



# Αποθήκες Δεδομένων και Εξόρυξη Δεδομένων

Ενότητα 4: Αποθήκες Δεδομένων – Μέρος Β'

Αναστάσιος Γούναρης, Επίκουρος Καθηγητής  
Τμήμα Πληροφορικής ΑΠΘ



# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.





# Αποθήκες Δεδομένων Μέρος Β'

Αρχιτεκτονικές συστημάτων OLAP, κυβοειδή,  
κατάλογοι και ευρετήρια



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
*επένδυση στην κοινωνία της γνώσης*

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

# Περιεχόμενα ενότητας

1. Περίληψη εισαγωγικών εννοιών – Τύποι σχημάτων.
2. Αρχιτεκτονικές υλοποίησης συστημάτων OLAP.
3. Μερική υλοποίηση κυβοειδών.
4. Επιλογή κυβοειδών για απάντηση.
5. Κατάλογοι – Ευρετήρια.
6. Γρήγορη απάντηση σε ερωτήματα.
7. Τρόποι υπολογισμού κυβοειδών.



# Σκοποί ενότητας

- Ανάλυση αρχιτεκτονικών υλοποίησης συστημάτων OLAP και υλοποίησης κύβων.
- Παρουσίαση βασικών εννοιών καταλόγων και ευρετηρίων.



# Τί είναι Αποθήκη Δεδομένων (επανάληψη)

- Έχει ορισθεί με πολλούς διαφορετικούς τρόπους αλλά όχι με την απαραίτητη σαφήνεια.
  - Μία ΒΔ για υποστήριξη λήψης αποφάσεων που διατηρείται ξεχωριστά από τις επιχειρησιακές ΒΔ.
  - Υποστήριξη επεξεργασίας πληροφοριών μέσω μίας πλατφόρμας ολοκληρωμένων, ιστορικών δεδομένων για ανάλυση.
- «Μία προσανατολισμένη προς το θέμα (subject-oriented), ολοκληρωμένη (integrated), χρονικά μεταβαλλόμενη (time-variant), και μη πτητική συλλογή δεδομένων με σκοπό την υποστήριξη λήψης αποφάσεων.» - W. H. Inmon.
- Αποθήκευση δεδομένων:
  - Η διαδικασία ανάπτυξης και χρησιμοποίησης Αποθηκών Δεδομένων.



# Διαφορές ΑΔ και ΒΔ

Χαρακτηριστικό	Σχεσιακό ΣΔΒΔ	Αποθήκη Δεδομένων
Σκοπός	«τρέξιμο» καθημερινών διεργασιών	Υποστήριξη αποφάσεων
Λειτουργία	Διεκπεραίωση συναλλαγών	Εξαγωγή πληροφορίας
Χρήστες	Κατώτεροι εργαζόμενοι, DBAs	Υψηλόβαθμα στελέχη, αναλυτές
Αριθμός Χρηστών	(μέχρι) χιλιάδες	(μέχρι) εκατοντάδες
Δεδομένα	Τρέχοντα, απομονωμένα	Ιστορικά, ολοκληρωμένα
Ενδεικτικό Μέγεθος	<100GB	10s-100s TB
Σχεδιασμός	ΟΣ – κανονικοποίηση	Μοντελ/ση διαστάσεων, αποκαν/ση
Χρήση	Επαναληπτική	Ad-hoc

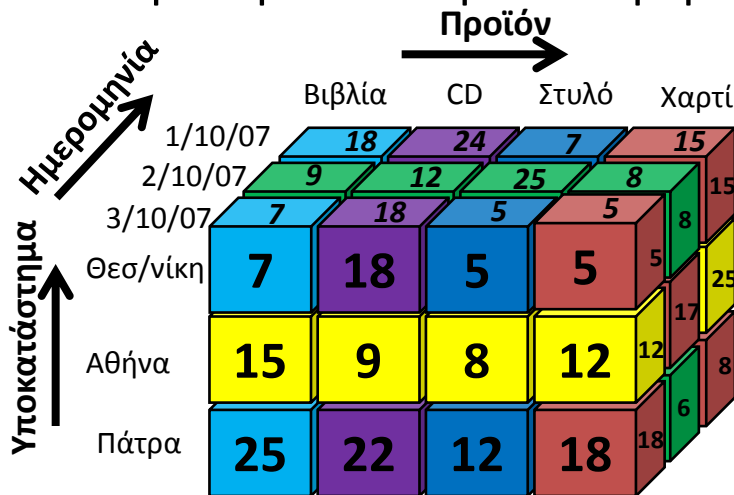
Χαρακτηριστικό	Σχεσιακό ΣΔΒΔ	Αποθήκη Δεδομένων
Προσπέλαση	Ανάγνωση/εγγραφή	(κυρίως) ανάγνωση
Ενημέρωση	Συνεχής	Περιοδική
Μονάδα εργασίας	Σύντομες, απλές συναλλαγές	Περίπλοκα ερωτήματα
Χρόνοι διεκπεραίωσης	<δευτερόλεπτα	Λεπτά-ώρες
#προσπελ. εγγραφές	10s	1000000s
Μονάδα απόδοσης	Συναλλαγές/sec	Χρόνος απόκρισης
ACID	Ναι	Όχι
Κατάλογοι	B-δένδρα	Κατάλογοι bitmap



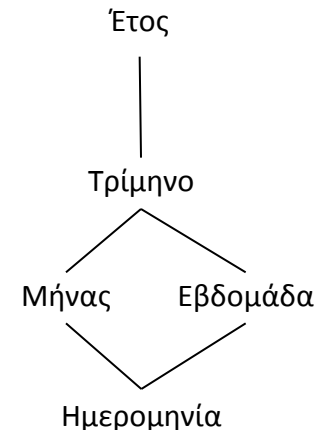
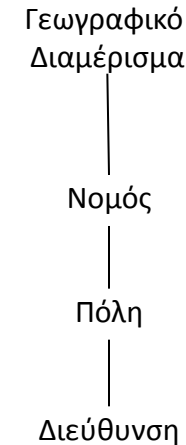
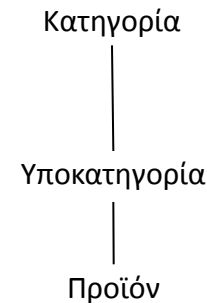


# Από τους πίνακες και τα λογιστικά φύλλα στους κύβους δεδομένων

- Μία ΑΔ βασίζεται σε ένα ΠΟΛΥΔΙΑΣΤΑΤΟ λογικό μοντέλο, με το οποίο τα δεδομένα είναι στη μορφή κύβου δεδομένων.
- Ένας κύβος δεδομένων (π.χ., πωλήσεις) επιτρέπει τη μοντελοποίηση και την παρουσίαση σε διάφορες διαστάσεις.



