

DB DRIVEN WEBSITES

MySQL, PHP, AND MVC OVERVIEW

2010

Τεχνολογίες και Προγραμματισμός εφαρμογών στον Ιστό
Οικονομικό Πανεπιστήμιο Αθηνών

DB Driven websites

2

- Αντί να αποθηκεύουμε το περιεχόμενο του Web site σε αρχεία είναι προτιμότερο να το αποθηκεύουμε σε "αποθήκες δεδομένων" (XML, ΒΔ, κτλ)
- Έπειτα συνδυάζουμε το περιεχόμενο με html & css χρησιμοποιώντας template για την δημιουργία της σελίδας
 - Ο διαχωρισμός του **κώδικα, παρουσίασης**, και **δεδομένων** είναι θεμιτός στόχος και δεν περιορίζεται στον προγραμματισμό σε εξυπηρετητές
 - Η χρήση ΒΔ δίνει τη δυνατότητα δημιουργίας δυναμικών ή και εξατομικευμένων σελίδων
 - προτιμήσεις χρηστών, προσφορές σε e-commerce, προσαρμοσμένες αναφορές, portals κτλ
 - Η χρήση ΒΔ μας δίνει τη δυνατότητα να δημιουργούμε συστήματα διαχείρισης περιεχομένου (Content Management Systems, CMS)

DB Driven websites

2

- Πότε δημιουργούμε τις σελίδες σε ένα site που χρησιμοποιεί ΒΔ (DB driven sites);
 - Όταν ζητούνται; Πριν ζητηθούν;
 - Η δυνατότητα της δημιουργίας σελίδων εκ των προτέρων εξαρτάται από το πόσο μεταβάλλεται το περιεχόμενό τους.

Συστατικά

4

- Βάση δεδομενων (Oracle, MS SQL Server, Sybase, DB2, MySQL κτλ)
- Middleware/ Προγραμματιστικό περιβάλλον (PHP, CGI/Perl, ASP/ASP.NET, CFM, JSP κτλ)
- Templating Environment
 - Μπορεί να είναι μέρος του Middleware/Προγραμματιστικού περιβάλλοντος

Συστατικά

5

- APIs και Drivers για τη σύνδεση με την βάση
 - ▣ Διαφάνεια ως προς τις λεπτομέρειες υλοποίησης
 - Ερώτηση: Πόσο συχνά θα αλλάζουν τα δεδομένα;
 - ▣ Drivers
 - ODBC, native drivers
 - ADO.NET, JDBC
- ▣ Εδώ μπορεί να γίνει και η διασύνδεση των πινάκων της ΒΔ με αντικείμενα

Διαχείριση ΒΔ

- Ένας πολύ απλούστερος τρόπος για την διαχείριση ΒΔ είναι με την χρήση του phpmyadmin

The screenshot displays the phpMyAdmin web interface. On the left is a sidebar with the '1&1' logo and a tree view showing the database 'db154377598' and its table 'users'. The main panel shows the 'Structure' tab for the 'users' table. At the top, it indicates the server is 'db299.perfora.net' and the database is 'db154377598'. Below this are navigation buttons: Browse, Structure (selected), SQL, Search, Insert, Export, Operations, Empty, and Drop. The table structure is shown in a table with columns: Field, Type, Attributes, Null, Default, Extra, and Action. The fields are: user_id (mediumint(8), UNSIGNED, No, auto_increment), first_name (varchar(35), No), last_name (varchar(35), No), login (varchar(16), No), and password (varchar(16), No). Below the table structure are buttons for 'Check All / Uncheck All' and 'With selected:'. There are also links for 'Print view' and 'Propose table structure'. Below these are options to 'Add' fields, with a dropdown set to 'user_id' and a 'Go' button. The 'Indexes' section shows a 'PRIMARY' index on 'user_id'. The 'Space usage' section shows 'Data' (44 Bytes), 'Index' (2,048 Bytes), and 'Total' (2,092 Bytes). The 'Row Statistics' section shows 'Statements' (dynamic), 'Rows' (1), 'Row length' (44), 'Row size' (2,092 Bytes), 'Need AutoIndex' (2), 'Creation' (Feb 14, 2006 at 01:25 PM), and 'Last update' (Feb 14, 2006 at 01:25 PM). At the bottom, there is a text area for SQL queries with the text 'SELECT * FROM `users` WHERE 1', a 'Fields' list on the right, and a 'Go' button. A checkbox 'Show this query here again' is also present.

