

Spring 2010

CIS611 Supporting Material	
16	Tech Talks. Schedule
15	Lab Assignment 3. Implementing Relational-Design Tool (pdf file) or (zip-file) Due
14	Lecture Notes. Design Algorithms – Further Dependencies (Chp 11. Elmasri, Chp 6 Maier) Click here for pdf-file or zip-file .
13	Lab Assignment 3. Implementing The Canonical Cover Algorithm (pdf file) or (zip-file) Due Wed. Marc 31 st .
12	Lecture Notes. Functional Dependencies & Normalization (Chp 10. Elmasri, Chp 6 Maier) Click here for pdf-file or zip-file .
11	Lab Assignment 2. Implementing an efficient Attribute-Closure Algorithm (pdf file) or (zip-file)
10	Lecture Notes. Covers for Functional Dependencies (Chp 5. D. Maier) pdf-file or zip-file .
9	Lecture Notes. Functional Dependencies (Chp 4. D. Maier) pdf-file or zip-file .
8	Notice: Due to bad weather last Wednesday and Monday's holiday, the test and homework due on Wed. 17 th are postponed to next Monday March April 1 st
7	Query-By-Example. Click here for slides (available from Database Management Systems, R. Ramakrishnan and J. Gehrke. 2 nd Ed. McGraw-Hill) Additional Notes: Click here . Link to original paper by M. Zloof introducing QBE.
6	Technical Topics. Click here for a list of topics. Select one and prepare a 30 min talk on the subject.
5	Test on Wed. Feb 17th – Includes Relational Tuple Calculus and QUEL.
4	Lab Assignment1. An Experience on QUEL-INGRES (Company Database)
3	Lecture Notes: A Tutorial on QUEL / Ingres Part1 and Part2
2	Lecture Notes: Additional Notes on Relational Algebra – Relational Tuple Calculus Lecture Notes: A lab experience using Relational Algebra
1	Lecture Notes: Relational Model – Relational Algebra

Cleveland State University

CIS 611 Relational Database Systems (3-0-3) – Spring 2010
Section 1 – Class Nbr. 9624. Mon, Wed – 6:00 – 7:15 PM

Prerequisites: CIS 505 and CIS 530

Instructor: Dr. Victor Matos
Office Location: BU342 **Phone:** 216 687-3911 **email:** v.matos@csuohio.edu
Webpage: <http://grail.cba.csuohio.edu/~matos>

Office Time: Mo, We 10:50 -12:00 and 4:00-5:50 PM (or by appointment)
Class Location: BU-112

Catalog Description: This course is a continuation of CIS-430/530. It presents a theoretical approach to logical and physical database design. The course emphasizes relational data modeling, abstract query languages, design theory, dependencies, synthesis and decompositions, normalization, and query optimization. Study of data organizations-strategies for improved performance. It continues with an exploration of emerging non-relational models, their underlying architectures, data manipulation languages and other relevant features.

Key Concepts: Relational databases, normalization theory, functional and multi-valued dependencies, synthesis and decomposition algorithms. Object-Oriented databases, persistency, Object-Relational Mapping (ORM), querying objects.

Expected Outcomes: Upon successful completion of the course the student will be able to create -or reengineer- ‘well designed relational’ databases mostly free from redundancy and abnormal update syndromes. To contrast the world of normalized relational databases the student will be exposed to examples of non-relational databases such as object-oriented, temporal, deductive, XML, Spatial and Multimedia databases. In particular, the students will learn how to create applications using object-oriented database systems.

List of Required Materials: Microsoft Visual Studio 2008 (available from the Microsoft Academic Alliance program: http://msdn02.e-academy.com/elms/Storefront/Home.aspx?campus=cleveland_cis)

Text:

1. “The Theory of Relational Databases”, D. Maier, Ed. Comp. Sc. Press, 1983 ISBN 0-914894-42-0. We have obtained the author’s permission to use electronic copies of his book. A PDF version of the book is included in the ACM SIGMOD Anthology Vol. 4 Issue 1. Fall 2000 (<http://www.sigmod.org/sigmod/dblp/db/about/cd4-1.html>). The book is also available from Dr. J. Freytag web site (Humboldt Universitat, Berlin) at <http://www.dbis.informatik.hu-berlin.de/~freytag/Maier/>.
2. Elmasri / Navathe. "Fundamentals of Database Systems". Ed. Addison/Wesley Pub Co. 5th Edition, (2007).