## Ιεραρχίες

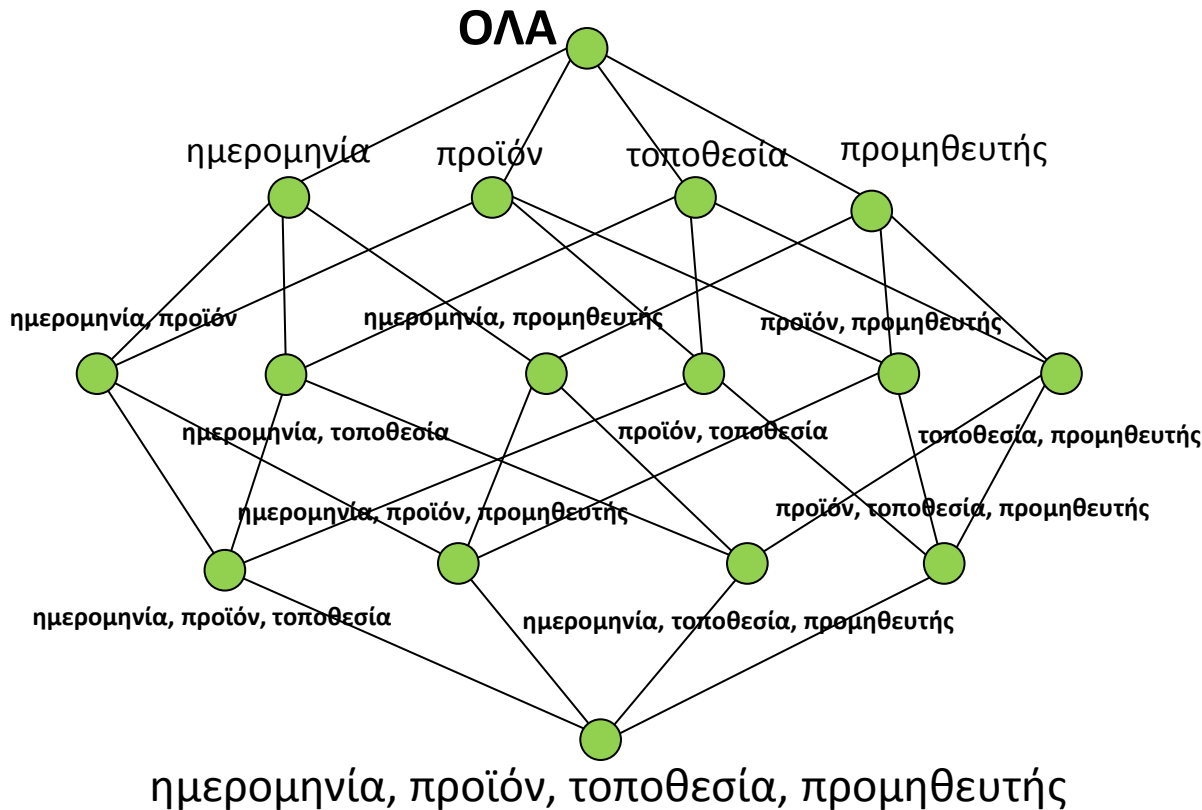


**Λειτουργίες OLAP: rollup, drill down, pivot, slice, dice**



# Κύβος Δεδομένων

- Κύβος δεδομένων: Πλέγμα Κυβοειδών (cuboids)



0-D(apex) cuboid

1-D cuboids

2-D cuboids

3-D cuboids

4-D(base) cuboid



# Μεταδεδομένα

- Τα μεταδεδομένα της ΑΔ ορίζουν τα αντικείμενα μέσα στην ΑΔ. Περιλαμβάνουν:
- Περιγραφή της δομής της ΑΔ - Σχήμα, όψεις, διαστάσεις, ιεραρχίες, τοποθεσίες αγορών δεδομένων, περιεχόμενα, ...
- Λειτουργικά μεταδεδομένα - Καταγωγή δεδομένων - data lineage: ιστορικό των δεδομένων από άλλες πηγές και των μετασχηματισμών, ενημερότητα (active, archived, or purged), παρακολούθηση (στατιστικά χρήσης, αναφορές λαθών, έλεγχοι).
- Αλγόριθμοι περίληψης.
- Η σύνδεση του επιχειρησιακού περιβάλλοντος με την ΑΔ.
- Δεδομένα σχετικά με την απόδοση, με την κυριότητα, την πολιτική τιμολόγησης, κλπ.



# Τρεις τύποι σχημάτων

- Σχήμα αστέρα: Ένας πίνακας γεγονότων στη μέση που συνδέεται με πολλούς πίνακες διαστάσεων.
- Σχήμα νιφάδας: Τροποποίηση του σχήματος αστέρα με την έννοια ότι κάποιοι πίνακες διαστάσεων είναι κανονικοποιημένοι, σχηματίζοντας σχήμα χιονο-νιφάδας.
- Αστερισμοί γεγονότων: Πολλαπλοί πίνακες γεγονότων μοιράζονται πίνακες διαστάσεων.



# Αρχιτεκτονικές υλοποίησης συστημάτων OLAP - ROLAP

- **Relational OLAP (ROLAP)**
  - Χρήση σχεσιακού (ή εκτεταμένου σχεσιακού) ΣΔΒΔ για την αποθήκευση και διαχείριση του κύβου + OLAP middleware.
  - 1 πίνακας γεγονότων + ξεχωριστοί πίνακες για διαστάσεις.
  - Βελτιστοποιημένο backend ΣΔΒΔ, μη απαραίτητη η δημιουργία κύβου, ύπαρξη προτύπων.
  - Πολύ καλή κλιμάκωση, ενδεχομένως μειωμένη ταχύτητα.



# Αρχιτεκτονικές υλοποίησης συστημάτων OLAP - MOLAP

- **Multidimensional OLAP (MOLAP)**
  - Ο πίνακας γεγονότων αποθηκεύεται σε (αραιούς) πολυδιάστατους πίνακες που έχουν την ίδια έννοια με τους πίνακες π.χ., στη C.
  - Πολύ γρήγορη εκτέλεση πράξεων/άμεση πρόσβαση στα κελιά.
  - Απαιτείται συμπίεση, μεγάλος χρόνος δημιουργίας κύβου, δεν υπάρχουν πρότυπα.



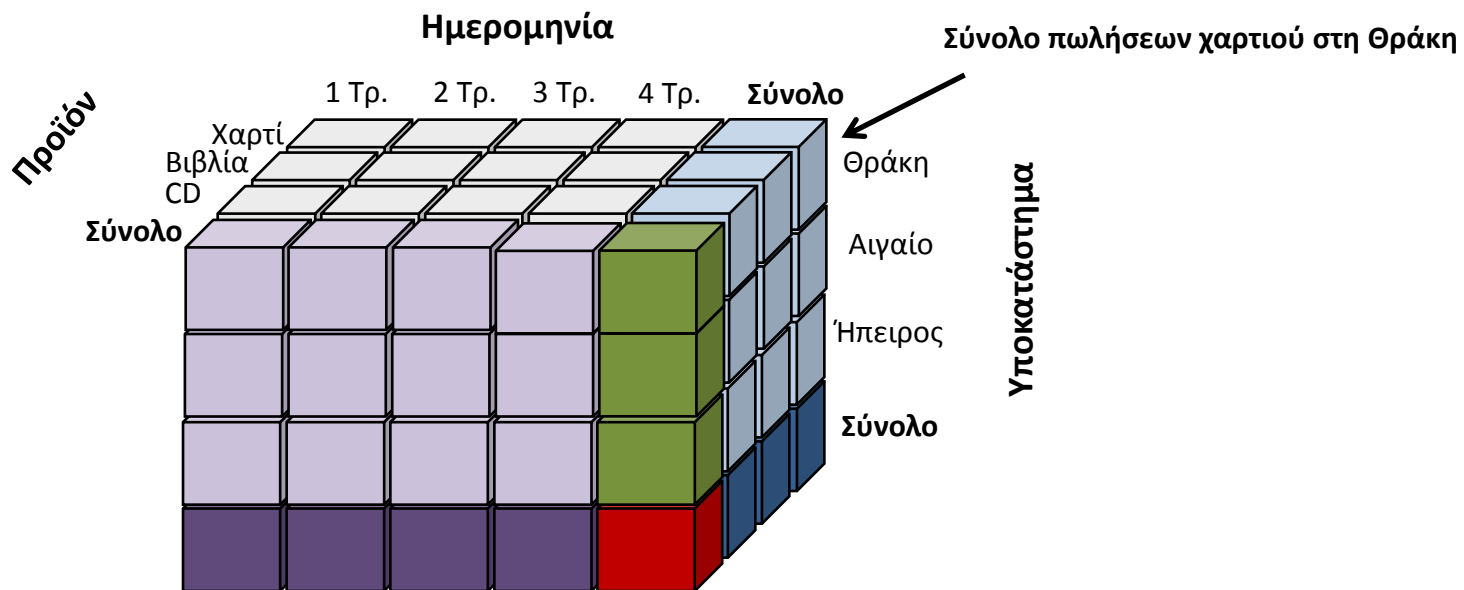
# Αρχιτεκτονικές υλοποίησης συστημάτων OLAP - HOLAP

- **Hybrid OLAP (HOLAP)** (e.g., Microsoft SQLServer)
  - Μεγαλύτερη ευελιξία:
    - Χαμηλό επίπεδο: relational.
    - Υψηλότερο επίπεδο: array.
  - Συνδυασμός γρήγορης εκτέλεσης και κλιμάκωσης.



# Υλοποίηση κυβοειδών

- Αν  $k$  διαστάσεις και δεν υπάρχουν ιεραρχίες, τότε υπάρχουν συνολικά  $2^k$  κυβοειδή (που αλλιώς ονομάζονται και όψεις).
- Αν κάθε διάσταση έχει  $L_i$  επίπεδα ιεραρχίας, τότε  $\prod(L_i + 1)$ ,  $i=1..k$
- Η υλοποίηση μπορεί να γίνει είτε σε ROLAP (π.χ., πίνακες υλοποιημένων όψεων), είτε σε MOLAP, με επέκταση του βασικού κύβου.



