IS 331
Database Design, Management and Applications

Instructor: Todd Will
Office: EarthLab, GITC4215
Phone: 973-876-1148
E-Mail: todd.will@njit.edu
Skype: toddapod
Office Hours: Thursday, 6-9PM, via Skype
Textbook: **D. Kroenke and D. Auer, DATABASE PROCESSING: FUNDAMENTALS, DESIGN & IMPLEMENTATION, 13 Edition, Pearson/Prentice-Hall, 2013. ISBN: 10: 0133058352 or 13: 978-0133058352**

Course Description: Businesses use databases extensively for analysis and decision-making because they provide efficient, large-scale information storage and rapid retrieval. Databases support the "back end functionality" of most large web systems. This course gives students extensive, pragmatic experience in designing, building, querying, updating, maintaining and managing relational databases, using the Structured Query Language (SQL). Proper database design principles are emphasized throughout the course, beginning with high level descriptions of relational databases using data modeling tools (such as ER (Entity Relationship) diagrams) and progressing to relational database design principles based on higher order normalizations. We will examine some poorly designed databases and show how these can be transformed into well designed databases. SQL will be extensively covered, and students will design and implement sophisticated SQL queries invoking self-joins, outer joins, correlated subqueries and related concepts. Students will explore and utilize design methodologies for input data validation and maintaining database integrity, and study related issues of database privacy and security. Advanced topics to be discussed include the role of the Database Administrator (DBA), database life cycle activities, database denormalizations, read-only databases and data warehouses. Hands-on experience will be gained by working with actual databases using industry-standard database management systems such as Oracle.

Prerequisite: Completion of a 100-level GUR course in computing, such as IS117 or CS101 or CS104 or CS113. (It is also assumed that students have some basic familiarity with Microsoft Office (2007 or 2010), particularly Microsoft Access.)

Class Communication Space/Learning Management System: We will be using Moodle, an open source Learning Management System at NJIT, for the posting of projects and class resources and other class announcements are postings. Students having questions on projects, etc., may contact Chao directly at cx26@njit.edu or, if the answer would benefit the class, post the question in the appropriate forum within Moodle. Students are obligated to log into Moodle on a near-daily basis, and to keep current.

Course Goals: To understand the design and development issues regarding databases. Students will obtain a strong conceptual foundation of the underpinnings of database design, as well as gain experience with some commercial database management products, ranging such as Oracle. IS331 seeks to provide the student with the conceptual and pragmatic aspects and issues related to designing, implementing, managing, deploying and utilizing database applications, with the emphasis on the student's comprehension of key concepts in database design (as opposed to programming or algorithms). The student will utilizes various DBMS products and software tools, including Oracle and Aqua Data Studio, to put into practice the database concepts presented.

Lecture Notes: Presentation slides will be downloadable each week from Moodle

Course Grade Components:

- Midterm Exam - 30%
- Final Exam - 30%
- Database conceptual assignments and DBMS projects (30%) – 7 throughout semester
- Class participation / attendance 10%.

Policy on Collaboration/Cheating: Every assignment/project is a 'home-mini-exam.' The NJIT Honor Code will be strictly upheld. Students found cheating/collaborating/plagiarizing will be **immediately** referred to the Dean of Students and the NJIT Committee on Professional Conduct and subject to possible Disciplinary Probation, a permanent marking on the record, **possible dismissal and a grade of 'F' in the course**. **All submitted assignments are carefully checked for similarities, and plagiarism and guilty students will be identified and referred to the Dean of Students for disciplinary action.**

Policy on Lateness of Submission: Every assignment/project will have a due date/time, and all submissions must be submitted by this due date/time. **Assignments submitted after the due date/time will NOT be accepted.**

Below are the TOPICs covered in the course and the related TEXTBOOK readings. Remember one of the keys to success in IS331 is your own self-discipline - your goal should be to maintain currency each week, and NEVER fall behind! In addition to viewing the slides every week and completing the homework, you also need to post a question, comment, or something interesting you read for the week in the forum each week; this will count as your class participation grade.

WEEK	WEEK STARTING	TOPICS	BOOK READINGS	HOMEWORK
1	**May 23**	**Welcome & Intro** **File processing systems** **Database Intro**		**Assignment 1** **Assignment 2**
2	**May 30**	**Conceptual Design 1** **Conceptual Design 2**	**Chapter 2**	**Assignment 3**
3	**June 6**	**Conceptual Design 3** **Conceptual Design 4**	**Chapter 3**	**Assignment 4**
4	**June 13**	**Logical Design 1** **Logical Design 2**	**Chapter 4**	
5	**June 20**	**Logical Design 3** **Logical Design 4**	**Chapter 5**	**Assignment 5**
6	**June 27**	**Midterm Exam** **Physical Design** **Data Definition Language** **Data Definition Language Lab**		
7	**July 4**	**Relational Algebra 1** **Relational Algebra 2**		
8	**July 11**	**Relational Algebra 3** **Relational Algebra 4**	**Chapter 7**	

9	**July 18**	**SQL Practice** **SQL Developer**	**Chapter 8** **Chapter 9**	**Assignment 6**
10	**July 25**	**PL SQL** **Creating Views and other objects**	**Chapter 10**	
11	**August 1**	**Transaction Processing** **Data Warehousing**	**Chapter 11 (optional)** **Chapter 12**	**Assignment 7**
12	**August 8**	**Final Examination**		
13	**August 11**	**Grades due at Registrar**		

Student Outcomes After Completing IS331

- The student can demonstrate the use of SQL to properly and optimally design and implement a database from a set of user requirements, insuring data integrity, and providing a high level description of the database, using Entity Relationship Data Modeling.
- The student will be able to identify poorly designed databases, and redesign them into well-designed databases.
- The student will be able to construct sophisticated SQL queries, using advanced SQL topics such as self-joins, inner and outer joins, and correlated sub-queries.
- The student can define the critical responsibilities of a database professional, particularly in regard to privacy of data, security, and integrity.
- The student is able to apply the conceptual ideas underlying relational database design, including design concepts such as referential integrity, conforming to higher order normalization designs, functional and multivalued dependencies and relationships.

Brief List of Topics to be Covered

- a. properties of databases
- b. flatfile databases vs. relational databases
- c. high-level descriptions of databases, using Entity-Relationship Diagrams with standard and IE Crow's Foot Nomenclature
- d. translating ER Models to Relational Designs
- e. methodologies for querying a database, including Relational Algebra operators and graphical query interfaces such as GQBE
- f. fundamental syntax of SQL, creating a database with SQL
- g. examining some poorly designed databases, and rectifying their design
- h. responsibilities of the database professional, including issues of database integrity, ethical obligations involving privacy/security of data
- i. functional dependencies and multivalued dependencies
- j. optimal database design using Normalization (1NF, 2NF, 3NF, BCNF, 4NF, DKNF)
- k. denormalization and when denormalization should be invoked l. updateable vs. read-only databases
- m. referential integrity, foreign key constraints and implications, and casual relationships n. data warehouses
- o. Aqua Data Studio & Oracle
- p. advanced SQL query design, including sophisticated SQL queries using self-joins and inner/outer joins
- q. input validation and database integrity r. database redesign
- s. correlated subqueries in SQL