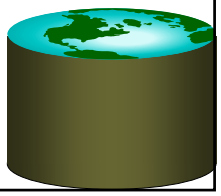
CS186 - Introduction to Database Systems

Spring Semester 2006
Prof. Michael Franklin

"Knowledge is of two kinds: we know a subject ourselves, or we know where we can find information upon it."
-- Samuel Johnson (1709-1784)

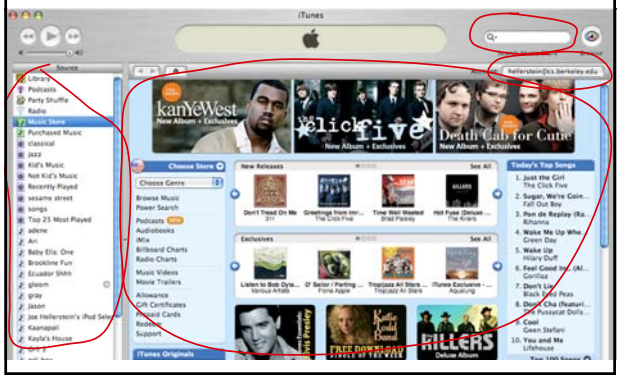


Database Systems: Then

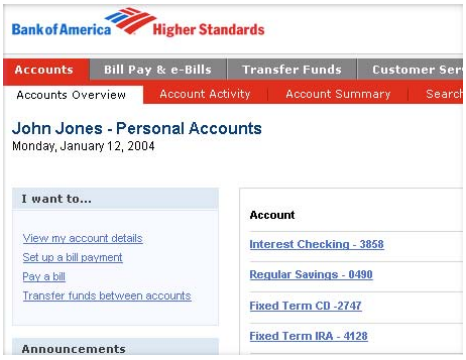
Database Systems: Today

From Friendster.com on-line tour

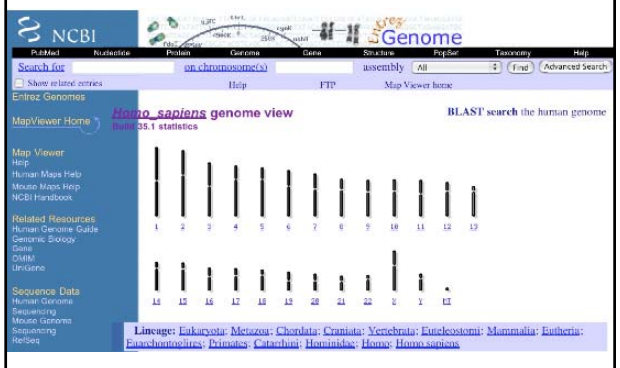
Database Systems Today



Database Systems Today



Database Systems Today



Other ways Databases Make Life Better?

- **"Players could finally sign up for the Star Wars Galaxies game last week as Sony opened up registration to the public."**
- **"Once players got in to the game they found that the game servers were offline because of database problems."**
- **"Some players spent hours tuning their in-game characters only to find that crashes deleted all their hard work."**
- **Source: BBC News Online, July 1, 2003.**

Other databases you may use

So...*What* Is a Database System?

- **Database:**
  a very large, integrated collection of data.
- **Models a real-world *enterprise***
  - Entities (e.g., teams, games)
  - Relationships
    (e.g., Cal *is playing against* Stanford)
  - More recently, also includes active components , often called "business logic". (e.g., the BCS ranking system)
- **A *Database Management System (DBMS)* is a software system designed to store, manage, and facilitate access to databases.**
- **More expansive definitions are possible (and more interesting...)**

Is the WWW a DBMS?

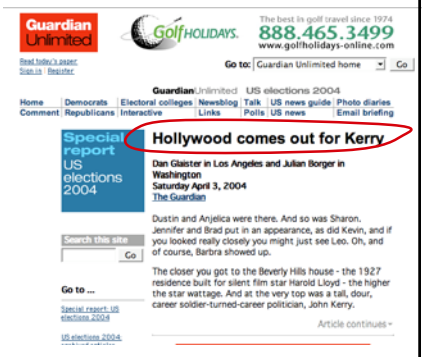
- **Fairly sophisticated search available**
  - crawler *indexes* pages on the web
  - Keyword-based search for pages
- **But, currently**
  - data is mostly unstructured and untyped
  - search only:
    - can't modify the data
    - can't get summaries, complex combinations of data
  - few guarantees provided for freshness of data, consistency across data items, fault tolerance, ...
  - Web sites typically have a DBMS in the background to provide these functions.
- **The picture is changing**
  - New standards e.g., XML, Semantic Web can help data modeling
  - Research groups (e.g., at Berkeley) are working on providing some of this functionality *across multiple web sites.*

"Search" vs. Query

What if you wanted to find out which actors donated to John Kerry's presidential campaign?

