School of Computing
CS6530 Advanced Database Systems
Fall 2017

Instructor: Feifei Li, *lifeifei@cs.utah.edu*, `http://www.cs.utah.edu/~lifeifei`, 801-585-6673.
Lectures: T&H 12:25pm to 1:45pm, WEB 1250.
Office Hours: T&H 2:00pm to 3:00pm or by appointment (the best way to reach me is via email.)
TMs: Zhao Chang, *zchang@cs.utah.edu*, office hours: Monday, 1:30-3:00 PM.
Yanqing Peng *ypeng@cs.utah.edu*, office hours: Wednesday, 1:30-3:00 PM.
Zhuoyue Zhao *zyzhao@cs.utah.edu*, office hours: 3:00-4:30 PM.
All TAs hold their office hours at the school's TA office (next to LCR, MEB 3147).

Course Description: Graduate-level course on the design and implementation of (relational) database system kernels, as well as other large-scale data management techniques and systems. Review the relational data model (including relational algebra) and relational query language: SQL. Examine in depth file organization, database storage, indexing and hashing, query evaluation and optimization, transaction processing, concurrency control and recovery, database integrity and security (if schedule allows).

In addition to the study of relational database kernels, this course also investigates latest development in other large-scale data management techniques and systems, e.g., the MapReduce framework (in particular, the Hadoop system), NoSQL systems, Key-Value stores (Cassandra, HBase, Google BigTable), other IO efficient techniques (if time permits), and in-memory computing platforms such as Spark.

Students will participate in a semester-long project and build a mini-database system by implementing several core modules in a relational database system. In summary, this course is about the principles of designing and implementing database kernels, as well as other relevant large data management techniques and systems. Please note that this is NOT a course on building database applications and introduction to database systems, i.e., we will not cover in this course how to build a database application (e.g., ER design, schema refinement, functional dependency, and database application development). Such topics will be covered in CS 5530.

Course Objectives: To learn the principles, algorithms, and system design and implementation issues of implementing a relational database kernel, as well as other large scale data management systems. To understand in depth how a relational database system and other large-scale data management systems work for storing, organizing and querying large amount of data.

Prerequisites: Undergraduate-level courses on the analysis of data structure and algorithms. Students must also possess good programming skills and experiences in Unix systems.

Other preferred knowledge (but not required) include an undergraduate-level introductory course on databases and database applications (e.g., CS5530), and an undergraduate-level operating systems.

Class Home Page: Canvass. We will use the canvass course homepage to disseminate your grade, as well as making important announceents. You are also encouraged to post questions there and participate the discussion forum.

Time and Place: Tuesday & Thursday 12:25 PM to 1:45 PM, WEB 1250.

Required Textbook: R. Ramakrishnan, J. Gehrke. Database Management Systems. Third

Edition. McGraw-Hill 2003, ISBN 0-07-246563-8. Additional reading material may be distributed by the instructor.

Recommended Reading: Database Systems, The Complete Book, 2nd Edition, by Hector Garcia-Molina, Jeff Ullman, and Jennifer Widom. And The Red Book, Readings in Database Systems, 5th Edition (available online).

Collaboration/Academic Honesty All course participants must adhere to the academic honor code of University of Utah. All instances of academic **dishonesty** will be reported to the university. Every student must write his/her own homework/code. Showing your code or homework solutions to others is a violation of academic honesty. It is your responsibility to ensure that others cannot access your code or homework solutions. Consulting related textbooks, papers and information available on Internet for your coding assignment and homework is fine. However, copying a large portion of such information will be considered as academic dishonesty. If you borrow a small piece of any such information, please acknowledge that in your assignment.

Important Dates: Midterm: Tuesday in class, October 17 (the first Tuesday after Fall break). Final: Tuesday in class, December 5. Last day to withdraw from the class: Friday, 10/20/2017.

Grading Policy: The course grade will break down as follows.

Programming Assignments	50%
Midterm Exam	25%
Final Exam	25%

In addition, we will distribute quizes that are not graded, but will provide additional exercising opportunities. Solutions to a written assignment will be released on its due date (but written assignments are NOT graded). Solutions to a quiz will be released immediate followed the quiz itself.

And your final grade will be assigned as follows (Tentative).

92-100	A	75-82	B+	60-66	B-	45-52	C
83-91	A-	67-74	B	53-59	C+	35-44	C-
25-34	D	15-24	E	0-14	F		

Course Policy

1. You are allowed to discuss written assignments.
2. The programming project will be carried out in a team of two.
3. Most of the programming projects will be based on Java. You will be provided with skeleton of codes with clearly defined API and header files as well as the Makefile and test programs. Your responsibility is to fill in the missing implementation for various functions. More information will be posted when the first module is out.
4. You are bound to attend all lectures unless notifying the instructor in advance with reasonable excuses.
5. Usually, Written assignment is due one week after it is out. The time allocated to the programming assignment will vary depending on the difficulty of each individual assignment.

Late Policy – Make up exams: Late assignments will not ordinarily be accepted. If, for some compelling reason, you cannot hand in an assignment on time, please contact me as far in advance as possible. No credit will be given to late programming projects. No make-up exams (except under extremely unusual circumstances).

More about the project: The semester-long project aims to build a mini-database system. There will be several major modules that you will work on, for example, the HeapFile for the database storage engine, the buffer management module, the disk-based B+ tree index, the join operator, etc. There will also be projects on other relevant large scale data management techniques, such as external merge-sort for sorting large scale data. We will ask each team to set up a private bitbucket repo for keeping track of your submissions.

Tentative Course Schedule

Week#	Topics	Readings
1	Introduction; Overview of a Database System Relational Model; SQL	Chapter 1 Chapter 3
2	SQL Storage and Files; Written assignment 1 is out	Chapter 3 Chapter 9
3	Storage and Indexing Project 1 is out	Chapter 8
4	Tree Structured Indexing Written assignment 2 and project 2 are out	Chapters 10
5	Hash-Based Indexing	Chapter 11
6	External Sorting Written assignment 3 is out Project 3 is out	Chapter 13
7	Query Evaluation and Optimization	Chapters 12, 14
8	Query Evaluation and Optimization (continued) Transaction Written assignment 4 is out Project 4 is out	Chapter 15 Chapter 16
9	Transaction and Concurrency Control	Chapters 16, 17
10	Crash Recovery Written assignment 5 is out	Chapter 18
11-12	MapReduce Framework, Spark, Spark SQL Written assignment 6 and project 5 are out	Additional Material
13-14	NoSQL systems, Key-Value stores	Additional Material
15	Course wrap-up	