Database

Displaying data

To start: Set up project first with git and GitHub

Copy folder

Git init

Git remote rm origin

Git remote add origin

Keys are all set so do not change them, if you get an error it is something else.

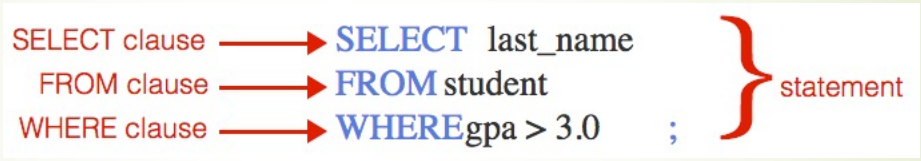
Create the live site, set it up.

1

Databases and SQL

Review SQL

SQL
Structured
Query
Language



Is the language we use to interact with the database.

UPPERCASE convention for sql reserved words

CamelCase convention for coder words (LastName vs last_name).

We break into separate lines by clause for readability

2

Databases - Tools

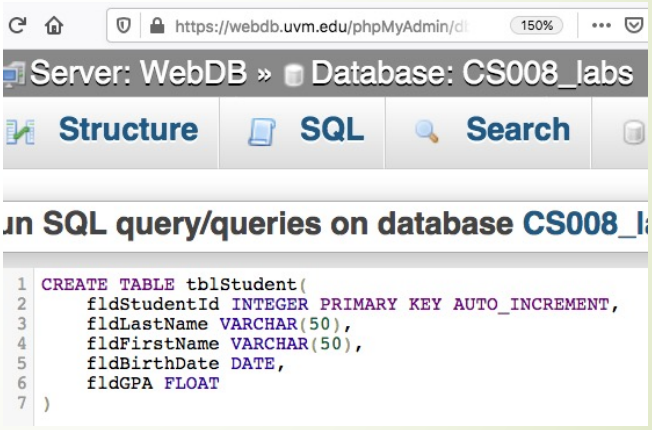
Web Based: phpMyAdmin

Type in your sql or fill out the form. In either case, you create a table by:

1. Defining the fields
2. Specify the data type
3. Specify other attributes.

NOTE: class naming conventions.

How to determine the size?



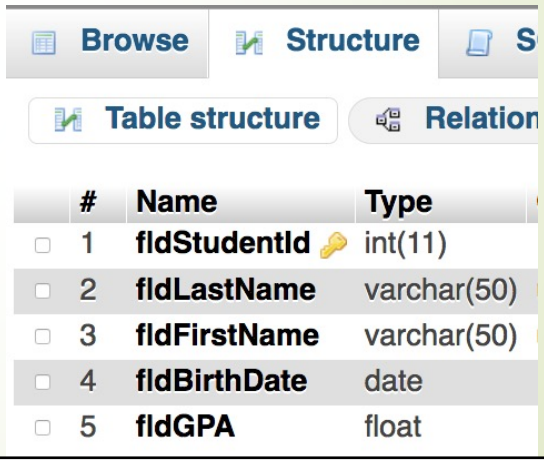
3

Relational Databases and SQL

Removing content from your code

I clicked on the Structure tab to see what it looked like.

Notice the Key symbol and int(11) vs integer



4

PHP and Databases

Constants

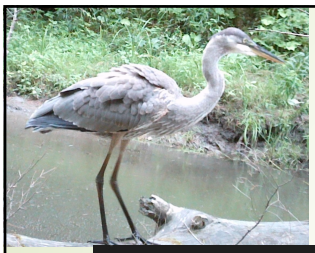
CONSTANTS – Values that do not change like pi. In code we often have values that are not going to change during the execution of the code but still need to be set.

For example in top.php you have two variables that are never going to change during the "run time" of the page.

Notice the constants do not have a dollar sign and are upper case.

```
define ('PHP_SELF', $_SERVER['PHP_SELF']);
define('PATH_PARTS', pathinfo(PHP_SELF));
```

5



PHP and Databases

lib – Library of common code

```
<!-- ****  include libraries  **** -->
<?php
include 'lib/constants.php';
```

Create a lib (short for library of common code)

File name is constants.php

Include constants.php in top.php so we have them in every file.

Be sure to add comments so they print in your html. It saves debugging time

6

PHP and Databases

Sample of constants

Our starter constants

Note how I sanitize the entire Server array.

```php
<?php
define('DEBUG', false);

$_SERVER = filter_input_array(INPUT_SERVER, FILTER_SANITIZE_STRING);

define ('SERVER', $_SERVER['SERVER_NAME']);

define('DOMAIN', '//' . SERVER);

define ('PHP_SELF', $_SERVER['PHP_SELF']);
define('PATH_PARTS', pathinfo(PHP_SELF));

define ('BASE_PATH', DOMAIN . PATH_PARTS['dirname'] . '/');

define('LIB_PATH', 'lib/');
```

7

PHP and Databases

Connecting to your database:

cs 008 method

Creates a database variable $pdo that we use to perform our SQL statements.