Field	Type	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> user_id	mediumint(8)	UNSIGNED	No		auto_increment	
<input type="checkbox"/> first_name	varchar(35)		No			
<input type="checkbox"/> last_name	varchar(35)		No			
<input type="checkbox"/> login	varchar(16)		No			
<input type="checkbox"/> password	varchar(16)		No			

Keyname	Type	Cardinality	Action	Field
PRIMARY	PRIMARY	1		user_id

Type	Usage
Data	44 Bytes
Index	2,048 Bytes
Total	2,092 Bytes

Statements	Value
Format	dynamic
Rows	1
Row length	44
Row size	2,092 Bytes
Need AutoIndex	2
Creation	Feb 14, 2006 at 01:25 PM
Last update	Feb 14, 2006 at 01:25 PM

```
SELECT * FROM `users` WHERE 1
```

Fields: user_id, first_name, last_name, login, password

☒ Show this query here again

Χρήση PHP & MySQL

18

```
define('DB_USER', 'root');
define('DB_PASSWORD', 'password here');
define('DB_HOST', 'localhost');
define('DB_NAME', 'mydatabase');

$dbc = mysql_connect(DB_HOST, DB_USER, DB_PASSWORD)
        OR die('Cound not connect to MySQL:' . mysql_error());
@mysql_select_db(DB_NAME)
        OR die('Could not select database:' . mysql_error());
```

- Αρχικά πρέπει να συνδεθούμε με την συνάρτηση **mysql_connect** και να συνδεθούμε σε μια βάση με την **mysql_select_db**
 - ▣ Χρήση σταθερών για μεγαλύτερη ευκολία
 - ▣ Έλεγχος για σφάλματα για αποφυγή προβλημάτων
- Οι νεότερες εκδόσεις της PHP υποστηρίζουν και **αντικειμενοστραφή προσέγγιση**

Χρήση PHP & MySQL

10

□ Δημιουργία πινάκων

```
$create_table = "CREATE TABLE users (  
    user_id MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    first_name VARCHAR(35) NOT NULL,  
    last_name VARCHAR(35) NOT NULL, login CHAR(16) NOT NULL,  
    password CHAR(16) NOT NULL, PRIMARY KEY (user_id));"  
$result = @mysql_query($create_table);  
echo "Database create: $result <br />";
```

□ Εισαγωγή δεδομένων:

```
$insert_statement = "INSERT INTO users  
(first_name,last_name,login,password)  
VALUES ('Vasilis','Vassalos','vassalos','supersecret');"  
$result = @mysql_query($insert_statement);  
echo "Database insert: $result <br />";
```


Select Query στη PHP

20

```
$query = 'SELECT login,first_name,last_name FROM users ORDER BY
login';
$result = mysql_query($query);
$numAccts = mysql_num_rows($result);
echo '<pre>';
if ($numAccts == 0) {
    echo 'No accounts';
} else {
    while ($row = mysql_fetch_array($result,MYSQL_NUM)) {
        echo "$row[0]\t\t$row[1]\t\t$row[2]\n";
    }
}
echo '</pre>';
```

Μετά την επιστροφή των πλειάδων από το query τις εμφανίζουμε μια – μια μέσα σε ένα βρόχο



