Logistics

Instructor:

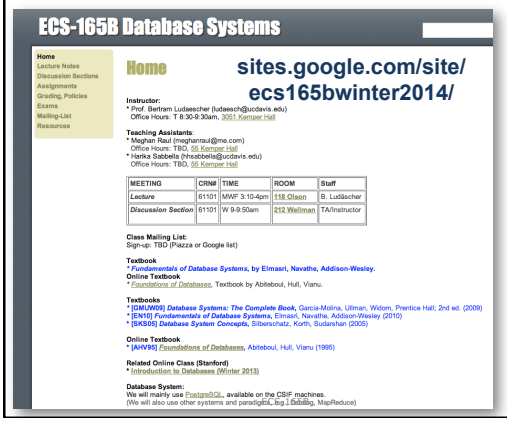
- Prof. Bertram Ludaescher (ludaesch@ucdavis.edu)
  - Office Hours**: T 8:30-9:30am, 3051 Kemper Hall**

Teaching Assistants:

- Meghan Raul (meghanraul@me.com)
- Harika Sabbella (hhsabbella@ucdavis.edu)
- Office Hours: TBD, 55 Kemper Hall

Discussion Sections:

- W 9-9:50am, 212 Wellman (TA / Instructor)
- **On demand** (not before **Week 2**)



More Logistics

Class page:

- https://sites.google.com/site/165bWinter2014

Class Mailing List:

Sign-up:

- https://piazza.com/ucdavis/winter2014/ecs165b/

Textbooks

- [GMUW09] Database Systems: The Complete Book, Garcia-Molina, Ullman, Widom, Prentice Hall; 2nd ed. (2009)
- [EN10] Fundamentals of Database Systems, Elmasri, Navathe, Addison-Wesley (2010)
- [SKS05] Database System Concepts, Silberschatz, Korth, Sudarshan (2005)

165A Course Topics **(pre-req for 165B)**

- Database Design, **E/R Model**
- Relational Model**, Relational Algebra**
- **SQL** (Structured Query Language)
- **Integrity Constraints**
- Storage structures, Indexing
- **Query Processing**
- Transactions
- Additional Topics & Current Trends
  - Logic/declarative queries ("**Datalog**")

Prerequisite Screening

- ECS-165**B** prereq: ECS-165**A**
- Campus is now a bit stricter about screening prereqs
- If you haven't done 165A (or didn't do so well), you might still take 165B
- I will email individuals who failed the pre-req screening.

165B Overview

- **Database Foundations** ("Theory")
  - Design Theory for Relational Databases (Normalization)
  - Recursive queries (Datalog, SQL-with-recursive)
  - Querying (XML) trees and graphs
- **Advanced & Hands-on Topics** ("Practice")
  - XML data management
  - Online Analytical Processing (OLAP), Data Cubes
  - Map-Reduce Parallelism Framework & "Big Data"
  - Other trends (NoSQL)

165B Course Topics (tentative)

- **Database theory:**
  - recursive queries, data integration
  - DB design: normalization
  - data provenance
- **Semistructured data on the web (XML)**
  - DTDs, XML Schema
  - XPath
  - XQuery
  - XSLT
- **Advanced database topics**
  - OLAP (vs OLTP)
  - Big Data, parallel processing e.g. MapReduce
  - Specialized topics (Fusion Tables? NoSQL?)

DB-Class @ Stanford

165B Course Topics

- Focus is on
  - **Foundations**
    - DB Theory (Normalization) and Datalog
  - **Practical experience** with SQL, XML, ….
    - We'll use PostgreSQL
      - A "real" (full-featured), scablable DBMS
      - Open source, available @CSIF and @home!
        - » Other systems: Oracle, SQL-Server, MySQL, SQLite, …
        - » Embedded SQL (e.g. with Python)
    - … and some other things …
      - Map-Reduce @ Amazon
- **Individual Assignments (~3-4)**
- **Group Projects (~3)**

