

CS222/CS122C Fall 2017: Principles of Data Management

Course Personnel
Announcements
Syllabus (Subject to Change)
Projects
Schedule & Location
Course Information
Course Description
Grading Breakdown
Textbooks
Other Readings
Class Discussion Forum
Programming Project
Comprehensive Exam Option
Notice about Projects
Project Late Policy
Cheating Policy
C/C++ Resources
Midterm Exam
Final Exam

Course Personnel

Instructor:

Chen Li

E-Mail: chenli AT ics DOT uci DOT edu

Office: DBH 2092

Office hours: Mondays 2:15 pm - 3:15 pm and Thursdays 9:45 am - 10:45 am

Assistant:

Jamshid Esmaelnezhad

E-mail: jesmaeln AT uci DOT edu

Office hours: Thursdays, 1-2pm and 2:15-3:15pm, CS building (building number 302) room #424F (or if I wasn't there, room #424A).

Seungjin Lee

E-mail: seungl13 AT uci DOT edu

Office hours: TBA.

Online Discussion: <https://piazza.com/uci/fall2017/cs222cs122c>

- Please use Piazza properly. It's a place for students to exchange ideas. Don't post "easy" or "random" questions without much thinking.
- To encourage students to actively participate in Piazza discussions and provide high-quality answers, we will select two students with the best Piazza performance. Each student will get 2% extra credits in the overall scores.

Announcements

- The deadline to drop the course is Oct 5, Thursday, before the lecture. After that, "you are in"!
- For students who are still on the waiting list, please continue attending the lectures.
- For all students, please notice that the first project is already online, and you are encouraged to start working on it soon.

Syllabus (Subject to Change)

ID	Date	Topic	Readings	Notes
1	09/28/17, Th	DBMS Architecture	Ch. 1; Chamberlin et al 1981	PPT1, Setup MySQL
2	10/03/17, Tu	Project 1, Disks and Files	Ch. 9.1, 9.7; Stonebraker 1981	PPT2
3	10/05/17, Th	Heap Files, Page Formats, Buffer Manager	Ch. 9.2-9.6	PPT3
4	10/10/17, Tu	Buffer Management, RAID, Catalogs, Project 2	Ch. 9.2, 9.4, 12.1	PPT4
5	10/12/17, Th	Schema versioning, PAX , Column Stores, OLTP/OLAP, File Organizations	Ch. 8.4	PPT5
6	10/17/17, Tu	Index Overview and ISAM Tree Index	Ch. 8.4, Ch. 10	PPT6
7	10/19/17, Th	B+ tree	Graefe 2011 (Sec. 1-3)	PPT7
8	10/23/17, Tu	Static Hashing, Extendible Hashing, Linear Hashing	Ch. 11; Fagin et al 1979; Litwin 1980	PPT8
9	10/25/17, Th	Indexing and Performance	Ch. 8.4-8.6	PPT9

	Th			
10	10/30/17, Tu	Project 3, Indexing and Performance	Ditto	Ditto
11	11/02/17, Th	Ditto	ditto	Ditto
12	11/07/17, Tu	In-class midterm		
13	11/09/17, Th	External Sorting	Ch. 13	PPT10

Projects

The project is structured as a four-part, team-based (teams of one or two students), software development exercise:

Project	Due Date	Topic	Teams	Proportions (50%)	Private Tests
Project 1	Mon, week 2, Oct. 9	Record-Based File Manager (RBF)	Solo Project	9%	Private Tests
Project 2	Mon, week 5, Oct. 30	Relation Manager (RM)	Solo or Pair Project	14%	
Project 3	Mon, week 8, Nov. 20	Index Manager (IX)	Solo or Pair Project	14%	
Project 4	Fri, week 10, Dec. 8	Query Engine (QE)	Solo or Pair Project	13%	

[Project team sign-up sheet](#)

Schedule & Location

Lecture

Days: Tuesdays and Thursdays

Time: 11:00-12:20p

Location: **MSTB 118**

Course Information

Course Description

This course provides an implementor's view of database management systems. It covers fundamental principles and implementation techniques, issues, and trade-offs related to database management systems. Topics covered include storage management, buffer management, record-oriented file systems, access methods, query processing, and query optimization. This course is a must for those students wishing to explore database management as an area of research and/or who plan on taking CS223 or CS224. A significant portion of database systems research as well as industrial database and information system development deals with adapting the basic database techniques covered in this course to new advances in hardware and software technologies or to the requirements of different applications and data types.