- **Try "hollywood kerry donations" in your favorite search engine.**

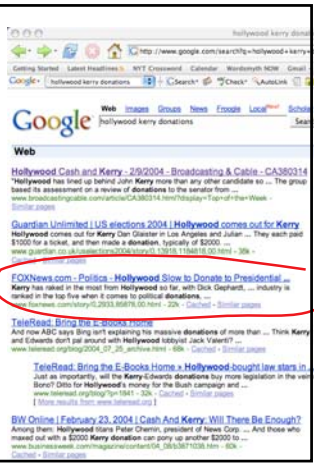
"Search" vs. Query



"Search" vs. Query

What if you wanted to find out which actors donated to John Kerry's presidential campaign?

- **Try "hollywood kerry donations" in your favorite search engine.**



"Search" vs. Query

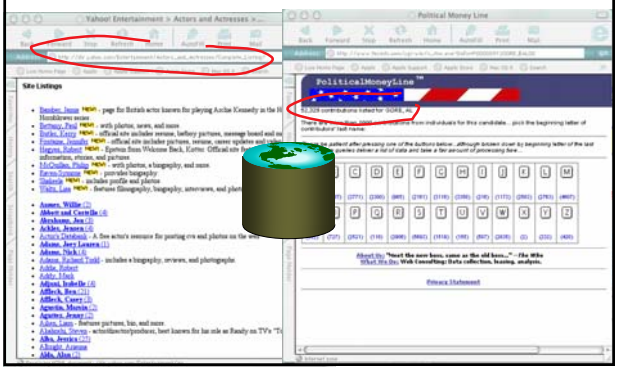
- **"Search" can return only what's been previously "stored".**

And, it's subject to the "spin" of whoever did the storing.

Also...

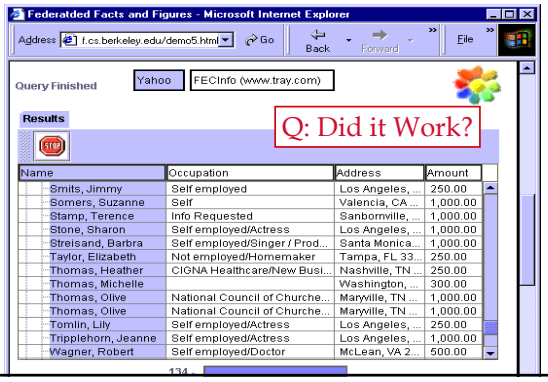
- **What if I wanted to find out the average donation of actors to each candidate?**
- **What if I wanted to compare actor donations this campaign to the last one?**
- **What if I wanted to find out who gave the most to each candidate?**
- **What if I wanted to know where the data came from, and how old it was?**

A "Database Query" Approach



"Yahoo Actors" JOIN "FECInfo"

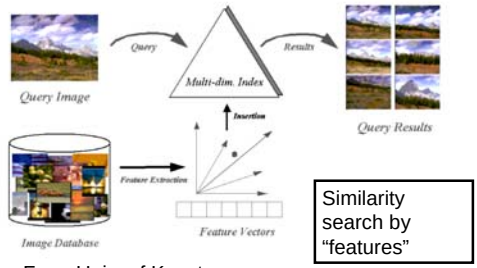
(Courtesy of the Telegraph research group @Berkeley)



What's going on here?

- **Unstructured Data**
  - Text-based search is based mostly on statistical models of similarity.
    - no real "understanding" of the data
  - Google's big step forward was to exploit some of the *structure* in web documents.
  - Still, web search places a large burden on *people* to do the last stage of filtering and interpretation.
- **Structure gives *computers* the ability to manipulate and maintain the data.**
- **Traditional (relational) Database systems are aimed at *structured* data.**

Other Unstructured Data - Images



Similarity search by "features"

Picture From Univ. of Konstanz

What about structured data?

- A *data model* is a collection of concepts for describing data.
- A *schema* is a description of a particular collection of data, using a given data model.

- The *relational model of data* is the most widely used model today.
  - Main concept: *relation*, basically a table with rows and columns.
  - Every relation has a *schema*, which describes the columns, or fields.

Example: University Database

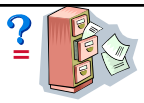
- **Conceptual schema:**
  - *Students*(sid: *string*, name: *string*, age: *integer*, gpa:*real*)
  - *Courses*(cid: *string*, cname:*string*, credits:*integer*)
  - *Enrolled*(sid:*string*, cid:*string*, grade:*string*)
    FOREIGN KEY sid REFERENCES Students
    FOREIGN KEY cid REFERENCES Courses
- **External Schema (View):**
  - *Course_info*(cid:string,enrollment:integer)
    Create View Course_info AS
    SELECT cid, Count (*) as enrollment
    FROM Courses
    GROUP BY cid

So, Don't you need both?

Good Old Text Search

Database Query

Is a File System a DBMS?



- **Thought Experiment 1:**
  - You and your project partner are editing the same file.
  - You both save it at the same time.
  - Whose changes survive?

A) Yours B) Partner's C) Both D) Neither E) ???

- **Thought Experiment 2:**
  - You're updating a file.
  - The power goes out.
  - Which of your changes survive?

