CS 2550 / Spring 2006
Principles of Database Systems

01 – Introduction

Alexandros Labrinidis
University of Pittsburgh

Quick Facts about the course

- When: Tue & Thu 2:30pm – 3:45pm
- Where: 5313 SENSQ
- Instructor: Alexandros Labrinidis (TA: Subrata Acharya)
- What: Database System Concepts **5th Edition**, 2005 by Silberschatz, Korth, Sudarshan
- Grading: assignments: **50%** (x 4-6)
  exam #1: **25%** (Feb 21)
  exam #2: **25%** (Apr 18)

Administrative

- web page: http://db.cs.pitt.edu/courses/cs2550/spring2006
  - also link from instructor's home page and main cs web page
  - check often!
- Email to `cs2550-staff@cs.pitt.edu` for all emails to instructor/TA
- class mailing list (make sure to put your email address on sign-up sheet)

What is a Database Management System?

- A Database Management System (DBMS) is a collection of interrelated data and a set of programs to access those data
- Database = collection of data
- Primary goal of DBMS: store database information
  - **conveniently**
  - **efficiently**

When to use a Database System?

- DBMS designed for managing large bodies of information
  - example: personal list of phone numbers VS phone book
- "manage" implies:
  - Data Storage
  - Data Manipulation capabilities
  - Safety guarantees
    - Surviving system failures
    - Preventing unauthorized access
    - Allowing for concurrency

Examples of DBMS usage

- Airlines – reservations and schedules (expedia.com)
- Banking – customer info and accounts (WellsFargo.com)
- Universities – student info, grades
- Government – taxes, budgets, legislation, census
- Sales – inventory, customer info (Amazon.com, eBay)
- Credit Cards – transactions, customer info
- Newspapers – track subscribers, advertising revenue
- Financial – stock prices, portfolio info
- Telecommunications – record of calls made
- many more!

When not to use a Database System?

- Disadvantages of a DBMS:
  - Expensive to buy
  - Expensive to maintain (need administrator)
  - "Expensive" to run (needs significant resources)
- Hence, it is "over-kill" to use when:
  - The database is small
  - The database has simple structure
  - The application is simple and not expected to change
  - Can tolerate failures
  - Do not require multiple, concurrent users

Data Management

- Information is crucial to most organizations
  ➔Large body of concepts & techniques for data management
- Purpose of the CS2550 course:
  - Provide in-depth knowledge of Database Management System design
  - Provide significant breadth, covering many aspects of concepts and techniques
  - Provide hands-on experience, with projects designed to
    - Understand principles
    - Evaluate implementation options
    - Implement techniques

Roadmap

- What is a DBMS
- Uses of a DBMS
- Database System vs File System
- Data Abstraction
- Data Models
- Database Languages
- Database Users
- Transaction Management
- Historical Perspective

Database Systems vs File Systems

Disadvantages of using file systems for data storage:

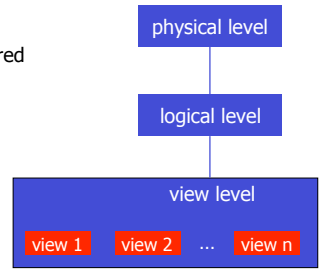
- Data redundancy and inconsistencies
- Difficulty in accessing data (when application changes)
- Data may be spread over multiple files
- Integrity problems (hard-code consistency constraints)
- Atomicity problems (on failures)*
- Concurrent access anomalies*
- Security problems (authorization)

Data Abstraction

- Provide users with abstract view of data, thus hiding certain details

Three abstraction levels:

- Physical Level
  - How the data are actually stored
- Logical Level
  - What data are stored
- View Level
  - Describe part of database



Data Abstraction (cont.)