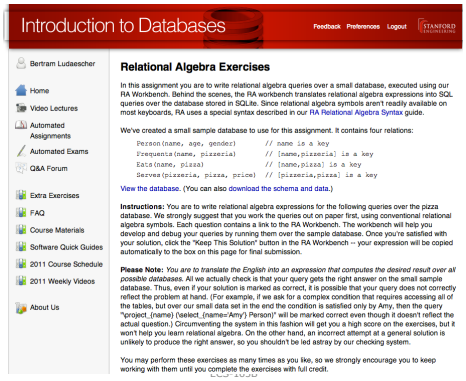
Assignments (Examples)

- Individual Assignments:
  - Datalog, SQL With Recursive
  - XML DTDs, XPath, XQuery, XML Accelerator
  - Design Theory (Normalization)
  - Data Provenance
- Group Projects:
  - Graph Data Visualization Tool
  - XML to Relational Mapping
  - Big Data / Map-Reduce (Amazon EC2)

ECS-199: Special Study for Advanced Undergraduates

- If you did very well in ECS-165A …
- … and you want to get some more additional hands-on experience!
- I can offer a couple (not too many) of these.
- Many possibilities:
  - Complex queries: on trees, graphs; Skylines; OLAP; temporal; spatial; …
  - Harvesting data from the web (DBpedia, …)
  - Analyzing (social) network data
  - Mobile DBs (Android)
  - …

http://class2go.stanford.edu/db/Winter2013



Grading and Policies

- Grading:
  - Base line:
    - 30% Individual Assignments
    - 30% Group Projects
    - 40% Exams
- Academic Conduct
  - Be polite
  - Don't cheat
- Ask when in doubt
- Make good use of the mailing-list/forum

Why study databases / data management?

- Critical to business, government, science, culture, society, …
- Determines success of many corporations (even their existence)
- Many tech companies built on data management (Google, Amazon, Yahoo!, Facebook, …)
- … or offer database products (Microsoft, IBM, Oracle)
- Database systems span major areas of computer science
  - Operating systems (file, memory, process management)
  - Theory (languages, algorithms, complexity)
  - Artificial Intelligence (knowledge-based systems, logic, search)
  - Software Engineering (application development)
  - Data structures (trees, hash-tables)
  - ***… and the DB research community continues to be very active***

Lots of Data Everywhere

- From http://en.wikipedia.org/wiki/Petabyte :
- **History**: According to Kevin Kelly in *The New York Times*, "the entire [written] works of humankind, from the beginning of recorded history, in all languages" would amount to 50 petabytes of data.[1]
- **Computer hardware**: Teradata Database 12 has a capacity of 50 petabytes of compressed data.[2][3]
- **Telecoms**: AT&T has about 16 petabytes of data transferred through their networks each day.[4]
- **Archives**: The Internet Archive contains about 3 petabytes of data, and is growing at the rate of about 100 terabytes per month as of March, 2009.[5][6]
- **Internet**: Google processes about 20 petabytes of data per day.[7]
- **Physics**: The 4 experiments in the Large Hadron Collider will produce about 15 petabytes of data per year, which will be distributed over the LHC Computing Grid.[8]
- **P2P networks**: As of October 2009, Isohunt has about 9.76 petabytes of files contained in torrents indexed globally.[9]
- **Games**: World of Warcraft utilizes 1.3 petabytes of storage to maintain its game.[10]

Science has been changing lately …

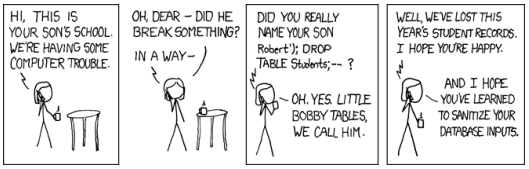
- *"All science is either physics or stamp collecting."*
  - Ernest Rutherford, British chemist & physicist (1871 - 1937)
    [J. B. Birks "Rutherford at Manchester" (1962)]
- That is, from few data, lots of thinking
- … to LOTS OF DATA and ANALYSIS

