Course Syllabus – Fall 2024

Course Information

CS 325 - Database Design - Section 10 - CRN 41055 - Fall 2024

Lecture times and location: Monday, Wednesday - 1:00 - 2:20 pm - BSS 166

Lab times and locations:
Section 11: Friday - BSS 313, 1:00 - 2:50 pm
Section 12: Friday - BSS 315, 3:00 - 4:50 pm
Section 13: Friday - BSS 315, 9:00 - 10:50 am

Professor Contact Information

Dr. Dongcheng Li

Instructor's office: BSS 328

Instructor's e-mail: dl313@humboldt.edu

Instructor's phone: (217) 926-3588

Instructor's office hours: Monday 2:20-3:20 pm and Thursday 5:50-6:50 pm, or by appointment

---

Course Description

From the HSU catalog: Introduction to database design and implementation. Relational model, entity-relationship model and diagrams, converting a model to a schema, elementary Structured Query Language (SQL), normalization. This course is an introduction to relational database implementation and design; it will enable you to realize the potential of available powerful database management systems (DBMS) software by teaching you how to model and design a database that will serve as a firm foundation for database applications. You will also learn and extensively practice using the SQL, used as a data definition language to create and destroy database tables, as a data manipulation language to maintain and query the data within database tables, and as a data control language to protect and enhance the integrity of the data within database tables. You will learn the fundamentals of the relational model, entity-relationship modeling, converting an entity-relationship model into a database schema/design, normalization, implementation of a database schema/design, and some fundamentals of transactions and concurrency management.

Course Prerequisites

[CS 112] OR [GSP 270 and (CS 111 or CS 232 or GSP 318)] OR [instructor's consent]

Student Learning Outcomes

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

- Understand the basic components of a database management system (DBMS), and of a database design/schema.
- Understand a database model, create a database model for at least a simple scenario, and convert a database model into a corresponding database design/schema.
- Understand the concept of normalization, and normalize a given set of relations and functional dependencies to at least Third Normal Form (3NF).
- Use basic SQL fluently for defining, manipulating, and querying database tables, and be able to create, execute, and debug SQL scripts run within an enterprise-level multi-user DBMS.
- Understand the concept of a constraint on a database to increase data integrity; be familiar with entity integrity constraints (implemented using primary keys) and referential integrity constraints (implemented using foreign keys).
- Produce an implemented database project whose final report includes a description of the database' scenario, a database model, a corresponding database design/schema, and technical prose describing how this database prototype can now be used within this scenario.

CS Program Learning Outcomes that this course addresses

This course addresses departmental learning outcomes of:

- Computational Thinking
- Technical Writing
- Communicating and Collaborating

Cal Poly Humboldt Learning Outcomes that this course addresses

Cal Poly Humboldt graduates will have demonstrated:

- Effective communication through written and oral modes.
- Critical and creative thinking skills in acquiring a broad base of knowledge and applying it to complex issues.
- Competence in a major area of study.

Required Course Materials

- Links to online required readings will be provided via Canvas.

Course Software

We primarily will be using an **Oracle implementation of SQL** for this course. Unless noted otherwise, you are expected to use the on-campus student Oracle database for course assignments, 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 that you can access the HSU Oracle student database from on- or off-campus using nrs-projects.humboldt.edu.

Throughout the semester, you will be making some use of the Linux operating system. Note that you may access nrs-projects.humboldt.edu by using the programs ssh (secure shell) and sftp (secure ftp); we will walk through how this can be done during an early class lab session. Versions of these can be accessed from both vlab.humboldt.edu and vlinux.humboldt.edu.

Recommended Textbooks and Materials

- Fundamentals of Database Systems (7th Edition) by Ramez Elmasri, Shamkant B. Navathe.
- Database System Concepts (7th Edition) by Avi Silberschatz, Henry F. Korth, and S. Sudarshan

---

Grading Breakdown

If you are a Computer Science (CS) major, note that you must earn at least a C- in CS 325 for this course to count towards your major.

The grade each student will earn from this class will be based on a weighted score calculated by using the following table:

Quiz	15%
Exam 1	12.5%
Exam 2	12.5%
Final Exam	15%
Assignments	15%
Lab	15%
Project	15%

NO homework grades are dropped, ALL homework grades count toward your homework average. Every homework includes important practice of course fundamentals. No late work is accepted. The one (1) lowest quiz grade will be dropped from your total quiz grade, and the one (1) lowest lab grade will be dropped from your total lab grade.

