Databases Uses and Introduction to the Relational Model

22 October 2012
Lecture 1

Topics for Today

- Managing Data
- Why use a database?
- Describing and Storing Data
- Queries
  - Concurrency
- Structure of a DBMS

- Source:
  - Ramakrishnan and Gehrke 1.1-1.9

Managing Data

- Databases are designed to enable easy management of large quantities of related information
- We will talk about in this course:
  - Database Design
  - Database Analysis
  - Using Databases
  - Concurrency and Robustness
  - Efficiency and Scalability

Database: Past and Present

History

1960s

- First DBMS by GE
- IBM: Information Management Systems (IMS) (SABRE)

Edgar Codd: *Relational Data Model*

- Foundation for most DBMS
- We'll talk about changes to that model next semester

1980s: IBM developed System R → SQL

Today

SQL:

- Standard by ANSI and ISO
- Relatively uniform; some features not implemented by various vendors

Applications (1980s-1990s):

- Data Warehouses: Rich data and storage specialization
- Enterprise Resource Planning
- Management Resource Planning
- Web Backend: PHP, Java, Apache, Javascript

Why a DBMS?

- For *large* amounts of structured data – say 500GB
  - Won't fit in memory!
  - To retrieve it you need something just to deal with the data files
  - Manage moving it on and off disk
  - Concurrency?
- Those are the major parts of a DBMS

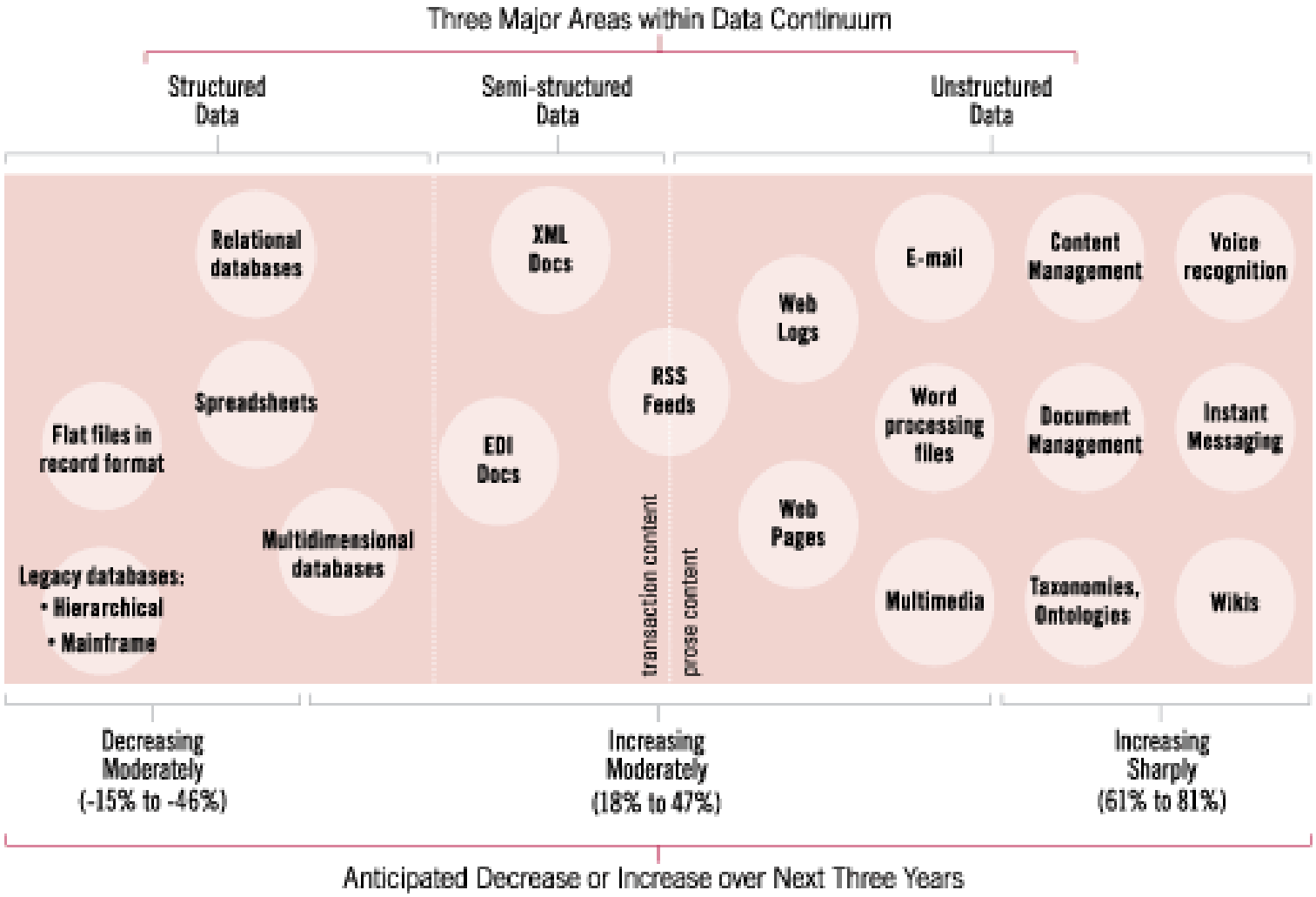
DBMS Advantages

- Data Independence
  - Don’t worry how it’s stored, retrieved, updated
  - Reduced application development time
- Efficient Data Access
  - Manage large quantities, distributed
- Data Integrity and Security
- Data Administration
  - Centralized and uniform management
- Concurrent Access
- Crash Recovery
  - When media fails, power fails

