CS 411 - Database Systems

Instructors:

CS 411 has been taught by various instructors in the CS department, and none of them have taught more than one semester consecutively. Recently it has been taught by Peixiang Zhao, Anna Yershova, Saurabh Sinha, Ryan Cunningham, and Kevin Chang. It is worth noting that person teaching the course isn't always a tenured professor, and can sometimes be a PhD candidate.

Prerequisites:

CS 225 Data Structures is the only listed prerequisite for the course, but the overlap is limited. Specifically, there is a brief discussion of B-Trees in CS 225 that becomes relevant in CS 411. However, most, if not all, of the material is reviewed in lecture.

More important than the intuition for data structures for this course is some practical programming experience in a web stack, similar to what many Computer Science majors get in CS 242 Programming Studio. There is a final project in which you create a database driven website, and there is no special treatment given to those who don't know how to go about making that happen. Though it's possible to pick it up, going in with some experience with PHP, Django, or some other web framework that can connect to a SQL database will make the final project far less painful. Alternatively, pick a partner who has some experience with a web framework.

Additionally, in recent semesters, MPs have been added to the course. While some of these are MySQL and relatively straight forward, others have been in Python, and were not as trivial to understand. Some experience with Python syntax, structure, and paradigms will help make the MPs more about Database Systems and less about learning Python. However, it is unclear if these MPs will remain a part of the course in the coming semesters.

When to Take It:

You can take this course whenever you want once you're ready for it. This course is not a major time commitment compared to most 400 level ECE & CS courses, so it's good to take if you need to squeeze a tech elective into a busy semester. Again, it's helpful to be comfortable with practical web programming before enrolling, but any specific skills required (JavaScript for client side anything, CSS/HTML, etc.) can be picked up along the way.

This course is typically offered every semester.

Class Content:

The course first covers relational models and algebra, and then the principles behind quality database schema design. The second part of the course then applies these concepts to SQL (Structured Query Language). Finally, the practical considerations of implementing a Relational Database Management System, including the considerations of hard disk design, query optimization, and database indexing. If there is time at the end of the course, sometimes the instructor will go into advanced topics, like MinHashing, the algorithm behind recommendation systems.

Work:

Each section of the course has an associated homework assignment (and sometimes an MP). Two weeks are given for the homeworks, and one week for the MPs. In terms of actual work, the amount of time required varies wildly based on the amount of experience you come into the course with, but if the material is well understood post-lecture, homeworks shouldn't take more than 5 hours total, and the MPs shouldn't take more than 5 to 8 hours.

The course typically has one midterm and one final, which share a similar format of having both multiple choice questions, short answer, and some other written problems. Typically the kinds of problems closely mimic those that have been on the semester's homework. Because there is high instructor turnover, current instructors also often look to previous midterms and finals for problem content.

Finally, there is a semester long project where teams of 2 to 4 create a database driven website. The expectations for quality are set fairly high. Apart from minimal functionality, grading TAs also look for a certain amount of polish. This requires some real effort, especially if nobody on the team has web programming experience.

Typically, there is also a requirement for advanced functions in the project, which is where team members without web programming experience can shine. These advanced features are sometimes a native mobile app, but can also be things like applied digital signal processing, machine learning, or other algorithmic data manipulation which doesn't strictly require knowledge of web implementation.

Life After:

Data driven websites are everywhere. Having experience with SQL, relational databases, and the practical web programming done in the semester long project opens up a lot of practical opportunities in industry. The project can be extremely beneficial for learning other practical topics, like data scraping and UI design as well.
Though this course focuses primarily on relational databases, it briefly covers NoSQL, and many real world applications use NoSQL databases. C S 425 (ECE 428) Distributed Systems covers some of the considerations when implementing a NoSQL database. On occasion, there is a CS 498 Special Topics course on Web Programming that also covers NoSQL topics.