

PHP Programming (IV)

Res. Asst. Sanem Sariel
sariel@itu.edu.tr

http://www2.itu.edu.tr/~sariel/course_notes/dbms2007/p4/slides.pdf
http://www2.itu.edu.tr/~sariel/course_notes/dbms2007/p4/codes/p4_codes.zip

2/26/2007

DBMS 2007, Sanem Sariel

1

Content

- Database Operations with PHP
 - Connection
 - Database Queries
 - Processing Query Results
- User Identification and Security with PHP
 - Cookies
 - Sessions

2/26/2007

DBMS 2007, Sanem Sariel

2

MYSQL Database Connection

- `$db_conn = mysql_connect($db_host, $db_uname, $db_passwd)`
 - ...
 - /* Database operations */
 - ...
 - `mysql_close($db_conn)`
- `$db_host` : the host name that the database resides on
`$db_uname` : username to connect the database
`$db_passwd` : password to connect the database
`$db_conn` : MYSQL link identifier or failure

2/26/2007

DBMS 2007, Sanem Sariel

3

Selecting MYSQL Database

- `$db_succ = mysql_select_db($db_name, $db_conn)`
- `$db_name`: the name of the database that is to be selected
- `$db_conn`: MYSQL link identifier (optional)
- `$db_succ` : true on success

```
$db_conn = mysql_connect('localhost',  
'root', 'knoppix') or die ('Connection  
cannot be established!');  
  
$db_name = 'movies';  
$db_selected = mysql_select_db($db_name);  
if (!$db_selected)  
    die ('Can\'t use $db_name: ' .  
    mysql_error());
```

2/26/2007

DBMS 2007, Sanem Sariel

4

Flexible Code Generation

- Host, Database name and User information can be changed
 - Create separate accessible files for database and other global variables
 - `config_db.php`
 - `open_db.php`
 - `close_db.php`
- Create a library of common functions and global variables

2/26/2007

DBMS 2007, Sanem Sariel

5

Configuration and Connection Files

- Include the configuration and connection files

```
<?php  
$db_host = 'localhost';  
$db_uname = 'root';  
$db_passwd = 'knoppix';  
$db_name = 'movies';  
?>
```

"config_db.php"

```
<?php  
$db_conn = mysql_connect($db_host, $db_uname, $db_passwd) or  
die ('Connection cannot be established!');  
mysql_select_db($db_name);  
?>
```

"open_db.php"

```
<?php  
mysql_close($db_conn);  
?>
```

"close_db.php"

2/26/2007

DBMS 2007, Sanem Sariel

6

Including the Configuration and Connection Files

```
<?php
require_once ('config_db.php');
require_once ('open_db.php');

/*
    ...Your code goes here...
    Database queries and other operations
*/

require_once ('close_db.php');
?>
```

"main_db.php"

2/26/2007

DBMS 2007, Sanem Sariel

7

MYSQL Database Queries

- **\$query_result =**
`mysql_query($query_string, $db_conn)`
- **\$query_string**: any type of query on database
 - can also be read from a file (e.g., SQL commands for creating tables)
- **\$db_conn**: MYSQL link identifier (optional)
- **\$query_result**: returns
 - **SELECT, SHOW or DESCRIBE**
 - Resource on success
 - **INSERT, UPDATE, DELETE, DROP, etc.,**
 - True ON success

2/26/2007

DBMS 2007, Sanem Sariel

8

Queries and MYSQL Functions

- `$query_result = mysql_create_db($db_name);`
- `$db_name= 'movies';`
- `$query_string= 'CREATE DATABASE '. $db_name;`
- `$query_result= mysql_query($query_string);`