Writer vs reader

Online never admin

lab8/connect-DB.php

```
<!--Connecting-->
<?php
$databaseName = '[redacted]_labs';
$dsn = 'mysql:host=webdb.uvm.edu;dbname=' . $databaseName;
$username = '[redacted]_writer';
$password = '[redacted]';

$pdo = new PDO($dsn, $username, $password);
?>
<!--Connection Completed-->
```

8

PHP and Databases

Selecting Records:

cs 008 method

```
$sql = 'SELECT fldArea, fldDetails, fldImportance FROM tblComponents';

$statement = $pdo->prepare($sql);
$statement->execute();

$records = $statement->fetchAll();
```

Define SQL statement

Prepare

Execute

Put results into array

9

PHP and Databases

Connecting to your database:

cs 148 method using a class

A class is like a blueprint for an object, that defines values and methods for things you can do with the object.

Our database object will

1. create the connection from php to our database (value).
2. Allow us to perform sql queries on the database (method).

The initial setup of the php code for a class is longer that we did in cs 008 but in the end provides better maintainable code

10

PHP and Databases

Create the object – Part A sets us up to connect

```php
<?php
class DataBase{
    public $pdo = '';

    const DB_DEBUG = false;

    public function __construct($dataBaseUser, $whichDataBasePassword, $dataBaseName) {
        $this->pdo = null;

        // passwords be sure to add pass.php to .gitignore
        include 'pass.php';
        $DataBasePassword = '';

        switch ($whichDataBasePassword) {
            case 'r':
                $DataBasePassword = $dbReader;
                break;
            case 'w':
                $DataBasePassword = $dbWriter;
                break;
        }

        $query = NULL;

        $dsn = 'mysql:host=webdb.uvm.edu;dbname=';
```

File name and class name are the same.

public - variable that can be used throughout the class file

const are defined and used differently

__construct is code that happens when you create an instance (like a variable) of this class. NOTE: there are two underscore lines

$pdo and $this->pdo

DB_DEBUG and self::DB_DEBUG

We don't pass in the password as we would have to edit that in all files. It is common to change your passwords

11

PHP and Databases

Create the object – Part B connects

CS 008 method

```
$pdo = new PDO($dsn, $username, $password);
```

```php
        try {
            $this->pdo = new PDO($dsn . $dataBaseName, $dataBaseUser, $DataBasePassword);

            if (!$this->pdo) {
                if (self::DB_DEBUG) echo '<p>You are NOT connected to the database!</p>';
                return 0;
            } else {
                if (self::DB_DEBUG) echo '<p>You are connected to the database!</p>';
                return $this->pdo;
            }
        } catch (PDOException $e) {
            $error_message = $e->getMessage();
            if (self::DB_DEBUG) echo "<p>An error occurred while connecting to the
            database: $error_message </p>";
        }
    }  // ends constructor
```

NOTE I skipped the debugging code here.

Try and catch allows a mistake to happen but the program does not end.

$this->pdo instead of $pdo

IF statement helpful for debugging

12

PHP and Databases

Need the password file

lib > pass.php

```php
<?php
$dbReader="xxxx";
$dbWriter="xxxx";
?>
```

13

PHP and Databases

Our first method: Selecting records

Our class files connects php to the database.

A method to return our records based on a query.

```php
    public function select($query, $values = '') {

        $statement = $this->pdo->prepare($query);

        if (is_array($values)) {
            $statement->execute($values);
        } else {
            $statement->execute();
        }

        $recordSet = $statement->fetchAll(PDO::FETCH_ASSOC);

        $statement->closeCursor();

        return $recordSet;
    }
}  // ends the class
?>
```

14

PHP and Databases

Using our Class – creating an instance

```
print '<!-- make Database connections -->';
require_once(LIB_PATH . '/Database.php');
//create constants for these in lab 3
$thisDatabaseReader = new Database('rerickso_reader', 'r','RERICKSO_cs148_lab2');
```

Create the instance in top.php

$thisDatabaseReader is our instance variable that we will use

15

PHP and Databases

Using our instance or variable

In index.php (or any file you need to:

1. Define your query
2. Set the data values
3. Assign variable to the instance

$animals will be an array of the records that match the query, or it will be empty if none match.

```
<?php
include 'top.php';
$sql  = 'SELECT pmkWildlifeId, fldType, fldCommonName, fldDescription, fldHabitat, ';
$sql .= 'fldReproduction, fldDiet, fldManagement, fldStatus, fldMainImage ';
$sql .= 'FROM tblWildlife ';
$sql .= 'ORDER BY fldCommonName';

$data ='';
$animals =  $thisDatabaseReader->select($sql, $data);
```

16

PHP and Databases

Once you have your array the foreach loop is used to display your data.

Constants are a way to set common values for your code that can be set at run time but will not change during run time

Classes are a way to organize code, more code to write in the beginning but make maintenance and coding easier.

We connect to our database in one place only so if the conection needs to change we only change one file.

Code is not to hard, but it is longer …

17