Introduction to Database Systems

Syllabus

Web Page

http://pdinda.org/db

Instructor

Peter A. Dinda
Seeley Mudd 3507
pdinda@northwestern.edu
Office hours: Thursdays, 2-5pm, or by appointment

Teaching assistants and peer mentors

Brian Suchy (TA)
Seeley Mudd 3301
BrianSuchy2022@u.northwestern.edu
Office hours: Mondays, 9-12, or by appointment
Wilkinson Lab

Michael Leonard (TA)
Seeley Mudd 3301
MichaelLeonard2018@u.northwestern.edu
Office hours: Wednesdays, 10-11am, 2-4pm (1st week also Friday, 1-4), or by appointment
Wilkinson Lab

Jin Han (PM)
Seeley Mudd 3301
JinHan2019@u.northwestern.edu
Office hours: Mondays, 1-4pm, or by appointment
Wilkinson Lab

Location and Time

Lectures:	MWF, 4-4:50pm, Tech L361
Optional Recitation w/TAs and PMs:	Wednesdays, 6pm, Tech M345
Midterm Exam:	TBD, midquarter, outside of class
Final Exam:	Friday, 3/22, 12-2, Tech L361

Prerequisites

Required	EECS 214 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 212 for example)

Recommended Some familiarity with a 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. While this is dated with respect to the presentation tier of a web application, it is still quite timely with respect to the interaction of the logic tier and data tier where the data tier is a relational database.
- The SQL examples given in this book are for the Oracle relational database, which is the database we will be using in projects.

Joe Celko, *SQL for Smarties: Advanced SQL Programming*, 5th edition, Morgan Kaufman, 2014. (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. We will use Perl extensively in the code provided with the projects.

David Flanagan, *JavaScript: The Definitive Guide*, O'Reilly and Associates, 2011 (Useful)

- This is an in-depth view of the JavaScript language, including its interaction with HTML in a web browser. JavaScript is **the** language used for the parts of web applications that run in browsers. We will use JavaScript for small parts of the projects where needed.

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 executes your queries using the relational algebra, the theoretical underpinning of database systems.
- How an RDBMS implements transactions.
- Special topics (if time): NoSQL/distributed databases, CAP theorem.

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, free 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.

The projects in the class can be done in groups of up to three students. We strongly encourage you to find a group early.

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 will 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.

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.

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.

Due to the number of students signed up for the course, and because I want the TAs to focus on the projects and other aspects of the course, we will simply verify that you have handed in each homework on time. We will not grade them. That is, if you handed in the homework on time, you will get the complete homework score. We will also make a solution to each homework available.

It is important that you make an effort to complete the homework and to understand the solutions we provide. Especially in preparation for the exams.

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 be in the university's scheduled date and time. 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 Handin
20%	Midterm
20%	Final

Grades are not a competition and this class is not curved. Your final score in the class will be computed as a *weighted sum* of the above elements. Final scores in the 90s will map to As, 80s to Bs, 70s to Cs and so on. Canvas will show you your current scores. Note that projects typically have extra credit.

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. You should also get an invitation to Piazza.

Lecture	Date	Topics	Readings	Homework and Project
1	1/7	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
2	1/9	SQL in a nutshell, Start walk through of RWB (SQL)	PG 1-7, Perl HO, JS HO, WOT HO	Note: you might find PG 10 useful reading
3	1/11	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, JS HO, WOT HO, GUW 9.3.9	
4	1/14	Returning to the big picture: Relational and distributed databases, Data modeling, transactions/ACID, queries, abstracting storage+indices, etc *Instructor may be out of town*	GUW 1; PG preface + 1	
5	1/16	Back to the nitty gritty: Perl	Perl HO	
6	1/18	Walk through RWB (Perl)	Perl HO	
1/21 - No Lecture - MLK Day				
7	1/23	Walk through RWB (Perl) *Instructor may be out of town*	Perl HO	
8	1/25	Slack time or special topic		

9	1/28	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	HW 1 out
10	**1/30**	**CANCELLED DUE TO WEATHER EMERGENCY - SCHEDULE PUSHED BACK ONE LECTURE USING PREVIOUS SLACK TIME ON 2/15**		Project A (RWB) in. Project B (Portfolio) out
11	2/1	Entity-Relationship Model: conversion to binary relationships, subclassing, design principles	GUW 4.1-4.4	
12	2/4	Entity-Relationship Model: constraints, weak entity sets	GUW 4.1-4.4	
13	2/6	Relational Data Model: basics, translating from ER to relational	GUW 2.2, 2.3, 4.5	HW 1 in HW 2 out
14	2/8	Relational Data Model: basics, translating from ER to relational	GUW 2.2, 2.3, 4.5	
15	2/11	Relational Data Model: subclasses, functional dependencies	GUW 4.6, 3.1-3.2	
16	2/13	Relational Data Model: Schema design and normal forms	GUW 3.3-3.5, 3.6.6	
17	2/15	Relational Data Model: Multivalued dependencies	GUW 3.6	HW 2 in
18	2/18	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
Midterm on Tuesday, 2/19, 6pm (90 minute exam) - Location TBD *Midterm will cover 1-17*				
19	2/20	Relational Algebra: Bags, equivalent expressions, some extended operators *Instructor may be away*	GUW 5.1-5.2	
20	2/22	Relational Algebra: grouping, constraints, data-mining *Instructor may be away*	GUW 5.1-5.2, 2.5	

21	2/25	Advanced SQL: strings, regular expressions, date/time, nulls, 3-valued logic, explain plan, subqueries in/exists/>all/>any, correlation *Instructor may be away*	GUW 6	
22	2/27	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	Project B in Project C out
23	3/1	Advanced SQL: Constraints, Triggers, systems aspects.	GUW 6, 7, 8	HW 3 in
24	3/4	Implementation: Storage and Representing Data	GUW 13	
25	3/6	Implementation: Indexes, Btrees	GUW 14.1, 14.2	
26	3/8	Implementation: Indexes, Hashes	GUW 14.3	
27	3/11	Implementation: Indexes, Bitmaps	GUW 14.7	
28	3/13	Implementation: Transactions (Logging, Locking)	GUW 17.1-17.4, 18.1-18.3	
29	3/15	Implementation: Transactions (Logging, Locking)	GUW 17.1-17.4, 18.1-18.3	Project C in
Final Exam, Friday, 3/22, 12 noon, in our classroom. *Final Exam will Cover Lectures 18-31* *We are in the process of possibly changing this. If it changes, the most likely days/times would be Tuesday or Wednesday of finals week, 6pm*				

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
JS HO = *JavaScript Model in a Nutshell* handout
WOT HO = *Using Databases in the Web of Things Environment* handout