CS 328 - Homework 2

Deadline

11:59 pm on **Friday, February 6, 2026**

Purpose

To consider further your choice for your "second" database (so that you can submit your chosen database's files as part of **Homework 3**), and to practice writing and (at-least-partially) validating more strict-style HTML.

How to submit

IF you are submitting your **"second" database files** now, submit those files using **`~st10/328submit`** on **nrs-projects**, with the special homework number of **`33`**.

Each time you wish to submit **any files for Problems 1-3**, submit those files using **`~st10/328submit`** on **nrs-projects**, with a homework number of **`2`**.

NEXT STEPS for your "second" database

The purpose of this section is to give you a heads-up about database files you will need to submit as part of the **next** homework, **Homework 3**. (But it also lets you know how can go ahead and submit them earlier, IF you wish.)

If you recall, Homework 1 - Problem 2 noted that you will be building application pieces atop **another**, **second** database in addition to the bookstore database, and you started considering what that database might be as part of Homework 1 - Problem 2's submission `second-db.txt`.

Scanning the responses submitted in your `second-db.txt` files for Homework 1 - Problem 2, there are indeed a number of you who would or might like to build atop a different second database that the one you built in CS 325.

Available Shared Databases for your "second" database

There are now **three** databases posted that previous CS 325/CS 328 students have given me permission to share. You can see them at the course Canvas site, under "Modules", scrolling down to the section "**Available Shared Databases**".

- If you do decide to use one of these, you will be required to leave the **original creator's name(s)** in those files.
  - You may modify your copies of them, adding **additional** comments of "modified by:" with your name and the date last modified when you do so.
- You will also be required to include a **"database designed by" credit** including their name(s) at the beginning of each application you build atop this database.

- (Understand that these have been shared **as-is**, so you might find errors that need to be fixed, or aspects you would like to change, etc.! Each time you do change them, you will be expected to submit the revised database file(s) along with the first homework that uses/depends on those revisions.)

You now have a chance to look over these, and consider if you would like to build atop one of these, or use your CS 325 database, or use another database (as long as you get my approval for it and create the files specified below **by Homework 3's deadline**).

When and How you will let me know your chosen "second" database

As part of **Homework 3**, everyone will be asked to submit the following for their selected "second" database:

- **NOTE**: I am considering writing scripts that search the homework submissions for files with names that end *precisely* as noted below, so if yours do not, you may lose some credit.
- Your selected database' **business rules** file, **in PDF format**, in a file whose name *ends* in **`biz-rules.pdf`**
- Your selected database' **ER model** file, **in PDF format**, in a file whose name ends in **`model.pdf`**
  - If you are creating a new database: note that this model should have at least five distinct entity classes, with at least four relationships, at least one of which is 1:N.
  - (Where does the "at least 7 relations" part come in, then? That is based on assuming some combination of N:M relationships, multi-valued attributes, and supertype/subtype entity classes in your model will result in at least two more relations when this model is converted to a design.)
- Your selected database' **design given in relation-structure form** (the form you used to express the bookstore database in Homework 1 - Problem 3's file `design-bks.txt`), **in plain-text format**, in a file whose name ends in **`design-rs.txt`**
- The **SQL script** implementing your selected database' **design**, in a file whose name ends in **`design.sql`**
  - The grader or I may need to create a version of your database at some point, so you may lose credit if this SQL script does not successfully create empty versions of your selected database's tables if we try it.
- The **SQL script** for an initial **population** of your selected database' design, in a file whose name ends in **`populate.sql`**
  - The grader or I may need to initially-populate a copy of your database at some point, so you may lose credit if this SQL script does not successfully initially-populate the tables created in your design script if we try it.

IF you ALREADY KNOW what "second" database you want to use and have the above files ready:

...it is fine for you to submit them now, using a special homework number of **33** so that I'll know you've chosen your "second" database and this is it.

- (By the way, it is perfectly fine if you improve any of these files later -- just re-submit them using the homework number of **33** and I'll then know that's an improved updated version of the files included.)

Homework 2 Requirements/Set-up

- For this homework's problems, do **not** include any CSS except for the external CSS `normalize.css` included in `html-template.html` and `html-style-play-before.html`
- If you wish to include an `img` element in the document for Problem 3, note that it needs to validate as strict-style HTML. If its URL does not do so, make a copy of the image in your nrs-projects account (if you can legally do so) or use a service such as such as tinyurl to avoid problematic characters.
- You are expected to put your HTML documents for Homework 2 into a **sub-directory** with permissions of **`711`** in your `public_html` directory on nrs-projects.

  In this case, I am asking **you** to **choose** the name for this sub-directory.

  ```
  cd ~/public_html             # make sure you are in your public_html
  mkdir name-you-choose        # make a directory within public_html
  chmod 711 name-you-choose    # make it world-executable
  cd name-you-choose           # go to that new subdirectory
  ```

  Remember that a world-readable file *my-doc.html* in the `public_html` subdirectory *name-you-choose* would have the URL:

  `https://nrs-projects.humboldt.edu/~`*your_user_name*/*name-you-choose*/*my-doc.html*

  - (Note: it is also perfectly fine if you choose to put your Homework 2 files in a "deeper" sub-directory within `public_html`.)