Templating

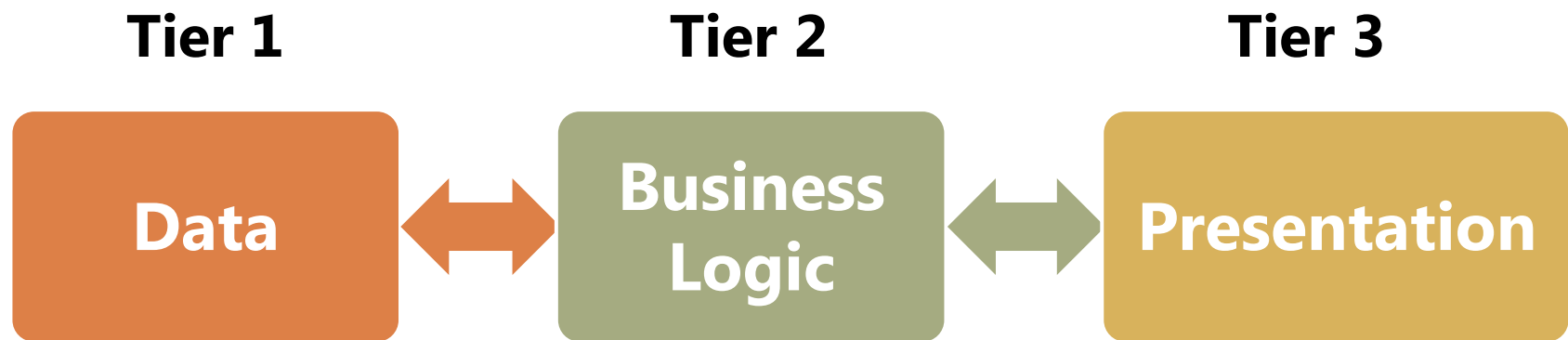
Αρχιτεκτονική Web Τριών Επιπέδων

26

- Υπάρχουν πολλά πλεονεκτήματα από την χρήση αρχιτεκτονικών τριών επιπέδων (3-tier Web site/application)
 - ▣ Κάθε επίπεδο μπορεί να εκτελείται σε διαφορετικό μηχάνημα, παρέχοντας **καλύτερη απόδοση**
 - ▣ Καλύτερη δομή που οδηγεί σε **ευκολότερη συντήρηση**
 - ▣ Καλύτερος διαχωρισμός αρμοδιοτήτων – εργασίας για κάθε επίπεδο που οδηγεί ξανά σε βελτιωμένη συντήρηση
- Φυσικά όπως κάθε αρχιτεκτονική μπορεί να δημιουργήσει προβλήματα αν δεν εφαρμοστεί σωστά ή αν δεν είναι απαραίτητη
 - ▣ Το πρόβλημα του "cargo cult" είναι σύνηθες κατά τον προγραμματισμό