Q: How do you write programs over a subsystem when it promises you only "???" ?
A: Very, very carefully!!

A) All B) None C) All Since last save D) ???

OS Support for Data Management

- **Data can be stored in RAM**
  - this is what every programming language offers!
  - RAM is fast, and random access
  - Isn't this heaven?
- **Every OS includes a File System**
  - manages *files* on a magnetic disk
  - allows *open, read, seek, close* on a file
  - allows protections to be set on a file
  - drawbacks relative to RAM?

Database Management Systems

- **What more could we want than a file system?**
  - Simple, efficient *ad hoc*[1] queries
  - concurrency control
  - recovery
  - benefits of good data modeling
- **S.M.O.P.[2]? Not really…**
  - as we'll see this semester
  - in fact, the OS often gets in the way!

[1]ad hoc: formed or used for specific or immediate problems or needs
[2]SMOP: Small Matter Of Programming

Why take this class?

A. Database systems are the core of CS

- **Shift from computation to information**
  - True in corporate computing for years
  - Web, p2p made this clear for personal computing
  - Increasingly true of scientific computing
- **Need for DB technology has exploded in the last years**
  - Corporate: retail swipe/clickstreams, "customer relationship mgmt", "supply chain mgmt", "data warehouses", etc.
  - Web:not just "documents". Search engines, e-commerce, blogs, wikis, other "web services".
  - Scientific: digital libraries, genomics, satellite imagery, physical sensors, simulation data
  - Personal: Music, photo, & video libraries. Email archives. File contents ("desktop search").

Why take this class?

B. DBs are incredibly important to society

- **"Knowledge is power." -- Sir Francis Bacon**
- **"With great power comes great responsibility." -- SpiderMan's Uncle Ben**

Policy-makers should understand technological possibilities.
Informed Technologists needed in public discourse on usage.

Why take this class?

C. The topic is intellectually rich.

- **representing information**
  - data modeling
- **languages and systems for querying data**
  - complex queries & query semantics*
  - over massive data sets
- **concurrency control for data manipulation**
  - controlling concurrent access
  - ensuring transactional semantics
- **reliable data storage**
  - maintain data semantics even if you pull the plug

* semantics: the meaning or relationship of meanings of a sign or set of signs

Why take this class?

D. The course is a capstone.

- **We will see**
  - Algorithms and cost analyses
  - System architecture and implementation
  - Resource management and scheduling
  - Computer language design, semantics and optimization
  - Applications of AI topics including logic and planning
  - Statistical modeling of data

Why take this class?

~~**E. It isn't that much work.**~~

- **Bad news: It is a lot of work.**
- **Good news: the course is front loaded**
  - Much of the hard work is in the first half of the semester
  - Load balanced with most other classes

Why take this class?

F. Looks good on my resume.

- **Yes, but why? This is not a course for:**
  - Oracle administrators
  - IBM DB2 engine developers
    - Though it's useful for both!
- **It is a course for well-educated computer scientists**
  - Database system concepts and techniques increasingly used "outside the box"
    - Ask your friends at Microsoft, Google, Apple, etc.
    - Actually, they may or may not realize it!
  - A rich understanding of these issues is a basic and (un?)fortunately unusual skill.

Administrivia Break: Workload

- **Projects with a "real world" focus:**
  - Modify the internals of a "real" open-source database system: PostgreSQL
    - Serious C system hacking
    - Measure the benefits of our changes
  - Build a web-based e-commerce application w/PostgreSQL, Apache, and PHP (almost "LAMP")
  - Other homework assignments and/or quizes
- **Exams – 2 Midterms & 1 Final**
  - We reserve the right to adjust final course grades for extreme (good or bad) exam performance relative to (group) project grades.
- **Programming Projects to be done in groups of 2**
  - Pick your partners ASAP
- **The course is "front-loaded"**
  - hardest project work is in the first two thirds...

Administrivia Break - Contacts

- **http://inst.eecs.berkeley.edu/~cs186**
- **Prof. Office Hours:**
  - 687 Soda Hall, M 11-12; Th 1-2
  - or by arrangement: `franklin@cs.berkeley.edu`
- **TAs** (Office Hours, locations TBD – see web page)**:**
  - Eirinaios Michelakis
  - Daisy Wang
  - and, if we're lucky... Eugene Wu

- **NO Discussion Sections This Week!**
- **Cancelling Tues Section (starting with 3)**
- **More details on Thursday**

More Administrivia

- **Textbook**
  - Ramakrishnan and Gehrke, 3rd Edition
  - Today's lecture covers Chapter 1 in R&G
  - Read Ch 3 (The Relational Model) for next class.
- **Grading, hand-in policies, etc. will be on Web Page**
- **Cheating policy: zero tolerance**
  - We have the technology...
- **Team Projects (subset of projects)**
  - Teams of 2
- **Class bulletin board - ucb.class.cs186 and blog**
  - read it regularly and post questions/comments.
  - mail broadcast to all TAs will not be answered
  - mail to the cs186 course account will not be answered