Introductions

• Name
• Program
• Any background with databases?
• Why taking 523?
Course Objectives

• Develop a general understanding of databases, and specific understanding of the relational database model.
• Gain experience with both the theoretical and practical aspects of database design and implementation.
• Be able to weigh, discuss, and justify database design decisions.
• Learn about concepts involved in database operation
• Gain an understanding of important ideas for databases in the future.
Course Website & Policies

• Open-web course website:
  http://www.ils.unc.edu/courses/2013_spring/inls523_001/
  (also available from the SILS Courses page)
  – Course information
  – Course policies
  – Lecture schedule and slides
  – Assignments and due dates

• Sakai site
  – Submit assignments
  – Take quizzes
  – View graded assignments
  – View your list of grades
Graded Work (part 1)

• Quizzes & In-class Exercises (10%)
  – Throughout the semester, there will be a number of quizzes designed to gauge your understanding of course material. There will also be several in-class exercises to give you hands-on experience applying concepts we discuss in class. In-class exercises may include a component that will be turned in as a “quiz”.

• Homework (30%)
  – In my experience, developing proficiency in database concepts requires a good deal of practice. As such, in this course, there will be frequent homework assignments designed to give you practice applying the concepts that are discussed in class. Homework assignments are to be completed individually.
Graded Work (part 2)

• **Database Project (20%)**
  – Over the course of the semester, students will work in pairs to specify, design, and implement a database project of their own design. There will be three parts to the project: 1) an initial description of the database requirements, 2) a completed model and design, and 3) a final implemented database, loaded with data and demonstrated with representative queries.

• **Participation (5%)**
  – Students are expected to regularly attend and participate in class. Throughout the semester, students will be called on in class to answer questions, and to present work from homework assignments, in-class exercises, and quizzes. Your participation grade will be based on regular class attendance, courteous behavior in class, being prepared for class, and being fully and actively engaged during class.

• **Exams**
  – There will be a mid-term exam (15%) and a comprehensive final exam (20%).
Textbook and Computing

• **Required textbook**

• **SQLite**
  – For this course, you will do most of your database work using a database package called SQLite. In class, we will discuss several options for installing SQLite software on your Mac, PC, or Linux computer.

• **Drawing tools**
  – You will need to use a drawing tool to create your E/R diagrams and database models. No specific tool is required – you may use the tool(s) that you prefer. Some commonly used tools are: Microsoft Visio, Omnigraffle, Powerpoint, LibreOffice Draw.

• **PDF**
  – You will need the ability to save Word processing files and diagrams as PDF files. Most current versions of Microsoft Office and LibreOffice support saving to PDF. You will also need a tool such as Acrobat Reader that will allow you to open and view PDF files.

• **In-class exercises**
  – We will do several in-class exercises that will require computer use. For these exercises, students are expected to bring their laptop computer to class. The dates you will need a computer in-class will be announced in advance.
Policies

• Review the Course Policies document
  – Due dates and Late work
  – Requests for Extensions and Absences
  – Statute of Limitations
  – Course Communication
  – Announcements
  – Assignments submitted to Sakai
  – Grades
  – Email
  – Honor code
  – Special Accommodations
What are we going to learn?

• Review the course schedule on the website:
  – Database concepts
  – Modeling and ER
  – Relational concepts
  – ER to Relational Mapping
  – SQL, SQL, SQL
  – Advanced SQL
  – Relational Algebra
  – Functional Dependencies
  – Normal Forms and Normalization
Great!

Let’s talk about databases!

I’ve heard about these “databases”, and I know they’re important, but what the heck are they really?
Anybody know any good databases?

Anybody used any databases recently?
Databases All Around Us

“databases play a critical role in almost all areas where computers are used” (EN6 p4)

• **Data**: facts, can be recorded, have meaning
  – Ex: names, phone numbers, addresses
• **Database**: collection of related data

But... this is a very general definition. (*Why?*)
Database Properties

• Database Properties
  – Represent some aspect of the “real” world
  – Has meaning as a collection
  – Design and built for a specific purpose
  – Many different sizes and complexities
  – Supports users
  – May be manual or computerized
DBMS

• DBMS = Database Management System

• “a collection of programs that enables users to create and maintain a database” (EN6 p5)

• Activities:
  – Defining the DB
    • Descriptive information, meta-data
  – Constructing the DB
    • Storing the data
  – Manipulating the DB
    • Querying, updating
  – Sharing
Databases and Users

EN6 Figure 1.1
Client-Server vs. Local

- **Client-server**
  - Data lives on the server
  - DBMS software lives on server
  - Applications run on clients
  - Applications can issue queries to the DBMS (1)
  - DBMS sends results back to the application (2)
DB Client-Server + WebServer