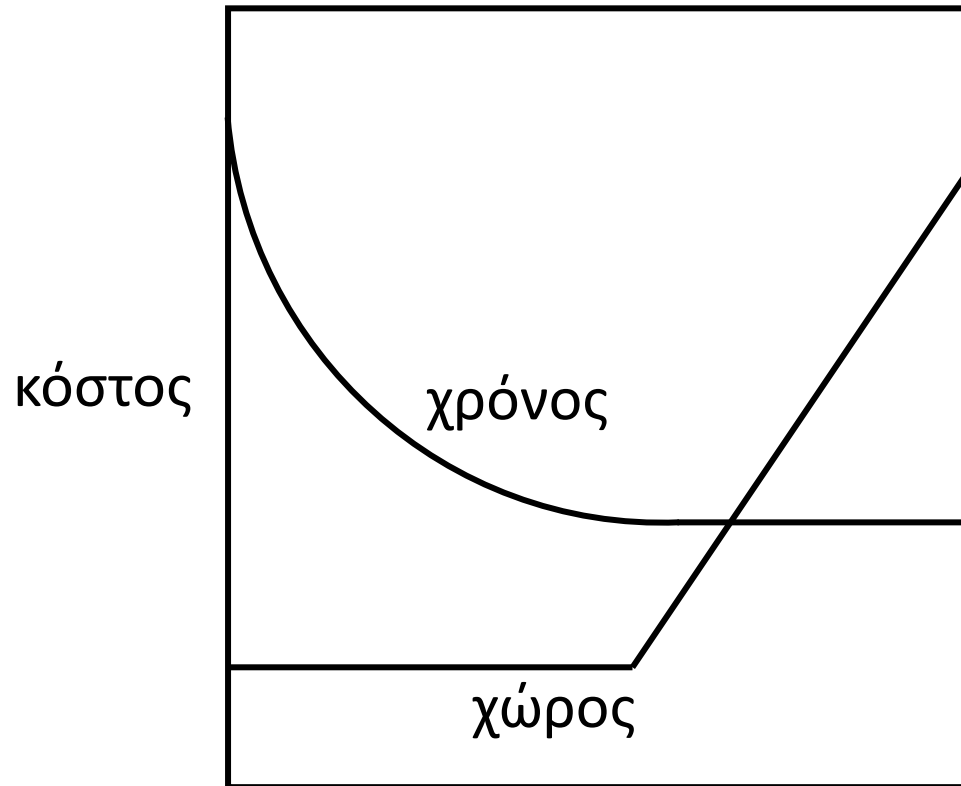


# Επιλογές υλοποίησης

- Φυσική υλοποίηση όλου του κύβου.
  - Καλύτερος χρόνος απόκρισης. Αλλά όχι ρεαλιστική λύση για μεγάλους κύβους λόγω των πολύ υψηλού κόστους χώρου, δημιουργίας κύβου, συντήρησης.
- Καθόλου υλοποίηση.
  - Για κάθε ερώτημα, προσπέλαση στα βασικά (raw) δεδομένα για τον υπολογισμό κάθε κελί (cell). Ισχυρός αντίκτυπος στο χρόνο απόκρισης, δίνει μεγάλη σημασία στο σύστημα που βασίζεται (RDBMS σε ROLAP).
- Υλοποίηση μέρους του κύβου.
  - Οι τιμές πολλών κελιών είναι υπολογίσιμες από άλλα κελιά του κύβου (**dependent cells**) Επίσης, ο αριθμός των υλοποιημένων κελιών εξαρτάται από τους περιορισμούς χώρου.



# Πλεονέκτημα μερικής υλοποίησης



#υλοποιημένες όψεις  
(materialized views)



# Περιορισμοί χώρου - ζητήματα

- Πόσα κελιά πρέπει να υλοποιηθούν για «καλή» απόδοση στην επεξεργασία ερωτημάτων;
- Δεδομένου συγκεκριμένου χώρου, ποιες όψεις πρέπει να υλοποιηθούν ώστε να ελαχιστοποιηθεί το μέσο κόστος ερωτημάτων;
- Αν ανεχόμαστε  $X\%$  υποβάθμιση του μέσου κόστους ερωτημάτων, πόσο χώρο μπορούμε να γλιτώσουμε;
  - Σύγκριση με την περίπτωση ολικής υλοποίησης.

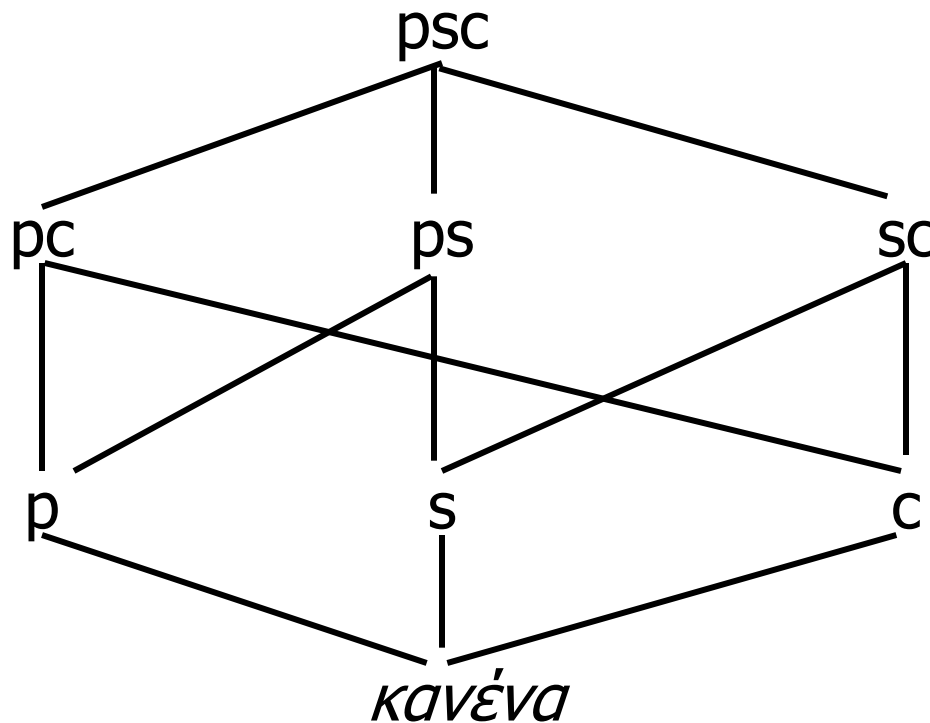


# lattice (δικτυωτό) πλαίσιο

- Το  $V1$  εξαρτάται από το  $V2$  ( $V1 \preceq V2$ ) αν και μόνο αν το  $V1$  μπορεί να απαντηθεί με αποτελέσματα του  $V2$ .
  - $(part) \preceq (part, cust)$
- Ο τελεστής  $\preceq$  καθορίζει μερική ταξινόμηση (partial order).
- Μερική ταξινόμηση + top element (από το οποίο όλα εξαρτώνται)  $\Rightarrow$  δίκτυο (lattice).
- Σχέσεις:
  - Ο  $b$  είναι πρόγονος του  $a$ :  $ancestor(a) = \{b \mid a \preceq b\}$
  - Ο  $b$  είναι απόγονος του  $a$ :  $descendent(a) = \{b \mid b \preceq a\}$
  - Ο  $b$  είναι ο επόμενος κόμβος του  $a$ :  
 $next(a) = \{b \mid a < b, \nexists c: a < c \wedge c < b\}$



