CS 541
Database Management Systems

Fall 2007
updated July 31, 2007
Note: Fall 2007 is the last semester that this course will be offered by interactive video from UTSI.

Logistics

CS 541 will meet Tuesdays & Thursdays, 2:10–3:25 p.m. Eastern Time / 1:10–2:25 Central Time, beginning Thursday August 23, at two locations:

- **UT-Knoxville**, in Studio C of the Communications Building, shown on the campus map[1] → "UT Knoxville Building Locator" → "Communications & Univ. Extension" → "Display Map". Once you find the Communications & Univ. Extension Building, go to room 61 for directions to Studio C (which is in room 43).
- **UT Space Institute**[2], in room E-113. This is where the class is transmitted from.

I've reserved the interactive video connection an additional half hour (3:25–3:55 Eastern / 2:25–2:55 Central Time) after each class for project discussions, individual questions/answers/consultation, etc.

There are cameras and microphones at both ends, so you can see and hear me, and I can see and hear you and the others in the class. That means it's not just a one-way lecture, students in Knoxville can ask questions, participate in project discussions, etc. We will use dual-video transmission, so that you'll be able to see me at the whiteboard simultaneously with a large display of my PC screen.

[1] http://www.utk.edu/maps/
[2] http://www.utsi.edu

Prerequisites

No previous background in database management is assumed, but you will need the equivalent of CS 311 Discrete Structures[3] and the ability to reason clearly and carefully about properties, relationships, and data dependencies among things that exist in the real world.

Handouts

You'll probably get a lot more out of class if you print out the handouts before class and bring them with you to refer to in class. I tend to be overly optimistic about how much we will cover in a day, so if we don't get to a handout that day, keep bringing it until we get to it. Handouts for the first day include the "Printable Syllabus" and "Project Steps" (buttons on the main course web page) and the example relations shown in Tables 1,2, and 3 of the Data Structure[4] handout.

After the first day, all handouts will be posted in the "Handouts" electronic notebook which I'll give you the password to on the first day of class. As I update each handout from last year's version, its timestamp will change from 2006 to 2007. So when you see a handout with a 2007 timestamp, that means that it's been revised for this year and is ready for you to print out and bring to class.

[3] http://www.cs.utk.edu/ mclennan/Classes/311/
[4] http://www.it.jcu.edu.au/Subjects/cp3020/1995/3/node3.html

Textbooks

In Fall 2007 we will use these two textbooks:

- Jan L. Harrington, *Relational Database Design Clearly Explained,* Second Edition, Morgan Kaufmann Publishers,[5] April 2002. ISBN: 1-55860-820-6
- Chris Fehily, *SQL: Visual QuickStart Guide, 2nd Edition*, Peachpit Press, 2005. ISBN: 0321334175

These textbooks will be supplemented extensively with additional readings available on the web, as indicated in the course schedule on the next three pages.

Topics

- **Database Architecture**

  External, Conceptual, and Internal Levels

- **The Relational Database Model**

  Domains and Attributes

  Tables and Relations

  Predicates and Queries

  Primary Keys and their Entity Integrity Rules

  Foreign Keys and their Referential Integrity Rules

- **SQL (Database Definition, Manipulation & Query Language)**

  The use of ISO/ANSI-standard SQL[6] for vendor-independence and portability.

- **Database Design**

  Functional Dependencies

  Data Normalization (1st, 2nd, 3rd, & Boyce-Codd Normal Forms)

  Choosing Primary Keys and Foreign Keys

  Specifying Integrity Constraints

  Specifying User Views

  Implementing the Design in SQL

- **Transaction Management**

  Transactions

  Concurrency

  Recovery

- **Entities, Relationships, and Objects**

  Entity-Relationship Modeling

  Object/Relational Mapping

In the course schedule shown on the next three pages, the dates given for exams and project presentations are definite and unlikely to change.

However, the dates associated with textbook readings are more approximate, telling you what you need to be reading to keep up with the class and to understand the concepts we are dealing with.

We will not by default spend class time going over the assigned reading — but we *will* go over anything that you have questions about. So you are encouraged to ask questions about anything in the reading that is not clear to you. We will devote as much class time as we need to answer all questions and to clarify whatever is unclear; then we will spend the rest of our time in class looking at the same concepts from other viewpoints and/or in greater depth.

