



# ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

---

## Προγραμματισμός στο Διαδίκτυο

### ΡΗΡ και Διασύνδεση με ΒΔ

Μανώλης Μαραγκουδάκης

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

---



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



## Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



## Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αιγαίου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

# Προγραμματισμός στο Διαδίκτυο

## Ενότητα 5γ

### PHP και Διασύνδεση με ΒΔ

Μανώλης Μαραγκουδάκης

Πανεπιστήμιο Αιγαίου

Τμήμα Μηχανικών Πληροφοριακών και  
Επικοινωνιακών Συστημάτων

# **ΕΙΣΑΓΩΓΙΚΑ ΓΙΑ ΒΔ**

# Εισαγωγή στις Βάσεις Δεδομένων / SQL



- Σχεσιακές Βάσεις Δεδομένων ή Αρχεία?
- Πλεονεκτήματα αποθήκευσης δεδομένων σε σχεσιακές βάσεις δεδομένων αντί σε αρχεία κειμένου:
  - Γρηγορότερη πρόσβαση στα δεδομένα
  - Δημιουργία ερωτημάτων/συνθηκών για την «εξαγωγή» δεδομένων
  - Ενσωματωμένος μηχανισμός για το χειρισμό ταυτόχρονης πρόσβασης
  - Ενσωματωμένα συστήματα δικαιωμάτων



## Πίνακας Customer

CustomerID	Name	Address	City
1	Κώστας Νικολάου	Παλαμά 15	Πάτρα
2	Πέτρος Πετρίδης	Κορνάρου 32	Τρίπολη
3	Παύλος Πέτρου	Παλαμά 35	Αθήνα

- Στήλες ή πεδία ή ιδιότητες
- Γραμμές ή εγγραφές
- Τιμές

# Κλειδιά



CUSTOMERS

Πρωτεύοντα κλειδιά

CustomerID	Name	Address	City
1	Παπαδόπουλος Στέλιος	Δημοκρατίας 4	Αθήνα
2	Μιρέλλη Μαρία	Βικέλα 37	Ερμούπολη
3	Σώκος Πέτρος	Πηλίου 10	Βόλος

ORDERS

Ξένο κλειδί

OrderID	CustomerID	Ammount	Date
1	3	32.90	2/2/2003
2	1	25.00	12/3/2003
3	2	20.10	1/4/2001
4	1	75.80	3/4/2003



- Ένα – προς – Ένα (π.χ. Αν είχαμε βάλει τους πελάτες και τις διευθύνσεις τους σε ξεχωριστούς πίνακες)
- Ένα προς Πολλά (π.χ. Customers – Orders)
- Πολλά – προς – Πολλά (π.χ. πίνακες Books – Authors)



# SQL - SELECT



Σύνταξη:

```
SELECT column_name [, column_name] ...  
FROM table_name
```

Παραδείγματα:

- SELECT name, comment FROM Students
- SELECT \* FROM Students

# SQL - WHERE



Σύνταξη:

*SELECT column FROM table*

*WHERE column operator value*

όπου *operator*: =, <>, >, <, >=, <=, BETWEEN, LIKE

Παράδειγμα:

*SELECT \* FROM Students WHERE id=1*

# SQL - INSERT INTO



Σύνταξη :

- `INSERT INTO table_name  
VALUES (value1, value2,....)`
- `INSERT INTO table_name (column1, column2,...)  
VALUES (value1, value2,....)`

Παραδείγματα:

- `INSERT INTO Students  
VALUES ('Apostolis',45,'Test entry')`
- `INSERT INTO Students (name,id)  
VALUES ('Apostolis2',48)`

# SQL - UPDATE



Σύνταξη :

*UPDATE table\_name*

*SET column\_name = new\_value*

*WHERE column\_name = some\_value*

Παράδειγμα:

UPDATE Students

SET comment ='New Comment'

WHERE id = 48

# SQL - DELETE



Σύνταξη :

*DELETE FROM table\_name*

*WHERE column\_name = some\_value*

Παραδείγματα:

- `DELETE FROM Students WHERE id = 48`
- `DELETE FROM Students WHERE id > 25`

# SQL – Άλλες Επιλογές



- ORDER BY:

```
SELECT * FROM Students ORDER BY name
```

- AND/OR:

```
SELECT * FROM Students
```

```
WHERE name='test' OR name='test2'
```

- IN:

```
SELECT column_name FROM table_name
```

```
WHERE column_name IN (value1,value2,..)
```

