CS411
Database Systems

01: Introduction

Saurabh Sinha

Welcome to CS411

- Web site:

  https://agora.cs.illinois.edu/display/cs411fa10/

- Announcements, syllabus, policies, schedule, …

- Please read the class syllabus, policies, and lecture schedule

Teaching Staff: The Front-End

- Yunliang Jiang

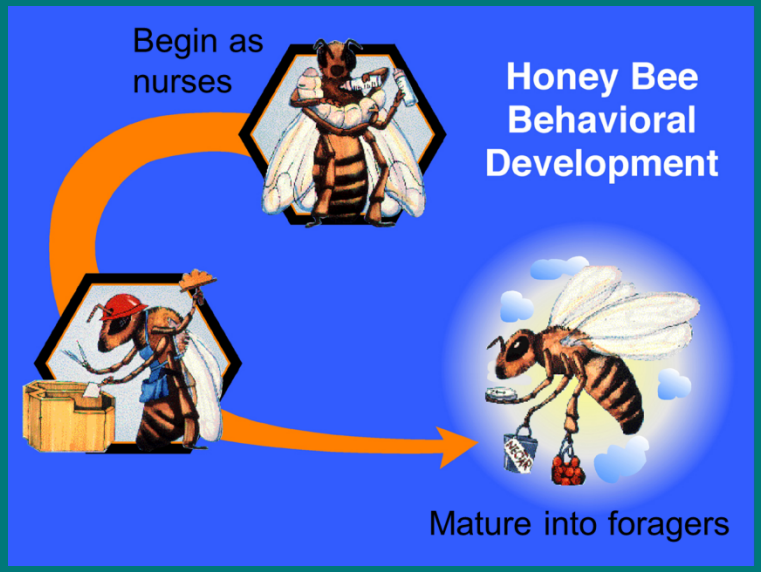
- Marina Danilevsky

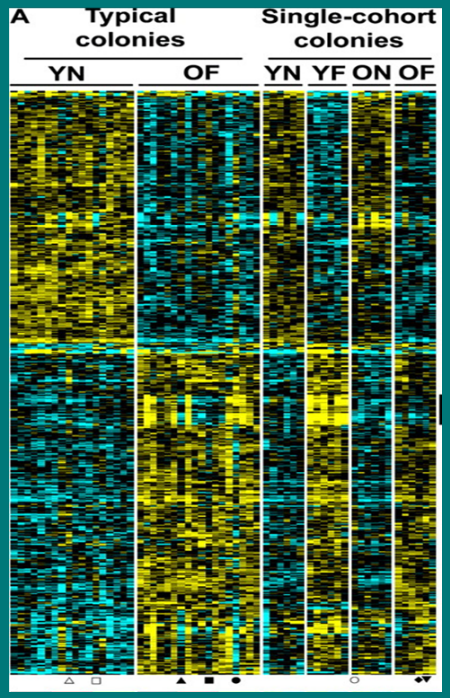
Teaching Staff: The Back-End

- Saurabh Sinha
- Research interest:
  - Bioinformatics / computational biology
- Brief history
  - India (B.Tech in CSE from IIT Kanpur)
  - Seattle, WA (PhD in CSE from U.W.)
  - New York, NY (Post-doc in bioinformatics, Rockefeller Univ.)
  - Illinois (Assistant Professor, now in 6th year here)

Projects in Sinha "lab"

