

# Επεξεργασία και βελτιστοποίηση ερωτήσεων

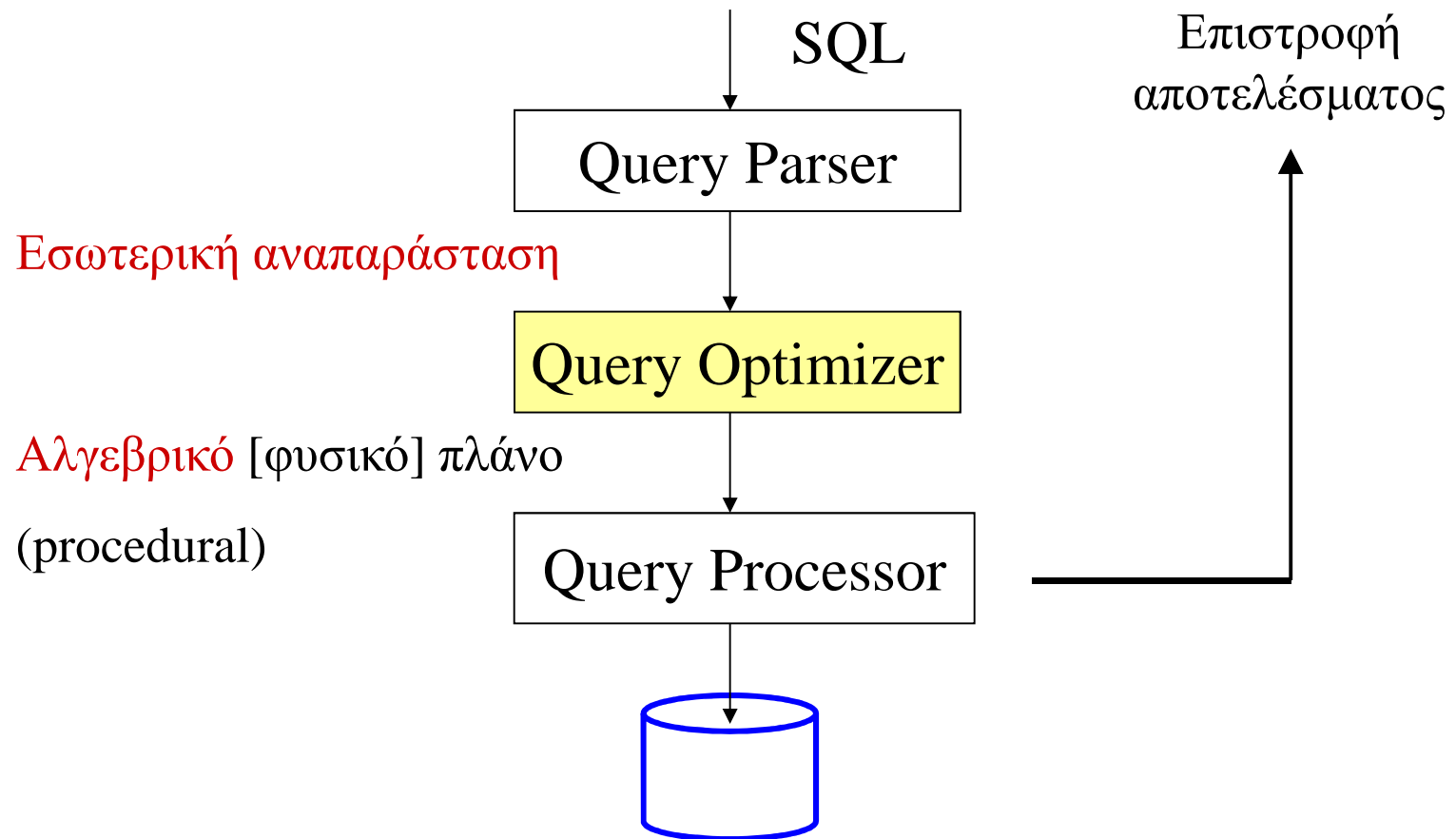
*Πολλές ευχαριστίες στους Πάνο Βασιλειάδη, Γ. Ιωαννίδη, Τ. Σελλή, Ε. Πιτουρά για την  
επαναχρησιμοποίηση κειμένων/διαφανειών τους*

*Οι εικόνες για την DB2 είναι από “DB2 Universal Optimizer” – παρουσίαση του G. Lohman*

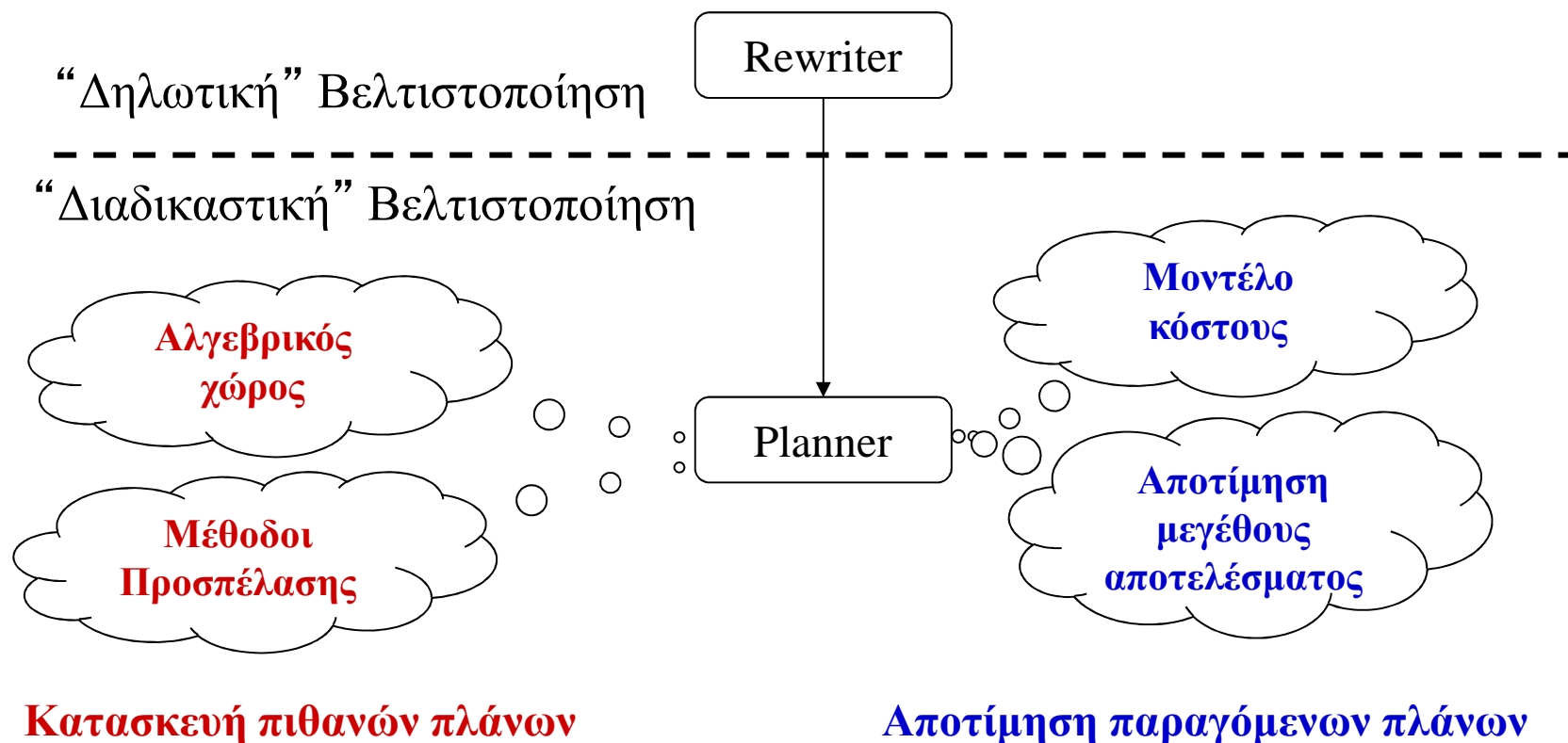
# Θεματολόγιο

- Βελτιστοποίηση ερωτήσεων:
  - Επανεγγραφή ερωτήσεων
  - Παραγωγή εναλλακτικών πλάνων
  - Αποτίμηση πλάνων
  - Πρόβλεψη μεγέθους

# Επεξεργασία ερωτήσεων



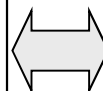
# Αφαιρετική δομή του βελτιστοποιητή



## Επίπεδα βελτιστοποίησης

- ✦ Υπάρχει ένα επίπεδο «δηλωτικής» βελτιστοποίησης, ή επανεγγραφής, όπου παράγουμε λογικά ισοδύναμους τρόπους να εκφράσουμε μια ερώτηση μέσω του **rewriter**

```
select *  
from emp  
where emp.dno in  
      (select dept.dno  
       from dept) and  
       sal > 100K
```

