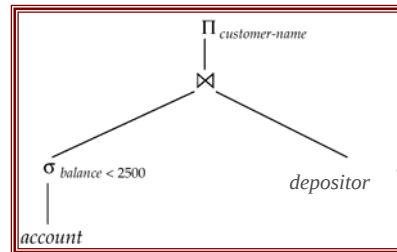
Pipelining



- Υπολογισμός πολλών λειτουργιών ταυτόχρονα, περνώντας τα αποτελέσματα της μιας λειτουργίας στην επόμενη.
- Π.χ. στο προηγούμενο παράδειγμα δεν αποθηκεύουμε το αποτέλεσμα του

$\sigma_{balance < 2500}(account)$

- Αντ' αυτού, περνάμε τις πλειάδες της έκφρασης κατευθείαν στην πράξη της σύνδεσης. Ομοίως, δεν αποθηκεύουμε τα αποτελέσματα της σύνδεσης, αλλά περνάμε τις πλειάδες του αποτελέσματος κατευθείαν στην πράξη της προβολής.
- Πολύ φθηνότερο από το materialization.
- Ενδέχεται να μην είναι πάντα εφικτό – π.χ., ταξινόμηση, hash-join.



Τελικό αποτέλεσμα: ένα σχέδιο εκτέλεσης



- Η έξοδος του optimizer είναι ένα **σχέδιο εκτέλεσης ερωτήματος** (query execution plan – QEP), το οποίο καθορίζει:

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

(β) πώς θα συντονιστεί η εκτέλεση των λειτουργιών.

