Introduction to Databases – Flipped Classroom FOUR

PHP, HTML and MySQL – NEED A LAPTOP FOR THIS!

General Instructions:

- You must have watched all the videos for Week 7 and Week 8, you must have completed the review exercises and completed Quiz 4 BEFORE doing the “flipped classroom” so that you are fully prepared to work on the exercises with your group.
- In your Owl group of students, TOGETHER as a group, discuss and complete the questions below on the paper.
- Work as hard as you can to get as many of today’s exercises done. All the exercises will help you with assignment 3!
- The goal is for your group to learn from each other, so it is fine, actually encourage, to brainstorm and discuss and problem solve! Feel free to surf the internet or watch the course videos again to figure out your answers.

YOUR GROUP NUMBER: ________________

Group Member Name (PRINT)	Present Today (Circle One)
1	YES NO
2	YES NO
3	YES NO
4	YES NO

This week, in owl, have ONE of your group members submit the following:

- The names of the people who attended today in the Flipped 4 Assignment Owl submission textbox.
- A picture of THIS page and each of your name on the page ((jpg, png or gif format please). Circle who showed up and who didn’t
- A group selfie in the classroom (jpg, png or gif format please). EVERYONE MUST BE GIVING A THUMBS UP IN THE PICTURE
- A link to your museum.php URL
- A link to the URL for each activity after the first one that you got done (so I want to see the links to show me how far your group got today)
- Attach the 2 files in Owl: museum.php and getmuseum.php (you will likely get farther than that, but you don’t need to attach any more code).

And remember to hand in your paper assignment sheet to Laura

Objectives:

- To practice creating webpages that connect to a database
- To practice writing PHP code
- To learn collaboratively how to problem solve

QUESTION 1 – Creating the database and connecting to a table in the database (About 45 Minutes)

Part 1 (Creating the Database): You are going to create a museum database with 2 tables that have a one to many relationship (museum has many works of art). There will be a museum table and a works of art table. Museums contain works of art. Then you are going to create a webpage that connects to the database and displays the information and adds new works of art.

1. Have one member go into their virtual machine and then go into mysql as follows:
   ```
   mysql -u root -p
   ```
2. Go to: http://www.csd.uwo.ca/~lreid/blendedcs3319/flippedclassroom/four/dbsetup.txt and copy the SQL statements in this file and paste them into your vm's mysql.
3. Do a SELECT on both the ***museum*** table and the ***workofart*** table to make sure you have data in your database.
4. Copy one of the INSERT lines you just pasted in, slightly modify it so that it adds a new museum with the musID of 5, make up a fun name for your group's museum and put it in your group's favourite city and country and set the number of visitors to be 100.
5. Copy one of the INSERT lines you just pasted in, slightly modify so that is adds a new work of art with the *artname* being one of your group member's favourite colours, then the *artist* being the first name of one of your group members, the *year* being the current year and the *museum (whichmus)* being the new one you just added (i.e. 5). THEN DO THIS FOR EACH PERSON IN YOUR GROUP WHO SHOWED UP TODAY, it should look similar to this but the yellow will be the first names of your group members who showed up for today's class:

   ```
   INSERT INTO workofart(artname, artist,year,whichmus) VALUES ('orange', 'Laura', 2025, 5);
   INSERT INTO workofart(artname, artist,year,whichmus) VALUES ('yellow, 'Bart', 2025, 5);
   INSERT INTO workofart(artname, artist,year,whichmus) VALUES ('red', 'Homer', 2025, 5);
   ```

6. Push up arrow till you get to your two select statements (SELECT FROM museum… and SELECT FROM workofart…) and make sure those new rows were added.
7. Exit mysql

8. Move to the /var/www/html directory as follows:

```
cd /var/www/html
```

9. Create a new directory called flipped4 as follows:

```
mkdir flipped4
```

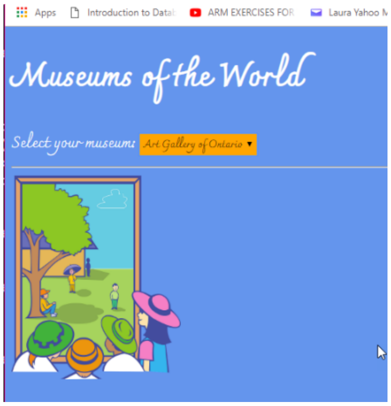
10. Move to that directory as follows:

```
cd flipped4
```

11. Create a new file called ***museum.php*** in that directory. You can use nano or emacs or vi as your editor (depending on what you installed). If you want, you could create it on your machine in notepad or textedit and then copy and paste it, or you can work right in your virtual machine but keep in mind, you won't have a backup of the file.

