CIS 6500 – Advanced Topics in Database Systems

Welcome to CIS 6500! We're going to learn about databases and build some stuff.

Instructor	Ryan Marcus
Email	rcmarcus@seas.upenn.edu
Office	Towne 219C
Office hours	Monday 5pm-6pm, Tuesday 1-2pm OR by appointment
Course website	Canvas
Favorite animal	Penguins
Ryan's website	https://ryanmarc.us

Course overview

CIS 6500 is a "seminar" style course where you will learn about the internals and implementation of database systems. Specifically, this course will focus on "analytics" or OLAP database systems, along with recent advances in machine learning that aid those systems. Topics will include indexing, query processing and compilation, query optimization, learned systems, and instance-optimized systems.

Prerequisites

This course will assume students are familiar with:

- The basic operation and usage of a SQL database (e.g., issuing queries, inserting data, creating tables, creating indexes)
- Data structures and algorithms knowledge
  - Hashing
  - Sorting
  - Big-O notation (asymptotic behavior)
- Programming knowledge, as the course involves a significant programming project
  - Some familiarity with a "systems" programming language, like C/C++ or Rust, will be helpful, but is not strictly required.
- Basic principles of statistics and probability

Learning goals

After completing this course, you should be able to:

1. Pick up and read an academic paper about database systems (e.g., from VLDB or SIGMOD)
2. Have practice giving a lecture and provoking discussion about an advanced technical topic
3. Understand the tradeoffs involved in different data layouts
4. Understand the principles of efficient query execution in modern systems
5. Understand the basic principles of query optimization in modern systems
6. Develop a deep understanding of index structures and access methods
7. Learn how machine learning techniques can improve (and complicate) traditional database management systems

Grade breakdown & course structure

Grade breakdown:

Short paper summaries	10%
Long paper summary	10%
Lecture and seminar	30%
Final exam	20%
Final project	20%
Attendance and participation	10%

This course will be split into two parts:

- Initial lectures: for the first three weeks of the class, there will be traditional lectures and discussions to ensure everyone has an appropriate level of background knowledge. These lectures will include instruction on how to read a research paper, along with basic database concepts.
- Seminar lectures: the remainder of the class time will be used for students to present papers. Each student will be assigned a paper and a class period, and on that class period the student will give a brief lecture and lead the class in a discussion about the paper. Students assigned to earlier class periods will get extra credit.

You will be responsible for several assignments:

1. Your paper presentation. When it is **your turn** to present:
    1. **Due 1 week before your seminar:** you will write a 3-page summary of the paper, using LaTeX, which should give an intuitive overview of the paper, highlighting important experimental results and including a worked example not found in the paper. These 3-page summaries will be combined together at the end of the semester to give everyone an annotated bibliography of the semester. Upload both a PDF and source files to Canvas. After

your summary is graded, you are welcome to work with me to correct any errors and make any needed improvements, and I will update your grade accordingly.

2. **On the day of your seminar:** you will plan to give an approximately ~45m lecture about the paper you were assigned, and then lead the class in a ~45m discussion. It is of **extreme importance** that you take this seriously, as if you do not put sufficient effort into your seminar, the entire class will suffer. You can work with me ahead of your seminar as much as you'd like to get your seminar ready. Aside from uncontrollable sickness or other medical emergencies, there is **no way to makeup or redo your seminar.**

2. Paper summaries. When it is **not your turn** to present:
    1. **Due before class**: you will write a 1 paragraph (maximum 300 words) summary of the paper, and turn it in via Canvas. These will be graded based on completion, *but keep in mind that you may bring all of your 1 paragraph summaries, as they were submitted, as "notes" to the final exam.*
    2. **During class**: actively engage with the lecture! You will be up there soon – don't leave your classmates hanging.
3. Final project: you will propose a final project **Oct 18**, and the final projects and reports will be due on **Dec 11.**
    1. Proposal: You will write < 1 page document describing what you will set out to do. More details will be provided about the project proposal on the last day of initial lectures.
    2. Projects and reports: a 2-5 page document describing what you did, what worked, and what didn't, along with any code for your project.
    3. Example final projects:
        1. Implementing and experimenting with an idea from a paper in the course, such as learned index structures or multi-armed bandits. Note that many of the papers will not be presented in class until after the proposal deadline, so you should skim the list of papers to see if anything strikes you as particularly interesting.
        2. A topic from your own research that is similar to a topic discussed in this class.
        3. "Canned" project 1: exploring the PostgreSQL query optimizer. Execute three workloads (for example, the JOB, TPC-C, and Stack) on PostgreSQL, computing the expected and actual cardinality estimates of each subpart. Analyze the cardinality estimation errors: do they correlate with the number of joins? Size of the table? Etc. Suggest a way to improve the estimates, and test it.
        4. "Canned" project 2: data structure implementation. Implement a column sketch or learned index on top of a data file, and analyze the performance of that data structure for a number of queries. Suggest a way to improve the structure, and test it.