- Social behavior in honeybees



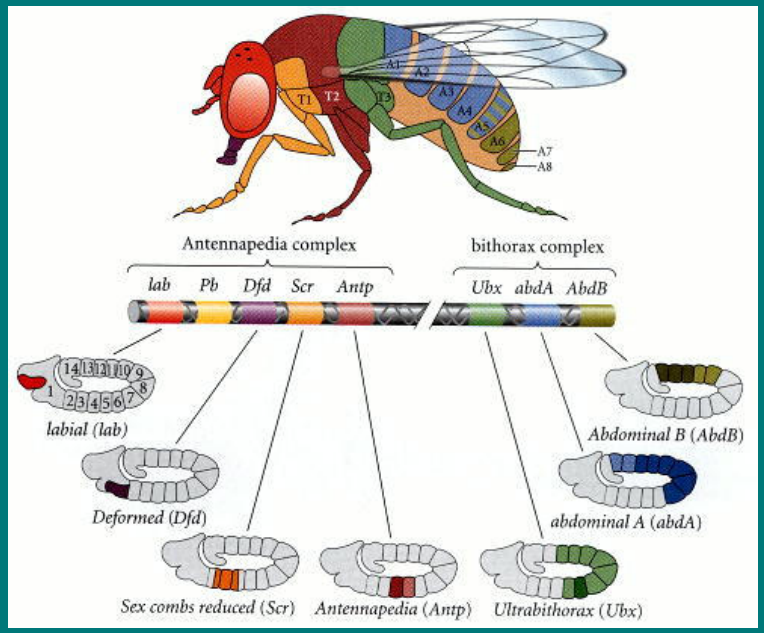


- Gene activity in the honeybee brain
- Algorithms to figure out the connection between genes and behavior (using Hidden Markov models)

Projects in Sinha "lab"

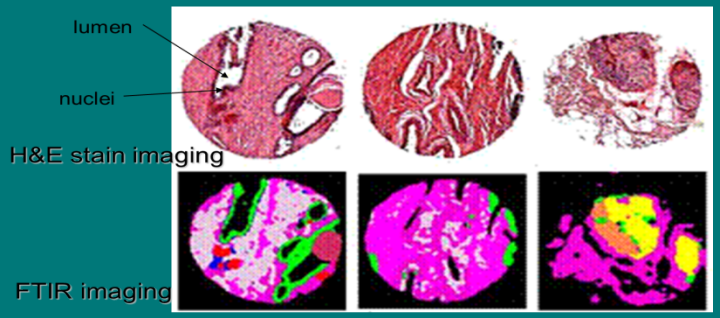
- Predicting regions of DNA that are involved in development

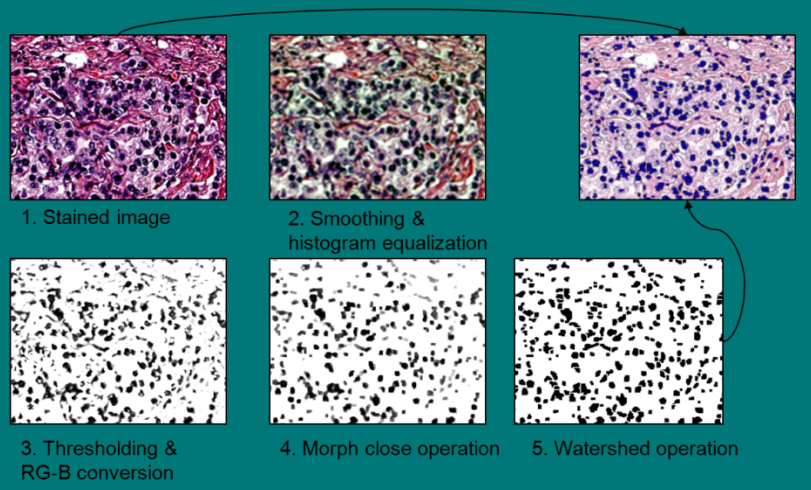
Probabilistic sequence alignments and Logistic Regression



Projects in Sinha "lab"

- Cancer diagnosis from spectroscopy data





You Tell Me --

- Why Are You Taking this Course?

- Are you interested more in being
  - An IT Guru at Goldman-Sachs or Boeing?
  - A System Developer at Oracle or Google?

