

## CS 328 - Homework 9

### Deadline

11:59 pm on Friday, April 10, 2026

### Purpose

To get more practice writing PHP documents that send requests to Oracle using OCI, to read and think about SQL injection, and to practice building a postback PHP that builds a dynamic form widget.

### How to submit

Each time you wish to submit your work-so-far, 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 (including documents generated by your PHP files) 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.

And when you re-run these, you should also re-run:

- **Homework 3's `new-titles.sql`**, to re-add your additional titles
- **Homework 5's `328hw5.sql`**, to re-create its stored subroutines
- **Homework 6's `328hw6.sql`**, to re-create its stored function.

## Problem 1 - bookstore-related practice with PHP and OCI

**Purpose:** to get more practice using `require_once` to include a PHP function from another file, and using OCI from PHP to request data from Oracle.

**Task:** Starting from the CS 328 `html-template.html`, create a PHP document `328hw9-1.php` that meets the following requirements.

First, recall that we "bundled" the steps needed for setting up a no-login connection between PHP running on your nrs-projects account and Humboldt's Oracle `student` database into a PHP function `hum_conn_no_login`, in a file named `hum_conn_no_login.php`.

Make a copy of this file in your directory for this problem:

```
cp ~st10/hum_conn_no_login.php . # DON'T FORGET the SPACE & PERIOD!!
```

Then, consider the `bookstore` database, and think of a query that:

- projects **at least 3 different columns**, and
- selects **at least 3 different rows**.

Then, in your `328hw9-1.php`:

- 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!)
- Within its **head** element, give the **title** element appropriate descriptive content.
- Within its **head** element: include a regular PHP tag or tags:
  - to enable PHP error reporting
  - to include the definition of function `hum_conn_no_login`, from your local copy of file `hum_conn_no_login.php`
- Within its **head** element, *after* the **link** element styling this document with `normalize.css`, add a second **link** element so that this will be *further* styled using an external CSS stylesheet `328hw9-1.css` that you are about to create for this problem.
  - **Do not include any inline or internal CSS rules in your `328hw9-1.php`.**
- Within its **body** element, include an **h1** element that somehow includes "your" bookstore's name (from previous homeworks' `about-bks.html`).
- Within its **footer** element near the end of the **body** element, add a **p** element whose content includes your name.
- Query at least one of the tables **created by `create-bks.sql`**,
  - such that you **project at least 3 different columns**, and
  - such that you **select at least different 3 rows**.
- Display the queried results using an appropriate **table** element.

- Include an **a**/anchor element (hypertext link) with appropriate text that links back to your **328hw9-1.php**.

Your **328hw9-1.css** should meet the following requirements:

- Include a comment including at least its file name, your name, and the last-modified date.
- If you change any of the default foreground and background colors, make sure that, for any text atop a background, the **contrast** between their colors is **at least WCAG 2 AA Compliant** based on the tester at: [https://snook.ca/technical/colour\\_contrast/colour.html](https://snook.ca/technical/colour_contrast/colour.html).
- Include at least one rule adding an attractive **border** to the **table**, **td**, and **th** elements.
  - You also should make appropriate use of the **border-collapse** property to prevent a doubled-border in the resulting displayed table.
- Add additional CSS rules as desired to further attractively format elements of this document.
- Make sure your resulting **328hw9-1.css** validates as valid CSS.

Strict-validate your **328hw9-1.php**'s result by running it from a browser, viewing its source, copying and pasting that source into a file named **328hw9-1.xhtml**, and put the URL of your **328hw9-1.xhtml** into the validator.

Submit your resulting files:

- **328hw9-1.php** (and all additional files it uses, if any),
- **hum\_conn\_no\_login.php**,
- **328hw9-1.xhtml**, and
- **328hw9-1.css**

## Problem 2 - your-"second"-db-related practice with PHP and OCI

**Purpose:** to get more practice using **require\_once** to include a PHP function from another file, and using OCI from PHP to request data from Oracle, in this case making use of your "second" database.

**Task:** Starting from the CS 328 **html-template.html**, create a PHP document **328hw9-2.php** that meets the following requirements.