- BETWEEN / AND:

```
SELECT column_name FROM table_name
```

```
WHERE column_name BETWEEN value1 AND value2
```

# SQL - CREATE



Σύνταξη :

- `CREATE DATABASE database_name`
- `CREATE TABLE table_name (  
column_name1 data_type,  
column_name2 data_type, ..... )`

Παράδειγμα:

```
CREATE TABLE Address (  
id int(11),  
adr varchar(100),  
city varchar(50),  
PRIMARYKEY(id) )
```

# SQL – ALTER TABLE



Σύνταξη:

- `ALTER TABLE table_name`  
`ADD column_name datatype`
- `ALTER TABLE table_name`  
`DROP COLUMN column_name`

Παραδείγματα:

- `ALTER TABLE Address ADD person varchar(30)`
- `ALTER TABLE Address DROP COLUMN person`



# SQL - DROP TABLE



Σύνταξη:

```
DROP TABLE table_name
```

Παράδειγμα:

```
DROP TABLE Address
```

# CREATE USER



## Σύνταξη (MySQL)

- `CREATE USER user_specification [, user_specification] ...`
- `user_specification: user [ IDENTIFIED BY [PASSWORD] 'password' ]`
- `user : 'user_name'@'host_name'`

## Παράδειγμα:

```
CREATE USER 'jeffrey'@'localhost'  
IDENTIFIED BY 'mypass';
```

# GRANT



Σύνταξη (MySQL):

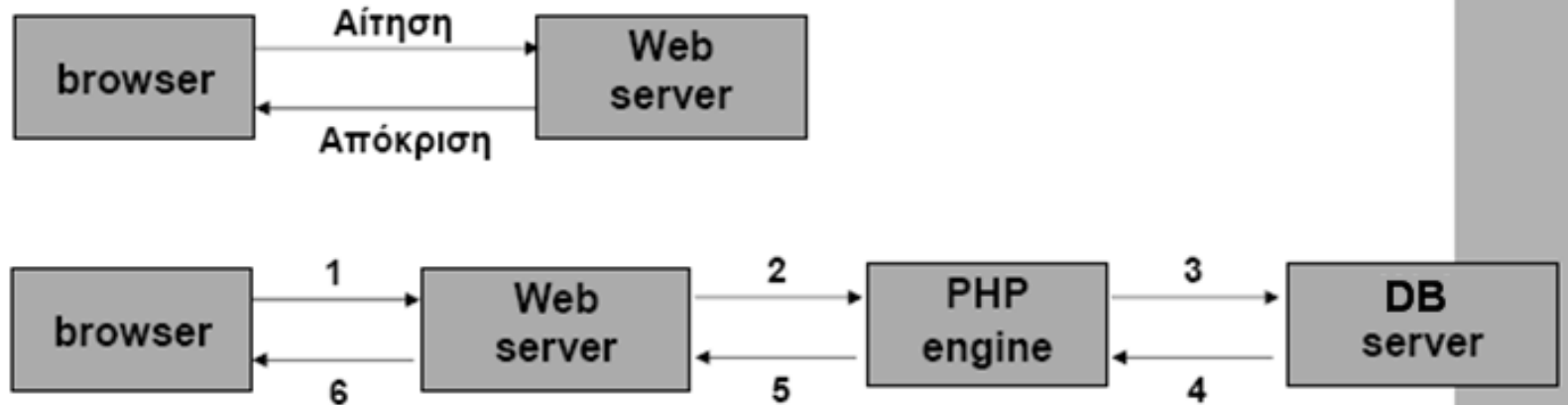
- *GRANT privilege [, privilege] ... ON db\_name.table\_name TO user\_specification [, user\_specification] ...*
- *privilege* : ALL, SELECT, INSERT, UPDATE, DELETE, INDEX, ALTER, CREATE, DROP, RELOAD, SHUTDOWN, PROCESS, GRANT, REVOKE

Παραδείγματα:

- `GRANT ALL ON db1.* TO 'jeffrey'@'localhost';`
- `GRANT SELECT ON db2.invoice TO 'jeffrey'@'localhost';`
- `GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, INDEX ON myblog.* TO 'jeffrey'@'localhost' IDENTIFIED BY 'mypass';`

# PHP ΚΑΙ ΒΔ

# Αρχιτεκτονική Web Βάσεων Δεδομένων



1. Αίτηση
2. Ο web server λαμβάνει αίτηση
3. Η PHP engine αρχίζει την ανάλυση του script, συνδέεται στον database server και στέλνει το ερώτημα
4. Ο database server επεξεργάζεται το ερώτημα και στέλνει πίσω τη πληροφορία
5. Μορφοποίηση των αποτελεσμάτων σε HTML και επιστροφή
6. Ο web server επιστρέφει την HTML σελίδα στον browser

