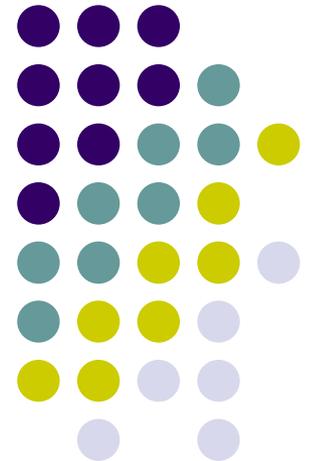


Databases and PHP

Accessing databases
from PHP



PHP & Databases



- PHP can connect to virtually any database
 - There are specific functions built-into PHP to connect with some DB
 - There is also generic ODBC functions that will work with many other DB
- Before you can connect with PHP you must already
 - have a database installed on the server machine
 - have the proper extensions added to PHP
 - have an account and password on the DB!



PHP & Databases

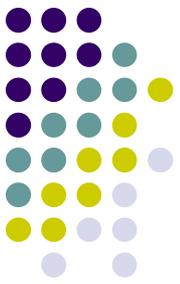
- These slides will discuss the basic elements of database connectivity to MySQL with PHP:
 - How to connect to a server from PHP
 - How to select a database from PHP
 - How to perform a query from PHP
 - How to format and view results from PHP
- More information on controlling MySQL from PHP and on using other DB with PHP can be found at:
 - <http://www.php.net/manual/>

Basic PHP functions for using MySQL



Function	Result
<code>mysql_connect()</code>	Opens a connection to the MySQL server. Requires a hostname, username, and password
<code>mysql_select_db()</code>	Selects a db on the MySQL server.
<code>mysql_query()</code>	Issues the SQL statement.
<code>mysql_fetch_array()</code>	Puts an SQL statement result row into an array
<code>mysql_result()</code>	Gets single element result data from a successful query.
<code>mysql_error()</code>	Returns a meaningful error message from MySQL.
<code>mysql_close()</code>	Closes a previously opened connection to a MySQL server.

Connecting to a MySQL server



- Must know the name of the server and a valid username and password.
- Syntax:

```
$conn = mysql_connect("hostName or IP", "userName", "password")  
or die(mysql_error() );
```
- die:
 - A built-in PHP function that prints an error message and exits the script.
 - The use here, with the `mysql_error()` function, will cause an error message to be printed.
 - Useful for debugging code.
- \$conn:
 - The `mysql_connect` function returns a pointer to a DB connection.
 - You will use this variable like a file pointer
 - Whenever you want to refer to this DB, use the `$conn` variable

Connecting to MySQL II



- Modern object-oriented technique.
- Syntax:

```
$conn = new mysqli($servername, $username, $password,  
$dbname);
```

```
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}
```

- **die:**
 - A built-in PHP function that prints an error message and exits the script.
 - will cause an error message to be printed.
- **\$conn:**
 - Contains an object that contains a DB connection.
 - You will use this variable like a file pointer
 - Whenever you want to refer to this DB, use the **\$conn** variable



Selecting a DB

- Must have already connected to MySQL
- Now must choose the DB to use

- **Syntax:**

If you connected via Method II the DB is already chosen

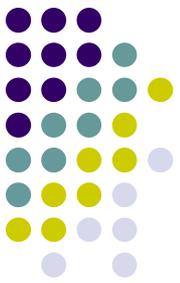
```
$db = mysql_select_db("DBname", $conn) or  
die(mysql_error) );
```

- Die: same use as before
- Must know the name of the database
- \$conn is the pointer returned from the mysql_connect function



Issuing a SQL command

- Must have already connected to MySQL *and* selected a DB
- Now can issue any SQL command that you have permission to use.
- Two steps:
 - Form the command into a string
 - Use either the `mysql_result` function or the `mysql_fetch_assoc` function.



Making a query

- Example:

```
$sql = "SELECT studentID, studentName FROM students
ORDER BY studentID ASC";
$sql_result = mysql_query($sql, $conn) or
die(mysql_error());
while ($row = mysql_fetch_assoc($sql_result)) {
    // process each row
}
```

- First line creates an SQL query from the *students* table.
- Second line sends the query to the mysql server represented by the variable `$conn`
- The result is placed in the `$sql_result` variable
- The while statement processes the results
 - `mysql_fetch_array` function returns the next row of the result (stored in variable `$sql_result`) as an associative array



Making a query, method II

- Example:

```
$sql = "SELECT studentID, studentName FROM students
ORDER BY studentID ASC";
$result = $conn->query($sql);
while($row = $result->fetch_assoc()) {
    // process each row
}
```

- First line creates an SQL query from the *students* table.
- Second line sends the query to the mysql server represented by the variable `$conn`
- The result is placed in the `$result` *object*
- The while statement processes the results
 - `fetch_assoc()` function returns the next row of the result (stored in variable `$result` object) as an associative array



