Databases and PHP
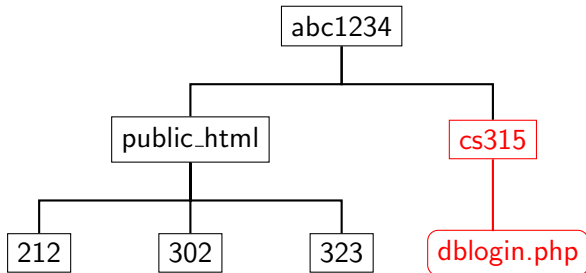
Class 30

PHP and Databases

- the safe, modern way to connect to a database from PHP uses the PDO library

- do not use mysqli or mysql_connect
- do not trust any code you see that uses these ancient deprecated systems

PHP Connection to a Database

- for security, create a file in a directory not within
  `public_html`
- name it something like `dblogin.php`
- give it permission 604

dblogin.php

the contents of dblogin.php:

```php
<?php
// your name

$db_host = 'borax.truman.edu';
$db_user = 'abc1234';
$db_pass = 'yourpassword';
$db_db = 'imdb';
?>
```

PHP Connection to a Database

```
require('../../cs315/dblogin.php');

$db = new PDO(
  "mysql:host=$db_host;dbname=$db_db;charset=utf8",
  $db_user, $db_pass,
  array(PDO::ATTR_EMULATE_PREPARES => false,
        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));
```

Simple Actions

- for actions that have no return value
- deletes and simple updates
- use the exec method

```
$db->exec('delete from user
           where last_login <
               date_sub(now(), interval 365 day)');
```

Simple Queries

- for very simple queries
- no parameters and return a single row
- use the query method
- the return value is an array of arrays
- outer array indexed by (row) numbers
- inner array indexed by column labels

```
$result = $db->query('select count(*) as cnt
                      from director');
$number_of_directors = $result[0]['cnt'];
```

More Complex SQL

- most SQL statements are more complex
- most SQL statements involve parameters
- query and exec are usually not safe or sufficient
- the general strategy is to use prepared statements

Prepared Statements

```
$sql = 'select name, format(rank, 2) as rank_f
        from movie
        where year = :year
        order by name asc';
$statement = $db->prepare($sql);
$statement->bindValue(':year', $year);
$statement->execute();
$results = $statement->fetchAll(PDO::FETCH_ASSOC);
foreach ($results as $row)
{
  $row['name'] ...
  $row['rank_f']...
}
```