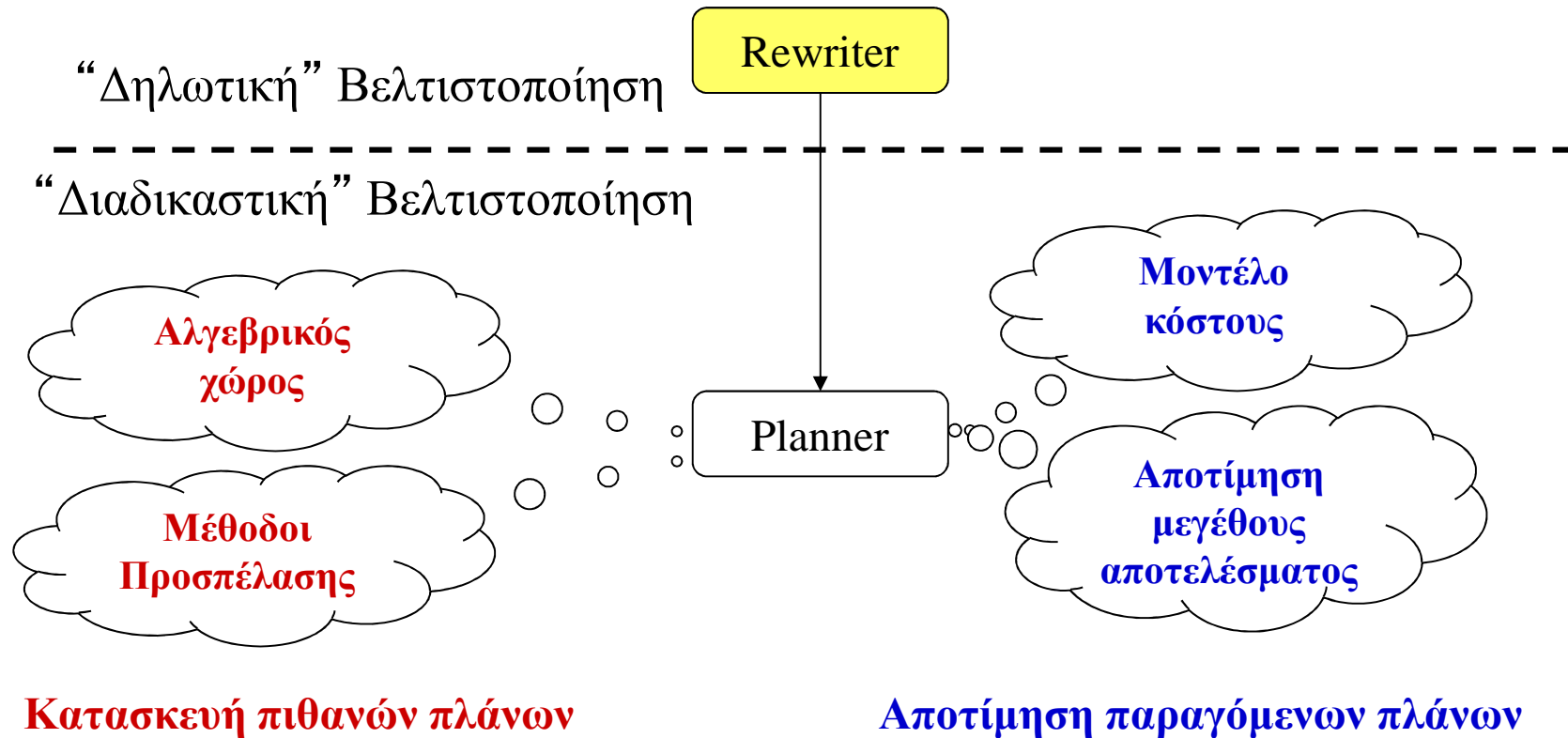


```
select name, age, sal, ndo  
from emp, dept  
where emp.dno = dept.dno  
       and sal > 100K
```

## Επίπεδα βελτιστοποίησης

- ✦ Υπάρχει ένα επίπεδο «διαδικαστικής» βελτιστοποίησης, όπου παράγουμε (όλα τα) διαφορετικά πλάνα εκτέλεσης μέχρι να διαλέξουμε το πιο αποδοτικό. Η δουλειά αυτή ανατίθεται στον **planner**.
- ✦ Ο planner οφείλει:
  - ✦ Να αποφασίσει ποια πλάνα εκτέλεσης θα δημιουργηθούν
  - ✦ Ποιο εξ αυτών είναι το καλύτερο

# Θεματολόγιο: Επανεγγραφή



## Επίπεδα βελτιστοποίησης

- ✦ Υπάρχει ένα επίπεδο «δηλωτικής» βελτιστοποίησης, ή επανεγγραφής, όπου παράγουμε λογικά ισοδύναμους τρόπους να εκφράσουμε μια ερώτηση μέσω του **rewriter**
- ✦ Αυτό συμπεριλαμβάνει, συνήθως:
  - Μετατροπή εκφράσεων σε «βολική» μορφή
  - Απλοποίηση εμφωλευμένων ερωτήσεων
  - Σημασιολογικά έξυπνες μετατροπές



## Ισοδυναμίες της σχεσιακής άλγεβρας

✦ Επιλογή:  $\sigma_{c_1 \wedge \dots \wedge c_n}(R) \equiv \sigma_{c_1}(\dots \sigma_{c_n}(R))$  (*Cascade*)

$$\sigma_{c_1}(\sigma_{c_2}(R)) \equiv \sigma_{c_2}(\sigma_{c_1}(R)) \quad (\text{Commute})$$

✦ Προβολή:  $\pi_{a_1}(R) \equiv \pi_{a_1}(\dots (\pi_{a_n}(R)))$  (*Cascade*)

✦ Σύνδεση:  $R \bowtie (S \bowtie T) \equiv (R \bowtie S) \bowtie T$  (*Associative*)

$$(R \bowtie S) \equiv (S \bowtie R) \quad (\text{Commute})$$

☐ Δ.Ο.:  $R \bowtie (S \bowtie T) \equiv (T \bowtie R) \bowtie S$

## Σύνθετες ισοδυναμίες

- $\sigma_\theta(\pi_A(R)) \equiv \pi_A(\sigma_\theta(R))$ , αρκεί τα πεδία που εμπλέκονται στη συνθήκη  $\theta$  να είναι υποσύνολο των πεδίων του  $A$
- $\sigma_\theta(R \times S) \equiv (R \triangleright \triangleleft_\theta S)$ , αρκεί τα πεδία που εμπλέκονται στη συνθήκη  $\theta$  να είναι από τις σχέσεις  $R$  και  $S$
- $\sigma_\theta(R \triangleright \triangleleft S) \equiv \sigma_\theta(R) \triangleright \triangleleft S$ , αρκεί τα πεδία που εμπλέκονται στη συνθήκη  $\theta$  να είναι αποκλειστικά της  $R$
- $\pi_A(R \triangleright \triangleleft S) \equiv \pi_{A'}(R) \triangleright \triangleleft S$ , με την  $A'$  να περιέχει (α) τα πεδία που είχε η  $A$  (αρκεί να είναι αποκλειστικά της  $R$ ) και (β) τα πεδία που εμπλέκονται στη συνθήκη σύνδεσης
  - *Δ.Ο. ισχύουν τα παραπάνω...*

# Επανεγγραφή Ερωτήσεων (παράδειγμα από IBM DB2)

## Distribute NOT

... WHERE NOT(COL1 = 10 OR COL2 > 3)

γίνεται

... WHERE COL1 <> 10 AND COL2 <= 3

## Μετασχηματισμοί τιμών:

... WHERE COL = YEAR('1994-09-08')

γίνεται

... WHERE COL = 1994

## Μεταβατική κλειστότητα

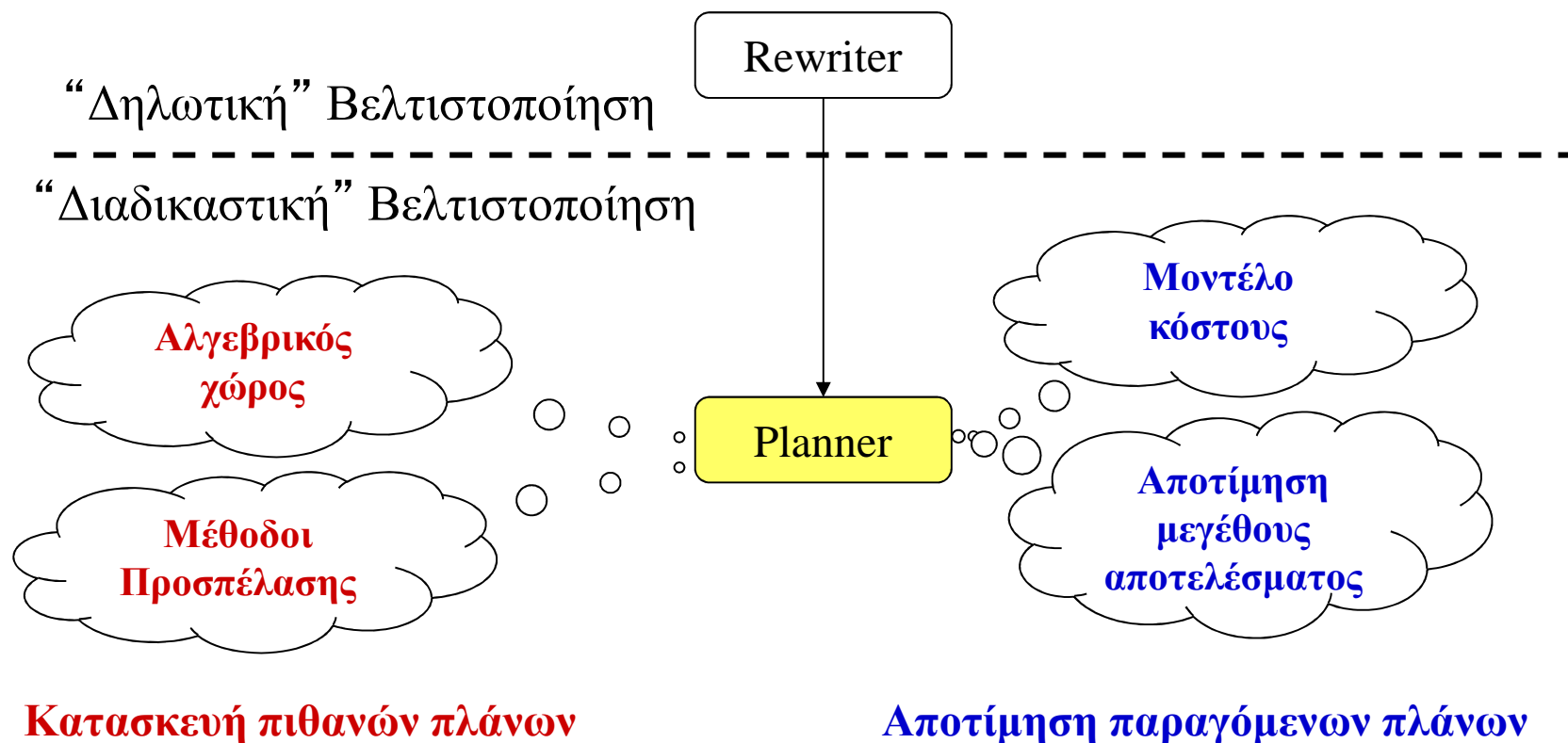
δοθέντος:

T1.C1 = T2.C2, T2.C2 = T3.C3, T1.C1 > 5

προστίθενται...