Prerequisites: CS 122A "Introduction to Data Management" and (ICS53 - "Principles in System Design" or CS143A - "Principles of Operating Systems"). Please don't attempt this class without the required background! Doing so has not proven to be a great idea for students historically. :-)

Grading Breakdown

Midterm Exam: 25%

Final Exam: 25%

Programming Projects: 50%

This mixture of grading criteria for CS222 is intended to give equal "excelling opportunities" to both thinkers and do-ers. It's also intended to avoid overly weighting either team-based performance or individual performance. **This is a hands-on project class, so expect to put a significant effort into your projects.** The exams will be comprehensive with respect to the course material and will also ping you individually with respect to project knowledge, so be sure that you and your project partner make design decisions together. The two exams will be closed-book, but you may bring an 8.5"x11" two-sided cheat sheet to each exam if you like.

For all the graded projects and exams, if you disagree with the grading, you can discuss with us within two weeks after they are returned. After that, all the grades will be finalized.

Textbooks

Required: *Database Management Systems*, 3rd edition, by R. Ramakrishnan and J. Gehrke, McGraw Hill, 2003.

Recommended: *Readings in Database Systems*, 4th edition, by J. Hellerstein and M. Stonebraker, MIT Press, 2005.

Other Readings

The following papers are mostly drawn from the Hellerstein and Stonebraker book or can be found via UCI's ACM Digital Library subscription (<http://dl.acm.org>). You will be responsible both for the material covered in the lectures and the material covered in the readings listed in the Syllabus below.

- Abadi, D., Madden, S., and Hachem, N. 2008. Column-stores vs. Row-stores: How Different are They Really?. *Proc. ACM SIGMOD Intl. Conf. on Management of Data* (Vancouver, Canada, June 2008).
- Alsubaiee, S., et al., 2014. AsterixDB: a scalable, open source BDMS. *Proc. VLDB Endow.* 7, 14 (Oct. 2014), 1905-1916.
- Chamberlin, D., et al, 1981. A history and evaluation of System R. *Commun. ACM* 24, 10 (Oct. 1981), 632-646.
- Dean, J., and Ghemawat, S., 2004. MapReduce: Simplified data processing on large clusters. *Proc. 6th USENIX Symp. on Op. Sys. Design and Impl. (OSDI'04)* (San Francisco, CA, December 2004).
- DeWitt, D. and Gray, J. 1992. Parallel database systems: The future of high performance database systems. *Commun. ACM* 35, 6 (June 1992), 85-98.
- Fagin, R., et al, 1979. Extendible Hashing—A fast access method for dynamic files. *ACM Trans. Database Syst.* 4, 3 (Sept. 1979), 315-344.
- Graefe, G., 2011. Modern B-Tree Techniques. *Foundations and Trends in Databases* 3, 4 (Sept. 2011) 203-402. (Available from the UCI intranet at: <http://dx.doi.org/10.1561/19000000028>)
- Guttman, A. 1984. R-trees: A dynamic index structure for spatial searching. *Proc. ACM SIGMOD Intl. Conf. on Management of Data* (Boston, Massachusetts, June 1984).
- Litwin, W., 1980. Linear hashing: A new tool for file and table addressing. *Proc. 6th Int'l. Conf. on Very Large Data Bases*, Montreal, Oct. 1980.
- Nievergelt, J., Hinterberger, H., and Sevcik, K. C. 1984. The Grid File: An adaptable, symmetric multikey file structure. *ACM Trans. Database Syst.* 9, 1 (March 1984), 38-71. (optional)
- Selinger, P. G. et al, 1979. Access path selection in a relational database management system. *Proc. ACM SIGMOD Int'l. Conf. on Management of Data* (Boston, Massachusetts, May-June 1979).
- Shapiro, L. D. 1986. Join processing in database systems with large main memories. *ACM Trans. Database Syst.* 11, 3 (Aug. 1986), 239-264.
- Stonebraker, M. 1981. Operating system support for database management. *Commun. ACM* 24, 7 (July 1981), 412-418.

Class Discussion Forum

We will be using Piazza for online class discussions. This system is highly catered to getting you help fast and efficiently from classmates, the Reader(s), and the instructor. Rather than emailing questions to any of us on the teaching staff, we'll ask you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@.... Please make Piazza participation (at least monitoring the activity there) a part of your regular CS122C/222 routine!

