**198:336** **Principles of Information and Database Management**

**SYLLABUS**

## Spring 2019

## Prof. Chon Lintakoon

## clintakoon@gmail.com

**Course Information**

# Prerequisites

The prerequisites for this course are 198:205 Discrete Structures I (or equivalent), for knowledge of First Order Logic; and 198:112 Data Structures, for data access techniques (hashing, search trees).

# TA and Project Help

# **Section 4, 6 TA:** Fei Deng email:fd213@rutgers.edu

# **Office hour:** Monday 10:00 AM - 11:00 AM at Hill 202

# **Section 5 TA:** Gang Qiao email**:** gq19@cs.rutgers.edu

**Office hours:** Monday 11:00 - 12:00 PM at Cubic J, CBIM (modular building near Busch student center), Busch campus

# **Project related help:**

# Varvara (Valia) Kalokyri via email only:[v.kalokyri@cs.rutgers.edu](mailto:v.kalokyri@cs.rutgers.edu)

# Course Goals

Unlike old-fashioned database courses, this course will cover aspects of ***using*** all kinds of information in the age of the Internet, including: structured data from databases; semi-structured data like XML, JSON and hyperlinked web pages; unstructured text; and even knowledge from which you can infer things. In general, the course will consider various *languages* for representing and accessing the different kinds of information, *methodologies* for using them, *theoretical principles* underlying them, and some *fundamental algorithms.* However, in contrast to standard database courses, we will be *very skimpy on implementation* aspects of DBMS, such as data storage and query optimization, which are covered in 198:437. If you want to be a database manager, you really must also take that course.

# Books

* [Database Management Systems](http://www.cs.wisc.edu/~dbbook), R.Ramakrishnan and M.Gehrke, *3rd edition !!!,*McGraw-Hill, 2003 (This book covers a surprisingly broad range of topics, including some XML, Information Retrieval and Data Mining)

The following authors also publish standard and excellent textbooks on *database systems*, which are widely used at other universities, in case you want to read alternate expositions

* [J.Ullman and J.Widom](http://www-db.stanford.edu/~ullman/fcdb.html) , "A First Course in Databases" (which is Volume 1 of the 2-volume [Garcia-Molina, Ullman and Widom: "Database Systems: The Complete Book"](http://www-db.stanford.edu/~ullman/dscb.html)). If this book, used in the first course on databases at Stanford, had material on information retrieval, it would have been my choice for a text, since it is also focused on the non-implementation aspects of databases. The accompanying web sites are outstanding! You are strongly encouraged to browse around to see what is available (lecture slides, exercises, exams, ...)
* ["Database systems: concepts, languages, and architectures",](http://dbbook.dia.uniroma3.it/) P. Atzeni, S. Ceri, S. Paraboschi, R. Torlone. (Excellent coverage of conceptual modeling, requirements, schema design. Out of print but **FREE** PDF below:

<http://dbbook.dia.uniroma3.it/dbbook.pdf>

* ["Fundamentals of Database Systems", R.Elmasri and S.Navathe](http://www.aw-bc.com/catalog/academic/product/0,1144,0321369572,00.html). Extensive coverage of conceptual modeling using Extended Entity Relational model and UML.
* ["Database System Concepts", A.Silebrschatz, H.Korth and S.Sudharshan](http://www.cs.yale.edu/homes/avi/db-book)
* ["Database systems: an application oriented approach", M.Kifer, A. Bernstein, P.Lewis. Addison-Wesley](http://www.pearsonhighered.com/educator/academic/product/0,1144,0321268458,00.html)

**BUT considerable material will be covered which is not in books. Students are responsible for knowing all the material covered in lectures, and *class attendance will be taken*.**

# Examinations

The only acceptable reason for not attending an exam is a **major** medical emergency, or a conflict with another exam. This will have to be documented, in order to be considered for a make-up, possibly oral, exam. (Please take this seriously. Leave early for the exam and be sure of its time and location.)

Exams will usually be cumulative, though we might post lists of specific topics. **Unless otherwise announced, all exams will be closed book.**

In addition, we *may* give short quizzes in lecture, following some of the assignments. These are intended to check that you did the assignments. There will be no make-up quizzes.

# Term work

There will be (bi-)weekly homework assignments and a 3-person team programming project.

