

Course Syllabus for CIS 315 – Database Design and Implementation – Fall 2010

Basic Course Information:

Instructor:	Sharon Tuttle	
Lecture time & location:	Monday	1:00 - 2:50 pm HGH 106
Lab time & location:	Wednesday	1:00 - 2:50 pm BSS 313
Instructor's office:	BSS 322	
Instructor's e-mail:	st10@humboldt.edu OR sharon.tuttle@humboldt.edu OR smtuttle@humboldt.edu	
Instructor's office phone:	(707) 826-3381	
Instructor's office hours:	Monday, Wednesday	3:15 - 4:45 pm
	Tuesday, Thursday	9:30 - 10:30 am
	or by appointment	
Course public web page:	follow link from: http://users.humboldt.edu/smtuttle/ OR follow link from course Moodle site	

Course Description:

[from the HSU catalog] Design/implementation concepts for relational model. Enterprise and entity-relationship modeling. Schema development: normalization; SQL data definition and data manipulation language; user-defined types, rules, and triggers to support the schema. Features to support integrity, ease of use, and control: concurrency, locking, distribution, performance.

This course is an introduction to database design and modeling; it includes fundamental database concepts including introductions to the relational model, relational operations, entity-relationship modeling, converting an entity-relationship model into a database schema or design, normalization, and transactions and concurrency management. It also includes an introduction to the fourth generation language SQL (Structured Query Language), used as both a data definition language (DDL) and a data manipulation language (DML) in the implementation and use of relational databases. We will be using an Oracle implementation of SQL.

Course Objectives:

After successfully completing this course, students should be able to: *

- understand what a database is, and what a database management system (DBMS) is;

* Some of these are adapted from the ACM Computer Science Curriculum 2001, available from link at: <http://www.acm.org/education/curricula-recommendations>

- understand the basic components of a database design/schema, and of a DBMS;
- understand a database model;
- create a database model for at least a simple scenario;
- convert a database model into a corresponding database design/schema;
- understand the advantages of creating a database design/schema based on a carefully-designed database model as opposed to simply attempting to create a database design/schema directly from scratch;
- understand the concept of normalization;
- normalize a given set of relations and functional dependencies to at least Third Normal Form (3NF);
- use basic SQL fluently:
 - creating and deleting tables;
 - creating tables with appropriate primary and foreign keys;
 - manipulating table contents: inserting, updating, and deleting rows;
 - querying table contents, including expressing the most important relational operations, selection, projection, and equi-join, using SQL;
 - using SQL `select` statement clauses such as `order by`, `group by`, and `having`;
 - using nested SQL `select` statements;
 - using SQL operators such as `exists` and `in`;
 - creating, executing, and debugging SQL scripts run within an enterprise-level multi-user DBMS;
- understand the concept of a constraint on a database: limitations to the possible values that can be stored, to increase data integrity;
- be familiar with entity integrity constraints (implemented using primary keys) and referential integrity constraints (implemented using foreign keys);
- understand that a DBMS might support further constraints to be defined/implemented, via such mechanisms as database triggers and SQL `check` clauses.

Course Prerequisites:

Either the three courses: CIS 230, CIS 250, **and** CIS 260,

or the course CS 233,

or instructor approval.

Also, the course MATH 253 is **recommended**.

Note that you must have received a grade of at least **C** in the required prerequisites to enroll for this course.

Required Course Text, Materials, etc.:

- Turning Point RF Response Clicker, available at the campus bookstore, or a Responseware license, available on-line (address to be provided).
- Links to on-line required readings will be provided.

Recommended Course Text:

- "Database Modeling and Design", Fourth edition, by Teorey, Lightstone, and Nadeau, published by Morgan Kaufman, ISBN #0-12-685352-5.
 - Required readings will not be assigned from this text, but you may find it to be a useful **reference** for some of the course topics.

Course Software:

As mentioned, we will be using an **Oracle implementation of SQL** for this course; you are expected to use the `student` Oracle database on the campus computer `nrs-labs.humboldt.edu` for the homework/lecture/lab assignments and for the course project, and exam questions involving SQL will be assumed to use Oracle SQL as well. The software you use for any additional practice is, of course, up to you.