12. Use the html tags in ***museum.php*** to create a page that looks like the image to the right. You can get the html code and css code that was used to create this blue webpage OR if you want, you can write your own html and css to make something similar to the page above. If you want use my code to start you off, you can get the code here:

    a. HTML CODE: (in Chrome, right click and select *view source* and copy the code to your *museum.php* file)



    http://www.csd.uwo.ca/~lreid/blendedcs3319/flippedclassroom/four/starter.html

    b. While in the museum.php file on your VM, change this line:

    ```
    <h1>Museums of the World </h1>
    ```

    to this:

    ```
    <h1>Museums of Group ???</h1>
    ```

    where ??? is your group number.

    c. CSS CODE: (create a file in the *flipped4* directory called ***museum.css*** and put the following code in it): http://www.csd.uwo.ca/~lreid/blendedcs3319/flippedclassroom/four/museum.css

    d. While in museum.css, change the line that sets the background-color for the body to be your group's favourite color. Here is a list of possible colours.

13. Go to the correct URL for the new page you just created and make sure that you can see it in a browser. It should be something like this (but change the virtual machine number and remember to put the / at the end of the url):

    http://cs3319.gaul.csd.uwo.ca/**vm001**/flipped4/

HINT/SUGGESTION: If possible, have one member of your group have a laptop open that just points to the link above (and refreshes it) while the other member types the code into the VM, then you don't have to keep switching between Chrome and your VM Terminal.

14. Click on the *museum.php* file and make sure it works.
15. Now we are going to write the code to connect this webpage to the database that you just created in step 2 above.
16. Create a new file called connecttodb.php in your virtual machine and put the following code in that file. Your group will have to figure out how to fill in the blanks below:

```
<?php
$dbhost = "localhost";
$dbuser= "root";
$dbpass = "__________";
$dbname = "___________";
$connection = mysqli_connect($dbhost, $dbuser,$dbpass,$dbname);
if (mysqli_connect_errno()) {
     die("Database connection failed :" .
     mysqli_connect_error() . " (" . mysqli_connect_errno() . ")"  );
    } //end of if statement
?>
```

Be careful here → if you cut and paste you MIGHT get the wrong type of quotes, so you might want to retype your quotes.

17. Go back and edit your ***museum.php*** and add the following 3 lines of code right after the **<body>** tag (**NOT** after the </body> at the end of the page, but rather put this code after the <body> tag near the top of the page).

```
<body>
<?php
  include "con.php";
?>
<h1>Museums of the World </h1>
```

18. I have purposely made a mistake on the middle line above because I want to make sure that your php errors are displaying. In the browser, reload the page again and make sure you get error messages. If you are NOT getting errors, go to Step 3 of Workshop to see if you can find a mistake (it might be on Step3 part 4, if you didn't modify your php.ini file) OR call your prof over to help you.
19. Change the middle line in step 15 above to be:

    include "connecttodb.php";
20. Reload your page again and make sure you are not getting errors.

21. We have now established a connection to our database, now we want to retrieve some data. Create another new file called ***getmuseum.php***. Put the following code in that file but fill in the blank ($query = “   “) with the correct SQL statement to show all the data in the museum table;

```
<?php
  $query = "        ";
  $result = mysqli_query($connection,$query);
  if (!$result) {
    die("databases query failed.");
  }
  while ($row = mysqli_fetch_assoc($result)) {
     var_dump($row);
     echo $row . "<br>";
   }
   mysqli_free_result($result);
?>
```

22. Go back to your ***museum.php*** file and put your cursor after the **</select>** tag (do not put it after the <select> tag just yet, you will do that but not till a later step, for now, make sure you are AFTER the </select> tag), add the following red lines:

```
</select>
<?php
  include "getmuseum.php";
?>
```

23. Reload your browser. You should now see a dump of all the data from the museum table that looks pretty gross. We want to put the museum names in the dropdown box. So, each time we get a museum row in our while loop, we want to surround it with <option> and </option> (look in museum.php to see how the <option> tag works with Art Gallery of Ontario, that is what we want). We want to create lines should look something like this:

    <option>National Museum of China</option>

    using echo statements, except it will change the museum name for each museum.
    Edit your getmuseum.php file and remove these lines:

    ```
    var_dump($row);
    echo $row . "<br>";
    ```

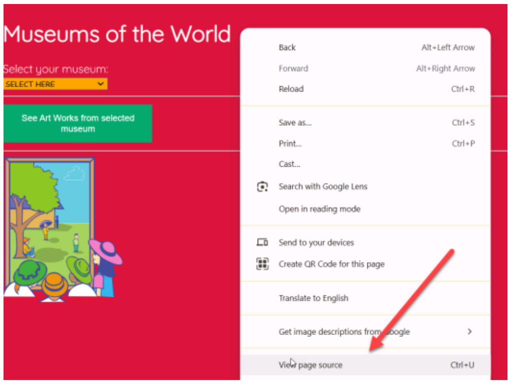
    and replace them with a new echo line (or lines) that will display the html tag ***<option>,*** then the contents of the *musname* column (you would do $row["musname"] to get that value), then a ***</option>*** html tag for each museum row in our museum table.

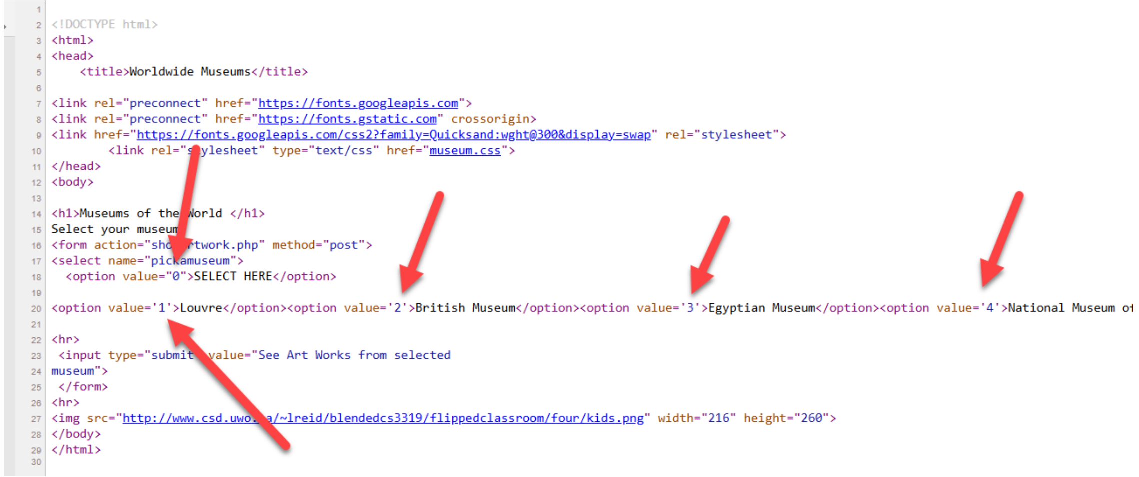
24. Save your ***getmuseum.php*** file and reload your page and make sure that you can now see the museum names.

25. The names are probably NOT in the dropdown box because in step 20 we put the 3 lines of code into ***museum.php*** AFTER the end of the dropdown box (the end of a dropbox is indicated by </select>). They should be **before** the </select>. Try moving the 3 lines in the box to the just above the </select> line, save your file and reload your browser. Make sure the museums are now in the dropdown box.

```
<?php
 include "getmuseum.php";
?>
```

26. Notice that we still have the *Art Gallery of Ontario* even though it is not in our database. Go the museum.php file and change the *Art Gallery of Ontario* to just say *"Select Here".*
27. Reload your file in the web browser, right click and select *View page source* to see the generated html. NOTICE THAT YOU CANNOT SEE THE PHP CODE, you can only see the html tags that your echo commands displayed.
28. Next you want to add the museum ids to your page as well as your museum names but the museum ids will be hidden in the html code and you will have to do *View Page Source* to see them, here is what it will look like AFTER you complete the next few steps:





```

<!DOCTYPE html>
<html>
<head>
    <title>Worldwide Museums</title>

<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Quicksand:wght@300&display=swap" rel="stylesheet">
        <link rel="stylesheet" type="text/css" href="museum.css">
</head>
<body>

<h1>Museums of the World </h1>
Select your museum
<form action="showartwork.php" method="post">
<select name="pickamuseum">
  <option value="0">SELECT HERE</option>

<option value='1'>Louvre</option><option value='2'>British Museum</option><option value='3'>Egyptian Museum</option><option value='4'>National Museum o

<hr>
 <input type="submit" value="See Art Works from selected
museum">
 </form>
<hr>
<img src="http://www.csd.uwo.ca/~lreid/blendedcs3319/flippedclassroom/four/kids.png" width="216" height="260">
</body>
</html>