# Υποστήριξη ΒΔ στην PHP



- Η PHP διαθέτει ενσωματωμένη υποστήριξη για τις περισσότερες ΒΔ, όπως **MySQL**, **dBase**, **FilePro**, **mSQL**, **Oracle**, **PostgreSQL**, και **Sybase**.
- Αν πρέπει να χρησιμοποιηθεί κάποιος τύπος ΒΔ που δεν υποστηρίζεται απ' ευθείας, όπως **Access** ή **SQL Server**, πρέπει να χρησιμοποιηθούν οι συναρτήσεις **ODBC** (**Open DataBase Connectivity**) της PHP μαζί με τα προγράμματα οδήγησης της συγκεκριμένης ΒΔ για να είναι δυνατή η επικοινωνία



- Πολύ δημοφιλές σύστημα διαχείρισης βάσεων δεδομένων ανοιχτού κώδικα
  - <http://www.mysql.com>
- Η MySQL διανέμεται δωρεάν ή σε πολύ χαμηλό κόστος αν πρόκειται να χρησιμοποιηθεί για εμπορικούς σκοπούς
- Ικανοποιητική απόδοση και λειτουργικότητα για εφαρμογές Παγκόσμιου Ιστού
- Συνεχής βελτίωση



## 3 τρόποι

- Βασικός τρόπος
  - php\_MySQL
  - Παραδείγματα από SQL injection
- Advanced τρόπος
  - Php\_MySQLi
    - Procedural
    - Object-Oriented
- Εναλλακτικός τρόπος
  - PDO\_MySQL: PHP Data Objects

## Βασικά βήματα

Δημιουργία σύνδεσης

Σχηματισμός ερωτήματος

Υποβολή του ερωτήματος

Επεξεργασία του αποτελέσματος

Προβολή στο χρήστη



# Βασικός Τρόπος:



## Κύριες συναρτήσεις php-mysql

- `mysql_connect()` – σύνδεση σε ΒΔ
- `mysql_pconnect()` – παραμένουσα σύνδεση
- `mysql_close()` – κλείσιμο σύνδεσης
- `mysql_create_db()` – δημιουργία νέας ΒΔ
- `mysql_drop_db()` – αφαίρεση ΒΔ
- `mysql_select_db()` – επιλογή ΒΔ ως ενεργής
- `mysql_query()` – εκτέλεση SQL query στη ΒΔ
- `mysql_fetch_array()` – επιστρέφει μια γραμμή από το σύνολο των αποτελεσμάτων που επιστρέφεται από ένα SQL query ως ένα συσχετιζόμενο πίνακα ή/και πίνακα με ακέραιους δείκτες
- `mysql_fetch_row()` – όμοιο με `mysql_fetch_array()` αλλά με πίνακα με ακέραιους δείκτες
- `mysql_affected_rows()` – αριθμός γραμμών που επηρεάστηκαν από το SQL query
- `mysql_result()` – επιστρέφει το περιεχόμενο ενός κελιού του πίνακα
- `mysql_list_dbs()` – λίστα των διαθέσιμων ΒΔ

# Βασικές συναρτήσεις php-mysql



`resource mysql_connect ( [string $server [, string $username [, string $password [, bool $new_link [, int $client_flags]]]] )`

- Συνδέεται σε έναν MySQL Server.
- Επιστρέφει το αναγνωριστικό της σύνδεσης ή **FALSE** σε περίπτωση αποτυχίας

`bool mysql_select_db ( string $database_name [, resource $link_identifier] )`

- Επιλέγει μια ΒΔ.
- Επιστρέφει **TRUE** σε επιτυχία ή **FALSE** σε αποτυχία

`bool mysql_close ( [resource $link_identifier] )`

- Κλείνει την σύνδεση με τον MySQL server.
- Επιστρέφει **TRUE** σε επιτυχία ή **FALSE** σε αποτυχία.