*If* your files for this application are in a different directory than those for Problem 1, then make a copy of the **hum\_conn\_no\_login** function in this problem's directory:

```
cp ~st10/hum_conn_no_login.php . # DON'T FORGET the SPACE & PERIOD!!
```

Also, make a **COPY** of your file **custom.css** from **Homework 7 - Problem 3** in this problem's directory on nrs-projects (so you will not interfere with your earlier homework's files!).

Then, consider your "second" database, and think of a query that:

- projects **at least 3 different columns**, and
- selects **at least 3 different rows**.

Then, in your **328hw9-2.php**:

- 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!)
- Within its **head** element, give the **title** element appropriate descriptive content.
- Within its **head** element: include a regular PHP tag or tags:
  - to enable PHP error reporting
  - to include the definition of function **hum\_conn\_no\_login**, from your local copy of file **hum\_conn\_no\_login.php**
- Within its **head** element, *after* the **link** element styling this document with **normalize.css**, add a second **link** element so that this will be *further* styled using your external CSS stylesheet **custom.css**.
  - **Do not include any inline or internal CSS rules in your 328hw9-2.php.**
- Within its **body** element, include an appropriate, descriptive **h1** element.
- Within its **footer** element near the end of the **body** element, add a **p** element whose content includes your name.
- Query at least one of the tables of your "second" database:
  - such that you **project at least 3 different columns**, and
  - such that you **select at least different 3 rows**.
- Display the queried results using an appropriate **table** element.
- Include an **a**/anchor element (hypertext link) with appropriate text that links back to your **328hw9-2.php**.

Your Homework 9 version of **custom.css** should meet the following requirements:

- It should continue to meet Homework 7 - Problem 3's requirements, and **also** the **additional** requirement below.
- Include at least one rule adding an attractive **border** to the **table**, **td**, and **th** elements.
  - You also should make appropriate use of the **border-collapse** property to prevent a doubled-border in the resulting displayed table.
- Make sure the resulting **custom.css** still validates as valid CSS.

Strict-validate your **328hw9-2.php**'s result by running it from a browser, viewing its source, copying and pasting that source into a file named **328hw9-2.xhtml**, and put the URL of your **328hw9-2.xhtml** into the validator.

Submit your resulting files:

- **328hw9-2.php** (and all additional files it uses, if any),
- **hum\_conn\_no\_login.php**,
- **328hw9-2.xhtml**, and
- **custom.css**

## Problem 3 - thinking about SQL injection

**Purpose:** to learn more about SQL Injection and some of the means for thwarting it.

**SQL injection** is when someone tries to "inject" additional SQL clauses into a **dynamic** SQL statement -- one built at run time, especially one built based on user input.

**Task:** To add more depth to your knowledge of SQL injection, **read** the two articles:

- **"SQL Injection Attacks by Example"** and
- **"How security flaws work: SQL injection"**

...that are posted along with this homework handout.

As you read these articles:

- pay special attention to the **"Mitigations"** section near the end of the first article, and
- note that an example of the second article's prepared statements are statements using OCI's bind variables that we will be discussing in class starting in Week 11.

Then, in a **plain-text file 328hw9-3.txt**:

- include your name
- list **at least THREE** important "take-aways" from these articles, and **for each**, explain **WHY** you chose it. (So, there are **SIX PARTS** to this -- your **three** selections, **AND WHY** you chose each.)

Submit your file **328hw9-3.txt**.

## Problem 4 - practice building a dynamic **select**/drop-down widget

**Purpose:** to practice using a query's results to build a dynamic form widget, and to continue to practice building PHP postback documents.

Consider:

- In-class example **328lect10-2.php** dynamically builds a **p** (paragraph) element with queried employee information.
- Your Week 10 Lab Exercise **328lab10.php** dynamically builds a **table** element with queried employee-dept-and-or-customer information

And, consider a **select**/drop-down box form widget. This, too, could be dynamically built using a query's results!

Then, you could build a drop-down box form widget without hard-coding the options, and that could always reflect the most up-to-date information from a database whenever a user uses that PHP document.

**Task:** For this problem, you are going to build a **PHP postback document** that either:

- creates a **form** with a **select** drop-down widget **dynamically** built based on a query's results, or
- crafts a response to that form carefully/safely noting which option the user selected from the submitted form.

In **Homework 4, Problem 4**, in the file **bks-isbn-choice.html**, you included a form allowing a user to select a title's ISBN, including just a few hard-coded titles/ISBNs.