• A common arrangement for web sites:
  – Database is on a DB server
  – Web site is on a different Web Server
  – Users access the web site using a browser (e.g. Firefox on my PC) (1)
  – Web site issue queries to the DBMS (2)
  – DBMS sends results back to the web site (3)
  – Web site displays results to users (4)
Client-Server vs. Local

• Local
  – Nowadays, local device can be almost computing device
    • Workstation, PC, tablet, smartphone
  – Data, DBMS, and applications are all on local device
  – Applications can issue queries to the DBMS (1)
  – DBMS sends results back to the application (2)
Movie DB Exercise

Let’s see what a database might really look like. Think about creating a movie database.

What *things* do we need to represent in the DB?

What *characteristics* do those things have that we want to store?

What *relationships* are there between / among the things?

(In-class, draw these as an ER diagram on the board)
# One possible movie DB...

## MOVIES

<table>
<thead>
<tr>
<th>MPAA #</th>
<th>Title</th>
<th>Year</th>
<th>ProdCo</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>Star Wars</td>
<td>1977</td>
<td>AA55</td>
</tr>
<tr>
<td>3456</td>
<td>Radiers</td>
<td>1981</td>
<td>AA55</td>
</tr>
<tr>
<td>5678</td>
<td>Matrix</td>
<td>1999</td>
<td>BC73</td>
</tr>
</tbody>
</table>

## ACTORS

<table>
<thead>
<tr>
<th>SAG #</th>
<th>Name</th>
<th>BDay</th>
</tr>
</thead>
<tbody>
<tr>
<td>9999</td>
<td>Harrison Ford</td>
<td>1942</td>
</tr>
<tr>
<td>8888</td>
<td>Carrie Fisher</td>
<td>1956</td>
</tr>
<tr>
<td>7777</td>
<td>Carrie-Anne Moss</td>
<td>1967</td>
</tr>
<tr>
<td>5555</td>
<td>Keanu Reeves</td>
<td>1964</td>
</tr>
</tbody>
</table>

## APPEARED_IN

<table>
<thead>
<tr>
<th>SAG #</th>
<th>MPAA #</th>
</tr>
</thead>
<tbody>
<tr>
<td>9999</td>
<td>1234</td>
</tr>
<tr>
<td>8888</td>
<td>1234</td>
</tr>
<tr>
<td>9999</td>
<td>3456</td>
</tr>
<tr>
<td>7777</td>
<td>5678</td>
</tr>
<tr>
<td>5555</td>
<td>5678</td>
</tr>
</tbody>
</table>

## PRODUCTION CO

<table>
<thead>
<tr>
<th>CorpID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA55</td>
<td>LucasFilm</td>
</tr>
<tr>
<td>BC73</td>
<td>WarnerB</td>
</tr>
</tbody>
</table>
DB Design Phases

• Requirements specification and analysis
  – What needs to be modeled? User requirements?

• Conceptual design
  – Requirements → ER model

• Logical design
  – ER model → data description/schema

• Physical design
  – Implementation in a specific DBMS
DB Approach vs. Files

• Data could be stored in files
  – e.g. text documents, spreadsheets

• Database approach offers many advantages
  – Self-describing (meta-data)
  – Data abstraction
    • Program-data independence
    • Program-operation independence
  – Multiple views of data
  – Data sharing, transactions processing, and access control
ACID Properties

- ACID properties ensure correctness
  - Atomicity
    • All or none of a transaction will occur
  - Consistency
    • Data integrity is valid
    • e.g. no entries in `appears_in` without a valid SAG# and MPAA##
  - Isolation
    • Transactions appear to be independent
  - Durability
    • Can recover from a crash at any point
The “Who” of Databases

- Database administrators
- Database designers
- End users
- Systems analysts
- Application programmers
- DBMS designers and implementers
- Tool developers
- Operators and maintenance personnel
DBMS Advantages

• Controlling redundancy
  – Separate tables for actor information
  – Separate tables for production company
  – Normalization

• Restrict/control access

• Persistent storage, backup & recovery
  – Data objects are stored on disk, can be retrieved later

• Multiple user interfaces

• Represent complex relationships among data

• Enforce integrity constraints
Extended DB Capabilities

• Scientific data
• “Big” data
• Image & video databases
• Data mining
• Spatial & geographical databases
• Time series applications

• Relationship to Information Retrieval
Exercises

• What views of the movie database might be needed by other users?
• What integrity constraints might there be in the movie database?
• How would you change an item in the movie database?
  – An actor changes their name?
  – An MPAA # was discovered to be wrong and needs to be updated?