CS411
Database Systems

01: Introduction

Kevin C. Chang

Welcome to CS411

- Web site:

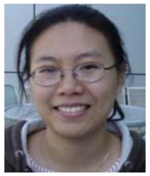
  https://agora.cs.illinois.edu/display/cs411sp11/
- Announcements, syllabus, policies, schedule, …
- Please read the class syllabus, policies, and lecture schedule; ask now if you have questions.

Teaching Staff: The Front-End

- **Marina Danilevsky**: I2CS, Homework, Tutorials

- **Mark Overholt**: Newsgroup, Homework, Website

- **Hanna Zhong**: Projects Track 1



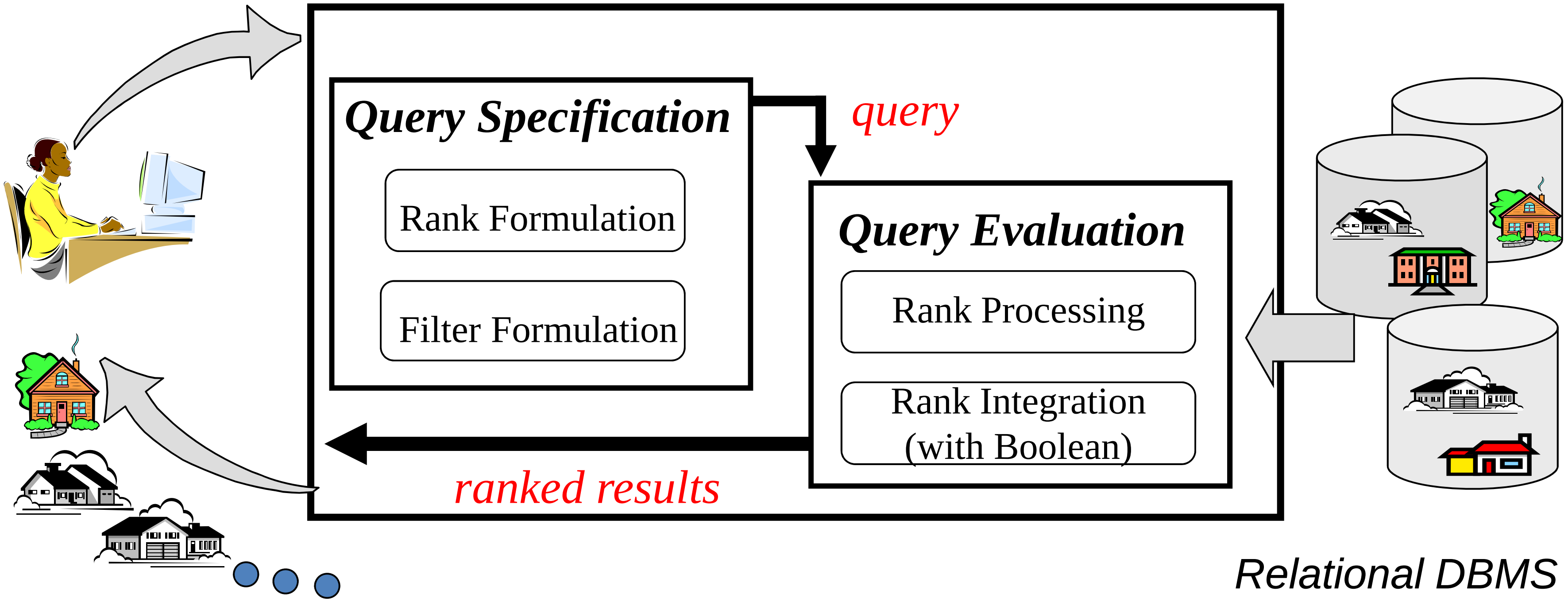
- **Rui Li**: Projects Track 2

Teaching Staff: The Back-End

- Kevin C. Chang
- Research interest:
  - database systems, Web integration and mining
  - current projects: AIM, MetaQuerier, WISDM, ARISE (we are recruiting!)
- Hobbies:
  - ocean diving, mountain climbing
- Brief history
  - Taiwan (BS in EE from National Taiwan University)
  - California (MS in CS, PhD in EE from Stanford)
  - Illinois (associate professor in CS, UIUC)
  - "Data mining": what can you predict? East or west? CS or EE?