In addition, the instructor may, at his discretion, issue a **non-passing semester grade** for the course if participation in **any** aspect of the class falls below a minimum of 40% of the total possible points

for that part of the class. For example, if missed homework assignments result in an average homework score of less than 40%, then a semester grade of D or F may be assigned, even if the overall grade average is above 70%. The purpose of this rule is to prohibit the practice of simply ignoring part of the course requirements with the thinking that the other parts will be enough to pass. Participation in all aspects of the course (lectures, labs, homeworks, and exams) must be maintained at acceptable levels in order to learn this material!

Grades will be assigned according to the scale using the following table:

Weighted Score	Grade
93.0 – 100	A
90.0 – 92.9	A-
87.0 – 89.9	B+
83.0 – 86.9	B
80.0 – 82.9	B-
77.0 – 79.9	C+
73.0 – 76.9	C
70.0 – 72.9	C-
67.0 – 69.9	D+
60.0 – 66.9	D
Below 60.0	F

Tentative Course Schedule:

The instructor reserves the right to modify this calendar as he deems necessary.

Week	Date	Topic
1	08/26	Class Introduction
1	08/28	Database processing and development; Intro to SQL
2	09/02	Holiday: Labor Day
2	09/04	Intro to SQL
3	09/09	Database processing and development; Intro to SQL
3	09/11	The relation model and relational operations; Writing relational operations in SQL
4	09/16	The relation model and relational operations; Writing relational operations in SQL
4	09/18	Entity-Relationship modeling; More where clause options and aggregate functions for SQL
5	09/23	Entity-Relationship modeling; More where clause options for SQL
5	09/25	Entity-Relationship modeling; More where clause options for SQL
6	09/30	Entity-Relationship modeling; Aggregate functions, Sub-selects, concatenating columns, and projecting literals
6	10/02	Entity-Relationship modeling; Aggregate functions, Sub-selects, concatenating columns, and projecting literals, Review for Exam I
7	10/07	Sub-selects, concatenating columns, projecting literals, and EXISTS predicate, Review for Exam I
7	10/09	Exam I
8	10/14	Intro to database security

8	10/16	Normalization of relations; Order by, group by, and having
9	10/21	Normalization of relations; Order by, group by, and having
9	10/23	Normalization of relations; Order by, group by, and having
10	10/28	Converting model into a design/schema; Set-theoretic operations, more on modifying data, intro to sequences
10	10/30	Converting model into a design/schema; Set-theoretic operations, more on modifying data, intro to sequences
11	11/04	Converting model into a design/schema; SQL ANSI join syntax; More join; Views
11	11/06	Converting model into a design/schema; SQL ANSI join syntax; More join; Views
12	11/11	Transactions and Concurrency; Simple reports
12	11/13	Transactions and Concurrency; Simple reports
13	11/18	Transactions and Concurrency; Simple reports; Review for Exam II
13	11/20	Exam II
14	11/25	FALL BREAK – No Classes
14	11/27	FALL BREAK – No Classes
15	12/02	Advanced SQL
15	12/04	Advanced SQL
16	12/09	Database Ethics
16	12/11	Project presentation and demo, Final Exam Review
17	12/18	Final Exam (12:40PM - 2:30PM)

Course Expectations

First: 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 12 hours a week spent outside of class for this 4-credit course.
However, you should be aware that:

- This is a junior-level CS major course; it has an accordingly-rigorous workload.
- You can only learn SQL, database modeling, database design, etc., by practicing them. Practicing as much as possible helps!
    - This can include playing around with in-class examples, experimenting to see if something you are curious about really works like you think, doing further research on topics of interest, and so on.
    - Think of a musical instrument -- you have to practice to master playing a guitar, violin, trumpet, drums, etc. You can't master it by just reading about the instrument. Think, also, of sports skills such as pitching, archery, etc. -- again, repetition and practice is required to hone such skills.
- As you likely know by this point, programming courses can be notorious time eaters. Occasionally, a problem with a SQL script will take large amounts of time to locate and fix (especially if you don't ask for help!).
    - 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?

- I cannot emphasize this enough: you need to start project parts early, to start homework assignments as soon as they are made available, to submit homework parts throughout the week, and to ask questions as you have them!
  - This is not a course that you can work on just one day a week, nor is it a course that you can take a week off from and expect to do well.
- The course will intensify as the semester progresses -- as you are able to do more, you will be expected to do more. Also, later concepts are built upon earlier concepts. 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 and project 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 them; likewise, they will typically not be extended because of hardware or network failure. (Admittedly, campus failures might affect deadlines. But don't assume so until you have heard from me definitively.)
  - 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, (if that assignment is a homework:) 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 assignment.

