USC Viterbi
School of Engineering

Course ID and Title: [550 Advanced Data Stores]
Units: 4
Term—Day—Time: [Spring 2025] — [TuTh 10–11:50 am]
Location: [Social Sciences Building (SOS) B2: Map Link]

Instructor: Ibrahim Sabek
Office: Ginsburg Hall GCS 505C
Office Hours: TuTh 4:00–5:00 pm (Location TBD)
Contact Info: sabek@usc.edu

Catalogue Description
Foundations of Data Management and Processing, Modern Database System Architectures, OLAP and OLTP Indexing, Query Pipelining, Materialization, Vectorization, Scheduling, Join Algorithms, Optimization, Consistency, Concurrency Control

Course Description
With the unprecedented volumes of data we face, database and data-intensive systems had to be revolutionized to remain efficient. During the last two decades, the data management and processing fields have exploded, especially with the massive decreases in the storage and computation costs that encouraged the development of scalable techniques. In this course, we provide a comprehensive introduction to the modern techniques and foundations of large-scale data management and processing while covering topics such as indexing for different workload types, query pipelining, vectorized execution, query optimization, concurrency control, and consistency. We also delve into the modern database system architectures that exploit specialized hardware (e.g., GPU, NVRAM, RDMA) and serve specific application requirements (e.g., in-memory, streaming, hybrid analytic, and transaction workloads). To show the practicality of these modern system architectures, we study some popular real systems, such as Snowflake and Spark, as use cases of them.

The course is designed for both MSc and PhD students who are interested in understanding the concepts of building efficient data systems and exercising such concepts using a hands-on project. Classes will be a mix of the instructor's lectures and student-led presentations on representative papers published in top database venues.

Learning Objectives

- Introducing the modern fundamentals of database management and analysis.
- Familiarizing the students with the current database system architectures.
- Evaluating the strengths and weaknesses of different approaches in the same topic.
- Improving the teamwork spirit and collaboration skills.

Prerequisite(s): CSCI-485 or CSCI-585 or permission of the instructor.

Recommended Preparation: Students are strongly recommended to be familiar with the SQL query language and the basics of database management systems that are covered in CSCI-485 and/or CSCI-585, including indexing structures (hash and B+-tree), query optimization, query execution, concurrency control, transaction processing techniques, and crash recovery protocols. Students must have good programming skills, preferably in C/C++ languages.

Course Notes

The grading type will be **Letter,** and its numerical mapping is described below. We will use Brightspace to post the syllabus, homework, and other class information. We will use Piazza for communication and discussion.

Technological Proficiency and Hardware/Software Required

Students will utilize personal laptops/desktops and should be familiar with using the ACM Digital Library and web search engines to access required research papers and materials.

Required Readings and Supplementary Materials

There will be no required textbook. The lectures given by the instructor and the students' presentations will be based on recent influential papers in top research venues of data management, processing, and analysis.

Description of Course Deliverables and Assessment

- Review Reports: Students are expected to deliver review reports for some topics in the class. Each review report should contain summaries and critical evaluations (detailed strengths and weakness points) for the papers on the corresponding topics.
- Review Presentation: Each student will be part of a group presentation on a paper selected by the group on one topic covered in the course. Presentations will be followed with in-class discussions. Students will be graded based on the clarity and quality of their presentations. Typically, presentations will occur in the second half of each class.
- Two Exams: There will be two exams, at the middle and the end of the semester, to evaluate the student's understanding of the topics in the class.
- Two Projects: Students will work on two projects during the course. In the first project, all students will be asked to extend a specific open-source database system with a small feature. The main objective of this project is to let students experience developing inside database systems, especially if this is their first time doing that. The second project will be a large-scale one, where students will select an open-source database system from their choice (not necessarily the same system used in

the first project) and propose extensions to this system to support more functionalities or specific workloads. In this project, students will write a project proposal, present their findings in a presentation at the end of the course, and write a conference paper-style project report. There will be one status update meeting along the way with either the instructor or the teaching assistant(s) to check the second project's progress and provide feedback. In both projects, students will work in groups of 2-3.