```

29. Modify your echo statement so that it not only displays the museum name but it also puts the museum's id into the value section for the <option> boxes because you will need that for the next step. For example, one of your option html tags should NOT be this anymore: **`<option>National Museum of China</option>`**

but instead should be this:

```
<option value='4'>National Museum of China</option>
```

NOTE: the mysql column names are case sensitive so use musID NOT musid

ALSO NOTE: in PHP **.** means concatenate (i.e. a dot **.** means to concatenate 2 strings)

ALSO NOTE: you need to break the above line into strings and variables and concatenate them together to get it to work, here is a hint to get you started with this line BUT YOU MUST COMPLETE THE LINE YOURSELF:

```
  echo "<option value='" . $row[" . . .
```

HINT: break this line into parts (so that you can concatenate the parts together) like this (I have colour coded the parts to help you):

```
<option value='4'>National Museum of China</option>
```

30. Save your ***getmuseum.php*** file and go to Chrome, do refresh, then right click and select *view the source* to make sure that it is putting the museum id value on the <option> html tag. (you won't be able to see it in the Chrome page, you MUST do view source to see it)
31. Now we want to make our code display all the works of art depending on the item that the user selects from the dropdown menu.

QUESTION 2 – Creating a Join (About 1 hour)

1. The tricky part is that AFTER we load our page, we have already gone to the web server (your virtual machine), got the data we needed and now we are back at our page on the client (your laptop). So now the user will pick one of the museums and we need to go back to the database to get all the art work for the museum the user picked. The only way to get art work data is load up another .php page or reload our page, both of which force us to go to the web server again to get more data. It is much easier to go to a new php page so we will do that (you can just go to a new page for all the tasks in your assignment 3, it is much easier than reloading the page).

2. Create a new file called ***showartwork.php*** . In the ***showartwork.php*** file add the following code:

```
<!DOCTYPE html>
<html>
<head>
      <title>Worldwide Museums - The Art</title>
      <link rel="stylesheet" type="text/css" href="museum.css">
      <link href="https://fonts.googleapis.com/css?family=Mali" rel="stylesheet">
</head>
<body>
<h1>This museum has this art work:</h1>

</body>
</html>
```

3. Save your file and to http://cs3319.gaul.csd.uwo.ca/**vm???**/flipped4/ and click on your new file showartwork.php just to make sure it opens.
4. We are going to put the name of the museum selected between the <h1> and </h1> first. Remove the <h1> line above and replace it with this:

```
…
<body>
<?php
    include "connecttodb.php";
    $querymus="SELECT musname FROM museum WHERE musID=1";
    $result = mysqli_query($connection, $querymus);
    if (!$result) {
      die("database querymus on museum failed. ");
    }
    $row = mysqli_fetch_assoc($result);
    echo "<h1>Art Works at " . __________ . " include </h1>";
?>
</body> …
```

5. Figure out what to put in the blank spot on the echo line so that it echo out the museum that was returned. It won't be the one we selected just yet but just make sure it returns a museum name.
6. Try reloading your showartwork.php page in the browser and seeing if it works. Change the musID on the $querymus SELECT line in showartwork.php to look for the museum with an musID of 4 instead of 1 and try it again and make sure it shows a different museum name.
7. We want to connect this page to the main page so that when the user selects a museum from the dropdown box, it jumps to the showartwork.php page with the correct museum. In order to do that, we need to modify the museum.php file.