A successful student in this class will:

- Attend every lecture and lab/activity.
- Participate in class (asking questions, paying attention, taking notes, being an attentive team member when project meeting/discussion).
- Complete reading assignments and reading quizzes in a timely fashion.
- Practice example questions on the required textbook.
- Take the opportunity to learn how to write your own thoughts; don't plagiarize. Be sure to give credit where credit is due and cite your sources.
- Ask specific questions -- in class, in lab/activity, in office-hours, and in e-mail.
- Read through each homework assignment as soon as it is posted.
- Start working on each homework early in the week, and on each project milestone early in its stage.
- E-mail the instructor with specific homework-related questions starting early in the week both to clarify what a question is asking for and when hitting roadblocks (being sure to include BOTH the code involved AND any error messages or descriptions of bizarre behavior).
  - (Likewise for project milestones, although earlier milestones will involve concepts rather than code)
  - NOTE: please do NOT use Canvas messages to contact me or to ask me a question -- please send e-mail instead. Handling Canvas messages is time-consuming and error-prone on my end.
- Always submit SOMETHING for an assignment, even if it is not complete.
  - I believe in partial credit on homeworks, believing that if you have at least started working on a problem, the posted example solution will be more helpful/understandable than if you have not.

    - With regard to project milestones, I believe in partial credit because of the importance of getting started on pieces of long-term projects!
- Compare their homework solutions to posted example solutions if/when they become available.
- Study with others for exams, and practice explaining concepts to one another.
- Attempt every exam problem, and carefully study over exams when they are returned.
- Practice SQL, database modeling, database design, etc. as much as possible.

Inclusivity

Students in this class are encouraged to speak up and participate in-class. Each of us must show respect for each other because our class represents a diversity of beliefs, backgrounds, and experiences. I believe that this is what will enrich all of our experiences together. I recognize that our individual differences can deepen our understanding of one another and the world around us, rather than divide us. In this class, people of all ethnicities, genders and gender identities, religions, ages, sexual orientations, disabilities, socioeconomic backgrounds, regions, and nationalities are strongly encouraged to share their rich array of perspectives and experiences. If you feel your differences may in some way isolate you from our classroom community or if you have a specific need, please speak with me early in the semester so that we can work together to help you become an active and engaged member of our class and community. *(Adapted from HSU Canvas Accessible Syllabus Template, which was in turn adapted from CSU Chico and Winona State University)*

Thus, spoken language and body language should emanate respect for everyone in your classroom community. This includes coming to class on time and prepared to listen and share. (Adapted from Jayne McGuire's syllabi language)

Lab Exercises

- Graded lab exercises will be given during most lab sessions.
- If you miss a lab session, typically its graded lab exercise cannot be made up later (except for extenuating circumstances - please let me know!). However, the **one (1) lowest lab exercise grades** will be dropped from the lab exercise portion of your semester grade.
- You will typically be using **pair programming** for lab exercises -- in pair programming, two programmers work on the same problem together, but also discussing along the way.
    - **Both** are **actively** involved in the programming process.
    - **Engage** in continuous **discussion** about the problem and code, collaboratively **analyzing** and evaluating best approach to identify the solution
    - **Both** students need to **submit** their lab work.
    - This software engineering practice can result in programs with fewer errors, amongst other potential benefits.
    - While learning to program, this practice can also give you more chances to discuss course concepts with other students, (along with the practical benefit of reducing the total number

of questions the instructor has to try to answer during lab sessions, hopefully also reducing your wait time for those answers).

- Note: if, for example, there is an odd number of students at a particular lab, or there are technical difficulties, we'll also sometimes have trios -- in that case, **all three** are working on the problem together, and also discussing along the way.
- It is **not acceptable** to simply sit back during a lab exercise and have your partner(s) do all the work -- you are expected to **actively participate** in your pair/trio.
- Please let me know of any issues that come up related to pair programming.
- Again, once you have completed a lab session's lab exercise, made sure that all of your pair's/trio's members submitted your work to Canvas.

Academic Honesty

Students are responsible for knowing policy regarding academic honesty. For more information, visit:

https://www2.humboldt.edu/studentrights/academic-honesty

Observe that among the actions that are unacceptable are submitting another's program, code, or file as your own, giving your program, code, or file to another to submit as THEIR own, and failing to quote material (that includes algorithms, code, and comments, too!) taken from another person's 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 then each pair-programmed file turned in will include **both** of the names of the students who worked on it as a pair.)

(**If** an assignment does explicitly specify that it is acceptable to pair program or work in 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 explicitly specified as permitting 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, in general, students may not work together to complete homework assignments, one student should not instruct another in how to write the code for a homework assignment, and **any type of copying or modifying of another person's computer files, OR of providing computer files to another, related to homework assignments 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.)