(Note: it is possible that the HSU Oracle database we use may change during the course of the semester; if this happens, you will be notified about what database you should be using for this course, and how to access it.)

Throughout the semester, you will be making some use of the UNIX operating system (since that's what `nrs-labs.humboldt.edu` uses). Note that you may access `nrs-labs.humboldt.edu` by using `ssh` (secure shell) and `sftp` (secure ftp); `ssh` may be downloaded for free from:

<https://experts.humboldt.edu/ditss/download/> or
<http://www.humboldt.edu/its/software>

...although in Fall 2008 some students had better luck downloading it from:

http://www.colorado.edu/its/docs/authenticate/printouts/win_ssh.html

...which also includes a nice illustrated tutorial for this Windows implementation of `ssh` and `sftp`.

I hope that we will also be making use of some free software from Oracle, **SQL Developer**, which has versions for Windows, Linux, and Mac OS X, and can be downloaded for free from:

<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

It should also be available in the following on-campus labs:

BSS 313, BSS 317, FOR 204A, LIBR 121, and LIBR 122

Note that you can access the HSU Oracle `student` database from on- or off-campus using `nrs-labs`; however, at this time, you can only use SQL Developer to access it from on-campus. We hope to allow off-campus access using SQL Developer some time during the semester. More on this as it develops.

Clickers:

We will be using Turning Technologies student response clickers (or, for those who prefer, Responseware on one's cell phone or laptop) in lecture and sometimes in lab. There is significant literature indicating that using clickers may increase student engagement and success in learning.

Students purchase this clicker or buy a Responseware license; purchased clickers can be returned at the end of the semester for a partial refund of the purchase price. Students with clickers register them once, at the beginning of the semester, by entering the large number (consisting of 6 characters/digits) on the back of the clicker at a special address that I will provide, and then bring them to every class meeting. Students using Responseware purchase the license, and then sign into Responseware at the beginning of each class meeting, using a special code that I will project at the beginning of each class meeting.

These clickers will be used for in-class questions, which will be interspersed during the lecture and lab sessions (although probably with more questions during lecture sessions, since lab sessions may also include lab exercises). The response system will record the overall class response percentages as well as keep track of individual answers. Students will receive **2 points** for a correct answer, **1 point** for an incorrect answer, and **0 points** for no answer, but with a maximum semester clicker-questions grade of **120**. Thus you will be rewarded for regular attendance and participation. If you miss a class session, you miss that day's clicker questions and cannot make them up. However, there will be at least **65** questions asked over the course of the semester, so there is opportunity for extra credit (up to a maximum clicker grade of **120**) (or to make up for a day that you are out due to illness, although note that you are still responsible for finding out what you missed on such days).

If you forget your clicker for a class meeting, then **up to 5 times** you may still receive some clicker credit, **minus a 2-point penalty**, by e-mailing me your clicker answers for that day, **by midnight on that day**, using the Subject: **CIS 315 Clicker Answers for <date>**. Later e-mails, or e-mails without the proper Subject: line, will not be accepted for credit.

The idea is that the clicker questions will help you to see if you are starting to understand concepts being discussed; sometimes they will also provide review of concepts discussed previously. Clicker questions are typically quite different from exam questions (since clicker questions are typically multiple-choice questions, while exam questions will rarely be multiple-choice). They still enable you to get some immediate feedback regarding whether you are grasping course concepts, whether you need to pay more attention to course discussions and/or readings, etc. They may even help me to know what concepts might need more explanation in-class.

I hope to run tests of the system during Wednesday's lab during the first week of the semester, and hope to begin asking questions that "count" during the second week of the semester. Therefore, you must purchase your clicker and register it (or purchase a Responseware license) as soon as possible. If there is an issue with this (for example, if the bookstore runs out of clickers), contact me immediately.

