Scope of this Course

- Directed at computer science *undergraduate* students
- Introduces 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

Database System Environment

<table>
<thead>
<tr>
<th>Users / Programmers</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATABASE SYSTEM</td>
</tr>
<tr>
<td>DBMS SOFTWARE</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Application Programs / Queries</td>
</tr>
<tr>
<td>Processing Queries / Programs</td>
</tr>
<tr>
<td>Accessing Stored Data</td>
</tr>
<tr>
<td>Stored Database Definition (Meta-Data)</td>
</tr>
<tr>
<td>Stored Database</td>
</tr>
</tbody>
</table>
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 and index structures
- Recovery
- Concurrency control
- Query execution
- Distributed databases

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

- Bundled with Gradiance software for homeworks

Instructor

- Peng Ning, associate professor of Computer Science
- Email policy
  - pning@ncsu.edu
  - Phone (919) 513-4457
  - Office: EBII-3258, Centennial Campus
  - Office hours: Tuesdays and Thursdays 9:45-10:45am, or by appointment

Teaching Assistant

- Ms. Yao Liu
- Office hours: Tuesdays and Thursdays, 4pm - 5pm
- Office: EBII-3323
- Email: yliu20 AT ncsu.edu
Now Tell Us about Yourself

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

Course Website

- [http://courses.ncsu.edu/csc440/lec/001/](http://courses.ncsu.edu/csc440/lec/001/)
- Tour of web sites
- Important: course announcements:
  - on the web site
- Learning objectives and lecture slides on the web

Assignments

- Reading assignments: see course web page
  - Chapter 1 and Section 8.5 now
- Eight homework assignments
  - 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 will be announced on the course website
- You need to start forming project teams
  - Talk to your classmates
  - Use the message board

Grading

- Quizzes 5%
- Assignments 10%
- Project (demo day April 23) 22%
- Midterm (February 28 in class) 30%
- Final exam (April 29, 8-11am) 33%

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 apply
- **Regrade policy**: 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.

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

Group Discussion
- What do you know already about the topic?
  - We’ll see next week!
- What questions do you have about the course content?
An Example of a Database

**STUDENT:** Name  StudentNumber  Class  Major

<table>
<thead>
<tr>
<th>Name</th>
<th>StudentNumber</th>
<th>Class</th>
<th>Major</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>17</td>
<td>1</td>
<td>CSC</td>
</tr>
<tr>
<td>Brown</td>
<td>8</td>
<td>2</td>
<td>ECE</td>
</tr>
<tr>
<td>Green</td>
<td>24</td>
<td>2</td>
<td>CSC</td>
</tr>
<tr>
<td>White</td>
<td>37</td>
<td>1</td>
<td>CSC</td>
</tr>
</tbody>
</table>

**GRADE_RPRT:** StudentNo  Course  Grade

<table>
<thead>
<tr>
<th>StudentNo</th>
<th>Course</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>CSC742</td>
<td>A-</td>
</tr>
<tr>
<td>17</td>
<td>CSC316</td>
<td>B</td>
</tr>
<tr>
<td>24</td>
<td>CSC742</td>
<td>C-</td>
</tr>
<tr>
<td>8</td>
<td>CSC742</td>
<td>B+</td>
</tr>
</tbody>
</table>

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

Users / Programmers

DATABASE SYSTEM

Application Programs / Queries

DBMS SOFTWARE

Processing Queries / Programs

Accessing Stored Data

Stored Database Definition (Meta-Data)

Stored Database

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 (cont’d)

- 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

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

DBMS Components
- Storage manager:
  - Stores on disk: data, metadata, indexes, logs
- Query processor:
  - 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
DBMS Structure in More Detail

- Strategy Selector
- Query Parser
- User Transaction
- Transaction Manager
- Concurrency Control
- Buffer Manager
- Recovery Manager
- Lock Table
- File Manager
- M.M. Buffer
- Log
- Statistical Data
- Indexes
- User Data
- System Data

People

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

Less Traditional Applications

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