The AIM Project: AIMing to the top

Supporting ranking in data retrieval



Query Specification

Rank Formulation

Filter Formulation

query

Query Evaluation

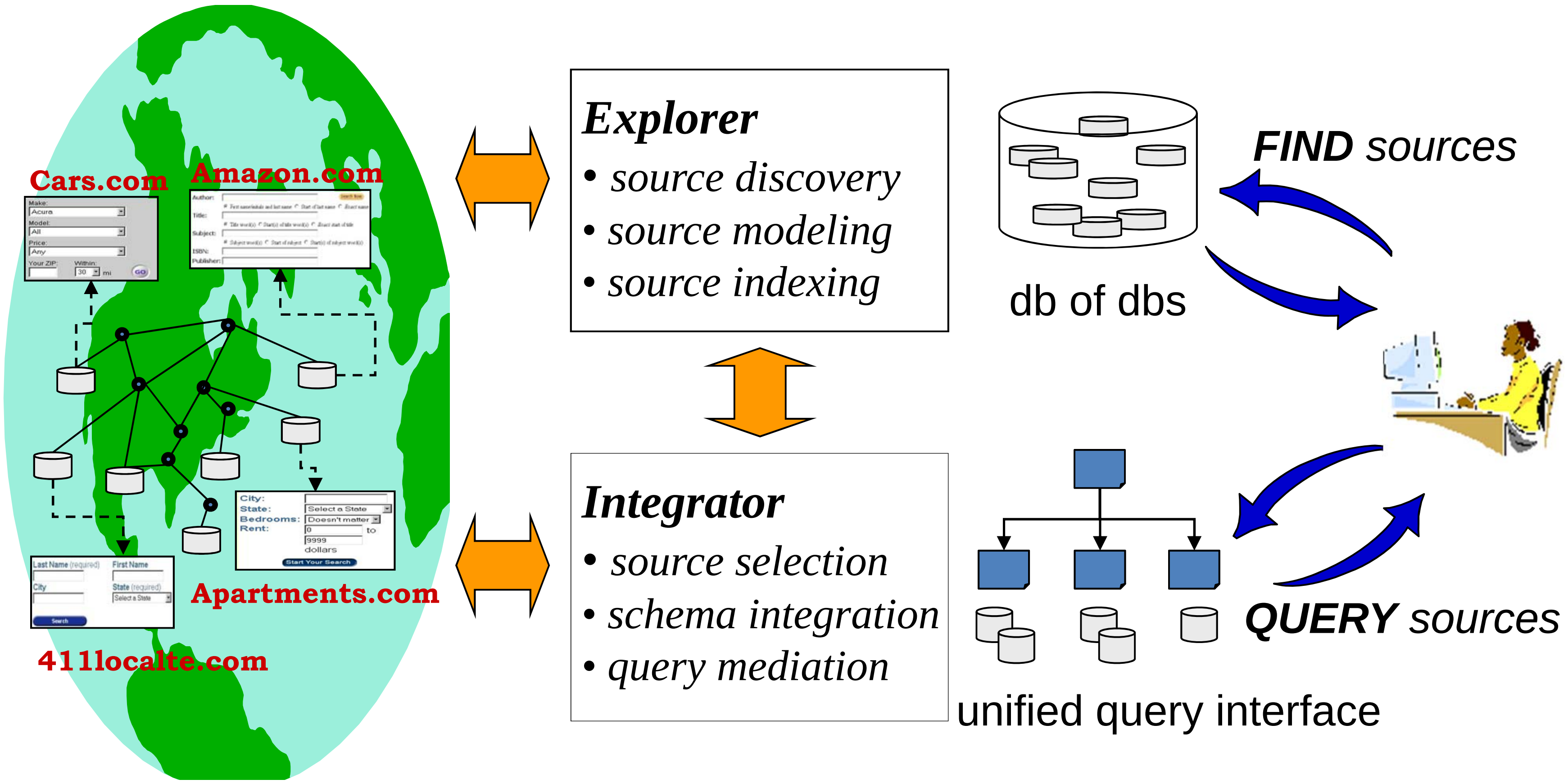
Rank Processing

Rank Integration (with Boolean)

ranked results

Relational DBMS
or middleware
sources (e.g,, Web)

