Lecture 4 – PHP and MySQL

INLS 760
Web Databases
Spring 2013
Rob Capra
MySQL accounts

• Access accounts from ruby.ils.unc.edu
• MySQL is running on pearl.ils.unc.edu

[rcapra@ruby rcapra]$ mysql -h pearl.ils.unc.edu -u webdb_1 -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 379585 to server version: 4.1.20

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> set password for webdb_1 = password('mypassword');
Recall basic MySQL commands

• show databases
• use <databasename>
• show tables
• describe <tablename>
• drop table
• create table
• select * from <tablename>
• exit
Create a table

mysql> describe swcantina;

+-------------+----------------+--------+---------+-----------------+----------------+
| Field       | Type           | Null   | Key    | Default         | Extra          |
|-------------+----------------+--------+---------+-----------------+----------------|
| pid         | int(10) unsigned|        | PRI     | NULL            | auto_increment |
| pname       | varchar(100)    | YES    |         | NULL            |                |
| price       | int(11)         | YES    |         | NULL            |                |
| pdesc       | varchar(200)    | YES    |         | NULL            |                |
+-------------+----------------+--------+---------+-----------------+----------------+
Create a table

mysql> describe swcantina;

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Key</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>pid</td>
<td>int(10) unsigned</td>
<td></td>
<td>PRI</td>
<td>NULL</td>
<td>auto_increment</td>
</tr>
<tr>
<td>pname</td>
<td>varchar(100)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>price</td>
<td>int(11)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>pdesc</td>
<td>varchar(200)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
</tbody>
</table>

mysql> create table swcantina
    -> (pid int unsigned not null auto_increment primary key,
    ->    pname varchar(100), price int, pdesc varchar(200));
MySQL Bulk Loader

Data file = lect4/swcantina.dat

Better Blaster|200|a middle-grade blaster
Droid oil|5|to keep your droids running smoothly
Blaster|100|a simple blaster

mysql> load data
-> local infile "swcantina.dat"
-> replace into table swcantina
-> fields terminated by '|'"'
-> (pname, price, pdesc)
-> ;

mysql> source bulk.sql
Query the DB

```
mysql> select * from swcantina;

<table>
<thead>
<tr>
<th>pid</th>
<th>pname</th>
<th>price</th>
<th>pdesc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Better Blaster</td>
<td>200</td>
<td>a middle-grade blaster</td>
</tr>
<tr>
<td>2</td>
<td>Droid oil</td>
<td>5</td>
<td>to keep your droids running smoothly</td>
</tr>
<tr>
<td>3</td>
<td>Blaster</td>
<td>100</td>
<td>a simple blaster</td>
</tr>
<tr>
<td>4</td>
<td>Map of Tatooine</td>
<td>10</td>
<td>for all your travels</td>
</tr>
<tr>
<td>5</td>
<td>Jedi cloak</td>
<td>30</td>
<td>look your best this spring (Jedi only)</td>
</tr>
<tr>
<td>6</td>
<td>Regular cloak</td>
<td>10</td>
<td>look your best this spring</td>
</tr>
<tr>
<td>7</td>
<td>Map of Hoth</td>
<td>10</td>
<td>navigate the ice with confidence</td>
</tr>
<tr>
<td>8</td>
<td>Master Blaster</td>
<td>300</td>
<td>the best blaster</td>
</tr>
</tbody>
</table>

8 rows in set (0.00 sec)
```
Accessing the DB from PHP

• What version of PHP are you using?
• What features does it have enabled?

```php
<?php phpinfo(); ?>
```

• `phpinfo()` shows lots of info about your system.
• My recommendation: only leave this accessible when you need it. After using, delete it.
• **We will be using the mysql interface**
  – Slightly different from mysqli shown in book
WARNING

• All PHP-MySQL examples in this lecture assume KNOWN SAFE strings are used.

• This is an unrealistic assumption for real-world applications.

