CSC 540
Database Management Concepts and Systems

Sections 001 and 601

Spring 2020

This presentation uses slides and lecture notes available from
http://infolab.stanford.edu/~ullman/dscb.html#slides

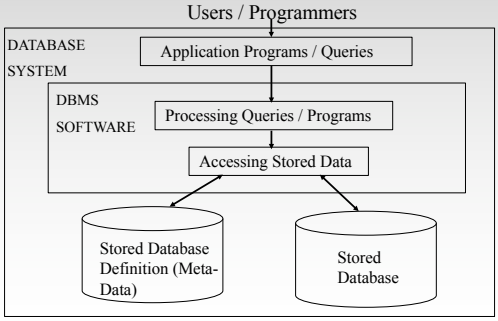
1

Scope of this Course

- Directed at computer science *graduate* students
- Emphasizes concepts and theory
- Requires design and development of a database application
- Implementation-specific details are not the focus of the course — you learn those on your own
- Intensive

2

Database System Environment



3

Contents

- Introduction
- Database-programming basics
- Data modeling: entity-relationship approach
- Relational data model
- Relational algebra
- SQL
- Constraints and triggers
- Transactions, security, and authorization in SQL

Contents (cont'd)

- Data-storage basics
- Representing data elements
- Index structures
- Recovery
- Concurrency control
- Query execution
- Distributed databases
- (More if time allows)

Prerequisites

- CSC 316 (Data structures for computer scientists)
- Knowledge of discrete mathematics and predicate logic
- Sufficient ability to program in Java or a willingness to acquire it through self-study

Course Textbook (required)

- *Database Systems: The Complete Book 2nd edition*, by Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom.
  - Has been requested onto library reserves – Hunt Library

Instructor

- Rada Chirkova, professor in the Department of Computer Science
- chirkova@csc.ncsu.edu
- Phone (919) 513-3506
- Office: EBII-2276
- Office hours (tentative): Mondays 1-2pm (for Section 601 students) and Wednesdays 12-1pm

About Me

- Questions?

Teaching Assistants

- Yunkai 'Kai' Xiao
- Jiaqing Yuan

Now Tell Us about Yourself

- Interview your neighbor and have him/her interview you
- Introduce your neighbor to the class

Course Website

- *On Moodle*
- On that page you will find:
  - course syllabus
  - announcements
  - learning objectives
  - assignments
  - and much more

Assignments

- Reading assignments: see course web page
  - Chapters 1, 2; Sec 4.1-4.4, 9.6 this and next week
- Eight homework assignments (plus warmup)
  - The homeworks use the Gradiance system
  - All work is to be done individually *unless otherwise specified.*
  - For the collaborative problems, you may form teams of 2-3 members (of students in this class) to cooperate *only* on those problems. After discussing the problems, please write up your answers individually. Indicate the names of the other members in your team, if any.

Project

- All students are required to complete a course project
- The details are announced on the course website
- You need to start forming project teams
  - I can help you find teammates
  - You can use the *Finding Teammates* message board

Grading

- Quizzes 5%
- Assignments 5%
- Project 30%
- Midterm exam (1 hour 15 min) 25%
- Final exam (3 hours, cumulative) 35%
- Extra credit up to 2%

Why the Homeworks Are Important

Self-Study Responsibilities

- Some of the topics are important but are either quite straightforward or not a main focus of this course.
- These topics will be identified as self-study topics on the course web page.
- Your knowledge of them will be evaluated as appropriate through exams, homework, programming assignments, or the project.

Miscellaneous

- Rules: The NC State University and Department of Computer Science rules regarding academic honesty and misconduct apply
  - Minimal penalty is 15% of course grade
- Regrade policy: see syllabus
- Computers and other gadgets: see syllabus
- Supporting fellow students in distress: see syllabus
- Safety on campus: see syllabus

Discussion

- Discuss the syllabus in pairs
- Ask me questions (later if you prefer)

Setting Goals and Expectations

- Write your goals for the semester (2-3 phrases)
- Write your expectations of me as a teacher (2-3 phrases)
- Anonymously: hand in rumors, if any, you have heard about the course or about me. We will discuss the rumors next time.
- You may also ask me questions.
- Feedback from students who have taken this course

