
Συστήματα Διαχείρισης και Ανάλυσης Δεδομένων

Γιάννης Κωτίδης

kotidis@aueb.gr



About me

- ◆ Professor, Dept of Informatics, Athens University of Economics and Business
 - URL: <https://pages.aueb.gr/users/kotidis/>
 - Email: kotidis@aueb.gr
 - Office: *A516, Πτέρυγα Αντωνιάδου, Κεντρικό Κτήριο ΟΠΑ*
- ◆ Interests: Big Data Systems/Algorithms, Data Streams, Time Series Databases, Graphs, Data Warehousing/Analytics, Data Mining

Recent Projects/Student Theses

(<http://pages.cs.aueb.gr/~kotidis/index.html>)



DeLorean: Compression, Indexing and Analysis Techniques for Time-Series Management



EasyFlinkCEP: Complex Event Processing using user-defined patterns



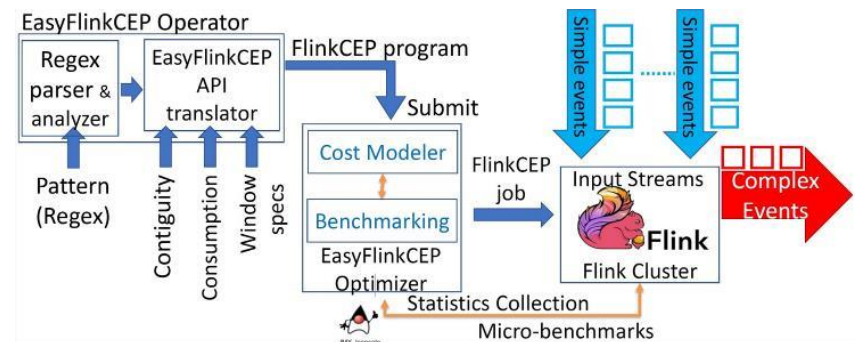
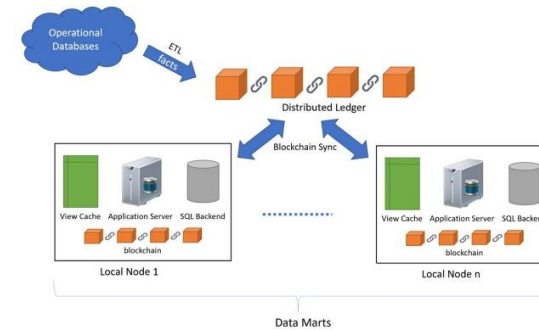
Smart-Views: Decentralized OLAP View Management using Blockchains



RECAST: Real time management of Complex Streams



Graph exploration, partitioning, augmentation



Επικοινωνία

- ◆ Γραφείο: Α516, Πτέρυγα Αντωνιάδου (5^{ος} όροφος)
- ◆ Ώρες γραφείου: Δευτέρα 12:00-13:00 & 15:00-16:00, Πέμπτη 10:00-11:00 ή άλλες ημέρες & ώρες έπειτα από συνεννόηση μέσω email
 - e-mail: kotidis@aub.gr
- ◆ *Web-page μαθήματος:*
<https://eclass.aub.gr/courses/INF245/>

Σημειώσεις/Συγγράμματα

- ◆ Διαφάνειες
 - Αναρτημένες στο e-class
- ◆ Συγγράμματα Μαθήματος
 - A) *ΘΕΜΕΛΙΩΔΕΙΣ ΑΡΧΕΣ ΣΥΣΤΗΜΑΤΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ*,
RAMEZ ELMASRI, SHAMKANT B. NAVATHE
 - B) *ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ*
Ε.ΓΙΑΝΝΑΚΟΥΔΑΚΗΣ

Βοηθοί/φροντιστήριο μαθήματος

- ◆ **Μάκης Καπέτης**, ΕΔΙΠ Τμήματος Πληροφορικής
- ◆ **Φροντιστήρια**
 - Τετάρτη 11-13 (Αμφ. Κινητής) και Τρίτη 15-17 (Δ12)
 - Ίδιο εκπαιδευτικό υλικό, θα παρακολουθείτε ένα από τα δύο
 - Θα αναρτηθεί στο eclass ενδεικτικό πρόγραμμα διαλέξεων και φροντιστηρίων
 - Θα ανακοινώνεται κάθε φορά στο μάθημα και στο eclass
 - Αν δεν υπάρξει ανακοίνωση, δε θα γίνεται φροντιστήριο τη συγκεκριμένη εβδομάδα

Εργασίες & Ασκήσεις

- ◆ Ατομικές εργασίες και ασκήσεις
 - 2 σειρές ασκήσεων (μαθησιακός στόχος: εξοικείωση με την θεωρία του μαθήματος)
 - 2 εργασίες (μαθησιακός στόχος: χρήση εργαλείων και αξιοποίηση της διδαχθείσας θεωρίας στην πράξη)
- ◆ Ενδεικτικό χρονοδιάγραμμα θα αναρτηθεί στο eclass

Πως θα περάσω το μάθημα? (in theory)

- ◆ 40% της βαθμολογίας προκύπτει από τις ασκήσεις/εργασίες, το υπόλοιπο 60% από το τελικό διαγώνισμα
- ◆ Θα πρέπει
 - Να παραδώσω επιτυχώς τουλάχιστον 3 από τις 4 εργασίες/ασκήσεις
 - Παραδίνω επιτυχώς = βαθμολογούμαι με τουλάχιστον 2.5/10
 - Να έχω μ.ο. τουλάχιστον 5, υπολογισμένο στο σύνολο των 4 εργασιών ανεξάρτητα από το όσες παρέδωσα
 - Να γράψω τουλάχιστον 4 στο τελικό διαγώνισμα
 - Ο συνολικός βαθμός να είναι τουλάχιστον 5

Πως θα περάσω το μάθημα? (in practice)

- ◆ Παρακολουθώ τις διαλέξεις και τα φροντιστήρια
 - Ρίχνω μία ματιά στις διαφάνειες πριν από κάθε διάλεξη
 - Κάνω ερωτήσεις για ότι δεν καταλαβαίνω
 - Στο μάθημα, στο διάλειμμα, σε ώρες γραφείου/teams, μέσω email
 - Δεν αφήνω κενά
- ◆ Προσπαθώ να καταλάβω πως δουλεύουν τα συστήματα, οι τεχνικές και οι αλγόριθμοι που παρουσιάζονται στο μάθημα
 - Δε παπαγαλίζω τη θεωρία ούτε απομνημονεύω τύπους
- ◆ Παραδίνω όλες τις ασκήσεις/εργασίες

Τελικό Διαγώνισμα

- ◆ Ερωτήσεις κατανόησης και ασκήσεις από όλη την ύλη του μαθήματος
 - Χωρίς «σκονάκι» ή σημειώσεις
- ◆ Μην απομνημονεύσετε λυμένες ασκήσεις από διαλέξεις και εργασίες
- ◆ Αν έχετε κατανοήσει την ύλη δε θα έχετε κανένα πρόβλημα

What is a Database?

◆ From Wikipedia:

- A **database** is a structured collection of records or data. A computer database relies upon software to organize the storage of data. The software models the database structure in what are known as database models. The model in most common use today is the relational model. Other models such as the hierarchical model and the network model use a more explicit representation of relationships ...
- **Database management systems** (DBMS) are the software used to organize and maintain the database. These are categorized according to the database model that they support. The model tends to determine the query languages that are available to access the database. A great deal of the internal engineering of a DBMS, however, is independent of the data model, and is concerned with managing factors such as performance, concurrency, integrity, and recovery from hardware failures. ...