• Next lecture, we will look at how to use input gathered from users by HTML forms in a (relatively) safe way.
Accessing the DB from PHP

```php
<?php
    echo "SW Cantina Products<br>
    $h = 'pearl.ils.unc.edu';
    $u = 'webdb_1';
    $p = 'yourpassword';
    $db = mysql_connect($h,$u,$p) or die('Could not connect');
    mysql_select_db('webdb_1') or die ('Could not select db');
    $foo = "select * from swcantina";
    $result = mysql_query($foo);
    while ($row = mysql_fetch_array($result, MYSQL_ASSOC))
    {
        echo $row['pname'] . " ($" . $row['price'] . ") -- " . $row['pdesc'];
        echo "<p>
    }
    echo "</p>
?>
```
# mysql vs. mysqli proc vs. mysqli oo

<table>
<thead>
<tr>
<th>mysql</th>
<th>mysqli proc</th>
<th>mysqli oo</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>$db = mysql_connect($h,$u,$p)</code></td>
<td><code>$db = mysqli_connect($h,$u,$p);</code></td>
<td><code>$db = new mysqli($h,$u,$p);</code></td>
</tr>
<tr>
<td><code>mysql_select_db($dbname);</code></td>
<td><code>mysqli_select_db($dbname):</code></td>
<td><code>$db-&gt;select_db($dbname);</code></td>
</tr>
<tr>
<td><code>$result = mysql_query($query);</code></td>
<td><code>$result = mysqli_query($db, $query):</code></td>
<td><code>$result = $db-&gt;query($query);</code></td>
</tr>
<tr>
<td><code>$row = mysql_fetch_array($result, MYSQL_ASSOC);</code></td>
<td><code>$row = mysqli_fetch_assoc($result );</code></td>
<td><code>$row = $result-&gt;fetch_assoc();</code></td>
</tr>
</tbody>
</table>
Accessing the DB from PHP

```php
<?php
    echo "SW Cantina Products<p>";
    $h = 'pearl.ils.unc.edu';
    $u = 'webdb_1';
    $p = 'yourpassword';
    $db = mysql_connect($h,$u,$p) or die('Could not connect');
    mysql_select_db('webdb_1') or die ('Could not select db');
    $query = "select * from swcantina";
    $result = mysql_query($query);
    while ($row = mysql_fetch_array($result, MYSQL_ASSOC))
    {
        echo $row['pname'] . " ($" . $row['price'] . ") -- " . $row['pdesc'];
        echo "<p>";
    }
    ?>
```

Any security issues with this approach?
PHP and MySQL Password Issues

• Problem
  – To log in to a MySQL DB, PHP must supply a password:
    \[ 
    \text{db} = \text{mysql} \_ \text{connect}($\text{host, user, password}) \]
  – PHP file must be ugo-readable
    • If all is working, web access will not reveal the password in the file
    • However:
      – Password in PHP file is an unnecessary risk
      – Other users on the system may be able to see the password since it must be in a ugo-readable file.
PHP and MySQL Password Issues

• Really two problems
  – Extra protection from web access
  – Hide password from other users on a multi-user web server

• Extra protection from web access
  – Use a separate PHP file to connect to the DB
  – Store this file outside the web directory tree
  – “Require” this file whenever need to connect to DB

What is “outside” the web directory tree?

• check httpd.conf file
• look at DocumentRoot
• look at UserDir
• look at other settings
<?php
    $h = 'pearl.ils.unc.edu';
    $u = 'webdb_1';
    $p = 'yourpassword';
    $dbname = 'webdb_1';
    $db = mysql_connect($h,$u,$p)
        or die('Could not connect');
    mysql_select_db($dbname)
        or die ('Could not select db');
?>

<?php
    echo "SW Cantina Products<p>";
    require "/export/home/r/rcapra/dbconnect.php";
    $query = "select * from swcantina";
    $result = mysql_query($query);
    while ($row = mysql_fetch_array($result, MYSQL_ASSOC))
    {
        echo $row['pname'] . " ($" . $row['price'] . ") -- " . $row['pdesc'];
    } echo "<p>";
?>
PHP and MySQL Password Issues

