Databases and DBMS’s

- A database is a large, integrated collection of data
- A database management system (DBMS) is a software system designed to store and manage a large amount of data
  - Declarative interface to define data stored, add data, update data, and query data
  - Efficient querying
  - Concurrent users
  - Reliable storage and crash recovery
  - Access control...

Outline

- Overview of databases and DBMS’ s
- Course topics and requirements
Earlier Database Applications

• OnLine Transaction Processing (OLTP)
  – Data with many small items, many queries, many updates
  – E.g., banking, airline reservations
  – E.g., university databases

Recent Database Applications

• OnLine Analytical Processing (OLAP), also known as Data Warehousing
  – Large amounts of data over years, complex queries, designed for analysis and reporting
  – Sales data analysis, e.g., Walmart, Target, ...
  – Fraud analysis, e.g., credit card use, insurance

More Recent DB Applications

• Electronic commerce
  – E.g., amazon.com, ebay.com
  – Integrating thousands of catalogs and orders

• Social networking
  – E.g., facebook.com, myspace.com, with 100’s million users or more at a popular site
  – Supporting many users, their profiles, communications
  – Real-time analysis of user behaviors

How does one build a database?

Example: The Internet Shop*

• DBDudes Inc.: a well-known database consulting firm
• Barns and Noble (B&N): a large bookstore specializing in books on horse racing
• B&N decides to go online but needs help
• Step 0: DBDudes makes B&N agree to
  – pay steep fees and
  – schedule a lunch meeting for requirements analysis

* The example and all related material was taken from “Database Management Systems,” Edition 3.
Step 1: Requirements Analysis

- "I’d like my customers to be able to browse my catalog of books and place orders online."
  - Books:
    - For each book, B&N’s catalog contains its ISBN number, title, author, price, year of publication, …
  - Customers:
    - Most customers are regulars with names and addresses registered with B&N.
    - New customers must first call and establish an account.
  - On the new website:
    - Customers identify themselves before browsing and ordering.
  - Shipping:
    - For each order, B&N ships all copies of a book together once they become available.

Step 2: Conceptual Design

- A high level description of the data in terms of the Entity-Relationship (ER) model.
  - Design review:
    - What if a customer places two orders of the same book in one day?
    - Modification: add "ordernum" to Orders.

Step 3: Logical Design

- Mapping the ER diagram to the relational model

  CREATE TABLE Books
  (isbn CHAR(10),
   title CHAR(80),
   author CHAR(80),
   qty_in_stock INTEGER,
   price REAL,
   year INTEGER,
   PRIMARY KEY(isbn))

  CREATE TABLE Customers
  (cid INTEGER,
   cname CHAR(80),
   address CHAR(200),
   PRIMARY KEY(cid))

  CREATE TABLE Orders
  (ordernum INTEGER,
   isbn CHAR(10),
   cid INTEGER,
   cardnum CHAR(16),
   qty INTEGER,
   order_date DATE,
   ship_date DATE,
   PRIMARY KEY(ordernum, isbn),
   FOREIGN KEY (isbn) REFERENCES Books,
   FOREIGN KEY (cid) REFERENCES Customers)

  CREATE VIEW OrderInfo
  (isbn, cid, qty, order_date, ship_date)
  AS SELECT O.isbn, O.cid, O.qty,
           O.order_date, O.ship_date
            FROM Orders O

- Access control: use views to restrict the access of certain employees to customer sensitive information

Step 4: Schema Refinement

- Redundant Storage!

Orders

<table>
<thead>
<tr>
<th>ordernum</th>
<th>isbn</th>
<th>cid</th>
<th>cardnum</th>
<th>qty</th>
<th>order_date</th>
<th>ship_date</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>0-07-11</td>
<td>123</td>
<td>40241160</td>
<td>2</td>
<td>Jan 3, 2006</td>
<td>Jan 6, 2006</td>
</tr>
<tr>
<td>120</td>
<td>1-12-23</td>
<td>123</td>
<td>40241160</td>
<td>1</td>
<td>Jan 3, 2006</td>
<td>Jan 6, 2006</td>
</tr>
<tr>
<td>120</td>
<td>0-07-24</td>
<td>123</td>
<td>40241160</td>
<td>3</td>
<td>Jan 3, 2006</td>
<td>Jan 26, 2006</td>
</tr>
</tbody>
</table>

