

# Επεξεργασία Ερωτήσεων

## Επεξεργασία Ερωτήσεων

Ερώτηση του χρήστη σε κάποια γλώσσα υψηλού επιπέδου (π.χ., SQL)

Επεξεργαστής Ερωτήσεων (Query Processor)

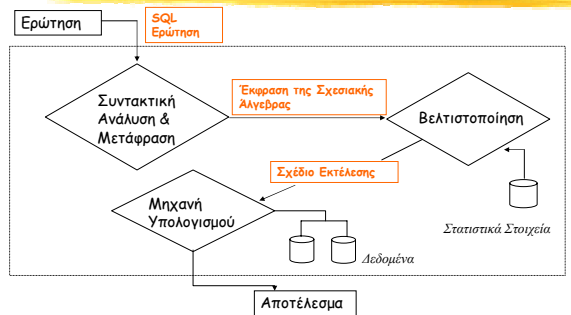
Χαμηλού επιπέδου εντολές επεξεργασίας δεδομένων (π.χ., υλοποιήσεις πράξεων της σχεσιακής άλγεβρας)

## Επεξεργασία Ερωτήσεων

⌘ Μεθοδολογία Εκτέλεσης της ερώτησης  
(τα βήματα που ακολουθούνται για την εκτέλεση της ερώτησης)

⌘ Βελτιστοποίηση Ερώτησης  
⌘ Πως καθορίζουμε το καλύτερο πλάνο εκτέλεσης

## Επεξεργασία Ερωτήσεων σε Κεντροποιημένο ΣΔΒΔ



## Επεξεργασία Ερωτήσεων σε Κεντροποιημένο ΣΔΒΔ

Τα βασικά βήματα στην επεξεργασία μιας ερώτησης είναι

1. Συντακτική Ανάλυση & Μετάφραση
2. Βελτιστοποίηση
3. Υπολογισμός

## Επεξεργασία Ερωτήσεων σε Κεντροποιημένο ΣΔΒΔ

### 1. Συντακτική Ανάλυση (Parsing) & Μετάφραση

Η SQL ερώτηση μεταφράζεται σε μια εσωτερική μορφή αφού γίνει ο απαραίτητος συντακτικός και σημασιολογικός έλεγχος (π.χ., τα ονόματα που αναφέρονται είναι ονόματα σχέσεων που υπάρχουν)

Αντικατάσταση των όψεων από τον ορισμό τους

Σε ποια εσωτερική μορφή: Έκφραση της σχεσιακής άλγεβρας

```
select A1, A2, ..., An
from R1, R2, ..., Rm      πA1, A2, ..., An (σP (R1 x R2 x ... x Rm))
where P
```

## 2. Βελτιστοποίηση

Μια SQL ερώτηση μπορεί να μεταφραστεί σε διαφορετικές (ισοδύναμες) εκφράσεις της σχεσιακής άλγεβρας

`select balance`  
`from account`  
`where balance < 25000`

- $\sigma_{\text{balance} < 2500} (\pi_{\text{balance}}(\text{account}))$
- $\pi_{\text{balance}} (\sigma_{\text{balance} < 2500} (\text{account}))$

### Ισοδυναμίες μετασχηματισμών

- Μία σχεσιακή ερώτηση μπορεί να εκφραστεί ως ο συνδυασμός διάφορων εκφράσεων της σχεσιακής άλγεβρας.
- Επηρεάζει τον τρόπο με τον οποίο θα υπολογιστεί το αποτέλεσμα και τη σειρά με την οποία θα γίνει ο υπολογισμός αυτός.

Διαφορετικά πλάνα εκτέλεσης για το ίδιο αποτέλεσμα.

### Δέντρο Τελεστών (operator tree)

Παράδειγμα

$\pi_{\text{TNUM, ROLE}} \sigma_{\text{CITY} = \text{'RHO'}} (\text{TASK Join DEPT\_TASK})$

Δύο ειδών τελεστές:

- μοναδιαίοι (unary) (επιλογή, προβολή (*select, project*))
- δυαδικό (binary) (καρτεσιανό γινόμενο, ένωση, διαφορά, τομή, ανένωση, φυσική συνένωση, ημι-συνένωση (*cartesian product, union, difference, intersection, join, natural join, semi-join*)).

### Κανόνες ισοδυναμιών μεταξύ συνδυασμών τελεστών

Δύο εκφράσεις είναι ισοδύναμες: αν αντικαταστήσουμε τη μία με την άλλη, το αποτέλεσμα θα παραμείνει το ίδιο, ανεξάρτητα από τα δεδομένα που περιέχει η βάση δεδομένων.