T1.C1 = T3.C3 AND T2.C2 > 5 AND T3.C3 > 5

# Αφαιρετική δομή του βελτιστοποιητή



## Επίπεδα βελτιστοποίησης

- ✦ Υπάρχει ένα επίπεδο «διαδικαστικής» βελτιστοποίησης, όπου παράγουμε (όλα τα) διαφορετικά πλάνα εκτέλεσης μέχρι να διαλέξουμε το πιο αποδοτικό. Η δουλειά αυτή ανατίθεται στον **planner**.
- ✦ Ο planner οφείλει:
  - ✦ Να αποφασίσει ποια πλάνα εκτέλεσης θα δημιουργηθούν
  - ✦ Ποιο εξ αυτών είναι το καλύτερο

## Ο planner οφείλει ...

- ✦ Να **κατασκευάσει ένα σύνολο πλάνων**, με βάση
  - ✦ Ένα **αλγεβρικό χώρο** για τη σειρά εκτέλεσης των λειτουργιών (π.χ., να αποφασίσει με ποια σειρά θα κάνει το  $R \triangleright \triangleleft S \triangleright \triangleleft T$ )
  - ✦ Ένα σύνολο από **μεθόδους προσπέλασης** στα δεδομένα (π.χ., full-index scan, full table scan, ...)
- ✦ Να **αποτιμά κάθε πλάνο** που παράγει, μέχρι στο τέλος να βρει το πιο αποδοτικό, με βάση
  - ✦ Ένα **μοντέλο κόστους** που προβλέπει πόσο χρόνο/disk I/O/... κοστίζει το κάθε πλάνο
  - ✦ Ένα **μοντέλο πρόβλεψης του μεγέθους**, κυρίως των ενδιάμεσων αποτελεσμάτων
  - ✦ Προσοχή: η αποτίμηση είναι πάντα προσέγγιση/πρόβλεψη και όχι ακριβής υπολογισμός...

## Υποθέσεις ...

- Θα κάνουμε τις εξής υποθέσεις εργασίας (που αφορούν πρακτικά το σύνολο των DBMS) σε ότι αφορά τις εναλλακτικές λύσεις που θα εξετάσουμε:
- Οι μέθοδοι προσπέλασης που έχουμε είναι (α) πλήρες διάβασμα ενός πίνακα (β) προσπέλαση μέσω ενός B+ ευρετηρίου
- Οι μέθοδοι σύνδεσης που έχουμε είναι (α) nested loops και (β) merge-join, στα οποία χρησιμοποιούμε και των δύο ειδών τις μεθόδους προσπέλασης

## Σύνδεση με Nested Loops

```
SELECT E.ename, D.mgr  
FROM Emp E, Dept D  
WHERE D.dname= 'Toys' AND E.dno=D.dno
```

Για κάθε  $D \in \text{Dept}$  [ & Dname= 'Toys' ]

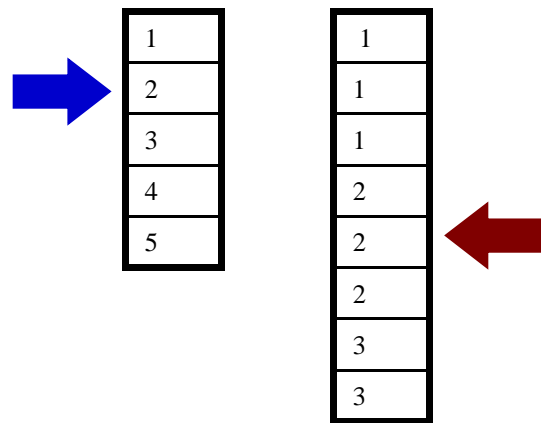
Για κάθε  $E \in \text{Emp}$

Αν ταιριάζουν στο πεδίο Dno

Τότε επέστρεψε E.ename, D.mgr



## Σύνδεση με Merge-Join



```
SELECT E.ename, D.mgr  
FROM Emp E, Dept D  
WHERE D.dname= 'Toys' AND E.dno=D.dno
```

Ταξινόμησε τα Emp, Dept με βάση το πεδίο Dno

Για κάθε  $D \in \text{Dept} [\& \text{Dname}= 'Toys' ]$

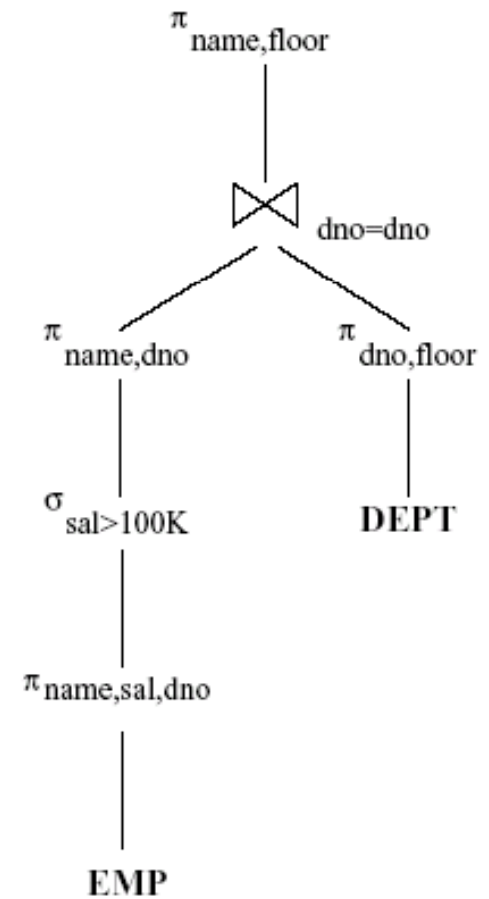
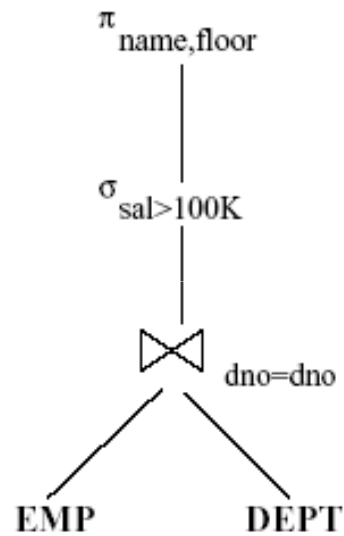
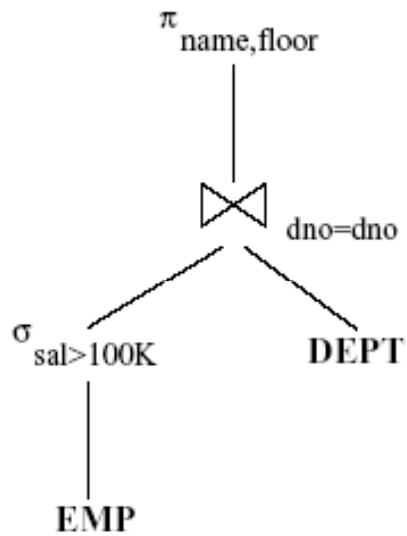
Για κάθε  $E \in \text{Emp}$  με ίδιο Dno

Επέστρεψε E.ename, D.mgr

## Αλγεβρικός χώρος: τι είναι ένα πλάνο

- Το πλάνο εκτέλεσης μιας SQL ερώτησης είναι ένα **δέντρο**, με:
  - ✦ Τις **σχέσεις** που συμμετέχουν στην ερώτηση, για **φύλλα**
  - ✦ **Αλγεβρικούς τελεστές** για **ενδιάμεσους κόμβους** και συγκεκριμένα τους  $\pi$ ,  $\sigma$  και  $\triangleright \triangleleft$
- Το πλάνο έχει **σειρά εκτέλεσης από κάτω και αριστερά προς τα πάνω**.
- Κοιτώντας ένα ενδιάμεσο κόμβο, ξέρουμε ότι τα παιδιά του έχουν εκτελεστεί και αυτός στέλνει το αποτέλεσμα προς τα πάνω

# Πλάνα εκτέλεσης



```
select name, floor
from emp, dept
where emp.dno = dept.dno
and sal > 100K
```

## Πλάνα εκτέλεσης

- Για μια απλή SELECT-FROM-WHERE ερώτηση SQL, ο αριθμός των ισοδύναμων εναλλακτικών πλάνων είναι τεράστιος.
- Υπάρχουν κάποιοι **λογικοί κανόνες**, που επιτρέπουν στον βελτιστοποιητή να μειώσει τον αλγεβρικό χώρο πλάνων