• What about other users on a multi-user system?
• Can be addressed with:
  – Permissions and groups
    • Apache/PHP process needs to read, but others do not
  – PHP include directories + permissions
    • Set up special include directories with permissions and groups to restrict access
  – Apache configuration file (httpd.conf)
    • Can set “default” user and password variables for MySQL for scripts that run in specific directories
    • Requires configuration for each user (set these defaults for each user’s public_html directory)
    • Limits user’s ability to change their MySQL password
More PHP and MySQL

- Formatting and sorting MySQL results

lect4/showdbsorted.php

```php
<h1>SW Cantina Products</h1>
<table border="1" cellpadding="2">
<tbody>
<?php
    require "/export/home/r/rcapra/dbconnect.php";
    $query = "select * from swcantina order by price";
    $result = mysql_query($query);
    while ($row = mysql_fetch_array($result, MYSQL_ASSOC))
    {
        echo "<tr><td>" . $row['pname'] . "</td><td>" . $row['price'] . "</td><td>" . $row['pdesc'] . "</td></tr>";
    }
    mysql_close($db);
?>
</tbody>
</table>
```
More PHP and MySQL

- Adding records to a MySQL DB from PHP

lect4/addtodb.php

```php
<h1>SW Cantina Products</h1>
Adding to the database<br>

<?php
    require "~/export/home/r/rcapra/dbconnect.php";
    $pname = "Map of Naboo";  $price = "5";  $pdesc = "home of Amidala";
    $query = "insert into swcantina values";
    $query = $query . " (NULL,'" . $pname . "," . $price . "," . $pdesc;
    $query = $query . ")";
    echo "query = " . $query . "<br>";
    $result = mysql_query($query);
    if ($result) { echo mysql_affected_rows() . " rows inserted okay<br>"; } else { echo "problem inserting values<br>"; }
    mysql_close($db);
?>```
More PHP and MySQL

- `addslashes()`

lect4/addtodb2.php

```php
<h1>SW Cantina Products</h1>
Adding to the database<br>

<?php
  require "/export/home/r/rcapra/dbconnect.php";
  $pname = "Map of Alderaan";  $price = "7";  $pdesc = "Leia's home";
  $pname = addslashes($pname);
  $price = intval($price);
  $pdesc = addslashes($pdesc);
  $query = "insert into swcantina values" . " (NULL," . $pname;
  $query = $query . "," . $price . "," . $pdesc . ")";
  echo "query = " . $query . "<br>";
  $result = mysql_query($query);
  if ($result) { echo mysql_affected_rows() . " rows inserted okay<br>"; } else { echo "problem inserting values<br>"; }
  mysql_close($db);
?>
```

Special characters need to be prefaced with slashes, or they may cause errors.
More PHP and MySQL

• Deleting records in a MySQL DB from PHP

lect4/delfromdb.php