[5] http://www.mkp.com/books_catalog/

[6] http://linux.oreillynet.com/pub/a/linux/2000/10/20/aboutSQL_1.html

Date	Event	Reading Topic	Harrington	Fehily
Aug 23		Introduction, Architecture: External, Conceptual, and Internal Levels	Chapter 1; pp. 52-54; page 138. Also see pp. 375-385	

The Relational Model of Data

Aug 23-28		Domains, Attributes, Tables, and Relations	pp. 20; 22-top 23; 73-77	pp. 21-top 24
Aug 28 - Sept 4		Predicates and Queries `http://en.wikipedia.org/wiki/First-order_predicate_calculus` (skip Peano Axioms) `http://www.it.jcu.edu.au/Subjects/cp3020/1995/3/node6.html` (In class, we will formulate queries C1-C11 in a more intuitive SQL-like syntax developed by C.J. Date. Then, later in the term, we will translate these queries into standard SQL.)		

Database Design

Sept 6	**Project Presentation I** (5 min or less, no visual aids, just say a few sentences to the class.) Identify the real-world system you'll be working on, and who is in the group that will be working on this project.			
Sept 6-11		First Normal Form; Functional Dependencies; Second Normal Form	Chapter 2 to page 17; pp. 95- top 106	pp. 33-36
Sept 11		Primary Keys and their Entity Integrity Rules; Choosing Primary Keys	pp. 77-82; 191	pp. 26-27
Sept 13		Foreign Keys and their Referential Integrity Rules; Choosing Foreign Keys	pp. 82-86; 191-194	pp. 28-32
Sept 18		Third Normal Form	pp. 106-top 108	pp. 32-33
Sept 20	**Project Presentation II:** Condition-response table and Data dictionary which define the structure and composition of your user's input & output user views (Project Steps 1 and 2: External Level). The purpose here is to capture the data attributes in your original user views **before** doing any normalization of that data.			
Sept 25		SQL (Database Definition, Manipulation & Query Language)	pp. 177-180; mid 184 -190	Chapter 3 (skip "Interval Types")
Sept 27		Specifying Integrity Constraints	pp. 194-197	pp. 331-380
Oct 2		Review		
Oct 4	**Midterm Exam**			
Oct 9		Boyce-Codd Normal Form	pp. 108-109	
Oct 11	No class (Fall Break)			

Date	Event	Reading Topic	Harrington	Fehily
Implementation				
Oct 16		Expressing Queries in SQL http://www.postgresql.org/docs/7.3/interactive/queries.html (But feel free to defer section 4.2.1.1. ''Joined Tables'' until we need it Oct 31)		Chapter 4; pp. 109-115; 134-136; 139-141; 278-top 281
Oct 18	**Project Presentation III:** The Boyce-Codd Normal Form (BCNF) tables in the database you have designed for your project. Present your final BCNF tables only. Do *not* present the 1NF, 2NF, 3NF tables that led up to it.			
Oct23		Data Manipulation in SQL		Chapter 10
Oct 25		Basic User Interfaces	http://www.php.net/, especially http://pgsql.designmagick.com/tutorial.php If you have a home page or directory on the CS web site, then you can ignore the installation sections in the tutorial above. Just put your .php pages in your directory on the CS web server, and it should handle them. For reference, the PHP manual is at http://www.php.net/manual, other documentation is at http://www.php.net/docs.php, and the section on PostgreSQL php functions is at http://www.php.net/manual/en/ref.pgsql.php	
Oct 30		Queries on Multiple Tables	bottom 112 - 119	Chapter 7; pp. 237-245; 285-286; 287-293
Nov 1		Summarizing and Grouping Data; Views	pp. 86 - top 88; 198-200	Chapter 6; Chapter 13
		The following readings on transactions, concurrency, and recovery, are links to CP3020 "Advanced Database Management" by H. Ghodosi et al., at http://www.it.jcu.edu.au/Subjects/cp3020/2003-1/CP3020.HTML		
Nov 6		Transactions http://www.it.jcu.edu.au/Subjects/cp3020/2002-1/Cns/lectureNotes/chapter19/chapter19.html (skip "Characterizing Schedules Based on Recoverability")		
Nov 8		Concurrency http://www.it.jcu.edu.au/Subjects/cp3020/2002-1/Cns/lectureNotes/chapter20/chapter20.html up to and including "Downgrading" (of Locks); then "Concurrency Control Based on Timestamp Ordering" to end of chapter.		
Nov 13	**Project Presentation IV**: Demonstration of a working PostgreSQL implementation and user interface for your project.			
Nov 15		Recovery http://www.it.jcu.edu.au/Subjects/cp3020/2002-1/Cns/lectureNotes/chapter21/chapter21.html up to and including "Recovery Techniques Based on Immediate Update"		

Date	Event	Reading Topic	Harrington	Fehily
Nov 20		Entity-Relationship Modeling	Rest of Chapter 2; pp. 94-95	
Nov 22		No class: Thanksgiving holiday		
Nov 27		Object/Relational Mapping "Mapping Objects to Relational Databases: O/R Mapping In Detail" by Scott W. Ambler. http://www.agiledata.org/essays/mappingObjects.html		
Nov 29- Dec 4		Catch-up if needed; Review		
	Final Exam: 2:45 - 4:45 pm Eastern Time, Monday December 10			

Computing Facilities

The SQL examples we develop in class and in project work for the course will be done in industry-standard SQL (Structured Query Language) which is an ISO/ANSI standard and which runs on all major commercial and open-source database management systems (DMBSs).

