COMP SCI 451 Database Design & Applications - Fall 2011
Course web site: [http://www.uwgb.edu/shayw/courses/451]
Meeting times: MW 8:00-9:20; in MAC 122

Professor: Bill Shay **Office**: MAC C308
Contact Information: The best way to contact me is to talk after class or by email: shayw@uwgb.edu
Office Hours: After class or by appointment. Email me or see me after class to schedule appointments.
Prerequisites: COMP SCI 257 Software Design II with C or better

Text: *Database Processing: Fundamentals, Design, & Implementation (11th ed)* by David M. Kroenke & David J. Auer
Books on Reserve: *Microsoft Visual C# 2010: An Introduction to Object Oriented Programming (4th ed)* by Joyce Farrell.
Pro C# 2010 and the .NET Platform (5th ed) by Andrew Troelsen
Inside SQL Server 2008 T-SQL Programming by Itzik Ben-Gan, et. al.
Pro T-SQL 2008 Programmers Guide by Michael Coles

Course:
Unlike other courses, which focus largely on program design, this course focuses in part on data modeling. Primarily we address the question: How can we organize and provide access to large amounts of inter-related data of different types? This course will mix the theory behind the relational database model with practical application and client/server database technology. There will be a semester-long project in which team members collaborate to design and build their own relational database subject to a variety of constraints and build applications which perform a variety of tasks. Teams will use *SQL Server 2008* to build a server database and C# (written in Visual Studio 2010) to create client applications that interact with it. Be aware, however, that this is NOT a course in C#, Visual Studio 2010, nor *SQL Server*. Certainly we need to use elements of all of these and some class time will be devoted to introducing the basics of each. However, because this is a 400-level course, I make the assumption that each student has the ability to become more knowledgeable in each of those environments through independent work. The C# application will utilize *ADO.NET* controls to access the database. *ADO.NET* defines a class hierarchy that can be used to access data from a wide variety of sources such as *SQL Server*, *Oracle*, *MS Access*, spreadsheets, and *XML* pages. This is a powerful, albeit complex, collection of classes that allows applications to be written independent of the data source.

Primary course goals include:

- Understand how to organize and define relationships among a wide variety of different data types in order to reflect a diverse set of requirements and constraints.
- Understand the operations and theory of the relational database model.
- Gain working knowledge of SQL (structured query language), stored procedures, and triggers used to access and manipulate data in a relational database.
- Gain practical experience with writing an application that interfaces with a relational database. Specifically, each person will write a C# application that interfaces with a relational database using *ADO.NET* data controls.
- Understand mathematical models that define the expressive capability of a query language and help to optimize data queries.
- Understand how to categorize relations and determine when they might be subject to anomalies.
- Introduce security measures required to protect data.
- Understand techniques used to guarantee the integrity of data in a database.
- Understand the problems of providing concurrent data access to multiple users.

Grading: Project: 50%; Midterm exam: 20%; Final Exam: 30%;
The final exam is scheduled for Friday, December 16, 2011 from 8:00-10:00 and must be taken at this time. This date is known many months in advance and any plans such as work schedules must be planned around this date.

Approximate syllabus: **minor changes are possible**

Week	Topics
1	Introduction to databases; overview of storage structures and access methods: linked lists, indexes, B-trees, hash functions. File organizations: Sequential, indexed, and hashed files. Introduction to SQL Server Management Studio and the SQL Server Database Management System. (Chapter 1; first few pages of chapter 10 ; Appendix D (online at http://wps.prenhall.com/bp_kroenke_database_11/127/32761/8386898.cw/index.html); SQL Server databases *CS451 Supplier* and *CS451 Packers*).
2-3	Entity-relationship (E/R) modeling: strong and weak entities; attributes; one-to-one, one-to-many, and many-to-many relationships, binary, ternary, and recursive relationships. (Chapter 5). In-class group activity: Project design.
4	Database design: tables, relationships, primary and foreign keys, referential and entity integrity rules, integrity constraints; creating tables from E-R diagrams. (Chapter 6 and 7; SQL Server database *CS451 Supplier*).
5	Structured Query Language, SQL, queries and subqueries, inner and outer joins; creating and updating views, (Chapters 2, 7, and 8, SQL Server database *CS451 Supplier*). Also [http://www.sqlcourse.com] and [http:www.sqlcourse2.com].
6-9	**Exam on concepts, design, and SQL**. Database access via ActiveX Data Objects (ADO.NET) data controls: connection objects, data adapters, datasets, currency managers; C# applications and controls: forms, listBoxes, datagrids, bindings, buttons, etc.; server-side programming using stored procedures and triggers (sections on stored procedures and triggers from chapter 10; first few sections of Chapter 11; Course demos; reserve books).
10	Data normalization: functional dependencies, update anomalies, multivalued dependencies, first through fourth normal forms, non-loss table decompositions (Chapter 3 and 4)
11	Deductive relational databases, rules, logic programming (SQL Server database *CS451 Family*).
12	Relational algebra, optimization, implementing relational operators (notes), query analysis from the demo databases.
13	Multiuser databases: Concurrency control, deadlock, locking, transaction processing, two-way commit, backup and recovery. (Chapter 9).
14	Group presentations and/or group meetings (Attendance required for all if we do presentations).