ISBN-10: 0321369572

Official Calendar

Please consult the page <http://www.csuohio.edu/enrollmentservices/registrar/calendar/index.html>

Final exam: W, May 12 6:00-8:00 PM.

First Saturday Class begins Jan 16
 Martin Luther King Day (University Holiday) Jan 18
 First Weekday Class Jan 19
 Last Day to Add Jan 22
 Last Day to Drop Jan 29
 President's Day (University Holiday) Feb 15
 Midterm Grades Mar 8-14

Spring Recess Mar 14-21
 Classes Resume Mar 22
 Last Day to Withdraw Apr 2
 Last Day of Classes May 7
 Final Exams May 8-14
 Commencement May 15
 Spring Incomplete Deadline Sept 24

Grading: The course grade is based on a student's overall performance through the entire Semester. The final grade is distributed among the following components:

- | | |
|---------------------------------|-------------------------------|
| 1. Exams (Mid & Final) | 60% (30% each) |
| 2. Computer Projects | 30% (about 6 lab assignments) |
| 3. Research Topic Presentation: | 10% |

A	94% +	A: Outstanding (student's performance is genuinely excellent)
A-	90% - 93%	
B	82% - 87%	B: Very Good (student's performance is clearly commendable but not necessarily outstanding)
B-	80% - 82%	
C	75% - 80%	C: Good (student's performance meets every course requirement and is acceptable; not distinguished)
D	65%-75%	D: Below Average (student's performance fails to meet course objectives and standards)
F	<65%	F: Failure (student's performance is unacceptable)

Examination Policy: Students are allowed to bring to the tests a summary page (standard letter size) with their own notes. During the exams: (1) the use of books, cell phones, calculators, or any electronic devices is prohibited, and (2) students must not share any materials.

Make-Up Exam Policy: No makeup exams will be given unless notified and agreed to in advance. Requests will be considered only in case of exceptional demonstrated need.

Homework Policy: The students are expected to attend all classes. The students are responsible for collecting the notes, handouts and any other course material distributed during the class period. All assignments must be individually and independently completed and must represent the effort of the student turning in the assignment. Should two or more students turn in *substantially the same solution* or output, in the judgment of the instructor, the solution will be considered group effort. All involved in group effort homework will receive a zero grade for that assignment. A student turning in a group effort assignment more than once will automatically receive an "F" grade for the course.

Late Assignment: All lab assignments are due at the beginning of class on the date specified. Laboratory Assignments handed in after the class has begun will be accepted with a 25% grade penalty for up to a week and then not accepted at all. All laboratory assignments must be completed. *Failure to do so will lower your course grade one additional letter grade.*

Student Conduct: Students are expected to do their own work. Academic misconduct, student misconduct, cheating and plagiarism will not be tolerated. Violations will be subject to disciplinary action as specified in the [CSU Student Conduct Code](http://www.csuohio.edu/studentlife/StudentCodeOfConduct.pdf). A copy can be obtained on the web page at: <http://www.csuohio.edu/studentlife/StudentCodeOfConduct.pdf> or by contacting Valerie Hinton Hannah, Judicial Affairs Officer in the Department of Student Life (MC 106 email v.hintonhannah@csuohio.edu). For more information consult the following web page *CSU Judicial Affairs* available at <http://www.csuohio.edu/studentlife/jaffairs/faq.html>

Course Schedule: The schedule of topics and their order of coverage is given below. Every effort will be made to follow the schedule, but topics covered may vary depending upon the progress made.

