An Introduction to Running a PHP Program with Embedded SQL

This is a hands-on exercise, and you will be awarded marks for completing the tutorial. Your submission will include a few screenshots of your work to show that you have successfully completed the tutorial. Canvas lets you upload multiple files, but you're welcome to put your results together into one file and submit that.

- **Late submission policy:** 25% penalty per day late

Introduction

From the previous two tutorials, you are familiar with using Java/JDBC and Oracle, as well as Oracle's SQL*Plus on the undergraduate servers. Again, it is not entirely necessary that you have completed the two tutorials before this tutorial.

In this tutorial, we will be using Oracle again, but this time, we'll deal with *embedded* PHP programming. Some of you may be using this platform for your project. PHP works well with Web page applications (HTML).

As you proceed, please note that, besides the instructions at the links below, you'll find troubleshooting information and other tips at https://www.students.cs.ubc.ca/~cs-304/resources/faq.html#php.

Feel free to get help from Piazza or the TAs' office hours when setting up and running this tutorial.

Instructions

What we're going to do is fairly lightweight. Basically, we want to get you familiar with PHP and Oracle. Most of this tutorial involves getting it to run properly and observing how PHP and Oracle work together. We want you to complete this tutorial and hand in some simple screenshots to show that you got it working, and that you modified our sample code to display the output rows, and got that working, too.

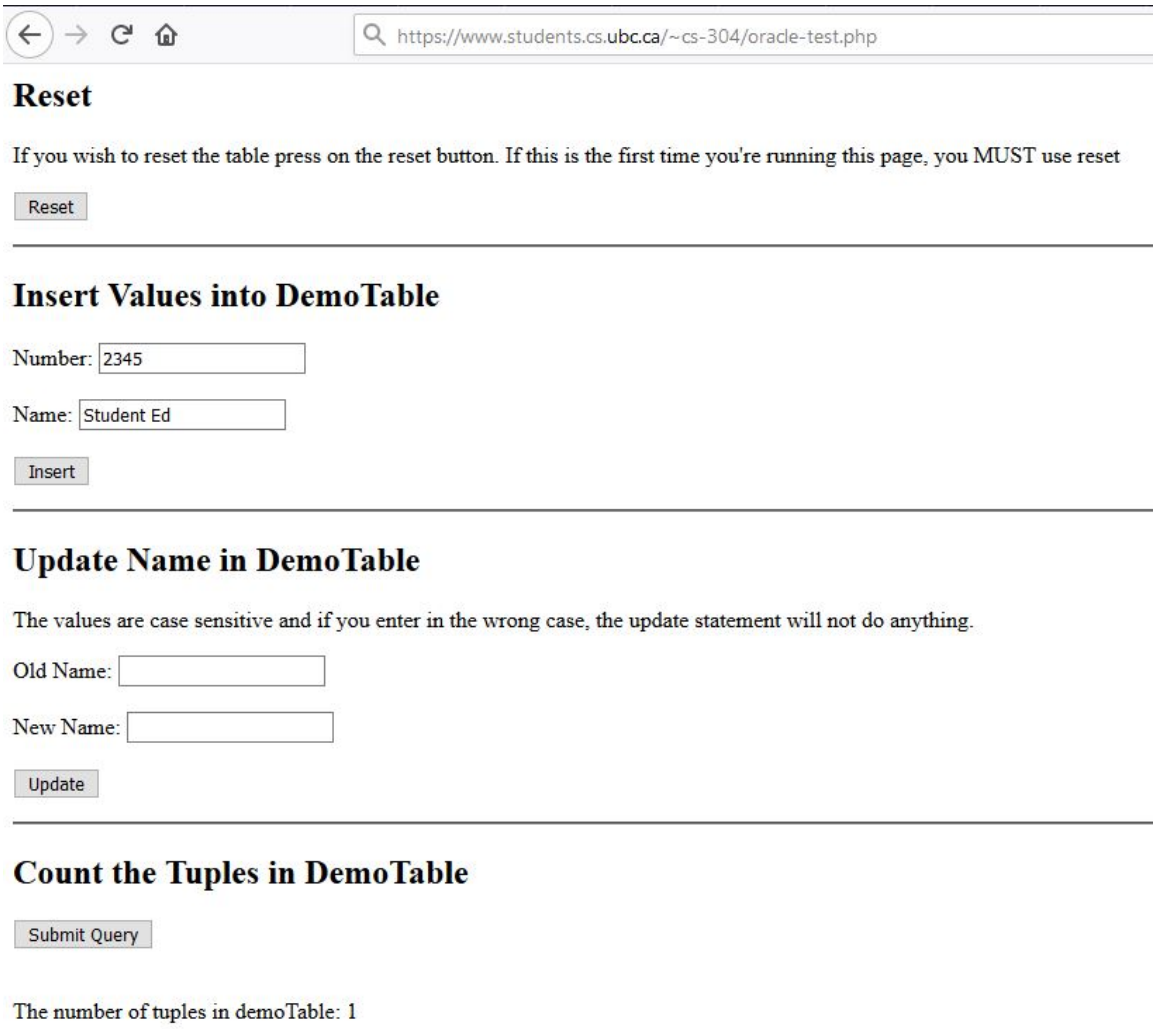
1. Go to this Web page:

   https://www.students.cs.ubc.ca/~cs-304/resources/php-oracle-resources/php-setup.html

   Then, follow the instructions. Note that there are a bunch of things to set up and test.

   Deliverable #1: A screenshot (or two) of the results of `oracle-test.php` or whatever you named your program. Give evidence that you got your PHP/Oracle application working. In other words, show a screenshot of your browser (upper left) that shows *both*

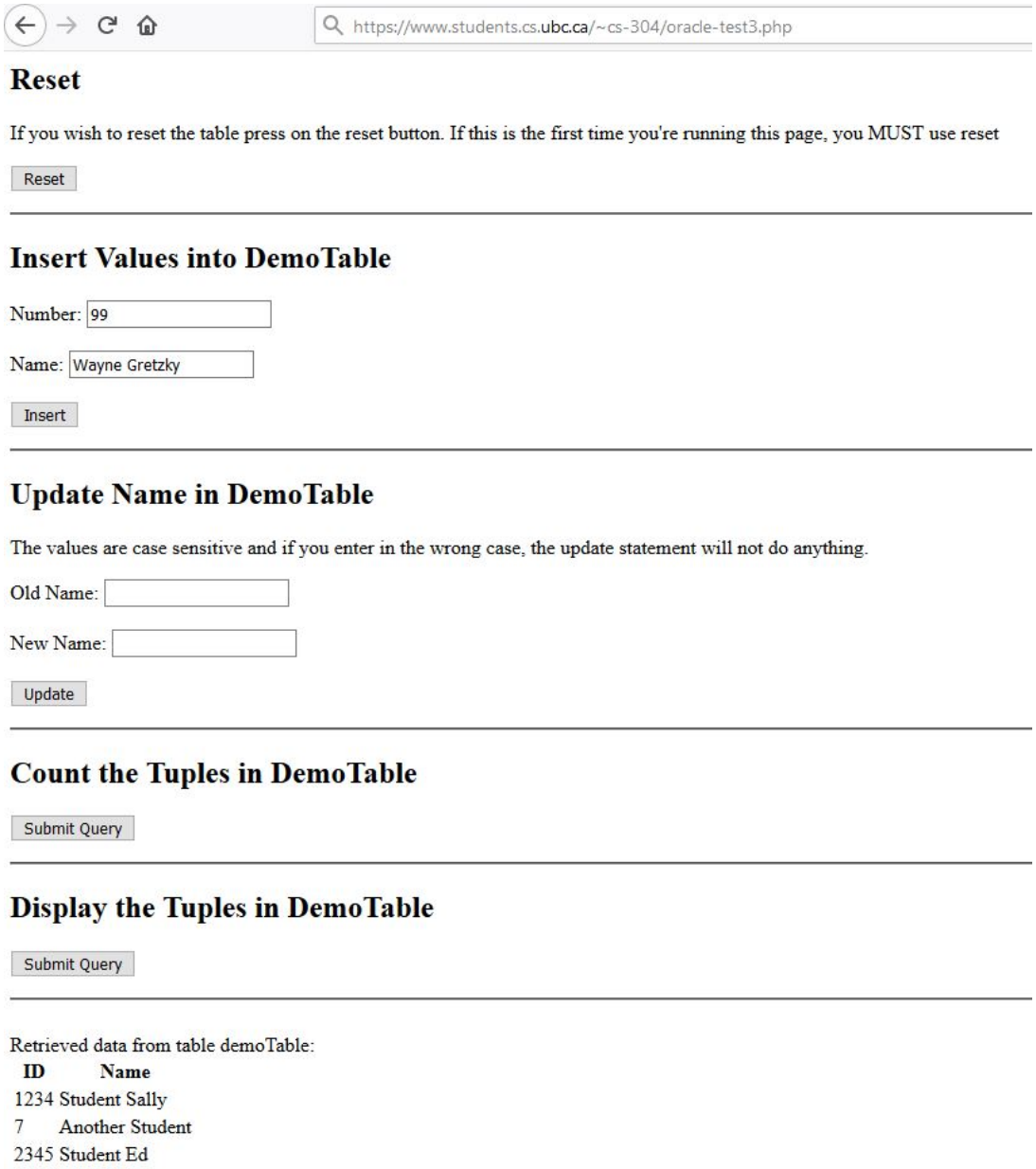
the URL in your browser, plus enough sample output to convince us that PHP/Oracle is working for you. For example, show us a count of the number of rows in the table after inserting a row into the table. Here is an example after inserting one row (not shown), displaying the count (shown), and starting to type in another insert (shown):