Likewise -- notice that this means that it is NOT okay to post homework answers or code into an online discussion!

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 also are encouraged to ask 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 and complete the reading questions on-time, who attend lecture and lab regularly, who start homeworks and project milestones promptly after they are made available from the public course web site, 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 the 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. Is it right?")

CS 325 E-mail Guidelines

- As noted above, sending questions to me **by e-mail** also can be a very effective way to ask for help.

    - **NOTE:** please do **NOT** use Canvas messages to contact me or to ask me a question -- please send **e-mail** instead. Handling Canvas messages is time-consuming and error-prone on my end.
- Include CS 325 along with the subject of your e-mail in the Subject: line of any class-related e-mail that you send me. This will help your e-mail be more recognizable as a class-related message, and will make it less likely that I will accidentally overlook it.
    - ALSO include a **descriptive subject** along with the CS 325 in that Subject: line -- this also increases the chances that I will notice and reply to your question more promptly. (In particular, do **not** just reply to a class e-mail message I have sent previously, and do **not** simply leave the Subject: line blank!)
- You are expected to **sign** each e-mail you send me with **your name** -- sometimes the sender's identity is not obvious from one's e-mail address, especially for an off-campus e-mail address.
- Please take a few minutes to ensure that your message reflects a professional tone. I know I have sent an email or two in the heat of the moment that I soon regretted. Take your time and communicate professionally. (*Adapted from Jayne McGuire's syllabi language*)
- I try to check my e-mail (dl313@humboldt.edu) **about once a day on weekdays**, and **about once over each weekend**. This is another reason to start assignments early, so you have time to receive a reply to any questions that might arise.
- So, in general, if I have not replied to your e-mail within 24 hours, please **re-send** it, just in case I have overlooked it or some glitch occurred.
    - (And if there seems to be a chance that your message is getting chomped by a spam filter -- rare, but not unprecedented! -- leave me a message at 217-926-3588 with the Subject: line of the email you are trying to send and the e-mail address you are using, and I will see if I have indeed received it!)
- All HSU students are responsible for checking their HSU email account for official communications. You are expected to check for course-related messages as well. Course-related messages from me will include CS 325 in the Subject: line.
    - While students may elect to redirect messages sent to their official HSU email address to another address, those who redirect their email to another address do so at their own risk.
    - HSU Email Policy: https://policy.humboldt.edu/p16-01-email-policy

---

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).

Additional Course Policies

- You are expected to read this syllabus and be prepared to verify in a required Canvas activity that you have received it, have read it, and understand its contents.
- Exam dates are given in the course schedule above. No Make-up exams.

- You are expected to check the public course web page and the course Canvas 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 Canvas site. You are expected to monitor your posted grades and let me know about any discrepancies.
- When reading assignments are given, you are expected to prepare (read and study) assigned readings before class and to participate in class discussions. Some reading assignments will have reading quizzes that must be completed by a given deadline.

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.

- 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 and lab notes are not posted, although many of the projected examples will be made available on the public course web site.

Campus policies

The following URL leads to useful links regarding HSU policies, procedures, and resources:
https://academicprograms.humboldt.edu/content/syllabus-addendum
ALL of the policies linked from the above are applicable to this class, and you are expected to be familiar with these policies!
The following are just a FEW highlights from this site:

Students with Disabilities

Persons who wish to request disability-related accommodations should contact the **Student Disability Resource Center**, **826-4678 (voice)** or **sdrc@humboldt.edu**. Disability accommodations must be pre-approved by the Student Disability Resource Center.
You can reach the Student Disability Resource Center's web site at:
https://disability.humboldt.edu/
Please note that some accommodations may take up to several weeks to arrange. 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:
https://registrar.humboldt.edu/registration/help#help
You can find the University policies for repeating classes at:
https://registrar.humboldt.edu/forms#policies
NOTE THAT THE ADD/DROP DEADLINE IS:
***** SEPTEMBER 9, 2019 *****

...WHICH IS THE DEADLINE TO ADD OR DROP CLASSES WITHOUT A SERIOUS AND COMPELLING REASON. And, please note: it is the Registrar's Office that determines what constitutes a "serious and compelling reason".
If you do drop the course, note that it is your responsibility to complete and submit the appropriate forms.

Attendance and disruptive behavior

Students are responsible for knowing policy regarding attendance and disruptive behavior: https://www2.humboldt.edu/studentrights/attendance-behavior

- **Late arrival to class:** Please attempt to come to class on time, with your headphones/earbuds/etc. put away and your cell phones/tablets/pads/gadgets/etc. 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://risksafety.humboldt.edu/sites/default/files/risksafety/hsu_eop-g_9.3.pdf

...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