Αρχιτεκτονική Web Τριών Επιπέδων

27



Αρχιτεκτονική Web Τριών Επιπέδων με PHP

2.0

Server-side

Presentation

Template
display code

Business Logic

Database
abstraction layer

**Persistent
Data Storage**

Main PHP script
Logic and flow of
the application

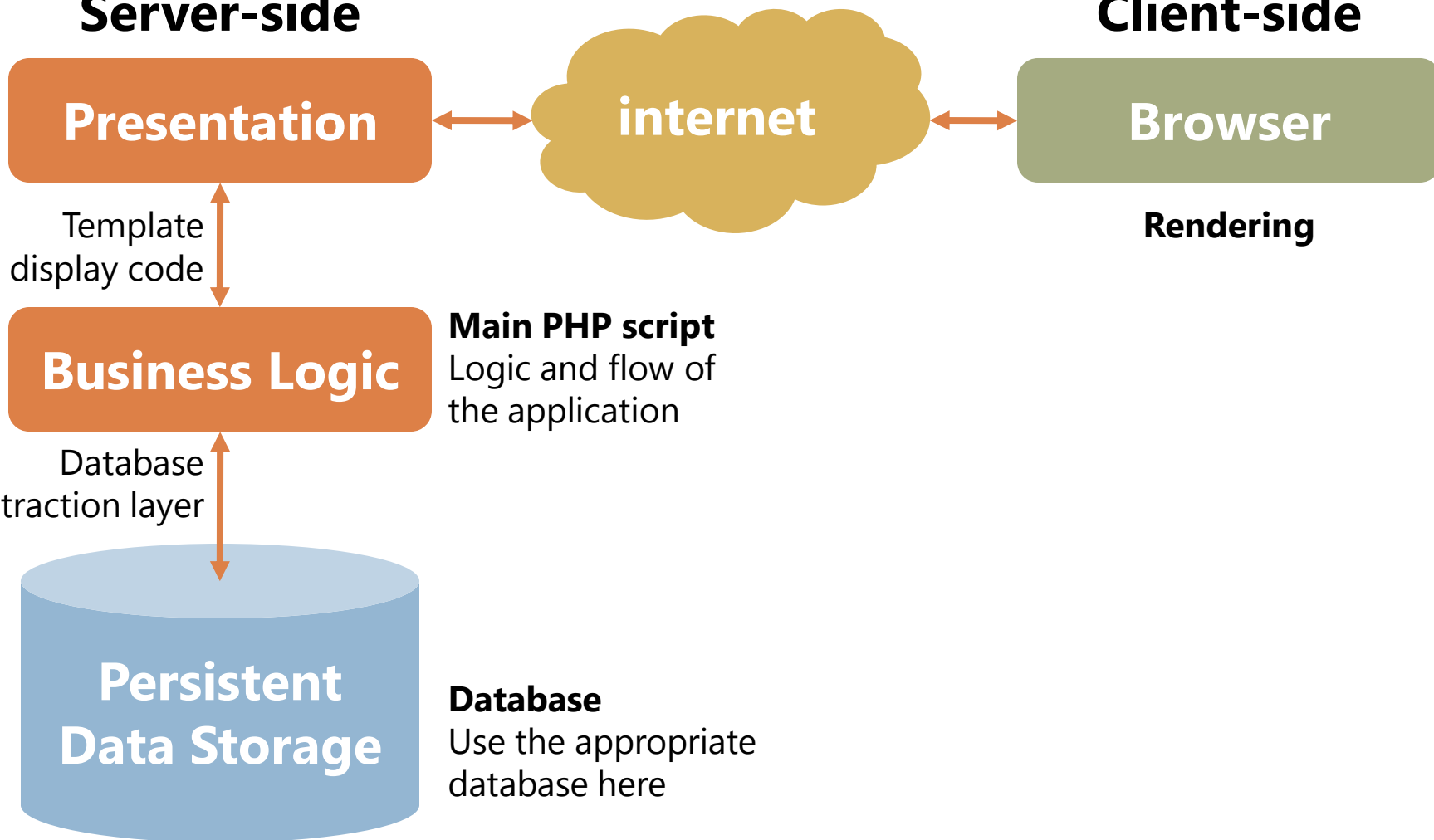
Database
Use the appropriate
database here

internet

Client-side

Browser

Rendering



Database Abstraction Layer

30

- Για την PHP είναι διαθέσιμο το παρακάτω:
 - ▣ <http://pear.php.net/package/DB>
- Το πλεονέκτημα είναι ότι παρέχεται διαφάνεια ως προς τις λεπτομέρειες της ΒΔ
 - ▣ Μεταφερσιμότητα
 - Εύκολη αλλαγή της ΒΔ
 - Ευκολότερη διανομή του κώδικα
 - ▣ Παρέχει αντικειμενοστρέφεια στην σύνδεση σε ΒΔ με χρήση PHP