Don’t use a Database when:

- Data is not structured
- Need to do complex manipulations
- Small quantity of data
- Need to manage file or byte level

Structured versus Unstructured



So Far

- Managing Data
- Why use a database?
- Describing and Storing Data
- Queries
  - Concurrency
- Structure of a DBMS

Describing and Storing Data

Data Model: How data is stored

- *Relational Data Model*
- Others: hierarchical, OO

Semantic Model: What the data means

- How data relates to other data
- *Entity-Relationship* semantic model

Relation: A set of records

- Unordered
- Some way to identify each record (a key)

Record: A structured *n*-tuple of fields (columns)

Schema: for a relation is the structure of its records

Example Schema

- Students (*sid:*string, *name*:string, *login*:string, *age:*integer, *gpa:*real)

Sid	*Name*	*Login*	*Age*	*Gpa*
53	Jones	ajones@cs	18	34.5
54	Jones	bjones@cs	30	91.3
12	Smith	smith@is	23	78.2
53	Cohen	cohen@math	19	80.2

- We can do more:
  - Identify rows by some set of fields (key)
  - Constraints on data ranges, types, integrity
  - Link one relation to another
  - Enforce that changing one automatically changes another
- Define a relation? Data Definition Language (DDL)

Other options?

- We didn’t need to use tables

- Object based databases (next semester)
- Hierarchical databases
  - Old, but getting back in vogue with XML
  - That’s another course
- No-SQL databases (popular with web sites)

Queries

Power of a database: Answers hard questions

- How much were sales last year?
- How many sales over $1,000 came during Jan-March from Tiberias?
- Which product produced the most profit in Bet Shean?

Queries: Questions we ask the DBMS:

- Read queries (Query Language)
- Write queries (add/delete/modify) (Data Manipulation Language)

SQL is both (and a DDL too)

Must understand the semantics (meaning) of queries

- Abstract query languages
- Ex: Relational Algebra underlies SQL

How to Query

1. Could log into the DBMS with a command line or GUI interface and ask queries directly.

2. Could use a remote DB tool which builds a connection to a DB and has a GUI to receive user queries and display results.

3. Could write application code which builds a database connection, sends queries, and does things with the results.

4. Could build a web site which has a backend application which runs queries implicitly or explicitly and shows the results on the page.