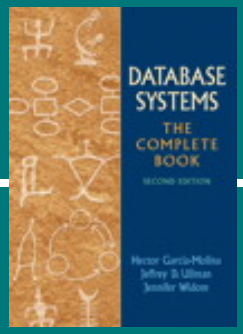
CS411 Goal: Two Perspectives of DBMS

- User perspective
  - how to use a database system?
  - conceptual data modeling, the relational and other data models, database schema design, relational algebra, and the SQL query language.
- Systems perspective
  - how to design and implement a database system?
  - data representation, indexing, query optimization and processing, transaction processing, concurrency control, and crash recovery

Prerequisite

- Must have data structure and algorithm background
  - CS 225 or equivalent
- Good programming skills
  - project will require lots of programming
  - need C++, Java, or PHP … to do a good job at talking to DB
  - you or your project group picks the language

Textbook



- Textbook:

  Database Systems: The Complete Book, 2/e, by Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer D. Widom

- Good references:
  - *Database Management Systems*, by Raghu Ramakrishnan an Johannes Gehrke, McGraw-Hill
  - *Database System Concepts*, by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, McGraw Hill
  - *Fundamentals of Database Systems*, by Ramez Elmasri and Shamkant Navathe, Addison Wesley
  - *An Introduction to Database Systems*, by C. J. Date, Addison Wesley

Course Format

- For all students
  - two 75-min lectures / week
  - 5 assignments planned
  - project (significant)
  - a midterm and a final exam
- Graduate students: 4 credits option
  - do an extra project or write a survey paper
  - will announce later

Lectures

- Lecture slides will be posted shortly before or after the lecture
  - are to complement the lectures
- Lectures are important for guiding your reading of textbook (and will be covered in exams and homeworks)
  - so please attend lectures regularly

Homework Assignments

- Mostly paper-based, some may involve light programming
- Due by noon of the due date; submit through Compass.
- No late homework will be accepted

Project:

- Database-Driven Web-based Information System
  - select an database application that needs a database
  - design and build it from start to finish
  - your choice of topic:
    useful, realistic, database-driven, web-based
- Team work
- Significant amount of programming (we will provide tutorials)
- Will be done in stages
  - you will submit some work at the end of each stage
- Will show a demo at semester end

Project Groups

- Project will be done in group of 3-4 students
  - learn how to work in a group: valuable skills
  - also use project group as study partners
- Try to form groups as soon as possible
  - can start by posting requests on the class newsgroup
- There will be a deadline soon for forming groups
  - if you have not formed groups by then
  - we will help assign you to groups
- Grading:
  - all members receive same grade
  - if someone drops out, the rest pick up the work

Exams

- Midterm and final
- There will be some brief review before each exam
- Check dates and make sure no conflict!
  - generally no makeup exams unless exceptional cases (see policy page)

In-Class Quiz

To encourage attending and participating in class:

- Three brief in-class quizzes.
- Each tests your participation in class.
- Each accounts for 1% of class grading.

Tentative Grading Breakdown

- Homework: 17%
- Project: 35%
- Midterm: 20%
- Final: 25%
- In-class quiz: 3%

How Do We Work Together?

Contacting the Staff --

Office Hours

- The best way for asking questions and clarifications
- Will have office hours three days a week.
- See course Web for schedule

Communications

- Website: "Announcements" page
- Newsgroup: class.fa10.cs411
  - vitally important!
  - make sure to check it regularly for questions/clarifications
  - announcements will appear here and the course Web
- If you have a question/problem
  1. talk to people in your group first
  2. post your question on newsgroup
  3. email TA
  4. go to office hours to talk to TA or instructor

Newsgroup: class.fa10.cs411

- Designed for you and your peer
  - to communicate and help one another
  - please do not post solutions/admin-requests
- TAs will monitor and try their best to help with your questions
- There can be many questions
  - may not be able to answer all of them timely manner
  - not good for more complex questions
  - hence should come to office hours or email TA

Database Management System (DBMS)?