# Βασικές συναρτήσεις php-mysql



resource **mysql\_query** ( string \$query [, resource \$link\_identifier] )

- Υποβάλει ένα ερώτημα στην ΒΔ.
  - Στην περίπτωση SELECT ερωτήματος επιστρέφει ένα resultset υπό τη μορφή resource σε περίπτωση επιτυχίας ή **FALSE** σε περίπτωση αποτυχίας.
  - Στην περίπτωση UPDATE, DELETE, INSERT ερωτήματος επιστρέφει **TRUE** σε περίπτωση επιτυχίας ή **FALSE** σε περίπτωση αποτυχίας.
- mixed **mysql\_result** ( resource \$result, int \$row [, mixed \$field] )
    - Επιστρέφει τα αποτελέσματα του ερωτήματος στη γραμμή row και στο πεδίο field.

# Επιπλέον συναρτήσεις php-mysql



`int mysql_affected_rows ( [resource $link_identifier] )`

- Επιστρέφει τον αριθμό των γραμμών οι οποίες επηρεάστηκαν στην τελευταία SQL εντολή.

`array mysql_fetch_array ( resource $result [, int $result_type] )`

- Επιστρέφει σε μορφή πίνακα τα αποτελέσματα της τρέχουσας γραμμής.
- Ο πίνακας μπορεί να είναι: είτε απλός (`result_type=MYSQL_NUM`), είτε συσχετιζόμενος (`result_type=MYSQL_ASSOC`), είτε και τα δύο (`result_type=MYSQL_BOTH`) όπου είναι και η προκαθορισμένη επιλογή.
- Εάν δεν υπάρχουν επιπλέον γραμμές επιστρέφει **FALSE**.

# Επιπλέον συναρτήσεις php-mysql



string **mysql\_error** ( [resource \$link\_identifier] )

- Επιστρέφει το λάθος το οποίο συνδέεται με την τελευταία MySQL λειτουργία.

int **mysql\_num\_fields** ( resource \$result )

- Επιστρέφει τον αριθμό των πεδίων στο αποτέλεσμα result.

int **mysql\_num\_rows** ( resource \$result )

- Επιστρέφει τον αριθμό των γραμμών στο αποτέλεσμα result.

bool **mysql\_free\_result** (resource \$result)

- Ελευθερώνει τους πόρους του result.
- Επιστρέφει **TRUE** σε επιτυχία ή **FALSE** σε αποτυχία

Πλήρης λίστα συναρτήσεων στο

<http://php.net/manual/en/book.mysql.php>

# Χειρισμός σφαλμάτων MySQL



- Στην προσπάθεια να χειριστούμε τα σφάλματα εκτός της συνάρτησης `mysql_error()` υπάρχουν και άλλα δύο εργαλεία της PHP.
- Το σύμβολο `@`, όταν χρησιμοποιείται μπροστά από κάποιο όνομα συνάρτησης εξαλείφει τυχόν μηνύματα σφάλματος ή προειδοποιήσεις που μπορεί να προκαλέσει η συνάρτηση αυτή.
- Η συνάρτηση `die()` τερματίζει την εκτέλεση ενός σεναρίου και στέλνει το μήνυμα που προσδιορίζεται ως όρισμά της στον browser.

# Διασύνδεση PHP με Βάση Δεδομένων (MySQL)



Βήμα 1 – Σύνδεση με database server

```
mysql_connect($host, $username, $password);
```

Βήμα 2 – Επιλογή βάσης

```
mysql_select_db($database);
```

Βήμα 3 – Εκτέλεση SQL εντολών

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

Βήμα 4 – Ανάγνωση αποτελεσμάτων

```
$num=mysql_numrows($r);
```

```
while ($i < $num) {
```

```
    $first=mysql_result($r, $i, "first");
```

```
    $last=mysql_result($r, $i, "last");
```

```
    print "$first $last";
```

```
    $i++
```

```
}
```

```
while ($row = mysql_fetch_array ($r)) {
```

```
    $first = $row['first'];
```

```
    $last = $row['last'];
```

```
    print "$first $last";
```

```
}
```

Βήμα 5 – Τερματισμός σύνδεσης

```
mysql_close();
```

# mysql\_connect.php



```
<?php // Script mysql_connect.php
// This script connects to the MySQL server.

// Address error handling.
ini_set ('display_errors', 1);
error_reporting (E_ALL & ~E_NOTICE);

// Attempt to connect to MySQL and print out messages.
if ($dbc = @mysql_connect ('localhost', 'guest', 'guest')) {

    print '<p>Successfully connected to MySQL.</p>';

    mysql_close(); // Close the connection.

} else {

    die ('<p>Could not connect to MySQL because: <b>' . mysql_error() . '</b></p>');

}
?>
```