Note

- ◆ Term “**database**” often used interchangeably for both the **data** and the **system** that manages it

Relational Database Usage

Relations

A	B	C	D	E

SQL Statements
(e.g. select columns and rows)

Select A,D
From R
Where <condition>



A	D

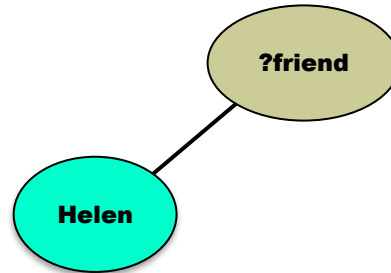
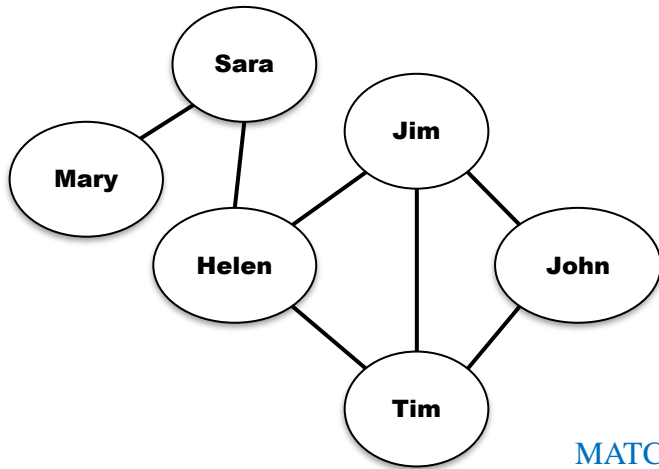


Results

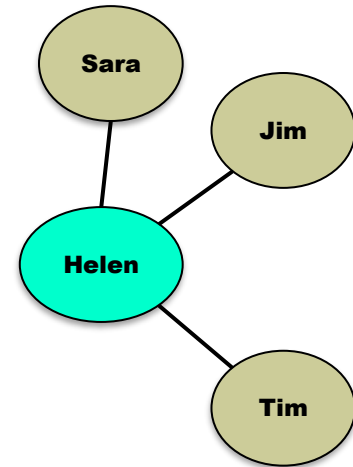
A	D

Graph Database Usage

Graph **→** Statements
(patterns: Helen and her friends) **→** Results



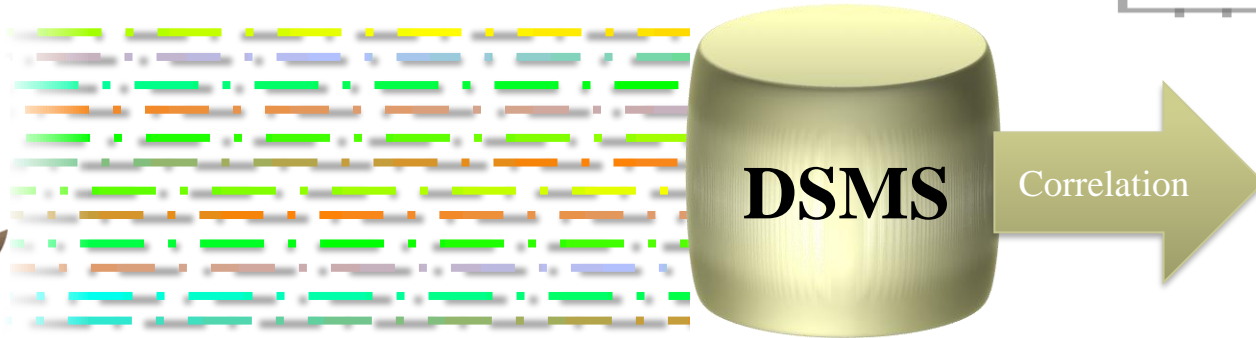
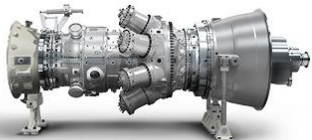
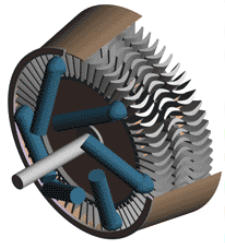
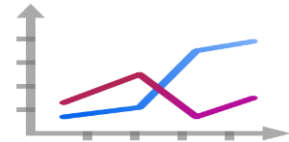
```
MATCH (n:Person {name:'Helen'})-[r:Knows]->(friend:Person)  
RETURN n.name, friend.name, r.since
```



Stream Database Example

- ◆ Intercept two streams (temperature, vibration)
- ◆ Group measurements over specified windows
- ◆ Joins streams, compute Pearson coefficient:

$$\text{Pearson}(u_i, u_j) = \text{cov}(u_i, u_j) / (\sigma_{u_i} * \sigma_{u_j})$$



Data Warehouses

OLAP

- Well defined computations over data categorized by multiple dimensions of interest
- Enables users to easily and selectively extract and query data in order to analyse it from different points of view

Data Mining

- Seek to find relationships and patterns in data
 - Frequent itemset
 - Association rules
 - Clustering

Machine Learning

- Build models for prediction, classification etc.
 - Image classification
 - Speech processing
 - Sentiment analysis
 - NLP

Database as a Service: Containerization Example

- ◆ See for example https://hub.docker.com/_/microsoft-mssql-server
- ◆ *Install: `docker run -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=myPassword12#' -p 1433:1433 -d --name mssql mcr.microsoft.com/mssql/server:2019-latest`*
- ◆ *Connect: `docker exec -it mssql /opt/mssql-tools/bin/sqlcmd -S localhost -U sa -P myPassword12#`*

Γιατί χρειαζόμαστε αυτό το μάθημα?

- ◆ Ένα ΣΔΒΔ βρίσκεται στην καρδιά ενός καλά σχεδιασμένου ΠΣ ή εφαρμογής
 - Διαχειρίζεται το πιο «πολύτιμο» αγαθό (asset): τα δεδομένα του ΠΣ/εφαρμογής
- ◆ Η απόδοση του ΣΔΒΔ αντανακλάται άμεσα στην καλή, απροβλημάτιστη λειτουργία της εφαρμογής
 - Κατανόηση του τρόπου λειτουργίας ενός ΣΔΒΔ.
 - Πως επηρεάζεται η απόδοση του από το διαθέσιμο υλικό?
 - Ποιοι αλγόριθμοι/τεχνικές αξιοποιούνται κατά την απόκριση σε ερωτήματα?
 - Πόσα resources χρειάζονται αυτοί οι αλγόριθμοι ώστε να είναι αποδοτικοί
 - Πως μπορούμε να επέμβουμε για να επιτύχουμε την επιθυμητή απόδοση?
 - Περισσότερα resources? Καλύτερη οργάνωση των δεδομένων?
 - Πως ένα ΣΔΒΔ μας βοηθάει να στήσουμε «σωστές» εφαρμογές και μας προστατεύει από σφάλματα?

Γιατί χρειαζόμαστε αυτό το μάθημα?