Finally, please note that use of another CIS 315 student's clicker, or having someone else use your clicker in CIS 315 class -- that is, pretending that a student is in class who actually is not -- is considered to be cheating, with the same policies applying as would be the case if you turned in someone else's work as your own or permitted someone else to copy your work. Please **ASK ME** if you are not sure what I mean by this.

Grading Breakdown:

If you are a Computer Information Systems (CIS) major, it is important that you note that you must earn **at least a C in CIS 315** for this course to count towards your major and to be able to take CIS 318 - Programming Database Applications. If you are a Computer Science (CS) major, it is important that you note that you must earn **at least a C in CIS 315** for it to be able to count as a CS major elective.

Your semester grade will be determined by the percentage of points that you earn, **subject to some minimum requirements**. Here are the grade percentages, followed by those minimum requirements:

Homework assignments:	20.0%
Lab exercises:	10.0%
Clicker questions:	10.0%

Exams:	Exam #1:	12.5%	
	Exam #2:	12.5%	
	Final Exam:	15.0%	Monday, Dec. 13, 12:40 - 2:30 pm, HGH 106
Project:		20.0%	

Grade Requirements:

- To earn a grade of **C or better** in this course, the following four requirements must **all** be met:
 - your overall semester average must **equal or exceed 72.5%** - this is to show a reasonable level of overall mastery of the course material.
 - the **average** of your Exam #1, Exam #2, and Final Exam grades must **equal or exceed 60%** - this is to show that you understand at least a minimal reasonable level of the most important course concepts.
 - the **average** of your Homework assignments must **equal or exceed 60%** - because SQL is an important component of this course, but SQL acumen is not tested as effectively on exams, this is to show at least a minimal level of SQL competence and experience in addition to course concept mastery. Also, past experience has shown that students who do not put a solid effort into the course homeworks do not do well on the course exams.
 - your Project grade must **equal or exceed 60%** - because, in a lot of ways, the project is tying together many of the most important aspects of this course, including some that cannot be tested as effectively on exams. If you have not met this requirement, you have not truly shown minimal competence in this course's material.
- If **all four** requirements above are **not** met, then your semester grade will be **either C-** or the letter grade computed according to the mapping given below, **whichever is lower**.
 (That is, if a student had an overall semester average of 74% but a Homeworks average of 55%, that student would receive a **C-** for his/her semester grade; if a student had a Homeworks average of 61% and an Exams average of 71%, but an overall semester average of 65%. then that student would receive a **D** for his/her semester grade. You are expected to ASK ME if this aspect of the grading policy is not clear to you.)
- Including the four requirements noted above, your semester grade will be computed according to the mapping given below:

Overall Percentage (based on the given weights)	Exams Average	Homeworks Average	Project Grade	Letter Grade
>= 93	>= 60	>= 60	>= 60	A
>= 90 and < 93	>= 60	>= 60	>= 60	A-
>= 87 and < 90	>= 60	>= 60	>= 60	B+
>= 83 and < 87	>= 60	>= 60	>= 60	B
>= 80 and < 83	>= 60	>= 60	>= 60	B-
>= 77 and < 80	>= 60	>= 60	>= 60	C+
>= 73 and < 77	>= 60	>= 60	>= 60	C
>= 73	< 60	any	any	C-

Overall Percentage (based on the given weights)	Exams Average	Homeworks Average	Project Grade	Letter Grade
>= 73	any	< 60	any	C-
>= 73	any	any	< 60	C-
>= 70 and < 73	any	any	any	C-
>= 67 and < 70	any	any	any	D+
>= 60 and < 67	any	any	any	D
< 60	any	any	any	F

Final Exam:

Again, the Final Exam for this course is scheduled for **Monday, December 13, 12:40 – 2:30 pm**, in **HGH 106** (unless I announce otherwise). Note this time and date BEFORE making your end-of-semester travel plans.

Students with Disabilities:

Persons who wish to request disability-related accommodations should contact the **Student Disability Resource Center** in the basement of the Library, Library 055, **826-4678 (voice)** or **826-5392 (TDD)**. Some accommodations may take up to several weeks to arrange. You can reach the Student Disability Resource Center's web site at:

<http://www.humboldt.edu/disability/>

Please note that some accommodations may take up to several weeks to arrange; also, if you are eligible for such accommodations, please contact me as soon as possible to discuss them.