# Παράδειγμα PHP διασύνδεση με MySQL (create\_tables.php)



```
<?php
$user="guest";
$password="guest";
$database="test";
$con = mysql_connect('localhost',$user,$password);
@mysql_select_db($database) or die( "Unable to select database");

// create table CONTACTS
$query="CREATE TABLE contacts (id int(6) NOT NULL auto_increment,
first varchar(15) NOT NULL,
last varchar(15) NOT NULL,
phone varchar(20) NOT NULL,
mobile varchar(20) NOT NULL,
fax varchar(20) NOT NULL,
email varchar(30) NOT NULL,
web varchar(30) NOT NULL,
PRIMARY KEY (id),
UNIQUE id (id),
KEY id_2 (id))";

if ($r = mysql_query($query))
    echo "<p>Table contacts created</p>";
else { // Query didn't run.
    print ('<p>Could create the table because: <b>' . mysql_error()
        . "</b>. The query was $query.</p>");
} // End of query IF.
```

# Παράδειγμα PHP διασύνδεση με MySQL (create\_tables.php)



```
// create table BOOKS
$query="CREATE TABLE books (isbn varchar(16) NOT NULL,
                             author varchar(100) NOT NULL,
                             title varchar(100) NOT NULL,
                             price real NOT NULL,
                             PRIMARY KEY (isbn))";

if ($r = mysql_query($query))
    echo "<p>Table books created</p>";
else { // Query didn't run.
    print ('<p>Could create the table because: <b>' . mysql_error()
          . "</b>. The query was $query.</p>");
} // End of query IF.

// create table USERS
$query="CREATE TABLE IF NOT EXISTS users (username varchar(50) NOT NULL,
                                           password varchar(50) NOT NULL)";

if ($r = mysql_query($query))
    echo "<p>Table users created</p>";
else { // Query didn't run.
    print ('<p>Could create the table because: <b>' . mysql_error()
          . "</b>. The query was $query.</p>");
} // End of query IF.
```

# Παράδειγμα PHP διασύνδεση με MySQL (create\_tables.php)



```
$query="INSERT INTO users (username, password) VALUES
      (' user ', PASSWORD('test')),
      ('demo', PASSWORD('demo'))";

if ($r = mysql_query($query))
    echo "<p>Table users updated</p>";
else { // Query didn't run.
    print ('<p>Could insert data into users table because: <b>' . mysql_error()
          . "</b>. The query was $query.</p>");
} // End of query IF.

mysql_close();
echo "Database ready";

?>
```

# Παράδειγμα PHP διασύνδεση με MySQL (display\_data.php)



```
<?php
$username="guest";
$password="guest";
$dbase="test";

mysql_connect('localhost', $username, $password);
mysql_select_db($dbase) or die( "Unable to select database");
$query="SELECT * FROM contacts";
$result=mysql_query($query);

$num=mysql_numrows($result);

mysql_close();

echo "<b><center>Database Output</center></b><br><br>";
```

# Παράδειγμα PHP διασύνδεση με MySQL (display\_data.php)



```
$i=0;
while ($i < $num) {

    $first=mysql_result($result,$i,"first");
    $last=mysql_result($result,$i,"last");
    $phone=mysql_result($result,$i,"phone");
    $mobile=mysql_result($result,$i,"mobile");
    $fax=mysql_result($result,$i,"fax");
    $email=mysql_result($result,$i,"email");
    $web=mysql_result($result,$i,"web");

    echo "<b>$first $last</b><br>Phone: $phone<br>Mobile: $mobile<br>
        Fax: $fax<br>E-mail: $email<br>Web: $web<br><hr><br>";

    $i++;
}
```

?>

# Παράδειγμα PHP διασύνδεση με MySQL (insert\_book.php)



```
<html>
<head>
<title>Book-Net Entry Results</title>
</head>
<body>
<h1>Book-Net - Αποτελέσματα εισαγωγής τίτλου</h1>
<?php
if (!$_POST["isbn"] || !$_POST["author"] || !$_POST["title"] || !$_POST["price"]) {
echo "You have not entered all the required details.<br>"
."Please go back and try again.";
exit;
}
```

# Παράδειγμα PHP διασύνδεση με MySQL (insert\_book.php)



```
$isbn = addslashes($_POST["isbn"]);  
$author = addslashes($_POST["author"]);  
$title = addslashes($_POST["title"]);  
$price = floatval($_POST["price"]);  
$db = mysql_pconnect("localhost", "guest", "guest");  
if (!$db) {  
    echo "Error: Could not connect to database. Please try  
    again later."  
    exit;  
}
```