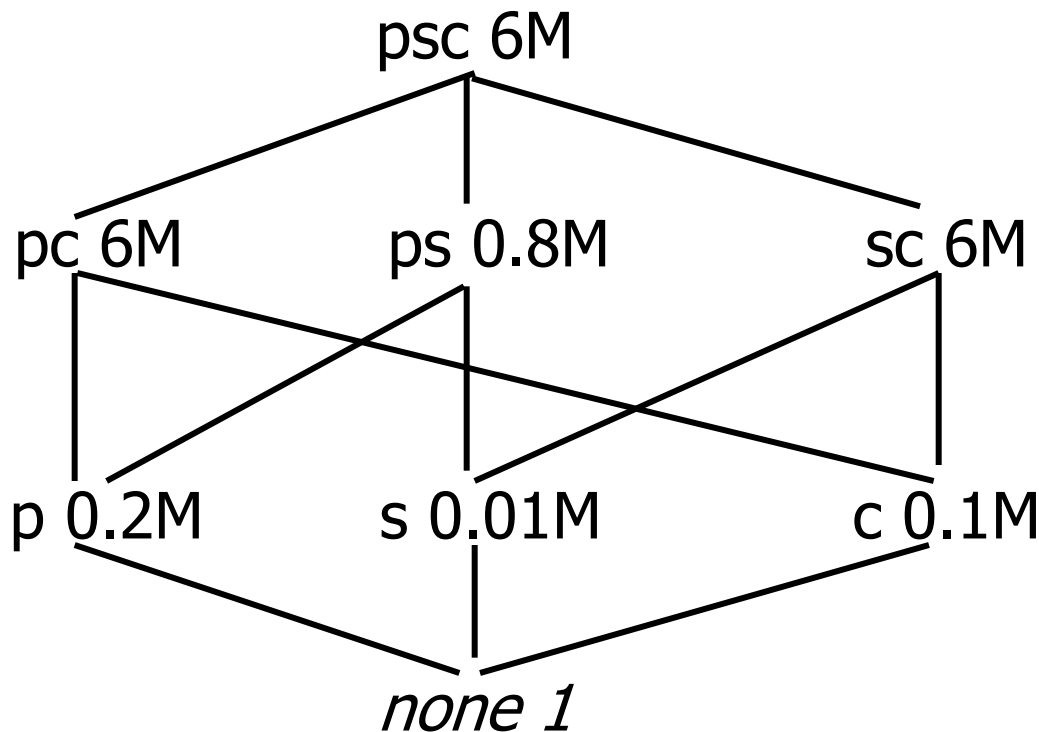
# Παράδειγμα



p: part  
s: supp  
c: cust



# Κέρδος από μερική υλοποίηση



- Το κόστος είναι ανάλογο του αριθμού των γραμμών της όψης.
  - επόμενη διαφάνεια.
- Αν έχω υλοποιήσει το **psc**
  - Δεν χρειάζεται να υλοποιηθούν τα **pc**, **sc**.
  - Το **psc** περιέχει τις απαντήσεις για **pc**, **sc** με το ίδιο κόστος (6M).
  - Γλιτώνει κανείς 12M και χρειάζεται μόνο 7.11M.
- Το βασικό κυβοειδές υλοποιείται πάντα.



# Γραμμικό μοντέλο κόστους (1)

- Αν η απάντηση για  $V$  είναι μέσω ενός προγόνου  $V_A$ , τότε το κόστος της απάντησης είναι: #γραμμές του  $V_A$  -- μέση περίπτωση.
  - Δηλ. το κόστος ισούται με σάρωση της όψης του προγόνου.
- Πολλά ερωτήματα απαιτούν πολύ λιγότερο από σάρωση ολόκληρης υλοποιημένης όψης.
  - Π.χ., πωλήσεις ενός συγκεκριμένου προϊόντος.
  - Αν είναι υλοποιημένο το συγκεκριμένο κυβοειδές και υπάρχει index, τότε το κόστος είναι  $O(1)$ .



# Γραμμικό μοντέλο κόστους (2)

- Πολλά ερωτήματα όμως απαιτούν πάνω από μία σάρωση υλοποιημένης όψης.
  - Π.χ., από την υλοποιημένη όψη (part, supplier), βρες όλες τις πωλήσεις για κάθε κομμάτι => ανάγκη για συναθροιστική συνάρτηση μέσα στην όψη.
    - Καλύτερη περίπτωση: ένα πέρασμα και δημιουργία hash table που χωράει στην μνήμη.
    - Χειρότερη περίπτωση: πολλαπλές σαρώσεις και υπολογισμός ξεχωριστά σε κάθε κομμάτι.
    - Πρακτικά απαιτούνται 1-2 σαρώσεις.





# Υπολογισμός μεγέθους

- Ένα γραμμικό μοντέλο απαιτεί γνώση του μεγέθους κάθε όψης.
- Δεν υλοποιούμε τα πάντα, οπότε πως θα γνωρίζουμε το μέγεθος;
  - Χρησιμοποιούμε ένα μικρό δείγμα, υλοποιούμε όλες τις όψεις σε αυτό το δείγμα και υπολογίζουμε έτσι το πραγματικό μέγεθος.
  - Για στατιστικά ανεξάρτητες όψεις, ο υπολογισμός πραγματοποιείται αναλυτικά:  
$$\#rows \text{ in } (part, supp) = \#distinct \text{ values in } (part) \times \#distinct \text{ values in } (supp)$$
- Πώς βρίσκουμε τον αριθμό των διαφορετικών τιμών σε κάποιο χαρακτηριστικό;
  - Haas et al: Sampling-based estimation of the number of distinct values of an attribute, VLDB 1995.



# Αλγόριθμος βελτιστοποίησης

- Απλοποιημένο πρόβλημα βελτιστοποίησης.
  - Ελαχιστοποίηση: μέσος χρόνος για επεξεργασία μιας όψης.
  - Περιορισμός: υλοποίηση αναγκαστικά  $k$  όψεων (ανεξαρτήτου χώρου που απαιτούν).
- NP-complete: ουσιαστικά είναι Set-Cover.
- Προσεγγιστική μέθοδος: άπληστος (greedy) αλγόριθμος.
  - Η καλύτερη λύση βρίσκεται βάσει προηγούμενων αποφάσεων.



# Ο άπληστος αλγόριθμος - Εισαγωγή

- Έστω  $C(v)$ , το κόστος προσπέλασης μίας όψης  $v$
- Υπολογισμός του κέρδους (benefit) του  $v$  ....
- λαμβάνοντας υπόψη πως μπορεί να βελτιώσει η υλοποίηση της όψης τον υπολογισμό άλλων όψεων και την ίδια την όψη.
- Από ένα επιλεγμένο σύνολο όψεων  $S$  (μαζί με το βασικό κυβοειδές) που δεν περιλαμβάνει το  $v$ , το κέρδος  $B(v, S)$  ορίζεται ως εξής:
  - Για κάθε  $w \preceq v$ ,
    - επέλεξε  $u$  ώστε  $C(u) = \min\{C(z) \mid z \in S, w \preceq z\}$
    - If  $C(v) < C(u)$ , then  $B_w = C(u) - C(v)$ ; otherwise  $B_w = 0$
  - $B(v, S) = \sum_{w \preceq v} B_w$



# Ο άπληστος αλγόριθμος

- Ουσιαστικά, για κάθε  $w$  που είναι απόγονος του  $v$ , συγκρίνουμε το κόστος υπολογισμού του  $w$  χρησιμοποιώντας
  - (1) το  $v$  και
  - (2) την όψη στο  $S$  που παρέχει τον πιο αποδοτικό τρόπο υπολογισμού του  $w$ .
- Αν το  $v$  μπορεί να αυξήσει την απόδοση, τότε η διαφορά είναι το κέρδος.
  - Δεν υπάρχουν αρνητικά κέρδη.

$S = \{\text{top view of lattice, i.e., basic cuboid}\}$

**for**  $i = 1$  **to**  $k$  **do** //  $k$ : αριθμός υλοπ. όψεων

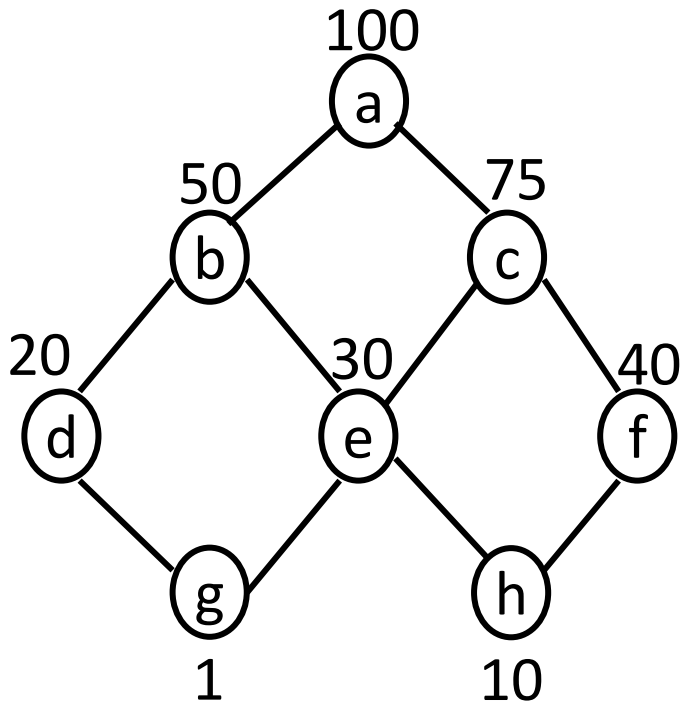
    select view  $v \notin S$  with  $\max B(v, S)$

