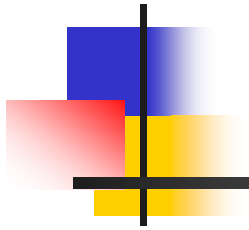


PHP-MySQL database applications





PHP-MySQL database applications

- Brief review of MySQL
- PHP MySQL functions
- examples



Brief Review Of MySQL

The MySQL comand line monitor

Creating database tables

Queries



Command Line Client

- Read the MySQL documentation
 - `c:\mysql\mysql\docs\manual_toc.html`
- Read MySQL install instructions on [MySQL.com](#)
- Command to enter monitor is
 - `mysql -u xxxx -p`
- To execute an SQL script use
 - `mysql -u xxxx -p < /path/script.sql`
- GUI client MyCC is better



Supply a
username
and a
password
when
prompted



Creating Database Tables

- Create **web_db** database to hold the tables:
 - **CREATE DATABASE web_db;**
- To create a table called notes:
 - **USE web_db;**
 - **CREATE TABLE notes (...);**
 - insert some rows for testing if necessary
- It is easy to write an sql script called **notes.sql** and use monitor to execute it



The Basic Queries

- CREATE
 - create databases and tables
- SELECT
 - select table rows based on certain conditions
- DELETE
 - delete one or more rows of a table
- INSERT
 - insert a new row in a table
- UPDATE
 - update rows in a table
- ALTER
 - alter the structure of a table



PHP MySQL Functions

- Connecting to a Database
- Making a query
- Using results of a query
- freeing resources
- closing the connection



Connecting To A Database

- `mysql_connect(server, username, password)`
 - connects to a MySQL server through a port
 - the default is the string `"localhost:3306"`
 - username is a string for the user name
 - password is a string for the password
 - returns FALSE on failure
- Example
 - ```
$db_link = mysql_connect("localhost:3306",
"test", "123");
```
- there is also the persistent `mysql_pconnect`





# Example From PHP Docs

---

```
<?php
 $link = mysql_connect("localhost", "mysql_user",
"mysql_password")
 or die("Could not connect: " . mysql_error());
 print ("Connected successfully");
 mysql_close($link);
?>
```



# Selecting A Database

---

- `mysql_select_db(name, link)`
  - select a database given by the string name
  - the link is optional and specifies the open link value such as `$db_link` returned by a connect statement.
  - if not supplied the last opened link is used.
  - returns TRUE on success else FALSE
- Example
  - `mysql_select_db( "web_db" );`



# Example From PHP Docs

---

```
<?php

$link = mysql_connect('localhost', 'mysql_user',
 'mysql_password')
 or die ('Not connected : ' . mysql_error());

// make foo the current db
mysql_select_db('foo', $link) or die ('Can\'t use foo
: ' . mysql_error());

?>
```



# Error Reporting (1)

---

- `mysql_error(link)`
  - Return an error string or error number
  - the link is optional
  - if not supplied the last opened link is used.
  - Empty string is returned if there is no error.
- Example
  - `mysql_error();`



## Error Reporting (2)

---

- `mysql_no(link)`
  - Return the error number
  - the link is optional
  - if not supplied the last opened link is used.
  - 0 is returned if there is no error.
- Example
  - `mysql_no();`



# Example From PHP Docs

---

```
<?php
 mysql_connect("localhost", "mysql_user",
"mysql_password");

 mysql_select_db("nonexistentdb");
 echo mysql_errno() . ": " . mysql_error() . "\n";

 mysql_select_db("kossu");
 mysql_query("SELECT * FROM nonexistenttable");
 echo mysql_errno() . ": " . mysql_error() . "\n";
?>
```



# Making A Query (1)

---

- `mysql_query(query, link)`
  - make a select query (link is optional)
  - query is a string for the MySQL query
  - Don't end the query with a semi-colon
  - Return value is a resource identifier or FALSE if the query is SELECT, SHOW or DESCRIBE
- Example (select all rows of books table)

```
$query = "SELECT * FROM books";
$result = mysql_query($query);
```