Add/Drop Policy:

Students are responsible for knowing the University policy, procedures, and schedule for dropping or adding classes. You can find these on the web at:

<http://www.humboldt.edu/registrar/students/regulations/schedadjust.html>

You can find the University policies for repeating classes at:

<http://www.humboldt.edu/registrar/students/regulations/repeat.html>

Note that the CSU (and thus HSU) policies on withdrawing from and repeating courses changed as of Fall 2009:

- Students may withdraw from no more than 18 semester-units after the first four weeks of instruction; that is, students may withdraw from no more than 18 semester-units between census and the final 20% of instruction, and only then with a serious and compelling reason.
- Students may repeat courses only if they earned grades lower than a C.
- Students may repeat up to 16 semester-units with grade forgiveness.
- Students may repeat up to an additional 12 semester-units with grades averaged.

Be careful – as of Fall 2009, HSU is being much more strict about what constitutes a “serious and compelling reason”.

The census date for Fall 2010 (before which you can drop without a W, and without it counting toward your 18 semester-units drop limit) is: **5:00 pm on Monday, September 20th**.

The last date for Fall 2010 to drop with a W on your transcript, with a serious and compelling reason, and subject to the 18 semester-unit drop limit, is: **Friday, November 12th**.

If you do drop the course, note that it is **your responsibility** to complete and submit the appropriate paperwork. As noted in the University policies for dropping courses, "As a matter of university policy, **the instructor in the course may not drop on behalf of the student**."

Incompletes:

Incompletes are rarely given and only in the case of a true emergency. They certainly are not appropriate for students who find they have fallen behind on assignments, missed a test, or taken on too much academic, work, or family responsibilities. For these situations, dropping the course would be appropriate (if that is still possible according to the University policies for dropping courses).

Time Expectations:

Remember the general rule of thumb for college-level courses --- to be successful in a course, you should plan to spend at least 3 hours outside of class for each 1 hour of college course credit. That implies an estimate of at least 9 hours a week spent outside of class for this 3-credit course.

However, you should be warned that:

- This is a junior-level CIS major course; it has an accordingly-rigorous workload.
- Programming courses can be notorious time eaters --- occasionally, a problem with a SQL script will take large amounts of time to locate and fix. Starting early enough so that you have time to ask me questions when you run into problems can help with this. Why spend 4 hours struggling with a frustrating roadblock the night before the assignment is due, when you can spend 10 minutes composing an e-mail early in the week, work on other problems while waiting for the answer, and then get a reply that makes everything clearer as soon as you read it?
- You can only learn SQL (and database modeling, and database design, etc.) by practicing them, and it takes some much longer than others to master them. Practicing as much as possible helps. (This means playing around with in-class examples, experimenting to see if something you are curious about really works like you think, and so on.)
- The course will intensify as the semester progresses – as you are able to do more, you will be expected to do more. If you ask me as soon as you realize that some concept is not clear to you, that can help keep you from falling behind.
- Homework deadlines will not be extended because you waited too late to start or because you did not allocate enough time before the deadline to work on it; likewise, they will not be extended because of hardware or network failures. You need to keep backups of your files at all times, and need to plan your schedule to be able to work on on-campus computers as necessary.
- If you have not completed an assignment by the deadline, your best choice is to submit whatever you have managed to do by then, as partial credit is your friend, to carefully study the posted example solution as soon as it is available, to ask me about anything there that is still unclear, and to get a good early start on the next homework.

Academic Honesty:

Students are responsible for knowing policy regarding academic honesty:

http://www.humboldt.edu/studentrights/academic_honesty.php

Observe that among the actions that are unacceptable are submitting another's program, code, or file as your own and failing to quote material taken from another person's written work. (Note that copying another student's comments is also unacceptable.)