## Λογικοί κανόνες βελτιστοποίησης

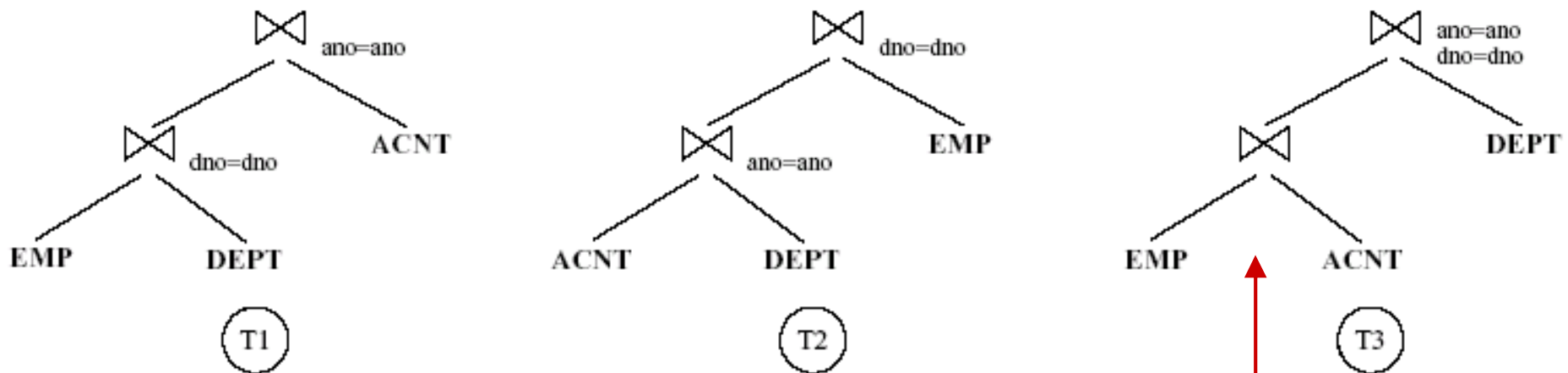
- Σπρώξε όλες τις επιλογές όσο πιο χαμηλά στο δέντρο γίνεται
- Ενσωμάτωσε τις προβολές μέσα στους άλλους τελεστές
- ... και πάλι όμως, ο αλγεβρικός χώρος παραμένει τεράστιος ...



## Λογικοί κανόνες βελτιστοποίησης

- Η βασική αιτία είναι οι ιδιότητες της σύνδεσης:
  - $R \bowtie S \Leftrightarrow S \bowtie R$
  - $(R \bowtie S) \bowtie T \Leftrightarrow R \bowtie (S \bowtie T)$
- Το αποτέλεσμα είναι ότι για **N** σχέσεις στο FROM clause έχω **N!** διατάξεις...
- Επιπλέον κανόνας:
  - **Ποτέ** μην κάνεις **καρτεσιανά γινόμενα**, εκτός κι αν πρέπει...

## Ποτέ μην κάνεις καρτεσιανά γινόμενα

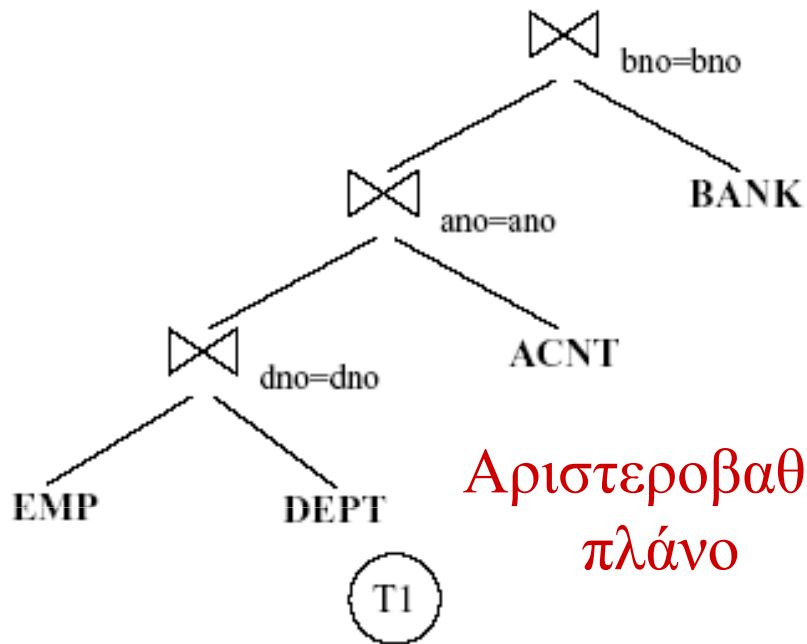


```
select name, floor, balance
from emp, dept, acnt
where emp.dno = dept.dno
and dept.ano = acnt.ano
```

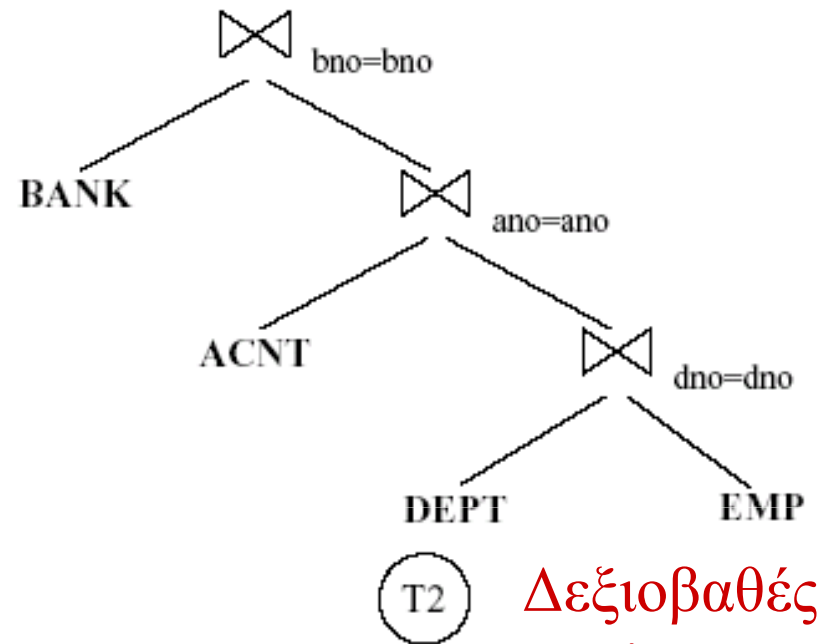
## Αριστεροβαθή Δέντρα

- Ακόμα και τώρα όμως, ο αλγεβρικός χώρος είναι μεγάλος
- Όλα τα σύγχρονα DBMS έχουν εισάγει τον ακόλουθο πρακτικό κανόνα:
- Η εσωτερική σχέση ενός τελεστή είναι ΠΑΝΤΑ μια σχέση της ΒΔ και ποτέ ενδιάμεσο αποτέλεσμα!
- Τα δέντρα που προκύπτουν έτσι, λέγονται **αριστεροβαθή (left-deep)**



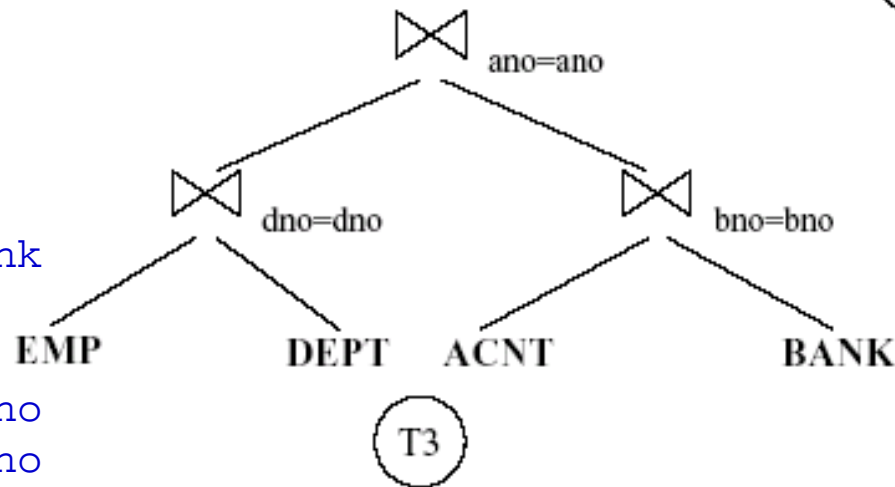


Αριστεροβαθές  
πλάνο



Δεξιοβαθές  
πλάνο

**Select** name, floor,  
balance, bank  
**from** emp,dept,acct,bank  
**where**  
emp.dno = dept.dno  
and dept.ano = acct.ano  
and acct.bno = bank.bno



Θαμνώδες  
(bushy) πλάνο

## Αριστεροβαθή Δέντρα

- ✦ Κέρδη από αριστεροβαθή δέντρα:
  - ✦ Μπορούμε εύκολα να χρησιμοποιούμε **ευρετήρια** για τις σχέσεις!
  - ✦ Τα αποτελέσματα από μια σύνδεση μπορούν να γίνουν **pipeline** σε μια επόμενη σύνδεση!

