# CMSI 486

## INTRODUCTION TO DATABASE SYSTEMS

*http://myweb.lmu.edu/dondi/fall2005/cmsi486*

**Fall 2005** — Pereira 202
TR 10:50am–12:05pm, 3 semester hours
Office Hours: TR 1:30–3pm, 4:30–6pm

John David N. Dionisio, PhD
e-mail: *dondi@lmu.edu*, AIM: *dondi2LMU*
Doolan 106; (310) 338-5782

## *Course Objectives*

The primary objective of this course is to introduce the essential principles that guide the design, implementation, and management of systems capable of managing large amounts of data efficiently. Topics include: database system structure, data modeling, relational databases, formal query languages, integrity and security, physical design of databases, indexing and hashing. Query processing and optimization, transaction processing, concurrency, and crash recovery are introduced.

## *Course Requirements*

CMSI 386 (Programming Languages); mastery of a high-level programming language such as Java or C++; some knowledge of computer systems and elementary logic; willingness to participate actively in class discussions.

## *Materials and Texts*

- Abraham Silberschatz, Henry F. Korth, and S. Sudarshan. *Database System Concepts*, Fifth Edition, McGraw-Hill 2006. — *a.k.a. "SKS"*
- Assorted handouts and sample code to be distributed throughout the semester.

Alternatively, much of the content in the above materials is available in various forms on the Worldwide Web; starter links are available on the class Web site. Do not hesitate to search for and find additional sources of information regarding the techniques, tools, and paradigms that we will discuss.

## *Course Work and Grading*

Graded coursework consists of accumulated homework (20%), 1 midterm (25%), 1 database project (30%), and 1 final exam (25%). Letter grades are determined as follows: ≥ 90% gets an A– or better; ≥ 80% gets a B– or better; ≥ 70% gets a C– or better. Fractions of a percent ≥ 0.5 round up to the next integral value. The instructor may curve your grade upward based on qualitative considerations such as degree of difficulty, effort, class participation, time constraints, and overall attitude throughout the course. Grades are never curved downward.

### Homework

Homework consists of questions, exercises, and programming assignments to be given throughout the semester. Homework is where you can learn from your mistakes without grading penalty: if you do the work and submit it on time, you will get full credit, regardless of correctness. What goes around comes around — the effort you put into your homework pays off in the tests and the project. The homework submission deadline is always the beginning of class on the designated due date. Any submissions after this deadline receive half credit. Occasionally, extra credit homework may be assigned. Fulfilling extra credit work is counted *on top of* the 20% allocation of homework to your final grade.

## Database Project

You will design and implement a database-driven end-user application using PostgreSQL, an open source database management system. The application should implement all four major database operations: inserts, updates, deletes, and retrievals. The project consists of these deliverables:

1. *White paper* describing/proposing what you will do: submit as a text file using CVS (see below). Due September 27, 10% of the project grade. Upon submitting your white paper, we will "process" your proposed project and should have it finalized on or before October 11.
2. *Midsemester progress report* — a session where you show me what you've done so far. We will also look at the state of your design documents, source code, and supporting files at this juncture. Currently scheduled for October 27, 10% of the project grade.
3. *Design documents* — the blueprint of your application; they represent the planning and forethought that you put into your project. At a minimum, submit a *data model* and a set of *use cases*. You may use anything to create these documents, as long as the final result is in a standardized, cross-platform format such as PDF. To be submitted via CVS (see below), 30% of the project grade.
4. *Source code and supporting files* — this should be everything a "customer" will need to compile, run, and use your project. 40% of the project grade. Deliverables #3 and #4 are to be submitted via CVS, due December 6 (your final submission == whatever is in CVS on December 6).
5. *Final presentation* — time to show off! This is a session at the end of the semester to share your labor of love with everyone in the class (and perhaps other folks in the department). Currently scheduled for December 8, 10% of the project grade.