$U$  : Μοναδιαίοι τελεστές  
 $B$  : Δυαδικό τελεστές  
 $R$  : Σχέσεις

Αντιμεταθετικότητα μοναδιαίων τελεστών  
 $U_1 U_2 R \Leftrightarrow U_2 U_1 R$

Αντιμεταθετικότητα δυαδικών τελεστών  
 $R_1 B R_2 \Leftrightarrow R_2 B R_1$

Προσεταιριστικότητα δυαδικών τελεστών  
 $R_1 B (R_2 B R_3) \Leftrightarrow (R_1 B R_2) B R_3$

Μοναδική ποσότητα μοναδιαίων τελεστών  
 $U R \Leftrightarrow U_1 U_2 R$

Επιμεριστικότητα μοναδιαίων τελεστών σε σχέση με δυαδικούς  
 $U(R_1) B U(R_2) \Leftrightarrow U(R_1 B R_2)R$

Οι ιδιότητες αυτές ισχύουν μόνο όταν κάποιες συνθήκες πληρούνται.

Παραδείγματα

- Αντιμεταθετικότητα μεταξύ επιλογής και προβολής  
*Ισχύει όταν τα πεδία που λαμβάνουν μέρος στη συνθήκη επιλογής είναι υποσύνολο των πεδίων που αφήνει η προβολή στο τελικό αποτέλεσμα*

$$\pi_{A_1} \sigma_{F_2} (R) = \sigma_{F_2} \pi_{A_1} (R), \text{ όταν } \text{Attr}(F_2) \subseteq A_1$$

- Προσεταιριστικότητα μεταξύ συνενώσεων  
*Ισχύει όταν τα πεδία στα οποία γίνεται η δεύτερη συνένωση είναι υποσύνολο των πεδίων της δεύτερης και της τρίτης σχέσης*

$$R_1 \Join_{F_1} (R_2 \Join_{F_2} R_3) \Leftrightarrow (R_1 \Join_{F_1} R_2) \Join_{F_2} R_3, \text{ όταν } \text{Attr}(F_2) \subseteq \text{Attr}(R_2) \cup \text{Attr}(R_3)$$

## Επεξεργασία Ερωτήσεων σε Κεντροποιημένο ΣΔΒΔ

- Μοναδική ποσότητα μεταξύ προβολών  
είναι επιτρεπτή όταν τα πεδία της προβολής του αριστερού μέλους της ισότητας είναι τα ίδια με αυτά της πρώτης προβολής του δεξιού και υποσύνολο των πεδίων της δεύτερης προβολής

$$\pi_A R = \pi_{A_1} (\pi_{A_2} R), \text{ όταν } A = A_1, A \subseteq A_2$$

- Προσεταιριστικότητα μεταξύ καρτεσιανού γινομένου και προβολής  
είναι επιτρεπτή όταν ισχύει η εξής συνθήκη:

$$\pi_A (R_1 \times R_2) = \pi_{A_1} (R_1) \times \pi_{A_2} (R_2), \text{ όταν } A_1 = A - \text{Attr}(R_1), A_2 = A - \text{Attr}(R_2)$$

## Επεξεργασία Ερωτήσεων σε Κεντροποιημένο ΣΔΒΔ

- Προσεταιριστικότητα μεταξύ ένωσης και σύνδεσης  
ισχύει πάντα

$$(R_1 \cup R_2) \Join (S_1 \cup S_2) = (R_1 \Join S_1) \cup (R_1 \Join S_2) \cup (R_2 \Join S_1) \cup (R_2 \Join S_2)$$

(η ιδιότητα αυτή είναι εξαιρετικά χρήσιμη στον υπολογισμό του αποτελέσματος συνδέσεων καταγεγραμμένων σχέσεων (αν φανταστούμε ότι τα  $R_1, R_2, S_1, S_2$  είναι τμήματα των καθολικών σχέσεων  $R, S$  και εκτελείται η ερώτηση  $R \Join S$ ).

## Κεντροποιημένο ΣΔΒΔ: Βελτιστοποίηση Ερωτήσεων

### Δέντρο ερώτησης

Φύλλα: σχέσεις εισόδου

Εσωτερικοί κόμβοι: πράξεις της σχεσιακής άλγεβρας

Εκτέλεση δέντρου ερώτησης

## Κεντροποιημένο ΣΔΒΔ: Βελτιστοποίηση Ερωτήσεων

### Ευριστικοί Κανόνες

- Διάσπαση των πράξεων επιλογής με συζευκτικές συνθήκες σε ακολουθίες πράξεων επιλογής
- Μετατοπίζουμε την πράξη επιλογής όσο πιο κάτω επιτρέπεται από τα γνωρίσματα που περιλαμβάνονται στη συνθήκη
- Επαναδιευθέτηση των φύλλων ώστε να εκτελούνται πρώτα οι σχέσεις που έχουν τις πιο περιοριστικές πράξεις επιλογής