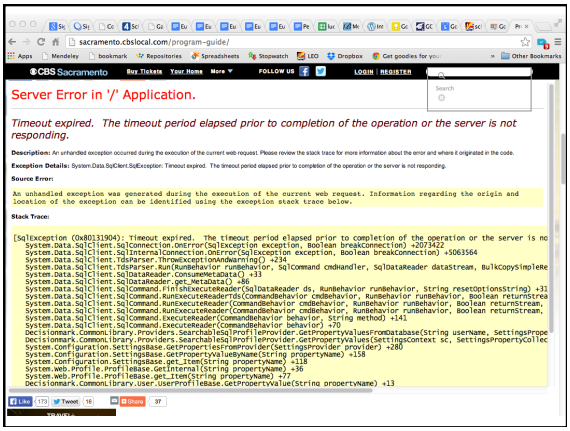
➔ "Data-driven" scientific discovery!

4th *paradigm*, in addition to hypothesis-driven science

Also: Data(bases) can be Yummy!

Exploits of a Mom http://xkcd.com/327/





Relational Query Languages

- SQL — SELECT ... FROM ... WHERE ...
- Relational Algebra (RA) — $\sigma, \pi, \bowtie, \delta, \cup, \setminus$
- Relational Calculus (RC) — $\forall x F,\ \exists x F,\ F \wedge G,\ F \vee G, \neg F$
- **Datalog** — $\approx$ RC + Recursion

EXAMPLE: Given relations `employee(Emp, Salary, DeptNo)` and `dept(DeptNo, Mgr)`, find all (employee, manager) pairs:

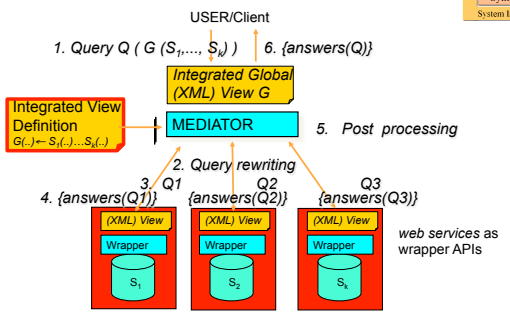
- SQL:
```
SELECT Emp, Mgr
FROM employee, dept
WHERE employee.DeptNo = dept.DeptNo
```
- RA: $\pi_{\texttt{Emp,Mgr}}(\texttt{employee} \bowtie \texttt{dept})$
- RC: $F(\texttt{Emp},\texttt{Mgr}) =$ $\exists \texttt{Salary, DeptNo} : (\texttt{employee(Emp, Salary, DeptNo)} \wedge \texttt{dept(DeptNo,Mgr)})$
- **Datalog**: `boss(Emp,Mgr) ← employee(Emp, Salary, DeptNo), dept(DeptNo,Mgr)`

Datalog

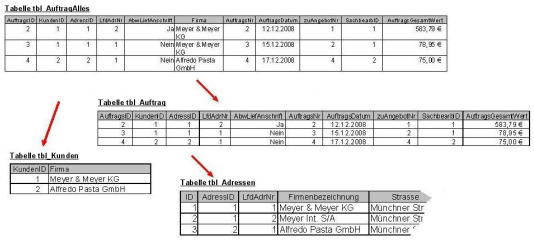
```
grandparent(X, Y) :-
    parent(X, Z), parent(Z, Y).
father(X, Y) :-
    parent(X, Y), male(X).
mother(X, Y) :-
    parent(X, Y), female(X).
brother(X, Y) :-
    parent(P, X), parent(P, Y), male(X), X != Y.
sister(X, Y)  :-
     parent(P, X), parent(P, Y), female(X), X != Y.

ancestor(X, Y) :- parent(X, Y).
ancestor(X, Y) :- parent(X, Z), ancestor(Z, Y).
```


Data Integration (Mediator System)