$S = S \cup \{v\}$

**return**  $S$



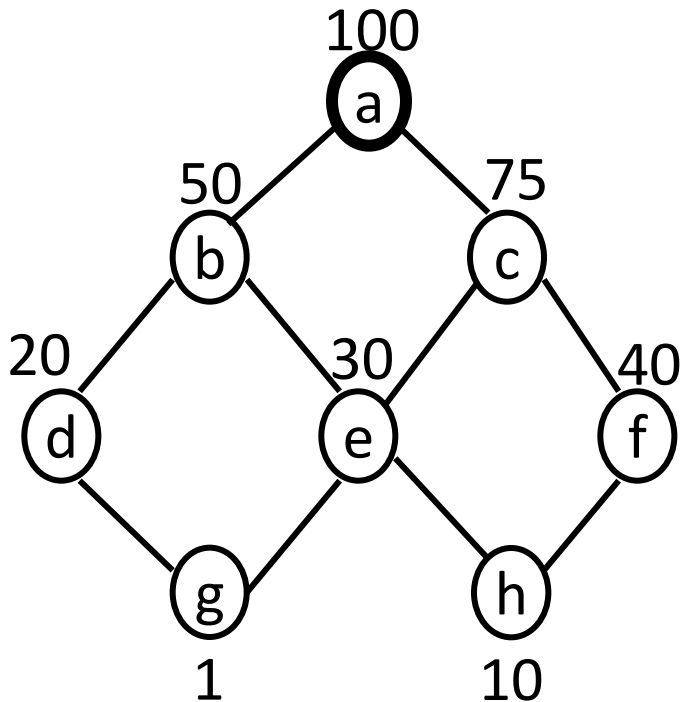
# Παράδειγμα εφαρμογής (1)



- Αρχικά,  $S = \{a\}$
- $k+1 = 4$  (δηλ., πρέπει να 3 επιπλέον)



# Παράδειγμα εφαρμογής (2)

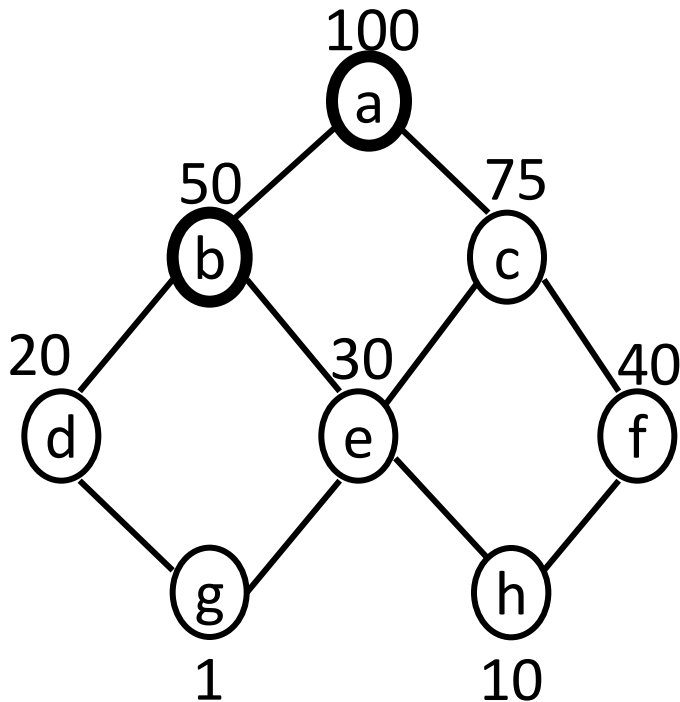


Αρχική επιλογή

- $b: 50 \times 5 = 250$
- $c: 25 \times 5 = 125$
- $d: 80 \times 2 = 160$
- $e: 70 \times 3 = 210$
- $f: 60 \times 2 = 120$
- $g: 99 \times 1 = 99$
- $h: 90 \times 1 = 90$



# Παράδειγμα εφαρμογής (3)

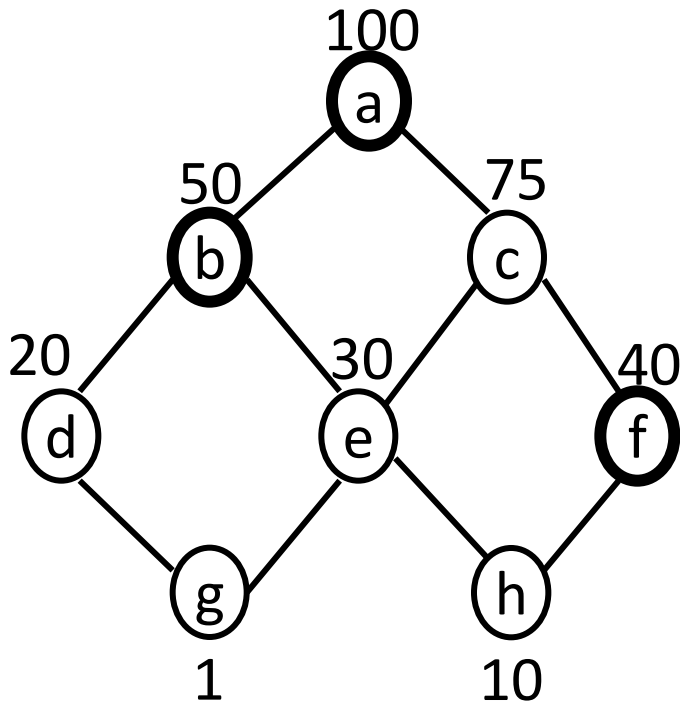


Second choice

- $c: 25 \times 2 = 50$
- $d: 30 \times 2 = 60$
- $e: 20 \times 3 = 60$
- $f: 60 + 10 = \mathbf{70}$
- $g: 49 \times 1 = 49$
- $h: 40 \times 1 = 40$



# Παράδειγμα εφαρμογής (4)



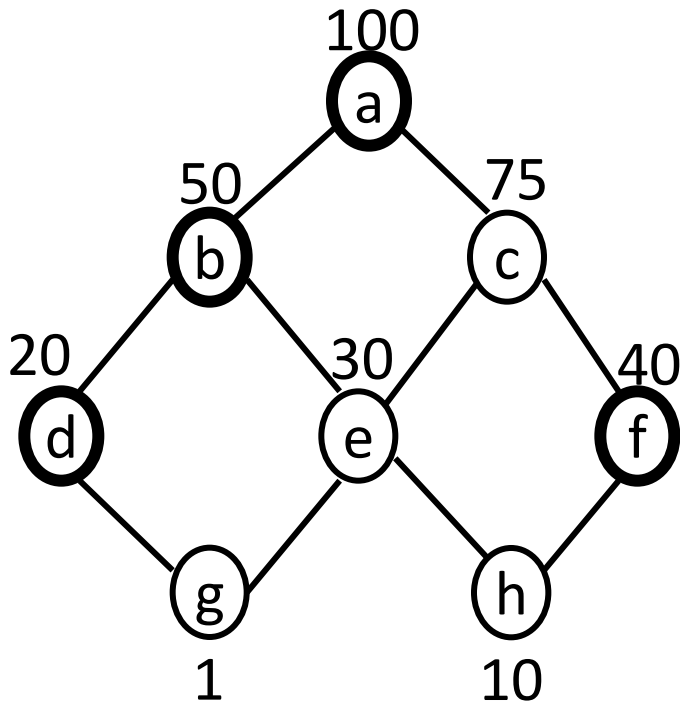
Third choice

- c:  $25 \times 1 = 25$
- d:  $30 \times 2 = \mathbf{60}$
- e:  $20 \times 2 + 10 = 50$
- g:  $49 \times 1 = 49$
- h:  $30 \times 1 = 30$





# Παράδειγμα εφαρμογής (5)



- Αν μόνο το a υλοποιούταν, το κόστος θα ήταν  $8 \times 100 = 800$ .
- Τώρα το κόστος είναι  $800 - 250 - 70 - 60$ , δηλ. 420.
- Τυγχάνει το 420 να είναι και η βέλτιστη λύση.
- Εγγύηση απόδοσης:

$$1 - \left( \frac{k-1}{k} \right)^k \geq 0.63$$



# Επιλογή κυβοειδών για απάντηση ερωτημάτων (1)

- Έστω ότι έχουμε να επεξεργαστούμε μία ερώτηση πάνω στο {brand, province\_or\_state} με συνθήκη “year =2004”.
- Υπάρχουν διαθέσιμα 4 υλοποιημένα κυβοειδή:
- 1) {year, item\_name, city}
  - ίδιο σύνολο διαστάσεων: brand  $\preceq$  item\_name(Δεκτό)
  - Μεγαλύτερο κόστος: brand  $\preceq$  item\_name, province\_or\_state  $\preceq$  city
- 2) {year, brand, country}
  - Country  $\preceq$  province\_or\_state (δεν μπορεί να επιλεχθεί)



# Επιλογή κυβοειδών για απάντηση ερωτημάτων (2)

- 3) {year, brand, province\_or\_state}
  - Δεκτό
- 4) {item\_name, province\_or\_state} where year = 2004
  - Δεκτό
- Εάν δεν υπάρχουν πολλές τιμες year που να σχετίζονται με το Items στο κύβο, κόστος\_cuboid3 < κόστος\_cuboid4.
- Εάν υπάρχουν αποδοτικά ευρετήρια για το cuboid4 τότε μπορεί να είναι καλύτερο από cuboid3.