## Making A Query (2)

---

- INSERT and UPDATE queries
  - for these queries a resource is not returned
  - TRUE is returned on success
  - FALSE is returned on failure
- Example (describe the books table)

```
$query = "DESCRIBE books";
```

```
$status = mysql_query($query);
```





# Example From PHP Docs

---

```
<?php
$result = mysql_query("SELECT my_col FROM my_tbl")
 or die("Invalid query: " . mysql_error());
?>
```



# Retrieving Table Information

---

- `mysql_list_fields(database, table, link)`
  - For a select query it retrieves information from given table in given database. link is optional
  - The returned resource can be used to obtain properties of the table such as names of the table columns and field type information
- Example
  - `$fields = mysql_list_fields("web_db", "books");`



# Number Of Table Columns

---

- `mysql_num_fields(result)`
  - return the numbers of columns in a table
  - result is the resource returned by a call to the `mysql_list_fields` function
- Example

```
$fields = mysql_list_fields("web_db", "books");
$num_columns = mysql_num_fields($fields);
```



# Names Of Table Columns

---

- `mysql_field_name(result, index)`
  - return the name of the table column whose position is given by index (0,1,...)
  - result is the resource returned by a call to `mysql_list_fields`
- Example: the first column name

```
$fields = mysql_list_fields("web_db", "books");
$isbn = mysql_field_name($fields, 0);
```



# Example From PHP Docs

---

```
<?php
$link = mysql_connect('localhost', 'mysql_user',
'mysql_password');

$fields = mysql_list_fields("database1", "table1",
$link);
$columns = mysql_num_fields($fields);

for ($i = 0; $i < $columns; $i++) {
 echo mysql_field_name($fields, $i) . "\n";
}
?>
```



# Accessing Table Rows (1)

---

- `mysql_fetch_row(result)`
  - each call returns the next row as an indexed array where result is a resource returned from a call to `mysql_query` (FALSE if no more rows)
- Example

```
$query = "SELECT * FROM books";
$result = mysql_query($query);
$row = mysql_fetch_row($result); // row 0
$isbn = $row[0]; // isbn for row 0
```



## Accessing Table Rows (2)

---

- `mysql_fetch_assoc(result)`
  - as in `mysql_fetch_row` but next row is returned as an associative array
- Example

```
$query = "SELECT * FROM books";
$result = mysql_query($query);
$row = mysql_fetch_assoc($result); // row 0
$isbn = $row['isbn']; // isbn for row 0
```

# Accessing Table Rows (3)

- `mysql_fetch_array(result)`
  - combines `mysql_fetch_row`, `mysql_fetch_assoc`
  - returns row information as both an associative array and an indexed array
- Example

```
$query = "SELECT * FROM books";
$result = mysql_query($query);
$row = mysql_fetch_array($result); // row 0
$isbnA = $row[0]; // isbn for row 0
$isbnB = $row['isbn']; // can also get it this way
```





# Accessing table rows (4)

---

- `mysql_num_rows(result)`
  - returns number of rows from a select query
  - result is the resource returned by the select query
- `mysql_affected_rows(result)`
  - used after an INSERT, UPDATE, or DELETE query to return the number of rows affected
  - result is the resource returned



# Other Functions

---

- `mysql_real_escape_string(string,link)`
  - returns a string safe to use in `mysql_query`
- In MySQL 4.1 there are `mysqli_...` functions which are supposed to be improved.
- There are many other MySQL functions that we will not use.
- See PHP function reference for complete list



# Freeing Resources

---

- `mysql_free_result(result)`
  - free memory associated with the given resource called result (after a select query).
  - Not necessary except for large result sets
  - Done automatically when script exits.
- `mysql_close(link)`
  - close the database connection associated with the given link
  - doesn't do anything for persistent links.