Παράδειγμα Template: Smarty

31

- Υπάρχουν πολλά templating engines για την PHP με το πιο δημοφιλές να είναι το [Smarty](#)
- Λόγοι για να χρησιμοποιήσετε templating engine:
 - ▣ Διαχωρισμός του κώδικα από το template
 - ▣ Caching των templates, white space stripping
 - ▣ Αποφυγή της PHP στο επίπεδο της παρουσίασης, χρήση μιας λιγότερο ισχυρής γλώσσας και έτσι βελτίωση της ασφάλειας (νομίζετε ότι αυτό ισχύει;)
 - ▣ Άλλοι: <http://www.smarty.net/rightforme.php>
- Σημείωση: Υπάρχουν επίσης βιβλιοθήκες για εξειδικευμένη έξοδο σε HTML όπως <http://phphtmlib.newsblob.com/>

Παράδειγμα Smarty

22

PHP

```
<?php
require './smarty/libs/Smarty.class.php'; // Path to Smarty
$smarty = new Smarty;
$smarty->template_dir = './templates/'; // Path to templates
$smarty->assign('Username', 'Mr. Smarty');
$smarty->display('index.tpl');
?>
```

Smarty
template

```
<!doctype html>
<html>
<head>
    <title>Hello Smarty</title>
</head>
<body>
    Hello world {$Username}!
</body>
</html>
```


Smarty DB Example

22

```
<?php
```

```
require ('../smarty/libs/Smarty.class.php');
```

```
$smarty = new Smarty;
```

```
$smarty->template_dir = '../templates/';
```

```
// Σύνδεση στη MySQL και επιλογή βάσης [παραλείπεται]
```

```
$query = 'SELECT login,first_name, last_name FROM users';
```

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

```
$users = array();
```

```
while ($auser = mysql_fetch_array($result)) {
```

```
    $users[] = $auser;
```

```
}
```

```
$smarty->assign('results', $users);
```

```
$smarty->display('displayusers.tpl');
```

```
?>
```

displayusers.tpl

34

```
<!DOCTYPE html>
<html>
<head>
    <title>List users with Smarty</title>
    <meta charset="utf-8" />
</head>
<body>
    <table>
        <caption>Users</caption>
        <tr><th>Login</th><th>Name</th><th>Surname</th></tr>
        {section name=nr loop=$results}
            <tr {if $smarty.section.nr.iteration is odd}
                                                style="background:#eee"{/if}>
                <td>{$results[nr].login}</td>
                <td>{$results[nr].first_name}</td>
                <td>{$results[nr].last_name}</td>
            </tr>
        {/section}
    </table>
</body>
</html>
```

Φαίνεται ενδιαφέρον αλλά...

25

- Αρκετοί προγραμματιστές είναι διστακτικοί στο να χρησιμοποιήσουν συστήματα για templates όπως το Smarty
 - ▣ Πρέπει να μάθουν καινούργια γλώσσα (σύνταξη)
- Μπορούμε να τα καταφέρουμε μόνο με PHP;
 - ▣ Φυσικά και μπορούμε! Παράδειγμα: Wordpress
 - ▣ Συνήθως σε PHP templates, για λόγους αναγνωσιμότητας προτιμάται η εναλλακτική γραφή για loops και short opening/closing tags (`<? ... ?>` και `<?= ... ?>`)

Το ίδιο παράδειγμα, χωρίς Smarty

36

```
<?php
```

```
// Σύνδεση στη MySQL και επιλογή βάσης [παραλείπεται]
```

```
$query = 'SELECT login,first_name, last_name FROM users';  
$result = mysql_query($query);
```

```
$users = array();  
while ($auser = mysql_fetch_array($result)) {  
    $users[] = $auser;  
}
```

```
// Now pass the results to the template  
include './templates/displayusers.tpl' ;
```

```
?>
```

displayusers.tpl χωρίς Smarty

37

```
<!DOCTYPE html>
<html>
<head>
    <title>List users with Smarty</title>
    <meta charset="utf-8" />
</head>
<body>
    <table>
        <caption>Users</caption>
        <tr><th>Login</th><th>Name</th><th>Surname</th></tr>
        <? foreach($results as $nr => $user): ?>
            <tr <? if($nr % 2 == 1): ?> style="background:#eee"<? endif; ?>>
                <td><?= $user['login'] ?></td>
                <td><?= $user['first_name'] ?></td>
                <td><?= $user['last_name'] ?></td>
            </tr>
        <? endforeach; ?>
    </table>
</body>
</html>
```