Processing a query

- Example (cont). You could process the data in the while loop like this:

```
echo "<table>";
while ($row = mysql_fetch_assoc($sql_result)) {
    $fullName = $row[studentName];
    $fullID = $row[studentID];
    echo "<tr><td>$fullName</td><td>$fullID</td></tr>";
}
echo "</table>";
```

Processing a query, method II



- Example (cont). You could process the data in the while loop like this:

```
echo "<table>";
if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "ID: " . $row["studentId"]. "Name: " . $row["studentName"].
        . $row["dorm"]. "<br>\n";
    }
} else {
    echo "0 results";
}
```

Processing a query: addendum



- There is also a php function
`mysql_fetch_array($sql_result)`
- This function does the same thing as
`mysql_fetch_assoc($sql_result)`
 - Except that the resulting array can be indexed by *either* names or numbers.
 - If you don't need to access the array by numbers, stick to using
`mysql_fetch_assoc($sql_result)`



Closing a DB connection.

- Closing a DB connection.
 - A DB connection is *automatically* closed when a script ends.
 - If your script is long, however, it is good to close the connection explicitly.
 - Reason: there are a limited number of connections that a MySQL server can make (depends on admin settings)
 - Syntax:

```
mysql_close();  
Or  
mysql_close($conn);
```
 - Example:

```
$conn = mysql_connect("147.129.16.1", "testUser",  
    "conn!now") or die(mysql_error() );  
// all the code to do things with the database  
mysql_close($conn);
```



Closing a DB connect method II

- Closing a DB connection.

- Syntax:

```
$conn->close();
```

- Example:

```
$conn = new mysqli($servername, $username, $password, $dbname);  
// all the code to do things with the database  
$conn->close($conn);
```



Complete example: [phpDB1.php](#)

- Database: “Ithaca”
- Tables in database: “courses” and “students”
- Courses table:

courseID	Descript	instrId
304212	Stuff	56564
319291	Junk	76765
304245	Stars	5654

students table:

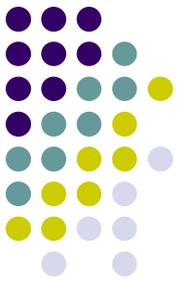
studentID	studentName	dorm
1111	John	Stanton
2222	Susan	Russian House
3333	Gwendolyn	Forbes
4444	Gabriel	Williams



Complete example: [phpDB1.php](#)

- The next program accesses the *students* table from the *Ithaca* database
 - Gets only the studentID and studentName
 - Prints the results into a table.

Complete example: phpDB1.php



```
<?php
// create connection
echo "<html>\n<head>\n<title>Our Students </title>\n</head>\n<body bgcolor=yellow>\n";
echo "<p>\n<h1 style='text-align:center'>Barr School</h1>\n</p>\n<p>\n";
echo "<table>\n";
// create the connection and choose the DB
$conn = new mysqli("localhost", "barrg", "ithaca", "lhaca");

// Check if connection was successfully made
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
```

Use your account name and password.
The third parameter is the DB name.

Use “localhost” if you’re connecting from the web, use the actual Linux server IP address (eg, 147.129.16.1) if you’re running this php script on a machine other than the Linux server

Complete example: phpDB1.php

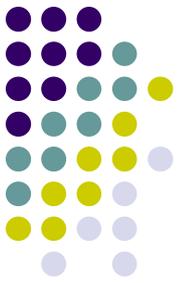


```
// create an SQL statement
$sql = "SELECT studentID, studentName FROM students ORDER BY studentID ASC";
$result = $conn->query($sql);
// Check whether query worked; if it didn't there will be 0 rows
if ($result->num_rows == 0) {
    die("Connection failed: " . $conn->connect_error);
}

while ($row = $result->fetch_assoc()){
    $fullName = $row['studentName'];
    $fullID = $row['studentID'];
    echo "<tr><td>$fullName</td><td>$fullID</td></tr>\n";
}
echo "</table>\n";
echo "</body></html>\n";
?>
```

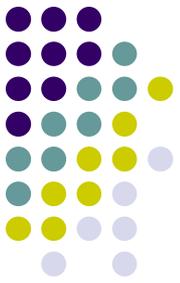
When there are no more rows, the `$result->fetch_assoc()` will return 0 which will be put in `$row`. But the result of the assignment statement is the value that is placed into the variable `$row`. The number 0 is interpreted as “false” so when there are no rows left, the loop will stop.

