CS 328 - Homework 9

Deadline

11:59 pm on **Sunday, April 7, 2024**

Purpose

To get ready to commit to a "second" database choice as part of the next homework, and to get more practice with PHP that sends requests to Oracle using OCI, including using it to build a dynamic form widget and using bind variables to help thwart SQL injection.

How to submit

Each time you wish to submit, submit your files using `~st10/328submit` on nrs-projects, with a homework number of `9`.

Important notes

- **NOTE**: you are welcome to use `require_once/include_once/require/include` in your PHP documents!
  - BUT when you use them in homework problems' documents, be sure to also SUBMIT copies of all files that you are requiring/including!
- **DO NOT USE ANY PHP FRAMEWORKS FOR THESE PROBLEMS** -- one of this course's purposes is to provide you with practice with "plain" PHP.
- Remember: You are required to use **`normalize.css`** for all of your web pages for CS 328, and to add **`link`** elements for additional CSS external stylesheets *after* this (but still **within** the `head` element).

  EXCEPT for `normalize.css`, **DO NOT USE ANY CSS FRAMEWORKS or PREDEFINED LIBRARIES for this course** (unless you get prior, explicit permission). One of this course's purposes is to provide you with some practice with the basics of "plain" CSS, so you can better make use of frameworks and predefined libraries later.
- Remember: There are now **CS 328 PHP Coding Standards so far** posted on the public course web site, under References -- you are also expected to follow these for all course PHP documents.
- You are expected to follow all course coding standards posted on the public course web site; course documents are also expected to validate as "strict"-style HTML, and valid CSS.
- 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.

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 10. (But it also lets you know how can go ahead and submit them earlier, if you wish.)

If you recall, back in Homework 1 - Problem 2, it was mentioned that you will be building application pieces atop **another**, **second** database in addition to the bookstore database.

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 build 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", in 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 if 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 10's deadline).

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

As part of **Homework 10**, everyone will be asked to submit the following for their selected "second" database:

- **NOTE**: I am considering writing scripts that search the homework submission 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`**
- 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.)

So you can start considering potential uses for your "second" database:

Note that, in remaining homeworks:

- You will be required to decide on and write a **PL/SQL stored procedure** or **PL/SQL stored function** for your selected "second" database.
- You will be creating PHP documents that create strict-style HTML documents styled by CSS and that use your selected "second" database.
  - You will definitely be asked to create a PHP document that **queries** something useful that results in **multiple rows** and displays it tastefully within an HTML **`table`** element.
  - You will definitely be asked to create a PHP document that **calls** your **PL/SQL stored procedure** or **PL/SQL stored function**.
  - You will definitely be asked to create a PHP **postback** document whose initial response is a **form** that includes form widget(s) (such as a **`select`**/drop-down element) **dynamically** built from a **query** of your database, and whose response when that form is submitted uses its data with the help of **bind variables** to query something requested by the user.
- You may also be asked to practice additional topics discussed during the remainder of the semester using your selected "second" database.

Problem 1 - more practice with PHP and OCI

Note that you can make a copy of **`hum_conn_no_login.php`**, that contains the helper function **`hum_conn_no_login`**, in your current nrs-projects directory with this command:

```
cp ~st10/hum_conn_no_login.php .  # remember the space and period!
```

Consider the **bookstore** database, and think of a query that:

- projects **at least 3 columns**, at least one of which is **numeric**, and at least one of which is **non-numeric**, and

- selects **at least 3 rows**.

Now consider the posted **328lect10-2-query-empls.php** from the Week 10 Lecture 2 posted examples.

