CSE177/EECS277 – DATABASE SYSTEMS IMPLEMENTATION

- Instructor: Florin Rusu
- Office: SE2-210
- Email: frusu@ucmerced.edu
- Web: http://faculty.ucmerced.edu/frusu

- TA: Yujing Ma (yma33@ucmerced.edu)

Meeting time.

- Lecture: M,W 9:00-10:15AM, KOLLIG 217
- Lab: W 7:30-10:20PM, SCIENG 138; R 4:30-7:20PM, SCIENG 138
- Exam: Tuesday, May 14 11:30-14:30, KOLLIG 217 or in the lab SCIENG 138

- Instructor office hours: F 3-5PM; SE2-210. Or by appointment.
- TA office hours: T 2-4PM; SE2-313

Description. This course studies the internals of a database management system, with emphasis on query execution. The final goal of the course is to build a fully-functional database execution engine consisting of all the standard components: storage manager, buffer manager, query execution engine, query optimizer, and query compiler.

Prerequisites.

- CSE 100 – Algorithm Design and Analysis
- CSE 111 – Database Systems

Textbook.

- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom – *Database Systems, The Complete Book*, 2nd Edition, 2009, Prentice Hall, ISBN: 978-0-13-815504-9.

References.

- Avi Silberschatz, Henry F. Korth, and S. Sudarshan – *Database System Concepts*, 6th Edition, 2010, McGraw-Hill, ISBN: 0-07-352332-1.
- Raghu Ramakrishnan, Johannes Gehrke – *Database Management Systems*, 3rd Edition, 2003, McGraw-Hill, ISBN: 0-07-246563-8.
- Ramez Elmasri, Shamkant B. Navathe – *Fundamentals of Database Systems*, 6th Edition, 2011, Addison-Wesley, ISBN: 978-0-136-08620-8.

Course format. The course consists of 3 hour lectures and 3 hour labs per week. The fundamentals of query processing algorithms are presented during the lectures. The understanding of the students is checked through weekly quizzes. The lab is entirely dedicated to the semester-long project which requires students to implement a fully-functional database system based on the concepts presented during the lectures.

Topics.

1. The Query Compiler (Chapter 16)
   - Parsing and Preprocessing
   - Algebraic Laws for Improving Query Plans
   - From Parse Trees to Logical Query Plans
   - Estimating the Cost of Operations
   - Introduction to Cost-Based Plan Selection
   - Choosing an Order for Joins
   - Completing the Physical-Query-Plan
2. Secondary Storage Management (Chapter 13)
   - The Memory Hierarchy
   - Disks
   - Accelerating Access to Secondary Storage
   - Disk Failures
   - Arranging Data on Disk
   - Representing Block and Record Addresses
   - Variable-Length Data and Records
   - Record Modifications
3. Query Execution (Chapter 15)
   - Introduction to Physical-Query-Plan Operators
   - One-Pass Algorithms
   - Nested-Loop Joins
   - Two-Pass Algorithms Based on Sorting
   - Two-Pass Algorithms Based on Hashing
   - Index-Based Algorithms
   - Buffer Management
   - Algorithms Using More Than Two Passes
4. Index Structures (Chapter 14)
   - Index-Structure Basics
   - B-Trees
   - Hash Tables
   - Multidimensional Indexes
   - Hash Structures for Multidimensional Data
   - Tree Structures for Multidimensional Data
   - Bitmap Indexes
5. Coping with System Failures (Chapter 17)
   - Issues and Models for Resilient Operation
   - UNDO Logging
   - REDO Logging
   - UNDO/REDO Logging
   - Protecting Against Media Failures

Learning goals. The goal of this course is to expose students to the internals of a database execution engine. At the end of the course, students will be able to understand how a query is executed from the moment a user launches it to getting back the result. This is achieved by a thorough analytical study of the components and algorithms involved in query processing and a hands-on project that requires the implementation of a full-fledged relational database engine.

Learning outcomes. Students will learn the fundamental data processing algorithms implemented inside all the existing relational database engines. They will get detailed exposure to the current research in data management and direct experience with research prototype systems. The progress students make in assimilating the class material will be continuously tested through weekly quizzes and a semester-long project in which the students are required to implement a full-fledged relational database system. In summary, the students enrolled in this course will get exposure to the current research in data management and will experience the latest innovations from the industry. These will benefit both students more interested in research aspects as well as students looking for a more hands-on experience.

Academic integrity policy. Each student in this course is expected to abide by the University of California, Merced's Academic Honesty Policy. Any work submitted by a student in this course for academic credit will be the student's own work. Students are encouraged to study together and to discuss information and concepts covered in lectures. Students can provide/receive "consulting" to/from other students. However, the permissible cooperation should never involve one student having possession of a copy of all or part of the work done by someone else, in the form of an email, an email attachment file, a storage device, or a hard copy. Should copying occur, both the student who copied work from another student and the student who gave material to be copied will receive zero credit for the corresponding assignment. Penalty for violation of this Policy can also be extended to include failure of the course and University disciplinary action. During examinations, each student has to do only their own work. Talking or discussing is not permitted, nor students comparing their papers, copying from others, or collaborating in any way. Any collaborative behavior during examinations will result in failure of the exam and may lead to failure of the course and University disciplinary action.

Disability service information. Accommodations for students with disabilities: The University of California, Merced is committed to ensuring equal academic opportunities and inclusion for students with disabilities based on the principles of independent living, accessible universal design diversity. I am available to discuss appropriate academic accommodations that may be required for students with disabilities. Requests for academic accommodations are to be made during the first three weeks of the semester, except for unusual circumstances. Students are encouraged to register with the Disability Services Center to verify their eligibility for appropriate accommodations.

Grading.

- **Project 80%** The project is divided into 6 stages that build on top of each other. The stages are spread over the entire semester. Each stage counts for 10% of the final grade, while the complete project presentation and demo count for the remaining score. The demo takes place during the final exam. In addition to these 6 stages, students enrolled in EECS277 are required to implement an additional stage that can be completely isolated from the main project. Alternatively, students from EECS277 may be part of smaller teams. The implementation of the project is mandatory in `C/C++`. We provide code that implements many of the low-level functionality and give clear requirements on what code to develop. Since the grade is largely based on the project, students should expect to spend a significant amount of time coding. This might go well beyond the allocated lab time.

- **Quizzes 20%** Weekly quizzes based on the material discussed during the lecture are meant to test the understanding of the students. The concepts have to be put in practice in the project.

- ≥ 950: A+; ≥ 900: A; ≥ 800: A-; ≥ 770: B+; ≥ 730: B; ≥ 700: B-; ≥ 670: C+; ≥ 630: C; ≥ 600: C-; ≥ 500: D; < 500: F.
- Curved grading may apply only in special situations.