## Με ποια σειρά?

- Ακόμα δεν μας είπες για τη **σειρά** των συνδέσεων!  
 $(R \triangleright \triangleleft S) \triangleright \triangleleft T \Leftrightarrow R \triangleright \triangleleft (S \triangleright \triangleleft T)$
- Ο planner, σε όλα τα εμπορικά DBMS χρησιμοποιεί ένα **αλγόριθμο δυναμικού προγραμματισμού** για να ανακαλύψει τη σειρά
- Προτού δώσουμε το γενικό τρόπο δημιουργίας των πλάνων, **θα κατηγοριοποιήσουμε τις ερωτήσεις** ως:
  - Ερωτήσεις που αφορούν **μία σχέση** στο FROM clause
  - Ερωτήσεις που αφορούν **πολλές σχέσεις** στο FROM clause

## Ερωτήσεις με μία σχέση στο FROM clause

- Οι ερωτήσεις αφορούν ένα συνδυασμό προβολών, επιλογών και συναθροίσεων. Η επιλογή του πλάνου γίνεται ως ακολούθως:
  - Εξετάζεται κάθε διαθέσιμη μέθοδος προσπέλασης (file scan / index) και επιλέγεται αυτή με το ελάχιστο κόστος
  - Οι τελεστές εκτελούνται, όσο το δυνατόν γίνεται, μαζί (π.χ., οι προβολές και οι επιλογές ενσωματώνονται στην προσπέλαση μέσω ευρετηρίου).

## Εκτίμηση κόστους για πλάνα μίας σχέσης

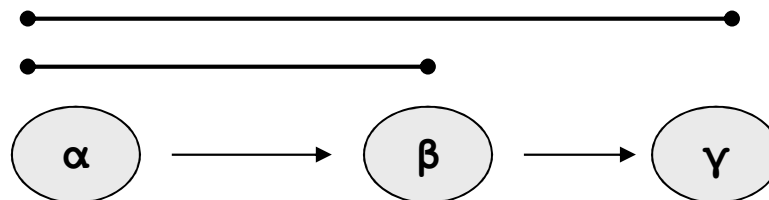
- ✦ Index I στο πρωτεύον κλειδί που χρησιμοποιείται για μια επιλογή:
  - ✦ Κόστος:  $\text{Height}(I)+1$  για B+ tree, περίπου 1.2 για hash index.
- ✦ Clustered index I που χρησιμοποιείται για μια ή περισσότερες επιλογές  $\sigma_1, \dots, \sigma_n$ :
  - ✦  $(\#Pages(I) + \#Pages(R)) * \prod_i (\text{sel}(\sigma_i)), i=1, \dots, n$
- ✦ Non-clustered index I που χρησιμοποιείται για μια ή περισσότερες επιλογές  $\sigma_1, \dots, \sigma_n$ :
  - ✦  $(\#Pages(I) + \#Tuples(R)) * \prod_i (\text{sel}(\sigma_i)), i=1, \dots, n$
- ✦ Sequential scan μιας σχέσης:
  - ✦  $\#Pages(R)$ .

## Αριστεροβαθή πλάνα για ερωτήσεις πολλών σχέσεων

- ✦ Τα αριστεροβαθή πλάνα διαφέρουν
  - ✦ στη **σειρά των σχέσεων**,
  - ✦ στη **μέθοδο προσπέλασης** για κάθε σχέση (index/file scan), και
  - ✦ στον **τρόπο εκτέλεσης** κάθε σύνδεσης (NLJ,SMJ,HJ)
- ✦ Οι πράξεις ORDER BY, GROUP BY, κλπ., εξετάζονται ως μια τελική πράξη που επικάθεται ενός πλάνου, ενδεχομένως ταξινομώντας το αποτέλεσμα των συνδέσεων αν αυτό δεν είναι ήδη βολικά ταξινομημένο.
- ✦ Και πάλι, όμως, ο αριθμός των υπό εξέταση πλάνων είναι εκθετικός σε σχέση με τον αριθμό των εμπλεκόμενων σχέσεων

## Δυναμικός προγραμματισμός

- Εφαρμόζεται σε προβλήματα, στα οποία η λύση μπορεί να εκφρασθεί ως μια **ακολουθία αποφάσεων**
- Εκμεταλλεύεται το **principle of optimality**: μια ακολουθία αποφάσεων (λύση) δεν μπορεί να είναι βέλτιστη, αν μια υπακολουθία της δεν είναι βέλτιστη



## Δυναμικός προγραμματισμός για επεξεργασία ερωτήσεων

- Πρόβλημα: **ποια η σωστή σειρά** για να εκτελέσω το  $R \triangleright \triangleleft S \triangleright \triangleleft T$ ?
- **Δυναμικός Προγραμματισμός:**
  1. Θα βρω όλους τους τρόπους για να προσπελάσω κάθε σχέση χωριστά
  2. Θα πάρω κάθε τέτοιο τρόπο προσπέλασης και θα φτιάξω το **καλύτερο** υποδέντρο με δύο φύλλα που του αντιστοιχεί
  3. Θα πάρω κάθε τέτοιο υποδέντρο και θα φτιάξω το καλύτερο υποδέντρο με τρία φύλλα που του αντιστοιχεί



## Ενδιαφέρουσες σειρές

- **Ενδιαφέρουσα σειρά** (interesting order) είναι μια λίστα από πεδία που εμπλέκονται σε μια ερώτηση είτε στο GROUP BY, είτε στο ORDER BY clause, είτε σε μία σύνδεση (Join)

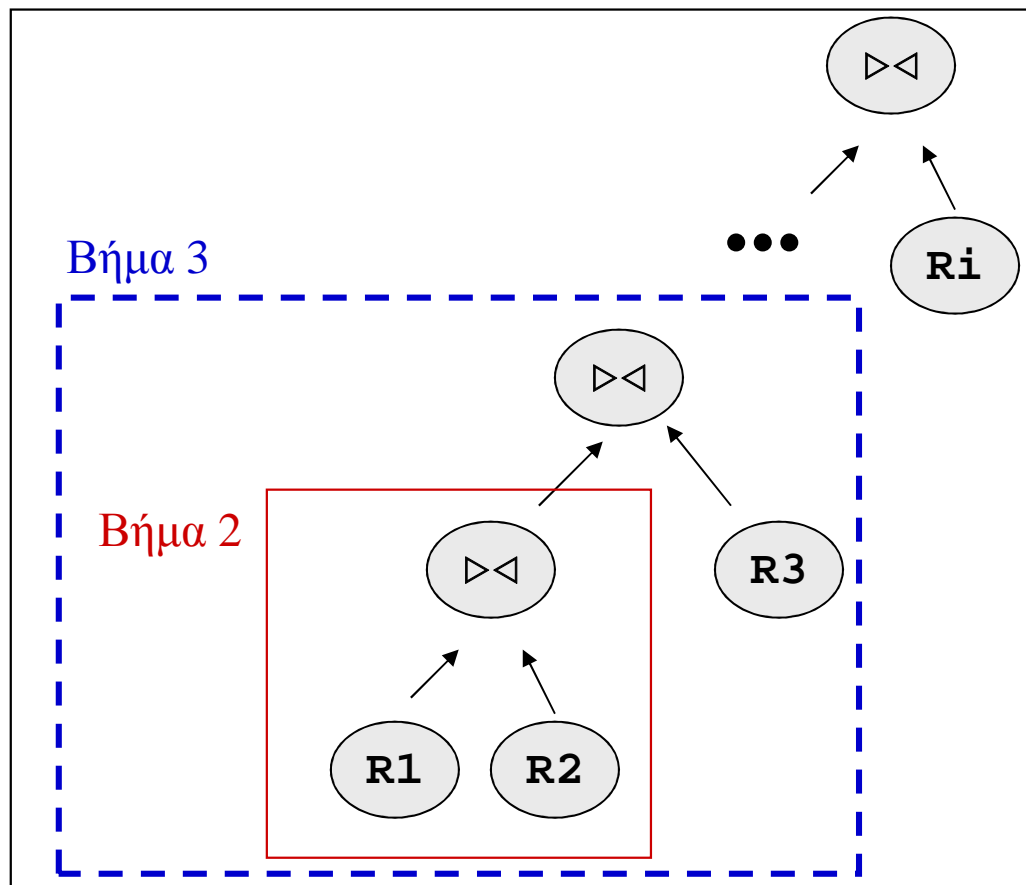
<b>SELECT X,Y,SUM(Z)</b>	Interesting orders
<b>FROM R,S</b>	{X,Y}
<b>WHERE R.A = S.A</b>	{A}
<b>GROUP BY X,Y</b>	{}

# Αριστεροβαθή πλάνα για ερωτήσεις πολλών σχέσεων

- Πρόβλημα: **ποια η σωστή σειρά** για να εκτελέσω το  $R_1 \bowtie R_2 \bowtie \dots R_N$ ?
- Κάνε  $N$  περάσματα, (αν συνδέουμε  $N$  σχέσεις):
  - Pass 0: Ομαδοποίησε τις μεθόδους προσπέλασης κάθε σχέσης σε σχέση με την ταξινόμηση των εγγραφών. Κάθε τέτοια ομάδα ονομάζεται «**ενδιαφέρουσα σειρά**» (**interesting order**)
  - Pass 1: Βρες το καλύτερο 1-relation πλάνο για κάθε ενδιαφέρουσα ομάδα μιας σχέσης
  - Pass 2: Βρες τον καλύτερο τρόπο σύνδεσης κάθε 1-relation plan (ως outer) με μια άλλη σχέση και φτιάξε όλα τα 2-relation plans για κάθε ενδιαφέρουσα ομάδα μιας σχέσης.
  - ... Για το υπο-πλάνο που συνδέει  $k$  σχέσεις, ( $k < N$ ), για κάθε interesting order, κάθε φορά, κρατάμε το καλύτερο πλάνο και κλαδεύουμε τα υπόλοιπα... Αν έχουμε  $m$  το πλήθος interesting orders, στο τέλος του βήματος μας μένουν  $m$  πλάνα
  - Pass  $N$ : Βρες τον καλύτερο τρόπο σύνδεσης κάθε  $(N-1)$ -relation plan (ως outer) με τη  $N$ -στη σχέση που του απομένει.