## Κεντροποιημένο ΣΔΒΔ: Βελτιστοποίηση Ερωτήσεων

- Συνδυασμός μιας πράξης καρτεσιανού γινομένου με μια πράξη επιλογής που ακολουθεί
- Διάσπαση και μετακίνηση των λιστών προβολής όσο πιο κάτω γίνεται στο δέντρο
- Εντοπισμός υποδέντρων με ομάδες πράξεων που μπορεί να εκτελεστούν με κοινό αλγόριθμο

## Επεξεργασία Ερωτήσεων σε Κεντροποιημένο ΣΔΒΔ

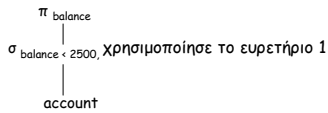
Κάθε πράξη της σχεσιακής άλγεβρας μπορεί να υλοποιηθεί με διαφορετικούς αλγόριθμους:

π.χ., για την υλοποίηση της επιλογής μπορεί είτε να σαρώσουμε (scan) όλο το αρχείο ελέγχοντας κάθε εγγραφή αν ικανοποιεί τη συνθήκη είτε αν υπάρχει π.χ., ένα  $B^+$  ευρετήριο σε κάποιο γνώρισμα να χρησιμοποιήσουμε το ευρετήριο

Άρα δεν αρκεί ο προσδιορισμός της πράξης - πρέπει να προσδιορίζεται και ο αλγόριθμος που θα χρησιμοποιηθεί για την υλοποίησή της

βασικές (primitive) πράξεις: πράξη + αλγόριθμος

Σχέδιο εκτέλεσης (execution plan): μια ακολουθία από βασικές πράξεις



- Τα διαφορετικά σχέδια εκτέλεσης έχουν και διαφορεικό κόστος

- **Βελτιστοποίηση:** η διαδικασία επιλογής του σχεδίου εκτέλεσης που έχει το μικρότερο κόστος

- Εκτίμηση του κόστους (συνήθως χρήση στατιστικών στοιχείων)

### 3. Εκτέλεση

Μηχανή εκτέλεσης που εκτελεί τις βασικές πράξεις

Τι θα συζητήσουμε:

1. Μετασχηματισμός ερωτήσεων που απευθύνονται σε καθολικές σχέσεις σε ερωτήσεις που απευθύνονται σε συγκεκριμένα τμήματα της κατακευμασμένης βάσης.
2. Μεθόδους που βελτιστοποιούν την αποτίμηση μιας ερώτησης.

Θέματα

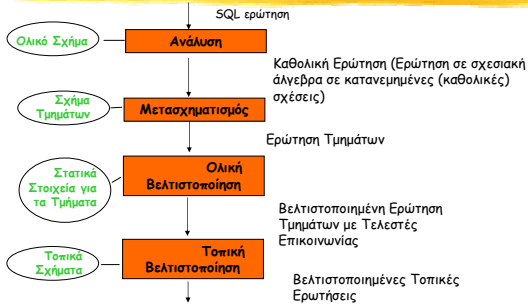
- ⌘ Το κόστος μετάδοσης δεδομένων στο δίκτυο
- ⌘ Πιθανό κέρδος από το ό,τι διάφοροι κόμβοι μπορεί να επεξεργάζονται μια ερώτηση παράλληλα (ταυτόχρονα) : διαφορά συνολικού χρόνου και χρόνου απόκρισης

**Καθολικές ερωτήσεις (global queries):** ερωτήσεις που απευθύνονται σε καθολικές σχέσεις και

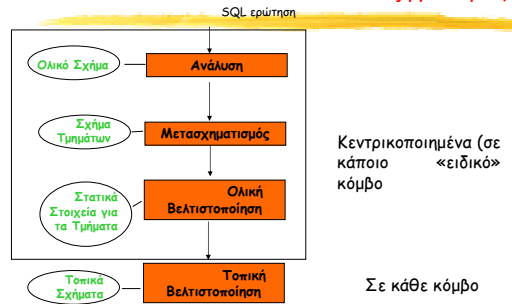
**Ερωτήσεις τμημάτων (fragment queries):** ερωτήσεις που απευθύνονται σε τμήματα.

Βασικοί στόχοι μετασχηματισμών: **πληρότητα** και **ορθότητα** (ο συνδυασμός των ερωτήσεων τμημάτων να επιστρέφει ακριβώς και μόνο το αποτέλεσμα της καθολικής ερώτησης).

