CS 328 - Homework 3

Deadline

Due by 11:59 pm on **Sunday, February 12, 2017**.

Purpose

To practice some more with PL/SQL stored procedures and functions, to practice with "strict"-style HTML5 , and to set up some HTML documents for future modification and use.

How to submit

Submit your files for this homework using `~st10/328submit` on nrs-projects, with a homework number of `3`.

Important notes

- Note: you *may* be presenting versions of some of these HTML5 pages to the class at some point.
- The **CS 328 SQL and PL/SQL Style Standards so far** as given in the CS 328 Homework 2 handout are now also posted on the public course web site,under **References** -- remember to follow these for all course SQL and PL/SQL code.
- There are now also **CS 328 HTML5 Coding Standards so far** posted on the public course web site, under **References** -- you are also expected to follow these for all course HTML documents.
  - Unless explicitly indicated otherwise, for the entire semester, all web pages submitted are expected to use "strict" style HTML5, as discussed in class, in the course textbook, and in these posted coding standards.
- 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.

Problem 1

Create a file `328hw3.sql`.

(Make sure that you have a copy of `pop-bks.sql` in the same directory as `328hw3.sql` -- it is called in this problem's testing script, to make sure the tests are run on "fresh", original versions of these tables, before the tests muck with them. Note that this script happens to already end with a `commit;` command.)

Start your `328hw3.sql` file with the following:

- comments containing at least your name, `CS 328 - Homework 3`, and the last-modified date
- include the command to `set serveroutput on`
- followed by a SQL*Plus `spool` command to spool the results of running this SQL script to a file named `328hw3-out.txt`
- followed by a `prompt` command including your name

Be sure to `spool off` at the end of this script (after your statements for the remaining problems).

Then, write a SQL*Plus `prompt` command that says `problem 1`. (You may add additional `prompt` commands around this to make it more visible, if you would like.)

Consider: sometimes, you just would like a nice unique next-primary-key for a table (and for the sake of some more PL/SQL practice, assume you have conveniently forgotten about the existence of sequences...!).

For example, consider the `order_needed` table. We're going to find that rows are added to this table when sales reduce a book's quantity below the order point -- so, it might be handy to have a function return the next suitable key value for the `order_needed` table's primary key, `ord_needed_id`.

Design and write a small PL/SQL stored function `next_ord_needed_id`; it doesn't need any parameters, and it should simply return a value that is one more than the largest current value of `ord_needed_id`. (BUT: what if the table is ever empty when this is called? Then this function should return a key of `1`.)

Separately (outside of `328hw3.sql`) run the posted testing script `next_ord_needed_id_test.sql` posted along with this homework handout, make sure the tests all pass, and submit the file it spools with the test results, `next_ord_needed_id_test_out.txt`, along with your files for this homework.

You may also add additional testing calls along with your `next_ord_needed_id` code in your `328hw3.sql` script if you would like.

Problem 2

In `328hw3.sql`, write a SQL*Plus `prompt` command that says `problem 2`. (Again, you may add additional `prompt` commands around this to make it more visible, if you would like.)

Then, design and write a PL/SQL stored function `is_on_order` that takes as its parameter an ISBN, and returns boolean `true` if that ISBN is currently on order (if its `on_order` attribute has the value `'T'`) and returns boolean `false` otherwise. (Note that we specifically want a return type of boolean!)

For exception-handling practice, within this function, you are required to use an exception section to handle the exception of the ISBN not being in the title table (let the system raise this `NO_DATA_FOUND` exception; your function should merely be able to handle it.) Have the function return boolean `false` in this case. (That is, if `is_on_order` is called with an ISBN not in the title table, then the function should simply return `false` -- it cannot be on order if it isn't a title -- rather than fail with an exception error message.)

Separately (outside of `328hw3.sql`) run the posted testing script `is_on_order_test.sql` posted along with this homework handout, make sure the tests all pass, and submit the file it spools with the test results, `is_on_order_test_out.txt`, along with your files for this homework. (Note that this script includes a useful little helper-function, `bool_to_string`.)

You may also add additional testing calls along with your `is_on_order` code in your `328hw3.sql` script if you would like.