# Processing column names

---

```
// Get resource for the field names
$fields = mysql_list_fields("web_db", "books");

// Get number of table columns

$num_cols = mysql_num_fields($fields);

// process them using a for loop

for ($i = 0; $i < $num_cols; $i++)
{
 $name = mysql_field_name($fields, $i)
 // do something here with $name
}
```



# Processing table rows (1)

```
while ($row = mysql_fetch_array($result))
{
 for ($i = 0; $i < count($row); *i
 {
 $col_value = $row[$i]'
 // Do something with $col_value here
 }
 // do end of row processing here
}
// do end of table processing here
```

From SELECT  
query

Here \$row is returned as an array so the inner loop is a for loop



## Processing table rows (2)

```
while ($row = mysql_fetch_assoc($result))
{
 foreach($row as $col_value)
 {
 // Do something with $col_value here
 }
 // do end of row processing here
}
// do end of table processing here
```

From SELECT  
query

Here \$row is returned as an associated array so the inner loop is a foreach loop. The foreach is easier to use.



# A db\_connect Function

---

- This function can be used in scripts to connect to a database. Put it in a file called **db\_connect.php** in your include path

```
<?php function db_connect($db_name)
{
 $host_name = "localhost:3306";
 $user_name = "xxxxxx"; $password = "yyyyyy";
 $db_link = mysql_connect($host_name,
 $user_name, $password)
 or die("Could not connect to $host_name");
 mysql_select_db($db_name)
 or die("Could not select database $db_name");
 return $db_link;
} ?>
```



# Books Display Example (1)

---

- First create the following books.sql file containing a sample database table

```
CREATE DATABASE IF NOT EXISTS web_db;
USE web_db;
CREATE TABLE books (
 isbn CHAR(15) PRIMARY KEY NOT NULL,
 title VARCHAR(100) NOT NULL,
 author VARCHAR(100) NOT NULL,
 pub VARCHAR(20) NOT NULL,
 year year NOT NULL,
 price DECIMAL(9,2) DEFAULT NULL
);
```





## Books Display Example (2)

---

- books.sql continued (insert some books)

```
INSERT INTO books VALUES (
 '0-672-31784-2',
 'PHP and MySQL Web Development',
 'Luke Welling, Laura Thomson',
 'Sams', 2001, 74.95
);
```

- Insert a few more (see [simple\books.sql](#))



## Books Display Example (3)

---

- Run **books.sql** through MySQL using the command
- **mysql -u xxxxx -p < c:/.../books.sql**
- Or use the gui client **MyCC**
- here **xxxxx** is your MySQL username and the -p option means to prompt for the password
- Now write a PHP script called **dbase.php** that displays the books in an HTML table



# Output

---

## Displaying the book database table using PHP

| isbn          | title                         | author                                   | pub               | year | price |
|---------------|-------------------------------|------------------------------------------|-------------------|------|-------|
| 0-672-31784-2 | PHP and MySQL Web Development | Luke Welling, Laura Thomson              | Sams              | 2001 | 74.95 |
| 0-13-066190-2 | Core MySQL                    | Leon Atkinson                            | Prentice Hall PTR | 2001 | 68.00 |
| 1-861003-02-1 | Professional Apache           | Peter Wainwright                         | Wrox Press Ltd    | 1999 | 74.95 |
| 0-13-089793-0 | Core WEB Programming, 2nd Ed  | Marty Hall, Larry Brown                  | Prentice Hall PTR | 2001 | 75.00 |
| 0-672-31880-6 | CGI in 24 Hours               | Rafe Colburn                             | Sams              | 2000 | 37.95 |
| 1-861003-14-5 | Beginning Perl                | Simon Cozens                             | Wrox Press Ltd    | 2000 | 59.95 |
| 0-596-00027-8 | Programming Perl, 3rd Ed      | Larry Wall, Tom Christianson, Jon Orwant | O'Reilly          | 2000 | 72.95 |
| 1-56592-243-3 | Perl Cookbook                 | Tom Christianson, Nathan Torkington      | O'Reilly          | 1999 | 56.95 |



# dbase.php (1)

---

- HTML header information

```
<?php require_once("db_connect.php"); ?>
<html>
<head>
<title>
Displaying the book database table using PHP
</title>
<h1>Displaying thebook database table using PHP</h1>
<?php
```



