CS 330: Database Theory and Design (3 Credits) Fall 2010 Syllabus

Catalog Description: Students explore database concepts and design; relational database systems, models and languages; and use of database software.

Goals: You are taking this course to learn how to design and implement effective relational database systems that support decision making. Through this course, you will become skilled in designing, implementing, and using databases that support applications that help professionals understand the data that describe their enterprise.

Connection to Mission: The Mission of Lewis University is to help give everyone an opportunity to develop their God-given abilities and talents and to encourage them to use those gifts to improve the lives of others. Organizations of all sizes and varieties, from the smallest of non-profits to the largest corporate entities, provide some kind of service to the public. Likewise, all organizations generate and collect data, and they should analyze their data to understand how they can improve their organization’s operations. By acquiring the skills and understanding the ideas taught in this course, you will be able to help organizations collect, organize, and analyze data so that they may better serve the people that depend on them. In fact, you’ll even get the opportunity to participate in some real-world database design and implementation projects that will provide important help to those who need database assistance.

Instructor: Dr. Ray Klump, Associate Professor of Mathematics and Computer Science, Room AS-137-S, klumpra@lewisu.edu, 815-836-5528

Class Meeting: 12pm MWF in AS-060-S

Office Hours: MWF 10am – 12pm

Textbook: Recommended text books for this course include Learning SQL by Alan Beaulieu. and Beginning PHP and MySQL 5: From Novice to Professional, Second Edition by W. Jason Gilmore. No textbook is required, as I strive to make the class notes as informative as possible.

Prerequisites: 70-210 (Data Structures)

Objectives: In successfully completing this course, you will
1. Understand how data are stored on magnetic disks.
2. Understand how modern storage mechanisms optimize data access times.
3. Discover how files can be organized for efficient data retrieval.
4. Appreciate the advantages of databases over flat file systems.
5. Understand the structure and functions of relational database management systems (RDBMSs).
6. Model data in a way that facilitates storage in an RDBMS.
8. Design databases that provide the back end for real-world applications.
9. Learn the syntax of standard Structured Query Language (SQL).
10. Write and execute SQL queries involving multiple tables in both inclusive and exclusive joins.
11. Write SQL queries that perform aggregation functions.
12. Explore relational algebra and relational calculus, the mathematical underpinnings of SQL.
13. Communicate with databases programmatically using Java and PHP.
14. Learn what ODBC is and how to set up and use an ODBC data source.
15. Understand how table indices work, how to create them for a given table, and how to choose which fields to include in the index.
16. Learn how indices are implemented physically in the file structures that underlie each database table.
17. Understand how databases handle transactions and concurrency through locking.
18. Appreciate database security risks.
19. Develop a security policy for a database based on access needs.
20. Learn how to obfuscate passwords in database tables and applications that use them.
21. Model data using the eXtensible Markup Language (XML) and DTDs.
22. Work as a team in building database solutions for real organizations that need help organizing and querying their data.
Outcomes: You will demonstrate that you have achieved these objectives by
1. Rating the performance of a disk drive given its physical attributes.
2. Implementing a basic file indexing system in a language of your choice
3. Designing a variety of database schemas for a number of hypothetical and real applications
4. Using MySQL to implement some of the database designs you create.
5. Developing a web-based database solution to a real-world problem, complete with database designs, sql script, and PHP code.
6. Illustrate by hand how data is organized for rapid recall using a variety of schemes.
7. Write a basic PHP or Java program that interacts with a database so that you may understand how to develop database-drive we applications.

In summary, you will be asked to demonstrate your understanding of the concepts through a combination of hand-written assignments and programming assignments. A good number of the assignments will be hand-written, as this is not intended to be a programming-intensive course. However, to fully appreciate how databases are used in today’s world, you need to write code that works with them. PHP is one of the most popular choices for interacting with databases because it integrates so seamlessly with the web, so that will be the programming emphasis. We will also cover Java’s approach to interacting with databases because it is representative of how other languages communicate with databases and because it will help you write database-driven desktop applications. You will not be an expert in using these languages after this course, but you will have the tools needed to continue to develop these skills.

