CS 328 - Week 11 Lab Exercise - 2024-04-05

Deadline

Due by the end of lab.

Purpose

To practice using PHP with OCI to call a PL/SQL stored procedure and a PL/SQL stored function.

How to submit

Submit your files for this lab 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** issues along the way.
- Make sure **BOTH** of your names appear in each file submitted.
- When you are done, before you leave lab, somehow e-mail or copy the lab exercise files so that **BOTH** of you have copies, and **BOTH** of you should submit these files using `~st10/328submit` on nrs-projects, with a lab number of **91**.

The Set-up

Since this PHP will muck with the contents of `empl` (and possibly `customer`), you should have a copy of `set-up-ex-tbls.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-tbls.sql .  # DON'T FORGET the SPACE and PERIOD!!
```

Consider a 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!)

Also consider a 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, 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!)

Your Tasks

Create a PHP document named **`328lab11.php`** that is a postback PHP, that **both** creates a form **and** crafts a response to that form when it is submitted (as demonstrated in the posted Week 9 Lecture 2 example **`php-form-handling.php`**).

- Fill in its opening comment block with the URL I can use to run your version, your name(s), and the last modified date.
  - It will be a penalty if your document does not display using this URL when I/the grader try it.
- Include an appropriate **`title`** element.
- You are welcome and encouraged to use the function **`hum_conn_no_login`** in `328lab11.php`!
  - You can get a copy of this AND the **`328footer-plus-end.html`** that it uses in your current directory on nrs-projects using:

    ```
    cp ~st10/hum_conn_no_login.php .  # remember the space and dot!
    cp ~st10/328footer-plus-end.html . # remember the space and dot!
    ```

  - Be sure to use **`require_once`** to include this in your **`head`** element, and then you can call `hum_conn_no_login` as desired in your `body` element.
- Include an appropriate **`h1`** element.
- Include **your last name** within a **`p`** element that you add to the CS-328-standard **`footer`** element (the one from `html-template.html`).

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 contain a **form** 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

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, 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 provided PL/SQL procedure **`terminate_empl`** to terminate that employee,
  - output a **`p`** element with content saying that that employee has been terminated (and include 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 the non-existent last name),

  ELSE, IF that `count_empl` call returns more than 1, your PHP should just output a `p` element with content saying that there are that many employees with that last name (including the last name), and that as a result it did not know which to terminate.
- Remember to call `oci_close` to close the connection object used when you are done.
- For more-convenient re-use, include a **hypertext** link with appropriate text that links back to your **`328lab11.php`**.

Reminder: How can one strict-validate the HTML resulting from a PHP?

Beware -- if you put the URL of this PHP directly into the validator at https://html5.validator.nu/, it looks like it does regular-HTML validation (not strict-validation) of just the response containing its form. This is not a bad start, but it is not strict-validation, nor does it check your PHP's response with a submitted form's values.

Here is one approach I have successfully used to strict-validate a multi-document PHP such as `328lab10.php`:

- Put your `.php`'s URL in a browser and view its source, copy and paste that source into a file with suffix `.xhtml`, (for example, `328lab10-1.xhtml`) and put the URL of that `.xhtml` document into the validator.
- Put your `.php`'s URL in a browser and submit its form, then view that *response*'s source, and copy and paste that *response*'s source into a file with suffix `.xhtml`, (for example, `328lab10-2.xhtml`) and put the URL of that `.xhtml` document into the validator.

Optional additions

- You may add an external CSS formatting your PHP document if you would like; if you do so, also submit that `.css` file.

Submit your resulting `328lab11.php` (and all additional files it uses, if any) with a lab number of **91**.

BEFORE you leave lab:

Make sure that you **both** have copies of the file `328lab11.php` (and all additional files it uses, if any), and you BOTH submit these using `~st10/328submit` on nrs-projects, with a lab number of **91**.

How the navigator can get today's lab files:

- Because this happen to be a file the nrs-projects web server has to be able to reach, the navigator should be able to get a copy of these files from the driver using an approach like this:
  - Assume the driver has username `ab12`, and the navigator has username `yz89`.
  - Also assume the driver created, in their `public_html` directory, a sub-directory named `328lab10`, and this sub-directory contains the files to be submitted for the lab exercise.
    - The NAVIGATOR `yz89` can now:
      - log in to THEIR nrs-projects account

```
cd public_html
mkdir 328lab11   # or other name they choose
chmod 711 328lab11
cp ~ab12/public_html/328lab10/328lab11.php . # note space & dot!
```

      - And now the navigator `yz89` has their own copy of this file.
- I will leave it up to the navigator to decide if they would like to UPDATE their `l328lab11.php` so their opening comment includes the URL to *their* copy, or if they want to leave the URL for the driver's copy.
  - HOWEVER: remember that you *will* lose some credit if this URL does not work when I or the grader paste your submitted file's URL into a browser, in either case.