Principles of Database Management Systems

CSE 544

Introduction
March 29th, 2000

Staff

- Instructor: Alon Levy
  - Sieg, Room 310, alon@cs.washington.edu
  - Office hours: by appointment.
- TAs: Bart Niswonger and Stefan Saroiu
  - Office hours: also by appointment.
- Mailing list: cse544@cs
- Web page: (a lot of stuff already there)
  http://www.cs.washington.edu/education/courses/544/00sp/

Course Times

- In general, WF, 12-1:20pm (with a 3 minute breather in the middle).
- Special dates:
  - Mondays, April 3, 10, 24.
- No classes on week of May 15th.

Goals of the Course

- Purpose:
  - Foundations of database management systems.
  - Issues in building database systems.
  - Introduction to current research issues in databases.
  - Have fun: *databases are not just bunches of tuples.*

Grading

- Homeworks: 35%
  - Very little regurgitation.
  - Meant to be challenging (I.e., fun).
- Project: 50%
  - More later.
- Participation: 15%
- No Exams.

Textbook

- **Database Management Systems**, Ramakrishnan and Gehrke.

Other Useful Texts

- Pair of books by Ullman, Widom and Garcia-Molina
- *Foundations of Databases* (Abiteboul, Hull & Vianu)
- *Parallel and Distributed DBMS* (Ozsu and Valduriez)
- *Transaction Processing* (Gray and Reuter)
- *Database Systems* (Silberschatz, Korth and Sudarshan)
- *Data and Knowledge based Systems* (volumes I, II) (Ullman)
- *Readings in Database Systems* (Stonebraker and Hellerstein)
- Proceedings of SIGMOD, VLDB, PODS conferences.

Prerequisites

Real Prerequisites

- Operating systems
- Data structures and algorithms
- Distributed systems
- Complexity theory
- Mathematical Logic
- Knowledge Representation
- User interface design
- Programming languages
- Artificial Intelligence (Search)
- *Greek, Hebrew, French, Romanian*

Why Use a DBMS?

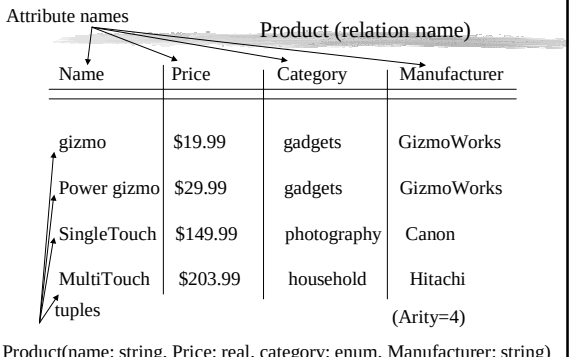
All programs manipulate data, so why use a database?

- Large amounts of data (Giga's, Tera's)
- Data is very structured
- Persistent data
- Valuable data
- Performance requirements
- Concurrent access to the data
- Restricted access to data

Functionality of a DBMS

- Persistent storage management
- Transaction management
- Resiliency: recovery from crashes.
- Separation between logical and physical views of the data.
  - High level query and data manipulation language.
  - Efficient query processing
- Interface with programming languages

Terminology

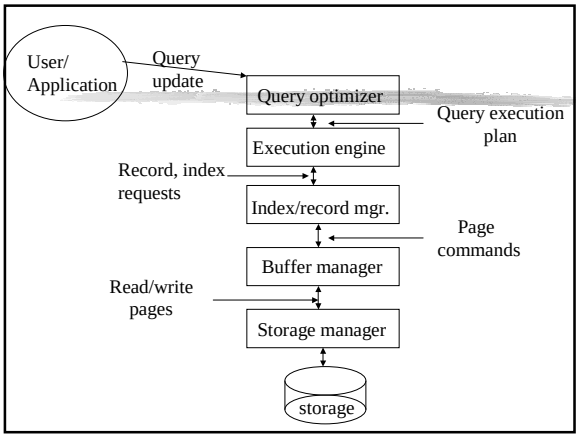


Product(name: string, Price: real, category: enum, Manufacturer: string)