Attendance and (lack of) virtual options

All classes will be held in person barring extreme circumstances. Since seminar courses are centered around discussion, the course will **not** be recorded so that students do not need to fear their questions or comments being immortalized. As a result, (1) participation and discussion is highly encouraged, and I suggest you bring a pen(cil) and paper to take notes and leave your electronics in your bag (still bring them – some lectures will have online interactive components), although this is not mandatory and you are welcome to take electronic notes if you prefer. (2) There is essentially no way to "make up" a missed class, so I will expect students miss no more than 4 class periods throughout the semester.

Schedule

Exact dates of each paper are subject to change. Proposal and final project deadlines will not be moved.

Date	**Assignment**	**Points**
August 30th	Cancelled!	
Part 1: Initial Lectures and Seminar Assignment		
September 6th	Welcome, syllabus, DB overview	
September 11th	Architecture of a DBMS	
September 13th	An overview of data layout and compression	
September 18th	An overview of query execution and index structures	
September 20th	How to read a research paper	
September 25th	An overview of query optimization	
September 27th	Teaser of machine learning for systems, caching, branch prediction, and final projects	
Part 2: Rows and Columns		
October 2nd	Key question: how do column stores take advantage of compression? "Integrating compression and execution in column-oriented database systems" Abadi et al. SIGMOD '06	+ 20/100
October 4th	Key question: how much, if any, of the advantages of column stores can we get from a row store? "Column-Stores vs. Row-Stores: How Different Are They Really?" Abadi et al. SIGMOD '08	+ 19/100
Part 3: Execution Models		
October 9th	Key question: how does one decompose a query execution plan into compile-able steps? What benefit does this bring?	+ 18/100

	"Efficiently compiling efficient query plans for modern hardware" Neumann. VLDB '11	
October 11th	Key question: compare and contrast compiled and vectorized execution engines. What are the pros and cons of each? "Everything you always wanted to know about compiled and vectorized queries but were afraid to ask" Kersten et al. VLDB '18	+ 17/100
Part 4: Index Structures		
October 16th	Key question: how dose the design of index structures change when we consider caching? "Making B+- trees cache conscious in main memory" Rao et al. SIGMOD '00	+ 16/100
October 18th	Key question: what is a column sketch, and how is it uniquely suited for column-oriented analytics databases? "Column Sketches: A Scan Accelerator for Rapid and Robust Predicate Evaluation" Hentschel et al. SIGMOD '18 **Project proposals due!**	+ 15/100
October 23rd	Key question: can statistical or learned models lead to improved index structures? If so, what is it about learned models that make performance better or worse? "The Case for Learned Index Structures" Kraska et al. SIGMOD '18 (can skip S4 and S5) "Benchmarking Learned Indexes" Marcus et al. VLDB '21	+ 14/100
October 25th	Key question: how do you design a workload-aware index for queries with predicates on multiple columns? Why can't you just use multiple single-column indexes? "Learning Multi-dimensional Indexes" Nathan et al. SIGMOD '20	+ 13/100
Part 5: Join Algorithms		
October 30th	Key question: what are the main problems with naive sort-merge-join? How can the algorithm be improved?	+ 12/100

	“Massively parallel sort-merge joins in main memory multi-core database systems” Albutiu et al. VLDB ‘12	
November 1st	Key question: what are the techniques and algorithms that make hash joins work well? Are there specific conditions or requirements for a hash join to have good performance? “To Partition, or Not to Partition, That is the Join Question in a Real System” Bandle et al. SIGMOD ‘21	+ 11/100
November 6th	Key question: can statistical or learned techniques improve something as fundamental as a sorting algorithm? “The Case for a Learned Sorting Algorithm” Kristo et al. SIGMOD ‘20	+ 10/100
November 8th	Key question: is there any advantage to using a learned function as a hash function? “Can Learned Models Replace Hash Functions?” Sabek et al. VLDB ‘22	+ 9 / 100
November 13th	Key question: even if we have a really good hash join, if we order the joins incorrectly, the query will still be slow. Is there another way? “Adopting worst-case optimal joins in relational database systems” Freitag et al. VLDB ‘20	+ 8 / 100
Part 6: Query Optimization		
November 15th	Key question: what is the “state of the art” in query optimization today? Where do current query optimizers struggle? “How Good Are Query Optimizers, Really?” Leis et al. VLDB ‘15	+ 7 / 100
November 20th	Key question: how can learned models help improve cardinality estimation? What are the advantages and disadvantages? “Learned Cardinalities: Estimating Correlated Joins with Deep Learning” Kipf et al. CIDR ‘19	+ 6 / 100
November 27th	Key question: these query optimizers sure seem complicated! Can we get the computer to just *learn* a model for us? “Balsa: Learning a Query Optimizer Without Expert	+ 5 / 100