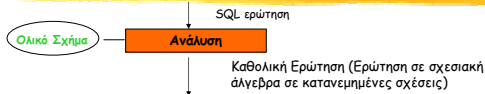
## Επεξεργασία Ερωτήσεων



## Επεξεργασία Ερωτήσεων



## 1ο Βήμα: Ανάλυση Ερωτήσεων



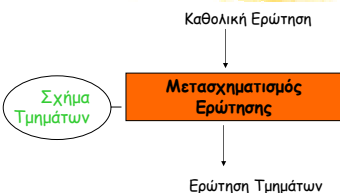
1. Λεξική και συντακτική ανάλυση
2. Παραγωγή του δέντρου ερώτησης (Σημείωση: τα φύλλα αντιστοιχούν σε καθολικές σχέσεις)
3. Πιθανές βελτιστοποιήσεις

## Ανάλυση: Παράδειγμα

```
select ENAME
from EMP, ASG, PROJ
where EMP.ENO= ASG.ENO
and ASG.PNO = PROJ.PNO
and ENAME <> "J. DOE"
and PNAME = "CAD/CAM"
and (DUR = 12 OR DUR = 24)
```

Δέντρο Ερώτησης

## 2ο Βήμα: Μετασχηματισμός Ερωτήσης



1. Καθορισμός των τμημάτων που συμμετέχουν στην ερώτηση
2. Αντικατάσταση της σχέσης από τα τμήματά της
3. Βελτιστοποίηση

## Παράδειγμα

```
SELECT AVG(S.age)
FROM Sailors S
WHERE S.rating > 3
AND S.rating < 7
```

- ⊗ Οριζόντιος καταταξιωματισμός: Οι πλειάδες με rating < 5 στη Shanghai, >= 5 στο Tokyo.
- ☒ Πρέπει να υπολογίσουμε SUM(age), COUNT(age) και στους δύο κόμβους.
- ☒ Αν το WHERE περιέχει μόνο S.rating > 6, θα αρκούσε ο υπολογισμός σε έναν μόνο κόμβο (απλοποίηση)

Μετασχηματισμοί: Παράδειγμα

Μετασχηματισμοί της ερώτησης

```
SELECT ...
FROM Sailors S
WHERE S.rating > κ1
AND S.rating < κ2
```

$$\sigma_{S.rating > \kappa1 \text{ and } S.rating < \kappa2} (S) = \sigma_{S.rating > \kappa1 \text{ and } S.rating < \kappa2} (S1 \cup S2) = \sigma_{S.rating > \kappa1 \text{ and } S.rating < \kappa2} (S1) \cup \sigma_{S.rating > \kappa1 \text{ and } S.rating < \kappa2} (S2)$$

Οριζόντιος κατατεμαχισμός

Εκτέλεση στη Shanghai

Εκτέλεση στο Tokyo

Δυνατές απλοποιήσεις:

Οριζόντιος κατατεμαχισμός:  
 Οι πλειάδες S1 με rating < 5 στη Shanghai, οι πλειάδες S2 με rating >= 5 στο Tokyo.

Μετασχηματισμοί: Παράδειγμα

```
SELECT AVG(S.age)
FROM Sailors S
WHERE S.rating > 3
AND S.rating < 7
```

Κάθετος κατατεμαχισμός: sid και rating στη Shanghai, sname and age στο Tokyo, tid και στους δύο κόμβους.  
 Πρέπει πρώτα να ξανά-σχηματιστεί η σχέση με συνένωση στο tid, και μετά να υπολογιστεί η ερώτηση

Μετασχηματισμοί: Παράδειγμα

Μετασχηματισμοί της ερώτησης

```
SELECT ...
FROM Sailors S
WHERE S.rating > κ1
AND S.rating < κ2
```

$$\sigma_{S.rating > \kappa1 \text{ and } S.rating < \kappa2} (S) = \sigma_{S.rating > \kappa1 \text{ and } S.rating < \kappa2} (S1 * S2) = \sigma_{S.rating > \kappa1 \text{ and } S.rating < \kappa2} (S1) \times \sigma_{S.rating > \kappa1 \text{ and } S.rating < \kappa2} (S2)$$

Κάθετος κατατεμαχισμός

Κάθετος κατατεμαχισμός:  
 sid και rating στη Shanghai, sname and age στο Tokyo, tid και στους δύο κόμβους.

