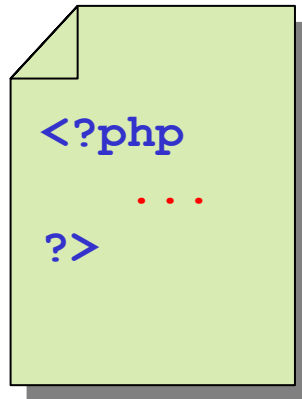


# COSC 2206 Internet Tools

---

PHP-MySQL database applications





# MySQL Database Applications

---

Brief review of MySQL  
PHP MySQL functions  
Bookstore display example  
WEB Database Applications



# 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
- 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



# Documentation

---

- The PHP documentation contains complete descriptions of the PHP functions needed to communicate with MySQL
- A nice feature of PHP is that MySQL support is built-in.
- You can search the PHP docs for a complete description of all MySQL functions.





# 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

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

// make foo the current db
mysql_select_db('foo', $lnk) 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 () ;`



# Error Reporting (3)

---

- There are also parameters that can be set in the php.ini file to control error reporting
- The @ operator can be used in front of a function call to suppress error messages.



# 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 simple/dbase.php](#)
- <http://localhost/php/simple/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>
<a href="<?php echo $delete_url?>">[ Delete ]</a>
```



# Suggestion Box

---

- Users make suggestions through a textarea in a form and submit them
- Suggestion is saved in a MySQL database table called sbbox in database web\_db
- Suggestions are displayed along with the time
- [view script sbbox/sbbox.php](#)
- <http://localhost/php/sbbox/sbbox.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:<br>
    <textarea name="suggestion" rows="5" cols="50"
        </textarea><br>
    <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 = $row['suggestion'];
        echo "<b>$time:</b> $suggestion<br>\n";
    }
}
else
```

# sbox.php (4)

```
{  
    echo "The suggestion box is empty";  
}  
?>  
<p><a href="<?php echo $self ?>?new=1">  
Submit a new suggestion</a></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 sbbox2, sbbox3, sbbox4
  - sbbox2 handles quotes and HTML text properly
  - sbbox3 adds error checking
  - sbbox4 is a simpler version of sbbox3 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:<br>
<textarea name="suggestion" rows="5" cols="50"
    </textarea><br>
<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(
            $row['suggestion']));
        echo "<strong>$time:</strong> $suggestion<br>
            \n";
    }
}
```



# sbox4.php (4)

---

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



# The Notes Database

---

- Form for submitting notes to the database
- button to submit a note
- button to display the current notes
- button to display notes based on search string
- notes can be edited and deleted
- [view script notes/notes.php](#)
- <http://localhost/php/notes/notes.php>



# notes.sql

---

```
CREATE DATABASE IF NOT EXISTS web_db;
USE web_db;
DROP TABLE IF EXISTS notes;
CREATE TABLE notes
(
    id          INT NOT NULL AUTO_INCREMENT,
    time        DATETIME NOT NULL,
    content     TEXT NOT NULL,

    PRIMARY KEY (id),
    INDEX (time)
)
```

- [view script notes/notes.sql](#)

# Submission Form And Display

## Notes Database

Enter a new note to submit or a search string:

**display\_notes  
does this  
display**

## Current Notes

**Time:** Wednesday, June 19th, 2002 at 08:58:48 AM

[\[ Edit \]](#) [\[ Delete \]](#)

Put more comments in the notes program

**Time:** Wednesday, May 1st, 2002 at 12:38:25 PM

[\[ Edit \]](#) [\[ Delete \]](#)

Use a cleaner logic for the notes program

**display\_note  
does each  
note**

# Search Display

## Notes Database

Enter a new note to submit or a search string:

**Search found 1 note matching "more"**

---

**Time:** Wednesday, June 19th, 2002 at 08:58:48 AM

[\[ Edit \]](#) [\[ Delete \]](#)