Make sure that you submit your `328hw3.sql` and `328hw3-out.txt` files as well as the `*_test_out.txt` files for each of the subroutines in Problems 1 and 2.

Problem 3

You created an initial `index.html` file as part of the Week 3 Lab Exercise -- but it was in your nrs-project's `public_html` subdirectory `328lab03`.

Now, within your nrs-project's `public_html` directory itself, create **another** `index.html` that meets the following minimum requirements:

- (**except** for what is included in the `html5-template.html`, **do not include any additional CSS**, internal or external, **yet** -- you'll add that for **future** course work.)
- Start with the posted `html5-template.html` as the initial basis for your `index.html` file.
- As you add elements to its `body` element, do so **before** the `<hr />` right before the first HTML5 validator link.
  - (that is, you should have a horizontal rule/line **between** the content that you add to the page and the required validator links at the bottom of the page.)
  - Do **not** remove these validator links.
- Give this page's `title` element appropriate content.
- Add an appropriate `h1` element.
- Add an un-numbered list with links to *some* (NOT all!) of your resulting HTML5 documents you will be creating for the problems below.
- Include your name visibly somewhere IN the page.
- You may add **additional** elements as long as the above requirements are met and your page validates as "strict"-style HTML5 (using at least the 2nd HTML5 validator given at the bottom of the page).
- Make sure your resulting `index.html` can be reached from:

  `http://nrs-projects.humboldt.edu/~`*your-user-name*
- BUT -- **ALSO** submit your resulting `index.html` file.

Problem 4

Along with this handout, you'll find the content for a file `hw3-warmup-before.html`. Currently, however, this content is NOT valid "strict" HTML5.

- Use this content as the initial content of a file `hw3-warmup-after.html`
- Add a level-2 heading (**after** the first heading element in the page) that indicates that you modified this page (including your name)
- Correct this content to make it valid "strict" HTML5 (that successfully validates as HTML5 using at least the 2nd HTML5 validator noted at the bottom of the page).
- **Submit** your resulting `hw3-warmup-after.html` file, then **"hide"** it by changing its permissions to make it **not** world-readable: `chmod 600 hw3-warmup-after.html`

  (do **NOT** add a link to this file from your `index.html`!!!)

Problem 5

Consider the database created and populated using `create-bks.sql` and `pop-bks.sql`. You are going to gradually add some database application pieces atop this database over the course of the semester.

As a starting point for future work, using the posted `html5-template.html` as the initial basis, create an

opening "strict" HTML5 "title page" `bks-start.html` that includes at least:

- your name (included visibly somewhere IN the page)
- CS 328 (included visibly somewhere IN the page)
- a name of your choice for this bookstore (you get to select the bookstore's "theme", you see).
- an image of your choice (fitting that theme).
- (note that, probably as part of Homework 4, you will be adding a login form to this page. But we need to discuss forms in class first!)
- (**except** for what is included in the `html5-template.html`, **do not include any additional CSS**, internal or external, **yet** -- you'll add that for **future** course work.)
- Make sure your page successfully validates as HTML5 using at least the 2nd HTML5 validator noted at the bottom of the page.
- Add a link from Problem 3's `index.html` to your resulting `bks-start.html`
- ALSO submit your resulting `bks-start.html` file.

Problem 6

Now consider "your" database, that you selected in Homework 1 and wrote a PL/SQL subroutine for in Homework 2. You are likewise going to gradually add some database application pieces atop this database over the course of the semester, also.

As a starting point for future work, using the posted `html5-template.html` as the initial basis, create an opening "strict" HTML5 "title page" `custom-start.html` that includes at least:

- your name (included visibly somewhere IN the page)
- CS 328 (included visibly somewhere IN the page)
- an appropriate name for your scenario
- an image of your choice (fitting your scenario).
- (note that, probably as part of Homework 4, you also will be adding a login form to this page.)
- (**except** for what is included in the `html5-template.html`, **do not include any additional CSS**, internal or external, **yet** -- you'll add that for **future** course work.)
- Make sure your page successfully validates as HTML5 using at least the 2nd HTML5 validator noted at the bottom of the page.
- Add a link from Problem 3's `index.html` to your resulting `custom-start.html`
- ALSO submit your resulting `custom-start.html` file.