Lab 5 – Read & Create: Take-Home
8 – Creating the "R" in CRUD

Creating and Populating your new table

- On Blackboard, download ***people.sql.***
- Upload the file to your turing *home* account
- Within your mysql engine, ***source people.sql***
- Verify the ***people*** table has been created and populated by typing the following within your mysql engine:
  SELECT * FROM people;

Before implementing CRUD

- Within ***public_html*** page create a folder named ***CRUD***
- Within this ***CRUD*** folder, upload the unzipped ***css*** and ***js*** folders as well as your ***included_functions.php***
- Add the following function to included_functions.php

```
function db_connection() {
        //  Complete require_once by replacing yourWebID with your own webid and DB.php with
        //  your php file that contains the DBHOST, USERNAME, PASSWORD, and DBNAME
        require_once("/home/yourWebid/DB.php");

        // 1. Create a database connection
        //  mysqli connect expects host, username, password, database name
        $mysqli = new mysqli(DBHOST, USERNAME, PASSWORD, DBNAME);

        // Test if connection successful
        if($mysqli->connect_errno) {
                die("Could not connect to server!<br />");
        }else {
                echo "Successful connection to ".DBNAME."<hr />";
        }
        return $mysqli;
}
```

- Add a parameter to the ***new_footer*** function, $mysqli: ***function new_footer($name="Default", $mysqli){***
  - At the end of the function add the line of code that will close your database:
    $mysqli -> close;

Download the following files from Blackboard, and upload to your CRUD folder on turing:

- ***session.php*** (session data that can be used while the browser is opened).
- ***index_INITIAL.php*** and rename to ***index.php*** (the primary file that is automatically rendered when this folder is selected to display – i.e., http://turing.cs.olemiss.edu/~*yourWebID*/CRUD/ will automatically display index.php)
- ***readPeople.php*** – A file you will modify to accomplish the "***R***" in CRUD
- ***addPeople.php*** – A file you will modify to accomplish the "***C***" in CRUD

Now, modify *readPeople.php* such that it connects to your database

- Look for the comment and write the code to create and execute the query to your database to select PersonID, FirstName, and LastName.
- Look for the comment and write the code to display the resulting FirstName and LastName within the while-loop (PersonID is not displayed but will be used later in the query string of an URL).
  - Remove the block comment that begins and ends with ***Uncomment Once Code Completed***
  - Recall that the tag to enter table data is <td> {data here} </td>
    You will need to use these tags when outputting first and last names

- Verify ***readPeople.php*** displays as follows:

Here is Who's Who

Name
Kevin Bacon
James Bond
George W. Bush
Bo Derek
Michael Jackson
Michael Jordan
Lee Majors
George Orwell
Molly Ringwald
George Washington
Oprah Winfrey

Add a person

Modify *index.php*

- Open the ***index.php*** file and to the line that invokes the ***new_header*** function, add the link location of index (second parameter). That is, add ***CRUD/*** to the link location. Recall that the new_header function should already point to:
  http://turing.cs.olemiss.edu/***~yourWebID***/ where ~yourWebID is your actual webid

  If done correctly, you should have a link to -> *http://turing.cs.olemiss.edu/**~yourWebID**/CRUD/*
  Verify that you can open this webpage

- Go to ***http://turing.cs.olemiss.edu/~yourWebID/CRUD/*** (NOTE: your ***index.php*** is automatically displayed. This page is eventually going to become our login page, but not until we complete our CRUD).
  - Click on the URL in the page and verify that you correctly redirect to readPeople.php
  - For now, the code for ***readPeople.php*** only displays those people found in the table. We want to add the ability to *create, update*, and *delete* to/from the people table.

Back in readPeople.php,

- Look for the comment and add the code to create 2 URLs (EDIT & DELETE) for each person
  - javascript is added to the delete tag to "confirm" delete should the user try to delete a person:

```
echo "<td> <a href='editPeople.php?id=".urlencode($row["PersonID"])."'>Edit</a>  </td>";

echo "<td> <a href='deletePeople.php?id=".urlencode($row["PersonID"])." ' onclick='return confirm('Are you sure?');'>Delete</a>  </td>";
```

- Verify that both links display for each person in readPeople.php
  - You will not be able to test if these two links successfully redirect since you don't have either the editPeople.php file or the deletePeople.php file. You can only verify that they show.

  - Once the files are created, note that ***href*** "passes along" the ID number of the person we want to edit/delete in the query string.

  - In each corresponding file, editPeople.php or deletePeople.php, we will be able to use $_GET["id"] to retrieve the id of the person we want to edit/delete

9 – Creating the "C" in CRUD

NOTE: ***readPeople.php*** already includes the link to ***addPeople.php***

Step 1 – Writing the Form

- Open ***addPeople.php***. Notice in lines 30-40, various statements assign a *value* to the $_SESSION key called *message* (recall that associative arrays use key-value pairs).
    - Now, open ***session.php***
        - *session_start()* is a php function that initiates a session. The session is active as long as the browser window is open.
        - The function *message*, will display those messages set by the $_SESSION key, *message*. In ***addPeople.php***, this *key* is assigned an error message *value* if there is an issue when trying to insert into your database.
- Look for the STEP 1. comments and add code that will create a HTML form:
    - This form should post to itself, like oneForm.php. So, post to ***addPeople.php***
    - Create text boxes for the following (input type is text and name is whatever you want to name it. Remember you will need to reference this name when using POST):
        - First Name
        - Last Name
        - Birthdate
        - Birth City
        - Birth State
        - Region
    - Add a submit input tag
        - Name should be *submit*
        - Value should be something like *Add Person*
        - Use the class *button tiny round*
- Verify Step 1 runs without errors

Step 2 – Inserting into Your Database

- Look for the Step 2. comments and add code that will insert the $_POST data into your database's people table.
  INSERT INTO people (FirstName, LastName, Birthdate, BirthCity, BirthState, Region) VALUES ('$_POST["FirstName"]' , '$_POST["LastName"]', …);

  NOTE: Birthdate, BirthCity, BirthState, and Region would have similar $_POSTs like FirstName & LastName

- The PHP is a little tricky for this query. Again, only writing up to the $_POST for last name, build your query as follows (DON'T FORGET the extra space at the end of each statement):

```
$query = "INSERT INTO people ";
$query .= "(FirstName, LastName, Birthdate, BirthCity, BirthState, Region) ";
$query .= "VALUES (";
$query .= " ' ".$_POST["FirstName"]." ', ";
$query .= " ' ".$_POST["LastName"]." ', ";
```

  NOTE: I have added spaces between the double-quotes (") and single-quotes(') to make it easier to read. Your actual code should not have spaces!

- Execute the query. If all works correctly, you will have a result for the variable ***$result*** and the ***if($result)*** statement will be true, displaying the message that your person was added to the database.

- Add your name to the database!!