Μετασχηματισμοί: Παράδειγμα

```
SELECT AVG(S.age)
FROM Sailors S
WHERE S.rating > 3
AND S.rating < 7
```

Αντίγραφα: αντίγραφα της σχέσης Sailors και στους δύο κόμβους.  
 Επιλογή του κόμβου με βάση το τοπικό κόστος και το κόστος για την μεταφορά του αποτελέσματος

Παράδειγμα (οριζόντια τμηματοποίηση)

```
EMP
EMP1 = σ_{ENO ∈ E3} (EMP)
EMP2 = σ_{E3 < ENO ≤ E6} (EMP)
EMP3 = σ_{ENO > E6} (EMP)
```

```
select ENAME
from EMP, ASG, PROJ
where EMP.ENO = ASG.ENO and ASG.PNO = PROJ.PNO and ENAME <> 'J. DOE' and PNAME = 'CAD/CAM' and (DUR = 12 OR DUR = 24)
```

```
ASG
ASG1 = σ_{ENO ∈ E3} (ASG)
ASG2 = σ_{ENO > E3} (ASG)
```

Αντικατάσταση του EMP με το (EMP1 ∪ EMP2 ∪ EMP3)  
 Αντικατάσταση του ASG με το (ASG1 ∪ ASG2)

Νέο δέντρο ερώτησης

Παράδειγμα

```
EMP ASG
EMP1 = σ_{ENO ∈ E3} (EMP) ASG1 = σ_{ENO ∈ E3} (ASG)
EMP2 = σ_{E3 < ENO ≤ E6} (EMP) ASG2 = σ_{ENO ∈ E3} (ASG)
EMP3 = σ_{ENO > E6} (EMP)
```

Παράλληλισμός

Χρήση της προεταριατικότητας μεταξύ ένωσης και συνένωσης

$$(R1 \cup R2) \Join (S1 \cup S2) = (R1 \Join S1) \cup (R1 \Join S2) \cup (R2 \Join S1) \cup (R2 \Join S2)$$

Νέο δέντρο ερώτησης

Απλοποιήσεις:

## Ελαχιστοποιήσεις για Οριζόντια Τμηματοποίηση

Γενικά η ελαχιστοποίηση για οριζόντια τμήματα αφορά (μετά την αναδόμηση των δέντρων ερώτησης) τον καθορισμό των υποδέντρων που παράγουν άδειες σχέσεις και την απαλοιφή τους

1. Ελαχιστοποίηση με επιλογή
2. Ελαχιστοποίηση με συνένωση

## Ελαχιστοποιήσεις για Οριζόντια Τμηματοποίηση

### Ελαχιστοποίηση με Επιλογή

Αν η συνθήκη της επιλογής αντιβαίνει τη συνθήκη επιλογής της κατάτμησης

Έστω τμήμα  $R_j$ ;  $R_j = \sigma_{P_j}(R)$

$\sigma_{P_i}(R_j) = \emptyset, \forall x \in R: \neg(P_i(x) \wedge P_j(x))$

EMP

EMP1 =  $\sigma_{ENO \neq E5}(EMP)$

EMP2 =  $\sigma_{E3 \neq ENO \neq E6}(EMP)$

EMP3 =  $\sigma_{ENO \neq E6}(EMP)$

Παράδειγμα

```
select *
from EMP
where ENO = "E5"
```

## Ελαχιστοποιήσεις για Οριζόντια Τμηματοποίηση

### Ελαχιστοποίηση με Συνένωση

Δυνατή αν η συνθήκη για την κατάτμηση είναι με βάση το γνώρισμα της συνένωσης. Σε αυτήν την περίπτωση, καταμερισμός των συνενώσεων και της ένωσης και απαλοιφή των άχρηστων συνενώσεων

## Ελαχιστοποιήσεις για Κάθετη Τμηματοποίηση

EMP

EMP1 =  $\pi_{ENO, ENAME}(EMP)$

EMP2 =  $\pi_{ENO, TITLE}(EMP)$

select ENAME

from EMP

Αντικατάσταση του EMP με

EMP1 J EMP2

Απλοποίηση

Γενικά η ελαχιστοποίηση για κάθετα τμήματα αφορά (μετά την αναδόμηση των δέντρων ερώτησης) τον καθορισμό των υποδέντρων που παράγουν άχρηστες ενδιάμεσες σχέσεις και την απαλοιφή τους

1. Ελαχιστοποίηση με προβολή

## Ελαχιστοποιήσεις για Κάθετη Τμηματοποίηση

### Ελαχιστοποίηση με Προβολή

Άχρηστη να τα γνώρισματα της προβολής δεν ανήκουν στο τμήμα!