Χρήσιμη όταν βάζουμε  
δεδομένα της μορφής  
π.χ. O'Reilly

# Παράδειγμα PHP διασύνδεση με MySQL (insert\_book.php)



```
mysql_select_db("test");  
$query = "insert into books values ('".$isbn."',  
"'.$author."', '".$title."', '".$price."')";  
$result = mysql_query($query);  
if ($result)  
echo mysql_affected_rows() . " books inserted into database."  
?> </body> </html>
```



# Δημιουργία απλού blog



- Θα δούμε πώς μπορούμε να αναπτύξουμε μια απλή ΒΔ η οποία δημιουργεί ένα ημερολόγιο Ιστού (blog)
- `create_db.php` : Δημιουργεί μια ΒΔ με όνομα `myblog` και την επιλέγει ως ενεργή.
- `create_table.php` : Δημιουργεί έναν πίνακα στην ΒΔ `myblog` με όνομα `blog_entries`

blog_entries Table	
COLUMN NAME	COLUMN TYPE
blog_id	Positive, non-null, automatically incrementing integer
title	Text up to 100 characters in length
entry	Text of any length
date_entered	A timestamp including both the date and the time the row was added

# Δημιουργία απλού blog



- **add\_entry.php** : εισάγει δεδομένα στο blog, δηλαδή στην αντίστοιχη ΒΔ. Δημιουργεί μια φόρμα που χρησιμοποιεί ο χρήστης για την εισαγωγή της πληροφορίας.
- **view\_blog.php** : ανάκτηση δεδομένων από τη ΒΔ και παρουσίασή τους στον χρήστη
- **delete\_entry.php** : λαμβάνει τον κωδικό αριθμό της καταχώρησης (blog\_id) από τη σελίδα view\_blog.php, επιβεβαιώνει ότι χρήστης επιθυμεί τη διαγραφή της εγγραφής και κατόπιν προχωρεί στη διαγραφή της.
- **edit\_entry.php** : τροποποιεί μια καταχώρηση του ημερολογίου. Επιτρέπεται να τροποποιηθούν ο τίτλος και το κείμενο μιας καταχώρησης, αλλά όχι η ημερομηνία ή ο κωδικός της καταχώρησης.

# create\_db.php



```
// Attempt to connect to MySQL and print out messages.
if ($dbc = @mysql_connect ("localhost", "guest", "")) {

    print '<p>Successfully connected to MySQL.</p>';

    if (@mysql_query ('CREATE DATABASE myblog')) {
        print '<p>The database has been created.</p>';
    } else {
        die ('<p>Could not create the database because: <b>' . mysql_error() . '</b></p>');
    }

    if (@mysql_select_db ("myblog")) {
        print '<p>The database has been selected.</p>';
    } else {
        die ('<p>Could not select the database because: <b>' . mysql_error() . '</b></p>');
    }

    mysql_close(); // Close the connection.
} else {

    die ('<p>Could not connect to MySQL because: <b>' . mysql_error() . '</b></p>');
}
```

# create\_table.php



```
<?php // create_table.php
// This script connects to the MySQL server, selects the database, and creates a table.

// Address error handling.
ini_set ('display_errors', 1);
error_reporting (E_ALL & ~E_NOTICE);

// Connect and select.
if ($dbc = @mysql_connect ("localhost", "guest", "")) {

    if (!@mysql_select_db ("myblog")) {
        die ('<p>Could not select the database because: <b>' . mysql_error() . '</b></p>');
    }

} else {
    die ('<p>Could not connect to MySQL because: <b>' . mysql_error() . '</b></p>');
}
```

# create\_table.php



```
// Define the query.
$query = 'CREATE TABLE blog_entries (
    blog_id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(100) NOT NULL,
    entry TEXT NOT NULL,
    date_entered DATETIME NOT NULL
)';

// Run the query.
if (@mysql_query ($query)) {
    print '<p>The table has been created.</p>';
} else {
    die ('<p>Could not create the table because: <b>' . mysql_error() .
        '</b>.</p><p>The query being run was: ' . $query . '</p>');
}

mysql_close(); // Close the connection.

?>
```

# add\_entry.php



```
<form action="add_entry.php" method="post">
<p>Entry Title: <input type="text" name="title" size="40" maxsize="100" /></p>
<p>Entry Text: <textarea name="entry" cols="40" rows="5"></textarea></p>
<input type="submit" name="submit" value="Add to the Blog!" />
</form>
</body>
</html>
```