# Homework

The written homework will be posted on the Sakai course web site, where students will submit solutions, and grades will be posted. *When uploading anything to sakai, please be sure to check it made it*; if not, try again; if the deadline is approaching, send email with the file before the due date, and then submit it late.  Assignments will be spot checked during the semester (this means that not all questions will be graded) and discussed in recitation.  **Do this homework!** It is intended to aid you in learning the material, and there is no substitute for trying problems. (For this reason, based on past experience, solutions will not be posted.) We will use material related to your homework on the tests.  Although you are welcome (even encouraged) to discuss homework questions with others, you must i) acknowledge such discussions on the solution submitted; ii) never leave the discussions with written notes (which means that simply copying solutions is considered a major breach of academic integrity).

# Project

The programming project will be graded principally on functionality, so fancy flashy images will not earn extra points (though you may find them a way to make the work more fun). On the other hand, nice user interface features, such as drop-down menus generated dynamically from a database, will be considered a plus. **In order to pass the course, a working programming project must be completed and handed in.  Individual contributions to the project will be measured and taken into account, so people trying to "coast on the coat-tails" of team-mates may fail.**In particular, the instructor may request an oral examination to further evaluate a student's understanding of the material involved and the way in which the program works; ultimately, this is what the grade is supposed to reflect!   
The only communication between teams should concern very general topics such as how to log in, how to install software and the like. Reusing software written by others or for other courses/projects is prohibited.

# Grading

**Expected Work:**

**1. Class Attendance and Participation 15%**

**2. Group Project 15%**

**3. Homeworks, Quizzes 20%**

**4. First Test (15%), Second Test (15%) 30%**

**5. Final Exam 20%**

**Grading standard:**

A 90-100

B Plus 86-89

B 80-85

C Plus 76-79

C 70-75

D 60-69

F 59 and below

**Regrading**: If you feel that an assignment or test has been improperly graded, please give it to the TA *within a week* of it having been returned to you, with a written note explaining the problem. No corrections will be considered beyond that limit.

# Policy on grading:

* To be fair to other students, current and past, unfortunately I have to be consistent, and stick to a longstanding policy we've evolved of denying requests to change *final grades*, even by doing additional work. The only exceptions to this are based on letters from the Dean explaining major personal difficulties. (I regret having to start with such a blunt message, but past experience ...)

# Attendance

You are responsible for all material I present or announce in lecture and assigned readings. As mentioned above, I will take attendance in lecture in order to determine grades at boundary points. If you expect to miss a class, please do not send me email; instead, use the University absence reporting website <https://sims.rutgers.edu/ssra/> to indicate the date and reason for your absence. An email is automatically sent to me (and your other instructors).

# Academic Integrity

Please read the [standard academic integrity policy](https://www.cs.rutgers.edu/academic-integrity/introduction)that has been developed for the Department of Computer Science.  I will strongly enforce this Policy and pursue all violations.

**Homework/Project Policy:** If get caught of copying homework or project from another students, all parties who have the same homework or project will receive zero and got put on probation on my list.  Second time caught cheating on the homework/project will be automatically given zero for the whole semester.

**Exam Policy:** NO chatting, NO restroom break, NO internet access through phone or other medium is allowed, no access to class related PowerPoint slides, or data files, no sharing of your knowledge with your neighbors or other students during test time. (Talking is strictly prohibited).

You will receive not only zero on that test, your file will be submitted to the dean's office for the consideration of dropping from the University.

Course Outline:

|  |
| --- |
| Topic |
| Overview of Database Systems |
| Database Design (ER) |
| Relational Model |
| From ER to Relational Model |
| Relational Algebra |
| Normalization |
| SQL |
| Transaction Management |
| Semistructured data (XML, JSON, AJAX, etc.) |

Adjustments to this schedule, as well as to the order of the topics will be made as necessary.

The following is a tentative list of topics to be covered, though the order is likely to be different in order to accommodate the project.

1. A functional view of an Information Service, illustrated with logic
   1. The I.S. as an abstract datatype with Ask and Tell operations: [a unifying framework for the course](http://www.cs.rutgers.edu/~borgida/442/2intro.pdf)
   2. *Datalog* Information Services (benefits: simple syntax)
   3. Negation and Recursion in Datalog (will be postponed)
2. Conceptual models: capturing domain information in ways natural to humans
   1. Extended Entity Relationship model
   2. Methodology for Conceptual Modeling
3. Structured Data: Relational Databases
   1. Overview of Database Management System services
   2. Relational Data Model
   3. Relational Algebra and its connection to Datalog
   4. SQL query language
   5. Advanced features of SQL: views, triggers, integrity constraints
   6. Web programming with database access
   7. Methodology of Relational Schema Design
      1. From EER Conceptual Models
      2. Functional dependency theory and Normal Forms
4. Semi-structured Data:
   1. XML: data as labelled tree
      1. Imposing structure: DTDs
      2. Querying:  XPath and XQuery
   2. JSON
5. Unstructured Data: Text ("Information Retrieval")
   1. Boolean retrieval
   2. Evaluation of IR systems
6. Concurrency control: transactions and levels of isolation
7. Advanced topics of current interest:
   1. NoSQL
   2. Data warehouses

