Introduction to Databases

Syllabus

Web Page

http://www.cs.northwestern.edu/~pdinda/db

Instructor

Peter A. Dinda
1890 Maple Avenue, Room 338
847-467-7859
pdinda@cs.northwestern.edu
Office hours: Thursdays, 2-4pm or by appointment

Teaching assistants

Ananth Sundararaj
1890 Maple Avenue, Room 332
847-491-7150
ais@cs.northwestern.edu
Office hours: Mondays, 10:15am-12:15pm, Wednesdays 11:15am-12:15pm
or by appointment

Bin Lin
1890 Maple Avenue, Room 224
847-491-7159
binlin@cs.northwestern.edu
Office hours: Tuesdays, 10-12am, Wednesdays 3:30-4:30pm
or by appointment

Location and Time

1890 Maple Avenue, CS Department classroom, MWF 9-9:50am

Prerequisites

Required	CS 311 or equivalent data structures course
Highly recommended	CS 213 or equivalent computer systems course
Highly recommended	Familiarity with concepts from discrete math such as set theory
Highly recommended	Some familiarity with Perl or other scripting language

Textbook and other readings

Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer D. Widom, *Database Systems: The Complete Book*, Prentice Hall, 2001 (Textbook)

- An in-depth introduction to databases and database implementation

Phillip Greenspun, *SQL for Web Nerds*, http://philip.greenspun.com/sql/. (Required)

- A great introduction to RDBMS systems from the perspective of a web application developer.

Joe Celko, *SQL for Smarties: Advanced SQL Programming*, 2nd edition, Morgan Kaufman, 1999. (Useful)

- A collection of wisdom on how working developers get useful things done in SQL.

Jim Gray, Andreas Reuter, *Transaction Processing: Concepts and Techniques*, Morgan Kaufman, 1993. (Related)

- Definitive book on transactions, a very important component of any modern database system.

Larry Wall, Tom Christiansen, Jon Orwant, *Programming Perl*, 3rd Edition, O'Reilly and Associates, 2000. (Useful)

- A detailed introduction to the Perl language. Your web-oriented projects in this class will be based on Perl CGI. You will need to know (or learn) only limited amounts of Perl.

Objectives, framework, philosophy, and caveats

This course introduces the underlying concepts behind data modeling and database systems using relational database management systems (RDBMS), the structured query language (SQL), and web applications (Perl DBI in CGI) as examples.

You will learn:

- How to model your data using the entity-relationship model
- How to design a normalized schema in the relational data model
- How to implement your schema using SQL
- How to keep your data consistent and safe with your schema using the ACID properties that a modern RDBMS gives you
- How to query your data using SQL
- How to interface to a modern RDBMS from a modern programming language.
- How such interfaces are used to create web applications
- How an RDBMS provides quick access to your data using indices, and how indices are implemented.
- How an RDBMS manages the storage hierarchy.
- How an RDBMS optimizes and execute your queries using the relational algebra, the theoretical underpinning of database systems.

- The history of database systems, including old ideas, like hierarchical databases, that are seeing a resurgence of interest today in the context of XML and LDAP.

The textbook I have chosen is actually a combination of two books, an introduction to the concepts and use of databases and an introduction to the implementation of RDBMS systems. We will cover mostly the former. However, this is a very useful and essentially timeless book to have on your bookshelf for both elements.

This is a learn-by-doing kind of class. You will get your hands dirty by creating a database-based web application that you will propose, design, and implement yourself. The majority of the programming in this class will be from scratch.

Be warned that this is the first iteration of this course and I am not a database researcher. Also, I may adjust the pacing of the class as we go based on feedback.

Project

At the beginning of the course, I will provide you with a simple web application, a tiny web log ("blog"). Microblog is based on an Oracle database and provides a web interface using a CGI application written in Perl that talks to the database via DBI. This is a very common web application model and one that I encourage you to use for the rest of the class. You will spend three weeks learning how Microblog works and extending it in several simple ways. The goal is to immediately introduce you to all the programming elements of the course.

For the remainder of the class, you will work on a self-defined project. You will propose your own problem and spend the remainder of the quarter designing and implementing your solution.

Detailed descriptions of the requirements of both projects are available on the course web site. Please be sure to read them now so that you know what you're getting yourself into.

Homework

There will be three to four homework problems sets that will be periodically assigned to help you improve your understanding of the material.

Exams

There will be a midterm exam and a final exam. The final exam will not be cumulative.

Grading

10 % Dry-run project
40 % Self-defined project
20 % Midterm

20 % Final
10 % Homework

Final grades will be computed in the following way. A final score from 0 to 100 will be computed as a weighted sum of each of the projects, the homeworks, and the exams. Scores greater than 90 or greater than 90th percentile will be assigned As, scores greater than 80 or greater than 80th percentile will be assigned Bs, scores greater than 70 or greater than 70th percentile will be assigned Cs, scores greater than 60 or greater than 60th percentile will be assigned Ds, and the remainder will be assigned Fs. Notice that this means that if everyone works hard and gets >90, everyone gets an A. Please choose wisely where you put your time.

