

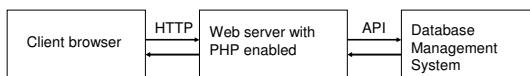
## IT420: Database Management and Organization

PHP and MySQL  
(Chapter 11 – PHP and MySQL  
Web Development)

## Last Time: PHP

- Arrays
  - Numerically indexed
  - Associative
  - Sort
- Files
  - Functions to work with files
  - File locking
- Functions
  - Optional/variable number of parameters
  - Variable scope

## Web Database Architecture



## Why Use DBMS?

## Goals Today

- Connect from PHP to MySQL

## MySQL

- Relational Database Management System
- Free
- Open source
- Portable
- High performance
- Support available

## Example Application



```
_(standard header stuff)
<html>
<head>
<title> Music search page </title>
</head>
<body>
<h1> Search for songs </h1>
<form action = "process_search.php"
      method = "post">
  <p>Enter search keyword: <input name
    = "searchterm" type = "text">
  <br/>
  <input type = "submit" value =
    "Search">
</p>
</form>
</body>
</html>
```

Database: dbmusic  
Table: songs(ISBN, Title, SingerID, Length)

## Use a database from web

- Check and filter data coming from user
- Connect to the database to use
- Send queries and retrieve results
- Process results
- Close connection
  
- All PHP functions return '*false*' if operation unsuccessful!

### process\_search.php

```
<?php
  $pageTitle = "Music Search Results";
  include('header.inc.php');
?>
<h1>Search results</h1>

<?php
$searchterm = $_POST['searchterm'];

//check input
$searchterm = trim($searchterm);
if (!$searchterm){
  echo '<p>No search term entered. Go back and try again</p>';
  include('footer.inc.php');
  exit;
}
if (!get_magic_quotes_gpc()){
  $searchterm = addslashes($searchterm);
}
```

```
process_search.php

//connect
$db = new mysqli('localhost', 'dbmusicwebuser', 'user123', 'dbmusic');
if ($mysqli_connect_errno()){
  echo "<p>Error: Could not connect to database.</p>";
  include('footer.inc.php');
  exit;
}

//construct query
$query = "select * from songs where Title like '%$searchterm%'";
//query
$results = $db->query($query);

//process results
if ($results){
  $numRows = $results->num_rows;
  echo "<p>Number of songs found: $numRows</p>";
  for ($i = 0; $i < $numRows; $i++){
    $row = $results->fetch_assoc();
    echo "<p>Title: '$row['Title']'</p>";
  }
  //free result
  $results->free();
}

//close connection
$db->close();
include('footer.inc.php');?>
```

## Sample Run



## Check Modification Results

`$results = $db->query($someQuery)`

- `$db->affected_rows`
- `$results === TRUE`

## String Manipulation Functions

- `string strip_tags(string stringVar [,string allowableTags])`
- Example:
  - `$inputStr = '<script> alert("hi"); </script>';`
  - Should not store this in the db!
  - `echo strip_tags($inputStr); //result: alert("hi");`

## SQL Injection Attacks!

- **SQL injection attack** occurs when data from the user is used to modify a SQL statement
- Example: users are asked to enter their alpha into a Web form textbox
  - User input: 081234 OR TRUE  
`SELECT * FROM STUDENT_GRADES WHERE Alpha = 081234 OR TRUE;`
  - Result?

## Class Exercise - dbmusic

- `songs(ISBN:varchar(25), Title:varchar(120), SingerID:int, Length:int)`
- Write PHP code to insert new songs in the database, based on user input from a form (POST method):
  - ISBN
  - Title
  - SingerID
  - Length

(extra space)

## PHP – DB Summary

- Check and filter data coming from user
- Connect to the database to use
- Send queries and retrieve results
- Process results
- Free results
- Close database connection

Always check and handle errors!