Introduction to Database Systems

Syllabus

Web Page

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

Instructor

Peter A. Dinda
Technological Institute L463
pdinda@northwestern.edu
Office hours: Thursdays, 2-5pm (not first week), or by appointment

Teaching assistants

Maciej Swiech
Ford 2-221
dotpyfe@u.northwestern.edu
Office hours: Tuesdays, 2-5pm, or by appointment

Location and Time

Lecture: Technological Institute, Tech L150, 4-4:50pm, MWF

Optional TA-led Recitation: Mondays, 6pm, different rooms:

Tech M128 on:	10/7, 10/21, 11/4, 11/11, 11/25, 12/2
Tech M152 on:	10/14, 11/18
Tech L324 on:	10/28, 12/9

Recommended, especially for asking questions to clarify lecture and for help with projects

Prerequisites

Required	EECS 311 or equivalent data structures course
Required	EECS 213 or equivalent computer systems course
Recommended	Familiarity with concepts from discrete math such as set theory (EECS 2/310 for example)
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*, 2nd Edition, Prentice Hall, 2009. (Textbook - Required)

- An in-depth introduction to databases and database implementation

Phillip Greenspun, *SQL for Web Nerds*, http://philip.greenspun.com/sql/. (Required, but available for free on the web)

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

Joe Celko, *SQL for Smarties: Advanced SQL Programming*, 4th edition, Morgan Kaufman, 2010. (Useful)

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

Tom Christiansen, brian d foy, Larry Wall, Jon Orwant, *Programming Perl*, 4th Edition, O'Reilly and Associates, 2012. (Useful)

- This is the bible for the Perl language

Eben Hewitt, *Cassandra: The Definitive Guide*, O'Reilly and Associates, 2010. (Useful)

- This describes the Cassandra distributed database

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/JavaScript/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.
- How an RDBMS implements transactions.
- Special topics: database security, including SQL injection attacks, and, if time: NoSQL/distributed databases, CAP theorem. (This may be student choice)

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. At the beginning of the course, we will also use a very practical, and highly irreverent, introductory book on relational databases and web applications. The idea is to dive in quickly and learn how to use a database as the core of a web application, and then to back up and consider data modeling, query modeling, and database systems more deeply.

This is a learn-by-doing kind of class. You will dive right in and modify a small database-based, mobile, geolocating web application. Next you will design and implement your own database-backed web application for financial portfolio management. Finally, you'll implement a B+Tree index data structure, a common index structure used in many database engines. The majority of the programming in this class will be from scratch. We will use SQL, Perl, JavaScript, and C++ on Linux systems.

Projects

At the beginning of the course, I will provide you with a simple web application that implements a mobile, map-based view of political candidates, committees, and contributors. This application is based on an Oracle database and provides a web interface using a combination of client-side JavaScript and a CGI application written in Perl that talks to the database via DBI. This is a very common form of web application. You learn how this application works, and then you will extend it in several ways, focusing on the database backend. The goal is to immediately introduce you to SQL right away using a substantial dataset, namely the Federal Election Commission's disclosure database from 1980 to the present. This project will take three weeks.

The second project is focused on developing a simple financial portfolio manager that tracks a user's investments, and allows the user to "mine" historical financial data in several ways. I will give you a set of requirements and access to about 10 years of stock price data, and you will design and implement a database-backed web-based system. This project will take four weeks.

The third project is to build a B+Tree data structure. B+Trees are common on-disk (as opposed to in-memory) data structures used in relational database systems and many other systems. I will provide you with a framework, starter code, and a test harness.

Homework

There will be three homework sets that will be periodically assigned to help you improve your understanding of the material. These will focus on the entity relationship model, the relational model, and relational algebra.

Exams

There will be a midterm exam and a final exam. The midterm exam will take place in the evening outside of class. The final exam will not be cumulative.

Grading

15%	Project A: Dry-run project ("Red, White, and Blue")
20%	Project B: Portfolio manager project
15%	Project C: B+Tree project
10%	Homework
20%	Midterm
20%	Final

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

Note that the schedule is subject to change due to travel and other factors. I will announce schedule and due-date changes via email. If you do not receive a welcome email from me, please let me know.