8. Open the museum.php file. We need to put a form around the dropdown box so that we can pass the value of whatever they selected to the showartwork.php page. Change that code as shown in below (add in the shaded bits):

```
…
<?php
   include "connecttodb.php";
?>
<h1>Museums of the World </h1>
  Select your museum:
  <form  action="showartwork.php" method="post">
    <select>
       <option value="1">SELECT HERE</option>
       <?php
         include "getmuseum.php";
        ?>
    </select>
 <hr>
      <input type="submit" value="See Art Works from selected
museum">
   </form>
```

9. Save your museum.php file and reload it in the browser. Notice the button at the bottom of the page. Click on it. If you did it correctly, it should take you to the showartwork.php page but it doesn’t show the correct museum yet. We need a way to pass whatever the user selected to the showartwork.php page. For any component (like a dropdown box or text box) on one page that you need to pass to another page, you must give the component a *name* attribute with a name that we make up.

10. We need to pass the value selected by the user from the dropdown. The dropbox was created using the <select> html tag so we need to give it a *name* and the name must have no spaces. Open your museum.php file again and change this line:

    ```
    <select>
    ```

    to be this:

    ```
    <select name="pickamuseum">
    ```

11. Now we need to make the receiving page (the showartwork.php page) gets a hold of the selected value from all the possible dropbox values. The way you do that is as follows:
    a. Open the showartwork.php file

b. Right under the line that opens your database, add a line that gets the value of the dropdown box and puts it into the PHP variable called $whichMus (add the shaded line below) and also add a line that checks to see if the value came across.

```
…
<body>
<?php
    include "connecttodb.php";
    $whichMus = $_POST["pickamuseum"];
    echo $whichMus . "<br>";
```

c. Refresh the museum webpage in the browser and select a museum and click on the button to make sure the value came across (it will show in the top left corner).

d. Now, go back into showartwork.php and figure out how to modify the sql select line below in showartwork.php so that it gets the selected museum by using the $whichMus variable and using concatenation:

```
$querymus="SELECT musname FROM museum WHERE musID=1";
```

e. Once you have modified the line above, save and reload your file in the browser and make sure when you select a museum, you get the correct museum in the second page header 1. You can remove the echo line too if you want that shows the museum id (but note: the echo command is really really useful in debugging, always try to echo your php variables if you can't figure out a mistake).

12. The next step is to get all the art works listed below for the selected museum. Right now, your code will look similar to this:

```
…
<body>
<?php
    include "connecttodb.php";
    $whichMus = $_POST["pickamuseum"];
    $querymus="SELECT musname FROM museum WHERE musID=" . $whichMus;
    $result = mysqli_query($connection, $querymus);
    if (!$result) {
      die("database querymus on museum failed. ");
    }
    $row = mysqli_fetch_assoc($result);
    echo "<h1>Art Works at " . $row["musname"] . " include </h1>";
?>
```

13. Now, in showartwork.php, under the area where you have the echo <h1> line, put the lines of code required to get all the art works for that museum. You will need a new PHP $query variable, for example $queryart and you will need a new SQL select statement and you will need a while loop like in Step 19 above on page 4.   You want to show all the art work for

each museum as a bulleted list. Also show the word *by* and then the artist who did the artwork, eg:

An Elephant by Rembrandt

The html tags for a bulleted list are <ul> to start the list and </ul> to end the list and you want to have a <li> html tag for each art work's title. Below is a shell of what you need to do (the gray areas). Try to figure out what should go in the blanks below (also, this link will remind you of the attribute names in your database):

```
…
<body>
<?php
    include "connecttodb.php";
    $whichMus = $_POST["pickamuseum"];
    $querymus="SELECT musname FROM museum WHERE musID=" . $whichMus;
    $result = mysqli_query($connection, $querymus);
    if (!$result) {
      die("database querymus on museum failed. ");
    }
    $row = mysqli_fetch_assoc($result);
    echo "<h1>Art Works at " . $row["musname"] . " include </h1>";
    $queryart = _______________________________ ;
    $result = mysqli_query(______, ____);
    if (!$result) {
      die("database queryart on museum failed. ");
    }
    echo "<ul>";
    while ($row = _________________ ) {


    }
    echo "</ul>";
    mysqli_free_result($result);
?>
```

14. Test that all the pieces of art work for each museum show up. If you are stuck, try outputting $queryart to make sure you did that part correctly. Also, it helps in debugging if you do "View Source" often to make sure the correct html is being generated.
15. Finally, add this line at the end of your showartwork.php page so that you can go back to your main page:
    `<a href="javascript:history.go(-1)">Go back to the Main Museum Page</a>`
    remember that if you put the above line between <?php and ?> then you need the echo command in front of the above line but if you put it after the ?>, then you will not need the echo command.
16. Test the back link. WELL DONE!

