

CS 328 - Week 12 Lab Exercise - 2026-04-16

Deadline

Due by the end of lab.

Purpose

To practice using OCI from PHP to request that Oracle call a PL/SQL stored procedure and a PL/SQL stored function, and to also practice calling `oci_commit` to request that a current Oracle database state be committed.

How to submit

Submit your files using `~st10/328submit` on nrs-projects, each time entering a lab number of **92**.

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` file for this lab exercise should have permissions **644**.
- When you are done, before you leave lab, **BOTH** of you should submit appropriate versions of this lab's files using `~st10/328submit` on nrs-projects, with a lab number of **92**.

Set up for this lab exercise

- Since this PHP will muck with the contents of `empl` (and possibly `customer`), you should have a copy of `set-up-ex-tb1s.sql` handy to recreate/restore these tables as needed. You can get a copy of it for today's lab use with the command:

```
cp ~st10/set-up-ex-tb1s.sql . # DON'T FORGET the SPACE and PERIOD!!
```

- Consider a PL/SQL stored **function** `count_empl`, that expects an employee's last name, and returns the number of employees with that last name. You can get a copy of it for today's lab use with the command:

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

- (and remember to **run** this script in **sqlplus** to **create** this stored function in your database!)

- Also consider a PL/SQL stored **procedure** `terminate_empl` -- it expects an employee's last name, and tries to have the side effects of:

- setting to `NULL` the `mgr` field of any employee managed by the employee to be terminated
- setting to `NULL` the `empl_rep` field of any customer whose employee rep is the employee to be

terminated

- then deleting that employee's row from **empl**

It does first verify that there is **only one** employee with that name (using stored function **count_empl!**), and does **nothing** if there are **none** or **more than 1**.

You can get a copy of it for today's lab use with the command:

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

- (and remember to **run** this script in **sqlplus** to **create** this stored procedure in your database!)

- And, for convenience, make a local copy of the file containing the **hum_conn_no_login** function definition:

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

Requirements for 328lab12.php

- Use the course HTML template to start up a PHP document **328lab12.php**:

```
cp ~st10/html-template.html 328lab12.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**
- **Do not include any inline or internal CSS rules in your 328lab12.php.**
 - (If you are sufficiently annoyed by the default formatting, you may **optionally** add a **328lab12.css** file to further format your **328lab12.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 **328lab12.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?

To concentrate for today's lab on calling a PL/SQL stored function and PL/SQL stored procedure from PHP using OCI, we'll keep this form simple. When your PHP is initially called, it should create a **form** element containing at least:

- a **label** logically connected to...
- ...a textfield asking the user to enter the last name of an employee to terminate

- a submit button (an **input** element with **type="submit"**)

Optional stretch goal

If you would like: instead have this part of your PHP set up a query to Oracle asking for employee last names, ordered by employee last names, and replace the textfield above with a **select**/drop-down widget with **option** elements whose contents and **value** attributes are those just-queried employee last names.

What form response?

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

- appropriately **sanitize** the employee last name submitted by this form
- use the sanitized employee last name as the argument to PL/SQL function **count_empl**
 - Remember to call **oci_free_statement** to free the statement object used to call **count_empl** when you are done.
- IF that **count_empl** call returns 1:
 - your PHP should then call PL/SQL procedure **terminate_empl** to terminate that employee,
 - output a **p** element with content saying that that employee has been terminated (**including** their last name in that paragraph's content)
 - then, as this is the end of this logical transaction, it should call **oci_commit** to commit this change.
 - Remember to call **oci_free_statement** to free the statement object used to call **terminate_empl** when you are done.
- ELSE, IF that **count_empl** call returns 0:
 - your PHP should just output a **p** element with content saying that there are no employees with that last name (**including** that non-existent last name in that paragraph's content)
- ELSE, IF that **count_empl** call returns more than 1:
 - your PHP should just output a **p** element with content **including** the number of employees with that last name (and **including** that last name in that paragraph's content), and noting that as a result it did not know which to terminate and so none were terminated.
- **NOTE** that you will lose **substantial** credit if you use concatenation to include the submitted employee last name within your PL/SQL subroutine call strings -- you are **required** to use a **bind variable** instead!
- Remember to call **oci_close** to **close** the connection object used when you are done.
- For more-convenient re-use, include an **a**/anchor element (hypertext link) with appropriate text that links back to your **328lab12.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...!)

Strict-validate both possible responses

Strict-validate the two parts generated by your **328lab12.php** as you did for the Week 9 Lab Exercise's **328lab09.php**:

- Put your **328lab12.php**'s URL in a browser and **view its source**, **copy and paste** that **source** into a file named **328lab12-1.xhtml**, and put the URL of your **328lab12-1.xhtml** into the validator.
- Put your **328lab12.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 **328lab12-2.xhtml**, and put the URL of your **328lab12-2.xhtml** into the validator.

BEFORE you leave lab:

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

- **328lab12.php** (and all additional files it uses, if any)
- **328lab12-1.xhtml** and **328lab12-2.xhtml**
- (If you created the optional **328lab12.css**, submit it, also.)

...and you BOTH submit these using **~st10/328submit** on nrs-projects, with a lab number of **92**.