Complete example: the Junk Store

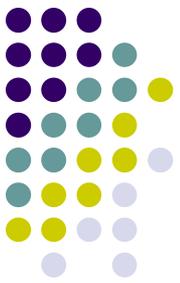


- A simple store application that uses a MySQL database
- Two scripts
 - junkStore.php Displays the items for sale
 - buyStuff.php receives an order, updates the database, sends cost information back to the browser

Complete example: the Junk Store



- A simple store application that uses a MySQL database
- Two scripts
 - junkStore.php Displays the items for sale
 - buyStuff.php receives an order, updates the database, sends cost information back to the browser

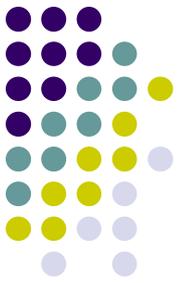


Complete example: the Junk Store

- Database: “Junk”
- Tables in database: “stuff”
- stuff table:

ID	Name	quant	Price	salePrice
1111	Watch	3	100.00	50.00
2222	Computer	4	999.99	799.00
3333	PDA	2	200.00	150.00
4444	Book	8	20.00	16.00
5555	Pickles	80	5.00	4.00

junkStore.php



```
<?php
// start the html page
echo "<html>\n<head>\n<title>John's Junk Jive</title>\n</head>\n<body
bgcolor=yellow>";
echo "<p><h1 style='text-align:center'>John's Junk Jive</h1></p><p>";

// create the connection
$conn = new mysqli("localhost", "barrg", "ithaca", "Junk");

// Check if connection was successfully made
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// echo "Connected successfully";
```

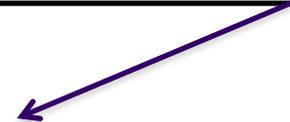


junkStore.php (continued)

```
// create an SQL statement
$sql = "SELECT ID, name, quant, price, salePrice FROM stuff ORDER BY ID ASC";
$result = $conn->query($sql);
// Check wether query worked; if it didn't there will be 0 rows
if ($result->num_rows == 0) {
    die("Connection failed: " . $conn->connect_error);
}

// Create the html table
echo "<table bgcolor=lightblue>\n";
echo "<form name=buyStuff method=POST action='buyStuff.php'>\n";
echo "<tr>\n<th>Item ID</th><th>Item Name</th><th>Quant Left</th><th>Price</th>";
echo "<th>Sale Price</th><th>Number Ordered</th>\n</tr>\n";
```

This line creates an html form that will call "buyStuff.php" when the "submit" button is clicked.



junkStore.php (continued)



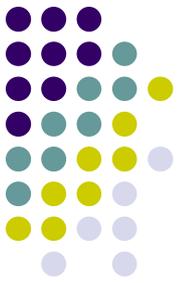
```
// get the info from the database
// fetch_assoc gets the next row of the query result
while ($row = $result->fetch_assoc()){
    $theName = $row['name']; // this gets the value associated with the 'name' field
    $theID = $row['ID'];
    $theQuant = $row['quant'];
    $thePrice = $row['price'];
    $theSale = $row['salePrice'];
    // the variable aRow will contain a string with all the html table info.
    // note that the variables that we created above are used to supply the values from the DB
    $aRow = "<tr>\n<td>$theID</td>\n<td>$theName</td>\n";
    $aRow = $aRow."<td>$theQuant</td>\n<td>$thePrice</td>";
    $aRow = $aRow."<td>$theSale</td>\n";
    $aRow = $aRow."<td><input type=text size=20 name=";
    $aRow = $aRow.$theName." value=0></td></tr>\n";
    echo $aRow;
}
echo "<input type=submit value='Buy Now'>\n"; // this is the button
echo "</form></table>";
?>
```

buyStuff.php

```
<?php
// create the web page
echo "<html>\n<head>\n<title>John's Junk Jive</title>\n</head>\n<body
bgcolor=yellow>";
echo "<p><h1 style='text-align:center'>John's Junk Jive</h1></p><p>";
echo "<h2>Thanks for buying the following stuff:</h2>\n</p>\n<p>";
// create a connection to the DB
$conn = new mysqli("localhost", "barrg", "ithaca", "Junk");
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// create an SQL statement
$sql = "SELECT ID, name, quant, price, salePrice FROM stuff ORDER BY ID ASC";
$result = $conn->query($sql);
// make sure that the query got results
if ($result->num_rows == 0)
    echo "0 results";
```



buyStuff.php (continue)



```
// create the html table
echo "<table border=1 bgcolor=lightblue>\n";
echo "<tr>\n<th>Item Name</th><th>Quant bought</th><th>Your Cost</th></tr>";
while ($row = $result->fetch_assoc()){
    $theName = $row['name'];
    $theID = $row['ID'];
    $theQuant = $row['quant'];
    $thePrice = $row['price'];
    $theSale = $row['salePrice'];
// foreach goes through each item received from web page that called this script
foreach ($_POST as $postName => $postValue){
    if ($postName == $theName && $postValue <= $theQuant && $postValue > 0){
        $totalCost = 0;
        $theQuant = $theQuant - $postValue;
        $totalCost = $totalCost + $postValue * $theSale;
        $aRow = "<tr style='text-align:center'>\n<td>$theName</td>\n";
        $aRow = $aRow."<td>$postValue</td>\n<td>\$totalCost</td>";
        $aRow = $aRow."</tr>\n";
        echo $aRow;
        $dbUpdate = "UPDATE stuff SET quant=$theQuant WHERE ID=$theID";
        $conn->query($dbUpdate);
    }
}
}
```

buyStuff.php (continue)

```
$conn->close();  
echo "</table>\n";  
echo "<a href='junkStore.php'>Shop More</a>\n";  
echo "</body></html>";  
?>
```