- System 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: many gigabytes at a minimum for big banks, more if keep history of all transactions, even more if keep images of checks -> Far too big for memory
- PERSISTENT: data outlives programs that operate on it

MULTI-USER Access

- MULTI-USER: many people/programs accessing same database, or even same data, simultaneously -> Need careful controls
- Alex @ ATM1: withdraw $100 from account #002

```
get balance from database;
if balance >= 100
        then balance := balance - 100;
dispense cash;
put new balance into database;
```

- Bob @ ATM2: withdraw $50 from account #002

```
get balance from database;
if balance >= 50
        then balance := balance - 50;
dispense cash;
put new balance into database;
```

MULTI-USER: What Can Go Wrong?

- Initial balance = 200. Final balance = ??

DBMS: More Requirements

- SAFE:
  - from system failures. E.g., money should not disappear from the account, due to a power failure!
  - from malicious users
- CONVENIENT:
  - simple commands to debit account, get balance, write statement, transfer funds, etc.
  - also unpredicted queries should be easy
- EFFICIENT:
  - don't search all files in order to get balance of one account, get all accounts with low balances, get large transactions, etc.

Why Direct Implementation Won't Work

- Early DBMS evolved from file systems
- Provided storage of MASSIVE amounts of PERSISTENT data, to some extent
  - Although, size limit by disk or address space
- SAFE?
  - when system crashes we may lose data
- EFFICIENT?
  - Does not intrinsically support fast access to data whose location in file is not known

Why Direct Implementation Won't Work

- CONVENIENT?
  - need to write a new C++/Java program for every new query
  - Schema change entails changing file formats; need to rewrite virtually all applications
- MULTI-USER ACCESS?
  - limited protection
  - need to worry about interfering with other users

That's why the notion of DBMS was invented!

