

CSC 261/461 – Database Systems

Fall 2017

MW 12:30 pm – 1:45 pm

CSB 601

Agenda

- *Administrative aspects*
- *Brief overview of the course*
- *Introduction to databases and SQL*

ADMINISTRATIVE ASPECTS

Teaching Staff

- **Instructor:**
 - Tamal Biswas
- **Graduate TA:**
 - Hensen Wen
- **Undergraduate TAs:**
 - Fuya Xu
 - Xuan Tang
 - Yuton He

How to contact me?

- Piazza
- Office hour
 - Tu 5pm-6pm
 - Immediately after lecture
- E-mail:
 - tbiswas2@cs.rochester.edu (Try to post on Piazza rather than sending email)
- Appointment (the last resort)
 - Friday afternoon (after 3 pm)

Work Load

- Heavy!
- Approx. 30 pages of reading per week
- Midterm exam (Oct 25, 2017)
- Final exam (Dec 13, 2017)
- A semester long project
- 2+ assignments/projects
- Finally, a quiz every week

Grading Policy

- **Quiz (20%)**
 - 10+ Quizzes (each worth 2%)
 - The lowest quizzes will be dropped
- **Exam component (30%)**
 - 1 midterm exam (15%)
 - 1 final exam (15%)
- **Written and programming assignment component**
 - Will not be graded, but a must for quizzes and projects
- **Project**
 - 2+ projects (45 %)
- **Class Participation**
 - Ask Questions in Class. Take part in Discussion
 - Provide Feedback to improve class
 - All the students will receive points between 0 and 5 for class participation

CSC/TCS 461

- **Term Paper**

- Need to review 1+ conference/journal papers, implement, add features, or survey the related topics.
- Students need to submit a report based on their results, surveys or findings.
- **8** additional points (making the total grade 108. The score will scaled to 100 for the final letter grade)

Letter Grades

Percentage score	Letter grade
90-100	A
85-89	A-
80-84	B+
75-79	B
70-74	B-
65-69	C+
60-64	C
55-59	C-
50-54	D
0-49	F

Policy

- Late submission:
 - No late submission allowed
- Incompletes & Make-up exams
 - Not given except in provably extraordinary cases! (see syllabus)



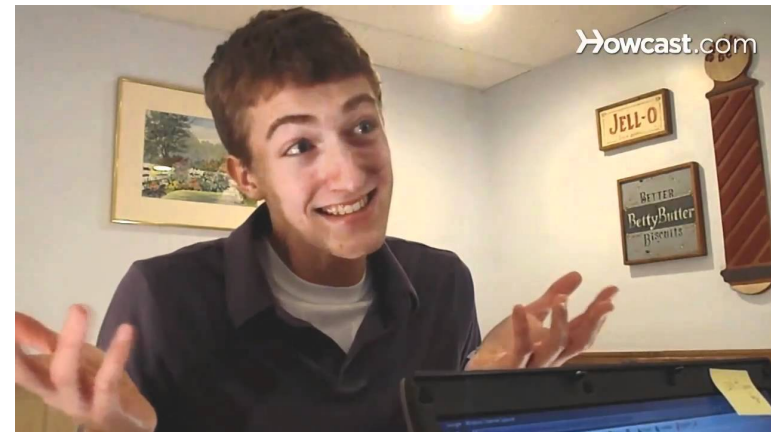
**teachers
call it
cheating,
we call it
teamwork**

Academic Honesty

- On plagiarism:
 - Zero on the particular assignment/exam/project
 - ‘E’ for the course on the second violation!
 - Apply to both parties!
- Group study/discussion is encouraged, but the submission must be your own work!
- Programming:
 - Discussions of ideas are welcomed, but no exchange of codes
 - If you use a piece of code from Mr. Google, say so!
- Please refer
<https://www.rochester.edu/college/honesty/policy.html>

No Lame Excuses, Please

- I want to go home early, can I take the final early?
- I had a fight with my girlfriend
- I've studied very hard, I understood the stuff very well, but I got a C-, please consider giving an A-



How to do well in the course?



How to do well in the course?

study

(verb)

The act of texting, eating
and watching TV with an
open textbook nearby.

How to do well in the course?



How to do well in the course?

- Participate: discuss, answer, ask questions
“the only stupid question is the question you don’t ask”
- Give suggestions, please! I’ll take them seriously
- Study every week
- Start early!