## Ελαχιστοποιήσεις

Οι κανόνες για την κεντρικοποιημένη περίπτωση συν:

1. Απαλοιφή άδειων σχέσεων που προκύπτουν από μη συμβατές επιλογές σε οριζόντια τμήματα
2. Απαλοιφή άχρηστων σχέσεων που προκύπτουν από προβολές σε κάθετα τμήματα
3. Κατανομή των συνενώσεων πάνω από τις ενώσεις και απαλοιφή των άχρηστων συνενώσεων

## Ελαχιστοποιήσεις για Υβριδική Τμηματοποίηση

EMP

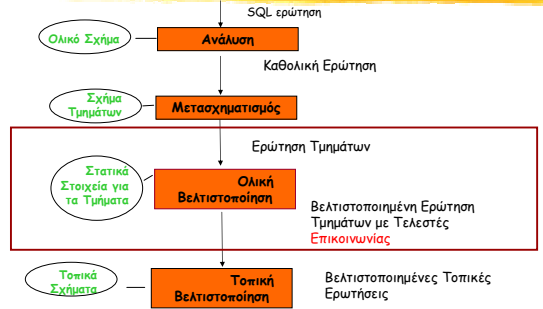
EMP1 =  $\sigma_{ENO \leq 4} (\pi_{ENO, ENAME} (EMP))$

EMP2 =  $\sigma_{ENO > 4} (\pi_{ENO, ENAME} (EMP))$

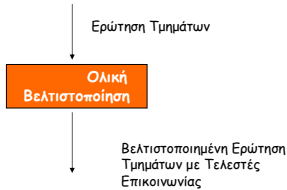
EMP3 =  $\pi_{ENO, TITLE} (EMP)$

```
select ENAME
from EMP
where ENO = "E5"
```

## Επεξεργασία Ερωτήσεων



## Ολική Βελτιστοποίηση



## Ολική Βελτιστοποίηση Ερωτήσεων

Εύρεση ενός καλού (όχι απαραίτητα βέλτιστου) ολικού πλάνου

1. Ελαχιστοποίηση κάποιας συνάρτησης κόστους
2. Καταγεγραμμένη επεξεργασία συνένωσης  
Bushy/linear δέντρα  
Τιες σχέσεις να μεταφέρουμε που  
Μεταφορά όλης της σχέσης ή όταν χρειάζεται
3. Χρήση ή όχι της ημι-συνένωσης
4. Μέθοδοι υπολογισμού συνένωσης

## Ολική Βελτιστοποίηση Ερωτήσεων

- ⌘ Υπολογισμός όλων των πλάνων - επιλογή αυτού με το μικρότερο κόστος
  - ☒ Διαφορά 1: Κόστος επικοινωνίας
  - ☒ Διαφορά 2: Τοπική αυτονομία κάθε κόμβου
  - ☒ Διαφορά 3: Νέοι τρόποι υπολογισμού καταγεγραμμένης συνένωσης
- ⌘ Κατασκευή ολικού πλάνου -- κάθε κόμβος εκτελεί το δικό του κομμάτι με τον καλύτερο (τοπικά) τρόπο

## Ελαχιστοποίηση Κόστους

- ⌘ Το κόστος μετάδοσης δεδομένων στο δίκτυο
- ⌘ Πιθανό κέρδος από το ό,τι διάφοροι κόμβοι μπορεί να επεξεργάζονται μια ερώτηση παράλληλα (ταυτόχρονα) : διαφορά συνολικού χρόνου και χρόνου απόκρισης



## Ελαχιστοποίηση Κόστους

**Ολικό Κόστος** = Κόστος Υπολογισμού (CPU) + I/O Κόστος + Κόστος Επικοινωνίας

- Σε δίκτυα ευρείας περιοχής (WAN)  
Το κόστος επικοινωνίας πολύ μεγαλύτερο από το I/O κόστος (20:1)
- Σε τοπικά δίκτυα (LAN)  
Το κόστος επικοινωνίας και το κόστος για την τοπική επεξεργασία (I/O και CPU) συγκρίσιμα (1:1.6)

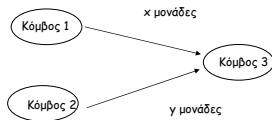
## Ελαχιστοποίηση Κόστους

**Χρόνος απόκρισης** για μια ερώτηση: χρόνος που περνά από την υποβολή της ερώτησης μέχρι την ολοκλήρωσή της

Χρόνος απόκρισης = Χρόνο CPU + χρόνο I/O + χρόνο επικοινωνίας

## Ελαχιστοποίηση Κόστους

**Παράδειγμα (διαφορά μεταξύ ολικού χρόνου και χρόνου απόκρισης)**



Ας θεωρήσουμε μόνο το χρόνο επικοινωνίας, έστω  $t_c$  ο χρόνος μετάδοσης

Ολικός χρόνος =  $x * t_c + \gamma * t_c$

Χρόνος απόκρισης =  $\max\{x * t_c, \gamma * t_c\}$

## Επεξεργασία Ερωτήσεων: Καταμεμημένη Συνένωση

LONDON

S

500 pages

PARIS

R

1000 pages

D κόστος εγγραφής/ανάγνωσης σελίδας

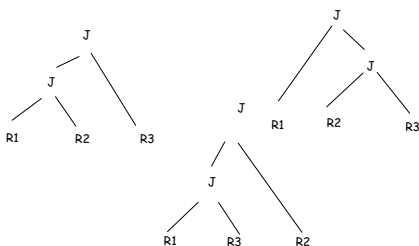
T κόστος μεταφοράς σελίδας

Υπολογισμός S J R

Πολλαπλές σχέσεις, πιο δύσκολο γιατί υπάρχουν πολλές δυνατές επιλογές και χρειάζεται ο υπολογισμός και του μεγέθους των ενδιάμεσων αποτελεσμάτων