The MetaQuerier Project
Exploring *and* *integrating* *deep Web*



Explorer

- *source discovery*
- *source modeling*
- *source indexing*

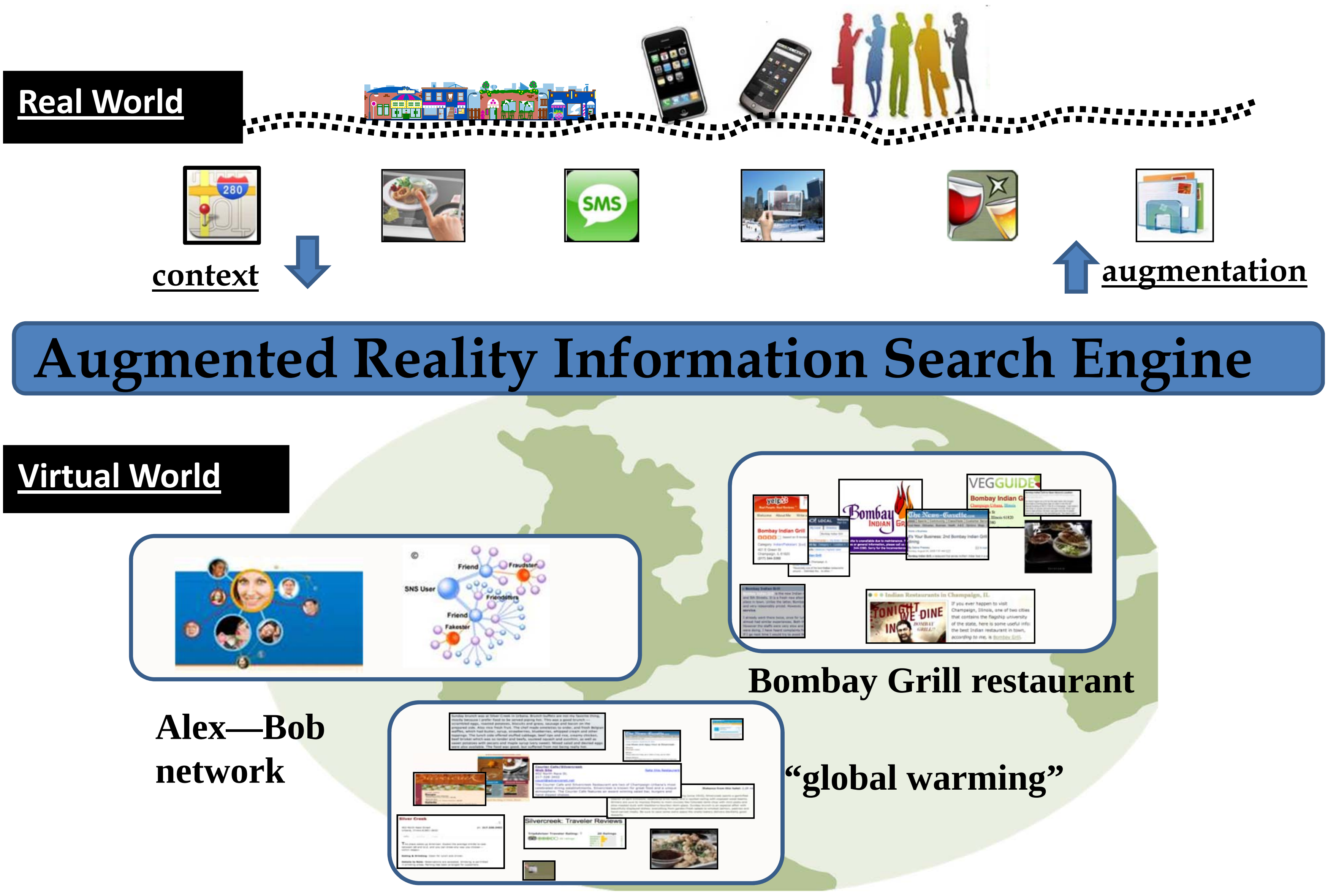
Integrator

- *source selection*
- *schema integration*
- *query mediation*

The WISDM Project:
Data Aware Search over the Web

Data Aware Search?