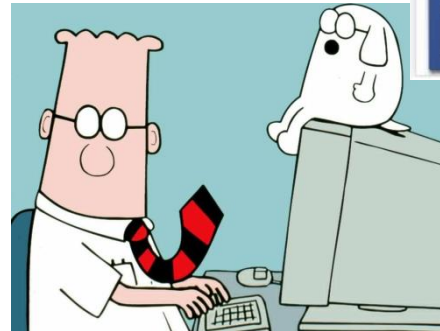
- ◆ Τα δεδομένα είναι το νέο πετρέλαιο!
 - Αποκτούν περισσότερη αξία μέσω της χρήσης και αξιοποίησης/ανάλυσης τους
 - Πως μπορούμε να δημιουργήσουμε ένα αποθετήριο το οποίο θα είναι η πηγή για την περαιτέρω επεξεργασία των δεδομένων και την ανακάλυψη νέας γνώσης από αυτά?
 - Πως μπορούμε να επεξεργαστούμε/αναλύσουμε σύνθετα/μεγάλα δεδομένα?
 - Οι αλγόριθμοι που χρησιμοποιούνται σε ΣΔΒΔ αποτελούν καλές πρακτικές!

Back to the future

- ◆ You work for a big telecom company
COSMOPHONE
 - Head of information analysis department (\$\$\$)
- ◆ Asked by CEO to report calls by your subscribers to the evil **VONNOTE** competitor



CEO



YOU



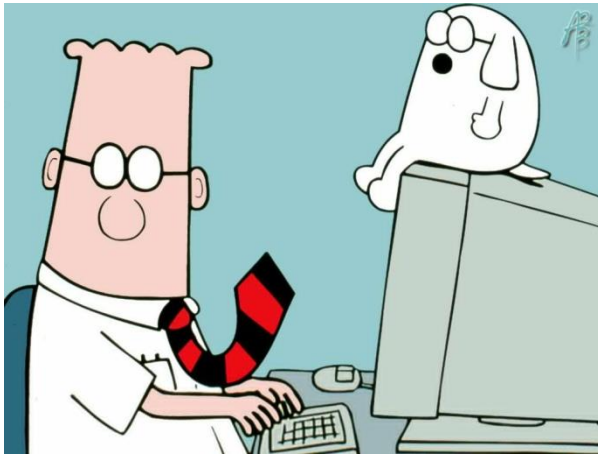
Ένα χρόνο πριν...





Μπορούμε να το κάνουμε;

- ◆ Έχουμε την τεχνολογία των ΒΔ!



SQL



Database server
powered by HAL 9000

Πίνακας (σχέση) με δεδομένα τηλεφωνικών κλήσεων

CALLEDDETAILS (εγγραφές για το 2023)

CustId	Call_start	Call_end	PhoneNo	CompId
100103	8:00	8:02	6995219694	COSMOPHONE
100105	7:55	8:20	6995811821	VONNOTE
100105	8:40	8:44	6991155123	VONNOTE
105812	8:55	8:59	6991058186	COSMOPHONE
...

SQL Επερώτηση

```
SELECT CustId, Call_end-Call_start  
FROM CALLEDDETAILS  
WHERE CompId="VONNOTE"
```

data

CustId	Call_start	Call_end	PhoneNo	CompId
100103	8:00	8:02	6995219694	COSMOPHONE
100105	7:55	8:20	6995811821	VONNOTE
100105	8:40	8:44	6991155123	VONNOTE
105812	8:55	8:59	6991058186	COSMOPHONE
...

query



results

CustId	Call_end-Call_start
100105	25
100105	4
...	...

Πως παράγεται αυτό το αποτέλεσμα?

Σημείωση

- ◆ Το περιεχόμενο του πίνακα CALLDETAILS είναι αποθηκευμένο σε κάποιο μέσο **μόνιμης** αποθήκευσης, ας υποθέσουμε σε έναν «σκληρό δίσκο» (HDD)
- ◆ Το ΣΔΒΔ πρέπει να αναζητήσει τις εγγραφές που θέλουμε και να τις μεταφέρει στην κύρια μνήμη (RAM) ώστε να τις επεξεργαστεί περαιτέρω και να επιστρέψει το αποτέλεσμα στο χρήστη (πχ στο τερματικό του)
- ◆ Αυτή η αναζήτηση μπορεί να διευκολυνθεί με τη χρήση **ευρετηρίων**

Ευρετήριο Όρων (βιβλίο)

Index entry
(algorithm,33)

Όρος αναζήτησης (search key) Θέση (page number)

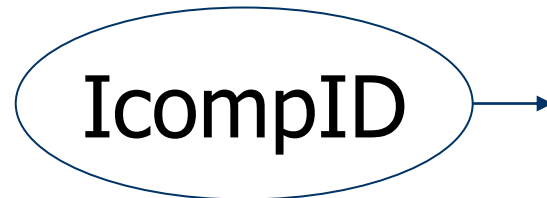
Index

fluid
incompressible, see incompressible fluid
fluid
Airy wave theory, 25
algorithm, 33
amortized running time, 33
amplitude, 26
angular frequency, 25
approximation, 7
Bessel function
zeroth order of the first kind, 30
big O notation, 33
cell, 7
cell face, 7
CFMM, 25, 32
Continuous Fast Multipole Method, see CFMM
convolution, 27
convolution filter, 27
convolution kernel, 27, 28
convolution theorem, 28
core, 33
density, 25
depth
effective, see effective depth
water, see water depth
differentiation
filter
convolution, see convolution filter
low-pass, see low-pass filter
Finite Volume Method, see FVM
fluid flux, 7
flux
fluid, see fluid flux
FMM, 32
Fourier transform
non-uniform, see non-uniform Fourier transform
reverse, see reverse Fourier transform
free surface elevation, 25
frequency domain, 26
FVM, 7
gradient, 26
gravitational acceleration, 25
grid point
surface, see surface grid point
Hankel transform
zeroth order, 30
incompressible fluid, 7
instability, 7
kernel
convolution, see convolution kernel

Δημιουργία ευρετηρίου στο γνώρισμα CompID

```
CREATE INDEX IcompID on CALLEDETAILS(CompId);
```

CompId=“**VONDOTTE**”?