## dbase.php (2)

---

- Make a database connection

```
$db_link = db_connect("web_db");
```

- This uses the function defined in the include file `db_connect.php`
- My include path in `php.ini` is  
`include_path=".;c:\Apache\php-`  
`includes"`



current directory



## dbase.php (3)

---

- Send a SELECT query for all columns

```
$query = "SELECT * FROM books";
$result = mysql_query($query)
 or die("SQL Query failed");
```

- Obtain table properties

```
$fields = mysql_list_fields("web_db", "books");
$num_columns = mysql_num_fields($fields)
```



## dbase.php (4)

---

- Display column headings in an HTML table

```
echo '<table border="1">', "\n";
echo "<tr>\n";
for ($i = 0; $i < $num_columns; $i++)
{
 echo "<th>", mysql_field_name($fields, $i),
 "</th>\n";
}
echo "</tr>\n";
```



## dbase.php (5)

---

- Display the books table in as an HTML table

```
while ($row = mysql_fetch_assoc($result))
{
 echo "<tr>\n";
 foreach ($row as $col_value)
 {
 echo "<td>$col_value</td>\n";
 }
 echo "</tr>\n";
}
echo "</table>\n";
```





## dbase.php (6)

---

- Free resources and close connection

```
mysql_free_result($result);
mysql_close($db_link);
?>
</body>
</html>
```

- [view script dbase.php](#)
- <http://localhost/users/MYSQL/dbase.php>



# Calling Script From Button (1)

---

- Give the button a name and a value

```
<input type="submit" name="choice" value="Display">
```

- When form is submitted the name will exist

```
if (isset($_REQUEST['choice']))
{
 // process the button click here
}
...
```

- Multiple submit buttons should have different names or same name, different values



## Calling Script From Button (2)

---

- Another approach is to use the submit button label (value) to distinguish choices

```
$choice = isset($_REQUEST['choice']) ?
 $_REQUEST['choice'] : "";
if ($choice == "Display")
{
 // process the button click here
} ...
```

- Multiple submit buttons can have the same name but different labels (values)



# Calling Script From Link

---

- A link can be clicked to invoke the script again or to pass a parameter to it using the GET method (query string in link)

```
<?php
$this_url = $_SERVER['PHP_SELF'];
$edit_url = "$this_url?choice=edit&id=$id";
$delete_url = "$this_url?choice=delete&id=$id";
?>
<a href="<?php echo $edit_url?>">[Edit]
<a href="<?php echo $delete_url?>">[Delete]
```



# Suggestion Box

---

- Users make suggestions through a textarea in a form and submit them
- Suggestion is saved in a MySQL database table called sbox in database web\_db
- Suggestions are displayed along with the time
- [view script sbox/sbox.php](sbox/sbox.php)
- <http://localhost/php/MYSQL/sbox/sbox.php>



# Suggestion Box Display

## Suggestion Box

2002-05-23 11:26:16: Help  
2002-05-23 11:50:42: Help Again  
2002-05-23 12:55:37: And again  
2002-06-19 09:02:44: Have fun in the sun  
2002-06-19 11:49:13: Get some sleep

[Submit a new suggestion](#)

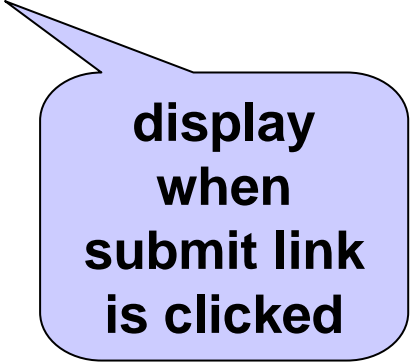


**initial  
display**

## Suggestion Box

Enter your suggestion:

Submit



**display  
when  
submit link  
is clicked**



# sbox.sql

---

```
CREATE DATABASE IF NOT EXISTS web_db;
USE web_db;
DROP TABLE IF EXISTS sbox;
CREATE TABLE sbox
(
 id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
 time DATETIME NOT NULL,
 suggestion TEXT NOT NULL
);
```

■ [view script sbox/sbox.sql](#)



# Script Logic

---

```
IF request for new suggestion THEN
 display form to get new suggestion
ELSE
 IF form was submitted THEN
 insert suggestion into database table
 ENDIF
 Retrieve suggestions from database table
 IF there are suggestions THEN
 display them
 ELSE
 Suggestion table is empty
 ENDIF
ENDIF

provide self link to enter a new suggestion
```





# sbox.php (1)

---

```
<?php require_once("db_connect.php"); ?>
<html><head><title>Suggestion Box</title></head>
<body><h1>Suggestion Box</h1>
<?php
$self = $_SERVER['PHP_SELF'];
if (isset($_REQUEST['new'])) // link was clicked
{
 ?>
 <form action="<?php echo $self ?>" method="POST">
 Enter your suggestion:

 <textarea name="suggestion" rows="5" cols="50"
 </textarea>

 <p><input type="submit" name="add"
 value="Submit"></p>
 </form>
<?php }
```



## sbox.php (2)

---

```
else
{
 $db_link = db_connect("web_db");
 if (isset($_REQUEST['add']))
 {
 $suggestion = $_REQUEST['suggestion'];
 $query = "INSERT INTO sbox SET time=NOW(),"
 . "suggestion='$suggestion'";
 mysql_query($query);
 }
}
```



**Forgot to use addslashes and stripslashes**



# sbox.php (3)

---

```
// Display all the suggestions

$query = "SELECT time, suggestion FROM sbox";
$result = mysql_query($query);
if (mysql_num_rows($result) > 0)
{
 while ($row = mysql_fetch_assoc($result))
 {
 $time = $row['time'];
 $suggestion = $row['suggestion'];
 echo "$time: $suggestion
\n";
 }
}
else
```

# sbox.php (4)

```
{
 echo "The suggestion box is empty";
}
?>
<p><a href="<?php echo $self ?>?new=1">
Submit a new suggestion</p>
</body>
</html>
<?php
}
?>
```

A trick for  
calling a  
script from a  
link with a  
parameter

- [view script sbox/sbox.php](#)



# Other Versions

---

- There are versions sbox2, sbox3, sbox4
  - sbox2 handles quotes and HTML text properly
  - sbox3 adds error checking
  - sbox4 is a simpler version of sbox3 for which the form is always displayed so no link is needed to add a new suggestion.



# sbox4.php (1)

---

```
<?php
 require_once ("db_connect.php");
 $db_link = db_connect ("web_db");
 $self = $_SERVER['PHP_SELF'];
?>
<html><head><title>Suggestion Box</title></head>
<body><h1>Suggestion Box</h1>
<form action="<?php echo $self ?>" method="POST">
Enter your suggestion:

<textarea name="suggestion" rows="5" cols="50"
 </textarea>

<p><input type="submit" name="add"
value="Submit"></p>
</form>
```



# sbox4.php (2)

---

```
<?php
// add a new suggestion if there is one

if (isset($_POST['add']))
{
 $suggestion = addslashes($_POST['suggestion']);
 if (strlen($suggestion) > 0)
 {
 $query = "INSERT INTO sbox SET time=NOW(),
 suggestion='$suggestion'";
 mysql_query($query);
 }
}
```



## sbox4.php (3)

```
// display all the suggestions
$query = "SELECT time, suggestion FROM sbox ORDER
 BY time DESC";
$result = mysql_query($query);
if (mysql_num_rows($result) > 0)
{
 while ($row = mysql_fetch_assoc($result))
 {
 $time = $row['time'];
 $suggestion = htmlspecialchars(stripslashes(
 $row['suggestion']));
 echo "$time: $suggestion

 \n";
 }
}
```





# sbox4.php (4)

---

```
 else
 {
 echo "The suggestion box is empty";
 }
?>
</body>
</html>
```