All course work is to be the work of each student, **individually**, unless it is **explicitly** stated otherwise at the beginning of that course work's description. Except for explicit exceptions, this is **not** a group or team programming course. When group work is explicitly permitted, the names of all students involved must be included on the work submitted. (For example, when you use **pair programming** in lab, the lab exercise will specify that, and the files you turn in will include both of the names of the students who worked as a pair.)

(When an assignment does specify that it is acceptable to work in pairs or groups, make sure that you don't get into the situation where you are merely watching someone else learn.)

For homework assignments (that are not specified as pair-programming), students may discuss general approaches **as long as no one involved in the discussion is writing anything down or typing anything during such discussions**. Students may also help one another in determining causes of program bugs, or in determining the meaning of compiler error messages. However, students may not work together to complete homeworks, one student should not instruct another in how to write the code for a homework, and **any type of copying or modifying of another person's computer files, OR of providing computer files to another, related to homeworks is definitely over the line, and never justified**. This applies to copying of documentation and comments as well as to copying of program code.

Note that it is **your** responsibility to ensure that course assignment files are read-protected. If you are careless about this, and someone else copies your work, you will share the penalty. (In particular, be very careful about leaving work on shared network drives in campus labs, or in UNIX/LINUX directories that are not read-protected.)

Learning takes hard work; when students turn in others' work as their own, it is a slap in the face to those seriously interested in learning. Not turning in an assignment results in no credit for that assignment, of course, but that is an honest grade. Work that violates the course honesty policy deserves a lower grade than that, and therefore the course policy is that work violating this policy will receive **negative** credit. A person providing a file for copying receives the same **negative** credit as the copier. Repeat offenses will be handled according to University policies.

Asking Questions/Getting Help:

- You are encouraged to ask me questions in class, in office hours, and by e-mail. The most successful students are those who are not afraid to ask questions early and often (I will gently let you know if you are overdoing it), who do the assigned reading, who attend lecture and lab regularly, who start homeworks promptly after they are made available from the course web page and start project milestones early, and who practice course concepts as much as possible.
 - It is better to ask a question sooner than later -- for example, it is better to send an e-mail with a specific question as soon as you think of it than it is to wait a day or two until the next class meeting or office hour. If you wait to ask such questions, you may not have time to complete the assignment.
 - It is perfectly reasonable if you send me a question and then end up finding out the answer yourself before you receive my answer; likewise, it is not a problem if you end up sending me several questions in separate e-mails (as you work on different parts of a homework while awaiting earlier answers).
- That said, I am expecting that you will ask **specific** questions – overly vague or broad questions are

problematic. (For example, an example of a specific question is, “When I try to run query: (paste in the query), I receive the following error message: (paste in the error message) Can you point me in the right direction about what is wrong?” An example of an overly vague or broad question is: “Here's my SQL script/database model/database design/etc. Is it right?”)

- I try to check my e-mail (st10@humboldt.edu or sharon.tuttle@humboldt.edu or smttuttle@humboldt.edu) about once a day on weekdays, and about once over each weekend. This is another reason to start assignments early, so that you have time to receive a reply to any questions that might arise. Include **CIS 315** and a general description of your topic in the Subject: line, both because including this makes it less likely that I'll overlook your question, and because it will make your message stand out if the spam filter gets confused and puts it in the university spam quarantine.
 - If I have not replied to your e-mail within 24 hours, please re-send it, just in case it did get overlooked somehow.
 - Also, try to avoid the word "password" in your e-mails to me -- pwd is a handy abbreviation to use instead -- because, due to phishing scams, HSU's spam filtering definitely does not like e-mails with that word in it! (Odd, but this was definitely the case in Spring 2010...)