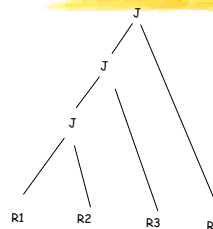
## Καταμεμημένη Συνένωση

Για N σχέσεις υπάρχουν  $O(N!)$  ισοδύναμα δέντρα συνένωσης

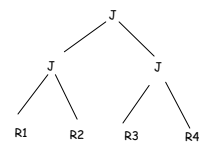


## Καταμεμημένη Συνένωση

Δυο σημαντικά είδη



Left Linear Join Tree  
(pipelining)



Bushy Join Tree  
(παραλληλισμός)

LONDON

S

500 pages

PARIS

R

1000 pages

⌘ Μεταφορά όλης της σχέσης (της μικρότερης;) σε έναν κόμβο: μεταφορά της R στο Λονδίνο

☑ Κόστος:  $1000 T + 4500 D$  (με ταξινόμηση/συγχώνευση; κόστος =  $3 * (500 + 1000) D$ )

☑ Αν το μέγεθος του αποτελέσματος είναι πολύ μεγάλο, μπορεί να συμφέρι να μεταφέρουμε και τις δύο σχέσεις στον κόμβο υποβολής της ερώτησης και να υπολογίσουμε τη συνένωση εκεί

LONDON

S

500 pages

PARIS

R

1000 pages

⌘ Μεταφορά όταν χρειάζεται, Εμφωλευμένος βρόγχος με τη σχέση S στον εξωτερικό βρόγχο :

☑ Κόστος:  $500 D + 500 * 1000 (D+T)$

LONDON

S

500 pages

PARIS

R

1000 pages

⌘ Μεταφορά όταν χρειάζεται, Εμφωλευμένος βρόγχος με τη σχέση S στον εξωτερικό βρόγχο :

☑ Αν η ερώτηση δεν είχε υποβληθεί στο Λονδίνο πρέπει να προσθέσουμε και το κόστος μεταφοράς του αποτελέσματος στον κόμβο που αρχικά υποβλήθηκε η ερώτηση

☑ Ο υπολογισμός μπορεί επίσης να γίνει στο Παρίσι

• Ημι-συνένωση

LONDON

S

500 pages

PARIS

R

1000 pages

• Ιδέα: Αποφυγή μεταφοράς όλης της σχέσης R στο Λονδίνο - αλλά μεταφορά μόνο των πλειάδων που συνενώνονται με πλειάδες της S

• Πως; Πρέπει να καθορίσουμε ποιες είναι αυτές οι πλειάδες

• Ημι-συνένωση

LONDON

S

500 pages

PARIS

R

1000 pages

- Στο Λονδίνο, προβολή (project) S στις στήλες του join
- Μεταφορά του αποτελέσματος στο Παρίσι
- Στο Παρίσι, συνένωση της προβολής S με τη R. Το αποτέλεσμα ονομάζεται **Ελάττωση (reduction)** της R σε σχέση με τη S.
- Μεταφορά της ελάττωσης της σχέσης R στο Λονδίνο
- Στο Λονδίνο, συνένωση S με την ελάττωση της R.

• Ημι-συνένωση

LONDON

S

500 pages

PARIS

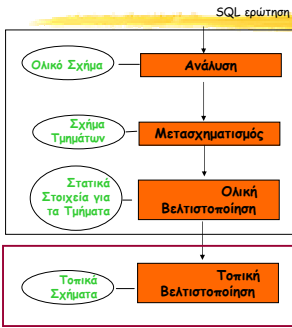
R

1000 pages

• Ιδέα: Αποφυγή μεταφοράς όλης της σχέσης R στο Λονδίνο -- με την επιβάρυνση του υπολογισμού και της μεταφοράς της προβολής της S και του υπολογισμού της συνένωσης της προβολής με τη R - κέρδος; Τότε;

• Ιδιαίτερα χρήσιμο όταν υπάρχει μια συνθήκη επιλογής στη σχέση S, και η απάντηση πρέπει να δοθεί στο Λονδίνο