Week of	Topic	Reading
---------	-------	---------

1, 2	<p>Review:</p> <p>The relational model of data. Attributes and atomic domains, Key and Referential Integrity rules (original Codd's paper).</p> <p><i>Classic Relational Query Languages:</i> <i>Algebra.</i> (S J P Todd, "The Peterlee Relational Test Vehicle - A System Overview", IBM Systems Journal, 15,4, pp.285–308, 1976) . <i>Tuple Calculus</i> (The Design and Implementation of INGRES. Stonebraker et al. ACM Transactions on Database Systems, Vol. 1, No. 3, September 1976, Pages 189-222). <i>Domain Calculus</i> (ZLOOF, M.M. Query by Example. Proc. AFIPS 1975 NCC, Vol. 44, AFIPS Press, Montvale, N.J., pp. 431-437, Chapter6 from Database Management Systems 3rd Edition by Raghu Ramakrishnan and Johannes Gehrke. Ed. McGraw-Hill. Available at http://pages.cs.wisc.edu/~dbbook/openAccess/thirdEdition/supporting_material.htm). <i>Mixed Algebra-Calculus: SQL</i> (System R, Structured English Query Language)</p>	Elmasri. Chp. 1-9 Listed papers.
3,4	Functional Dependencies. Propositional logic. Inference axioms, application of the Armstrong axioms. B-Axioms, RAP sequences. Completeness of the axioms. Testing membership in F+	Maier - Chp. 4 Elmasri - 10, 11
5	Covers for Functional Dependencies. Covers and equivalence. Non-redundant covers. Extraneous attributes. Canonical covers. Minimum and optimal covers.	Maier - Chp. 5 Elmasri - 10, 11
6,7	Normalization Theory. Database and database schemes. Redundancy and abnormal behavior. Normal forms, 1NF, 2NF, 3NF, BCNF. Techniques for mechanical design. Lossless-Join decompositions. Decompositions that Preserve Dependencies. Normalization through Decomposition and Synthesis. Shortcomings of each technique. Problems with BCNF.	Maier - Chp. 6 Elmasri - 10, 11
8	Multi-valued Dependencies Properties, MVDs and functional dependencies. Inference axioms for MVDs. Fourth normal form (4NF).	Maier - Chp. 7 Elmasri - 11
9	Query Modification. Symbolic optimization. Computing Access Paths for Efficient Execution. QUEL Query decomposition. Instantiation and Iteration. Improving queries containing aggregations.	Maier - Chp. 11 Elmasri - 15
10-11	<p>Case Studies: Non relational technologies</p> <ul style="list-style-type: none"> • Object-Relational: Using Cache DBMS from Intersys. • Object Oriented: Using Fast-Objects from Versant. • Associative Data Model: Using Sentences from Lazy Software. • XML Databases: Oracle XML Features. • Raw Objects: Using DB4O persistency methods. 	Elmasri 20, 21, 27. Lecture Notes
12-13	<p>Enhanced Data Models for Advanced Applications</p> <ul style="list-style-type: none"> • Temporal Databases • Spatial and Multimedia Databases • Deductive Databases 	Elmasri 24
14	Understanding LINQ: Language Integrated Query	Notes

NOTE: The instructor reserves the right to retain, for pedagogical reasons, either the original or a copy of your work submitted either individually or as a group project for this class. Students' names will be deleted from any retained items.

List of Assignments:

1. **Lab Phase1:** Relational Database Explorer Project: Consulting the MS-Access database dictionary: Tables (and their composition), Queries (and their underlying SQL, parameters), Reports, etc. Programming on the .NET platform.
2. **Lab Phase 2:** Database Explorer: Constructing a library of C# (or VB.NET) routines to provide SQL support to relational algebra operators.
3. **Lab Phase 3:** Develop a Syntax Oriented Translator to parse and execute algebra statements using the API you created in the previous lab assignment.
4. **Lab Phase 4:** Computing the Closure of a set of Attributes using a linear time algorithm.

5. **Programming Experience.** Discovering functional dependencies in an arbitrary data set.
6. **Lab Phase 5:** Computing the Canonical cover of a set of functional Dependencies, detecting dependency preservation and lossless join decompositions.
7. **Lab Phase 6:** Detecting potential abnormal behavior of relational schema dependencies F. Implementing the synthesis algorithm to obtain 3NF databases, implementing BCNF databases.
8. **Lab Phase 7:** Computing the dependency basis of a set of attributes X subject to a set M of multivalued dependencies. Obtaining 4NF decompositions. Evaluating quality of the design.
9. **Programming Experience.** Implement (design/populate) an Object-Oriented version of the Company Database (use design provided in class). Use LINQ to phrase the queries (to be) given in class.
10. **Programming Experience.** Repeat previous programming experience; this time use Java Objects and provide persistency using the DB4O OODBMS.