CS 361 – Database Systems
Syllabus – Spring 2017

INSTRUCTOR: Tom Naps

OFFICE: Halsey 214, phone 424-1388

EMAIL: naps@uwosh.edu

OFFICE HOURS: MWF 9:30 - 10:30, Thur 10:00 - 11:30

CLASS MEETING TIME: MWF 11:30 - 12:30

REFERENCES:

- Any handouts, readings, guide-sheets, assignments etc. that are needed for a class session will be posted by midnight of the day preceding the class meeting. They comprise a high-level, but not detailed, outline of what we will cover during class. Get a copy of them before class, organize them, take notes on and about them. Handouts that are not liberally saturated with your own explanatory notes will likely prove useless when you need them most.
- Review problems posted on D2L at the end of each class. Complete them before the start of following class meeting and be ready to discuss the answers you have submitted.
- Optional but good references (all three have provided material for my lecture slides, review problems, exam problems):
  - *Database Management Systems – the "Cow Book"*, 3rd edition, Raghu Ramakrishnan and Johannes Gehrke. For DB modeling techniques, we will employ the notation used in this book.
  - *Database Systems: The Complete Book (2nd Edition)*, Hector Garcia-Molina, Jeffrey D. Ullman and Jennifer Widom, Prentice Hall, 2008
  - Jennifer Widom's *free* Database MOOC at Stanford `https://class.stanford.edu/courses/DB/2014/SelfPaced/about`. Some of the material there matches pretty well what we will cover under topics 1-6, 10, 13 below

Topic Coverage

1. Overview of database systems
2. Introduction to database design
3. The relational model
4. Relational algebra and calculus
5. Schema refinement and normal forms
6. SQL
7. Database application development
8. Storage and indexing
9. Query evaluation and optimization
10. Transaction management
11. Concurrency control
12. Crash recovery
13. Non-relational ("NOSQL") data models – XML, JSON, MongoDB

Learning Outcomes

Given our coverage of these topics, you will be expected to . . .

- Fundamental concepts of database design
  - model customer requirements of a relational database with an Entity-Relational diagram (E-R diagram)
  - transform an E-R diagram to a database schema
  - write a query to a relational database in SQL
  - formulate a query to a relational database from the basic operators in relational algebra
  - design a database to provide the necessary information for an organization while minimizing redundancy and null entries.
- SQL
  - design and create a database using the Data Definition Language of a Database Management System
  - write queries to a relational database in an interactive mode
- Design of "good" relations, Schema Refinement, Concept of normalization and other theoretical issues
  - formulate the integrity constraints in the form of functional dependencies
  - eliminate extraneous attributes in a functional dependency
  - eliminate redundant functional dependencies
  - develop a cover from a set of functional dependencies
  - evaluate a proposed relational schema and determine whether it is in Third-Normal-Form (3NF) or Boyce-Codd-Normal-Form (BCNF)
  - implement a normalization program that checks whether a proposed relational schema is in 3NF or BCNF
  - decompose proposed relational schemas that are not in 3NF or BCNF into 3NF or BCNF
- Basic file organization and various file structure methods
  - determine the access time of records based on the file organization and file structure

  - specify the type of stable and non-stable storage in the design of a database management system.
  - analyze the requirements and select the design of an index (Hash, B+ tree)
  - organize data on disk to minimize disk accesses for various queries.

- Algorithms and implementation of large database systems

  - analyze the need of a database operator (scan, equality search, range search, insert, delete etc) and determine an appropriate and/or efficient algorithm (external sorting, hash, B+, clustered vs. unclustered, various join algorithms) in its implementation.
    Implement a Relational Operation (Example: Join)

- Transaction processing, concurrency issues, and recovery

  - identify and prevent deadlocks in concurrent database accesses
  - be able to describe the recovery process of databases
  - design the data structure and program of a database recovery mechanism
  - provide accurate, consistent, and efficient transactions within the context of concurrency issues and the possibility of various kinds of failures, such as a system crash.

