CS1315: Introduction to Media Computation

Databases

Why databases?

- Rather: Why not just use files?
  - **Why do we care about using some extra software for storing our bytes?**
- Databases provide efficient access to data in a standardized mechanism.
  - **Databases are fast.**
  - **Databases can be accessed from more than one place in more than one way.**
  - **Databases store *relations* between data**

Databases are fast because of indices

- Filenames are *indexed* just by name.
- Usually, you care about information that is found by something other than a filename.
  - **For example, you may care about someone's information identified by last name or by SSN or even birthdate or city/state of residence.**

Databases are standardized

- There are many different standard databases.
  - **In the UNIX and open source markets: bsddb, gdbm, MySQL**
  - **In the commercial markets: Microsoft Access, Informix, Oracle, Sybase**
- Information stored in a standard database can be accessed and manipulated via many different tools and languages.

Databases store relations

- Databases can store *names* for the *fields* of data
- They can store which fields are important (and thus *indexed* for rapid access), and how fields are related (e.g., that each pixel has three color components, that each student has one transcript)

Powerful, relational databases

- Modern databases are mostly *relational*
- Relational databases store information in *tables* where *columns* of information are named and *rows* of data are assumed to be related.
- You work with multiple tables to store complex relationships.

Simple to understand fundamentals and simple to use

Difficult to set up and administer!

Database Management Systems

- If you do any database work in your professional life, you *will* be using relational databases and the standard database language SQL.
  - **We haven't asked you to do that for homework in this class.**
  - **We may ask you about SQL on exams but not how to write queries in SQL.**
- The software that runs a database is called a database management system (DBMS)
- A standard, free DBMS is MySQL
  - **If you want to figure out how to use MySQL on your own computer, use the material in the book.**

A simple table

Fields

Baseball

Team	Place
Braves	Atlanta
Cubs	Chicago
White Sox	Chicago

The implied relation of this row is that the Braves play in Atlanta.

More complex tables

Picture	PictureID
Class1.jpg	P1
Class2.jpg	P2

StudentName	StudentID
Katie	S1
Brittany	S2
Carrie	S3

PictureID	StudentID
P1	S1
P1	S2
P2	S3

How to use complex tables

- What picture is Brittany in?
  - **Look up her ID in the student table**
  - **Look up the corresponding PictureID in the PictureID-StudentID table**
  - **Look up the picture in the Picture table**
    - Answer: Class1.jpg

StudentName	StudentID
Katie	S1
Brittany	S2
Carrie	S3

Picture	PictureID
Class1.jpg	P1
Class2.jpg	P2

PictureID	StudentID
P1	S1
P1	S2
P2	S3

Another Use

- Who is in "Class1.jpg"?
  - **Look up the picture in the Picture table to get the ID**
  - **Look up the corresponding PictureID in the PictureID-StudentID table**
  - **Look up the StudentNames in the Student picture**
    - Answer: Katie and Brittany

StudentName	StudentID
Katie	S1
Brittany	S2
Carrie	S3

Picture	PictureID
Class1.jpg	P1
Class2.jpg	P2

PictureID	StudentID
P1	S1
P1	S2
P2	S3

A Database *Join*

- We call this kind of access across multiple tables a *join*
- By joining tables, we can represent more complex relationships than with just a single table.
- Most database systems provide the ability to join tables.
- Joining works better if the tables are well-formed:
  - **Simple**
  - **Containing only a single relation per row**

SQL: Structured Query Language

- SQL is usually pronounced "S.Q.L." or "Sequel"
- It's a language for database creation and manipulation.
  - **Yes, a whole new language, like Python or Java**
  - **It actually has several parts, such as DDL (Data Definition Language) and DML (Data Manipulation Language), but we're not going to cover each part.**
- We're not going to cover *all* of SQL
  - **There's a lot there**
  - **And what's there depends, on part, on the database you're using.**

SQL: Selecting data in a database

- Select *column1,column2* from *tablename*
- Select *column1,column2* from *tablename* where *condition*

select * from Baseball

select Team,Place from Baseball

select * from Baseball where Place='Chicago'

select Team,Place from Baseball where Place='Chicago'

Doing this from Python

- When you use a *select* from Python,
  - **Your cursor has a <u>variable</u> *rowcount* that tells you how many rows were selected.**
    - This is called an *instance variable*
    - It's a variable known just to that object, similar to how a method is a function known just to that object.
  - **Method *fetchone()* gives you the next selected row.**
- Fetchone() returns a list

SQL in Python

```
def showPersons(con):
 con.execute("select name, age from Person")
 for i in range(0,con.rowcount):
  results=con.fetchone()
  print results[0]+" is "+str(results[1])+" years old"
```

We create an SQL command as a string and send it off to the database. (compare similar treatment of HTML)

Why is a database useful for a big web site?

- For CNN.com:
  - **Can have multiple authors and editors creating multiple stories distributed all over a network.**
  - **Can pull the content automatically via a program and merge all the stories into one big website**
- Works similarly for other kinds of large websites
  - **Amazon.com**
    - Where do you think their catalog and review is stored?
  - **EBay.com**
    - Where do you think all those pictures and descriptions and bid information is stored?