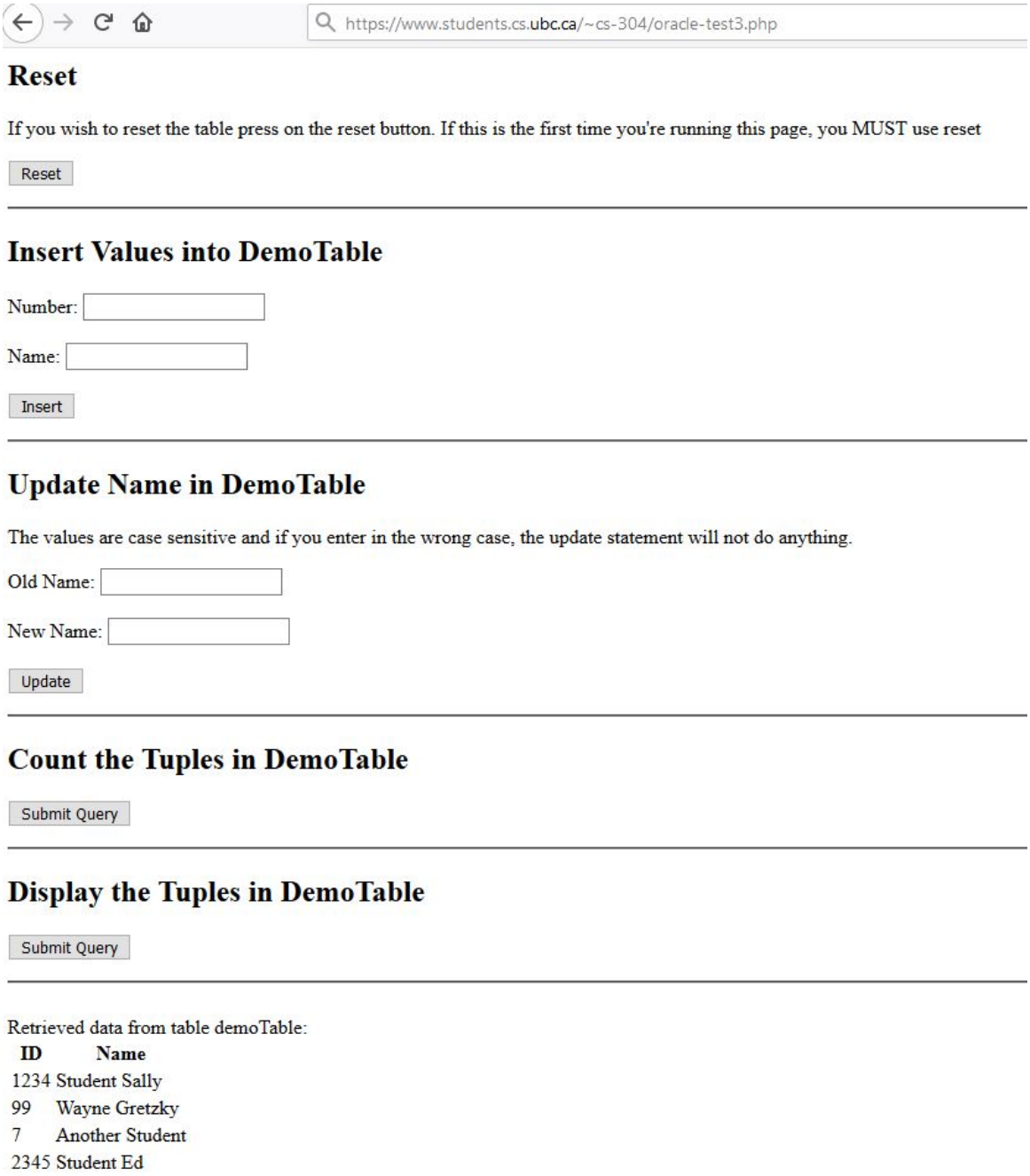


2. Let us extend the example, and modify some of the `oracle-test.php` script. Make a copy of it, and use a different file name. For your modifications, provide an extra option to display the output of all the rows of the demoTable on the screen.

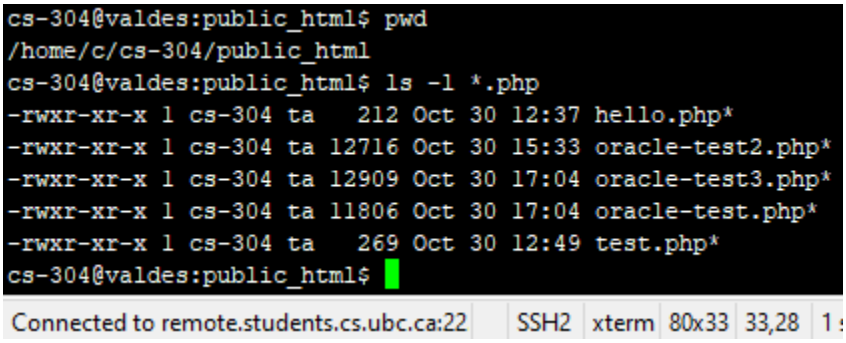
   Note: since we are changing the file name for deliverable 2/3, we will need to also change ALL our form methods in our new php script to redirect to itself. I.e. if we call our new file "new-deliverable" we would need to change our post action from <form method="POST" action="oracle-test.php"> TO <form method="POST" action="new-deliverable.php">

Deliverable #2 and #3: Provide a "before" and "after" screenshot of the result of this to convince us that you got things working. Include your CWL ID in the URL in the screenshots so that we know it's you and not your friend. Here is an example of two such screenshots:




https://www.students.cs.ubc.ca/~cs-304/oracle-test3.php
Reset
If you wish to reset the table press on the reset button. If this is the first time you're running this page, you MUST use reset
Reset
Insert Values into DemoTable
Number:
Name:
Insert
Update Name in DemoTable
The values are case sensitive and if you enter in the wrong case, the update statement will not do anything.
Old Name:
New Name:
Update
Count the Tuples in DemoTable
Submit Query
Display the Tuples in DemoTable
Submit Query
Retrieved data from table demoTable:
ID Name
1234 Student Sally
99 Wayne Gretzky
7 Another Student
2345 Student Ed

3. **Deliverable #4**: Provide a screenshot of your files with your timestamps – and again, with your userid. An example follows, but it doesn't have to be from the command line interface.



4. **Deliverable #5**: Provide a link to your PHP/Oracle application. We understand that you have already included this information in your screenshot but this deliverable is meant to make it easier for your TA to do a random spot check (if necessary).

   Submit your results on Canvas.

   And that's it!