## Επεξεργασία Ερωτήσεων



Κεντροκοιμημένα (σε κάποιο «ειδικό» κόμβο

Σε κάθε κόμβο

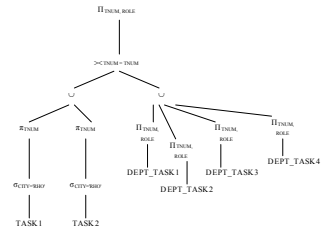
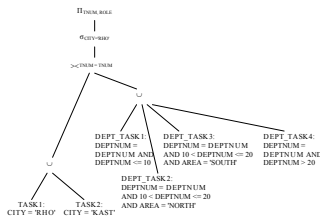
## Τοπική Βελτιστοποίηση



Πλάνο εκτέλεσης

Χρησιμοποιεί τεχνικές κεντροκοιμημένης βελτιστοποίησης για την εύρεση του καλύτερου (τοπικά) τρόπου εκτέλεσης

## Επεξεργασία Ερωτήσεων



## Επεξεργασία Ερωτήσεων

### Περίληψη Ευριστικών Κανόνων

#### Κανόνας 1

Βασίζομενοι στην επιμεριστικότητα των μοναδιαίων τελεστών σε σχέση με τους διαδοχικούς, σπρώχνουμε επιλογές και προβολές όσο πιο χαμηλά γίνεται στο δέντρο (με αποτέλεσμα να μειώνουμε γρήγορα και τοπικά το μέγεθος της εμπλεκόμενης πληροφορίας).

Για παράδειγμα, αντί για να κάνουμε την επιλογή  $\sigma_{CITY='RHO'}$  (TASK J DEPT\_TASK), προτιμήθηκε να σπρώξουμε την επιλογή μόνο στα τμήματα TASK1 και TASK2, ώστε να μειώσουμε το κόστος μεταφοράς δεδομένων.

## Επεξεργασία Ερωτήσεων

#### Κανόνας 2

Βασίζομενοι στην αντιμεταθετικότητα και τη μοναδική ποσότητα των μοναδιαίων τελεστών, παράγουμε συνδυασμούς από επιλογές και προβολές σε κάθε εμπλεκόμενη σχέση.

Στο προηγούμενο παράδειγμα, αντί να εμπλέξουμε όλο το TASK1, προτιμήθηκε να χρησιμοποιηθεί η έκφραση  $\pi_{PINUM}(\sigma_{CITY='RHO'}(TASK1))$ , η οποία σαφώς μειώνει το μέγεθος της αποτιμούμενης σχέσης.

**Κανόνας 3**

Μπορούμε να απαλείψουμε φύλλα του δέντρου (τμήματα δηλαδή των καθολικών σχέσεων), αν ο συνδυασμός των αλγεβρικών εκφράσεων που τα ορίζουν με τις αλγεβρικές εκφράσεις που τους επιβάλλονται έρχεται σε σύγκρουση.

Για παράδειγμα, η έκφραση  $\sigma_{CITY='RHO'}$  (TASK2) ισοδυναμεί με την έκφραση  $\sigma_{CITY='RHO' \wedge CITY='KAST'}$  (TASK) η οποία προφανώς περιέχει αντίφαση και δεν έχει νόημα να αποτιμηθεί.

(Αντίστοιχα, σε μια περίπτωση κάθετης κατάντησης, θα μπορούσε να συμβαίνει το ίδιο με μια προβολή).

**Κανόνας 4**

Μπορούμε να εκτελούμε τις συνενώσεις πριν από τις ενώσεις (ανεβάζουμε τις ενώσεις όσο πιο ψηλά μπορούμε στο δέντρο).

Με τον τρόπο αυτό, εκτελούμε τις συνενώσεις τοπικά, με αποτέλεσμα να μειώνουμε νωρίς την πληροφορία που διακινούμε στο δίκτυο.

**Κανόνας 5**

Όπου έχουμε αυξημένη μετα-πληροφορία, μπορούμε να τη χρησιμοποιούμε για να αποφεύγουμε άσκοπες συνδέσεις.

Για παράδειγμα, αν γνωρίζουμε ότι  $DEPTNUM < 10 \Rightarrow AREA = NORTH$ , έστω και αν αυτό δε φαίνεται στον ορισμό των τμημάτων, μπορούμε να αποφύγουμε τη συνένωση ενός τμήματος με  $AREA = SOUTH$  με κάποιο τμήμα  $DEPT\_TASK$  με συνθήκη  $DEPTNUM < 10$ .

**Κανόνας 6**

Χρησιμοποιούμε όσο το δυνατό πιο συχνά semi-joins αντί για joins, με σκοπό να μειώσουμε τη διακινούμενη πληροφορία.