Grading Breakdown

Assessment Tool (assignments)	**% of Grade**
Review Reports	20
Review Presentation	15
Exam 1	15
Exam 2	15
Project 1	10
Project 2	25
TOTAL	100

Grading Scale

- A: 94 – 101
- A-: 90 – 93
- B+: 87 – 89
- B: 83 – 86
- B-: 80 – 82
- C+: 77 – 79
- C: 73 – 76
- C-: 70 – 72
- D+: 67 – 69
- D: 63 – 66
- D-: 60 – 62
- F: < 60

Submission Policy

All deadlines for review presentations and course projects are firm. Late submissions are allowed for review reports only. Each late day will reduce the maximum credit a student can get for a report by 20%, with a maximum of 3 days late per report.

Attendance

In-person lecture attendance is required unless a special permission of absence is granted by the instructor.

Academic Integrity

This course will follow the expectations for academic integrity as stated in the USC Student Handbook. The general USC guidelines on Academic Integrity and Course Content Distribution are provided in the subsequent "Statement on Academic Conduct and Support Systems" section.

Please ask the instructor or TA if you are unsure about what constitutes unauthorized assistance on an exam or assignment or what information requires citation and/or attribution.

You may not record this class without the express permission of the instructor and all other students in the class. Distribution of any notes, recordings, exams, or other materials from a university class or lectures — other than for individual or class group study — is prohibited without the express permission of the instructor.

Use of Generative AI in this Course

Generative AI is not permitted: Since creating, analytical, and critical thinking skills are part of the learning outcomes of this course, all assignments should be prepared by the student working individually or in groups as described on each assignment. Students may not have another person or entity complete any portion of the assignment. Developing strong competencies in these areas will prepare you for a competitive workplace. Therefore, using AI-generated tools is prohibited in this course, will be identified as plagiarism, and will be reported to the Office of Academic Integrity.

Course Evaluations

Course evaluation occurs at the end of the semester university-wide. It is an important review of students' experience in the class. The process and intent of the end-of-semester evaluation should be provided. In addition, a mid-semester evaluation is recommended practice for early course correction.

Course Schedule

	Lecture	Topics/Daily Activities	Readings/Preparation	Deliverables
Week 1	01/14			
	01/16	Introduction & an Overview of Relational Database Systems	Relational Database Systems	
Week 2	01/21	Data Storage Layouts	Data Storage Layouts	
	01/23	OLAP Indexing	Indexing	
Week 3	01/28	OLTP Indexing	Indexing	
	01/30	Query Optimization Implementation – Part 1	Query Optimization	Report 1 Due
Week 4	02/04	Query Optimization Implementation – Part 2	Query Optimization	
	02/06	Query Pipelining & Materialization	Query Pipelining & Materialization	
Week 5	02/11	Query Scheduling	Query Scheduling	Project 1 Due
	02/13	Join Algorithms – Hashing	Join Algorithms	Report 2 Due
Week 6	02/18	Join Algorithms – Sorting	Join Algorithms	
	02/20	Exam 1		
Week 7	02/25	Vectorized Execution	Vectorized Execution	
	02/27	Query Compilation	Query Compilation	
Week 8	03/04	Concurrency Control – Part 1	Concurrency Control	Report 3 Due
	03/06	Concurrency Control – Part 2	Concurrency Control	
Week 9	03/11	Consistency – Part 1	Consistency	
	03/13	Consistency – Part 2	Consistency	
Week 10	03/18	**Spring Recess**		
	03/20	**Spring Recess**		
Week 11	03/25	Modern Database Systems Architectures – Part 1	System Architectures	Report 4 Due
	03/27	Modern Database Systems Architectures – Part 2	System Architectures	
Week 12	04/01	Status updates of Project 2		
	04/03	Modern Database Systems Architectures – Part 3	System Architectures	
Week 13	04/08	System Case Study – Spark	Spark	Report 5 Due
	04/10	System Case Study – Snowflake	Snowflake	
Week 14	04/15	System Case Study – DuckDB	DuckDB	
	04/17			Project 2 Due
Week 15	04/22	Exam 2		
	04/24	Project 2 Presentations (Part 1)		
Week 16	04/29	Project 2 Presentations (Part 2)		