# Δυναμικός προγραμματισμός για επεξεργασία ερωτήσεων

Βήμα i



Εν παραλλήλω,  
φτιάχνω πολλά  
δέντρα.

Σιγά σιγά όμως,  
μειώνω τον αριθμό  
τους, κρατώντας  
μόνο το πιο φτηνό  
για κάθε  
ενδιαφέρουσα σειρά

## Και τι πάει να πει «καλύτερο» πλάνο?

- ✦ **Μοντέλο κόστους:** ένα σύνολο αριθμητικών εκφράσεων που μου επιτρέπει να υπολογίζω πόσο θα κοστίσει ο κάθε φυσικός τελεστής
- ✦ Π.χ., για το nested loops του  $R \triangleright \triangleleft S$  :
  - ✦  $\text{Size}(R) + \text{Size}(S)$  αν ένα εκ των δύο χωρά στη μνήμη
  - ✦  $[\text{Size}(R) / \text{Size}(\text{buffers}) - 1] * \text{Size}(S)$ , αλλιώς

## Και τι πάει να πει καλύτερο ?

➤ Κι αν έχω  $((R \triangleright \triangleleft S) \triangleright \triangleleft T)$ , ήτοι, πρώτα το  $R \triangleright \triangleleft S$  και, μετά, το αποτέλεσμα του με το  $T$ , τότε τι κόστος θα έχω ?

➤ Απλό:

$$\color{blue}{+} \text{ Cost}(R \triangleright \triangleleft S) = [\text{Size}(R) / \text{Size}(\text{buffers}) - 1] * \text{Size}(S)$$

$$\color{red}{+} \text{ Cost}((R \triangleright \triangleleft S) \triangleright \triangleleft T) =$$

$$[\color{red}{\text{Size}(R \triangleright \triangleleft S)} / \text{Size}(\text{buffers}) - 1] * \text{Size}(T)$$

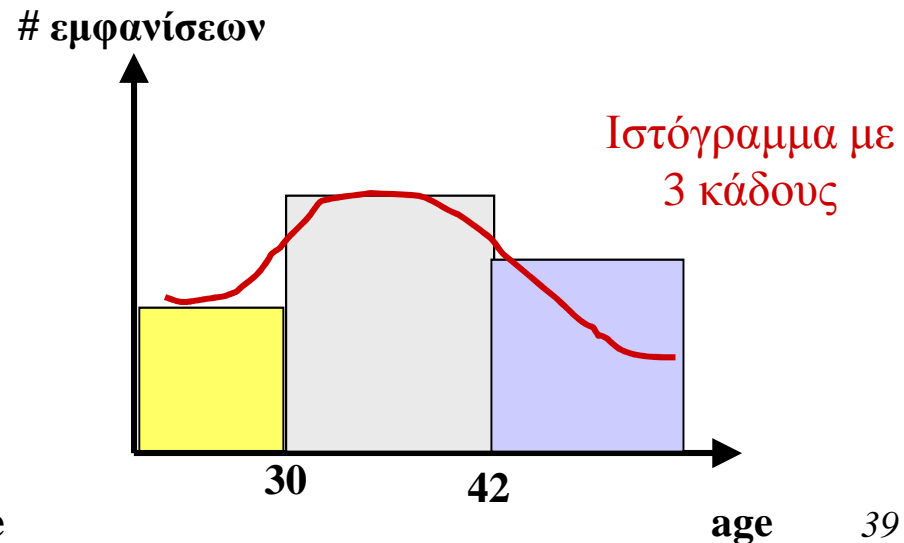
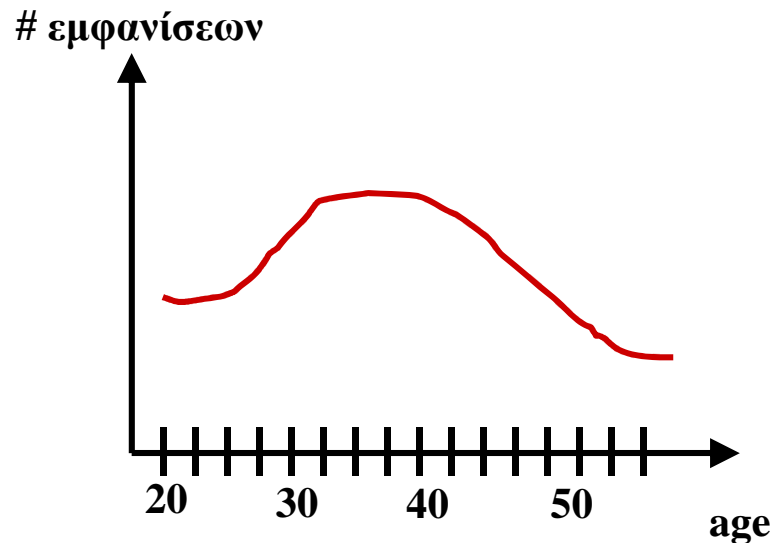
➤  $\color{red}{\text{Size}(R \triangleright \triangleleft S)???$  Και πού το ξέρουμε αυτό ??

## Εκτίμηση μεγέθους

- ✦ Για να δουλέψουν οι φόρμουλες κόστους που έχουμε, πρέπει να μπορούμε να **αποτιμήσουμε το μέγεθος των ενδιάμεσων αποτελεσμάτων**
- ✦ Εν γένει, δεν είμαστε πολύ καλοί σ' αυτό, πρακτικά οι τρόποι εκτίμησης που έχουμε δουλεύουν σε δέντρα με ύψος πάνω από 5...
- ✦ Η πιο καλή τεχνική που έχουμε είναι τα **ιστογράμματα**

# Ιστογράμματα

- Σ' ένα ιστόγραμμα, διαιρούμε το εύρος των τιμών ενός πεδίου σε **κάδους** (αγγλιστί: buckets)
- Για κάθε τιμή που παίρνει το πεδίο, μετράω τον αριθμό που αυτή εμφανίζεται



# Ιστογράμματα

- ✦ Και πώς αποφασίζω πόσους κάδους?
  - ✦ **Ιστογράμματα ίσου πλάτους:** κάθε κάδος έχει τον ίδιο αριθμό τιμών στον άξονα των  $x$
  - ✦ **Ιστογράμματα ίσου ύψους:** κάθε κάδος έχει το ίδιο ύψος στον άξονα των  $y$
  - ✦ **Σειριακά ιστογράμματα:** οι συχνότητες ενός κάδου είναι μεγαλύτερες από αυτές του προηγούμενου



## Ιστογράμματα

- ✦ Αν κάνω μια επιλογή  $\sigma_{\text{age} > 43}(\text{emp})$  το DBMS μπορεί να εκτιμήσει περίπου πόσες εγγραφές θα μου επιστραφούν
- ✦ Αντίστοιχα, αν κάνω μια σύνδεση  $R \bowtie S$  πάλι μπορεί να κάνει την αντίστοιχη εκτίμηση ανά ζεύγος κάδων.

## Ιστογράμματα

- Είναι σαφές ότι όσο πιο πολλές πράξεις, τόσο πιο πολύ απομακρύνεται η εκτίμηση από την πραγματικότητα...
- (Λανθασμένες) **Υποθέσεις εργασίας**: οι τιμές των πεδίων είναι ισοπίθανα μοιρασμένες + τα πεδία είναι ανεξάρτητα μεταξύ τους...

# Ιστογράμματα

Name	Salary	Department
Zeus	100K	General Manager
Poseidon	80K	Defense
Pluto	80K	Justice
Aris	50K	Defense
Ermis	60K	Commerce
Apollo	60K	Energy
Hefestus	50K	Energy
Hera	90K	General Manager
Athena	70K	Education
Aphro	60K	Domestic Affairs
Demeter	60K	Agriculture
Hestia	50K	Domestic Affairs
Artemis	60K	Energy

Department	Frequency
General Manager	2
Defense	2
Education	1
Domestic Affairs	2
Agriculture	1
Commerce	1
Justice	1
Energy	3

# Ιστογράμματα

Department	Απόλυτη Συχνότητα	Συχνότητα Κάδου
Agriculture	1	1.50
Commerce	1	1.50
Defense	2	1.50
Domestic Affairs	2	1.50
Education	1	1.75
Energy	3	1.75
General Manager	2	1.75
Justice	1	1.75

*#πραγματικών εμφανίσεων*

*εκτίμηση ιστογράμματος*

- ✦ **Ιστόγραμμα ίσου πλάτους:** κάθε κάδος έχει τον ίδιο αριθμό τιμών στον άξονα των x
- ✦ Εδώ: δύο κάδοι, ο πρώτος από A – D και ο άλλος από E-Z
- ✦ Συχν. Κάδου:  $\Sigma(x)/\text{count}(x)$ ,  $x \in \text{κάδο}$