Problem 1

FUN FACT: There is an HTML element **`cite`** that should be used for a creative work such as the title of a book or the name of a newspaper.

- **`cite` can** have content.
- **`cite`** is an **inline** element; it should be within an appropriate "containing" element, for example `p` or `li`.
- According to the Web Hypertext Application Technology Working Group (WHATWG) (as reported at https://developer.mozilla.org/en-US/docs/Web/HTML/Element/cite), **`cite`** should ***not*** include author name(s) within its content.

To practice thinking about *structuring* content using HTML, consider the following desired *content*:

```
Recent Noteworthy Children's Picture Books

These are the top five children's picture books from the New York
Times Best Sellers for February 8, 2026. These were obtained from the
```

```
list at https://www.nytimes.com/books/best-sellers/ on January 29,
2026.

Top Five Children's Picture Books

1. Little Blue Truck's Valentine, by Alice Schertle and Jill
McElmurry
2. For the Fans!, by Angela Song
3. How to Catch a Loveosaurus, by Alice Walstead and Andy Elkerton
4. Grumpy Monkey Valentine Gross-out, by Suzanne Lang and Max Lang
5. Chicka Chicka I Love You, by Bill Martin Jr., John Archambault,
and Julien Chung
```

Starting from the **`html-template.html`** posted on the course public site and along with this homework handout, create a "strict-style" HTML document meeting the following requirements:

- Include **`prob1`** somewhere in its file name, and give its file name the suffix **`.html`** .
- Fill in the opening comment block as specified, putting in your name, the last modified date, and the URL that can be used to run your document.
- Include **your last name** along with appropriately-descriptive content in its **`title`** element.
- Use HTML to include and structure the above given desired content appropriately.
- Include HTML to make the URL that appears within this desired content a **hyperlink** that leads to that URL.
- Remember: some of our course style standards are not things that validators check. BUT, you can check if your result follows at least some of the strict-style HTML syntax as you did in the Week 2 Lab Exercise:
  - On nrs-projects, in the directory where your ***your-file-name*-`prob1.html`** resides, run:

    `cp` *your-file-name*-`prob1.html` *your-file-name*-`prob1.xhtml`
  - Paste the URL of your copy of ***your-file-name*-`prob1.xhtml`** into the Address field at https://validator.w3.org/nu or https://html5.validator.nu/ to check it.

Submit your ***your-file-name*-`prob1.html`** and ***your-file-name*-`prob1.xhtml`** files.

Problem 2

Along with this handout, you'll find the content for a file **`html-style-play-before.html`**. Currently, however, this content is **NOT** valid strict-style HTML.

- Use this content as the initial content of a file that includes **`prob2`** somewhere in its file name, and give its file name the suffix **`.html`** .
- Fill in the opening comment block as specified, putting in your name, the last modified date, and the URL that can be used to run your improved document.
- **Add** your last name to the current content in its **`title`** element.

- **Correct** this content to make it follow HTML strict-style and CS 328 course style standards.
- REMEMBER that, while it does not check for all of the CS 328 required style standards, you should use an **`.xhtml`** copy of your document with https://validator.w3.org/nu or https://html5.validator.nu/ to assist you in correcting this, and you should submit this `.xhtml` copy of your file.

Submit your ***your-file-name*-`prob2.html`** and ***your-file-name*-`prob2.xhtml`** files.

Problem 3

Consider the small bookstore database created and initially-populated by **`create-bks.sql`** and **`pop-bks.sql`**, which you described in relation-structure form in **`design-bks.txt`** as part of Homework 1.

You get to choose the "**theme**" along with a **name** for the bookstore represented by your instance of the bookstore database, and will use these as you build application pieces atop this database during the semester. You should not feel limited in any way by the titles that **`pop-bks.sql`** happens to insert. (That is, you could choose a literary theme, or a technical theme, or a fantasy theme, or a musical theme, or a game-based theme, etc.)

Decide on a bookstore theme and a name, and, using the posted **`html-template.html`** as the initial basis, create a strict-style HTML document **`about-bks.html`**. This needs to contain at least the following:

- Fill in the opening comment block as specified, putting in your name, the last modified date, and the URL that can be used to run your document.
- In its **`head`** element, include a **`title`** element with appropriate content.
- In its **`body`** element, include at least:
  - A name for your bookstore, included in a top-level/level-1 heading.
  - A paragraph describing your bookstore, included in a **`p`** element.
  - Within its **`footer`** element, include a **`p`** element whose contents include your name.
  - You may include additional appropriate content, in appropriate strict-style HTML elements, as you would like.
- REMEMBER that, while it does not check for all of the CS 328 required style standards, you should use an **`.xhtml`** copy of your document with https://validator.w3.org/nu or https://html5.validator.nu/ to assist you in validating and checking over your document, and you should submit this `.xhtml` copy of your file.

Submit your **`about-bks.html`** and **`about-bks.xhtml`** files.