Συμπεράσματα

38

- “... *the point of template engines should be to separate your business logic from your presentation logic, not separate your PHP code from your HTML code.*” —http://www.massassi.com/php/articles/template_engines/
- Παρόλα αυτά η ιδέα των templates είναι σχετικά καλή ιδέα, άσχετα με το πως υλοποιείται
 - ▣ Ευκολότερες αλλαγές
 - ▣ Διαχωρισμός της δουλειάς των προγραμματιστών και των σχεδιαστών



Βάσεις δεδομένων & αντικείμενα

Από τους πίνακες στα αντικείμενα

□ ORM

- *“Object-Relational Mapping (aka ORM, O/RM, and O/R mapping) is a programming technique for converting data between incompatible type systems in relational databases and object-oriented programming languages.” —[Wikipedia](#)*
- Υπάρχουν [αρκετές υλοποιήσεις](#)

ORM: Παράδειγμα

42

- Στην Cake ας υποθέσουμε ότι έχουμε ένα πίνακα "Pets" με στήλες name, type, breed, color και ένα πρωτεύον κλειδί.
- Ένα σύστημα ORM θα δημιουργήσει μια αντιστοίχιση (mapping) σε ένα αντικείμενο με όνομα "Pet" και παρόμοιες ιδιότητες
- **Ερώτηση:** Πώς αντιστοιχίζεται το πρωτεύον κλειδί;
 - **Απάντηση:** Με τη χρήση ενός "Id"

Scaffold

43

- Στον προγραμματισμό και στο Web το scaffolding είναι γενικός όρος που περιγράφει frameworks που διαχειρίζονται πίνακες ΒΔ και διάφορα αρχεία ρυθμίσεων (config files)
- Για παράδειγμα στην Ruby δημιουργούνται scaffolds ως εξής:
 - ▣ `ruby script\generate model Pet`
 - ▣ `ruby script\generate controller Pet`
 - ▣ ή απλά `ruby script\generate scaffold Pet`

Scaffold

- Με το Scaffold είναι διαθέσιμες σελίδες παρουσίασης, διαγραφής, αλλαγής και εισαγωγής δεδομένων στην ΒΔ
- Μια αλλαγή στη ΒΔ μπορεί να γίνει γρήγορα αντιληπτή από την εφαρμογή
 - ▣ Είτε αυτόματα είτε με ένα trigger για να ξαναχρησιμοποιηθεί το Scaffold
- Τώρα αυτό που απομένει είναι η βελτίωση του γραφικού ώστε να είναι εμφανίσιμο αλλά και εισαγωγή οποιαδήποτε άλλης προσαρμοσμένης επιχειρηματικής λογικής

Συμπεράσματα

45

- Τελικά κάνουμε το ίδιο μέγεθος δουλειάς;
 - ▣ Για να ετοιμάσουμε τα templates
 - ▣ Για να κάνουμε κάτι πρωτότυπο
- Ένα επιχείρημα υπέρ των εφαρμογών που δημιουργούνται με αυτά τα συστήματα είναι ότι θα είναι πιθανώς γνωστά και σε κάποιο άλλο προγραμματιστή που γνωρίζει το σύστημα, όμως και πάλι σε αυτή τη περίπτωση έχουμε διαθέσιμα τα coding standards
 - ▣ Τελική ερώτηση: Τι συμβαίνει όταν δεν γνωρίζουμε το πρότυπο;