Put more comments in the notes program

---

**display\_search\_hits  
does this display**

**display\_note  
does each  
note**



# Edit Form

---

## Notes Database

### Note Editing Form

Put more comments in the notes program

Update

Cancel





# Note Script Logic

```
IF first time THEN display note entry form
ELSIF note submitted THEN
    insert note, display entry form and notes
ELSIF display button clicked THEN
    display entry form and notes
ELSIF search button clicked THEN
    display entry form and search hits
ELSIF delete link clicked for a note THEN
    display entry form and notes
ELSIF edit link was clicked THEN
    display note edit form
ELSIF update button on edit form clicked THEN
    update note, display entry form and notes
ELSIF cancel button on edit form clicked THEN
    display entry form and notes
ELSE error ENDIF
```

# notes.php (setup)

```
<?php
require_once ("db_connect.php");
$database = "web_db";
$table = "notes";

// Get form parameters. Non-existent values
// are mapped to empty string

$choice = isset($_REQUEST['choice']) ?
    $_REQUEST['choice'] : "";
$note = isset($_REQUEST['content']) ?
    $_REQUEST['content'] : "";
$id = isset($_REQUEST['id']) ? $_REQUEST['id'] : "";
$choice = strtolower($choice);
$note = addslashes($note);
```

id is a hidden field

# notes.php (logic 1)

```
start_html();  
$db_link = db_connect($database);  
if ($choice == "")  
{  
    display_entry_form();  
}  
elseif ($choice == "submit")  
{  
    insert_note($db_link, $table, $note);  
    display_entry_form();  
    display_notes($db_link, $table);  
}
```

first time  
no buttons  
clicked

submit  
button was  
clicked

# notes.php (logic 2)

```
elseif ($choice == "display")  
{  
    display_entry_form();  
    display_notes($db_link, $table);  
}
```

**display  
button was  
clicked**

```
elseif ($choice == "search")  
{  
    display_entry_form();  
    display_search_hits($db_link, $table, $note);  
}
```

**search  
button was  
clicked**

# notes.php (logic 3)

```
elseif ($choice == "delete")
{
    if (delete_note($db_link,$table,$id))
    {
        display_entry_form();
        display_notes($db_link, $table);
    }
    else
    {
        echo "<p><string>A note with ID# $id ",
            "was not found</b></p>\n";
    }
}
```

delete  
button was  
clicked

# notes.php (logic 4)

```
elseif ($choice == "edit")
{
    if ($row = select_note($db_link,$table,$id))
    {
        display_edit_form();
    }
    else
    {
        echo "<p><string>A note with ID# $id ",
            "was not found</b></p>\n";
        display_notes($db_link, $table);
    }
}
```



edit link  
was clicked

# notes.php (logic 5)

```
elseif ($choice == "update")
{
    update_note($db_link, $table, $id, $note);
    display_entry_form();
    display_notes($db_link, $table);
}
elseif ($choice == "cancel")
{
    display_entry_form();
    display_notes($db_link, $table);
}
else { // shouldn't get here }
mysql_close();
end_html();
exit;
```

**update  
button on  
edit form  
was clicked**

**cancel  
button on  
edit form  
was clicked**



# display\_entry\_form (1)

```
function display_entry_form()
{
    $script_url = $_SERVER['PHP_SELF'];
    print <<<END_HTML
        <form method="POST" action="$script_url">
        Enter a new note to submit or a search string:
        <p><textarea name="content" rows="5" cols="50"
        wrap=hard"></textarea></p>
        <p>
        <input type="submit" name="choice" value="Submit">
        <input type="submit" name="choice" value="Display">
        <input type="submit" name="choice" value="Search">
        </p>
        </form>
    END_HTML;
}
```





# display\_notes (1)

```
function display_notes ($db_link, $table)
{
    echo "<hr>\n<h2>Current Notes</h2>\n<hr>\n";
    $query =
        "SELECT id,"
        . "DATE_FORMAT(time, '%W, %M %D, %Y at %r')"
        . "AS time2, content FROM $table "
        . "ORDER BY time DESC";
    $result = mysql_query($query, $db_link) or
        die("Query failed");
    $num_rows = mysql_num_rows($result);
}
```