Course Reading Lists

This is a tentative list of the papers that will be discussed on each topic. We will adjust the list when new influential papers appear.

Relational Database Systems

- M. Stonebraker, et al., Readings in Database Systems, 5th Edition, 2015 (Chapters 1 and 2)

Data Storage Layouts

- M. Stonebraker, et al., C-Store: A Column-Oriented DBMS, in VLDB 2005
- D. Abadi, et al., Column-Stores vs. Row-Stores: How Different Are They Really?, in SIGMOD 2008

Indexing

- B. Hentschel, et al., Column Sketches: A Scan Accelerator for Rapid and Robust Predicate Evaluation, in SIGMOD 2018
- Z. Wang, et al., Building A Bw-Tree Takes More Than Just Buzz Words, in SIGMOD 2018
- V. Alvarez, et al., A Comparison of Adaptive Radix Trees and Hash Tables, in ICDE 2015
- V. Leis, et al., The Adaptive Radix Tree: ARTful Indexing for Main-Memory Databases, in ICDE 2013
- J. Levandoski, et al., The Bw-Tree: A B-tree for New Hardware, in ICDE 2013
- L. Sidirourgos, et al., Column Imprints: A Secondary Index Structure, in SIGMOD 2013
- Y. Li, et al., BitWeaving: Fast Scans for Main Memory Data Processing, in SIGMOD 2013

Query Pipelining and Materialization

- P. Boncz, et al., MonetDB/X100: Hyper-Pipelining Query Execution, in CIDR 2005
- L. Shrinivas, et al., Materialization Strategies in the Vertica Analytic Database: Lessons Learned, in ICDE 2013
- M. Kester, et al., Access Path Selection in Main-Memory Optimized Data Systems: Should I Scan or Should I Probe?, in SIGMOD 2017

Query Scheduling

- V. Leis, et al., Morsel-Driven Parallelism: A NUMA-Aware Query Evaluation Framework for the Many-Core Age, in SIGMOD 2014
- I. Psaroudakis, et al., Scaling Up Concurrent Main-Memory Column-Store Scans: Towards Adaptive NUMA-aware Data and Task Placement, in VLDB 2015
- B. Wagner, et al., Self-Tuning Query Scheduling for Analytical Workloads, in SIGMOD 2021
- I. Sabek, et a., LSched: A Workload-Aware Learned Query Scheduler for Analytical Database Systems, in SIGMOD 2022

Vectorized Execution

- P. Menon, et al., Relaxed Operator Fusion for In-Memory Databases: Making Compilation, Vectorization, and Prefetching Work Together At Last, in VLDB 2017
- O. Polychroniou, et al., Rethinking SIMD Vectorization for In-Memory Databases, in SIGMOD 2015
- H. Lang, et al., Make the Most out of Your SIMD Investments: Counter Control Flow Divergence in Compiled Query Pipelines, in VLDB Journal 2020

Query Compilation

- T. Neumann, Efficiently Compiling Efficient Query Plans for Modern Hardware, in VLDB 2011
- A. Shaikhha, et al., How to Architect a Query Compiler, in SIGMOD 2016
- A. Kohn, et al., Adaptive Execution of Compiled Queries, in ICDE 2018
- K. Krikellas, et al., Generating Code for Holistic Query Evaluation, in ICDE 2010
- T. Kersten, et al., Everything You Always Wanted to Know About Compiled and Vectorized Queries But Were Afraid to Ask, in VLDB 2018
- H. Lang, et al., Data Blocks: Hybrid OLTP and OLAP on Compressed Storage using both Vectorization and Compilation, in SIGMOD 2016