Step 5: Internet Application Development

- Presentation tier
  - Client Program (Web Browser)
  - HTML
  - JavaScript
  - Cookies

- Application logic tier
  - Application Server (Apache Tomcat, …)
  - PHP, JSP, Servlets, XSLT

- Data management tier
  - Database System (MySQL, DB2, …)
  - Relational, XML

An Example Internet Store

The Internet BookShop

Welcome to the Internet Bookshop. We have just launched our Internet Bookstore. Now you can browse and purchase books from the comfort of your favorite Internet browser. And of course, membership is free of charge! See our up-to-date list of all the books and prices.

And of course, membership is free of charge! See our up-to-date list of all the books and prices. And of course, membership is free of charge! See our up-to-date list of all the books and prices.
Example SQL Queries

**Search Page**
```
SELECT isbn, title, author, price
FROM Books
WHERE author = '<SearchString>'
ORDER BY title
```

**Login Page**
```
SELECT cid, username, password
FROM Customers
WHERE username = '<SpecifiedUsername>'
```

Step 6: Physical Design

- Auxiliary data structures, **indexes**, to speed up searches, e.g., B+-tree, R-tree, hash index

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Price</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spectacular</td>
<td>Edward L. Bowen</td>
<td>29.95</td>
<td>2003</td>
</tr>
<tr>
<td>Bid</td>
<td>Dan Mearns</td>
<td>24.95</td>
<td>2000</td>
</tr>
<tr>
<td>Legacies of the Turf</td>
<td>Timothy Capps</td>
<td>16.95</td>
<td>2001</td>
</tr>
</tbody>
</table>

Hash Index on Books.author

What is inside DBMS?

- **DBMS Architecture**
  - Query Processor
  - Transactional Storage Manager
  - Disk Manager
  - Buffer Manager
  - Log Manager
  - Access Methods

Course Topics

- **Fundamentals**
  - Data modeling
  - Relational design
  - Query languages (relational algebra + SQL)
- **Database implementations**
  - Storage and indexing
  - Query processing and optimization
  - Transaction management
- **Advanced technologies**
  - Web application development (PHP + MySQL)
  - Big data analytics (if time permits)

Prerequisites

- **Prerequisites:**
  - CMPSCI 220/230: Computer architecture & programming
  - CMPSCI 311: Introduction to Algorithms
- **Or, upon consent of the instructor**
  - Data structures and algorithms
  - Sufficient programming experience
Yanlei Diao, University of Massachusetts Amherst

Grading

- Homework: 30%
- Midterm: 20%
- Final: 25%
- Course Project: 20%
- Attendance & Participation: 5%

Homework: 30%

- 5 assignments throughout the semester
  - Written problem sets
  - Programming exercises with query languages including SQL
- Posted on the assignments page
- Dates that each assignment: see the schedule page
- Submission via Moodle before 11 am on due date
- Policy on late submissions:
  - Illness with a doctor’s note, or with the instructor’s content before the due date. No exception!

Project: 20%

- General theme: build a web application using PHP + MySQL backend
- Groups of 3
- Project work will include:
  - Schema design
  - DB implementation
  - Web site design
- Multiple milestones & deliverables
  - See the schedule page
  - See the projects page for details
- Submission: via Moodle, before 11 am on due date

Exams

- Midterm (20%)
  - In-class, closed-book exam
  - At the beginning of the 9th week
- Final (25%)
  - Closed-book exam
  - Waiting to be scheduled in the final exam period

Attendance & Participation (5%)

- Attend every class
- Ask questions, contribute to answers
- Participate in in-class exercises and discussions

Academic Honesty

- All submitted work must be your own!
  - Although students are encouraged to study together, each student is expected to produce his or her own solutions to the homework problems.
  - Copying or using sections of someone else’s program or assignment (even if it has been modified by you), or copying a solution from an external source, is not acceptable.
  - The University guidelines for academic misconduct: http://www.umass.edu/dean_students/codeofconduct/acadhonesty/
  - The staff of CS 445 will be vigorous in enforcing them.