The ARISE Project:
Augmented Reality Information Search Engine



Real World

context

augmentation

Augmented Reality Information Search Engine

Virtual World

Alex—Bob network

Bombay Grill restaurant

“global warming”

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

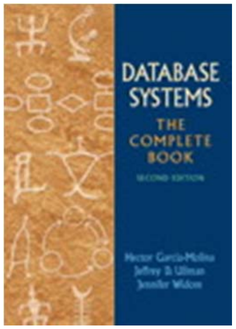
- User perspective
  - how to use a database system?
  - conceptual data modeling, the relational and other data models, database schema design, relational algebra, and the SQL query language.
- System perspective
  - how to design and implement a database system?
  - data representation, indexing, query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

Prerequisite

- Must have data structure and algorithm background
  - CS 225 or 400 equivalent

- Good programming skill
  - project will require lot of programming
  - need C++, Java, or PHP … to do a good job at talking with 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

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
- Without your participation, a lecture is impossible
  - so please attend lectures regularly

Homework Assignments

- Mostly paper-based, some may involve light programming
- Due by noon of the due date
- Submission by PDF to Compass
- No late homework will be accepted

Tutorials: 7 Sessions

- Homework Tutorial:
  - One for each homework assignment, to teach problem solving.

- Exam Tutorial:
  - One for each exam, to review and practice old exams.

- Scheduled a few days before due. Likely at Tu or Th 5pm.
- Will be recorded.

Project:

- Application Track: 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

- System Track: Take an open source DBMS (SQLite) and hack it.
  - strongly encouraged if you already have app experience

- 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 grading
  - 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 quiz.
- Each tests your participation in class.
- Each accounts for 1.5% of class grading.

Tentative Grading Breakdown

- Homework: 15%
- Project: 35%
- Midterm: 20%
- Final: 30%

How Do We Work Together?

Contacting the Staff --

Office Hours

- The best way for asking questions and clarifications
- Will have office hours every day
- See course Web for schedule

Communications

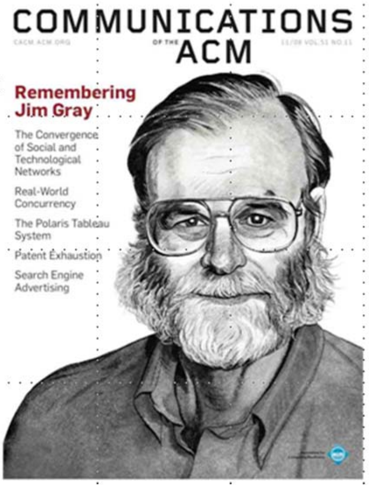
- Website: “Announcements” page
- Newsgroup: class.cs411
  - vitally important!
  - make sure to check it regularly for questions/clarifications
  - announcements will appear here and the course Web
- If you have a question/problem
  1. talk to people in your group first
  2. post your question on newsgroup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

Newsgroup: class.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

Data Management Evolution


COMMUNICATIONS OF THE ACM
Remembering Jim Gray
The Convergence of Social and Technological Networks
Real-World Concurrency
The Polaris Tableau System
Patent Exhaustion
Search Engine Advertising

Jim Gray: *Evolution of Data Management*.
IEEE Computer 29(10): 38-46 (1996):

- Manual processing: -- 1900
- Mechanical punched-cards: 1900-1955
- Stored-program computer-- sequential record processing: 1955-1970
- Online navigational network DBs: 1965-1980
  - many applications still run today!
- Relational DB: 1980-1995
- Post-relational and the Internet: 1995-

Database Management System (DBMS)?

- System for providing EFFICIENT, CONVENIENT, and SAFE MULTI-USER storage of and access to MASSIVE amounts of PERSISTENT data

DBMS Examples

- Most familiar use: many Web sites rely heavily on DBMS's
  - Examples?

- And many non-Web examples

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 = ??

Why Direct Implementation Won't Work

- Storing data: file system is limited
  - size limit by disk or address space
  - when system crashes we may loose data
  - Password/file-based authorization insufficient
- Query/update:
  - need to write a new C++/Java program for every new query
  - need to worry about performance

- Concurrency: limited protection
  - need to worry about interfering with other users
  - need to offer different views to different users (e.g. registrar, students, professors)
- Schema change:
  - entails changing file formats
  - need to rewrite virtually all applications
- That’s why the notion of DBMS was motivated!