Using this as a basis, create your own PHP document that includes **hw9-prob1** in its file name that meets the following requirements:

- Include your name and last modified date in its opening comment, **AND** the URL this can be run from
  - (You *will* lose some credit if this URL does not work when I or the grader paste it into a browser!)
- Include an appropriate **title** element.
- Use **require_once** to include the definition of function **hum_conn_no_login**, from your local copy of file **hum_conn_no_login.php**, in its **head** element.
- Include an appropriate **h1** element that somehow includes "your" bookstore's name (from previous homeworks' about-bks.html and bks-start.html)
- Query at least one of the tables **created by create-bks.sql**,
  - making appropriate use of the function **hum_conn_no_login**,
  - such that you **project at least 3 columns**, at least one of which is **numeric**, and at least one of which is **non-numeric**, and
  - such that you **select at least 3 rows**.
- Display the queried results using a **table** element.
- Include a **hypertext** link with appropriate text that links back to your ***hw9-prob1*.php**.
- Include **your last name** within a **p** element that you add to the CS-328-standard **footer** element (the one from html-template.html).
- It is up to you whether you would like to add additional CSS styling to this using an external CSS -- BUT if you do, also submit the .css file used.

Submit your file ***hw9-prob1*.php** (and any files besides hum_conn_no_login.php that it uses, if any).

Problem 2 - more practice building dynamic select/drop-down widgets and using bind variables

In the Week 10 Lab Exercise, you built a PHP postback document that:

- creates a form with a select drop-down widget dynamically built based on a query's results, and
- crafts a response to that form when it is submitted with the help of a dynamic select statement that uses bind variables rather than concatenation (to help thwart SQL injection).

Consider:

The bookstore has several kinds of data that would be more convenient for a user to be able to select than to type in -- for example, publishers, book titles, or author last names.

Decide on information a user might like to ask about such data -- as just a *few* of the possible examples:

- What is the city and state of a particular publisher?
- What is the minimum order for a discount, and the discount they give, for a particular publisher?
- What are the titles and their authors that are published by a particular publisher?
- What is the price and currently quantity of a particular title?
- Which of the bookstore's titles are by a particular author, and what publisher publishes each?

Create a PHP postback document that includes **`hw9-prob2`** in its file name that **both** creates a form **and** crafts a response to that form when it is submitted, that also meets the following requirements:

- Include your name and last modified date in its opening comment, **AND** the URL this can be run from
  - (You *will* lose some credit if this URL does not work when I or the grader paste it into a browser!)
- Include an appropriate **`title`** element.
- Use **`require_once`** to include the definition of function **`hum_conn_no_login`**, from your local copy of file **`hum_conn_no_login.php`**, in its **`head`** element.
- Include an appropriate **`h1`** element that somehow includes "your" bookstore's name (from previous homeworks' `about-bks.html` and `bks-start.html`)

Form requirements

- Its initial **`form`** element should have **`method="post"`** and an **`action`** attribute whose action is:

  `"<?= htmlentities($_SERVER["PHP_SELF"], ENT_QUOTES) ?>"`
- It should include a `label` logically-connected to a **`select`**/drop-down widget with **`option`** elements whose contents are **not** hard-coded, but are queried from the database.
  - For example, you could build this drop-down to contain publisher names, or titles, or order numbers.
- It may contain additional form widgets if you would like.
- It should include a submit button.

Response requirements

When its form is submitted using `method="post"`, your PHP should:

- Appropriately **sanitize** each piece of information submitted by this form.
- Build a dynamic `select` statement that projects **at least two attributes**, and uses at least one **bind variable** (instead of concatenation!) in its **`where`** clause based on the user's selection from the form's **`select`**/drop-down widget.
- Add the `select`'s results to the response in a pleasing, strict-HTML-style way.

- It may include more (based on additional optional form widgets) if you would like.

Note that you will lose substantial credit if you use concatenation to include any submitted information within your `select` statement string -- you are **required** to use bind variable(s) instead!

After your form or response:

- Include a **hypertext** link with appropriate text that links back to your **`*hw9-prob2*.php`**.
- Include **your last name** within a **`p`** element that you add to the CS-328-standard **`footer`** element (the one from `html-template.html`).
- It is up to you whether you would like to add additional CSS styling to this using an external CSS -- BUT if you do, also submit the `.css` file used.

Submit your file **`*hw9-prob2*.php`** (and any files besides `hum_conn_no_login.php` that it uses, if any).