# Ιστογράμματα

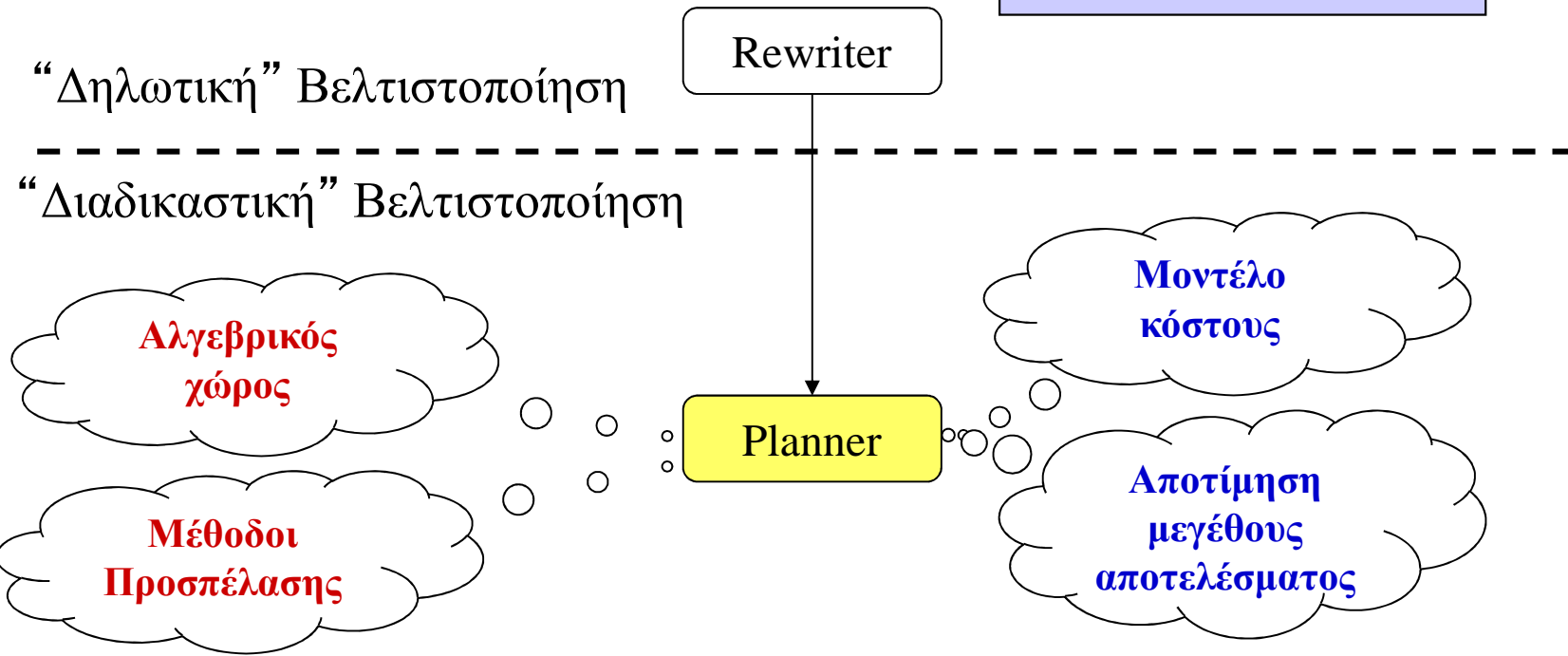
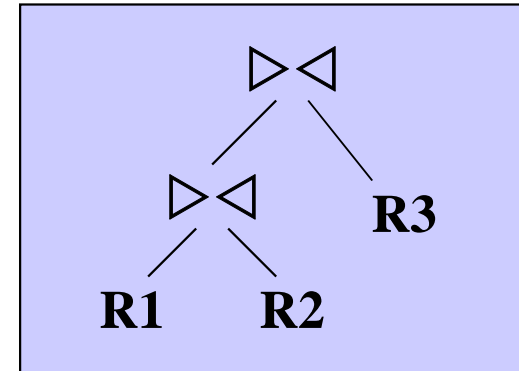
Department	Απόλυτη Συχνότητα	Συχνότητα Κάδου
Agriculture	1	1.33
Commerce	1	1.33
Defense	2	1.33
Education	1	1.33
General Manager	2	1.33
Justice	1	1.33
Domestic Affairs	2	2.50
Energy	3	2.50

*#πραγματικών εμφανίσεων*

*εκτίμηση ιστογράμματος*

- ✦ **Ιστόγραμμα σειριακό:** οι συχνότητες του δεύτερου κάδου είναι μεγαλύτερες από αυτές του πρώτου

# Επανάληψη



**Κατασκευή πιθανών πλάνων**

**Αποτίμηση παραγόμενων πλάνων**

# **Παράδειγμα Παραγωγής πλάνων στην βελτιστοποίηση ερωτήσεων**

## Σχήμα αναφοράς

`emp(name, age, sal, dno)`

`dept(dno, dname, floor, budget, mgr, ano)`

`acnt(ano, type, balance, bno)`

`bank(bno, bname, address)`

- Κάθε υπάλληλος (emp) εργάζεται σε ένα τμήμα (dept)
- Κάθε τμήμα έχει ένα λογαριασμό (acnt) ο οποίος βρίσκεται σε μια τράπεζα (bank)
- Έστω η ερώτηση  $\pi_{\text{NAME,BUDGET,BALANCE}}(E \bowtie D \bowtie A)$

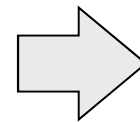


# Υποθέσεις - Access paths

## *Access paths*

### ✦ EMP

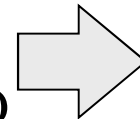
- ✦ 100.000 tuples, 100 σελίδες
- ✦ Όχι ταξινομημένη
- ✦ B+ tree στο E.NAME



E. FTS  
E.INDX.NAME

### ✦ DEPT

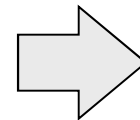
- ✦ 10.000 tuples, 10 σελίδες
- ✦ Ταξινομημένη βάσει του D.DNO
- ✦ B+ tree στο D.DNO



D.FTS  
D.INDX.DNO

### ✦ ACNT

- ✦ 10.000 tuples, 10 σελίδες
- ✦ Όχι ταξινομημένη
- ✦ B+ tree στο A.ANO



A.FTS  
A.INDX.ANO

*FTS: Full Table Scan*

## Υποθέσεις - Παρατηρήσεις

- ✦ Έστω ότι έχω αρκετή μνήμη για να δίνω  $\sqrt{|R|}$  buffers στη μνήμη σε μια σχέση R
  - ✦ 10 buffers για EMP,
  - ✦ 3 buffers για DEPT, ACNT
- ✦ **Επιλεκτικότητα** = 1 για όλα τα joins (έστω ότι έχω FK στα DNO, ANO)
  - ✦  $|E \bowtie D| = 100$  pages
  - ✦  $|D \bowtie A| = 10$  pages
  - ✦  $E \bowtie D \bowtie A = 100$  pages

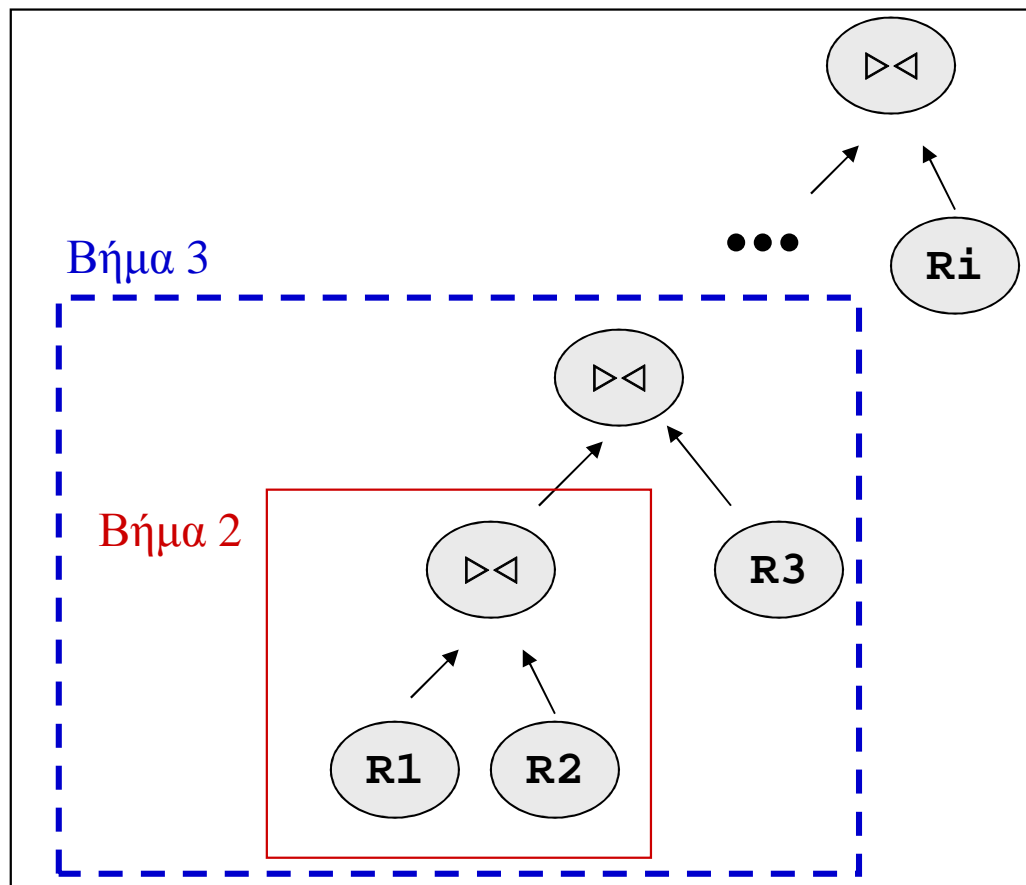
(κάνοντας τον χονδροειδή υπολογισμό ότι μία σελίδα εξακολουθεί να έχει 1000 εγγραφές)

## Ενδιαφέρουσες Ταξινομήσεις

- ✦ Ενδιαφέρουσες ταξινομήσεις (interesting orders)
  - ✦ DNO (E.DNO, D.DNO)
  - ✦ ANO (D.ANO, A.ANO)
  - ✦ No order
- ✦ Θυμηθείτε ότι οι ενδιαφέρουσες ταξινομήσεις είναι ομάδες πεδίων και περιλαμβάνουν τα πεδία των συνδέσεων, τα πεδία των GROUP BY, ORDER BY clauses, καθώς και το no-order.

# Δυναμικός προγραμματισμός για επεξεργασία ερωτήσεων

Βήμα i



Εν παραλλήλω,  
φτιάχνω πολλά  
δέντρα.