Additional Assignment-Related Policies:

- You should not expect to be able to finish course assignments during the lab sessions--- like any college-level course, you should expect to put in a (potentially) large amount of time outside of scheduled class meetings (lectures and labs) doing the assigned reading, working on homework assignments and project milestones, and practicing concepts discussed.
- Each assignment must be submitted as is specified on its handout to be accepted for credit. This may vary for different assignments. Often, parts of assignments will be submitted using a special tool on nrs-labs. Scripts that do not run using the HSU Oracle DBMS will not receive credit; remember that it is your responsibility to verify that your scripts run on the HSU Oracle DBMS before submitting them, regardless of where you developed them.
- Each assignment will be clearly marked with one or more due dates (a single assignment could have multiple parts with multiple due dates -- this is especially true of the course project).
 - **No assignments will be accepted late. If you wish to receive any credit for an assignment, then you must turn in whatever you have done, even if it is incomplete, by the deadline. Partial credit is usually preferable to no credit.** Note that "the computer/network/etc. going down" is no excuse --- if you leave an assignment for the last minute and there are technical problems, you still must turn in whatever you have by the deadline. As with any work done on computer, make frequent back-ups of your files!
 - You may submit multiple versions of assignment files before the deadline; I will grade the latest pre-deadline submission unless you inform me otherwise. This is to encourage you to turn assignment parts in early (since you will know that you can always turn in an improved version if further inspiration strikes). You also don't have to worry about forgetting to submit something that has already been submitted.
- The tool that you will be using to submit some assignment parts results in a file that serves as your "receipt" for having submitted items. You are expected to retain these "receipt" files at least until a grade has been posted to the course Moodle site for that assignment. If there is a system glitch or other hardware/software/network problem, you may be asked to make me a copy of one or more receipt files; if you do not have them, then you will not receive credit for the files involved. These receipt files are for your protection!

- It is nearly impossible to write unambiguous specifications. If you have questions about "what she means", get them resolved very early in the development cycle by **asking**.
- There is more to SQL scripts than simply whether they run or not...
 - Part of your grade will be determined by how well your work meets the written requirements. Work that you turn in is expected to meet handout specifications precisely; when one eventually works within a team on large projects, following the specifications precisely is vital, and can mean the difference between a working product and one that just sits there.
 - Note that work may be graded on **style** as well as on whether it runs properly and whether it precisely meets the homework specifications and requirements. Discussions on style will be ongoing throughout the semester.
- Some course work may be graded simply based on whether it has been attempted (the instructor's decision is final as to whether this is the case) -- other course work may be graded for correctness, style, and whether it meets specifications. You will not know in advance which will be the case.

Additional Grading-Related Policies:

- If you would like me to e-mail certain course grades to you during the semester, then you must give me permission in writing on the course information form.
- Clicker questions will be given during most lectures and labs (although probably more clicker questions will be asked during lecture sessions); graded lab exercises will be given during most lab sessions. The two lowest lab exercise grades will be dropped from your grade. Between the ample quantity of clicker questions and the dropped lab exercise grades, then, you can be absent several times from non-exam lecture or lab sessions without direct penalty, for whatever reason (although you are, of course, still responsible for the material covered on those days, and it is your responsibility for determining what that material is).
- Note: **NO** homework grades are dropped; **ALL** homework grades count toward your homework average. Every homework includes important practice of course fundamentals. Likewise, no project milestone grades are dropped, and all of them likewise count toward your project grade.

Additional Course Policies:

- You are expected to read this syllabus and be prepared to sign a statement that says you have received and understand these policies.
- Exam dates are given in the course schedule below. Make-up exams are only possible by special prior arrangement, or because of a valid medical excuse.
- You should monitor your e-mail for course-related messages. The University provides a means for you to specify your preferred e-mail address, so if you wish to receive e-mail into an account other than the one HSU provides, change your preferred e-mail address in both Account Center and Moodle accordingly. Course-related messages from me will include **CIS 315** in the Subject: line.
- You are expected to check the public course web page and the course Moodle site regularly --- course handouts, homework assignments, examples from lectures and labs, and possibly more will be posted to the public course web page, and grades will be posted to the course Moodle site. You are expected to monitor your posted grades and let me know about any discrepancies.
- Typically, we will discuss conceptual topics such as general database design and theory during the course lectures, and will discuss Oracle and SQL during the course lab sessions. The labs thus have

more of a "hands-on lecture" orientation than you may be used to --- I'll typically be discussing SQL syntax, demonstrating it, and building a **SQL script** for later reference.