Εγγραφή στο
δίσκο



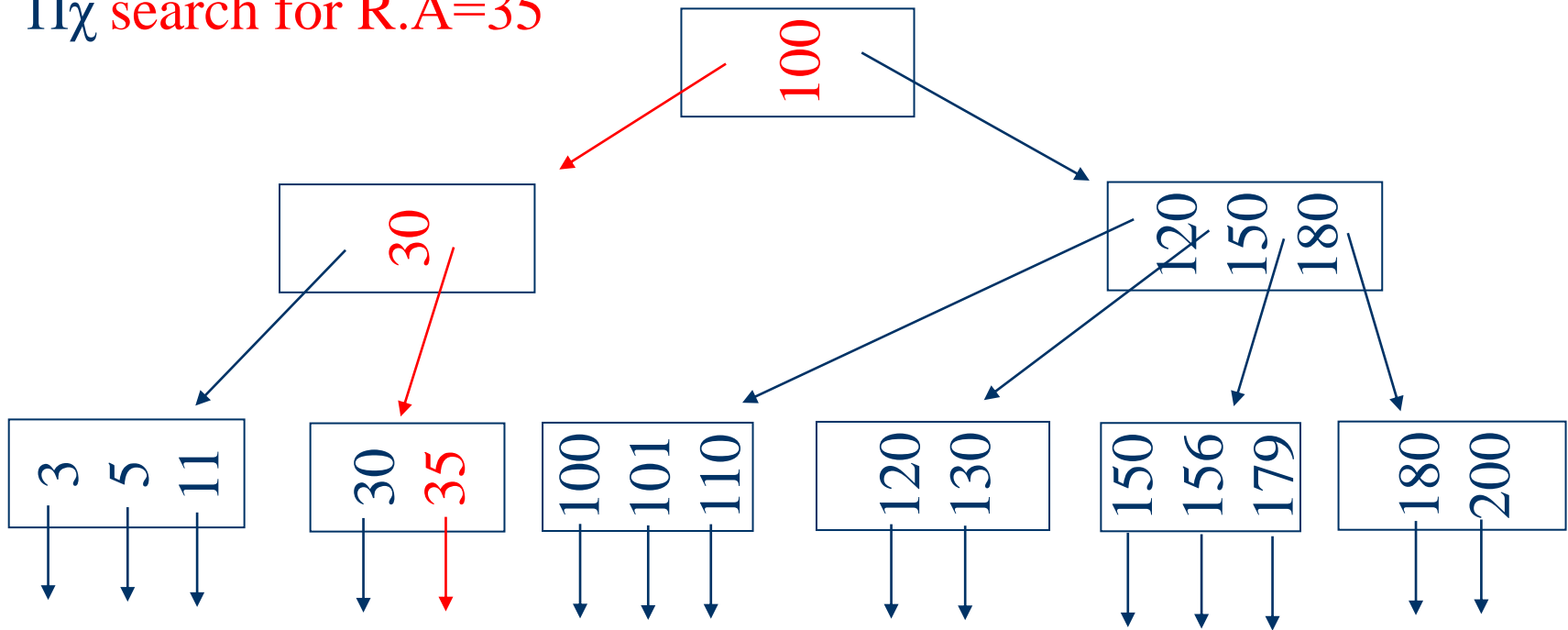
- ◆ Το ευρετήριο μας «κατευθύνει» προς τις κατάλληλες εγγραφές

Παράδειγμα: B+-Tree

(για πεδίο αναζήτησης R.A τύπου int)

Root (Ρίζα)

Πχ search for R.A=35



Προς θέση (πχ σελίδα ή row-id ή primary key value)
στο δίσκο της εγγραφής με τιμή R.A=35

Πόσο χρόνο θα πάρει η εκτέλεση της επερώτησης?

- ◆ Η συγκεκριμένη επερώτηση δεν κάνει κάτι χρονοβόρο με αυτές τις εγγραφές από τη στιγμή που αυτές έρθουν στη μνήμη

```
SELECT CustId, Call_end-Call_start  
FROM CALLEDETAILS  
WHERE CompId="VONNOTE"
```

- ◆ Επομένως θα επικεντρωθούμε στην καθυστέρηση που προκαλεί η μεταφορά εγγραφών από το δίσκο στη μνήμη (I/O: Input/Output)

Πόσο χρόνο θα πάρει η εκτέλεση της επερώτησης?

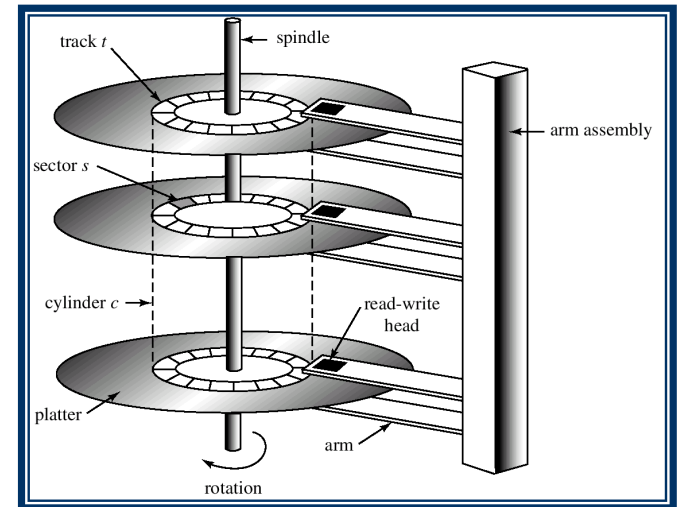
- ◆ Τι πληροφορίες χρειαζόμαστε για να υπολογίσω το χρόνο?
 1. Στατιστικά στοιχεία (δεδομένα): πόσες δεδομένα περιμένω να διαβάσω από το δίσκο?
 2. Τεχνικά στοιχεία (υλικό): πόσο γρήγορα δουλεύει ο δίσκος μου?

Στατιστικά

- ◆ 1,000,000 phone-calls per day
- ◆ 365M εγγραφές για όλο το έτος
- ◆ 1% είναι κλήσεις στην VONDOTE = 10,000/day
- ◆ 3.65M εγγραφές στον πίνακα CALLEDETAILS
αφορούν κλήσεις προς τη VONDOTE
 - Αυτές θα πρέπει να τις διαβάσω από το δίσκο

Πόσο χρόνο θα πάρει η εκτέλεση της επερώτησης?

- ♦ Με την αναζήτηση μέσω του ευρετηρίου, βρίσκω τη θέση (αριθμό σελίδας) στο δίσκο όπου βρίσκεται αποθηκευμένη η εγγραφή
- ♦ Δεδομένου ότι οι εγγραφές στο συγκεκριμένο παράδειγμα δεν είναι οργανωμένες με κάποιο ιδιαίτερο τρόπο (πχ ταξινομημένες ως προς το γνώρισμα compID), για κάθε εγγραφή θα ανακαλώ τη σελίδα της κάνοντας **τυχαία προσπέλαση** (random access)



IndexSeek: ανάγνωση μέσω ευρετηρίου

- ◆ Υποθέτουμε 20msecs **disk random access time**
 - Αγνοούμε όλες τις άλλες καθυστερήσεις
 - Υποθέτω ότι το ευρετήριο βρίσκεται στη μνήμη
 - Αγνοώ πιθανά cache-hits (buffers)
- ◆ $3.65\text{M records} * 0.02\text{sec/record} =$
20+ hours