	Demonstrations” Yang et al. SIGMOD ‘22 Optional, related: “Neo: A Learned Query Optimizer” Marcus et al. VLDB ‘19	
November 29th	Key question: if techniques like Balsa have a high potential for failure, are there more practical, less error prone ways to improve a query optimizer with learning? “Bao: Making Learned Query Optimization Practical” Marcus et al. SIGMOD ‘21	+ 4 / 100
December 4th	Key question: if an optimizer is going to see the same query many times, what techniques can we use to get better performance than simply running a traditional query optimization algorithm each time? “Kepler: Robust Learning for Faster Parametric Query Optimization” Doshi et al. SIGMOD ‘23	+ 3 / 100
December 6th	Key question: most query optimizers pick a query plan and then ship that plan off to the executor. What if we mixed together query planning and query execution? What are the advantages and disadvantages? “Eddies: Continuously Adaptive Query Processing” Avnur et al. SIGMOD ‘00	+ 2 / 100
December 11th	Key question: query optimization is only a small part of the puzzle of making a “smart” database. What other challenges and potential for automation exists?” “Self-Driving Database Management Systems” Pavlo et al. CIDR ‘17 **Final projects due!**	+1 / 100
TBD	Final exam	

Academic integrity

This course uses the academic integrity policy used by Professor Caldwell for MUSC 2300:

Intellectual development requires honesty, responsibility, and doing your own work. Taking ideas or words from others -- plagiarism -- is dishonest and will result in a failing grade on the paper or assignment and possibly other disciplinary actions. If you are unsure about what constitutes plagiarism, ask me or consult Academic Integrity at the University of Pennsylvania: A Guide for Students, which can be found here: https://catalog.upenn.edu/pennbook/code-of-academic-integrity/

A note on generative AI (e.g., ChatGPT): I view ChatGPT as a writing aid, you may use it freely for any part of this course. You do not need to "cite" or consider ChatGPT to be "an author." Keep in mind that large language models are prone to hallucination, and will happily produce vapid nonsense. Responsibility for verifying the correctness of your answers and claims rests solely on you. "But ChatGPT said X!" is not a valid argument for the correctness of X.

Communication and getting help

The best way to get help about a course-related matter is by posting on the course's Ed Discussion page. There, you can post anonymously, or privately only to me and the TAs (this course does not currently have a TA). You can also email me, but I am less likely to give a quality response via email.

Wellness

The Weingarten Center offers a variety of resources to support all Penn students in reaching their academic goals. All services are free and confidential. To contact the Weingarten Center, call 215-573-9235. The office is located in Stouffer Commons, 3702 Spruce Street, Suite 300.

Academic Support

Learning consultations and learning strategies workshops support students in developing more efficient and effective study skills and learning strategies. Learning specialists work with undergraduate, graduate, and professional students to address time and project management, academic reading and writing, note-taking, problem-solving, exam preparation, test-taking, self-regulation, and flexibility.

Undergraduates can also take advantage of free on-campus tutoring for many Penn courses in both drop-in and weekly contract formats. Tutoring may be individual or in small groups. Tutors will assist with applying course information, understanding key concepts, and developing course-specific strategies. Tutoring support is available throughout the term but is best accessed early in the semester. First-time users must meet with a staff member; returning users may submit their requests online.

Disability Services

The University of Pennsylvania is committed to the accessibility of its programs and services. Students with a disability or medical condition can request reasonable accommodations through the Weingarten Center website. Disability Services determines accommodations on an individualized basis through an interactive process, including a meeting with the student and a review of their disability documentation. Students who have approved accommodations are encouraged to notify their faculty members and share their accommodation letters at the start of each semester. Students can contact Disability Services by calling 215-573-9235.