# display\_notes (2)

```
if ( $num_rows == 0)
{
    echo "The database is empty";
}
else
{
    while ($row = mysql_fetch_assoc($result))
    {
        display_note($row);
    }
}
mysql_free_result($result);
}
```



# display\_search\_hits (1)

```
function display_search_hits ($db_link, $table,
    $search_string)
{
    $search_string = trim($search_string);
    if ($search_string = "")
    {
        echo "<p><b>Please enter a search term</b></p>";
    }

    $query = "SELECT id, "
        . "DATEFORMAT(time, '%W, %M, %D, %Y at %r') "
        . "AS times2, content FROM $table WHERE "
        . "content LIKE '%$search_string%' "
        . "ORDER BY time DESC";
```



# display\_search\_hits (2)

```
$result = mysql_query($query, $db_link)
    or die("Query failed");
$num_rows = mysql_num_rows($result);
echo "<h2>Search found ",
    $num_rows == 0 ? "nothing" : "$num_rows note",
    $num_rows > 1 ? "s" : "", " matching \"",
    stripslashes($search_string),
    "\"</h2>\n<hr>\n";

if ($num_rows > 0)
{
    while ($row = mysql_fetch_assoc($result))
    {
        display_note($row);
    }
}

mysql_free_result($result);
}
```

# display\_note (1)

```
function display_note ($row)
{
    $id = $row['id'];
    $note = htmlspecialchars (stripslashes (
        $row['content']));
    $time = $row['time2'];

    // Use get method to call this script for
    // either the edit or delete choice

    $this_url = $_SERVER['PHP_SELF'];
    $edit_url = "$this_url?choice=edit&id=$id";
    $delete_url = "$this_url?choice=delete&id=$id";
}
```

id is a hidden  
field



# display\_note (2)

```
print <<<END_HTML
<table width="100%">
  <tr valign="top">
    <td width="80%"><b>Time:</b> $time
    <pre style="margin-left:2em;margin-top:0.5em;
margin-bottom:0.5em">$note</pre>
  </td>
  <td width="10%"><a href="$edit_url">[ Edit ]
  </a></td>
  <td width="10%"><a href="$delete_url">[ Delete ]
  </a></td>
</tr>
</table>
<hr>
END_HTML;
}
```

# display\_edit\_form

```
function display_edit_form($row)
{
    $script_url = $_SERVER['PHP_SELF'];
    $id = $row['id'];
    $note = htmlspecialchars(stripslashes(
        $row['content']));
    print <<<END_HTML
    <b>Note Editing Form</b><br />
    <form method="POST" action="$script_url">
    <input type="hidden" name="id" value="$id" />
    <textarea name="content" rows="5" cols="50"
        wrap=hard>$note</textarea><p>
    <input type="submit" name="choice" value="Update" />
    <input type="submit" name="choice" value="Cancel" />
    </p></form>
    END_HTML;
}
```

id is a  
hidden field



# delete\_note

```
function delete_note($db_link, $table, $id)
{
    $query = "DELETE FROM $table WHERE id = '$id'";
    $result = mysql_query($query, $db_link) or
        die("<p>Deletion failed</p>\n");
    return $result > 0
}
```

**If a row was deleted then  
\$result will be 1 so the  
return value is true**





# insert\_note

```
function insert_note($db_link, $table, $note)
{
    if (trim($note) != "")
    {
        $query = "INSERT INTO $table SET time = NOW(), "
            . "content = '$note'";
        $result = mysql_query($query, $db_link)
            or die("<p>Insertion failed</p>\n");
    }
}
```



# select\_note

```
function select_note($db_link, $table, $id)
{
    $query = "SELECT * FROM $table WHERE id = '$id'";
    $result = mysql_query($query, $db_link)
        or die("<p>Select note failed</p>\n");
    $row = mysql_fetch_assoc($result);
    return $row;
}
```

**If a note with given id is  
not found then a false  
value will be returned**



# update\_note

---

```
function update_note($db_link, $table, $id)
{
    $query = "UPDATE $table SET content = '$note' "
            . "WHERE id = '$id'";
    $result = mysql_query($query, $db_link)
        or die("<p>Select note failed</p>\n");
}
```