```php
<h1>SW Cantina Products</h1>
Deleting from the database<br>
<?php
    require "/export/home/r/rcapra/dbconnect.php";
    $query = "delete from swcantina where pdesc LIKE '%Leia%'";
    echo "query = " . $query . "<br>";
    $result = mysql_query($query);
    if ($result) { echo mysql_affected_rows() . " rows deleted okay<br>"; } else { echo "problem deleting values<br>"; }
    mysql_close($db);
?>
```
WARNING

• All PHP-MySQL examples in this lecture assume KNOWN SAFE strings are used.

• This is an unrealistic assumption for real-world applications.

• Next, we will look at how to use input gathered from users by HTML forms in a (relatively) safe way.
Prepared statements – steps

– Prepare
  • Construct statement, send to MySQL
  • MySQL parses and stores statement (ready to go)

– Bind parameters
  • Connects specific values with query parameters

– Execute
  • Executes the query with the bound parameters
  • Returns results

– Bind results
  • Can bind results to variables

IMPORTANT:
Parsing occurs ONCE
mysqli + prepared statements
Example (procedural)

<?php
$h = 'pearl.ils.unc.edu';
$u = 'webdb_1';
$p = 'webdb_1';
$dbname = 'webdb_1';
$dbh = mysqli_connect($h,$u,$p,$dbname);
if (mysqli_connect_errno()) {
    echo "Connect failed" . mysqli_connect_error();
    exit();
}
?>
<?php

echo "SW Cantina Products<p>";
require "/export/home/r/rcapra/dbconnect-i.php";
$sql = "select pname, price, pdesc from swcantina " . "where pname=?";
if ($stmt = mysqli_prepare($dbh, $sql)) {
    $myname = 'Map of Naboo';
    mysqli_stmt_bind_param($stmt,'s',$myname);
    mysqli_stmt_execute($stmt);
    mysqli_stmt_bind_result($stmt,$pname,$price,$pdesc);
    while (mysqli_stmt_fetch($stmt)) {
        echo "$pname  $price  $pdesc"
        echo "<br>";
    }
} else {
    echo "prepare failed"
}
mysqli_stmt_close();
mysqli_close();
?>
<?php

echo "Insert two rows<p>";
require "/export/home/r/rcapra/dbconnect-i.php"
$sql = "insert into swcantina (pname, price, pdesc)
    values (?, ?, ?)";
if ($stmt = mysqli_prepare($dbh, $sql)) {
    mysqli_stmt_bind_param($stmt,'sis',$fred,$ethel,$lucy);
    // do first row insert
    $fred = "Map of Naboo";
    $ethel = 10;
    $lucy = "like-new condition";
    mysqli_stmt_execute($stmt);
    // do second row insert
    $fred = "Map of Hoth";
    $ethel = 15;
    $lucy = "mint condition";
    mysqli_stmt_execute($stmt);
} else { echo "prepare failed"; }
mysqli_stmt_close();
mysqli_close();

echo "Done.<p>";
?>
<?php

echo "SW Cantina Products<p>";
require "/export/home/r/rcapra/dbconnect-i.php";
$sql = "select pname, price, pdesc from swcantina where
   price >= ? and price <= ? and pname LIKE ?";
if ($stmt = mysqli_prepare($dbh, $sql)) {
   $min = 10;
   $max = 50;
   $mylike = "%cloak%";
   mysqli_stmt_bind_param($stmt,'iis',$min,$max,$mylike);
   mysqli_stmt_execute($stmt);
   mysqli_stmt_bind_result($stmt,$pname,$price,$pdesc);
   while (mysqli_stmt_fetch($stmt)) {
      echo "$pname  $price  $pdesc"
      echo "<br>";
   }
} else {
   echo "prepare failed";
}
mysqli_stmt_close();
mysqli_close();
?>
mysqli + prepared statements
Example (object oriented)

```php
<?php
$h = 'pearl.ils.unc.edu';
$u = 'webdb_1';
$p = 'webdb_1';
$dbname = 'webdb_1';
$mysqli = new mysqli($h, $u, $p, $dbname);
if (mysqli_connect_errno()) {
    echo "connection failed" .
        mysqli_connect_errno();
    exit();
}
?>
```
mysqli + prepared statements
Example (object oriented)

```php
<?php
echo "SW Cantina Products<p>");
require "/export/home/r/rcapra/dbconnect-io.php"
$sql = "select pname, price, pdesc from swcantina 
  . "where pname=?"
if ($stmt = $mysqli->prepare($sql)) {
  $myname = 'Map of Naboo';
  $stmt->bind_param('s',$myname);
  $stmt->execute();
  $stmt->bind_result($pname,$price,$pdesc);
  while ($stmt->fetch()) {
    echo "$pname $price  $pdesc 
    echo "<br>
  }
} else {
  echo "prepare failed"
}
$stmt->close();
$mysqli->close();
?>
```
mysqli + prepared statements

• Links

http://dev.mysql.com/tech-resources/articles/4.1/prepared-statements.html
http://devzone.zend.com/article/686