- Examples of Views:
  - Given a database of universities, show only public ones
  - Given an employee database, show all data except for salary
- Views provide:
  - Convenience
  - Security
- Database Schema = overall design of database
  - E.g., variable declaration
- Database Instance = collection of info stored in db
  - E.g., value of variable

Roadmap

- What is a DBMS
- Uses of a DBMS
- Database System vs File System
- Data Abstraction
- Data Models
- Database Languages
- Database Users
- Transaction Management
- Historical Perspective

Data Models

- A collection of conceptual tools for describing data, data relationships, data semantics and consistency constraints
- Entity-Relationship Model (E-R model)
  - entities = objects in the real world
  - attributes describe entities
  - relationship = association among several entities
- E-R diagram = graph of overall logical structure of db*
  - rectangles → entity sets
  - ellipses → attributes
  - diamonds → relationships among entity sets
  - lines → link attribute to entity set OR entity set to relationship

Relational Model

- The relational model uses a collection of tables to represent both data and relationships among data.
- Example:
  - customer-account-depositor tables
- Important property of relational model: **CLOSED-NESS**
  - Everything is a table
  - Operations on tables, return tables

Database Languages: DDL

- Data Definition Language (DDL)
  - specify database schema
- Example:
  - create table account
    (account-number char(10),
    balance integer)
  - execution creates table, updates data dictionary
- Data dictionary contains metadata (data about the data)
  - Data dictionary is consulted before reading/modifying data
- Consistency constraints are defined using DDL
  - Checked before updating data

Database Languages: DML

- Data Manipulation:
  - Retrieval of information (=query)
  - Insertion of new information
  - Deletion of information from the db
  - Modification of information in the db
- Data Manipulation Language (DML) enables users to access or manipulate data
  - Procedural DMLs
    - specify what data needed and how to get it
  - Declarative DMLs (non-procedural)
    - only specify what data needed

Database vs DBMS vs Application

- DBMS
  - Database Management System
- Database
  - collection of data in DBMS
- Application Program
  - program that is used to interact with the DBMS
- Database access for application programs:
  - Program interface for host language, OR
  - Extend host language to embed DML statements

Roadmap

- What is a DBMS
- Uses of a DBMS
- Database System vs File System
- Data Abstraction
- Data Models
- Database Languages
- Database Users
- Transaction Management
- Historical Perspective

Database Users

- Naïve users
  - Invoke application programs
- Application programmers
  - Develop db-enabled application programs
- Sophisticated users
  - Interact with DBMS using db query language
  - Advanced applications: OLAP/Data Mining
- Specialized users
  - Write specialized db applications, complex queries

Database Administrators (DBAs)

- DBA = person who has central control of both data & programs that access those data
- The DBA can:
  - Define the db schema
  - Define storage structures and access methods
  - Modify the schema and physical organization
  - Grant/revoke authorization for data access
  - Perform routine maintenance:
    - **Backups!**
    - Upgrade hardware (mostly disks)
    - Monitor performance to identify problems

Transaction Management

- Transaction = several operations on the database that form a single unit of work
  - Example: transfer $50 from account A to account B
- **Atomicity**
  - all-or-none requirement (in the presence of failures)
- **Consistency**
  - database is in consistent state before and after execution of the transaction (by itself)
- **Isolation**
  - each transaction unaware of others running concurrently
- **Durability**
  - after a transaction completes, changes are permanent

ACID properties

- Ensuring transaction consistency is the responsibility of the application programmer
- Ensuring atomicity, isolation and durability is the responsibility of the DBMS.
  - esp. hard in the presence of failures
- Recovery
  - restore database in the state before uncompleted transaction started execution
- Concurrency Control

Historical Perspective

- First database system
  - clay tablets to record transactions of goods (6000 years ago)
- Clay → Papyrus → Parchment → Paper
- 1890, Hollerith: punched cards for data processing
- 1950s: computers / magnetic tapes for data storage
- 1960s/70s: magnetic disks
- 1980s: commercial relational DBMSs
- Early 1990s: design of SQL
- Late 1990s: WWW boom
  - Databases are deployed extensively
  - Databases are online and available 24x7

Overview

- What is a DBMS
- Uses of a DBMS
- Database System vs File System
- Data Abstraction
  - Physical, Logical, View levels
- Data Models
- Database Languages
- Database Users
- Transaction Management
  - Atomicity, Consistency, Isolation, Durability
- Historical Perspective

Questions?

Do not forget to:

- check the web page frequently
  http://db.cs.pitt.edu/courses/cs2550/spring2006
  - Should be ready next week