CIS 318 - Homework 3

Deadline:
Due by 11:59 pm on Wednesday, February 15

How to submit:
Submit your files for this homework using ~st10/318submit on nrs-projects, with a homework number of 3

Purpose:
To give you an opportunity for more PL/SQL practice, more XHTML practice (including practice with XHTML forms and tables), and some CSS practice.

Important notes:
• Remember to follow the CIS 318 SQL and PL/SQL Style Standards as given in the CIS 318 Homework 1 handout 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 XHTML pages must include the image-link to the W3C XHTML 1.1 validator as shown in example pages xhtml-template.html and xhtml-css-template.html, and all must validate as valid XHTML 1.1. Each page that does not will cause a loss of points on the homework problem involved.
  – if a page also uses CSS, it must include the image-link to W3C CSS validator as shown in example page xhtml-css-template.html, and it must validate as valid CSS level 2.1. Each page that does not will cause a loss of points on the homework problem involved.

• I'm not requiring specific indentation for XHTML 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 styles, you are expected to indent the contents of all { }’s by at least 3 spaces, and each { and } should be on its own line (as seen in posted class examples).

The Problems:

Problem 1
Create a SQL script 318hw3.sql, and start it off with comments including your name, CIS 318 - Homework 3, and the last-modified date.

Next, add the command to run the pop-bks.sql script each time this script is run, so that you have
"fresh", original versions of these tables. (Their contents are mucked with below, so it is important that these are "reset" here.)

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 `318hw3-out.txt`. 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.)

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 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

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

**Problem 2**

In `318hw3.sql`, write a SQL*Plus prompt command that says `problem 2`.

Consider the `order_needed` table in the bookstore database. The idea/hope here is that, when a title's quantity on hand becomes less than the `order_point`, then a row should be added to the `order_needed` table indicating that, well, an order is needed for that title. Initially, the `date_placed` attribute for that new `order_needed` row is NULL -- the order is needed, but it has not yet been placed.

When, later, the order IS placed, then the `date_placed` attribute for the corresponding `order_needed` row is filled accordingly. (I hope to have you write a trigger on an upcoming homework that will take care of this.)

At any given time, then, the rows in `order_needed` that have a NULL value for `date_placed` indicate orders that, well, need to be made. One could think of these as "pending" `order_needed` rows.

There could be times (indeed, as part of an application you will be working on later this semester) where you would like to know if, for a given ISBN, there is a "pending" `order_needed` row for that ISBN (if there is a row with that ISBN whose `date_placed` attribute is NULL).

Write this little PL/SQL stored function `pending_order_needed` that expects an ISBN, and returns whether or not that ISBN has such a "pending" `order_needed` row.

Also write a small tester PL/SQL stored procedure named `test_pending` -- it takes an ISBN as its only parameter, and it calls function `pending_order_needed` to see if that ISBN currently has a "pending" `order_needed` row. It should print a message to the screen that says whether that ISBN currently does or not (and it should include the particular ISBN in the message).
Follow your function and procedure with the following testing code:

```sql
prompt **********************
prompt demo pending_order_needed
prompt **********************
prompt

update title
set qty_on_hand = qty_on_hand - 5
where isbn = '0805322272';

exec insert_order_needed('0805322272', 50)

prompt SHOULD be TRUE:
exec test_pending('0805322272')

prompt SHOULD be FALSE (its order_needed not pending):
exec test_pending('025602796X')

prompt SHOULD be FALSE (it's a title, but not in order_needed at all):
exec test_pending('0131103628')

prompt SHOULD be FALSE (not a title):
exec test_pending('1111111111')

prompt undoing temporary testing changes
rollback;
```

You may add additional testing calls if you would like.

Follow all of this with spool off and rollback commands; the resulting files 318hw3.sql and 318hw3-out.txt should now be ready to submit.

**Problem 3**

Consider your XHTML page `bks-splash.html` from CIS 318 Homework 2, Problem 4.

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 2's version of this (since that could affect your Homework 2 grade!)

For this problem, make the following changes to this new copy of this page (in its new directory):

- remove the link to your file `hw2-warmup-after.html`
- modify the XHTML comment containing the URL I can use to view your `bks-splash.html` from your nrs-projects account to reflect this copy's new directory. (Note that, for full credit, this URL must successfully display Homework 3's version of this page when I paste it into a browser.)
- add a link (with appropriate descriptive text) leading to `order-status.html` (to be created in
Problem 4 below)

• add a link (with appropriate descriptive text) leading to insert-o-needed.html (to be created in Problem 5 below)

Your resulting bks-splash.html file is not quite ready to submit yet...

Problem 4

Now 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 XHTML 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 to the database tier?