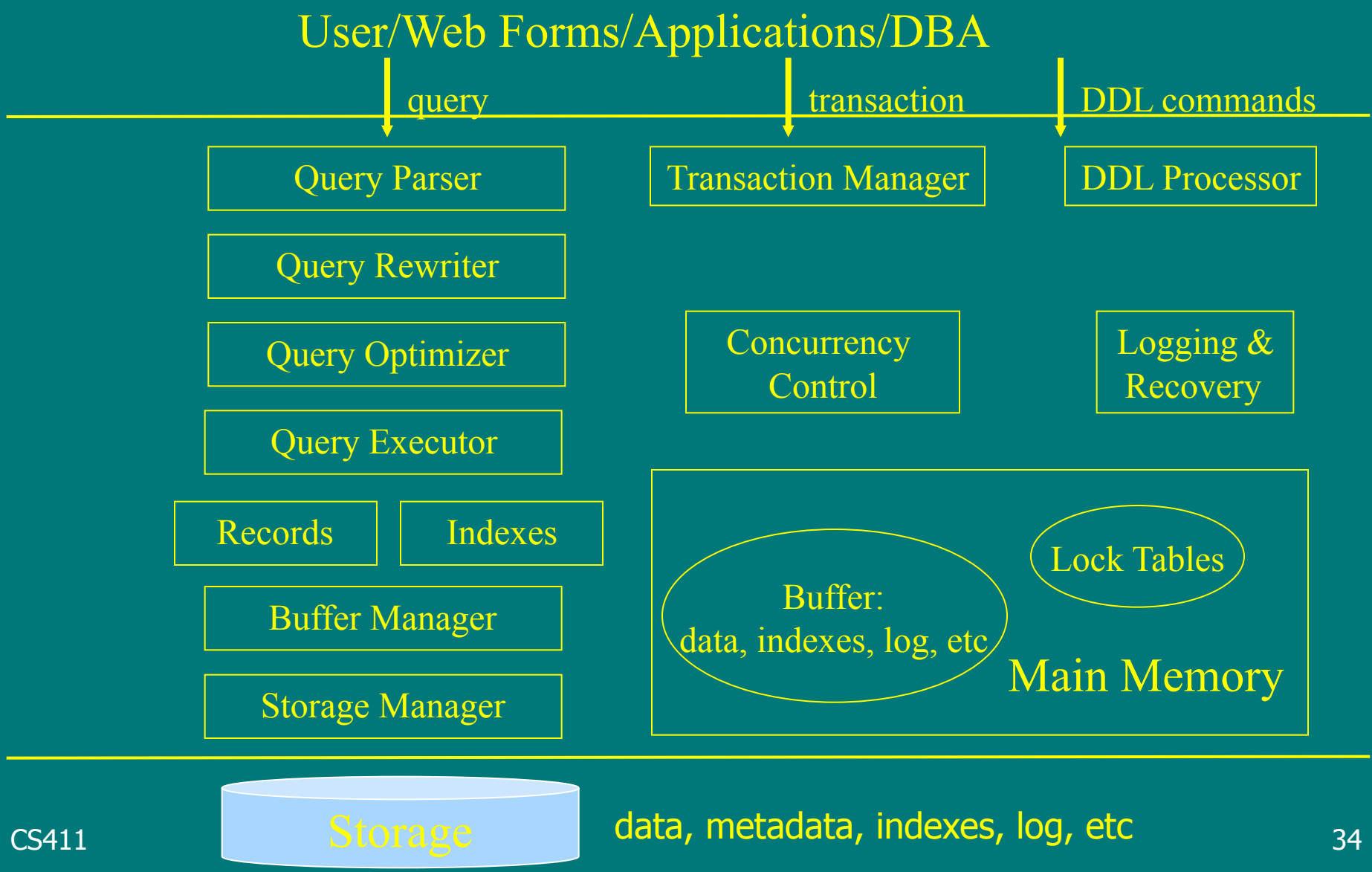
DBMS: A Software System

- Buy, install, set up for particular application
- Available for PC's, workstations, mainframes, supercomputers
- Major vendors:
  - Oracle
  - IBM (DB2)
  - Microsoft (SQL Server, Access)
  - Sybase
  - ➢ all are "relational" (or "object-relational") DBMS

DBMS Examples

- Most familiar use: many Web sites rely heavily on DBMS's
- And many non-Web examples

DBMS Architecture



Data Model and Schemas

- Data model or schema:
  - conceptual structuring of data stored in database
  - E.g., data is set of records, each with student-ID, name, address, courses, photo
  - E.g., data is graph where nodes represent cities, edges represent airline routes
- Schema versus data
  - schema describes how data is to be structured, defined at set-up time, rarely changes (also called "metadata")
  - data is actual "instance" of database, changes rapidly

DDL and DML

- Data definition language (DDL)
  - commands for setting up schema of database
- Data manipulation language (DML)
  - Commands to manipulate data in database:
  - RETRIEVE, INSERT, DELETE, MODIFY
  - Also called "query language"

People

- DBMS user: queries/modifies data
- DBMS application designer
  - set up schema, loads data, …
- DBMS administrator
  - user management, performance tuning, …
- DBMS implementer: builds systems

First ½ Topics: User Perspective

- Entity-Relationship Model
- Relational Model
- Relational Database Design
- Relational Algebra
- SQL and DBMS Functionalities:
  - SQL Programming
  - Queries and Updates
  - Indexes and Views
  - Constraints and Triggers

Second ½ Topics: System Perspective

- Storage and Representation
- Indexing
- Query Execution and Optimization
- Transaction Management

Project Tutorial Lectures

- Anatomy of a large scale database-driven web-based info system.
- Web programming.
- DB programming.

Special Topics: 2-3 Lectures

- Databases in the Real World
- Advanced Database Research

How to Get the Most out of CS411?

- Read and think before class
  - welcome to ask questions before class!
- Study and discuss with your peers
  - discuss readings to enhance understanding
  - discuss assignments but write your own solution!
- Use lectures to guide your study
  - use it as a roadmap for what's important
  - lectures are **starting** points– they do not cover everything you should read

Questions?

- Any questions? Please come talk to me.