Late Policy

For each calendar day after the due date for a homework or a lab, 10% is lost. After 1 day, the maximum score is 90%, after 2 days, 80%, etc, for a maximum of 10 days.

Cheating

Since cheaters are mostly hurting themselves, we do not have the time or energy to hunt them down. We much prefer that you act collegially and help each other to learn the material and to solve development problems than to have you live in fear of our wrath and not talk to each other. Nonetheless, if we detect blatant cheating, we will deal with the cheaters as per Northwestern guidelines.

Schedule

Lecture	Date	Topics	Readings	Homework and Project
1	9/24	Class mechanics Introductory material, Web applications, client/server, and three-tier	GUW Intro; PG preface + 1	Project A (Microblog) out
Introduction to Unix Session (9/25, 6-8pm)				
2	9/26	More introductory material: why a database is different from a filesystem and what it helps you with. Data modeling, transactions/ACID, queries, abstracting storage+indices, some history lessons (Hierarchical, Network, Relational, Object, Object Relational, Hierarchical again)	GUW 1; PG preface + 1	
3	9/29	How web applications work.	PG 1-7, Perl	

		Apache, CGI, Perl, DBI, RDBMS, SQL in a nutshell	HO, Oracle HO	
4	10/1	SQL in a nutshell, Walk through Microblog (SQL)	PG 1-7, Perl HO, Oracle HO	Note: you might find PG 10 useful reading
5	10/3	Perl in a nutshell	PG 1-7, Perl HO, Oracle HO	
6	10/6	Walk through Microblog (Perl)	PG 1-7, Perl HO, Oracle HO	
7	10/8	Walk through Microblog (Perl)	PG 1-7, Perl HO, Oracle HO	HW 1 (ER Modeling) out
8	10/10	Data models and Data modeling: Why? Goals; and start Entity-Relationship: Entity sets, attributes, relationships, ER diagrams, instances, multiplicity, roles, multiway	GUW 2	
9	10/13	Entity-Relationship Model: conversion to binary relationships, subclassing, design principles	GUW 2	
10	10/15	Entity-Relationship Model: constraints, weak entity sets	GUW 2	Project A (Microblog) in. Project B (Self-defined) out
11	10/17	Relational Data Model: basics, translating from ER to relational	GUW 3	HW 1 in,
12	10/20	Relational Data Model: subclasses, functional dependencies	GUW 3	HW 2 out
13	10/22	Relational Data Model: Schema design and normal forms	GUW 3	
14	10/24	Relational Data Model: Multivalued dependencies	GUW 3	Project B proposal due
15	10/27	Other data models: OO and ODL	GUW 4	
16	10/29	Other data models: Object-relational	GUW 4, 9.4, 9.5	HW 2 due
Midterm Exam, Thursday, 10/30, 6-8:30pm, CS Classroom *Covers Everything Through Lecture 15.*				

17	10/31	Other data models: XML	GUW 4	Project B specification due
18	11/3	Relational Algebra: Sets: union, intersection, difference, selection, projection, Cartesian product, joins	GUW 5	HW 3 out
19	11/5	Relational Algebra: Bags, equivalent expressions, some extended operators	GUW 5	
20	11/7	Relational Algebra: grouping, constraints, data-mining	GUW 5	Project B ER diagram due
21	11/10	SQL: strings, regular expressions, date/time, nulls, 3-valued logic, explain plan, subqueries in/exists/>all/>any, correlation	GUW 6	
22	11/12	SQL: insert/update/delete, multi-statement transactions using PL/SQL; create schemas: bit-fields, decimal, blob; drop, alter; indexes; views	GUW 6	
23	11/14	SQL: Constraints, Triggers, systems aspects.	GUW 7, 8.1, 8.2, 8.3, 8.4, 8.6	Project B Relational Schema, DDL, DML, data due, HW 3 due, HW 4 out
24	11/17	Implementation: Storage *Instructor out of Town*	GUW 11	
25	11/19	Implementation: Representing Data *Instructor out of Town*	GUW 12	
26	11/21	Implementation: Indexes, Btrees	GUW 13, 14.4	Project B application logic due
27	11/24	Implementation: Indexes, Hashes	GUW 13, 14.4	
Thanksgiving Break				
28	12/1	Implementation: Indexes, Bitmaps	GUW 13, 14.4	
29	12/3	Implementation: Transactions	GUW 18	HW 4 due
30	12/5	Slack day		Project B Web front-end due

Final Exam, Wednesday, 12/10, 9am-11am

PG = Phillip Greenspun, *SQL for Web Nerds*
GUW = Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer D. Widom, *Database Systems: The Complete Book*