# Poll Example

---

- Simple form that gives a list of choices for the poll
- Save poll results in a database
- Don't allow user to do the poll more than once from the same URL
- Include a "show results" button that gives the percentages for each entry in the list of choices
- [view script poll/poll.php](http://localhost/php/poll/poll.php)  
<http://localhost/php/poll/poll.php>



# Poll Results Database Table

```
CREATE TABLE poll_results
```

■ [view script poll/poll.sql](#)

```
(  
    votes INT UNSIGNED NOT NULL DEFAULT 0,  
    yes   INT UNSIGNED NOT NULL DEFAULT 0  
);
```

```
INSERT INTO poll_results VALUES (0,0);
```

**This table keeps track  
of the total number of  
votes and the total  
number of yes votes**



# IP Address Database Table

---

```
CREATE TABLE poll_ips
(
    ip VARCHAR(255) NOT NULL,
    PRIMARY KEY (ip)
);
```

**When a user votes the ip address is stored in this table so that noone from this location can vote again**

# Voting Displays

## Poll Question

Do you like Java programming?

☒ Yes  
☐ No

Submit

Results

display  
after  
voting

display  
after trying  
to vote  
again

## Poll Results

Do you like Java programming?

Yes	100 %	1 votes
No	0 %	0 votes

[View Source](#)

Someone at your location has already voted

## Poll Results

Do you like Java programming?

Yes	100 %	1 votes
No	0 %	0 votes

[View Source](#)



# Script Logic

```
Open a database connection
Get user IP address
IF results button was clicked THEN
    display the poll results
ELSEIF submit button clicked AND vote entered THEN
    Get the vote from radio button
    IF user has already voted THEN
        Warn user
    ELSE
        submit the vote
    ENDIF
    display the poll results
ELSE
    display the poll voting form
ENDIF
Close the connection
```





# poll.php (1)

```
<?php
require_once ("db_connect.php") ;
start_html () ;
$ip_address = $_SERVER['REMOTE_ADDR'] ;
$poll_question = "Do you like Java programming?" ;

// Table for ip addresses

$ip_table = "poll_ips" ;

// Table for total votes and total yes votes

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



# poll.php (2)

```
if (isset($_REQUEST['results']))
{
    display_poll_results($poll_question, $vote_table);
}
elseif ( isset($_REQUEST['submit']) &&
         isset($_REQUEST['vote']) )
{
    $vote = $_REQUEST['vote'];
    if ( has_voted($ip_address, $ip_table) )
    {
        echo '<span class="warn">Someone at your
            location has already votes</span>';
    }
}
```



# poll.php (3)

```
else
{
    submit_poll_vote($ip_address, $vote,
                    $vote_table, $ip_table);
}
display_poll_results($poll_question, $vote_table);
else
{
    display_poll_form($poll_question);
}
mysql_close($db_link);
end_html();
exit(0);
```



# poll.php (4)

```
function start_html ()
{
    ?>
    <html>
    <head>
    <title>Web Poll using MySQL</title>
    <style type="text/css">
        .warn {font-weight: bold; font-size: small;
            color: #FF0000 }
        .bg1 { background-color: #AEC6D9 }
        .bg2 { background-color: #0099CC }
    </style>
    </head><body>
    <?php
}
```



# poll.php (5)

---

```
function end_html ()  
{  
    ?>  
    </body>  
    </html>  
    <?php  
}
```



# poll.php (6)

```
function has_voted($user_ip, $ip_table)
{
    // return false // comment when testing is complete

    $query = "SELECT ip FROM $ip_table WHERE
                                                    ip = '$user_ip'";
    $result = mysql_query($query) or
                                                    die("CheckIP query failed");
    return mysql_num_rows($result) > 0;
}
```



# poll.php (7)

