

CS 328 - Week 11 Lab Exercise - 2026-04-09

Deadline

Due by the end of lab.

Purpose

To practice using OCI from PHP to request data from Oracle, including using it to build a dynamic form widget and using a bind variable to help thwart SQL injection, and also writing some CSS as a little Exam 2 review.

How to submit

Submit your files using `~st10/328submit` on nrs-projects, each time entering a lab number of **91**.

Requirements

- You are required to work in **pairs** for this lab exercise.
 - This means **two** people working at **ONE** computer, one typing ("driving"), one saying what to type ("navigating"),
while **BOTH** are looking at the **shared** computer screen and **discussing** concepts/issues along the way.
- Make sure **BOTH** of your names appear in each file submitted.
- Make a `public_html` **sub-directory** for today's lab exercise with a **not-too-obvious name** of your choice, and be sure to **chmod** it to permissions **711**.
- And, your **.php** and **.css** files for this lab exercise should all have permissions **644**.
- When you are done, before you leave lab, **BOTH** of you should submit appropriate versions of these files using `~st10/328submit` on nrs-projects, with a lab number of **91**.

Set up your files for this lab exercise

- Use the course HTML template to start up a PHP document **328lab11.php**:

```
cp ~st10/html-template.html 328lab11.php
```
- **Copy** this file into the **same** directory as your file `328lab11.php`:

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

Requirements for 328lab11.php

In your file **328lab11.php**:

- Fill in the opening comment block, putting in your **names**, the last modified **date**, and the **URL** that can be used to run your document.
 - (You *will* lose some credit if this URL does not work when I or the grader paste it into a browser!)
- **Within** the **head** element, edit the **title** element, giving it appropriate content.
- **Within** the **head** element: include PHP tag(s) with statement(s):
 - 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 the head element**, after the **link** element specifying that this document is being styled by **normalize.css**, add a second **link** element following course style standards specifying that this document will then be further styled using a file **328lab11.css** that you are about to create for this lab exercise.
 - **Do not include any inline or internal CSS rules in your 328lab11.php.**
- Within the **body** element, include an **h1** element with appropriate content.
- **Within the footer element** near the end of the **body** element, add a **p** element whose content includes your names.

You are now going to add to this so that **328lab11.php** will be a postback document, so that it **either** creates a form **or** crafts a response to that form when it is submitted.

What form?

When your PHP is initially called, it should set up a query to Oracle asking for department numbers and department names, ordering them by department name. It should use them to create a **form** element containing at least:

- a **label** logically connected to...
- ...a **select**/drop-down widget, with **option** elements:
 - ...whose **contents** are those just-queried **department names**
 - ...whose **value** attributes are those just-queried department numbers
- a submit button (an **input** element with **type="submit"**)

What form response?

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

- appropriately **sanitize** the department number submitted by this form
- use it as the value for a **bind variable** within a **select** statement querying for at least the **employee last names** and **job titles** for employees who work in the selected department, ordering them by employee last name.
 - You may choose to project *additional* information for each employee in the selected department if you would like.
 - **Note** that you will lose **substantial** credit if you use concatenation to include the submitted department number within your **select** statement string -- you are **required** to use a **bind variable** instead!
- display these within a **table** element that meets class style standards.
- Include an **a**/anchor element (hypertext link) with appropriate text that links back to your **328lab11.php**., but using an **href** attribute value of:

```
"<?= htmlentities($_SERVER["PHP_SELF"], ENT_QUOTES) ?>"
```

(so that this links back to the CALLING PHP, just in case the navigator does change the URL in the opening

comment to THEIR copy...!)

What goes in 328lab11.css?

Your **328lab11.css** should meet the following requirements:

- Include a comment including at least your **file's** name, *your names*, 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.
- *If* you did *not* happen to include a `fieldset` element within your **form**, then include rule(s) adding a visible, attractive border to this **form**.
- Include rule(s) that will result in the **form** *not* taking up the entire width of the body.
- Include rule(s) that will result in the **submit button** being **centered** within the **form**.
- 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 rules *if* desired to further attractively format elements of this document.
- Make sure your resulting **328lab11.css** validates as valid CSS.

Strict-validate both possible responses

Strict-validate the two parts generated by your **328lab11.php** as you did for the Week 9 Lab Exercise's 328lab09.php:

- Put your **328lab11.php**'s URL in a browser and **view its source**, **copy and paste** that **source** into a file named **328lab11-1.xhtml**, and put the URL of your **328lab11-1.xhtml** into the validator.
- Put your **328lab11.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 **328lab11-2.xhtml**, and put the URL of your **328lab11-2.xhtml** into the validator.

BEFORE you leave lab:

Make sure that you **both** have copies of the files:

- **328lab11.php** (and all additional files it uses, if any)
- **328lab11-1.xhtml** and **328lab11-2.xhtml**
- **328lab11.css**

...and you BOTH submit these using **~st10/328submit** on nrs-projects, with a lab number of **91**.