Lecture 19

SQLite, PDO and ...

Start an Integrated Example.

* Course logo spider web photograph from Morguefile openstock photograph by Gabor Karpati, Hungary.
Extended Example

- Begin with the overall structure
- Modular with extras, e.g. Bootstrap
- Clean look – thanks to Jaime Ruiz
- Functionality
  - Creates database if one not present
  - Provides alternative sorted views
- Includes error handling
<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>create.php</td>
<td>Loading this page creates the database and also reports on the process</td>
</tr>
<tr>
<td>index.php</td>
<td>This is the example homepage including reference to original music list</td>
</tr>
<tr>
<td>lec18.zip</td>
<td>Zip file so students can download a complete copy</td>
</tr>
<tr>
<td>music2.db</td>
<td>The SQLite database file that is the foundation of this example</td>
</tr>
<tr>
<td>select.php</td>
<td>This page presents the database table as sorted by Rank, Artist or Year</td>
</tr>
<tr>
<td>inc</td>
<td>footer.php</td>
</tr>
<tr>
<td>inc</td>
<td>header.php</td>
</tr>
<tr>
<td>inc</td>
<td>page_setup.php</td>
</tr>
<tr>
<td>inc</td>
<td>sidenav.php</td>
</tr>
<tr>
<td>lib</td>
<td>album.php</td>
</tr>
<tr>
<td>lib</td>
<td>artist.php</td>
</tr>
<tr>
<td>lib</td>
<td>config.php</td>
</tr>
<tr>
<td>lib</td>
<td>database.php</td>
</tr>
<tr>
<td>lib</td>
<td>maintience.php</td>
</tr>
<tr>
<td>lib</td>
<td>session.php</td>
</tr>
<tr>
<td>lib</td>
<td>user.php</td>
</tr>
<tr>
<td>lib</td>
<td>users.tsv</td>
</tr>
<tr>
<td>lib</td>
<td>utils.php</td>
</tr>
<tr>
<td>source_file</td>
<td>list.php</td>
</tr>
<tr>
<td>source_file</td>
<td>Top+500+Albums.csv</td>
</tr>
<tr>
<td>assets</td>
<td>css</td>
</tr>
<tr>
<td>assets</td>
<td>img</td>
</tr>
<tr>
<td>assets</td>
<td>css</td>
</tr>
</tbody>
</table>

*csu-logo-footer.png* | CSU White Text Logo
*csu-logo.png*       | CSU Logo with Ram
In this page we demonstrate how to initialize a database. There is a lot going on here so study each step in turn.

1. Start a connection
   
   Connection successful.

2. Delete old tables. Here we want to use exec() because the drop command does not return a queryset but instead number of rows effected. (Note: Always returns 0 when deleting or creating tables.)
   
   Delete artist:
   
   Number of rows effected: 0
   
   Delete albums:
   
   Number of rows effected: 0

3. Create Tables. The SQL here is the same as if you were doing it via command line
   
   Artist Table
   
   Number of rows effected: 0
   
   Albums Table
   
   Number of rows effected: 0

4. Load Values Into Database

5. Check Values (should be 100 albums and fewer artists since The Beatles and Led Zeppelin Rock.
   
   Number of artists: 65
   
   Number of albums: 100
It Starts with a PDO Object

Start a connection

```php
<?php
try {
    $dbh = new PDO("sqlite:./music2.db");
    echo '<pre class="bg-success">';
    echo 'Connection successful.';
    echo '</pre>;
} catch (PDOException $e) {
    /* If you get here it is mostly a permissions issue
     * or that your path to the database is wrong
     */
    echo '<pre class="bg-danger">';
    echo 'Connection failed: ' . $e->getMessage();
    echo '</pre>;
    die;
}
```
PHP Data Object

- Extremely useful general purpose API!

The **PHP Data Objects** (PDO) extension defines a lightweight, consistent interface for accessing databases in PHP. Each database driver that implements the PDO interface can expose database-specific features as regular extension functions. Note that you cannot perform any database functions using the PDO extension by itself; you must use a database-specific PDO driver to access a
Execute Database Query

- Common pattern
  1. Assemble query as PHP string
  2. Then exec on the database

```php
<p>Delete artist:</p>
<?php
/*
 * This will only delete the table if it actually exists
 */
$sql = "DROP TABLE IF EXISTS artist";
$status = $dbh->exec($sql);
```
Create Example 1

- Note the use of try/catch.
- Note the use of the `<pre>` tag.
- Note the PHP die command.
- Experiment with directory permissions.
- Note the PDO exec command.
  - What does it return.
Create Example 2

- Pay Attention to the Schemas
  - Is there a FOREIGN KEY?
  - What does ASC do?
- Note the way albums are inserted.
  - Here you see PDO prepare used.
  - PDOstatment (object) execute is used.
  - Who is setting artist integer keys?
- Relations between artists and albums maintained by PHP code.
Create Example 3

- First use of PDO query.

```php
<?php
    $artist_num = $dbh->query("SELECT count(*) FROM artist");
    $album_num = $dbh->query("SELECT count(*) FROM album");
    echo "Number of artists: " . $artist_num->fetchColumn() . '<br />
    echo "Number of albums: " . $album_num->fetchColumn() . '';
?>
```

Note that SELECT count(*) is unusual. It returns 1 row with 1 column and the actual data is not pulled from the database. Instead, as the name implies, it is a summary of how many elements (rows) there are in a table.