Topic #1: A Brief Introduction to DBMS: The Big Picture

An Example of a Database

STUDENT:

Name	StudentNumber	Class	Major
Smith	17	1	CSC
Brown	8	2	ECE
Green	24	2	CSC
White	37	1	CSC

GRADE_RPRT:

StudentNo	Course	Grade
17	CSC742	A-
17	CSC316	B
24	CSC742	C-
8	CSC742	B+

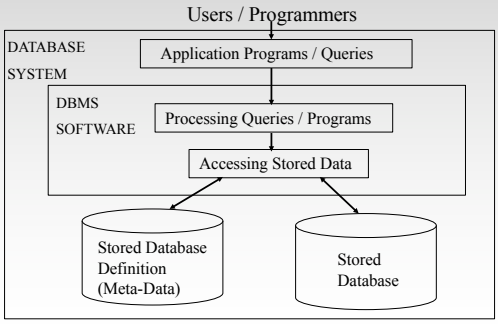
Database: What It Is

- Coherent collection of data with inherent meaning
  - Random assortment of data is not a database
- About an aspect of the world
  - Changes in the world are reflected in the database
- Fit to use for its intended purpose
  - Somebody is going to use the database

Group Discussion

- Come up with 2-3 examples of databases

Database System Environment



Example: Banking System

- Data = information on accounts, customers, balances, current interest rates, transaction histories, etc
- Massive
- Persistent
- Multi-user

Example (2 of 4)

- Jane at ATM1: withdraw $100 from account #55
  - Get balance from database
  - If balance > 100 then
    - balance := balance – 100
    - dispense cash
    - put new balance into database

Example (3 of 4)

- John at ATM2: withdraw $50 from account #55
  - Get balance from database
  - If balance > 50 then
    - balance := balance – 50
    - dispense cash
    - put new balance into database
- ***Initial balance = 100***
- ***Final balance = ??***

Example (4 of 4)

- Safe
- Convenient
- Efficient

Database Management System

- Specialized software
- Buy, install, set up for particular application
- Available for PC's, workstations, mainframes, supercomputers
- Is expected to:

Database Management System

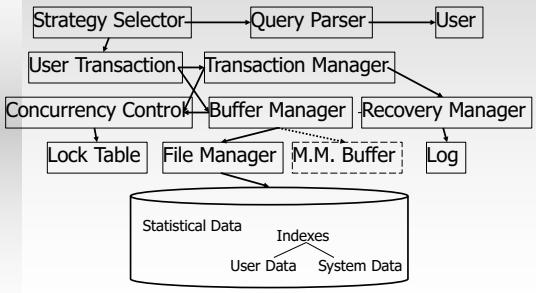
- Specialized software
- Buy, install, set up for particular application
- Available for PC's, workstations, mainframes, supercomputers
- Is expected to:
  - Allow users to *create* new databases (schema)
  - Give users the ability to *query/modify* the data
  - Support the *storage* of very large amounts of data
  - Control access to data from *many users* at once

Database Management System (continued)

- Major vendors/products:
  - Oracle
  - IBM (DB2)
  - Microsoft (SQL Server, Access)
- Powerful tool for providing *efficient, convenient,* and *safe multi-user* storage of and access to *massive* amounts of *persistent* data

DBMS Structure in More Detail

(slide courtesy of Dr. Hector Garcia-Molina)



DBMS Components

- Storage manager:
  - Stores on disk: data, metadata, indexes, logs
- Query compiler + execution engine:
  - Parses queries, optimizes by selecting query plan, executes the plan on the data
- Transaction manager:
  - Logs database changes to support recovery after system crashes
  - Supports concurrent execution of transactions

People

- DBMS implementer: builds systems
- Database designer: sets up schema, loads data
- Database user: queries/modifies data
- You in the course project

The Project: What You Will Need

- DBMS
- SQL (DDL and DML)
- Required - host languages (Java, C/C++, Perl, …)
- Web application servers (optional)
- SQL editors (optional) – e.g., Toad
- Tools for user interface (optional): forms, reports, etc.

Less Traditional Applications

- Real-time, historical data and queries, "active" databases
- Distributed, heterogeneous databases
- Scientific data