COURSE OVERVIEW

Course website

- <http://www.cs.rochester.edu/courses/261/fall2017/>

What you will learn from this course

- Database Systems

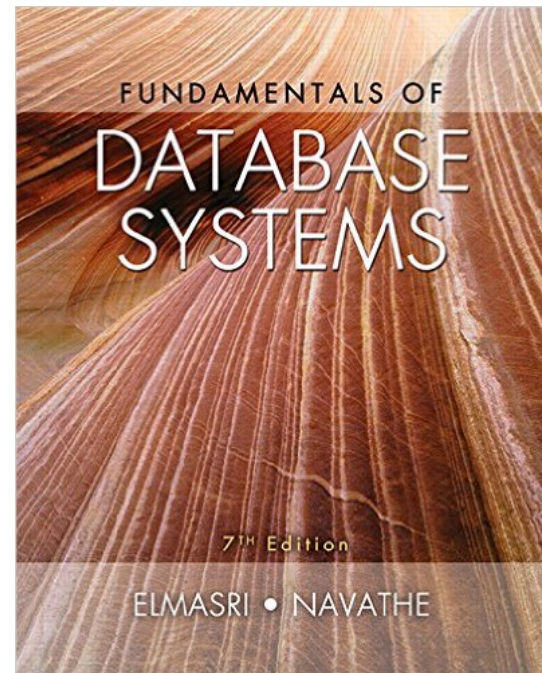
- Efficiently store and retrieve data
- Relational Databases and SQL
- Entity-relationship modeling and database design, Relational Algebra
- Web database programming (using PHP)
- Distributed databases and NOSQL
- Data Mining and Big Data

What you will not learn from this course

- *A lot!*
- Why?
 - Database Systems is a huge topic
 - Big Data
 - <https://www.youtube.com/watch?v=TzxmjbL-i4Y>

Textbook

- [Fundamentals of Database Systems, 7th Edition](#)
 - (ISBN: 0133970779) by
- Ramez Elmasri & Shamkant B. Navathe



INTRODUCTION TO DATABASES AND SQL

Databases and Database Users

- Types of Databases and Database Applications
- Basic Definitions
- Typical DBMS Functionality
- Example of a Database
- Main Characteristics of the Database Approach
- Advantages of Using the Database Approach
- Current Database Technology
- Extending Database Capabilities

Types of Databases and Database Applications

- Traditional Applications:
 - Numeric and Textual Databases
- More Recent Applications:
 - Multimedia Databases
 - Geographic Information Systems (GIS)
 - Biological and Genome Databases
 - Data Warehouses
 - Mobile databases
 - Real-time and Active Databases

Recent Developments

- Social Networks
 - Facebook
 - Twitter
 - Linked-In
- Search Engines
 - Google, Bing, Yahoo
- Big Data storage
- NOSQL (Not Only SQL) systems

Basic Definitions

- **Database:**
 - A collection of **related data**.
- **Data:**
 - Known facts that can be **recorded** and have an implicit meaning.
- **Mini-world:**
 - Some part of the real world about which data is stored in a database. For example, student grades and transcripts at a university.
- **Database Management System (DBMS):**
 - A software package/system to facilitate the creation and maintenance of a computerized database.
- **Database System:**
 - The DBMS software together with the data itself. Sometimes, the applications are also included.

