CSC 560: Special Topics in Databases - Modern DBMS Architectures
Fall 2014
Course Syllabus

September 11, 2014

Instructor: Alexander Dekhtyar
email: dekhtyar@csc.calpoly.edu
office: 14-215

What	When		Where
Lecture	TR	9:10 – 11:00pm	20-143 (Engineering East)
Final Exam	December 11 (Thursday)	10:10 - 1:00pm	20-143

Office Hours

	When	Where
Monday	9:30 - 11:30pm	14-215
Tuesday	11:10am - 12:00pm	14-215
Wednsday	10:10am - 12:pm	14-215

Description

The past decade has given us more new DBMS architectures than the previous thirty years combined. Relational DBMS are still dominant on the market, but new emerging DBMS architectures prove to be more appropriate for specific applications. In this course we will study the current DBMS architectures used actively in modern software applications: both relational, and the newly emerging, so called "NoSQL" DBMS. We will discuss their internal organization, the trade-offs posed by each architecture and the applications for which each DBMS architecture is appropriate.

Textbook, Readings

The class does not have an official textbook. The first part of the course is a rehash of selected topics from CSC 468 and is taught from the CSC 365/366/468 textbook:

> *Database Systems. The Complete Book*, 2nd Ed. H. Garcia-Molina, J.D. Ullman, J. Widom., 2009, Prentice Hall.
>
> or
>
> *Database System Implementation*, H. Garcia-Molina, J.D. Ullman, J. Widom, 2000, Prentice Hall.

The remainder of topics will be covered via a combination of academic papers, white papers, outside speakers, recently published books, and sheer luck.

Topics

The approximate list of topics is:

No.	Topic	Duration (weeks)
1.	ACID properties of DBMS and Relational DBMS	2
2.	Distributed Data Management (MapReduce)	1
3.	NoSQL DBMS: Overview	1
4.	NoSQL DBMS: Key-Value DBMS	1
5.	NoSQL DBMS: Document DBMS	1
6.	NoSQL DBMS: Column-Family DBMS	1-2
7.	NoSQL DBMS: Graph DBMS	1-2

The order of topics is not fixed and will depend on a number of circumstances.

Outside speakers

A significant portion of this course will, most likely, be outsourced to outside speakers. Some speakers will talk about specific data management applications and their approaches to success. Some speakers will present specific DBMS architectures and discuss their uses.

Grading

Team Project	50%
Literature Survey	10-20%
Assignments/Presentations	20-25%
Class Participation	5-10%

Course Policies

Exams

We have a scheduled time to meet for the final exam (Thursday, Dec 11, 10:10am - 1:00pm). There will be no written exam for the course. This time will be used either for oral presentations or demos of the quarter-long team project.

Project

The best way to understand the DBMS architecture, and to understand how non-relational DBMS work is to build one from scratch. Our project this quarter is the construction of a prototype of a non-relational DBMS *from scratch*. During the first week of classes we will form individual teams. Each team will choose a type of DBMS it wants to implement.

The project will affect how the majority of the course is taught. In a significant portion of the course, we will follow the "need-to-know"/"just-in-time knowledge" paradigm, where each team will determine what information it needs in order to succeed in building their DBMS, and then, jointly with the instructor, researches this information and presents it in class.

Team Assignments/Presentations

Unlike previous courses, where each topic was visited once during the quarter, we will investigate the various DBMS architectures (the ones that the teams are implementing) *in parallel*, with the team implementing the specific architecture responsible for on-going narrative about their DBMS. To that end, each team will prepare a series of presentations (about 20-30 mins each) and as we proceed through the course, will deliver them to the class. This way, we will wind up studying the non-relational DBMS architectures in parallel and will have ample opportunity for a compare-and-contrast throughout the quarter.

Outside of the project, we do not expect other significant programming assignments, although I reserve the right to revise it, should a short-term programming assignment present itself as being very benefitial to everyone.

In the ramp-up to the project, there may be some team assignments with written deliverables.

Literature Surveys

This is the only major compenent of the course that comes as an individual assignment. Each student will propose a database-related topic for literature survey, and, upon getting it approved, will select a number of papers on this topic, and prepare an academic literature survey. Because the course is devoted to essentially cutting-edge database research, your literature survey will have to include papers that have been published in major database conferences in the past five years (i.e., we are interested in problems that the database research community is working on now, rather than the ones it was addressing 10-20 years ago).

Please note, that the set of papers described in the literature survey shall be different from the papers each team studies as part of the project/in-class presentations preparation.

Late Submissions

There is very little room in the course for late submissions. Most written deliverables will have deadlines that are very hard to violate (e.g., last class of the quarter). Most in-person deliverables (demos, presentations) will have to happen on specific days or else.

Please don't make me come up with complicated rules for what happens to late submissions by not submitting anything late.

Web Page

Class web page can be found at

http://www.csc.calpoly.edu/~dekhtyar/560-Fall2014

Through this page you will be able to access all class handouts including homeworks, project information, reading materials and lecture notes (should the latter be written).

Mailing List

The course has a mailing list:

csc-560-01-2148@calpoly.edu

I tend to use mailing lists actively during the quarter (especially in upper-division/graduate courses), so, please, make sure, you are monitoring the email subscribed to this mailing list.

Academic Integrity

University Policies

Cal Poly's Academic Integrity policies are found at

http://www.academicprograms.calpoly.edu/academicpolicies/Cheating.htm

In particular, these policies define *cheating* as (684.1)

> "...*obtaining or attempting to obtain, or aiding another to obtain credit for work, or any improvement in evaluation of performance, by any dishonest or deceptive means. Cheating includes, but is not limited to: lying; copying from another's test or examination; discussion*

> *of answers or questions on an examination or test, unless such discussion is specifically authorized by the instructor; taking or receiving copies of an exam without the permission of the instructor; using or displaying notes, "cheat sheets," or other information devices inappropriate to the prescribed test conditions; allowing someone other than the officially enrolled student to represent same."*

Plagiarism, per University policies is defined as (684.3)

> "... the act of using the ideas or work of another person or persons as if they were one's own without giving proper credit to the source. Such an act is not plagiarism if it is ascertained that the ideas were arrived through independent reasoning or logic or where the thought or idea is common knowledge. Acknowledgement of an original author or source must be made through appropriate references; i.e., quotation marks, footnotes, or commentary."

University policies state (684.2): "Cheating requires an "F" course grade and further attendance in the course is prohibited." (appeal process is also outlined, see the web site above for details.). Plagiarism, per university policies (684.4) can be treated as a form of cheating, although a level of discretion is given to the instructor, allowing the instuctor to determine the causes of plagiarism and effect other means of remedy. It is the obligation of the instructor to inform the student that a penalty is being assessed in such cases.

Course Policies

First, all traditional warnings concerning cheating apply in this course. In particular, solicitation of help from people not involved in the course and submission of materials/code etc.. not developed by you are absolutely prohibited. Any outside materials used in preparation of homeworks, reports, project assignments must be properly documents. For example, you must properly cite all papers you refer to, all web resources used in perparation. You must also note any open source, off-the-shelf, etc. . . software or code fragments that you have incorporated in your solution. If you have questions concerning allowable use of such materials, please consult me **in advance**.

For example, if an assignment is to design and implement an XML parser, you are supposed to build one from scratch and not use any available parser code (which is plentiful). On the other hand, if you want to use an open-source library, or some code developed by one of the team members prior to the course as part of a project solution, this may qualify as allowable use, if the code is used in support of the main tasks of the project.