And, in **Homework 7, Problem 2**, you added to your **bks.css** to style a possibly-modified version of **bks-isbn-choice.html**.

Now you will create a PHP document **328hw9-4.php** that REPLACES the hand-written/hard-coded **select**/drop-down form widget in **bks-isbn-choice.html** with a *dynamically-generated* **select**/drop-down box form element!

Create a PHP document **328hw9-4.php** that meets the following requirements:

- If your files for this application are in a different directory than those for Problems 1 or 2, then make a copy of the `hum_conn_no_login` function in this problem's directory:  

```
cp ~st10/hum_conn_no_login.php . # DON'T FORGET the SPACE & PERIOD!!
```
- Also, make a **COPY** of your file **bks.css** from **Homework 7 - Problem 2** in this problem's directory on nrs-projects (so you will not interfere with your earlier homework's files!).
- Make sure **328hw9-4.php** includes 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!)
- **Within** its **head** element, edit the **title** element, giving it appropriate content.
- Within its **head** element: include a regular PHP tag or tags:
  - to enable PHP error reporting
  - to include the definition of function `hum_conn_no_login`, from your local copy of file `hum_conn_no_login.php`
- **Within** its **head** element, add a second **link** element so that this will be further styled using Homework 9's version of **bks.css**.
  - **Do not include any inline or internal CSS rules in your `bks-isbn-choice.php`.**
- Within its **body** element, include an **h1** element with appropriate content that somehow includes "your" bookstore's name (from previous homeworks' **about-bks.html**).
- Within its **footer** element near the end of the **body** element, add a **p** element whose content includes your name.
- Your **328hw9-4.php**'s logic should be designed so that its response includes **either** a form, **or** a response to its form -- its response should never include both.
- The initial **form** element generated by your **328hw9-4.php** should be a **variation** of your **form** from Homework 7 - Problem 2's **bks-isbn-choice.html**, meeting the following requirements:
  - its **action** attribute should have as its action:  

```
"<?= htmlentities($_SERVER["PHP_SELF"], ENT_QUOTES) ?>"
```
  - instead of being hard-coded, the **option** elements of ISBN-title pairs in its **select**/drop-down **must** be **dynamically built** based on **querying** for the current ISBNs and titles of books in the

bookstore databases' **title** table.

- AS in Homework 4 - Problem 4:
  - Set up this **select**'s **option** elements so that the user **sees**, in the **select**/drop-down box, ISBN-title name pairs, **but...**
  - ...when a particular option is selected, the **value** in the resulting name=value pair for this option is **JUST** the ISBN for the user's choice.
  - (for example, if the user selects an option that is displayed as:  
9780131103627 - The C Programming Language  
... the form will submit a name=value pair whose value is just 9780131103627 )
  - **ASK ME** if you are not sure what I am asking for here.
- When this form is submitted, the response generated by your **328hw9-4.php** should include the following:
  - Appropriately **sanitize** the ISBN submitted by this form.
  - Create a **p** (paragraph) element whose content includes the (appropriately sanitized) ISBN the user selected.
  - The response should **ALSO** include an **a**/anchor element (hypertext link) with appropriate text that links back to your **328hw9-4.php**.

Your Homework 9 version of **bks.css** should meet the following requirements:

- Actually, it is fine if you decide that you do not need to make any changes to this, as long as it meets Homework 7 - Problem 2's requirements.
- But, whether it is changed or not, make sure the resulting **bks.css** still validates as valid CSS.

Strict-validate **BOTH** parts generated by your **328hw9-4.php** as you did for the Week 9 Lab Exercise's 328lab09.php:

- Put your **328hw9-4.php**'s URL in a browser and **view its source**, **copy and paste** that source into a file named **328hw9-4-1.xhtml**, and put the URL of your **328hw9-4-1.xhtml** into the validator.
- Put your **328hw9-4.php**'s URL in a browser **and fill out and submit its form**, *then view that response's source*, and **copy and paste** that *response's source* into a file named **328hw9-4-2.xhtml**, and put the URL of your **328hw9-4-2.xhtml** into the validator.

Submit your resulting files:

- **328hw9-4.php** (and all additional files it uses, if any),
- **hum\_conn\_no\_login.php**,
- **328hw9-4-1.xhtml** and **328hw9-4-2.xhtml**, and
- **bks.css**.