# add\_entry.php (συν.)



```
if (isset ($_POST['submit'])) { // Handle the form.

    // Connect and select.
    if ($dbc = @mysql_connect ('localhost', 'guest', '')) {

        if (!@mysql_select_db ('myblog')) {
            die ('<p>Could select the database because: <b>' . mysql_error() . '</b></p>');
        }

    } else {
        die ('<p>Could not connect to MySQL because: <b>' . mysql_error() . '</b></p>');
    }

    // Define the query.
    $query = "INSERT INTO blog_entries (title, entry, date_entered) VALUES ('".$_POST['title']."',
        '".$_POST['entry']."', NOW())";

    // Execute the query.
    if (@mysql_query ($query)) {
        print '<p>The blog entry has been added.</p>';
    } else {
        print "<p>Could add the entry because: <b>" . mysql_error() . "</b>. The query was $query.</p>";
    }

    mysql_close();
}
```

# view\_blog.php



```
// Connect and select.
if ($dbc = @mysql_connect ('localhost', 'guest', '')) {

    if (!@mysql_select_db ('myblog')) {
        die ('<p>Could select the database because: <b>' . mysql_error() . '</b></p>');
    }

} else {
    die ('<p>Could not connect to MySQL because: <b>' . mysql_error() . '</b></p>');
}
```



# view\_blog.php



```
// Define the query.
$query = 'SELECT * FROM blog_entries ORDER BY date_entered DESC';

if ($r = mysql_query ($query)) { // Run the query.

    // Retrieve and print every record.
    while ($row = mysql_fetch_array ($r)) {
        print "<p><h3>{$row['title']}</h3>
        {$row['entry']}<br />
        <a href=\"edit_entry.php?id={$row['blog_id']}\">Edit</a>
        <a href=\"delete_entry.php?id={$row['blog_id']}\">Delete</a>
        </p><hr />\n";
    }
} else { // Query didn't run.
    die ('<p>Could create the table because: <b>' . mysql_error() . "</b>. The query was $query.</p>");
} // End of query IF.
```

# delete\_entry.php



```
if (isset ($_POST['submit'])) { // Handle the form.

    // Define the query.
    $query = "DELETE FROM blog_entries WHERE blog_id={$_POST['id']} LIMIT 1";
    $r = mysql_query ($query); // Execute the query.

    // Report on the result.
    if (mysql_affected_rows() == 1) {
        print '<p>The blog entry has been deleted.</p>';
    } else {
        print "<p>Could delete the entry because: <b>" . mysql_error() . "</b>. The query was $query.</p>";
    }

} else { // Display the entry in a form.
```

# delete\_entry.php (cont.)

```
} else { // Display the entry in a form.

    // Check for a valid entry ID in the URL.
    if (is_numeric ($_GET['id'])) {

        // Define the query.
        $query = "SELECT * FROM blog_entries WHERE blog_id={$_GET['id']}";
        if ($r = mysql_query ($query)) { // Run the query.

            $row = mysql_fetch_array ($r); // Retrieve the information.

            // Make the form.
            print '<form action="delete_entry.php" method="post">
<p>Are you sure you want to delete this entry?</p>
<p><h3>' . $row['title'] . '</h3>' .
            $row['entry'] . '<br />
<input type="hidden" name="id" value="' . $_GET['id'] . '" />
<input type="submit" name="submit" value="Delete this Entry!" /></p>
</form>';

        } else { // Couldn't get the information.
            print "<p>Could retrieve the entry because: <b>" . mysql_error() . "</b>.
            The query was $query.</p>";
        }

    } else { // No ID set.
        print '<p><b>You must have made a mistake in using this page.</b></p>';
    }
} // End of main IF.
```

# edit\_entry.php



```
<body>
<?php // edit_entry.php
// This script edits a blog entry using an UPDATE query.

// Address error handling.
ini_set ('display_errors', 1);
error_reporting (E_ALL & ~E_NOTICE);

// Connect and select.
if ($dbc = @mysql_connect ('localhost', 'guest', '')) {

    if (!@mysql_select_db ('myblog')) {
        die ('<p>Could select the database because: <b>' . mysql_error() . '</b></p>');
    }

} else {
    die ('<p>Could not connect to MySQL because: <b>' . mysql_error() . '</b></p>');
}
```

