CS2Bh: Current Technologies

Introduction to XML and Relational Databases

Spring 2005

- Instructor: Wenfei Fan
- Office: AT 2.07
- Office Hour: Wednesday 2:00-3:00
- Email: wenfei@inf.ed.ac.uk

- Web: http://homepages.inf.ed.ac.uk/wenfei/cs2/home.html

What the part is about -- data

Given a large collection of data,

- how to organize/represent the data? – data models
  - critical to storage, retrieval and management of the data
- how to efficiently retrieve information from the data? – queries
- how to efficiently manage the data? – updates

Data is typically found in documents and databases

- XML documents: XML is the prime standard for data exchange
- Relational databases: relational database systems are the dominant commercial database systems

This part gives an introduction to XML and databases

Documents vs. Databases

- Documents are typically small, while databases can be large
- Documents are usually static, whereas databases are typically dynamic
- A documents has an implicit structure, while a database has an explicit structure
- Documents are usually semi-structured (without an explicit type), while databases are structured, constrained by a schema
- Documents are human friendly, while databases are machine friendly
- Concerns about documents include presentation, editing, character encoding, language; while databases focus on models, queries, concurrency control, performance

Why study XML?

- Huge demands for data exchange
  - Across platforms
  - Across enterprises
- Huge demands for data integration
  - Heterogeneous data sources
  - Data sources distributed across different locations
- XML (eXtensible Markup Language) has become the prime standard for data exchange on the Web and a uniform data model for data integration.

Why not HTML?

Example:

- ✓ Amazon publishes a catalog for book sale
  - Data source: a relational database
  - Publishing: HTML pages generated from the relational database
- ✓ Customers want to query the catalog data:
  - they can only access the published Web pages (and hence need a parser)
  - They are only interested in information about books on Iraqi WMD authored by Bush -- in SQL:

```
select   B
from     book  B
where    B.title contains "WMD"   and   B.author = "Bush"
```


What is wrong with HTML?

HTML (HyperText Markup Language)

```
<h3>  Book  </h3>
      <ul>
      <il> <i>  I found WMD in Iraq </i>  G. Bush <br>
                <b>   2003 </b>
      <il> <b>  How to cheat  </b>  Bill Clinton <br>  …
      </ul>
```

A minor format change to the HTML document may break the parser – and yield wrong answer to the query

Why? HTML tags are

- ✓ predefined and fixed
- ✓ describing display format rather than structure of data

HTML is good for presentation (human friendly), but does not help automatic data extraction by means of programs

An XML solution

XML (eXtensible Markup Language):

```
<book >
    <title>   I found WMS in Iraq </title>
    <author>  G. Bush  </author>
    <year>  2003  </year>
</book>
<book  id  =  “B2” >
    <title>   How to cheat  </title>
    <author   Bill Clinton  </author>
</book>
. . .
```


XML vs. HTML

✓ XML tags:

- user-defined
- describing the structure of the data

XML is both human friendly and computer friendly.

✓ HTML is human friendly but not computer friendly;

HTML tags:

- predefined and fixed
- describing display format rather than structure of data indented for human consumption

What we shall learn about XML

- ✓ XML basics: elements, attributes, tree model
- ✓ Document Type Definition
  - "types": element type definition
  - "constraints": ID/IDREF
- ✓ XML query Languages
  - XPath
  - XQuery, XSLT

Why study databases?

We run into problems when

- ✓ The structure of data gets complicated
- ✓ The database gets large
- ✓ Efficiency of query evaluation, storage, management
- ✓ Many people want to use/update the data simultaneously
- ✓ The data must satisfy certain types of consistency constraints
  - The data should be protected, i.e., security policies should be enforced such that different users have different permissions to access different subsets of the data
  - The data must be restored to a consistent state if the system crashes while changes are being made.

Databases

- A ***database*** is a collection of data, typically containing the information about one or more related organizations.
- A ***database management system*** (***DBMS***) is a software package designed to store and manage databases.

The database community has developed in the past 40+ years:

- Query languages, query processing techniques
- Integrity constraints for consistency of the data
- Database views, updates
- Secondary storage, indexing
- Concurrency control, recovery, security
- . . .

data models

- A **data model** is a collection of concepts for describing data.

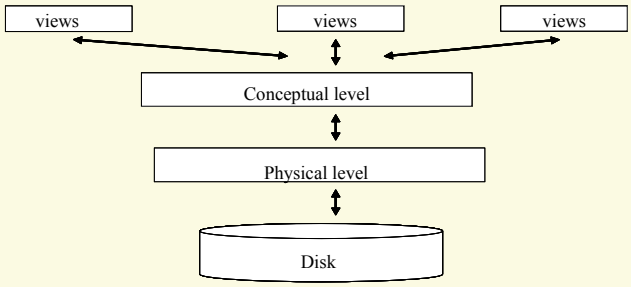
  Data model in database vs. type system in programming language.
- A **schema** is a description of a particular collection of data, using the given data model.

  Schemas in database vs. types in programming language.
- An **instance** of a schema (database) is the collection of information stored in the database at a particular moment.

  Instances of schemas vs. values of types

Database architecture

- It is common to describe databases in two ways:
  - The logical structure: what users see. The program or query language interface.
  - The physical structure: how files are organized. What indexing mechanisms are used.
- Further it is traditional to split the "logical" level into two components: the overall database design (conceptual level) and the views that various users get to see.



An example relational schema

A university database schema:

Students(name: string, sid: string, email: string, gpa: real)

Courses(title: string, cid: string, credits: integer)

Enroll(sid: string, cid: string, grade: string)

Name	Sid	email	gpa
John	0001	John@nimbus.ocis	3.0
Joe	0002	Joe@nimbus.ocis	3.6
Mary	0003	Mary@nimbus.ocis	2.8
Grace	0004	Grace@nimbus.ocis	4.0

An example relational query

Find the names of students with gpa greater than 3.0

```
SELECT   name
FROM     Students
WHERE    gpa > 3.0
```


What we shall learn about relational databases

- ✓ Relational data model
- ✓ Relational query languages
  - Relational algebra
  - SQL

There is much more to learn:

- ✓ Query languages: relational calculus, QBE, Datalog
- ✓ Database design: the ER model, functional dependencies, normal forms
- ✓ The physical structure of DBMS
- ✓ Query optimization
- ✓ Concurrency control and recovery
- ✓ Security, …

Traditional data models

- ✓ The relational data model – the dominant model

  Relational database systems: SQL server, Oracle, Sybase
- ✓ Object-oriented data model.

  Object-oriented database systems: ObjectStore, O2, …
- ✓ Object-relational model.

  Object-relational database systems: UniSQL, Informix Universal Server, etc.
- ✓ Semantic data model, e.g., the **ER** model.
- ✓ Other data models:
  - Network model
  - Hierarchical model
  - The functional data model
  - …

Summary: what this part is to cover

An introduction to XML and XML querying

- XML basics
- Document Type Definition (DTD)
- XPath
- XQuery and XSLT

An introduction to databases

- Relational data model
- Relational algebra
- SQL