# Πότε ο greedy αλγόριθμος δεν αποδίδει καλά;

- αλλά και πότε αποδίδει;
- *V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. In Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data, pages 205-216, Montreal, Canada, June 1996.*



# Ανάγκη για νέους τύπους καταλόγων

- Αναφερόμαστε σε συστήματα ROLAP.
- Ανάγκη για γρήγορη εκτέλεση ερωτημάτων σε ΑΔ:
  - Οι ΑΔ είναι πολύ μεγάλες, και οι τυπικοί τύποι καταλόγων σε ΣΔΒΔ δεν είναι πλέον αποδοτικοί.
  - Το μεγαλύτερο ποσοστό ερωτημάτων περιέχουν ανάκτηση πολλών δεδομένων, ενώ οι τυπικοί κατάλογοι επικεντρώνονται σε σενάρια με μικρό όγκο δεδομένων ανάγνωσης και ενημερώσεις.
  - Πιο περίπλοκα ερωτήματα + συνθήκες.
- Ανάγκη για περισσότερους από 1 κατάλογο ανά στήλη.
- Τα ερωτήματα περιλαμβάνουν πολλές συνδέσεις.
  - Συνδέσεις μεταξύ του πίνακα γεγονότων και των πινάκων διαστάσεων.

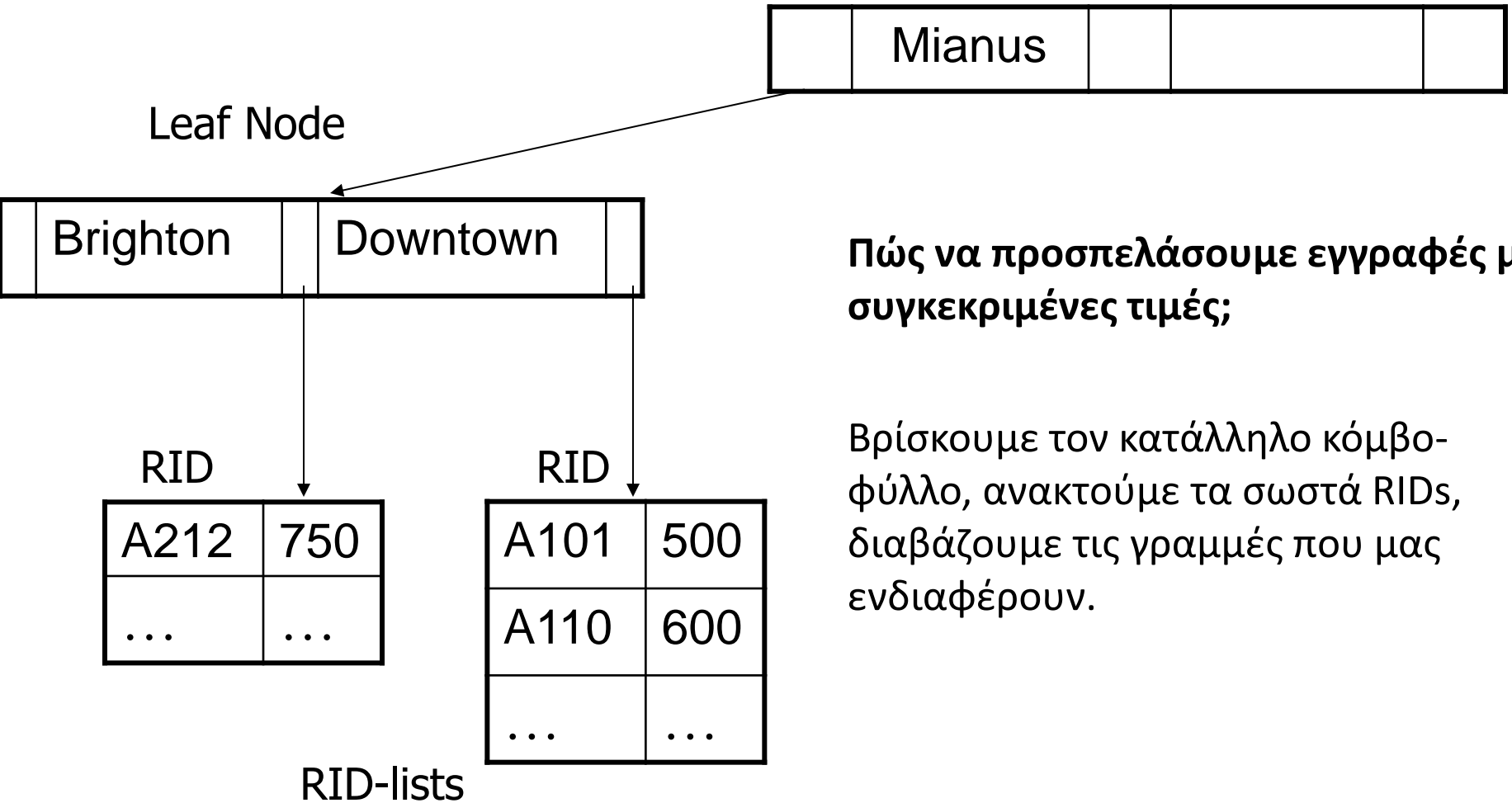


# Κατάλογοι-Ευρετήρια vs Υλοποίηση

- Η υλοποίηση κυβοειδών επιτρέπει τη γρήγορη απάντηση ερωτημάτων.
  - Υπάρχει όμως όριο στον αριθμό των διαστάσεων που μπορούν να αναπαρασταθούν σε προ-υπολογισμένους πίνακες.
- Σε ερωτήματα με πολλά κριτήρια επιλογής, πρέπει να προσπελάσουμε τα βασικά δεδομένα πολλές φορές.
- Ειδικοί κατάλογοι βοηθούν σημαντικά στην αποδοτική προσπέλαση βασικών δεδομένων, όταν αυτό χρειάζεται.



# B+ δένδρα



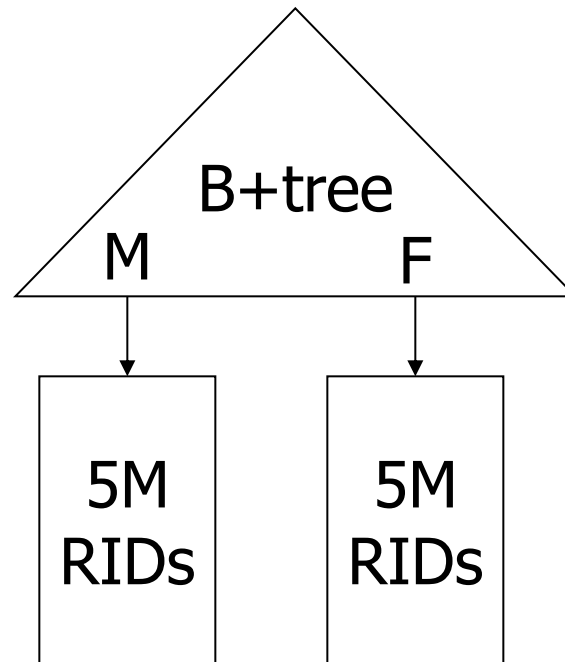
Πώς να προσπελάσουμε εγγραφές με συγκεκριμένες τιμές;

Βρίσκουμε τον κατάλληλο κόμβο-φύλλο, ανακτούμε τα σωστά RIDs, διαβάζουμε τις γραμμές που μας ενδιαφέρουν.



# Πρόβλημα με B+ δένδρα

- Αν μια στήλη έχει μικρό αριθμό δυνατών τιμών, τότε υπάρχουν λίγες και πολύ μεγάλες λίστες με RIDs.
- Παράδειγμα: Πίνακας με 10,000,000 εγγραφές, χαρακτηριστικό “Gender” με δύο δυνατές τιμές: {M, F}.



Επιλογή των ‘M’ γραμμών:  
προσπέλαση σε 5M RIDs =  
 $5M * 4 \text{ bytes} \sim 20 \text{ MB}$ .





# Bitmap Κατάλογοι

- Ένα bitmap ανά στήλη ανά τιμή: πίνακας με bits. Το  $i$ -ο bit είναι 1 αν η  $i$ -η εγγραφή έχει την αντίστοιχη τιμή.
- Bitmap κατάλογος: αναφέρεται σε επίπεδο ιδιότητας και αποτελείται από ένα bitmap για κάθε τιμή.
- Εύκολη συμπίεση και γρήγορη εκτέλεση δυαδικών πράξεων.