Join Algorithms

- C. Balkesen, et al., Main-Memory Hash Joins on Multi-Core CPUs: Tuning to the Underlying Hardware, in ICDE 2013
- S. Blanas, et al., Design and Evaluation of Main Memory Hash Join Algorithms for Multi-core CPUs, in SIGMOD 2011
- S. Richter, et al., A Seven-Dimensional Analysis of Hashing Methods and its Implications on Query Processing, in VLDB 2015
- S. Schuh, et al., An Experimental Comparison of Thirteen Relational Equi-Joins in Main Memory, in SIGMOD 2016
- M. Bandle, et al., To Partition, or Not to Partition, That is the Join Question in a Real System, in SIGMOD 2021
- C. Balkesen, et al., Multi-Core, Main-Memory Joins: Sort vs. Hash Revisited, in VLDB 2013
- M.-C. Albutiu, et al., Massively Parallel Sort-Merge Joins in Main Memory Multi-Core Database Systems, in VLDB 2012
- I. Sabek, et a., The Case for Learned In-Memory Joins, in VLDB 2023

Query Optimization

- G. Graefe, et al., The Volcano Optimizer Generator: Extensibility and Efficient Search, in ICDE 1993
- G. Graefe, The Cascades Framework for Query Optimization, in IEEE Data Engineering Bulletin 1995
- S. Chaudhuri, An Overview of Query Optimization in Relational Systems, in PODS 1998
- E. Begoli, et al., Apache Calcite: A Foundational Framework for Optimized Query Processing Over Heterogeneous Data Sources, in SIGMOD 2018
- J. Chen, et al., The MemSQL Query Optimizer, in VLDB 2017
- M. Soliman, et al., Orca: A Modular Query Optimizer Architecture for Big Data, in SIGMOD 2014
- L.D. Shapiro, et al., Exploiting Upper and Lower Bounds In Top-Down Query Optimization, in IDEAS 2001

Concurrency Control

- Y. Wu, et al., An Empirical Evaluation of In-Memory Multi-Version Concurrency Control, in VLDB 2017
- T. Neumann, et al., Fast Serializable Multi-Version Concurrency Control for Main-Memory Database Systems, in SIGMOD 2015
- J. Böttcher, et al., Scalable Garbage Collection for In-Memory MVCC Systems, in VLDB 2019
- S. Tu, et al., Speedy Transactions in Multicore In-Memory Databases, in SOSP 2013
- X. Yu, et al., TicToc: Time-Traveling Optimistic Concurrency Control, in SIGMOD 2016

- X. Yu, et al., Staring into the Abyss: An Evaluation of Concurrency Control with One Thousand Cores, in VLDB 2014

Consistency

- S. Ghandeharizadeh, J. Yap, et al., Strong consistency in cache augmented SQL systems, in Middleware 2014.
- DeCandia et al., Dynamo: Amazon's Highly Available Key-value Store, in SOSP 2007
- W Vogels, Eventually Consistent - Revisited. All Things Distributed (Blog), 2008 [URL: https://www.allthingsdistributed.com/2008/12/eventually_consistent.html]