Simplified Database System Environment

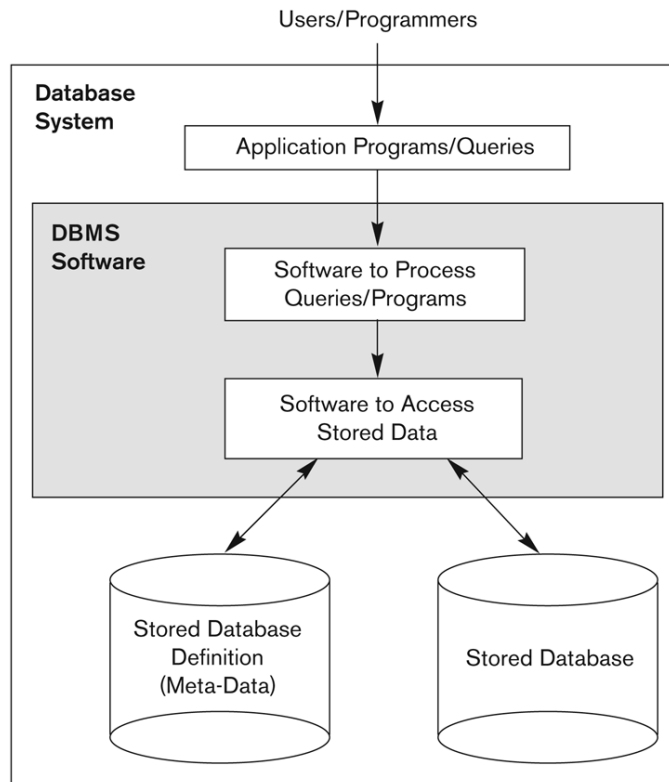


Figure 1.1
A simplified database system environment.

Typical DBMS Functionality

- *Define* a particular database in terms of its data types, structures, and constraints
- *Construct* or Load the initial database contents on a secondary storage medium
- *Manipulating* the database:
 - Retrieval: Querying, generating reports
 - Modification: Insertions, deletions and updates to its content
 - Accessing the database through Web applications
- *Processing and Sharing* by a set of concurrent users and application programs – yet, keeping all data valid and consistent

Example of a Database

- **Mini-world for the example:**
 - Part of a UNIVERSITY environment.
- **Some mini-world *entities*:**
 - STUDENT_s
 - COURSE_s
 - SECTION_s (of COURSE_s)
 - (academic) DEPARTMENT_s
 - INSTRUCTOR_s
- http://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all

Relationships

- **Some mini-world *relationships*:**
 - SECTIONs *are of specific* COURSEs
 - STUDENTs *take* SECTIONs
 - COURSEs *have prerequisite* COURSEs
 - INSTRUCTORs *teach* SECTIONs
 - COURSEs *are offered by* DEPARTMENTs
 - STUDENTs *major in* DEPARTMENTs
- expressed in a conceptual data model, such as the ENTITY-RELATIONSHIP (ER) data model

Example of a simple database

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2

A database that stores student and course information.

Data on the Web and E-commerce Applications

- Web contains data in **HTML** (Hypertext markup language) with links among pages.
- This has given rise to a new set of applications and E-commerce is using new standards like **XML (eXtended Markup Language)** and **JSON (JavaScript Object Notation)**.
- Script programming languages such as **PHP** and **JavaScript** allow generation of dynamic Web pages that are partially generated from a database
- <http://www.cs.rochester.edu/courses/261/spring2017/termpaper/best.html>

Big Data Technologies and NOSQL databases

- **Big Data:**
 - New data storage, management and analysis technology was necessary to deal with the onslaught of data in petabytes a day (10^{15} bytes or 1000 terabytes) in some applications .
 - This started being commonly called as “Big Data”.
- **Hadoop and MapReduce:**
 - Hadoop (which originated from Yahoo) and Mapreduce Programming approach to distributed data processing (which originated from Google) as well as the Google file system have given rise to Big Data technologies.
 - We will learn **Spark**
- **NOSQL:**
 - NOSQL (Not Only SQL- where SQL is the de facto standard language for relational DBMSs) systems have been designed for rapid search and retrieval from documents, processing of huge graphs occurring on social networks, and other forms of unstructured data with flexible models of transaction processing
 - We will learn **MongoDB**

WHY DATABASE?

Questions to ponder

- Why not Lists? Why database?
- How related tables avoid problems associated with lists?

Problems with Lists

- Multiple Concepts or Themes:
 - Microsoft Excel vs Microsoft Access
- Redundancy
- Anomalies:
 - Deletion anomalies
 - Update anomalies
 - Insertion anomalies

http://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all

List vs Database

- Lists do not provide information about relations!
- Break lists into tables
- Facilitates:
 - Insert
 - Delete
 - Update
- Input and Output interface (Forms and Reports)
- Query!

Structured Query Language (SQL)

- http://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all

Again, Why database?

- To store data
- To provide structure
- Mechanism for querying, creating, modifying and deleting data.
- **CRUD** (**C**reate, **R**ead, **U**ppdate, **D**elete)
- Store information and relationships

Last Words before we end today

- You'll learn as much from me as I will learn from you
- Welcome, again!

Acknowledgement

- Some of the slides in this presentation are taken from the slides provided by the authors.
- Thanks to YouTube for providing many useful videos.