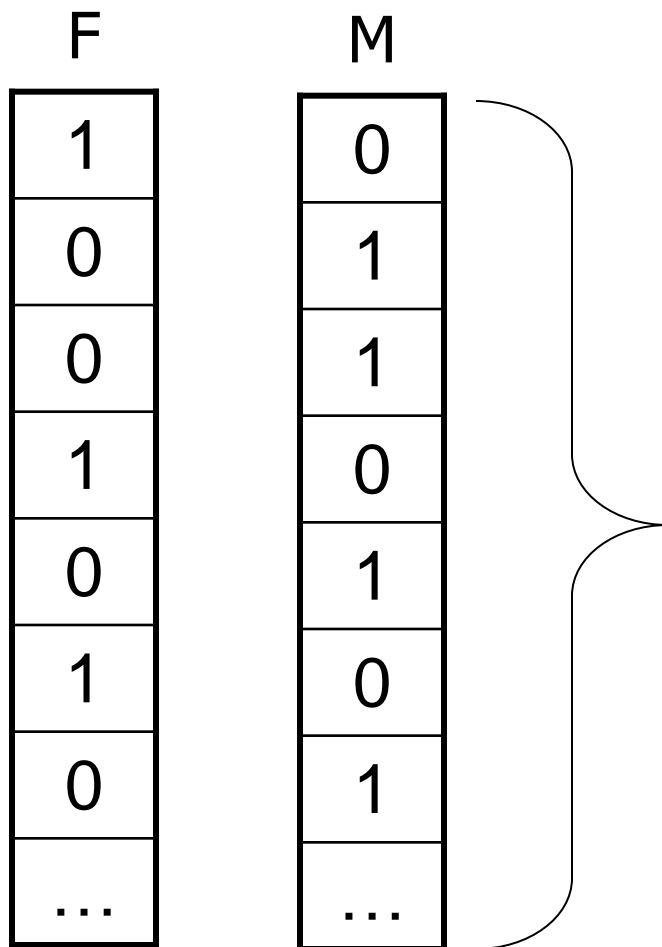
Cust	Region	Rating
C1	N	H
C2	S	M
C3	W	L
C4	W	H
C5	S	L
C6	W	L
C7	N	H

RID	N	S	E	W
1	1	0	0	0
2	0	1	0	0
3	0	0	0	1
4	0	0	0	1
5	0	1	0	0
6	0	0	0	1
7	1	0	0	0

RID	H	M	L
1	1	0	0
2	0	1	0
3	0	0	1
4	1	0	0
5	0	0	1
6	0	0	1
7	1	0	0



# Κατάλογοι φύλου με bitmaps



Πίσω στο παράδειγμα της δ. 38  
Η επιλογή των 'M' γραμμών  
προσπελαύνει μόνο 10M bits ~  
1.25 MB



# Πλεονεκτήματα καταλόγων bitmap

- Καλύτερη χωρική αποδοτικότητα απ' ότι λίστες με RID σε κατάλογο όπως π.χ., τα B+ δένδρα.
  - Όχι συμπίεση:
    - $|RID|=32\text{bits}$ ,  $\#\text{row}=n$ ,  $\#\text{distinct value}=m$
    - Αν  $m < 32$ ,  $m*n < 32*n$
  - Συμπίεση: Run-length encoding (εύκολα).
  - Αποδοτικότητα χώρου => αποδοτικότητα I/O
- Κάθε bitmap είναι μικρό και αυτά που χρησιμοποιούνται συχνά μπορεί να παραμείνουν cached στην κύρια μνήμη.



# Πλεονεκτήματα καταλόγων bitmap

- Πολύ μικρό κόστος CPU: οι πράξεις απλοποιούνται σε δυαδικές (σε επίπεδο bit) αυξάνοντας πολύ σημαντικά την ταχύτητα εκτέλεσης.
  - Τελεστές Boolean
    - Select Region From CustInfo  
Where Region='N' AND Rating='H'
    - RegionIndex(N) & RatingIndex(H) = 100001 & 1001001 = 1000001
  - Count
    - Select count(\*) From CustInfo  
Where Rating='H'
    - Count bits in RatingIndex(H) = 3
- Υποστηρίζονται από πολλά διαδεδομένα ΣΔΒΔ
  - Π.χ., Model 204, TargetIndex (Redbrick), IQ (Sybase), Oracle.



# Bit-Sliced Κατάλογοι

- Οι Bit-sliced κατάλογοι
  - Γενικά χρησιμοποιούνται σε ιδιότητες μετρήσεων...
  - ...με τιμές θετικούς ακεραίους.
    - Ή δεκαδικούς συγκεκριμένης ακρίβειας.
    - Παράδειγμα: Μετατροπή του \$5.67 σε 567 cents.
- Bit-sliced κατάλογος στο χαρακτηριστικό A.
  - Θεώρηση του A ως πολλαπλές στήλες με δυαδικές τιμές.
    - Στήλη A1 = λιγότερο σημαντικό bit του A.
    - Στήλη A2 = 2ο λιγότερο σημαντικό bit του A.
    - Κ.ο.κ.
  - Αποθήκευση κάθε στήλης ως ξεχωριστό bitmap.



# Παράδειγμα Bit-Sliced Καταλόγου

Amount
5
13
2
6
7

Binary
0101
1101
0010
0110
0111

## Bit-Sliced Index

B4: 01000

B3: 11011

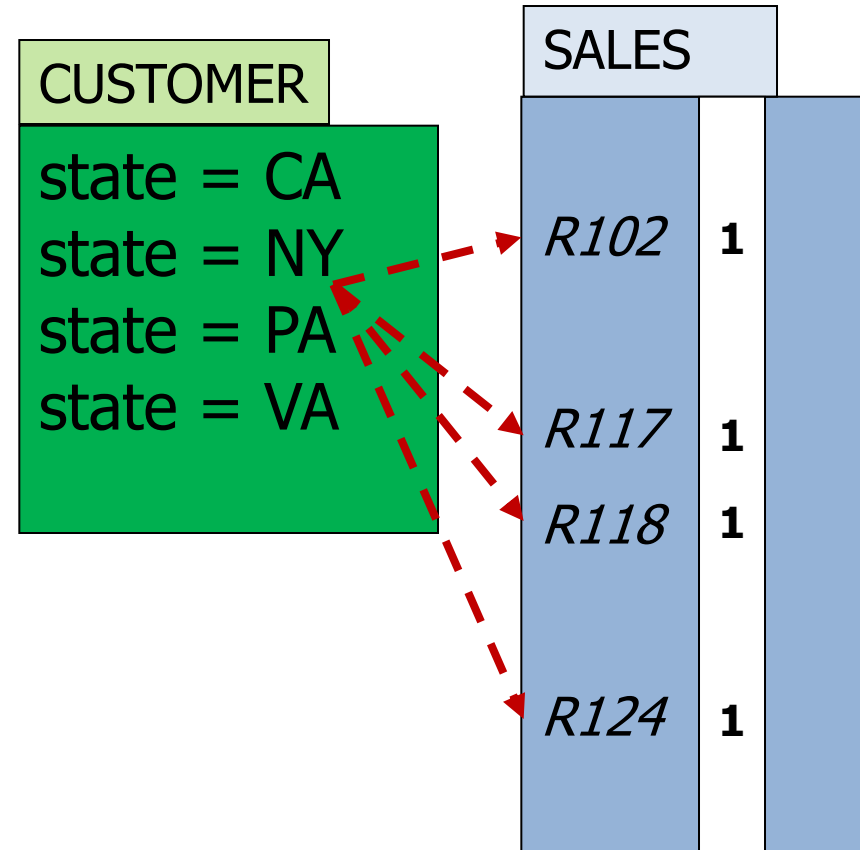
**B2: 00111**

B1: 11001



# Κατάλογοι Σύνδεσης

- Αποφυγή πολλαπλών συνδέσεων.
- Ο κατάλογος σύνδεσης συσχετίζει τις τιμές των πινάκων διαστάσεων σε σχήμα αστέρα με τις εγγραφές στον πίνακα γεγονότων.
  - Ένας κατάλογος σύνδεσης στον πίνακα *customer* διατηρεί για κάθε πολιτεία μία λίστα με τα αντίστοιχα RIDs στον πίνακα γεγονότων.
- Όταν χρησιμοποιούνται απλά RIDs, ο κατάλογος ονομάζεται Κατάλογος Σύνδεσης Αστέρα.
- Μπορεί να δημιουργείται πρόβλημα με το χωρικό κόστος.



# Bitmap Κατάλογοι σύνδεσης (1)


Αναφέρονται σε πολλούς πίνακες και αυξάνουν την απόδοση ερωτημάτων σύνδεσης.