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

• carefully consider what kind of XHTML form component would be most appropriate for allowing the user to indicate the order number for the order in which he/she is interested.

• 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: this is an input tag with a type attribute value of "password".
  – (Its possible attributes and behavior and look is very like a textfield, except what the user types is obscured, so that someone looking over a user's shoulder can't read something sensitive, such as a password, being typed into it. There is an example of one of these in posted in-class example try-textarea-reset-password.html)

• it should include your name, CIS 318, 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)

• include an XHTML comment containing the URL I can use to view your order-status.html from your nrs-projects account. (Note that, for full credit, this URL must successfully display your page when I paste it into a browser.)

• include a link (with appropriate descriptive text) leading back to Problem 3's bks-splash.html

Your resulting order-status.html file is not quite ready to submit yet...

Problem 5

Consider CIS 318 Homework 2, Problem 2's PL/SQL stored procedure insert_order_needed. What might an XHTML form look like that could submit the information needed by this procedure to the application tier, so that it could call this stored procedure?
Create an XHTML page `insert-o-needed.html` containing such a form that meets the following specifications:

- carefully consider what kind of XHTML form component would be most appropriate for allowing the user to indicate each "piece" of necessary information expected by this stored procedure
- 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, CIS 318, 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 calling `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)
- include an XHTML comment containing the URL I can use to view your `insert-o-needed.html` from your nrs-projects account. (Note that, for full credit, this URL must successfully display your page when I paste it into a browser.)
- include a link (with appropriate descriptive text) leading back to Problem 3's `bks-splash.html`

Your resulting `insert-o-needed.html` file is not quite ready to submit yet...

**Problem 6**

Consider Problem 1 and Problem 4 above (the query or queries that gave information about the order 11009, and the XHTML page `order-status.html`).

Later, we'll generate resulting XHTML based on a query or queries like these generated by submitting a form such as `order-status.html` above. Right now, I want to just use this scenario for XHTML practice.

Consider the result of the Problem 1 query for the order number 11009.

- "Hard code" the information that results from that PARTICULAR query into an attractive XHTML page (whose name is `order-info.html`)
- make appropriate use of at least one XHTML table
  - (Make sure this is clear: you are NOT actually querying the database from this example -- you are actually typing in the data resulting from Problem 1's query within an XHTML page, as XHTML practice involving an XHTML table.)
- include an XHTML comment containing the URL I can use to view your `order-info.html` from your nrs-projects account. (Note that, for full credit, this URL must successfully display your page when I paste it into a browser.)

Your resulting `order-info.html` file is not quite ready to submit yet...
Problem 7

Consider a rather typical plain XHTML form, in 318hw03-before.html, adapted from: http://srv13.fountainheadcollege.com/mustafa.eminoglu/ws201/registration.html

Copy this example into a file 318hw03-after.html and modify it as you like with an external style sheet named hw3.css. Be sure to:

- include a comment including at least your name and the last-modified date in hw3.css
- add your name to 318hw03-after.html in some visible fashion of your choice
- include at least five distinct
  selector
  {    
      property: value
  }
  ...styles in your external style sheet that have a visible effect on this page (but you can certainly add more if you are inspired!);

- include an XHTML comment containing the URL I can use to view your 318hw05-after.html from your nrs-projects account. (Note that, for full credit, this URL must successfully display your page when I paste it into a browser.)

Your resulting files 318hw05-after.html and hw3.css are now ready to submit.

(Note: it is possible that your answers to this problem may be posted from the public course Moodle site.)

Problem 8

Design/create a new XHTML page of your choice (it MUST be separate from your XHTML page for Problem 7!):

- whose file name includes prob8 somewhere within it
- uses your hw3.css file from Problem 7 as its external style sheet
- also defines an internal style sheet that visibly changes at least one of those styles for an element used at least THREE times in this page
- also uses an inline style sheet that changes the styles for EXACTLY ONE instance of that element (from the previous bullet) that is used at least THREE times in this page
- includes a form that includes:
  - at least 2 radio button instances (that together make a logical group)
  - at least 2 checkbox instances
  - at least one select/drop-down box with at least 4 options to choose from
  - a text area
– (the "theme" for your form is your choice)

• include an XHTML comment containing the URL I can use to view your file from your nrs-projects account. (Note that, for full credit, this URL must successfully display your page when I paste it into a browser.)

Your resulting XHTML page is now ready to submit.

(Note: it is possible that your answers to this problem may be posted from the public course Moodle site.)

**Problem 9**

Consider bks-splash.html, 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, and modify these XHTML pages to use this style sheet. NOW these resulting modified XHTML pages are ready to submit, as is bks.css

(Note: it is possible that your answers to this problem may be posted from the public course Moodle site.)