Database System Architectures

- J. Arulraj, et al., Write-Behind Logging, in VLDB 2016
- H. Kimura, FOEDUS: OLTP Engine for a Thousand Cores and NVRAM, in SIGMOD 2015
- J. Power, et al., Toward GPUs Being Mainstream in Analytic Processing, in DaMoN 2015
- J. Arulraj, et al., Let's Talk About Storage & Recovery Methods for Non-Volatile Memory Database Systems, in SIGMOD 2015
- I. Oukid, et al., SOFORT: A Hybrid SCM-DRAM Storage Engine for Fast Data Recovery, in *DaMoN* 2014
- S. Harizopoulos, et al., OLTP Through the Looking Glass, and What We Found There, in SIGMOD 2008
- V. Leis, et al., LeanStore: In-Memory Data Management beyond Main Memory, in ICDE 2018
- J. DeBrabant, et al., Anti-Caching: A New Approach to Database Management System Architecture, in VLDB 2013
- L. Ma, et al., Larger-than-memory Data Management on Modern Storage Hardware for In-memory OLTP Database Systems, in DaMoN 2016
- Q. Cai, et al., Efficient Distributed Memory Management with RDMA and Caching, in VLDB 2018
- J. Yang, et. al., F1 Lightning: HTAP as a Service, in VLDB 2020
- A. J. Elmore, et al., Zephyr: Live Migration in Shared Nothing Databases for Elastic Cloud Platforms, In SIGMOD 2011

DuckDB

- M. Raasveldt, et al., DuckDB: an Embeddable Analytical Database, in SIGMOD 2019
- M. Raasveldt, et al., Data Management for Data Science Towards Embedded Analytics, in CIDR 2020

Snowflake

- B. Dageville, et al., The Snowflake Elastic Data Warehouse, in SIGMOD 2016

Spark

- A. Behm, et al., Photon: A Fast Query Engine for Lakehouse Systems, in SIGMOD 2022
- P. Jain, et al., Analyzing and Comparing Lakehouse Storage Systems, in CIDR 2023
- M. Armbrust, et al., Delta Lake: High-Performance ACID Table Storage over Cloud Object Stores, in VLDB 2020

Materialize

- F. McSherry, et al., Materialize: A Platform for Building Scalable Event-based Systems, in DEBS 2022

Statement on Academic Conduct and Support Systems

Academic Integrity:
The University of Southern California is a learning community committed to developing successful scholars and researchers dedicated to the pursuit of knowledge and the dissemination of ideas. Academic misconduct, which includes any act of dishonesty in the production or submission of academic work, comprises the integrity of the person who commits the act and can impugn the perceived integrity of the entire university community. It stands in opposition to the university's mission to research, educate, and contribute productively to our community and the world.

All students are expected to submit assignments that represent their own original work, and that have been prepared specifically for the course or section for which they have been submitted. You may not submit work written by others or "recycle" work prepared for other courses without obtaining written permission from the instructor(s).

Other violations of academic integrity include, but are not limited to, cheating, plagiarism, fabrication (e.g., falsifying data), collusion, knowingly assisting others in acts of academic dishonesty, and any act that gains or is intended to gain an unfair academic advantage.

The impact of academic dishonesty is far-reaching and is considered a serious offense against the university. All incidences of academic misconduct will be reported to the Office of Academic Integrity and could result in outcomes such as failure on the assignment, failure in the course, suspension, or even expulsion from the university.

For more information about academic integrity see the student handbook or the Office of Academic Integrity's website, and university policies on Research and Scholarship Misconduct.

Please ask your instructor if you are unsure what constitutes unauthorized assistance on an exam or assignment, or what information requires citation and/or attribution.

Course Content Distribution and Synchronous Session Recordings Policies
USC has policies that prohibit recording and distribution of any synchronous and asynchronous course content outside of the learning environment.

Recording a university class without the express permission of the instructor and announcement to the class, or unless conducted pursuant to an Office of Student Accessibility Services (OSAS) accommodation. Recording can inhibit free discussion in the future, and thus infringe on the academic freedom of other students as well as the instructor. (Living our Unifying Values: The USC Student Handbook, page 13).

Distribution or use of notes, recordings, exams, or other intellectual property, based on university classes or lectures without the express permission of the instructor for purposes other than individual or group study. This includes but is not limited to providing materials for distribution by services publishing course materials. This restriction on unauthorized use also applies to all information, which had been distributed to students or in any way had been displayed for use in relationship to the class, whether obtained in class, via email, on

the internet, or via any other media. (Living our Unifying Values: The USC Student Handbook, page 13).