QUESTION 3 – Creating Users (About 15 Minutes).

Now you are going to write a page to check the password of a user before they can look at your database.

1. View the source for this webpage: http://www.csd.uwo.ca/~lreid/blendedcs3319/flippedclassroom/four/checkuser.html and copy the source code (the html tags) into a new file on your virtual machine in the flipped4 folder called ***checkuser.php***.
2. Look at the html tags for the above page you just created and figure out what name to give the php webpage that will check the user's id and password (HINT: look for the form tag)
3. Create a new PHP page and give it the name you figured out in the step above.
4. In this new file add the following lines:

```
<?php
   $dbhost = "localhost";
   $dbuser=  ________________________;
   $dbpass = $_POST['pwd'];
   $dbname = "flipped4db";
   $connection = mysqli_connect($dbhost, $dbuser,$dbpass,$dbname);
   if (mysqli_connect_errno()) {
        die("Database connection failed checking user");
   }
?>
```

5. Figure out what should go in the blank line above, it should look VERY similar to the line below it with the $dbpass. HINT: Look at both files to figure what you should do to make the username get passed from checkuser.php to checkpassword.php
6. Save both your files and reload the checkuser.php file in the browser, put the user of *root* and make sure that it fails if an incorrect password is added.
7. Notice that if you supply user *root* and the correct password, it does nothing.

8. You need to tell it which page to go to if the password works. To do that, open your password checking file and add the red lines.

```
    if (mysqli_connect_errno()) {
         die("Database connection failed checking user");
    } else {
         header('Location: museum.php');
    }
    exit;
?>
```

9. These lines say: go to the location of the file called ***museum.php*** and then exit from this page. Save your file and retry to make sure it goes to the museum page if the password was correct.
10. Now let's try adding a new user and see if we can make it work for someone other than the root user. Get into mysql (*mysql -u root -p*)
11. Create a new user called ***homer*** and give ***homer*** the password ***bart*** as follows:

```
CREATE USER 'homer'@'localhost' IDENTIFIED BY 'bart';
```

12. Reload your ***checkuser.php*** webpage again and see if it works when you type in ***homer*** with a password of ***bart***. It probably fails because even though you have created a new user, you haven't given your new user any privileges.
13. Go back into mysql and figure out how to allow *homer* to view all the data in the *flipped4db* database (Google the mysql Grant command). HINT: here is the start of the command you will need:

```
GRANT _____ ON _____   TO '___'@'localhost';
```

14. Reload your ***checkuser.php*** webpage and make sure that homer can see all the museums and the works of art for each museum.

RECAP: If you want to validate a user you need to:

a. Create a password webpage
b. Create a user in mysql with a password
c. Grant the privileges you want the user to have in mysql

QUESTION 4 – Inserting data into tables and debugging (About 20 Minutes).

In this next section, you are going to write code that adds a new work of art to a museum. I will give you ALMOST WORKING code with a few errors; your group needs to find the errors.

1. Go to this page: http://www.csd.uwo.ca/~lreid/blendedcs3319/flippedclassroom/four/addnewartwork.html view and copy the source code, and paste it into a new file called ***addnewartwork.php*** on your virtual machine. Save this file and exit from it. When you submit the new page you just created, it will call a page named ***addthework.php.*** Create a new file called ***addthework.php*** and copy the following code into it:

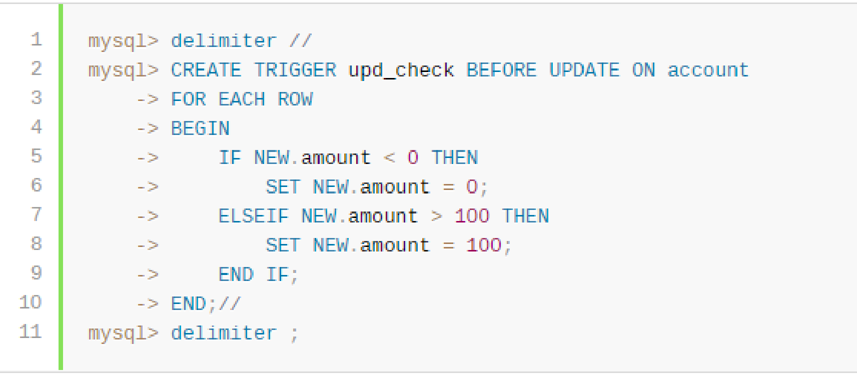
```
<?php
   include 'connecttodb.php';
   $art = $_POST["work"];
   $artist = $_POST["artist"];
   $theyear = $_POST["theyear"];
   $whichmus = $_POST["whichmus"];

   $query= 'INSERT INTO artwork (artname, artist, year, whichmus)  VALUES ("' . $art .'","' . $artist
. '","' . $theyear . ',' . $whichmus . ');';

   if (!mysqli_query($connection,$query)) {
      die ("Error while trying to add new art". mysqli_error($connection));
   } else {
     header('Location: museum.php');  //send back to museum page once it is done
     exit;
  }
?>
```

