CS145 Lecture Notes #1
Introduction

What is Database Management System (DBMS)?

A system for providing *efficient*, *convenient*, and *safe multiuser* storage of and access to *massive* amounts of *persistent* data

Example: Banking System

Data = Information on branches, accounts, customers, interest rates, transaction histories, etc.

- **Massive:** Many gigabytes at least for big banks, more if keep history of all transactions, even more if keep images of checks $\leadsto$ far too big for memory
- **Persistent:** Data outlives programs that operate on it
- **Multiuser:** Many people/programs accessing same database, or even same data, simultaneously $\leadsto$ needs careful control
  Example: simultaneous withdrawals

  Appears similar to concurrent programming problems
  —but data not main-meory variables
  Appears similar to file system concurrent access
  —but want to control at finer granularity
- **Safe:**
  - From system failures
    Example: balance transfer
  - From malicious users

- **Convenient:**
  - Simple commands to manipulate data: debit account, get balance, write statement, transfer funds, etc.
  - Complex, unpredicted queries: get large transactions that resulted in low balances during the last statement period, etc.
- **Efficient:** Don't search all files in order to: get balance of one account, get all accounts with low balances, etc. Massive data $\leadsto$ DBMS is carefully tuned for performance

DBMS Software

- "Big three" DBMS companies: Oracle, Informix, and Sybase are among the largest software companies in the world
- IBM also plays, with its DB2
- Microsoft plays, with its SQLServer
- mSQL is free, and very popular on Web
- Traditional DBMS are challenged by newer, object-oriented DBMS: O2, ObjectStore, Objectivity, etc.

Some Terminology and Concepts

Data Model

Describes conceptual structuring of data stored in the database.
Example: data is stored as a table of records, each with acct#, name, balance, etc. Another example: data is a graph where nodes represent cities, edges represent airline routes.

Schema vs. Data

- Schema describes how database is to be structured, defined at set-up time, rarely changes (also called *metadata*)
- Data is actual “instance” of database, changes rapidly

$\leadsto$ Compare to types and variables in programming language

People

- Database designer: sets up schema to model real-world domains (145)
- Database user: queries and modifies data (145)
- Database programmer: develops tools and applications allowing users to access the database (145)
- DBMS administrator: “root” of DBMS (145, 245 and lots of manual reading)
- DBMS implementor: builds the DBMS (245, 346, 347)