## Sales

Cid	Pid	Dollar_sales	Unit
0100	1	1200	1
0101	3	2600	2
0110	2	1600	1
0111	1	1200	1

Pid	Brand	P_type	Size
1	Dell	P1	14
2	HP	P2	13
3	Sony	P1	15
4	Dell	P1	14
5	HP	P2	14
6	IBM	P3	12
7	HP	P2	12

## Product



Dell	HP	Sony	IBM
1	0	0	0
0	0	1	0
0	1	0	0
1	0	0	0

**Bitmap join index για το «brand»**





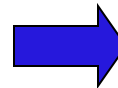
# Bitmap Κατάλογοι σύνδεσης (2)

## Customer

Cid	State
0100	CA
0101	NY
0110	CA
0111	PA

## Sales

Cid	Pid	Dollar_sales	Unit
0100	1	1200	1
0101	3	2600	2
0110	2	1600	1
0111	1	1200	1



CA	NY	PA
1	0	0
0	1	0
1	0	0
0	0	1

**Bitmap join  
index για το  
«state»**

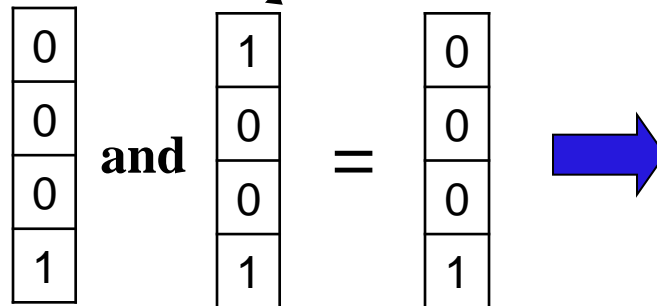


# Χρησιμοποίηση bitmap καταλόγων σύνδεσης

```
Select Sum(Dollar_sales)
From Sales S Natural Join
Product P Natural Join
Customer C
Where P.Brand='Dell'
AND C.State='PA'
```

CA	NY	PA
1	0	0
0	1	0
1	0	0
0	0	1

Dell	HP	Sony	IBM
1	0	0	0
0	0	1	0
0	1	0	0
1	0	0	0



Cid	Pid	Dollar_sales	Unit
0100	1	1200	1
0101	3	2600	2
0110	2	1600	1
<b>0111</b>	<b>1</b>	<b>1200</b>	<b>1</b>



# Απάντηση ερωτημάτων γρήγορα

- Σύγχρονη τάση (επηρεασμένη από τις δημοφιλείς μεθόδους έρευνας στο διαδίκτυο): η έμφαση είναι σε απάντηση ερωτημάτων όπου ο χρήστης ενδιαφέρεται για τα πρώτα «καλύτερα» αποτελέσματα και θέλει απάντηση όσο το δυνατόν γρηγορότερα.
  - Διαισθητικά: αν οι χρήστες δεν βρουν αυτό που ψάχνουν στις 1-2 πρώτες σελίδες, ξαναυποβάλλουν τροποποιημένο ερώτημα.
- Συνυφασμένο με το προηγούμενο είναι και η προσεγγιστική απάντηση όσο το δυνατόν γρηγορότερα, και κατόπιν συνεχής βελτίωση της ακρίβειας των αποτελεσμάτων.



# Top N Ερωτήματα

- Έστω ότι θέλουμε να βρούμε τα 10 φθηνότερα αυτοκίνητα.
- Δεν είναι αποδοτικό να βρούμε το κόστος όλων των αυτοκινήτων και να τα ταξινομήσουμε.
  - Βασική Ιδέα: εύρεση ενός κόστους  $c$ , τέτοιου που τα 10 φθηνότερα αυτοκίνητα είναι φθηνότερα από  $c$ , αλλά αυτά που κοστίζουν φθηνότερα από  $c$  δεν είναι πολύ περισσότερα από 10. Έτσι, η συνθήκη επιλογής « $cost < c$ » μπορεί να προστεθεί στο ερώτημα.
    - Αν η αρχική επιλογή του  $c$  είναι σωστή, τότε όλα OK.
    - Αν δεν είναι, πρέπει να ξαναγίνει εκτίμηση του  $c$ .



# Τορ Ν Ερωτήματα – Επιλογή του $c$

- Πώς γίνεται η επιλογή του  $c$ ?
  - Αν γνωρίζουμε την κατανομή (π.χ., με ιστογράμματα), τότε μπορούμε να επιλέξουμε τιμή με σχετική ασφάλεια.
  - Αλλά τα ιστογράμματα είναι προσεγγιστικά.
  - Η ύπαρξη άλλων συνθηκών επιλογής στο ερώτημα περιπλέκει την επιλογή.



# Άμεση (online) συνάθροιση

- Έστω ένα συναθροιστικό ερώτημα, π.χ., εύρεση του μ.ο. πωλήσεων σε μία περιοχή.
- Παροχή στον χρήστη προσέγγισης της απάντησης πριν δοθεί η τελική ακριβής απάντηση.
  - Η τρέχουσα τιμή του μ.ο. δίνεται.
  - Συνεχώς βελτιώνεται η ποιότητα απάντησης.
- Ελκυστική λύση, καθώς υπάρχουν γνωστοί αλγόριθμοι για προσεγγιστικές απαντήσεις.
- Χρήση non-blocking αλγορίθμων για σχεσιακούς τελεστές.
  - Ένας αλγόριθμος είναι blocking αν δεν παράγει κανένα αποτέλεσμα πριν επεξεργαστεί όλη την είσοδο.
    - Sort-merge: blocking.
    - Nested-loops/pipelined hash-join: non-blocking.



# Παράδειγμα

Status	State	AVG(sales)	Confidence	Interval
83%	Alabama	5232.5	97%	103.4
23%	Alaska	2832.5	93%	132.2
91%	Arizona	6432.5	98%	52.3
...	...	...	...	...
17%	Wyoming	4243.5	92%	152.3

**Επιλογή: καθορισμός προτεραιοτήτων στη σειρά υπολογισμού**



# Ερωτήματα παγόβουνου (iceberg) (1)

---

```
select R.A1, R.A2, ..., R.Ak, agr(R.B)
from Relation R
group by R.A1, R.A2, ..., R.Ak
having agr(R.B) > = constant
```





# Ερωτήματα παγόβουνου (iceberg) (2)

```
select P.custid, P. item,  
sum(P.qty)  
from Purchases P  
group by P.custid, P.item  
having sum (P.qty) > 5
```

```
select P.custid  
from Purchases P  
group by P.custid  
having sum (P.qty) > 5
```

**Q1**

```
select P.item  
from Purchases P  
group by P.item  
having sum (P.qty) > 5
```

**Q2**

**Δημιουργία (custid, item) μόνο βάσει των custid του Q1 και item του Q2**



# Αποδοτικός Υπολογισμός Κύβων

- Εκτός από το ποιοί κύβοι πρέπει να υλοποιηθούν, ένα άλλο ερώτημα είναι το πώς θα υπολογισθούν αποδοτικά αυτοί οι κύβοι.
- Εφαρμογή ταξινόμησης, κατακερματισμού, ομαδοποίησης για συσταδοποίηση σχετικών πλειάδων (ROLAP).
- Οι συναθροιστικές συναρτήσεις υπολογίζονται από προηγούμενες συναθρ. Συναρτήσεις, αντί απευθείας από τον πίνακα γεγονότων, όπου γίνεται.



# Τεχνικές Αποδοτικού Υπολογισμού Κύβων

- **Μικρότερος-πρόγονος:** υπολογισμός κυβοειδούς από το μικρότερο κυβοειδές που έχει ήδη υπολογισθεί.
- **Cache-results:** caching των αποτελεσμάτων ενός κυβοειδούς, από το οποίο επόμενα κυβοειδή θα υπολογισθούν για να μειωθεί το κόστος I/O.
- **Amortize-scans:** υπολογισμός όσο το δυνατόν περισσότερων κυβοειδών παράλληλα.
- **Share-sorts:** διαμοιρασμός κόστους ταξινόμησης.
- **Share-partitions:** διαμοιρασμός κόστους κατακερματισμού.



# Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Αναστάσιος Γούναρης.  
«Αποθήκες Δεδομένων και Εξόρυξη Δεδομένων. Ενότητα 4. Αποθήκες  
Δεδομένων – Μέρος Β'». Έκδοση: 1.0. Θεσσαλονίκη 2014.

Διαθέσιμο από τη δικτυακή διεύθυνση:<http://eclass.auth.gr/courses/OCRS182/>



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά - Μη Εμπορική Χρήση - Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>





# Τέλος ενότητας

Επεξεργασία: Ανδρέας Κοσματόπουλος  
Θεσσαλονίκη, Χειμερινό Εξάμηνο 2013-2014



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



ΑΡΙΣΤΟΤΕΛΕΙΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΟΝΙΚΗΣ

---

# Σημειώματα

# Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

