Advanced Databases (CPT-S 580-06, 2016 Spring)

Instructor: Yinghui Wu
Office: EME 49
Email: yinghui@eecs.wsu.edu
Web: http://eecs.wsu.edu/~yinghui/

Description: This course is an advanced study of database theory and systems, and advanced research topics in connection with large-scale data management and information extraction techniques. The course will start by overviewing fundamentals of data management, including database theory, complexity theory, and necessary background of discrete mathematics (set and graph theory). We will introduce modern (e.g., noSQL, newSQL, in-memory) database models, query languages, and systems adopted in industry and science applications. We will study the internals of database management systems (indexing, query optimization, data consistency), and application-driven databases including distributed/parallel databases, in-memory databases, stream processing, and graph databases. The course will also overview new and critical challenges in big data management (data quality, data privacy, crowdsourcing, OLTP/OLAP, data warehousing), with applications in in social analytics, cyber security, and information networks, among others that are already in public eye.

Course Objectives: This course will help students to achieve the following objectives.

- Understand theoretical foundations and principles of database design
- Learn new ways to query and model data.
- Get familiar with the expanding role of database technology in big data era.
- Understand and practice the research skills (algorithm design, problem solving, paper review and presentation)

Course Format: This is a seminar course. Lectures are to provide background as needed. Students will be expected to read related research papers or textbook on a provided list. Each student will also be expected to complete and present either a survey or a course project independently.

In keeping the research seminar nature of the course, there will be no exams. Instead, students are required to read research papers, write reviews, complete a course project and present the project in class. Final grades will be determined as follows:

- Reviews: 40%
- Project: 45%
- Project report and presentation: 15%

You should read as many papers on a provided list as possible. You should also select 8 papers from the list, and write reviews for those papers. Each review should be about one-page long, and should consist of a compelling mix of summary, key ideas and contributions, motivation for studying the problems, criteria for the line of research, evaluation, and possible extensions.

Project: Projects will be developed during the class. A project requires (a) design and development of an algorithm that deals in more depth with a topic encountered during the semester, **or** (b) a comprehensive survey of a line of research. You are expected to complete a project *independently*, and write and present a final project report.

- **Algorithm design project**: You may take any topic from a given list, and are encouraged to come up with your own. The same project may be taken by multiple people, but must be done

separately. Develop or implement an algorithm for one of the following problems below. Each problem accounts for a project.

- FunFacts: implement a fast keyword search in a fraction of Google knowledge graph (Freebase), to answer queries such as: "what are interesting/fun fact of an entity?"
- FastReach: implement an index for scalable reachability/distance queries over large graph/social network.
- Make "Big data" small: Develop effective data compression, sampling, reduction and selection techniques to support a query class (e.g., SQL, SPARQK, top-k skyline queries, reachability queries) in large-scale databases.
- Rule-based Data cleaning: implement a data-cleaning approach to fix inconsistencies in "dirty" data in knowledge bases.
- Design an algorithm to discover association rules in network data (e.g., Amazon co-purchasing recommendation network)

- **Survey Topics:** If you select survey as your course project, pick any of the topics from a list below, and write a comprehensive survey on the topic.
  - State-of-the-art SQL, noSQL and newSQL databases
  - Distributed search engines with applications
  - Algorithms for data cleaning in knowledge bases.
  - Distributed data stores and data models
  - Distributed algorithms for temporal graph and graph streaming.
  - Q&A systems over Web data and knowledge bases
  - In situ data processing
  - In-memory databases
  - Big data applications in Health Data (or a specific application domain: Power system, sports data, Internet of Things, Scientific data, Cloud computing).
  - Crowdsourcing + database systems
  - Big data benchmarks

You are encouraged to come up with your own project topic. Please discuss with the instructor.

Prerequisites: Students are expected to have basic programming experiences and knowledge of algorithm design, equivalent to a data structure course CPTS 223. Some background in basic linear algebra is necessary. The course will cover basics in graph theory and relational databases.