Students and Disability Accommodations:
USC welcomes students with disabilities into all of the University's educational programs. The Office of Student Accessibility Services (OSAS) is responsible for the determination of appropriate accommodations for students who encounter disability-related barriers. Once a student has completed the OSAS process (registration, initial appointment, and submitted documentation) and accommodations are determined to be reasonable and appropriate, a Letter of Accommodation (LOA) will be available to generate for each course. The LOA must be given to each course instructor by the student and followed up with a discussion. This should be done as early in the semester as possible as accommodations are not retroactive. More information can be found at osas.usc.edu. You may contact OSAS at (213) 740-0776 or via email at osasfrontdesk@usc.edu.

Support Systems:

Counseling and Mental Health - (213) 740-9355 – 24/7 on call
Free and confidential mental health treatment for students, including short-term psychotherapy, group counseling, stress fitness workshops, and crisis intervention.

988 Suicide and Crisis Lifeline - 988 for both calls and text messages – 24/7 on call
The 988 Suicide and Crisis Lifeline (formerly known as the National Suicide Prevention Lifeline) provides free and confidential emotional support to people in suicidal crisis or emotional distress 24 hours a day, 7 days a week, across the United States. The Lifeline is comprised of a national network of over 200 local crisis centers, combining custom local care and resources with national standards and best practices. The new, shorter phone number makes it easier for people to remember and access mental health crisis services (though the previous 1 (800) 273-8255 number will continue to function indefinitely) and represents a continued commitment to those in crisis.

Relationship and Sexual Violence Prevention Services (RSVP) - (213) 740-9355(WELL) – 24/7 on call
Free and confidential therapy services, workshops, and training for situations related to gender- and power-based harm (including sexual assault, intimate partner violence, and stalking).

Office for Equity, Equal Opportunity, and Title IX (EEO-TIX) - (213) 740-5086
Information about how to get help or help someone affected by harassment or discrimination, rights of protected classes, reporting options, and additional resources for students, faculty, staff, visitors, and applicants.

Reporting Incidents of Bias or Harassment - (213) 740-5086 or (213) 821-8298
Avenue to report incidents of bias, hate crimes, and microaggressions to the Office for Equity, Equal Opportunity, and Title for appropriate investigation, supportive measures, and response.

The Office of Student Accessibility Services (OSAS) - (213) 740-0776

OSAS ensures equal access for students with disabilities through providing academic accommodations and auxiliary aids in accordance with federal laws and university policy.

USC Campus Support and Intervention - *(213) 740-0411*
Assists students and families in resolving complex personal, financial, and academic issues adversely affecting their success as a student.

Diversity, Equity and Inclusion - *(213) 740-2101*
Information on events, programs and training, the Provost's Diversity and Inclusion Council, Diversity Liaisons for each academic school, chronology, participation, and various resources for students.

USC Emergency - *UPC: (213) 740-4321, HSC: (323) 442-1000 – 24/7 on call*
Emergency assistance and avenue to report a crime. Latest updates regarding safety, including ways in which instruction will be continued if an officially declared emergency makes travel to campus infeasible.

USC Department of Public Safety - *UPC: (213) 740-6000, HSC: (323) 442-1200 – 24/7 on call*
Non-emergency assistance or information.

Office of the Ombuds - *(213) 821-9556 (UPC) / (323-442-0382 (HSC)*
A safe and confidential place to share your USC-related issues with a University Ombuds who will work with you to explore options or paths to manage your concern.

Occupational Therapy Faculty Practice - *(323) 442-2850 or* otfp@med.usc.edu
Confidential Lifestyle Redesign services for USC students to support health promoting habits and routines that enhance quality of life and academic performance.