A well-documented[7] open-source PostgreSQL[8] relational DBMS running standard SQL on a Linux server will be available throughout the semester for hands-on course work. An SQL interface into this database management system will be accessible by all students in the course from any web browser.

I'll also be providing each project group with some server space managed by an electronic notebook[9] that will allow members of the group to post, share, and revise their drafts on various parts of the design and implementation.

Evaluation

50% exams (midterm & final); 50% project work.

There are two in-class exams, a 75-minute midterm and a 2-hour final. Exams cover both the readings listed in the course schedule and the material presented in class. The exams are open book and open note, since their purpose is not to test memorization, but rather to test your understanding of concepts and ability to apply these concepts in a problem-solving situation.

The project will be evaluated according to the following criteria. These criteria will be evaluated in the context of the level of difficulty of what was attempted. The idea here is that doing reasonably well on something difficult can be more significant than doing perfectly on something straightforward.

- Design of normalized tables
- Integrity design/enforcement
  - Domain integrity
  - Entity integrity
  - Referential integrity
- Implementation
  - Quality from technical perspective
  - Quality from user's perspective
  - How likely is it that the implementation would be successful in real-world use?
- Clarity and Understandability
- Having real users and an actual real-world system that guides and constrains your design.
  - This is not an absolute requirement, but it is an evaluation criterion, in order to fairly take into account that, when you are trying to meet the needs of an actual real-world system, most of the steps in the project become more difficult.

Grading scale (out of 100 points possible in the course):

90-100 A
80-90 B
below 80 C

There is no fixed cutoff between B and B+; those near the top of the B range will receive B+.

This scale typically results in a reasonable grade distribution (nearly all A's and B's, with C's being rare), but if it does not, then the A/B cutoff or the B/C cutoff will be adjusted downward as necessary to give a reasonable distribution. Under no circumstances will the cutoffs be moved upward; if everyone earns above 90% of the points in the course, then everyone would get an A.

[7] http://www.postgresql.org/docs/
[8] http://www.postgresql.org/
[9] http://www.epm.ornl.gov/ geist/java/applets/enote/

Project

The project is to design and implement a database for a real-world application chosen by the student. Ideally, the project should be chosen to meet a real-world need that actually exists and that you are familiar with or can find out about in the first few weeks of the term.

This project is an essential part of the course learning experience. The reason for the real-world emphasis is that some of the things you need to know to be a good database designer cannot easily be learned any other way. The simple fact is that the real world is much richer and messier than any textbook example, with exceptions and special cases conspiring either to invalidate your design or to make it hopelessly complex. It is sometimes difficult to judge how much of this richness and messiness you should try to capture in your design. Many principles of database design which seem very straightforward when applied to textbook examples suddenly seem much less clear when you try to fit them to a real-world situation.

Since designing and implementing a real-life database is a lot of work, course participants are strongly encouraged to work together in groups, each group choosing a separate project. You're welcome to work with any others in the class that you wish, regardless of who is taking the class for credit and who is not. The project consists of about a dozen well-defined steps, to be divided up among the members of the group. The ideal group size would be 3–4 people, but any size from 2 to 6 people will be acceptable.

Early in the term, I will discuss these project steps[10], which are based on several prior offerings of the course. Especially if your group is large, you'll probably want to meet with me after class early in the term to get some help dividing up the tasks among the people in your group fairly, and taking into account the different talents in the group.

If you have an idea for a real-world project and would like to interest others in working with you on it, I'll be happy to help you publicize your idea to the class if you wish. Let me know as early in the term as possible, or even before the term starts.

For Further Information

If you have questions or a project idea, feel free to contact me.

— *Bruce Whitehead*
Assoc. Prof., Computer Science
Univ. of Tenn. Space Institute[11]
Tullahoma, TN 37388-9700

bwhitehe <AT> *utsi* <DOT> *edu*
931-393-7296

I check my email and voice mail regularly during the week, but not on weekends.

[10] http://www.utsi.edu/cs/541/files/Project/steps.html

[11] http://www.utsi.edu/