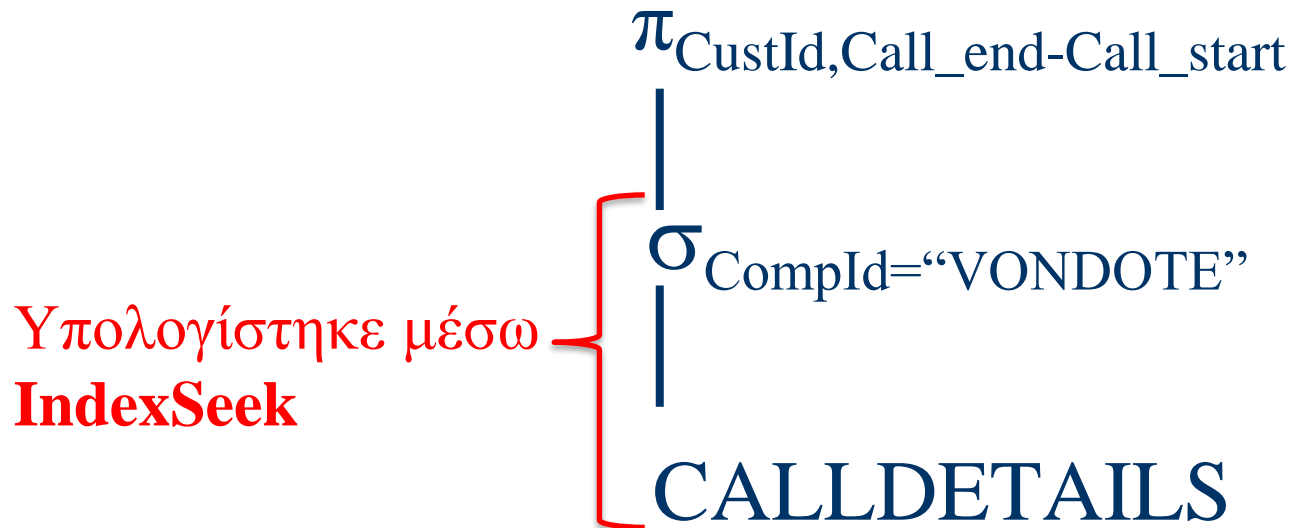
Μετά ο CEO σας ρωτάει

- ◆ Μπορείς να επαναλάβεις τον υπολογισμό για το 2022?
 - ... άλλες 20 ώρες

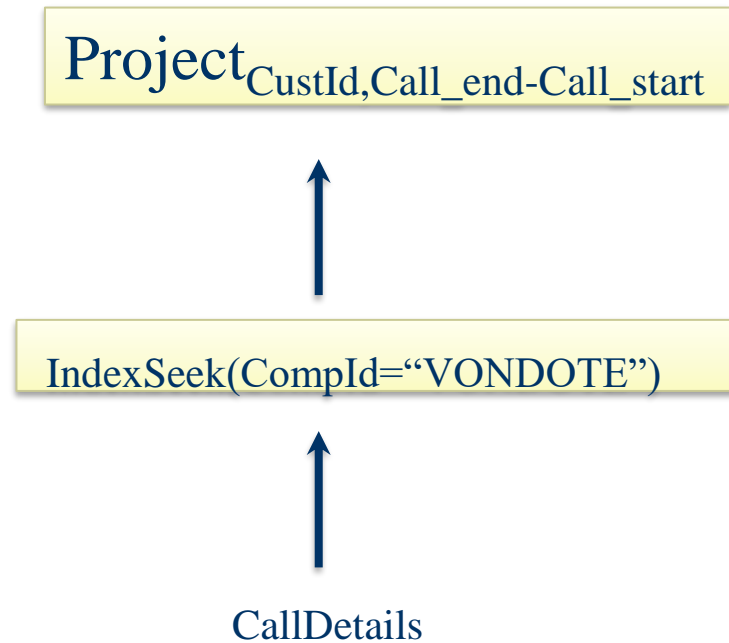
Γιατί τόσο απλές ερωτήσεις καθυστερούν τόσο πολύ?

◆ ?

Λογικό πλάνο Επερώτησης



Φυσικό πλάνο 1

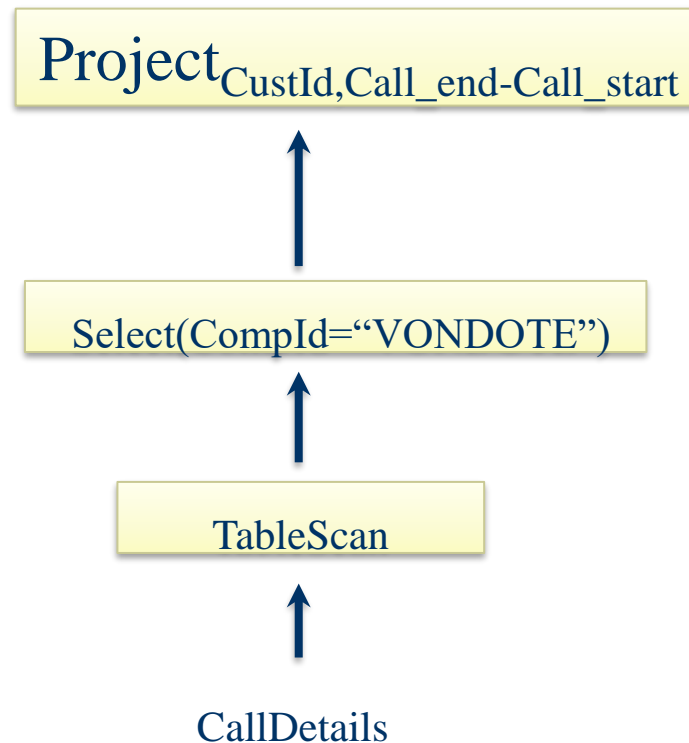


Μια Διαφορετική Στρατηγική

- ◆ Ας μη χρησιμοποιήσουμε το ευρετήριο
 - **TableScan:** Διαβάζω όλο τη σχέση από την αρχή μέχρι το τέλος
 - **Select:** κρατάω μόνο εγγραφές που με ενδιαφέρουν

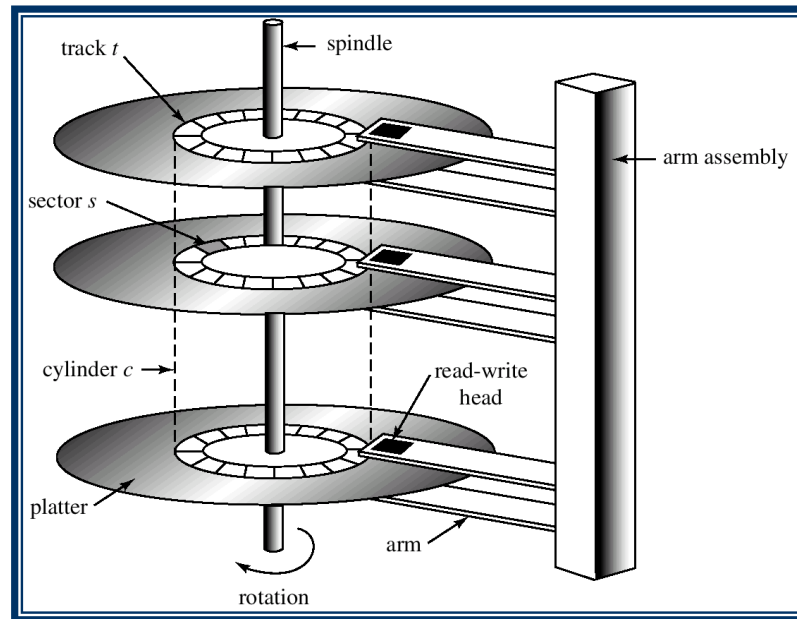
Εναλλακτική Εκτέλεση

Φυσικό πλάνο 2



Πίσω στο δίσκο

- ◆ 1,000,000 phone-calls per day, **128 bytes/record**
- ◆ $365 * 128 * 10000000$ bytes/year = 43.5GB



Χαρακτηριστικά Δίσκου

- ◆ 4KB/sector
- ◆ 128sectors/track
- ◆ Ταχύτητα περιστροφής
5400 rpm



Θα υποθέσω ότι οι σελίδες της σχέσης βρίσκονται αποθηκευμένες **ακολουθιακά** στην επιφάνεια του δίσκου