Lecture	Date	Topics	Readings	Homework and Project

1	9/25	Class mechanics Introductory material, Web applications, client/server, and three-tier	GUW 1, 9.1, 9.3.1,9.3.2; PG preface + 1	Project A (RWB) out
If you're unfamiliar with Unix, now would be a good time to view the Unix introduction video available from the course web site.				
2	9/27	SQL in a nutshell, Start walk through of RWB (SQL)	PG 1-7, Perl HO, WOT HO	Note: you might find PG 10 useful reading
3	9/30	How web applications work. Apache, CGI, Perl, JavaScript, DBI, RDBMS, SQL in a nutshell, continue walk through of RWB (SQL)	PG 1-7, Perl HO, WOT HO, GUW 9.3.9	
4	10/2	Returning to the big picture: Relational and distributed databases, Data modeling, transactions/ACID, queries, abstracting storage+indices,	GUW 1; PG preface + 1	
5	10/4	Back to the nitty gritty: Perl	Perl HO	
6	10/7	Walk through RWB (Perl)	Perl HO	
7	10/9	Walk through RWB (Perl) *Instructor may be out of town*	Perl HO	
8	10/11	Database security topics or catchup *Instructor may be out of town*	GUW 10.1 (although lecture will focus elsewhere)	*Optional SQL Injection Attack Challenge*
9	10/14	NoSQL/Distributed Databases		
10	10/16	Data models and Data modeling: Why? Start Entity-Relationship: Entity sets, attributes, relationships, ER diagrams, instances, multiplicity, roles, multiway	GUW 2.1, 4.1-4.4	Project A (RWB) in. Project B (Portfolio) out HW 1 out
11	10/18	Entity-Relationship Model: conversion to binary relationships, subclassing, design principles	GUW 4.1-4.4	
12	10/21	Entity-Relationship Model: constraints, weak entity sets	GUW 4.1-4.4	

13	10/23	Relational Data Model: basics, translating from ER to relational	GUW 2.2, 2.3, 4.5	HW 1 in HW 2 out
14	10/25	Relational Data Model: basics, translating from ER to relational	GUW 2.2, 2.3, 4.5	
15	10/28	Relational Data Model: subclasses, functional dependencies	GUW 4.6, 3.1-3.2	
16	10/30	Relational Data Model: Schema design and normal forms	GUW 3.3-3.5, 3.6.6	
17	11/1	Relational Data Model: Multivalued dependencies	GUW 3.6	
18	11/4	Slack time or Midterm Review		HW 2 in
Midterm Exam: Tuesday, 11/5, 6pm, Tech L221 *Midterm Exam will cover Lectures 1-18*				
19	11/6	Relational Algebra: Sets: union, intersection, difference, selection, projection, Cartesian product, and cross, inner, outer, left, right joins	GUW 2.4, 5.1-5.2	HW 3 out
20	11/8	Relational Algebra: Bags, equivalent expressions, some extended operators	GUW 5.1-5.2	
21	11/11	Relational Algebra: grouping, constraints, data-mining	GUW 5.1-5.2, 2.5	
22	11/13	Advanced SQL: strings, regular expressions, date/time, nulls, 3-valued logic, explain plan, subqueries in/exists/>all/>any, correlation	GUW 6	Project B (Portfolio) in Project C (B+Tree) out
23	11/15	Advanced SQL: insert/update/delete, multi-statement transactions using PL/SQL; create schemas: bit-fields, decimal, blob; drop, alter; indexes; views	GUW 6, 7, 8	
24	11/18	Advanced SQL: Constraints, Triggers, systems aspects.	GUW 6, 7, 8	HW 3 in
25	11/20	Implementation: Storage and Representing Data	GUW 13	
26	11/22	Implementation: Indexes, Btrees	GUW 14.1, 14.2	

27	11/25	Implementation: Indexes, Hashes	GUW 14.3	
27a	11/27	*(probably no class)*		
Thanksgiving Break (6pm, 11/27; classes resume on 12/2)				
28	12/2	Implementation: Indexes, Bitmaps	GUW 14.7	
29	12/4	Implementation: Transactions (Logging, Locking)	GUW 17.1-17.4, 18.1-18.3	
30	12/6	Implementation: Transactions (Logging, Locking)	GUW 17.1-17.4, 18.1-18.3	Project C (B+Tree) in
Final Exam, Thursday, 12/12, 7-9pm, in our classroom. Covers Lectures 19-30				

PG = Phillip Greenspun, *SQL for Web Nerds*
GUW = Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer D. Widom, *Database Systems: The Complete Book*
Perl HO = *Perl in a Nutshell* handout
WOT HO = *Using Databases in the Web of Things Environment* handout