- When reading assignments are given, you are expected to prepare (read and study) assigned readings before class and to participate in class discussions. Projected examples will be utilized frequently during discussion. You should understand that there may be material in the reading that will not be discussed in lecture/lab, and material in the lectures/labs that may not be found in the reading. You are responsible for both.
- **Attendance and disruptive behavior:** Students are responsible for knowing policy regarding attendance and disruptive behavior:
http://www.humboldt.edu/studentrights/attendance_behavior.php
- Regular attendance at lecture and lab sessions is expected. If you should happen to miss a lecture or a lab, then you are responsible for finding out what you missed. "I wasn't there that time" is never an acceptable excuse. Lecture notes are not posted, although many of the projected examples will be made available on the public course web site. Clicker questions and graded lab exercises missed **cannot** be made up later.
- As previously mentioned, during lab sessions, there may be lab exercises due during that lab session. Once a lab's lab exercise is complete, the remaining lab time should be used to continue work on the current course homework assignment, to practice course concepts, and/or to ask questions about course-related topics.
- **Late arrival to class:** Please attempt to come to class on time, with your headphones put away and your cell phones turned off. If you must arrive late or leave early, please do so with the least possible distraction to other students. If your late/early habits become disruptive, you may be asked to leave the class permanently.
- **Class disruption:** University policy requires that instructors eliminate disruptions to the educational process. Distractions such as excess talking, ringing cell phones, working on assignments for other classes, inappropriate or distracting laptop/tablet/smartphone/gadget use, demonstrations of affection, packing of books early, loud music leaking from headphones, chronic late arrivals or early departures, excessive comings and goings or other behaviors that disrupt the class are not acceptable. Students indulging in such behaviors will first be warned before being required to leave the class permanently.
- **Emergency Evacuation:** Please review the evacuation plan for the classroom (posted on the orange signs), and review the campus Emergency Preparedness web site at:
http://www.humboldt.edu/emergencymgmtprogram/campus_emergency_preparedness.php
for information on campus Emergency Procedures. During an emergency, information regarding campus conditions can be found at **826-INFO** or:
<http://www.humboldt.edu/emergency>

Tentative Course Schedule: (subject to change!)

Week 1: August 23, 25

- Lecture: Introduction to course
- Lab: UNIX refresher; accessing Oracle; clicker tests

Week 2: August 30, September 1

- Reading: Reading Packet - "Database processing and development"
- **Project Proposal sub-milestone** due by **5:00 pm** on **Monday, August 30**
- Lecture: Introduction to database processing and development
- Lab: Introduction to SQL*Plus and SQL; running SQL scripts and saving script output; **HW #1 out**

Week 3: September 8

- Monday, September 6 - Labor Day Holiday - NO CLASS
- Reading: Reading Packet - "More database fundamentals"
- Optional Reference: Teorey/Lightstone/Nadeau, Appendix, Sections A.2, A.3, A.3.2
- Lab: **HW #1 due**; More database fundamentals; database integrity; the role of SQL: DML, DDL, and DCL; database security part 1: SQL `grant` command
- **Project Proposal final milestone** due by **5:00 pm** on **Friday, September 10**

Week 4: September 13, 15

- Reading: Reading Packets: "The relation model and relational operations" and "Writing relational operations in SQL"
- Optional Reference: Teorey/Lightstone/Nadeau, Appendix, Section A.3.1
- Lecture: Introduction to the relation model and relational operations
- Lab: Writing relational operations using SQL; **HW #2 out**

Week 5: September 20, 22

- Reading: Reading Packets: "Entity-relationship modeling, part 1" and "More where clause options and aggregate functions"
- Optional Reference: Teorey/Lightstone/Nadeau, Chapter 1, Section 1.3, Chapter 2, Section 2.1, Appendix, Section A.3.1
- Lecture: Introduction to Entity-Relationship modeling
- Lab: **HW #2 due**; More options for the SQL `select` statement `where` clause; introduction to SQL aggregate functions; database security part 2: using `&` for "interactive" queries and the definition of SQL injection; **HW #3 out**
- **Project Model sub-milestone** due by **5:00 pm** on **Friday, September 24**