Course Grading Policies

Your grade for the course will be based on the following weighted factors:

Factor	**Weight**
Homework Assignments (4 or 5)	22.5%
"Real-world" project (Done in four phases – conceptual design, relational schema design and refinement, SQL database definition and implementation of essential queries, application development and testing)	22.5%
Exams (three – dates to be announced):	45%
Daily Class Participation	5%
Review problems	5%

To get the 45% contribution to your grade from the three exams, I will use the formula:

$$E = 0.10 \times E_{worst} + 0.20 \times E_{best} + 0.15 \times E_{other}$$

where E_{worst} is your worst exam score and E_{best} is your best exam score. At the end of the term, your work in all of these areas will contribute to a numerical grade for the course based on a 100-point scale. Grade cutoff levels on this final scale are:

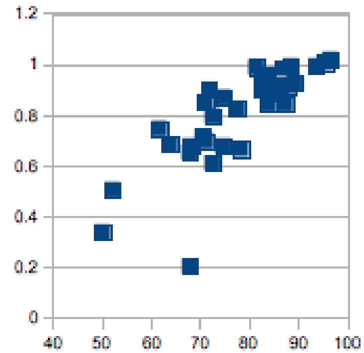
A $\geq$ 92	B $\geq$ 82	C $\geq$ 72	D $\geq$ 62
A- $\geq$ 90	B- $\geq$ 80	C- $\geq$ 70	D- $\geq$ 60
B+ $\geq$ 88	C+ $\geq$ 78	D+ $\geq$ 68	F $<$ 60

FAQ

Do I have to come to class? You are expected to arrive prepared to ALL the course sessions. Furthermore you are expected to participate in the classroom discussions and activities to the best of your abilities. This includes your taking advantage of the opportunity to present your solution to a review problem when called upon to do so. Your doing this will not only ensure that you get the full 5% "class participation" component factored into your grade; more importantly the learning that transpires during these activities will help tremendously in preparing you for what is on the exams. It is difficult to envision a student missing and/or arriving unprepared to a number of the class sessions and still succeeding in the course.

What if I'm late in completing and submitting a homework assignment or phase in the project? It will be accepted but will be penalized at the rate of 10% of point value the first day late, *an additional* 20% the second, *an additional* 30% the third ...

What if I'm late in completing the daily review exercises? You will get zero credit for that set of review exercises. Although the review problems only count 5% of your grade, the following correlation from a previous course between review-problem-percentage (on a 0 to 1.0 vertical scale) and overall percent in the course (on a horizontal scale of 0 to 100) is indicative of their true importance.



We will always discuss the review problems assigned on a given day at the beginning of the next class. If you have participated in class the day the review problem was distributed, have made a good faith effort to work on the review problem, and are "stuck" on it, I will be more than happy to help you with it if you come my office anytime within three days after the review problem has been assigned. After those three days, *because you have made the choice to not learn efficiently*, you are on your own in terms of grappling with these review problems. In particular, I will not entertain questions about past review problems as an exam approaches.

If I miss an exam, can I make it up? If you are unable to take a scheduled exam, it may be possible to take a make-up exam provided that you do BOTH of the following, which are then subject to my approval:

- Make arrangements prior to the scheduled exam (for last minute emergencies, telephone me at 424-1388 or leave a message at the computer science office, 424-2068). No after-the-fact notifications will be accepted . . . *AND*
- Have a written medical excuse signed by the attending physician OR have a note of justification from the Dean of Students Office.

Only one make-up exam will be given. It will be a rigorous comprehensive exam given at an arranged time during the last week of the semester.

Can I work with others on the homework assignments or the project? You have the choice to work on your own or with *one* partner, in effect working in *pair learning/programming* style. (See research by A. Cockburn and L. Williams indicating that, when done conscientiously, working in paired environments will slightly increase the time you put in and significantly improve the quality of the work.) You are not allowed to collaborate with anyone else for assignments. If you decide to work with a partner, here is the way that it must be done:

- *Both* you and your partner must "declare" via emails sent to me *three days* before an assignment is due that you will be working together. Once that declaration is made, it will remain in effect until you email me again that you no longer wish to work as partners.
- If you decide to work with a partner, only one of you will submit the assignment, and that is what I will grade. You will both receive the same grade.
- You may not share or even discuss your work with anybody but your partner unless you can live with a zero and the other potential academic sanctions of cheating (see the UWO Student Discipline Code, Chapter UWS 14).

If you decide to work on your own, then the third bullet point above also applies (minus the partner).