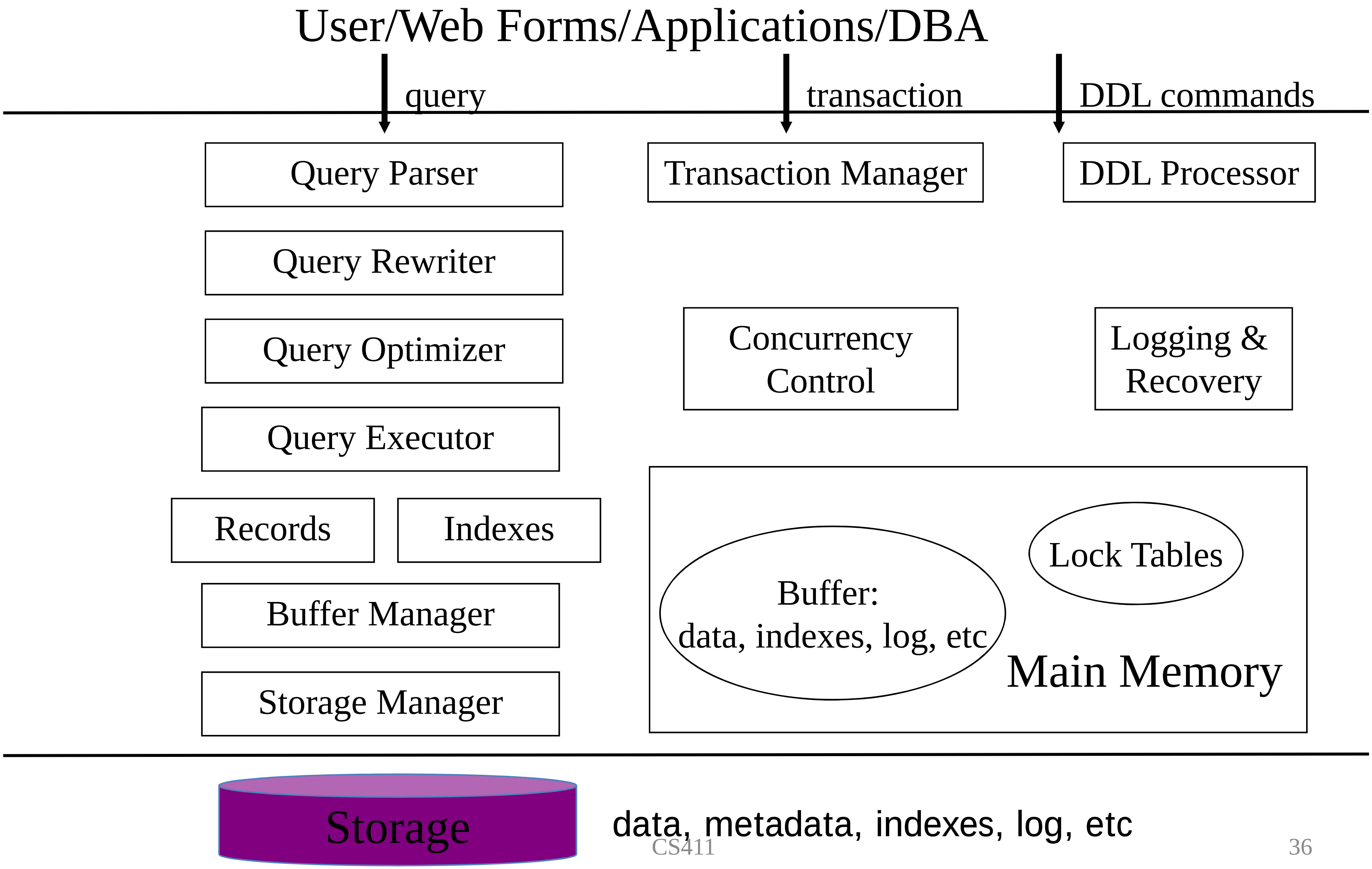
DBMS: More Requirements

- SAFE:
  - from system failures
  - 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.
  - massive data! -> DBMS's carefully tuned for performance

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 Architecture



Data Structuring: Model, Schema, Data

- Data model:
  - conceptual structuring of data stored in database
  - ex: data is set of records, each with student-ID, name, address, courses, photo
  - ex: 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
  - vs. types and variables in programming languages
- 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

Please nominate topics you like to hear about!

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.