Ας σκεφτούμε

- ◆ 4KB/sector
- ◆ 128sectors/track
- ◆ Ταχύτητα περιστροφής
5400 rpm

Σε ένα λεπτό διαβάζω

$$5400 * 128 * 4 \text{ KB}$$

Δηλαδή 45 MB/sec

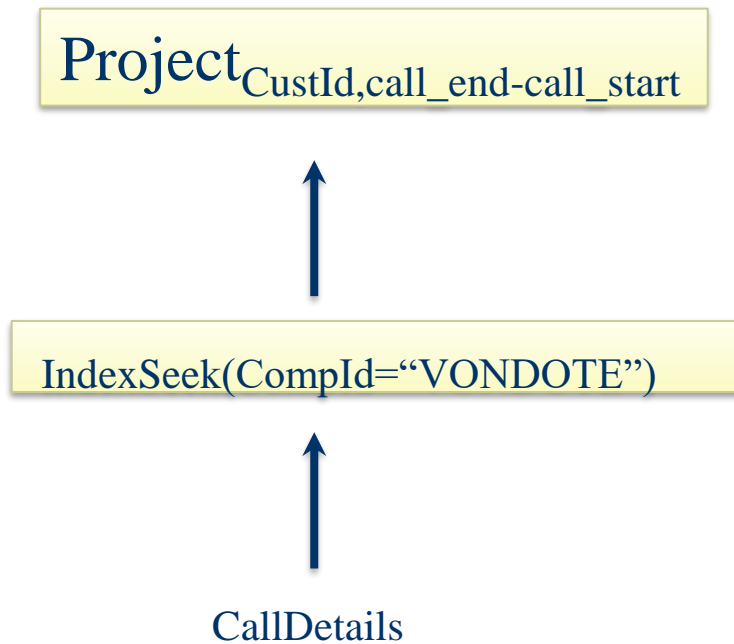
Για να διαβάσω 43.5GB
χρειάζομαι (τουλάχιστον)

$$1024 * 43.5 / 45 / 60$$

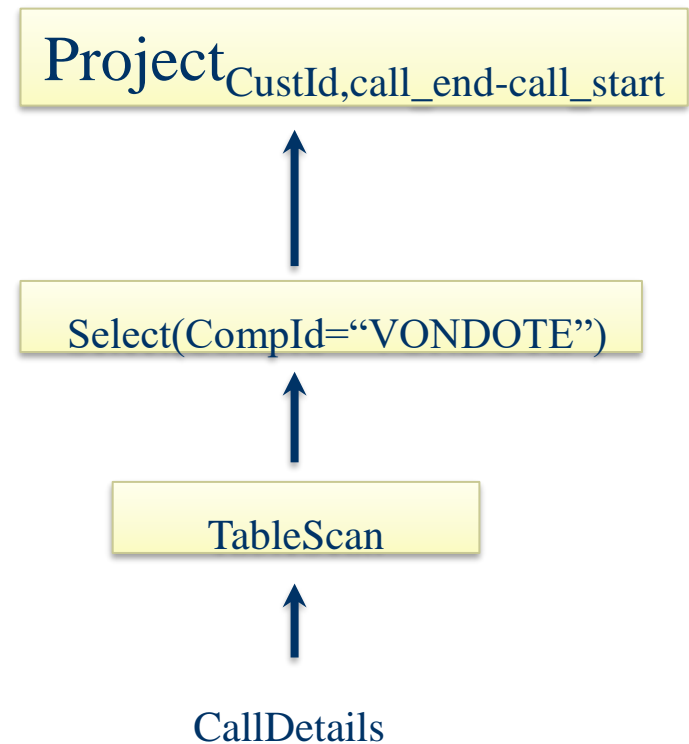
$$= 16.5 \text{ mins}$$

Ας συγκρίνουμε

Με χρήση ευρετηρίου: 20 ώρες



«Απλοϊκή» εκτέλεση: 16.5 λεπτά



Μπορώ ακόμα καλύτερα;

- ◆ Επένδυση σε υλικό
 - ➔ ■ CPU?
 - Μνήμη?
 - Ταχύτερο HDD, RAID?
 - SSD?
- ◆ Αλλαγή στη σχεδίαση της Βάσης
 - ???

Μπορώ ακόμα καλύτερα;

- ◆ Επένδυση σε υλικό

- CPU?

**Μέγεθος σχέσης=
43.5GB**



- Μνήμη?


- Ταχύτερο HDD, RAID?

- SSD?

- ◆ Αλλαγή στη σχεδίαση της Βάσης

- ???

Μπορώ ακόμα καλύτερα;

- ◆ Επένδυση σε υλικό
 - CPU?
 - Μνήμη?
 - Ταχύτερο HDD, RAID?
 -  ■ SSD?
- ◆ Αλλαγή στη σχεδίαση της Βάσης
 - ???

SSD

```
-----  
CrystalDiskMark 3.0.1 x64 (c) 2007-2010 hiyohiyo  
Crystal Dew world : http://crystalmark.info/  
-----  
* MB/s = 1,000,000 byte/s [SATA/300 = 300,000,000 byte/s]  
  
Sequential Read : 253.647 MB/s  
Sequential Write : 220.949 MB/s  
Random Read 512KB : 207.864 MB/s  
Random Write 512KB : 225.650 MB/s  
Random Read 4KB (QD=1) : 18.410 MB/s [ 4494.7 IOPS]  
Random Write 4KB (QD=1) : 41.489 MB/s [ 10129.3 IOPS]  
Random Read 4KB (QD=32) : 194.360 MB/s [ 47451.2 IOPS]  
Random Write 4KB (QD=32) : 113.936 MB/s [ 27821.4 IOPS]  
  
Test : 1000 MB [C: 18.4% (22.0/119.1 GB)] (x1)  
Date : 2012/08/21 19:23:13  
OS : windows 7 [6.1 Build 7600] (x64)
```

HDD

```
-----  
CrystalDiskMark 3.0.1 x64 (c) 2007-2010 hiyohiyo  
Crystal Dew world : http://crystalmark.info/  
-----  
* MB/s = 1,000,000 byte/s [SATA/300 = 300,000,000 byte/s]  
  
Sequential Read : 125.698 MB/s  
Sequential Write : 122.913 MB/s  
Random Read 512KB : 39.305 MB/s  
Random Write 512KB : 51.902 MB/s  
Random Read 4KB (QD=1) : 0.437 MB/s [ 106.7 IOPS]  
Random Write 4KB (QD=1) : 0.990 MB/s [ 241.7 IOPS]  
Random Read 4KB (QD=32) : 0.972 MB/s [ 237.3 IOPS]  
Random Write 4KB (QD=32) : 1.004 MB/s [ 245.1 IOPS]  
  
Test : 1000 MB [C: 73.7% (343.2/465.8 GB)] (x1)  
Date : 2012/09/11 16:19:50  
OS : windows 7 SP1 [6.1 Build 7601] (x64)
```