You can find our class page at: <https://piazza.com/uci/fall2017/cs222cs122c>

Programming Project

This class is intended to teach the principles involved in DBMS implementation, so it will include a significant programming component. The project will be aimed at giving you a sense of what goes on inside a DBMS, and what the issues and challenges are in building a system "now" that will be managing data of various user-defined shapes and sizes "later". You will try your hand at record-based file storage, indexing, and relational runtime system programming (a.k.a. query operators). The language for the project will be C++ - hopefully you either have C++ experience or you have Java experience and can learn C++ quickly. Our assistant(s) will run the project, and they can provide you with additional suggestions for material to help you "spin up" on C++ programming, debugging, and tools in case you need to do a bit of remedial work on the side in order to tackle the project successfully. The course has the same requirements for both graduate students and undergraduate students.

Comprehensive Exam Option

Computer Science graduate students wishing to satisfy the MS Comprehensive Exam requirement in Database Systems via CS222 should notify the instructor and the Student Advising Office (SAO) of this intention via e-mail before the day of the final exam. For such students, a weighted mix of your midterm and final exam performance will be used to determine your MS Comprehensive Exam grade (P or F) at the end of the course, and this grade will be reported to SAO. Students who do not pass successfully can try the exam again the next time CS 222 is offered by taking the midterm and final exams in that offering of the course; hopefully this will not be necessary for anyone. (Students who do this are advised to informally audit the next offering of the class; this is because the course emphasis and expected knowledge can vary a bit by term and by instructor, and also because you'll probably need the review to pass.)

Notice about Projects

- **Making teams:** Project grades will be team based. Students should work on project 1 individually, but they can form teams of up to two students starting from project 2. Please let us know about your group by adding your team members' names [here](#) in front of your group id (please do so even if you are alone in your group). A student can always leave his/her team during the quarter with a notice at least two weeks prior to the next project's deadline. Also, please note that it is not permitted to join a new team after splitting.
- The projects are doable by individuals, which can be more challenging but rewarding.
- Students are allowed to add attributes, methods, and classes, as long as the public signatures provided in the codebase are kept and implemented. We will be testing your code by calling the methods that are declared in the signatures.
- Students also have the freedom to use other open-source libraries or packages (e.g., Gtest and Boost) to implement and test their code. Just please make sure that you: (1) get the okay from us before using any additional packages; (2) list such packages clearly in your project document; and (3) write your makefile correctly so that your project can still be built by calling the 'make' command.

Project Late Policy

- The official due date for each assignment is listed here on the wiki, and it is expected that students will turn most work in on or before that date.
- We will offer a 48-hour "grace period" for each assignment, and will therefore accept solution submissions that are turned in within 48 hours of the due date (i.e., less than two days late). No need to ask - this time is yours in the event that you need it for some unforeseen reason. If you use it, your score will be deducted by 10% (out of 100%). You are recommended NOT to use the grace period, as you will lose not only 10%, but also valuable time from the next assignment's working period for every hour that you run late.
- Late assignments after the grace period will NOT be accepted, so do always aim to be on time! (Please don't even ask, as this is what the two-day grace period is intended for. :-))

Cheating Policy

Cheating is an area where the instructor for this course has absolutely no patience or sympathy whatsoever. You are all here to learn, and cheating defeats that purpose and is unfair to your fellow students. Students will be expected to adhere to the UCI and ICS Academic Honesty policies (and should see <http://www.editor.uci.edu/catalogue/appx/appx.2.htm#academic> and http://www.ics.uci.edu/ugrad/policies/index.php#academic_honesty to read their details). We run a software package to search for cheating cases in projects. Any student found to be cheating or aiding others in cheating will be academically prosecuted to the maximum extent possible — so you may fail the course with no option for a second chance. This policy will be strictly adhered to; no exceptions will be possible regardless of the impact it may have on your studies, your work plans, your visa status, or anything else. Don't cheat and you won't have issues. Just say *no* to cheating!

C/C++ Resources

- A helpful reference for C/C++ libraries: <http://www.cplusplus.com/reference/>
- A tutorial on programming in C/C++ and guide on related resources: <http://www.cprogramming.com/tutorial.html#c++tutorial>
- A tutorial on learning C++ for Java programmers: <http://pages.cs.wisc.edu/~hasti/cs368/CppTutorial/index.html>

Midterm Exam

Time: Nov. 7, Tuesday, In-class. Closed books. You can bring an A4-sized "cheat sheet." [sample midterm](#)

Room Assignment

Overflow room location: [SSTR 101](#)

Final Exam

Tuesday, December 12, 2017, 10:30 a.m. - 12:30 p.m

Closed books, Closed notes. You can bring an A4-sized "cheat sheet." The exam will cover all materials taught in this quarter, with a more emphasis on those after the midterm.

► Attachments