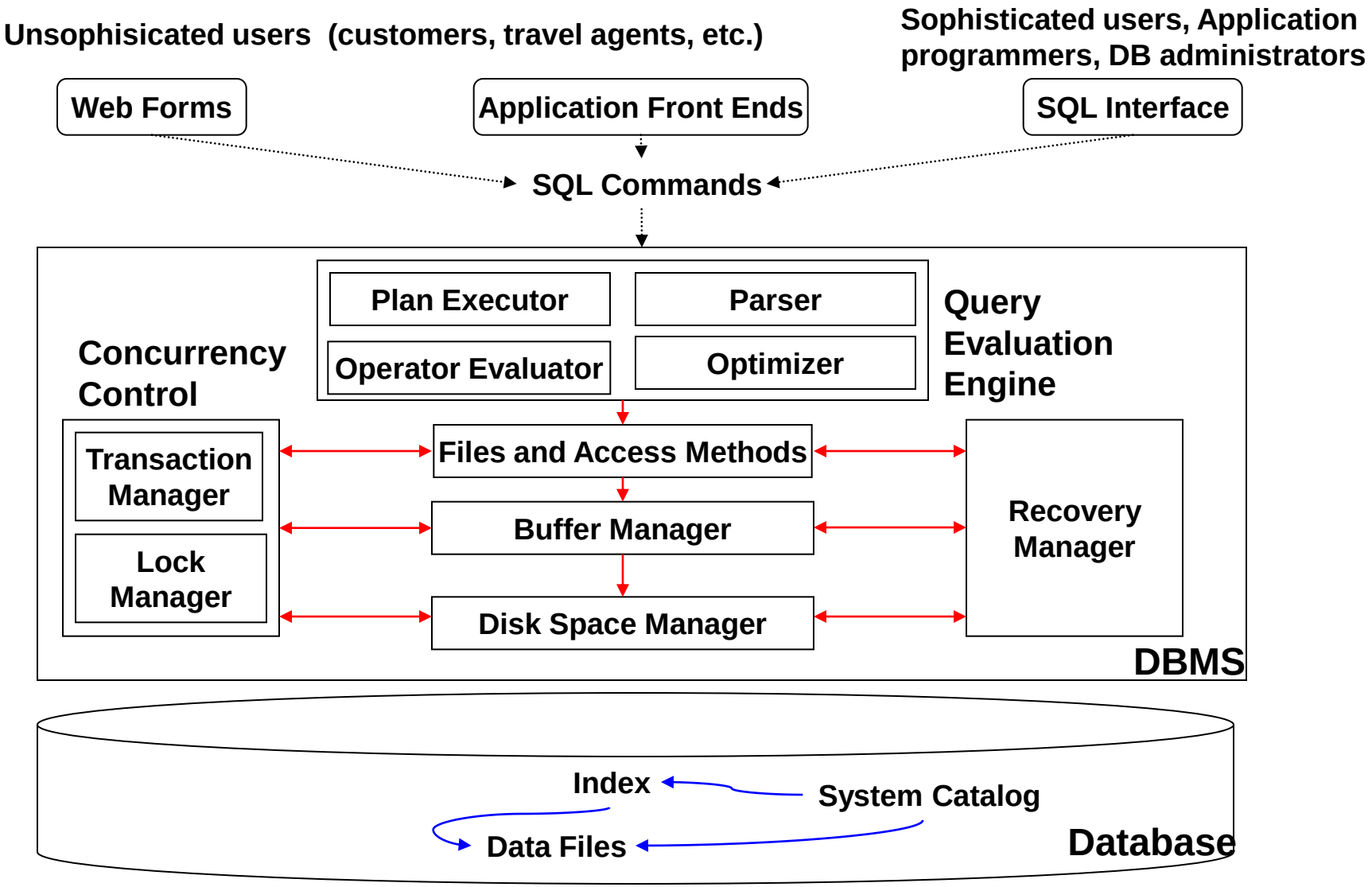
Concurrency

- All modern databases allow multiple concurrent users

- Management is via transactions:
  - *An atomic series of commands from a single connection (user/application)*
- Transaction management is done behind the scenes
  - We'll talk about it more next semester

- Purpose: Many users can work on the same data at once without noticing each other.
- Purpose: Enable orderly recovery from crashes or cut off conversations.

Structure of a DBMS



Summary

- Managing Data
- Why use a database?
- Describing and Storing Data
- Queries
  - Concurrency
- Structure of a DBMS

Personal Introduction

Instructor: **Michael J. May**

Ph.D. in Computer Science University of Pennsylvania, Philadelphia

Courses I teach:

- Information Systems Engineering 1 הנדסת מערכות מידע
- Database Systems מעכרות מסדי נתונים
- Introduction to Computer Networks מבוא לרשתות מחשבים
- E-Commerce and Network Security אבטחת תקשורת ומסחר אלקטרוני
- Database Systems Engineering הנדסת מסדי נתונים

Next year:

- Distributed Information Systems מערכות מידע מבוזרות
- Distributed Algorithms in Network Communication אלגוריתמים מבזורים ברשתות תקשורת

I am also involved in the 4th year projects for Information Systems Engineering students.

About My Courses

My home page: **http://www2.kinneret.ac.il/mjmay**
Every course has a web page and a Telem page

- On the course page:
  - Syllabus, course location and time
  - Lecture schedule
  - Quiz dates
  - Lecture slides (PDF)
  - Targil summaries (if I run them)
  - Reminders about quizzes and assignment due dates (normally)
  - Code, programming data
- On Moodle:
  - Assignment solutions and grades
  - Quiz files and grades
  - Previous tests
  - Other course announcements (changes, extensions, updates, etc).
  - Targil materials (if I don't run them)
  - Lecture slides (PPTX, sometimes and not always complete)

What I expect from you

- Come to class on time (after 10 minutes, the door is locked)
- Don't talk during class or disturb others – raise your hand.
- Read what I ask you to
- Perform and submit assignments on time
  - Late policy (same for all courses)
  - Don't ask for personal extensions
  - Miluim, births, etc. as per the college policy
- Come to office hours if you need extra help
- If you have a complaint or request – try asking me first
- Don't copy or turn in duplicate assignments
  - Group work policy per course
  - "We worked on it together" is not acceptable

- Correct me (respectfully) if I make mistakes, say something silly, etc.

Course Introduction

Topics: Databases, Database Management Systems, and SQL

Prerequisites:

- 1-02-215: Introduction to Information Systems Engineering (may be taken at the same time)
- 1-02-222: File Organization and Processing
- 1-02-220: Logic

Targilim

- Targil will consist of:
  - Exercising what we learned
  - Asking/Answering questions
  - Trying out queries on MS SQL Server
  - Working on Project

- No grade for anything done during Targil

Optional Assignments

- Four self check assignments over the course of the semester

- Exercise concepts and skills explained in class:
  - ERD
  - Schema development
  - Basic SQL
  - More advanced SQL – views, stored procedures

- Will help you in working on your projects

Semester Project

- Design and implement a database and interactive GUI front end using the tools and techniques learned in the classroom
  - Can be done in groups of 2
- Developed in phases over the course of the semester
  - Whole thing graded at the very end
  - Meet with me after each phase submitted
- Final project submission
  - Project Report
  - Project Presentation in last Targil