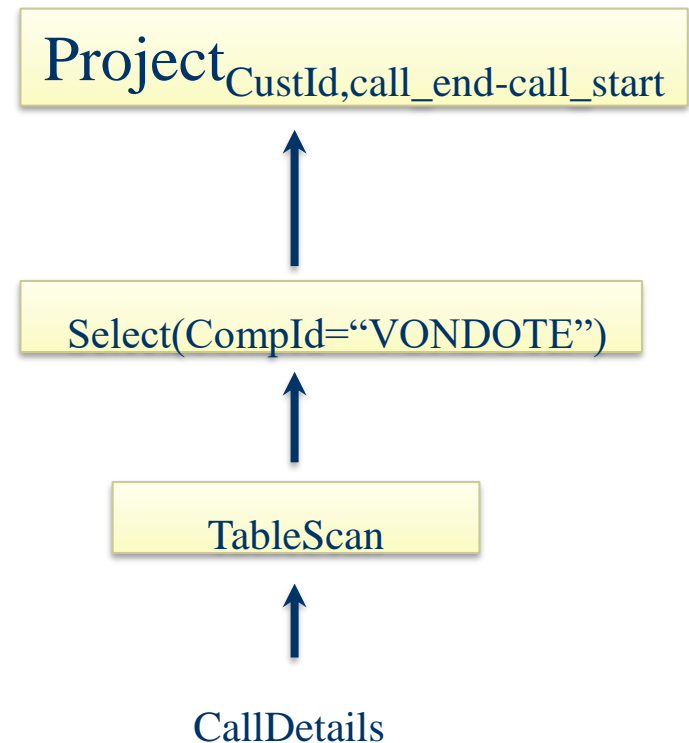
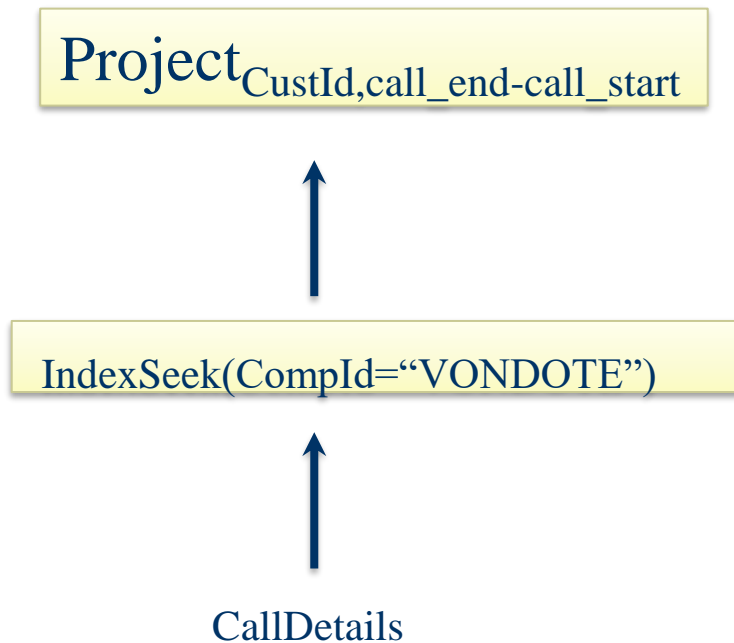
Ας δούμε το παράδειγμα του SSD

- ◆ Σειριακή ανάγνωση (Sequential Reads):
275MB/sec
- ◆ Τυχαία προσπέλαση 45K IOPS
 - Πλάνο 1: $3.65\text{M I/O} / 45000 = 81\text{secs}$
 - Πλάνο 2: $43.5\text{GB} / 275\text{MB/sec} = 161\text{secs}$

Σύνοψη

Αν χρησιμοποιήσω ευρετήριο **20 hours με HDD**
81 secs με SSD

«Απλοϊκή» εκτέλεση: **16.5 mins με HDD**
161 secs με SSD



Μπορώ ακόμα καλύτερα;

- ◆ Επένδυση σε υλικό
 - CPU?
 - Μνήμη?
 - HDD?
 - SSD?
- ◆ Αλλαγή στη σχεδίαση της Βάσης
 - Ευρετήριο συστάδων

Ευρετήριο συστάδων

- ◆ Ευρετήριο συστάδων (clustering index) ορίζεται στο πεδίο διάταξης το οποίο όμως δεν είναι κλειδί

Παράδειγμα: σχέση $R(A, \dots)$

- ♦ Παραδοχή: οι εγγραφές με τιμή $A=19$ είναι μαζεμένες σε όσο το δυνατό λιγότερες σελίδες στο δίσκο



A	
10	
10	
35	

19	
19	
19	

19	
42	
37	

Με ευρετήριο συστάδων στο compID

- ◆ 3.65M εγγραφές, 128 bytes/εγγραφή
- ◆ 4KB μέγεθος σελίδας
- ◆ Το αποτέλεσμα (οι εγγραφές που ψάχνουμε) χωράει σε **$3650000 * 128\text{bytes} = 446\text{ MB}$**
 - Υποθέτω ότι οι σελίδες αυτές είναι σειριακά γραμμένες στο δίσκο....
- ◆ Ανάγνωση με HDD @ 45MB/sec = ~ 10 secs
- ◆ Ανάγνωση με SSD @ 275MB/sec = ~ 2 secs

Σύνοψη

Καλύτερη λύση και για τις 2 τεχνολογίες

Αν χρησιμοποιήσω ευρετήριο: **20+ hours με HDD**

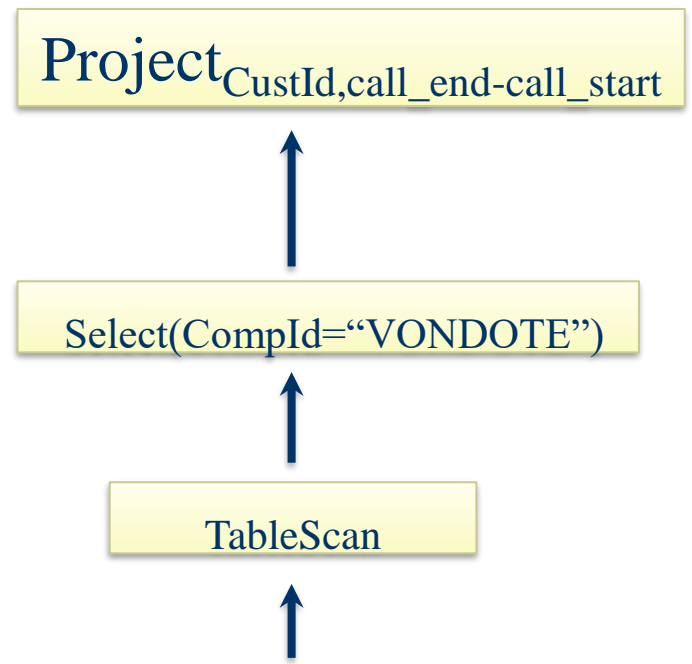
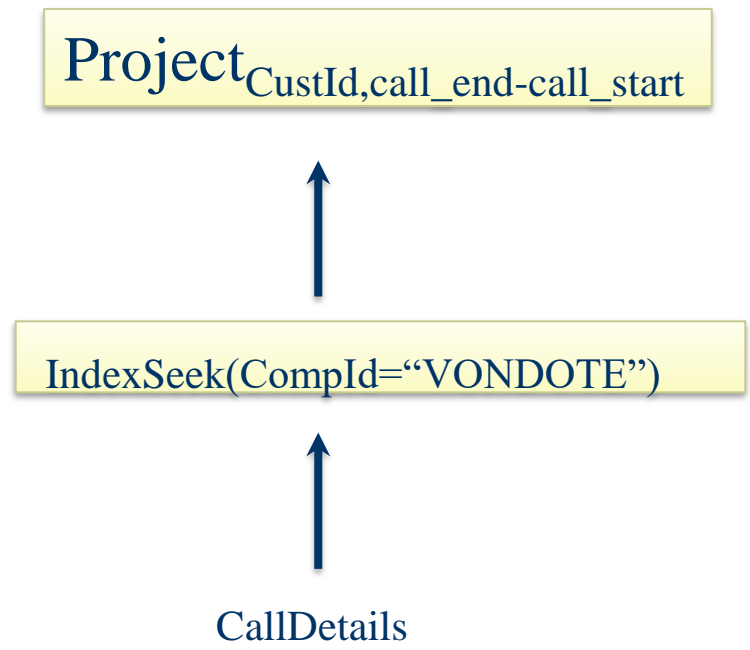
«Απλοϊκή» εκτέλεση: **16.5 mins με HDD**