Σιγά σιγά όμως,  
μειώνω τον αριθμό  
τους...

# Βήμα 1

Έστω τα παρακάτω κόστη ανάκτησης των σχέσεων EMP,  
DEPT, ACNT:

- E.FTS 100 pages
- E.INDX.NAME 110 pages
  
- D.FTS 10 pages
- D.INDX.DNO 11 pages
  
- A.FTS 10 pages
- A.INDX.ANO 11 pages

# Βήμα 1

Έστω τα παρακάτω κόστη ανάκτησης των σχέσεων EMP,  
DEPT, ACNT:

➤ E.FTS	100 pages	}	No-order
➤ E.INDX.NAME	110 pages		
➤ D.FTS	10 pages	}	DNO
➤ D.INDX.DNO	11 pages		
➤ A.FTS	10 pages		No-order
➤ A.INDX.ANO	11 pages		ANO

# Βήμα 1

Έστω τα παρακάτω κόστη ανάκτησης των σχέσεων EMP,  
DEPT, ACNT:

➤ E.FTS 100 pages → No-order

~~➤ E.INDX.NAME 110 pages~~

➤ D.FTS 10 pages → DNO

~~➤ D.INDX.DNO 11 pages~~

➤ A.FTS 10 pages No-order

➤ A.INDX.ANO 11 pages ANO

## Βήμα 2

- ✦ Υπολογισμός **join** ανά δύο σχέσεις, για όλα τα πιθανά **access paths** και ομαδοποίηση ανά **interesting order** για να διαλέξουμε το φθηνότερο
  - ✦  $E \triangleright \triangleleft D, D \triangleright \triangleleft E$
  - ✦  $D \triangleright \triangleleft A, A \triangleright \triangleleft D$
  - ✦ Προσοχή: όχι καρτεσιανά γινόμενα ( $E \times A$ )
- ✦ Έστω ότι έχω μόνο **NLJ, MSJ** ως τρόπους εκτέλεσης των **joins**



	Formula	Estimated Cost	Result size (pages)	Output order
$E \bowtie_{NLJ} D$	$100 + \frac{100}{10} * 10$	200	100	No-order
$E \bowtie_{MSJ} D$	$100 * 2.00 + 100 + 10$ ( $E * \log E + D * \log D + E + D$ )	310	100	DNO
$D \bowtie_{NLJ} E$	$10 + \frac{10}{3} * 100$	343.00	100	No-order
$D \bowtie_{MSJ} E$	Ίδιο με πριν	310	100	DNO
$D \bowtie_{NLJ} A.FTS$	$10 + \frac{10}{3} * 10$	43.33	10	DNO
$D \bowtie_{NLJ} A.INX$	$10 + 10 * (1 + 1)$ (scan D + $\forall d \in D, tree + A$ )	30	10	DNO
$D \bowtie_{MSJ} A.FTS$	$10 * 1.0 + 10 * 1.0 + 10 + 10$	40	10	ANO
$D \bowtie_{MSJ} A.INX$	$10 * 1.0 + 11 * 1.0 + 10 + 11$	31	10	ANO
$A.FTS \bowtie_{NLJ} D$	$10 + \frac{10}{3} * 10$	43.30	10	No order
$A.INX \bowtie_{NLJ} D$	$11 + \frac{11}{3} * 11$	47.66	10	ANO
$A.FTS \bowtie_{MSJ} D$	Ίδιο με πριν	40	10	ANO
$A.INX \bowtie_{MSJ} D$	Ίδιο με FTS	33	10	ANO

	Formula	Estimated Cost	Result size (pages)	Output order
$E \bowtie_{NLJ} D$	$100 + \frac{100}{10} * 10$	200	100	No-order
$E \bowtie_{MSJ} D$	$100 * 2.00 + 100 + 10$ ( $E * \log E + D * \log D + E + D$ )	310	100	DNO
<del><math>D \bowtie_{NLJ} E</math></del>	<del><math>10 + \frac{10}{5} * 100</math></del>	<del>343.00</del>	<del>100</del>	<del>No-order</del>
<del><math>D \bowtie_{MSJ} E</math></del>	<del>Ίδιο με πριν</del>	<del>310</del>	<del>100</del>	<del>DNO</del>
<del><math>D \bowtie_{NLJ} A.FTS</math></del>	<del><math>10 + \frac{10}{3} * 10</math></del>	<del>43.33</del>	<del>10</del>	<del>DNO</del>
$D \bowtie_{NLJ} A.INX$	$10 + 10 * (1 + 1)$ (scan D + $\forall d \in D, tree + A$ )	30	10	DNO
<del><math>D \bowtie_{MSJ} A.FTS</math></del>	<del><math>10 * 1.0 + 10 * 1.0 + 10 + 10</math></del>	<del>40</del>	<del>10</del>	<del>ANO</del>
$D \bowtie_{MSJ} A.INX$	$10 * 1.0 + 11 * 1.0 + 10 + 11$	31	10	ANO
$A.FTS \bowtie_{NLJ} D$	$10 + \frac{10}{3} * 10$	43.30	10	No order
<del><math>A.INX \bowtie_{NLJ} D</math></del>	<del><math>11 + \frac{11}{3} * 11</math></del>	<del>47.66</del>	<del>10</del>	<del>ANO</del>
<del><math>A.FTS \bowtie_{MSJ} D</math></del>	<del>Ίδιο με πριν</del>	<del>40</del>	<del>10</del>	<del>ANO</del>
$A.INX \bowtie_{MSJ} D$	Ίδιο με FTS	33	10	ANO

Για κάθε ζεύγος σχέσεων, για κάθε ενδιαφέρουσα σειρά, διαλέγω το πιο φθηνό πλάνο

Στις ισοπαλίες διαλέγω τυχαία κάποιο

## Νικητές του βήματος 2

	Estimated Cost	Result size (pages)	Output order	To be joined with
$E \bowtie_{NLJ} D$	200	100	No-order	A.FTS, A.INX
$E \bowtie_{MSJ} D$	310	100	DNO	

	Estimated Cost	Result size (pages)	Output order	To be joined with
$D \bowtie_{NLJ} A.INX$	43.33	10	DNO	E
$D \bowtie_{MSJ} A.INX$	31	10	ANO	
$A.FTS \bowtie_{NLJ} D$	43.30	10	No order	

## Βήμα 3

- Όσα δένδρα επιβίωσαν από τη φάση 2, θα συνδεθούν με ένα δένδρο ύψους 1 (ήτοι, με την επόμενη –και τελευταία- σχέση)
- Όλα τα αποτελέσματα (οφείλουν να) έχουν το ίδιο μέγεθος αποτελέσματος (ποιο?)
- Προσοχή στα μη αριστεροβαθή δέντρα!

	Formula	Estimated Cost	Overall query cost	Output order
$(E \bowtie_{NLJ} D) \bowtie_{NLJ} A.FTS$	<del>400</del> + $\frac{100}{10} * 10$ (λόγω pipelining)	100	+200 = <b>300</b>	No order
$(E \bowtie_{NLJ} D) \bowtie_{NLJ} A.INX$	100*(1+1)	200	<b>400</b>	No order
$(E \bowtie_{NLJ} D) \bowtie_{MSJ} A.FTS$	100*2.00 + 10*1.0 + 100 + 10	320	<b>520</b>	ANO
$(E \bowtie_{NLJ} D) \bowtie_{MSJ} A.INX$	100 * 2.0 + 100 + 11	311	<b>511</b>	ANO
$A.FTS \bowtie_{NLJ}(E \bowtie_{NLJ} D)$	<b>Προσοχή: ΔΕΞΙΟΒΑΘΕΣ!</b>			
$(E \bowtie_{MSJ} D) \bowtie_* A.*$	Οι υπολογισμοί παραλείπονται. Είναι εύκολο να δει κανείς κατ' ευθείαν ότι θα προκύψουν μεγαλύτερα νούμερα από τα παραπάνω, ούτως ή άλλως. Γιατί?			



	Formula	Estimated Cost	Overall query cost	Output order
$(D \bowtie_{NLJ} A.INX) \bowtie_{NLJ} E$	<del>40</del> + $\frac{10}{3} * 100$	333.00	+30 = <b>363.00</b>	DNO
$(D \bowtie_{NLJ} A.INX) \bowtie_{MSJ} E$	<del>40*1.0</del> + 100*2.0 + <del>40</del> + 100	300	<b>330</b>	DNO
$(D \bowtie_{MSJ} A.INX) \bowtie_{NLJ} E$	<del>40</del> + $\frac{10}{3} * 100$	333.00	+ 31 = <b>364.00</b>	DNO
$(D \bowtie_{MSJ} A.INX) \bowtie_{MSJ} E$	<del>40*1.0</del> + 100*2.0 + <del>40</del> + 100	300	<b>331</b>	DNO
$(A.FTS \bowtie_{NLJ} D) \bowtie_{NLJ} E$	<del>40</del> + $\frac{10}{3} * 100$	333.00	+ 43.33 = <b>376.33</b>	ANO
$(A.FTS \bowtie_{NLJ} D) \bowtie_{MSJ} E$	10*1.0 + 100*2.0 + 10 + 100	320	<b>363.33</b>	DNO

## Οπότε ...

- ✦ Νικά η λύση  $(E \bowtie_{NLJ} D) \bowtie_{NLJ} A.FTS$ 
  - ✦ με συνολικό κόστος 300 σελίδες.
- ✦ Αν θα είχα ζητήσει το αποτέλεσμα ORDER BY DNO, ή ORDER BY ANO, τι θα έκανα?
  - ✦ Min (total winner sorted by the requested order, best solution already having the requested order)