```
function display_poll_form($poll_question)
{
    $script_url = $_SERVER['PHP_SELF'];
    ?>
    <h2>Poll Question</h2>
    <h3><?php echo $poll_question?></h3>
    <form method="POST" action=
        "<?php echo $script_url ?>">
    <input type="radio" name="vote" value=1 />Yes<br />
    <input type="radio" name="vote" value=0 />No<br />
    <input type="submit" name="submit" value="Submit"/>
    <input type="submit" name="results" value="Results"/>
    </form>
    <?php
}
```



# poll.php (8)

```
function display_poll_results($poll_question,
                              $vote_table)
{
    $total_votes = 0;
    $total_yes = 0; $total_no = 0;
    $percent_yes = 0; $percent_no = 0;
    $query = "SELECT votes, yes FROM $vote_table";
    $result = mysql_query($query)
              or die("Query failed");
    if ( mysql_num_rows($result) == 1 )
    {
        $row = mysql_fetch_assoc($result);
        $total_votes = $row['votes'];
        $total_yes = $row['yes'];
        $total_no = $total_votes - $total_yes;
    }
}
```





# poll.php (9)

---

```
if ($total_votes != 0)
{
    $percent_yes = round( (($total_yes /
                           $total_votes)*100), 1);
    $percent_no = round( (100 - $percent_yes), 1);
}
```



# poll.php (10)

---

```
?>
<h2>Poll Results</h2>
<table border="0" cellpadding="5">
  <tr>
    <td class="bg2" colspan="3"><b>
      <?php echo $poll_question ?></b></td>
  </tr>
```



# poll.php (11)

```
<tr>
<td class="bg1">Yes</td>
<td class="bg1"><?php echo $percent_yes ?> % </td>
<td class="bg1"><?php echo $total_yes ?> votes
</td>
</tr>
<tr>
<td class="bg1">No</td>
<td class="bg1"><?php echo $percent_no ?> %</td>
<td class="bg1"><?php echo $total_no ?> votes
</td>
</tr>
```



# poll.php (12)

---

```
</table>
<p><a href="seeit.php">View Source</a></p>
<?php
mysql_free_result($result) ;
}
```



# poll.php (13)

```
function submit_poll_vote($user_ip, $user_vote,
                          $vote_table, $ip_table)
{
    $total_votes    = 0;
    $total_yes      = 0;

    // Get the current total votes and total yes vote

    $query = "SELECT votes, yes FROM $vote_table";
    $result = mysql_query($query)
               or die("Query failed");
```



# poll.php (14)

```
if ( mysql_num_rows($result) == 1 )
{
    $row = mysql_fetch_assoc($result);
    $total_votes = $row['votes'];
    $total_yes = $row['yes'];
}
else // initialize the poll
{
    $query = "INSERT INTO $vote_table
              SET votes = '0', yes = '0'";
    $result = mysql_query($query)
              or die("<p>Query failed</p>");
}
```



# poll.php (15)

```
// Update total votes and total yes votes

$total_yes = $total_yes + $user_vote; // 1 = yes
$total_votes++;
$query = "UPDATE $vote_table
        SET votes = '$total_votes', yes = '$total_yes'";
$result = mysql_query($query)
        or die("<p>Update vote failed</p>\n");

// Record the browser ip so user can only vote once
$query = "INSERT INTO $ip_table
        SET ip = '$user_ip'";
$result = mysql_query($query)
        or die("<p>Insertion of ip failed</p>\n");
}
```



# Authentication with MySQL

---

- Instead of using basic authentication that is implemented using HTTP headers it is better to use a database to store user names and passwords.
- A session variable can be used to identify a valid user.
- First create a data base with fields for the user id and the password:





# login.sql (user database)

```
USE web_db;
DROP TABLE IF EXISTS login;      ■ view script login-db/login.sql

CREATE TABLE login
(
    name VARCHAR(10) NOT NULL,
    password VARCHAR(30) NOT NULL,
    PRIMARY KEY (name)
);
# insert a few users and encrypt the passwords

INSERT INTO login VALUES ('test', PASSWORD('123'));
INSERT INTO login VALUES ('look', PASSWORD('kool'));
INSERT INTO login VALUES ('Fred', PASSWORD('Jones'));
```



# user database

---