Week 6: September 27, 29

- Reading: Reading Packets: "Entity-relationship modeling, part 2" and "Sub-selects, concatenating columns, and projecting literals"
- Optional Reference: Teorey/Lightstone/Nadeau, Chapter 2, Section 2.2

- Lecture: Introduction to Entity-Relationship modeling, continued
- Lab: **HW #3 due**; SQL `select` statement sub-selects and correlated queries; concatenating columns, projecting literals

Week 7: October 4, 6

- Lecture: Review for Exam #1
- Lab: **Wednesday, October 6: EXAM #1**

Week 8: October 11, 13

- **Project Model final milestone due by 5:00 pm on Monday, October 11**
- Reading: Reading Packet: "Normalization"
- Optional Reference: Teorey/Lightstone/Nadeau, Chapter 6, Sections 6.1, 6.2, 6.3, 6.4
- Lecture: Introduction to normalization
- Lab: in-lab Project Model review exercise; **HW #4 out**

Week 9: October 18, 20

- Reading: Reading Packets: "Database design, part 1" and "Order by, group by, and having"
- Optional Reference: Teorey/Lightstone/Nadeau, Chapter 5
- Lecture: Introduction to the basics of database design
- Lab: **HW #4 due**; More SQL `select` clauses: `order by`, `group by`, and `having`; **HW #5 out**

Week 10: October 25, 27

- Reading: Reading Packets: "Database design, part 2" and "Set-theoretic operations, more on modifying data, SQL*Loader, and sequences"
- Optional Reference: Teorey/Lightstone/Nadeau, Chapter 5, Appendix, Sections A.3.1, A.3.2, A.3.3
- Lecture: Introduction to the basics of database design, continued
- Lab: **HW #5 due**; SQL `union`, `intersect`, and `minus`; the "full" SQL `select` statement; more on SQL `update` and `delete`; introduction to SQL*Loader; introduction to sequences

Week 11: November 1, 3

- Lecture: Review for Exam #2
- Lab: **Wednesday, November 3: EXAM #2**
- **HW #6 out**
- **Project Design milestone due by 5:00 pm on Friday, November 5**

Week 12: November 8, 10

- (Thursday, November 11 - Veterans Day Holiday - does not affect CIS 315)
- Reading: Reading Packets: "Transaction management, part 1" and "Views; & simple reports, part 1"
- Optional Reference: Teorey/Lightstone/Nadeau, Appendix, Section A.3.4
- Lecture: Introduction to transaction management (transactions and concurrency)
- Lab: **HW #6 due**; Introduction to SQL views; database security part 3: how views and the `grant` command can improve security; creating simple ASCII reports, part 1; **HW #7 out**

Week 13: November 15, 17

- Reading: Reading Packets: "Transaction management, part 2" and "Simple reports, part 2"
- Lecture: Introduction to transaction management (transactions and concurrency), continued
- Lab: **HW #7 due**; SQL*Plus commands for creating simple ASCII reports, part 2; some handy SQL string-, date-, and time-related functions for prettier reports; **HW #8 out**
- **Project Population milestone** due by **5:00 pm** on **Friday, November 19**

Thanksgiving Holiday - Monday-Friday, November 22-26**Week 14: November 29, December 1**

- Reading: Reading Packet: "A few words on OODBMS's and ORDBMS's"
- Lecture: When databases and object-orientation collide: discussion of OODBMS and ORDBMS
- Lab: **HW #8 due**; Introduction to outer-joins; introduction to PL/SQL triggers; **HW #9 out**

Week 15: December 6, 8

- Reading: Reading Packet: "A few words on databases and ethics"
- Lecture: Discussion of databases and ethics
- Lab: **HW #9 due**; Review for Final Exam
- **Project Final milestone** due by **5:00 pm** on **Friday, December 10**

Final Exam:

MONDAY, December 13, 12:40 - 2:30 pm, in HGH 106 (unless I announce otherwise)