- `$query_result= mysql_drop_db($db_name);`
- `$query_string= 'DROP DATABASE '. $db_name;`
- `$query_result= mysql_query($query_string);`

create_table.php

2/26/2007

DBMS 2007, Sanem Sariel

9

Processing Data from Database

- `$result_array = mysql_fetch_array($query_result, $result_type);`
 - **\$result_type**:
 - **MYSQL_ASSOC** -> associative indices
 - **MYSQL_NUM** -> number indices
 - **MYSQL_BOTH** -> both types of indices
- `$result_array = mysql_fetch_assoc($query_result);`
- `$result_array = mysql_fetch_row($query_result);`
- **These commands also move the internal data pointer ahead**

select.php

2/26/2007

DBMS 2007, Sanem Sariel

10

Processing Data from Database, Cont.

```
mysql_query("SELECT id, name FROM tb_phbook");
...
while ($row = mysql_fetch_array($result, MYSQL_NUM)) {
    printf("ID: %s Name: %s", $row[0], $row[1]);
}
while ($row = mysql_fetch_row($result)) {
    printf("ID: %s Name: %s", $row[0], $row[1]);
}
while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {
    printf("ID: %s Name: %s", $row["id"], $row["name"]);
}
while ($row = mysql_fetch_assoc($result)) {
    printf("ID: %s Name: %s", $row["id"], $row["name"]);
}
while ($row = mysql_fetch_array($result, MYSQL_BOTH)) {
    printf("ID: %s Name: %s", $row["id"], $row[1]);
}
...

```

2/26/2007

DBMS 2007, Sanem Sariel

11

Processing Data from Database – Utilities

- `$row = mysql_fetch_row($query_result);`
 - `$row[0], $row[1]`
- `$list($id, $name) = mysql_fetch_row($query_result);`
 - `$id, $name`
- `$num_rows = mysql_num_rows($query_result);`
- `$num_fields = mysql_num_fields($query_result);`
- `mysql_free_result ($query_result)`: to free large result data set

form_database.htm

2/26/2007

DBMS 2007, Sanem Sariel

12

Cookies

- Small bits of data that are stored by a web browser on the user's hard drive.
- Set and read by the web server.
- Allow users to be "branded" by bits of data, so that web sites can identify repeat visitors
- Only contain data that the user supplies to a web site.

2/26/2007

DBMS 2007, Sanem Sariel

13

Setting and Reading Cookies

- Cookies are set by the web server
`setcookie('CookieName', 'CookieData');`
- Cookies are stored in the Global array `$_COOKIE`.
- Cookie values are accessed as `$_POST` variables:
`echo $_COOKIE['CookieName'];`
- A cookie isn't available within the same script that sent it because it is read from the user's browser when the page is loaded.
- Developers should set up their browsers to prompt for acceptance of cookies, in order to view the cookie content.
- `setcookie` call should be placed prior to any output.

2/26/2007

DBMS 2007, Sanem Sariel

14

Other Cookie Parameters

- `setcookie('name', 'value', expire, 'path', 'domain', secure, httponly);`
- `expire`: determines how long a cookie lives.
 - `setcookie('name', 'value', time() + 3600, '', '', 0,1);`
- `path`: denotes the folder in which a cookie is valid
- `domain`: denotes the domain for which a cookie is valid
- `secure`: determines whether the connection is secured
- `httponly`: determines whether the cookie is accessible only through HTTP protocol
- `setcookie('CookieName', '')` deletes the cookie

cookie.php

2/26/2007

DBMS 2007, Sanem Sariel

15

Sessions

- Sessions allows to store information about users.
- Sessions, store the data on the server.
- Sessions are more secure.
 - no data transmitted back and forth
- Sessions can store more data.
- Sessions can work even if user has cookies turned off.

2/26/2007

DBMS 2007, Sanem Sariel

16

Sessions in PHP

- Sessions are started by `session_start()` function
 - Must be called prior to anything being output to the browser
- Sends a cookie to the browser called `PHPSESSID` with the session ID which serves as a reference to the session and its data

2/26/2007

DBMS 2007, Sanem Sariel

17

Using Session Values

- Assign values to the session using the `$_SESSION` array.
`$_SESSION['uname'] = 'root';`
- Each time values are assigned to the array, a temporary file on the server is updated with the data.
- Values stored in the `$_SESSION` array are accessed just like other array values:
`echo $_SESSION['uname'];`

2/26/2007

DBMS 2007, Sanem Sariel

18

Deleting Session Values

- Session key and value pairs can be deleted using:

```
unset($_SESSION['uname']); // destroys the  
"uname" key and value
```

```
unset($_SESSION); // destroys all key and  
value pairs
```

```
session_destroy(); destroys all data  
registered to the current session
```

```
session_page1.php
```