Chapter 10: MySQL & PHP

PHP and MySQL • CIS 86 • Mission College

Tonight’s agenda

- Drop the class?
- Login file
- Connecting to a MySQL database
- Object-oriented PHP
- Executing a query
- Fetching a result
- Fetching a row: indexed array vs. associative array
- Adding data (a record) to your table
- Auto increment

Drop the class?

- Deadline: Nov. 18 (tomorrow)
- http://www.missioncollege.edu/class_schedule/2017_summer-fall/documents/calendar_summer-fall_2017.pdf
- You must drop the class yourself. It will not happen automatically and the school and the teacher cannot initiate it.

Connecting to MySQL

- On **MAMP** (Mac)
  - Your user name is `root`
  - Your password is `root`
  - `/Applications/MAMP/Library/bin/mysql -h localhost -u root -p`

- On **WAMP** (Windows)
  - Your user name is `root`
  - Your password is `blank (just hit return)`
  - `c:\wamp\bin\mysql\mysql5.6.17\bin\mysql -h localhost -u root -p`

- On the **Mission College** PHP server:
  - Your user name is `ca86_xx` (use your own account)
  - Your password is `php`

The overall process

1. Create a data table on your home computer using the MySQL monitor (command line).
2. Save the log so you can execute these mysql commands again later.
3. Write PHP code that accesses your table and displays the result.
4. When the PHP code works, upload it to the Mission College PHP server.
5. Log in to the Mission College PHP server and execute the same MySQL commands to create the data table there.

Create a table if necessary

```
create table classics (
    author (varchar(128),
    title varchar(128),
    type varchar(16),
    year char(4));
```

AUTO_INCREMENT

- Page 181. Page 252.

```
ALTER TABLE classics ADD id INT UNSIGNED NOT NULL
AUTO_INCREMENT KEY;
```

- Causes MySQL to set a unique value in this field for every row in the database. They generally start at 1 and increment, but we don't count on it. We just care that the value is unique.

- We can use this value to differentiate between records that might otherwise appear to be the same.

- For example, this might let us remove a duplicate entry, because the duplicate would have a different id.

Populate the table if necessary

```
insert into classics (author, title, type, year) values
('Jane Austen', 'Pride and Prejudice', 'Fiction',
'1811');

insert into classics (author, title, type, year) values
('Charles Darwin', 'The Origin of Species', 'Non-
Fiction', '1856');

insert into classics (author, title, type, year) values
('Charles Dickens', 'The Old Curiosity Shop',
'Fiction', '1841');

insert into classics (author, title, type, year) values
('William Shakespeare', 'Romeo and Juliet', 'Play',
'1594');
```

Examine the table

```
DESCRIBE classics;

SELECT * FROM classics;
```

Login file

- Textbook page 234

```
<?php // login.php
    $hn = ‘localhost’;
    $db = ‘ca86_99’;
    $un = ‘ca86_99;
    $pw = ‘php’;
?>
```

Multiple login files

- You may want a login file for the Mission College php server, and a different one for your home xAMP server.
- How can you create 2 different files and choose between them?

```
<?php
    if ($_SERVER['SERVER_NAME'] ==
        'php.missioncollege.edu')
        require_once ('login.mission.php');
    else
        require_once ('login.home.php');
?>
```

Connect to the database

```
<?php
    require_once ('login.php');
    $conn = new mysqli ($hn, $un, $pw, $db);
?>
```

Execute a query

```
<?php
    $query = “SELECT * FROM classics”;
    $result = $conn->query($query);
?>
```

Object Oriented PHP

```
$result = $conn->query($query);
```

- `$conn` is an OBJECT
- `query()` is a METHOD on the `$conn` object
- We use the `->` operator to get access to the methods.
- It is not the same as the `=>` operator we use for associative arrays.

Fetch a result as associative array

```
$rows = $result->num_rows;

for ($i=0; $i<$rows; $i++) {
    $result   = data_seek($i);
    $fields   = $result->fetch_assoc();
    $author   = $fields['author'];
    $title    = $fields['title'];
    $category = $fields['category'];
    $year     = $fields['year'];
    $isbn     = $fields['isbn'];
}
```

Fetch a result as indexed array

```
// http://php.net/manual/en/mysqli-result.fetch-row.php

$rows = $result->num_rows;

for ($i=0; $i<$rows; $i++) {
    $result    = data_seek($i);
    $fields    = $result->fetch_row();
    $author    = $fields[0];
    $title     = $fields[1];
    $category  = $fields[2];
    $year      = $fields[3];
    $isbn      = $fields[4];
}
```

Associative vs. Indexed?

- I hope it is apparent that associative array code is easier to follow.
- Associative:
  - `$author    = $fields['author']; // obvious`
- Indexed
  - `$author    = $fields[0];  // why??`
- With Associative Array code, you don't have to keep track of all the indexes and which one goes with which field.

Adding a record to a table

```
$query = insert into classics (author, title, type,
year) values (’Herman Melville', ’Moby Dick',
'Fiction', '1851');

$result = $conn->query ($query);

$query = insert into classics (author, title, type,
year) values (’Mark Twain', ’Huckleberry Finn',
'Fiction', '1885');

$result = $conn->query ($query);
```

Activity = Form

- Create a form to add new books to this database.