CS 328 - Homework 4

Deadline
11:59 pm on Friday, February 21, 2014

How to submit
Submit your files for this homework using ~st10/328submit on nrs-projects, with a homework number of 4

Purpose
To practice some more with "strict" HTML5 and to practice with CSS3.

Important notes
• Note: I hope to have you present some of these HTML5 pages to the class at some point.
• Remember to follow the CS 328 SQL and PL/SQL Style Standards as given in the CS 328 Homework 1 and Homework 2 handouts for all SQL and PL/SQL code.
• Make sure that you have executed the scripts create-bks.sql and pop-bks.sql, and that the bookstore tables are successfully created and populated.
• Unless explicitly indicated otherwise, for the entire semester, all web pages submitted are expected to use "strict" style HTML5, as discussed in class and in the course textbook.
• Likewise, unless explicitly indicated otherwise, all web pages submitted are expected to include the links to the W3C experimental HTML5 validator and the CSS3 validator as well as the link to http://validator.nu/ as shown in example page html5-template.html, and all must validate/pass the tests from all 3. Each page that does not could cause a loss of points on the problem involved.
• I'm not requiring specific indentation for HTML5 yet - I reserve the right to do so, however, if necessary. In the meantime, find a readable way of indenting it, and consistently do so...
• However, for CSS rules, you are expected to indent the contents of all { }'s by at least 3 spaces, and each { and } should be on its own line, lined up with the beginning of the selector (as seen in posted class examples).
• Unless explicitly indicated otherwise, you are expected to use external CSS style sheets, and not internal or inline CSS style sheets.
• HTML5 forms should not use the table element for layout purposes -- CSS should be used for such layout, instead. The table element should only be used for truly-tabular data (as in Problem 4 below).
Problem 1

Create a SQL script `328hw4.sql`, and start it off with comments including your name, CS 328 – Homework 4, and the last-modified date.

Include a SQL*Plus spool command to spool the results of running this SQL script to a file named `328hw4-out.txt`.

Write a prompt or prompts that indicates that what is to follow is the current status for order number 11009.

Now write a query or queries (your choice) that provide the following information: for order number 11009, what are:

- the date this order was placed
- the publisher for this order
- the book title(s) in this order
- the quantity received to-date for each book in this order

Follow this/these with a spool off command; the resulting files `328hw4.sql` and `328hw4-out.txt` should now be ready to submit.

You'll be using the results of this query in one of the problems below.

Problem 2

Consider the query-or-queries you wrote for Problem 1. What if you would like this information for any order number entered (rather than always for order 11009)? What might an HTML5 form look like that could submit a desired order number to the application tier, so that it could build the appropriate query/queries and send the resulting query/queries to the data tier?

Create an HTML5 page `req-order-status.html` containing such a form, meeting the following specifications:

- Carefully consider what kind of HTML5 form control would be most appropriate for allowing the user to indicate the order number for the order in which he/she is interested.
  - Hint: You would hard-code parts of this at this point that might be able to be dynamically generated later in the semester.
- This page also needs to allow the user to enter his/her Oracle username and password.
  - For the password entry, use a password field instead of a textfield.
- It should include your name, CS 328, and the name of your bookstore from `bks-splash.html`.
- Your form's action can be the URL of any functioning web page -- later, we'll replace this with a URL that will actually attempt to process this form, building and requesting that the database server perform the appropriate query.
- Your form's method should be "post" (although while you are debugging you can use "get", as long as you replace it with "post" for the version that you submit).
• Add a link from your index.html on nrs-projects to your resulting req-order-status.html
  
  – Note: In index.html's unordered list, if you would like to convert this into an unordered list
  with unordered sub-lists for Homework 3’s links and Homework 4's links, that would be fine
  indeed.

Your resulting req-order-status.html file is not quite ready to submit yet (unless you want to
submit a "partial" version early on).

Problem 3

Consider Homework 3, Problem 1’s PL/SQL stored procedure insert_order_needed. What
might an HTML5 form look like that could submit the information needed by this procedure to the
application tier, so that the application tier could call this stored procedure?