Assessment: Grades will be determined according to the following set of weights:
- 20% Exam 1 – in class, Wednesday, October 6
- 20% Exam 2 – in class, Wednesday, November 10
- 20% Final Exam, Monday, December 13, 1:30p – 3:30p (cumulative)
- 25% Homework – assigned approximately weekly
- 15% Quizzes – in class, approximately weekly

Exemption from Final: If your average at the end of the semester, computed from the weights listed above, is 93% or higher, you will be exempt from the final exam. Your average must meet this threshold to qualify for the exemption, as I will not round grades upward. If your average at the end of the semester between 90% and 93%, you may opt not to take the final, but you will receive an A- instead of an A. Again, I will not round up your grade in making this decision.

Grading Scale: Letter grades will be assigned according the following scale:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90-100</td>
</tr>
<tr>
<td>B+</td>
<td>87-89</td>
</tr>
<tr>
<td>B</td>
<td>83-86</td>
</tr>
<tr>
<td>B-</td>
<td>80-82</td>
</tr>
<tr>
<td>C+</td>
<td>77-79</td>
</tr>
<tr>
<td>C</td>
<td>73-76</td>
</tr>
<tr>
<td>C-</td>
<td>70-72</td>
</tr>
<tr>
<td>D+</td>
<td>67-69</td>
</tr>
<tr>
<td>D</td>
<td>63-66</td>
</tr>
<tr>
<td>D-</td>
<td>60-62</td>
</tr>
<tr>
<td>F</td>
<td>59 and below</td>
</tr>
</tbody>
</table>

Attendance and Participation: If you miss a class, it is your responsibility to make up the work you missed. If you miss a quiz or an exam, you must have a good reason, or you will not be allowed to make it up. Oversleeping is not a good reason.

Surfing the web during class is about as useful as just not showing up at all. Do not surf the web during class. I will not help you if you come to me with questions that were covered during class if I strongly suspect that you spent the time surfing.

Late Homework: Late homework will not be accepted. You will always have adequate time to complete homework assignments if you budget your time properly.

Cheating: If I find that you have copied any portion of another person’s work for a homework assignment or exam, you will receive 0 points for that work, and your grade will be reduced by one level at the end of the semester. If you are found cheating a second time, you will receive an F for the course. This policy does not mean that you can’t discuss how to solve a problem with your classmates; in fact, that sort of communication is a good thing. Stealing another person’s work, however, is wrong. I will not be lenient in enforcing this policy.
Web Support: Class notes, discussion boards, and various other support tools will be posted on the course’s Blackboard shell throughout the semester. To access Blackboard, go to http://lewisuniversity.blackboard.com. I will often send email through Blackboard, so you must make sure your email address posted on Blackboard is correct.

Advice: This isn’t an easy course. Why am I telling you this? I am simply advising you that, if you want your experience of this class to be a positive one, YOU CANNOT LET ANYTHING SLIDE. If you start to fall behind, you will find it very hard to catch up. You must keep up, or this class will not be very much fun.

What does this mean on a daily basis?

• Take good notes. Don’t rely just on the notes that I hand out. Actively listen to and document the ideas we discuss in class.
• DO NOT SURF THE WEB DURING CLASS!
• Review your notes after each and every class.
• Feel free to ask me questions when you are stuck. My job and my desire are to help you.
• Start working on your programming assignments as soon as they are assigned.

If you do these things, you’ll do very well in this class. If you don’t, you won’t.

Closing: I am confident that this can be the most valuable course you take in the computer science curriculum. Every organization for which you will work will conduct its business using databases and applications that support databases. The skills you will learn in this class will therefore serve you well as you begin your careers. The great relevance of this course should motivate you to try your best to master the concepts we discuss.