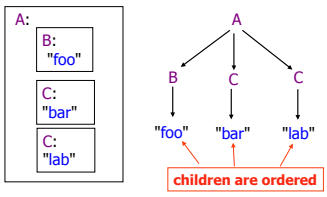
Database Design Theory: Normalization



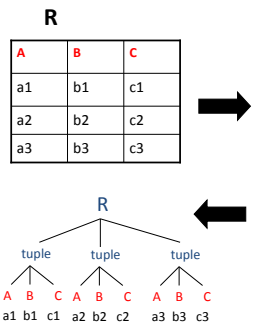
Pure XML -- Instance Model

- XML 1.0 Standard:
  - syntax of well-formed and valid (wrt. a DTD) documents
- implicit data model:
  - nested containers ("boxes within boxes")
  - labeled ordered trees (=a semistructured data model)
  - relational, object-oriented, other data: easy to encode

```
<A>
  <B>foo</B>
  <C>bar</C>
  <C>lab</C>
</A>
```



Relational Structures vs. XML




```
<R>
  <tuple>
    <A> a1 </A>
    <B> b1 </B>
    <C> c1 </C>
  </tuple>
  <tuple>
    <A> a2 </A>
    <B> b2 </B>
    <C> c2 </C>
  </tuple>
  ...
</R>
```

Introduction to XQuery

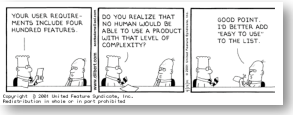
XQuery Tutorial

Peter Fankhauser, Fraunhofer IPSI
Peter.Fankhauser@ipsi.fhg.de
Philip Wadler, Avaya Labs
wadler@avaya.com

Data model

XML

```
<BOOK YEAR="1999 2003">
  <AUTHOR>Abiteboul</AUTHOR>
  <AUTHOR>Buneman</AUTHOR>
  <AUTHOR>Suciu</AUTHOR>
  <TITLE>Data on the Web</TITLE>
  <REVIEW>A <EM>fine</EM> book.</REVIEW>
</BOOK>
```



XQuery

```
element BOOK {
  attribute YEAR { 1999, 2003 },
  element AUTHOR { "Abiteboul" },
  element AUTHOR { "Buneman" },
  element AUTHOR { "Suciu" },
  element TITLE { "Data on the Web" },
  element REVIEW { "A", element EM { "fine" }, "book." }
}
```

XQuery Examples

Titles of all books published before 2000

```
/BOOKS/BOOK[@YEAR < 2000]/TITLE
```

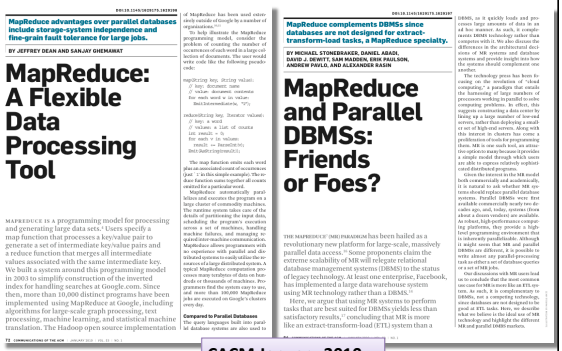
Year and title of all books published before 2000

```
for $book in /BOOKS/BOOK
where $book/@YEAR < 2000
return <BOOK>{ $book/@YEAR, $book/TITLE }</BOOK>
```

Books grouped by author

```
for $author in distinct(/BOOKS/BOOK/AUTHOR) return
  <AUTHOR NAME="{ $author }">{
    /BOOKS/BOOK[AUTHOR = $author]/TITLE
  }</AUTHOR>
```


MapReduce vs Parallel Databases



MapReduce advantages over parallel databases include storage-system independence and fine-grain fault tolerance for large jobs.

BY JEFFREY DEAN AND SANJAY GHEMAWAT

MapReduce: A Flexible Data Processing Tool

MapReduce complements DBMSs since databases are not designed for extract-transform-load tasks, a MapReduce specialty.

BY MICHAEL STONEBRAKER, DANIEL ABADI, DAVID J. DEWITT, SAM MADDEN, ERIK PAULSON, ANDREW PAVLO, AND ALEXANDER RASIN

MapReduce and Parallel DBMSs: Friends or Foes?

CACM January 2010