*Sample ideas:* an e-commerce application; a personal possessions tracker; a media library manager. Virtually anything these days would benefit from a database back-end — if you need to manage information, particularly in large volumes, you probably have a credible database project.

*Use of CVS:* CVS stands for Concurrent Versions System — it is a source code management system that many development shops use for managing, storing, and updating their source code. We will use CVS to manage your project files and mark milestones in your code. Your Keck Lab accounts automatically come with CVS set up in your home directories.

*Project grading criteria:* Project deliverables #1, #2, and #5 follow the same policy as homework — what matters is that you fulfill them on time, and you will get full credit for them; half credit if they're late. Deliverables #3 and #4 are your meat and potatoes, and will be graded on the following criteria:

1. *Design (30%):* How good is the overall structure of the database and application? Is it clear, easy to understand, flexible, and easy to maintain? Is it elegant or innovative? How closely does it follow the "one change, one place" rule of thumb?
2. *Functionality (30%):* How well do the database and application work? Do they satisfy the document use cases? Are results accurate or correct? Are tasks performed in a reasonable amount of time?
3. *Naming (20%):* Are tables, fields, classes, subroutines, variables, etc., clearly and consistently named? Do their names correspond to their functions and roles?
4. *Comments (15%):* Are comments provided where appropriate? Are the comments clear and well-written? Does the project take advantage of any special support for comments provided by the project language or platform (e.g. Javadoc in Java)?
5. *CVS use (5%):* Do you commit your code at reasonable intervals? Do you provide adequate descriptions of your commits?

## Tests

The midterm is initially scheduled for October 20. The final exam is scheduled for December 13. All tests are open-*paper*-everything; no sharing. Electronic lookups may also be allowed depending on the scope or subject matter. You may neither solicit nor give help while an exam is in progress. Late and/or missed tests will be handled on a case-to-case basis; in all instances, talk to me about them.

## Attendance

I am not a stickler for attendance, but I do like having a full class. Remember that the late registration and change of program deadline is September 2. The deadline for withdrawal or credit/no-credit status is November 4.

## University Policy on Academic Honesty

Loyola Marymount University expects high standards of honesty and integrity from all members of its community. Applied to the arena of academic performance, these standards preclude all acts of cheating on assignments or examinations, plagiarism, forgery of signatures or falsification of data, unauthorized access to University computer accounts or files, and removal, mutilation, or deliberate concealment of materials belonging to the University Library.

## *Course Topics and Schedule*

This schedule may change based on the actual ebb and flow of the class; deadlines, exams, and university dates (italicized) are less likely to change than lecture topics.

| | |
|---|---|
| **September** | Working with databases — modeling, management, querying, programming |
| *September 2* | *Late registration and change of program deadline* |
| September 27 | Project prospectus due |
| **October** | Theory: the relational model, algebra, and calculus; functional dependencies, normalization; integrity constraints |
| October 11 | Projects finalized |
| October 20 | Midterm |
| *October 25* | *Undergraduate holiday; no class* |
| October 27 | Midsemester progress report |
| **November** | Implementation: security, file structure and storage, indexing, query processing and optimization, transactions |
| *November 4* | *University withdraw/credit/no-credit deadline* |
| *November 24* | *Thanksgiving; no class* |
| **December** | Concurrency, recovery, architecture, and performance; miscellaneous topics (time permitting) |
| *December 6* | *Final submission of source code/supporting files* |
| *December 8* | *Final project presentations* |
| *December 13* | *Final exam, 11:00 AM* |

You can view the class calendar on the Web at *http://ical.mac.com/dondi/LMU*. If you have an iCalendar-savvy client (i.e. Mozilla Calendar, Ximian Evolution, KOrganizer, Apple iCal, etc.), you can subscribe to the class calendar at *webcal://ical.mac.com/dondi/LMU.ics*. On-the-fly updates and adjustments to the class schedule will be reflected in this calendar.