2. Now, in your browser, go to the directory we are putting all the files in on your virtual machine, it will be something like: ***http://cs3319.gaul.csd.uwo.ca/vm???/flipped4*** and click on the page called ***addnewartwork.php***. Enter some data and see if it works (it should NOT work). Use echo commands to print out what is in the variables and to double check your $query and see if you can find the mistakes. There are **3** mistakes your group will need to fix.

QUESTION 5 – Triggers (About 15 Minutes).

If you have time, create a trigger on the ***workofart*** table inside of mysql that is triggered whenever a new work of art is added. If the new row to be inserted has a null value for the artist or has a value of "" then set the artist to be **"Unknown"**. To refresh your memory, the image to the right is a mysql trigger. Use the webpage that you built in Question 4 above to test your trigger by adding a new piece of art where the artist is empty. You should NOT have to change or touch the php code if your group creates your trigger correctly.

--



```
mysql> delimiter //
mysql> CREATE TRIGGER upd_check BEFORE UPDATE ON account
    -> FOR EACH ROW
    -> BEGIN
    ->     IF NEW.amount < 0 THEN
    ->         SET NEW.amount = 0;
    ->     ELSEIF NEW.amount > 100 THEN
    ->         SET NEW.amount = 100;
    ->     END IF;
    -> END;//
mysql> delimiter ;
```

Hints:

- Build your trigger in Notepad, then you can change it if doesn't work the first time.
- The IF command uses only one equals for comparison, so put IF x=3… rather than IF x==3..
- Use the command *DROP TRIGGER name* if it doesn't work, then try rebuilding it.

Then go to ***addnewartwork.php*** and add a new artwork that has a blank for the artist name and then double check in ***museum.php*** to prove that it put ***Unknown*** for the artist's name.

QUESTION 6 – Getting your code onto your Repository

1. Hopefully you have set up your repository in ~/assignments. Move to that area as follows and make a new directory called flippy4:
```
cd ~/assignments/YOURUSERID
mkdir flippy4
cd flippy4
```

2. Then copy all the code you just created into flippy4 as follows:
```
cp /var/www/html/flipped4/*.*  ~/assignments/YOURUSERID/flippy4/
```

3. Make sure all the files are now in *~/assignments/youruserid/flippy4* as follows:
```
ls
```

4. Now do the following commands to put all your work into your gitlab.csd.uwo.ca:
```
git add .
git commit -m "flipped 4"
git push
```
(or it might be just ***git push*** or it might be ***git push origin HEAD:master***)

5. That should push all your code to your repository so that you have a copy of it

DO THE FOLLOWING BEFORE LEAVING TODAY:

1. **Take a selfie of all the members who showed up today in the classroom giving a "THUMBS UP"**
2. **Have one member in your group type the first names of every group member who showed up today in your Flipped 4 Assignment Owl textbox**
3. **Have one member add your museum.php file and your getmuseum.php file as attachments to the Flipped 4 Assignment in Owl.**
4. **Upload your selfie as an attachment to the Flipped 4 Assignment.**
5. **Take a picture of the front page of this paper and upload that picture as an attachment.**
6. **Include the link to your flipped4 museum.php file on your virtual machine to prove your museum.php worked in the Flipped 4 Assignment Textbox under your group names**
7. **A link to the URL for the farthest step you completed in the owl assignment textbox.**
8. **Hand in your worksheet to Laura**

NOTE: If you do NOT do the 8 steps before 11PM today above YOU WILL LOSE MARKS!