CSc 460 — Database Design
Spring 2026 (McCann)

http://www2.cs.arizona.edu/classes/cs460/spring26/

Homework #1

(120 points)

Due Date: February 19th, 2026, at the beginning of class

Directions

1. **This is an INDIVIDUAL assignment; do your own work! Submitting answers created by other people or generated by AIs (e.g., ChatGPT) are a few examples of NOT doing your own work.**
2. Write complete answers to each of the following questions, in accordance with the given directions. Create your solutions as a PDF document such that each question is on a separate page; all parts of a multi–part question may be on the same page. Show your work, when appropriate, for possible partial credit.
3. If you have questions about any aspect of this assignment, help is available from the class staff via piazza.com and our office hours.
4. When your answers are ready to be turned in, do so on gradescope.com. Be sure to assign pages to problems after you upload your PDF. Need help? Visit https://help.gradescope.com/ and search for "Submitting an Assignment."
5. Remember that you can use at most one late day on a homework assignment, because we will be distributing solutions after that time.
6. Start early! Getting help is much easier n days before the due date/time than it will be n hours before.

Connolly/Begg Chapter 1: Introduction to Databases

1. (5 points) Our textbook describes the "DreamHome" data management situation in Section 1.2 (pages 8–14). What are the pros and the cons of using a DBMS to store and manage the DreamHome data? Give three of each.

C/B Chapter 2: Database Environment

2. (5 points) We presented two types of data independence within the context of the ANSI/SPARC model. What are those types, and what do they each hope to achieve?

C/B Chapter 3: Database Architectures

3. (5 points) Why would a system architecture be designed as a multi–tier (a.k.a. n–tier) architecture, rather than a basic two–tier (Client–Server) architecture?

4. (5 points) We'll be using the Oracle DBMS (www.oracle.com) when we work with SQL and JDBC later in the semester. Find Oracle's documentation on–line, and see what you can learn about how Oracle supports (if it supports!) the ideas of a service–oriented architecture. Service–oriented architectures (SOAs) are covered in Section 3.2.2 and in the extra slides of Topic 2.

C/B Chapter 4: The Relational Model

5. (10 points) In Topic 4 we covered three different types of data integrity, and spent some time defining the (very simple) Bank database schema. Explain how each of those three types of data integrity apply to that schema. If any of them do not apply, explain why not. (Considering data integrity within the context of Chapter 4 of our text may help. That's where the authors introduce the ideas of data integrity.)

(Continued . . .)

6. (10 points) Our authors introduce a small Hotel database schema at the start of the exercises section at the end of Chapter 4, on page 118. Create some fake data for those four tables, at least four records per table, that follow the rules for referential integrity.

C/B Chapter 12: Entity–Relationship Modeling

7. (5 points) While presenting the three styles of E–R diagrams, we also presented the idea of multiplicity constraints on relationships. What do multiplicity constraints add to the information provided by an E–R diagram, by comparison to Chen's original relationship labeling?

8. (10 points) A company has hired you to create a database design. The company has four departments. Each department employs at least one employee, and each employee only works for a single department. Employees may or may not have dependents (e.g., children), but dependents can be associated with as many as two employees. For employees who have worked elsewhere, the company maintains an employment history for each of them.

   Draw an E–R model (your choice of any of the three we presented in class) for this database design and its restrictions. You may draw it by hand, or you may use any of the many diagramming tools that exist. (You may find that learning to use one of them is more time–consuming than drawing by hand!) If you do use a tool, tell us which one you used.

C/B Chapter 13: Enhanced Entity–Relationship Modeling

9. (5 points) What are the constraints that need to be followed when a specialization/generalization relationship is used?

Questions on Topics Not Well–Covered in the Textbook:

10. (15 points) Probabilities of disk failure. For all parts, assume an AFR probability of $p_f = 0.015$.

    (a) What is the probability of failure of a striped system with 6 total disks? Show your work.

    (b) When your boss hears this, he's appalled. "That's too high! Reconfigure the system to be mirrored instead of striped, using the same quantity of disks. I read in Busy–Body Business Boss Bulletin that doing so will reduce the probability of the whole system failing!" Will it or won't it? Explain your answer, showing the math so that even your boss will understand.

11. (10 points) We covered just four of the standard RAID levels in class. A popular non–standard RAID level is RAID 10 (a.k.a. RAID 1+0). Using reliable sources, learn the details of RAID 10, and write, in your own words, a description of no more than three paragraphs describing how it works, its best advantage, and its biggest disadvantage. Conclude with a list of the sources that you consulted.

12. (10 points) In Program #2, you linearly–hashed (lite–ly!) using key values expressed in ASCII. In class, I explained Extendible hashing using binary key values. For this exercise, you'll use extendible hashing with numeric keys expressed in Base 3. Assume that hash buckets are disk blocks that can hold at most three keys each. Build an extendible hashing index structure using the keys listed below, inserted in order from left to right, and draw the final structure. Be sure to include all global and local depth values. Feel free to draw intermediate results, too, for possible partial credit.

    210, 222, 010, 201, 002, 202, 212, 021, 012

13. (15 points) Assuming a B^+–tree of Order 3 (based on our in–class (Comer's) B–Tree definition, and storing keys to the left of their matching separators) that holds upper–case letters, insert the sequence of letters shown below, in order from left to right, into an initially–empty tree. Show the tree after each insertion that causes the quantity of tree nodes to grow, and show the final tree (if different from the most–recently–drawn tree).

    `A, O, E, U, I, D, H, T, N, S, Q, J, K, X, B, M, W, V, Z`

14. (10 points) Delete, from your final tree from the last problem, the following keys, in order, and show the resulting tree after each deletion: `T, U, A`. When a key being deleted has a copy in use as a separator, replace the separator copy, too.