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: Fridays, 10-12, or by appointment

Teaching assistant

Ananth Sundararaj
1890 Maple Avenue, Room 332
847-491-7150
ais@cs.northwestern.edu
Recitation section: Mondays, 6-7pm, CS Small Classroom (342)
Office hours: Wednesdays 10-12, Office (332)
or by appointment

I would like to hire at least one undergraduate assistant for this course. Let me know if you know anyone who might be interested.

Location and Time

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

Prerequisites

Required	CS 311 or equivalent data structures course
Required	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
Highly recommended	Familiarity with C/C++

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 dive right in and modify a small database-based web application, a web log. Next, you will propose, design, and implement your own database-based web application. Finally, you will implement an on-disk BTree index. The majority of the programming in this class will be from scratch. We will use SQL, Perl, and C/C++ on Linux systems.

Projects

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. You will spend three weeks learning how Microblog works and extending it in several simple ways. The goal is to immediately introduce you to SQL right away and bring you up to speed on the programming elements of the course that you'll need for the second project.

The second project is self-defined project. You will propose your own problem and spend the remainder of the quarter designing and implementing your solution. This project will take four weeks.

The final project is to implement an on-disk BTree index, a key data structure used by a modern database system. This is lower level programming project that you will do in C or C++. This project will take three weeks.

Detailed descriptions of the requirements of the 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 (Microblog)
25 % Self-defined project (Your own web application)
15% Implementation project (Btree index)
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	9/22	Class mechanics Introductory material, Web applications, client/server, and three-tier	GUW Intro; PG preface + 1	Project A (Microblog) out
Introduction to Unix Session (9/28, 6-8pm, CS Large Classroom)				
2	9/24	More introductory material: why a database is different from a filesystem and what it helps you with. Data modeling, transactions/ACID, queries, abstracting	GUW 1; PG preface + 1	

		storage+indices, some history lessons (Hierarchical, Network, Relational, Object, Object Relational, Hierarchical again)		
3	9/27	How web applications work. Apache, CGI, Perl, DBI, RDBMS, SQL in a nutshell	PG 1-7, Perl HO, Oracle HO	
4	10/29	SQL in a nutshell, Walk through Microblog (SQL)	PG 1-7, Perl HO, Oracle HO	Note: you might find PG 10 useful reading
5	10/1	Perl in a nutshell	PG 1-7, Perl HO, Oracle HO	
6	10/4	More Perl	PG 1-7, Perl HO, Oracle HO	
7	10/6	Walk through Microblog (Perl)	PG 1-7, Perl HO, Oracle HO	
8	10/8	Walk through Microblog (Perl)	PG 1-7, Perl HO, Oracle HO	
9	10/11	Walk through Microblog (Perl)	PG 1-7, Perl HO, Oracle HO	HW 1 (ER Modeling) out
10	10/13	Data models and Data modeling: Why? Start Entity-Relationship: Entity sets, attributes, relationships, ER diagrams, instances, multiplicity, roles, multiway	GUW 2	
11	10/15	Entity-Relationship Model: conversion to binary relationships, subclassing, design principles	GUW 2	Project A (Microblog) in. Project B (Self-defined) out
12	10/18	Entity-Relationship Model: constraints, weak entity sets	GUW 2	
13	10/20	Relational Data Model: basics, translating from ER to relational	GUW 3	HW 1 in, Project B proposal due, HW 2 out
14	10/22	Relational Data Model: subclasses, functional dependencies	GUW 3	
15	10/25	Relational Data Model: Schema design and normal forms	GUW 3	

16	10/27	Relational Data Model: Multivalued dependencies *Instructor out of Town – Bin Lin Lectured*	GUW 3	Project B spec/ER/relational due; HW 2 due
Midterm Exam, Thursday, 10/28, 6-7:30pm, L150 Tech Covers Everything Through Lecture 15.				
17	10/29	Other data models: OO and XML	GUW 4	
18	11/1	Relational Algebra: Sets: union, intersection, difference, selection, projection, Cartesian product, joins	GUW 5	HW 3 out
19	11/3	Relational Algebra: Bags, equivalent expressions, some extended operators	GUW 5	
20	11/5	Relational Algebra: grouping, constraints, data-mining	GUW 5	Project B DDL/DML/ Queries/App Logic Due
21	11/8	SQL: strings, regular expressions, date/time, nulls, 3-valued logic, explain plan, subqueries in/exists/>all/>any, correlation *Instructor out of Town*	GUW 6	
22	11/10	SQL: insert/update/delete, multi-statement transactions using PL/SQL; create schemas: bit-fields, decimal, blob; drop, alter; indexes; views *Instructor out of Town*	GUW 6	Project B Final Handin; Project C out
23	11/12	SQL: Constraints, Triggers, systems aspects.	GUW 7, 8.1, 8.2, 8.3, 8.4, 8.6	
24	11/15	Implementation: Storage	GUW 11	
25	11/17	Implementation: Representing Data	GUW 12	HW 3 due
26	11/19	Implementation: Indexes, Btrees	GUW 13, 14.4	
27	11/22	Implementation: Indexes, Hashes	GUW 13, 14.4	
Thanksgiving Break				
28	11/29	Implementation: Indexes, Bitmaps	GUW 13, 14.4	
29	12/1	Implementation: Transactions	GUW 18	HW 4 due
30	12/3	Slack day		Project C due

Final Exam, Thursday, 12/9, 9am-11am, CS Large Classroom

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