81 secs με SSD

161 secs με SSD

Ευρετήριο συστάδων:

10 secs με HDD
2 secs με SSD



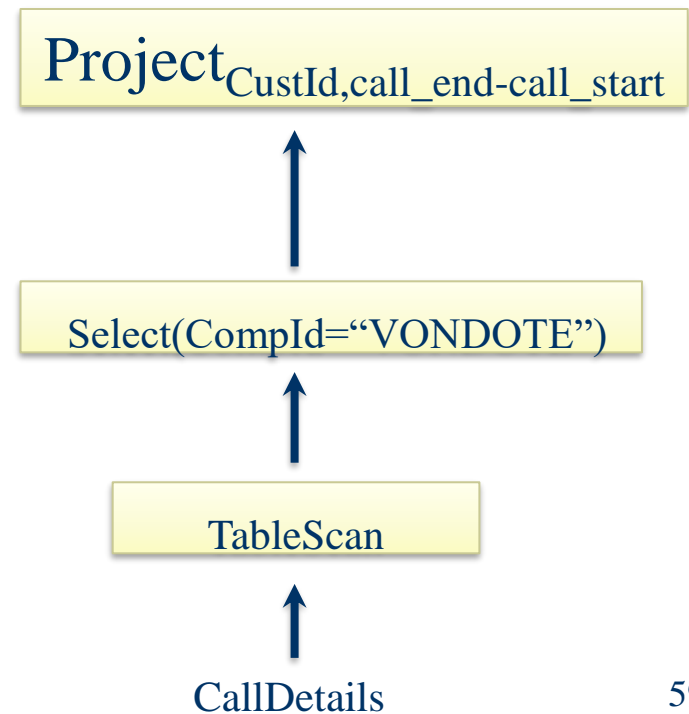
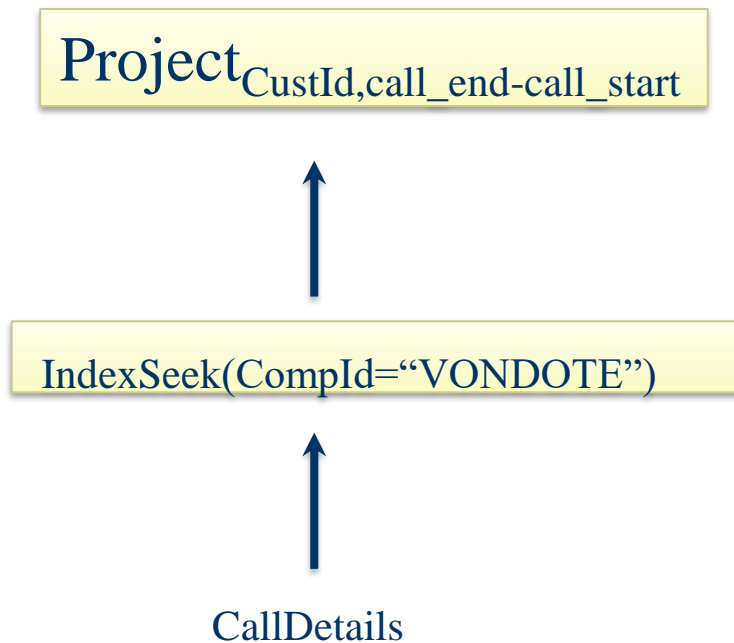
Ευρετήριο συστάδων, κακή οργάνωση στο δίσκο

- ◆ Το αποτέλεσμα χωράει σε 446 MB
 - Αν οι σελίδες δεν είναι σειριακά αποθηκευμένες στο δίσκο → **Τυχαία προσπέλαση** (random I/O)
 - Υποθέτω το χειρότερο σενάριο...
- ◆ 446 MB είναι 114176 σελίδες των 4KB
 - Ανάγνωση με HDD @ 20 msec/page = 38 mins
 - Ανάγνωση με SSD @ 45K IOPS = ~ 3 secs

Σύνοψη

Αν χρησιμοποιήσω ευρετήριο: **20 hours με HDD** «Απλοϊκή» εκτέλεση: **16.5 mins με HDD**
81 secs με SSD 161 secs με SSD

Ευρετήριο συστάδων: 10 secs–38 mins με HDD
2-3 secs με SSD



Περιεχόμενο Μαθήματος

- ◆ Πως το hardware επηρεάζει τη λειτουργία ενός ΣΔΒΔ
 - Ιεραρχία Μνήμης
 - Moore's law
 - Λειτουργία δίσκων HDD, σύγκριση με SSD
 - Ο κανόνας των 5 λεπτών

Περιεχόμενο Μαθήματος

- ◆ Ευρετήρια
 - Τύποι ευρετηρίων
 - Απλά ευρετήρια
 - B-trees
 - Κατακερματισμός (Στατικός, Δυναμικός)
 - Space filling curves (Z-ordering, Hilbert)
 - Πολυδιάστατα ευρετήρια (kd-trees, R-trees, quad-trees, etc)

Περιεχόμενο Μαθήματος

- ◆ Βελτιστοποίηση Επερωτήσεων
 - Λογικό πλάνο
 - Αλγεβρικοί μετασχηματισμοί
 - Μοντέλο υλοποιητή, μοντέλο επαναλήπτη

Περιεχόμενο Μαθήματος

- ◆ Φυσικά πλάνα
 - Υπολογισμός κόστους
 - Στατιστικά
 - Ταξινόμηση
 - Ερωτήματα με `distinct()`, `group by`
 - Αλγόριθμοι για σύζευξη

Περιεχόμενο Μαθήματος

- ◆ Δοσοληψίες
 - ACID Properties
 - OLTP συστήματα
 - Διαχείριση δοσοληψιών σε επίπεδο ΣΔΒΔ
 - Ανάνηψη από σφάλματα

Περιεχόμενο Μαθήματος

- ◆ Αποθήκες Δεδομένων
 - Ορισμός, αρχιτεκτονική, διαφορές από OLTP συστήματα
 - Σχεσιακά σχήματα αποθήκης δεδομένων
 - Πολυδιάστατο μοντέλο δεδομένων, παραδείγματα ανάλυσης OLAP
 - Ο τελεστής του Κύβου
 - Ορισμός, χρήση, επιλογή και αξιοποίηση όψεων

Περιεχόμενο Μαθήματος

- ◆ Μη σχεσιακές βάσεις δεδομένων
 - Το παράδειγμα των graph databases
- ◆ Ανάλυση Δεδομένων
 - Εισαγωγή στην εξόρυξη δεδομένων