**Course Outlines:**

|  |  |  |
| --- | --- | --- |
| Week # | **Chapter Discussion** | Comments |
| Week #1  Jan. 22, 25 | * Course Administration * A functional view of an Information Service, illustrated with logic * The I.S. as an abstract datatype with Ask and Tell operations: [a unifying framework for the course](http://www.cs.rutgers.edu/~borgida/442/2intro.pdf) |  |
| Week #2  Jan. 29, Feb 1 | * *Datalog* Information Services (benefits: simple syntax) * Conceptual models: capturing domain information in ways natural to humans   + - Extended Entity Relationship model | Testing of Datalog rules using facts using SWISH |
| Week #3  Feb. 5, 8 | * Conceptual models: capturing domain information in ways natural to humans   + - Methodology for Conceptual Modeling * Structured Data: Relational Databases   + Overview of Database Management System services | * Feb 5-HW0 (Create Amazon AWS access) due * Feb 8-HW1 (Create ER Diagram) due |
| Week #4 Feb. 12, 15 | * + Structured Data: Relational Databases   + Relational Data Model   + Relational Algebra and its connection to Datalog   + SQL query language | * Project Team Form (Feb 12) * Project Detailed Proposal (Feb 15) * Feb 15-HW2 (Datalog testing using facts and rules with SWISH) |
| Week #5 Feb. 19, 22 | * Methodology of Relational Schema Design * From EER Conceptual Models to SQL * Views, Outer join, IN, GROUP BY, HAVING, ORDER BY, NOT NULL | * Project ER Diagram (Feb 19) due * Feb 22-HW3 (Mapping ER diagram to Relation Schema) |
| Week #6  Feb. 26, Mar. 1 | * Methodology of Relational Schema Design | * Project Relational Schema (Feb 26) due * Mar. 1-HW4(Convert Relation Schema to Datalog and Relational Algebra) |
| **Week #7 Mar. 5, 8** | * **Test #1 Review** * **Test #1 in class (Mar. 8)** | **Mar 8 (Fri) in class Test 1** |
| Week #8 Mar. 12, 15 | * Test #1 result discussion * Normalization (1st, 2nd and 3rd Normal Forms) | Project Login Page (Mar 12) Due |

|  |  |  |  |
| --- | --- | --- | --- |
| Week #9  Mar. 19, 22 | * **Spring Recess (No Class)** | Mar 19-HW5-SQL due  Mar 22-HW6-Normal Form | |
| Week #10  **Mar. 26,** 29 | * Integrity constraints - Constraints (CHECK), Assertions, triggers * **TA explains Project Related (How to: JDBC and JSP)** | **Undergraduate Drop Period with a "W" grade (Mar. 26)** |
| Week #11  Apr. 2, 5 | * Functional dependency theory * BCNF Normal Form | Apr 9-HW7- Trigger due  **Final Project Due (Apr 8)** |
| Week #12  **Apr.** 9, **12** | * + Midterm 2 Review with TAs (Recitation Class ONLY)   + Data Warehouses, Data Mining |  |
| Week #13  Apr. 16, 19 | * **Apr. 16- Midterm 2 (In Class)** * Concurrency control: transactions and levels of isolation | Midterm 2 includes Midterm 1 scope and Integrity Constraints, BCNF, and Normalization  Apr 21-HW8-Data Mining |
| Week #14  Apr. 23, 26 | Concurrency control: transactions and levels of isolation Semi-structured Data:   * XML: data as labelled tree * Imposing structure: DTDs * Querying:  XPath and XQuery   Unstructured Data: Text ("Information Retrieval")   * JSON, NoSQL   Final Exam Review |  |
| **Week #15**  **Apr 30** | * **Final Exam Part 1 (In Class)** | **May 1, 2 University Calendar of Reading Days** |
| **Week #16**  **May 9** | * **Final Exam Part 2 (In Class)** |  |