```
mysql> use web_db;
Database changed
mysql> SELECT * FROM login;
+-----+-----+
| name | password          |
+-----+-----+
| test | 773359240eb9a1d9 |
| look | 7d74a0bb51520618 |
| Fred | 64099a8d551f7d81 |
+-----+-----+
3 rows in set (0.00 sec)

mysql>
```



# Login script logic

```
Start a session
IF username AND password were submitted THEN
    Check that these values are alphanumeric.
    IF not THEN set them to empty strings END IF
    IF there is a matching row in login table THEN
        Set a 'valid-user' session variable having
        value the username as value.
    ELSE
        Display login page with form to login
    END
ELSE
    Display login page with form to login
END
```



# login.php (1)

- [view script login-db/login.php](#)

```
<?php
require_once ("db_connect.php") ;
session_start() ;
if (isset($_REQUEST['userid'] &&
                                isset($_REQUEST['password'])))
{
    // Check for alphanumeric values

    $id = ereg ("^[a-zA-Z0-9]+$" , $_REQUEST['userid'])
        ? $_REQUEST['userid'] : "" ;
    $pass = ereg ("^[a-zA-Z0-9]+$" , $_REQUEST['password'])
        ? $_REQUEST['password'] : "" ;

    // now try to authenticate these values
```



# login.php (2)

```
if (isAuthentic($id, $pass))
{
    $_SESSION['valid_user'] = $id;
    display_members_page();
}
else
{
    display_login_page("Invalid login, try again");
}
else // first time so display form to login
{
    display_login_page("Please log in");
}
?>
```



# login.php (3)

```
<?php
function isAuthenticated($id, $password)
{
    $db_link = db_connect("web_db");

    $query = "SELECT * FROM login WHERE name='$id'"
        . "AND password = PASSWORD('$password')";

    $result = mysql_query($query, $db_link);
    $valid = mysql_num_rows($result) > 0;
    mysql_free_result($result);
    mysql_close($db_link);
    return $valid;
}
?>
```



# login.php (4)

---

```
<?php
function display_login_page ($message)
{
?>
<html>
<head><title>Members Login Page</title></head>
<body>
<h1>Login Page</h1>
<h2><?php echo $message ?></h2>
<form method="POST">
<table border="1">
<tr><td>
```



# login.php (5)

```
<table border="0">
<tr><td>User Name:</td>
<td><input type="text" name="userid"></td></tr>
<tr><td>Password:</td>
<td><input type="password" name="password"></td>
</tr>
<tr><td colspan=2 align=center>
<input type="submit" value="Log in"></td></tr>
</table>
</td></tr>
</table>
</form></body></html>
<?php
}
?>
```





# login.php (6)

```
<?php
function display_members_page ()
{
?>
<html><head><title>Members Page</title></head>
<body>
You have successfully logged in as user
<strong><?php echo $_SESSION['valid-user']?></strong>
<p>
<a href="members.php?<?php echo SID?>">Member
pages</a><br>
<a href="logout.php?<?php echo SID?>">Logout</a>
</p>
</body></html>
<?php } ?>
```



# logout.php

- [view script login-db/logout.php](#)

```
<?php
session_start();
unset($_SESSION['valid-user']);
session_destroy();
?>
<html>
<head><title>Logout Page</title></head>
<body>
<h1>Logout Page</h1>
If you were logged in you have been logged out
<p><a href="login.php">Login</a></p>
</body>
</html>
```



# members.php (1)

■ [view script login-db/members.php](#)

```
<?php
session_start();
if (! isset($_SESSION['valid-user']))
{ ?>
<html><head><title>Login Error</title></head>
<body>
<h1>Login Error</h1>
You are not authorized to view this page, please
  <a href="login.php?<?php echo SID?>">login</a>
</body>
</html>
<?php
exit();
}
?>
```



# members.php (2)

```
<html>
<head><title>Member Page</title></head>
<body>
<h1>Member Page</h1>
This is a member page.<br>
You are logged in as user
<strong><?php echo $_SESSION['valid_user']?></strong>.
<p><a href="logout.php?<?php echo SID?>">
Logout</a></p>
</body>
</html>
```