Querying a Database

```
SELECT  S.sname, phone
FROM Purchase P, Person Q
WHERE P.buyer=Q.name AND
        Q.city='seattle' AND
        Q.phone > '5430000'
```

- SQL (Structured Query Language)
- An acquired taste…
- Datalog: kinder, gentler language

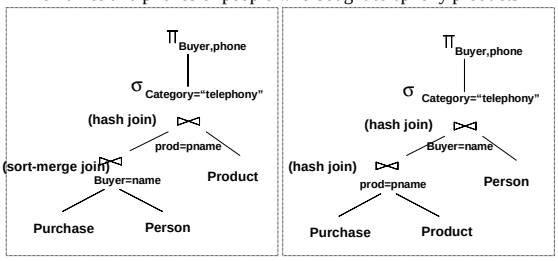


Storage Management

- Becomes a hard problem because of the interaction with the other levels of the DBMS:
  - What are we storing?
  - Efficient indexing, single and multi-dimensional
  - Exploit "semantic" knowledge
- Issue: interaction with the operating system. Should we rely on the OS?

Query Execution Plans

Find names and phones of people who bought telephony products



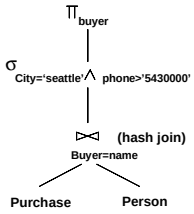
Imperative programs for evaluating queries. Many choices to make.

Query Optimization

Goal:

Declarative SQL query → ***Imperative query execution plan:***

```
SELECT  S.sname,phone
FROM Purchase P, Person Q
WHERE P.buyer=Q.name AND
    Q.city='seattle' AND
    Q.phone > '5430000'
```



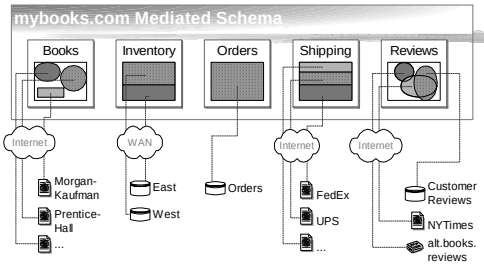
Plan: Tree of R.A. ops, with choice of alg for each op.

Ideally: Want to find best plan. Practically: Avoid worst plans!

TP and Recovery

- For efficient use of resources, we want concurrent access to data.
- Systems sometimes crash.
- A "real" database guarantees **ACID**:
  - Atomicity: all or nothing of a transaction.
  - Consistency: always leave the DB consistent.
  - Isolation: every transaction runs as if it's the only one in the system.
  - Durability: if committed, we really mean it.
- *Do we really want **ACID**?*

Data Integration



Uniform query capability across *autonomous, heterogeneous* data sources on LAN, WAN, or Internet

XML: Semi-structured Data

eXtensible Markup Language:

- *Emerging format for data exchange on the web and between applications*.

```
<db>
 <book>
    <title>Complete Guide to DB2</title>
    <author>Chamberlin</author>
 </book>
 <book>
    <title>Transaction Processing</title>
    <author>Bernstein</author>
    <author>Newcomer</author>
 </book>
 <publisher>
   <name>Morgan Kaufman</name>
   <state>CA</state>
 </publisher>
</db>
```

Database Industry

- Relational databases are a great success of theoretical ideas.
- Oracle has a market cap of over $200B
- Other players: IBM, MS, Sybase, Informix
- Trends:
  - warehousing and decision support
  - data integration
  - XML, XML, XML.

Course (Rough) Outline

- The basics: (quickly)
  - The relational model
  - SQL
  - Views, integrity constraints
- XML
- Physical representation:
  - Index structures.

Course Outline (cont)

- Query execution: (Zack Ives)
  - Algorithms for joins, selections, projections.
- Query Optimization
- Data Integration
- semi-structured data
- Transaction processing and recovery (Phil Bernstein)

Projects

- Goal: identify and solve a problem in database systems.
- (almost) anything goes.
- Groups of 2-3
- Groups assembled end of week 2;
- Proposals, end of week 3.
- Touch base with me: *every* two weeks.
- Example projects on web site.
- Start Early.