Create an HTML5 page insert-o-needed.html containing such a form that meets the following
specifications:

• Carefully consider what kind of HTML5 form control would be most appropriate for allowing the
user to indicate each "piece" of necessary information expected by this stored procedure.
  
  – Again, it is fine if you hard-code parts of this at this point that might be able to be dynamically
  generated later in the semester.

• This page also needs to allow the user to enter their Oracle username and password. For the
password entry, use a password field rather than a textfield.

• It should include your name, CS 328, and the name of your bookstore from bks-splash.html.

• Your form's action can be the URL of any functioning web page -- later, we'll replace this with a
URL that will actually attempt to process this form, building a call to the PL/SQL stored procedure
insert_order_needed.

• Your form's method should be "post" (although while you are debugging you can use "get", as
long as you replace it with "post" for the version that you submit).

• Add a link from your index.html on nrs-projects to your resulting insert-o-needed.html

Your resulting insert-o-needed.html file is not quite ready to submit yet (unless you want to
submit a "partial" version early on).

Problem 4

To get a little bit of practice with the HTML5 table element (although in a hard-coded, rather than a
generated, way...)

Consider Problems 1 and 2 above (the query or queries that gave information about the order 11009,
and the HTML5 page req-order-status.html).

Later, we'll generate resulting HTML5 based on a query or queries like these generated by submitting a
form such as req-order-status.html above. Right now, though, I want to JUST use this
scenario for HTML5 practice.
Consider the result of the Problem 1 query/queries for the order number 11009.

- "Hard code" the information that results from that PARTICULAR query into an attractive HTML5 page (whose name is order-info.html)
- Make appropriate use of at least one HTML5 table element. (This is tabular data, after all.)
  - (Make sure this is clear: you are NOT actually querying the database from this example -- you are actually typing/pasting in the data resulting from Problem 1's query within an HTML5 page and adding HTML5 table tags, as HTML5 practice involving an HTML5 table.)
- Add a link from your index.html on nrs-projects to your resulting order-info.html

Your resulting order-info.html file is not quite ready to submit yet (unless you want to submit a "partial" version early on).

**Problem 5**

Consider a rather typical plain HTML5 form, in 328hw4-before.html, adapted from: http://srv13.fountainheadcollege.com/mustafa.eminoglu/ws201/registration.html

Copy this example into a file 328hw4-after.html and modify it as you like with an external style sheet named hw4.css. Meet the following specifications:

- Include a comment including at least your name and the last-modified date in hw4.css.
- Add your name to 328hw4-after.html in some visible fashion of your choice.
- Include at least five distinct rules in your external style sheet that have a visible effect on this page (but you can certainly add more if you are inspired!).
  - Remember that your course textbook describes many interesting properties in Chapters 3 and 4.
- Add a link from your index.html on nrs-projects to your resulting 328hw4-after.html

Your resulting files 328hw4-after.html and hw4.css are now ready to submit.

**Problem 6**

Consider your HTML5 page bks-splash.html from Homework 3, Problem 5.

Make a new copy of this page in a different directory, since you will be modifying it and you don't want to change Homework 3's version of this (since that could affect your Homework 3 grade!)

Add a link from your index.html on nrs-projects to this new version of bks-splash.html (making clear somehow this is Homework 4's version, and NOT removing the earlier link to Homework 3's version!!)

Now consider this new version of bks-splash.html, as well as req-order-status.html, insert-o-needed.html, and order-info.html from the problems above.

- Design an external style sheet bks.css that you would like to use for these and other pages we create making use of the database created by create-bks.sql.
  - Include a comment including at least your name and the last-modified date in bks.css.
– Include at least five distinct rules in your external style sheet that have a visible effect on these pages (but you can certainly add more if you are inspired!).
– Include rules to nicely lay out the forms on these pages.
• Modify these four HTML5 pages to use this style sheet.

NOW these resulting modified HTML5 pages are ready to submit, as is bks.css.