# edit\_entry.php (συν.)



```
if (isset ($_POST['submit'])) { // Handle the form.

    // Define the query.
    $query = "UPDATE blog_entries SET title='{$_POST['title']}', entry='{$_POST['entry']}'
    |         | WHERE blog_id={$_POST['id']}";
    $r = mysql_query ($query); // Execute the query.

    // Report on the result.
    if (mysql_affected_rows() == 1) {
        print '<p>The blog entry has been updated.</p>';
    } else {
        print "<p>Could update the entry because: <b>" . mysql_error() . "</b>."
        |         | The query was $query.</p>";
    }
} else { // Display the entry in a form.
```

# edit\_entry.php (συν.)



```
} else { // Display the entry in a form.

// Check for a valid entry ID in the URL.
if (is_numeric ($_GET['id'])) {

// Define the query.
$query = "SELECT * FROM blog_entries WHERE blog_id={$_GET['id']}";
if ($r = mysql_query ($query)) { // Run the query.

    $row = mysql_fetch_array ($r); // Retrieve the information.

// Make the form.
    print '<form action="edit_entry.php" method="post">
<p>Entry Title: <input type="text" name="title" size="40" maxsize="100" value="' . $row['title'] . '" /></p>
<p>Entry Text: <textarea name="entry" columns="40" rows="5">' . $row['entry'] . '</textarea></p>
<input type="hidden" name="id" value="' . $_GET['id'] . '" />
<input type="submit" name="submit" value="Update this Entry!" />
</form>';

} else { // Couldn't get the information.
    print "<p>Could retrieve the entry because: <b>" . mysql_error() . "</b>. The query was $query.</p>";
}

} else { // No ID set.
    print '<p><b>You must have made a mistake in using this page.</b></p>';
}

} // End of main IF.
```



- Σε μεγαλύτερες εφαρμογές Ιστού συνιστάται η τοποθέτηση του κώδικα της σύνδεσης με τη ΒΔ και της επιλογής της ΒΔ σε ένα ξεχωριστό αρχείο, το οποίο θα βρίσκεται έξω από τον κατάλογο Ιστού. Κάθε σενάριο PHP που ζητάει τη ΒΔ μπορεί να συμπεριλαμβάνει (include) αυτό το εξωτερικό αρχείο.
- Αν τα μαγικά εισαγωγικά (`magic_quotes_gpc`) είναι ενεργοποιημένα στην εγκατάσταση της PHP τότε θα πρέπει να χρησιμοποιούμε τη συνάρτηση `stripslashes()` για να απομακρύνουμε τις επιπλέον καθέτους από τις τιμές που υποβλήθηκαν σε μια φόρμα.
- Είναι καλή ιδέα να εφαρμόζουμε τη συνάρτηση `trim()` για να αφαιρούμε τα περιττά κενά διαστήματα από τις τιμές που εισάγονται σε μια ΒΔ.

# ΔΙΑΣΥΝΔΕΣΗ PHP ΜΕ ΒΔ ODBC



# Διασύνδεση PHP με ΒΔ (ODBC)



Βήμα 1 – Σύνδεση με ODBC βάση

```
$connect = odbc_connect ($database,$username,$password);
```

Βήμα 2 – Εκτέλεση SQL εντολών

```
$result = odbc_exec($connect, $query);
```

Βήμα 3 – Ανάγνωση αποτελεσμάτων

```
while (odbc_fetch_row($result)) {  
    $name = odbc_result($result, 1);  
    $surname = odbc_result($result, 2);  
}
```

Βήμα 4 – Τερματισμός σύνδεσης

```
odbc_close($connect);
```

# Βασικές συναρτήσεις php-ODBC



resource **odbc\_connect** ( string \$dsn, string \$user, string \$password [, int \$cursor\_type] )

- Επιστρέφει τον αριθμό της ODBC σύνδεσης ή 0 (**FALSE**) σε περίπτωση λάθους.

resource **odbc\_exec** ( resource \$connection\_id, string \$query\_string [, int \$flags] )

- Υποβάλει ένα ερώτημα στη ΒΔ. Επιστρέφει ένα resource σε περίπτωση επιτυχίας ή **FALSE** σε περίπτωση αποτυχίας.

string **odbc\_result** ( resource \$result\_id, mixed \$field )

- Επιστρέφει το περιεχόμενο του πεδίου field (είτε με την χρήση αριθμού είτε με την χρήση του ονόματος του πεδίου).

void **odbc\_close** ( resource \$connection\_id )

- Κλείνει η σύνδεση με τη ΒΔ.

# Επιπλέον συναρτήσεις php-ODBC



**bool odbc\_fetch\_row** ( resource \$result\_id [, int \$row\_number] )

- Επιλέγει την επόμενη σειρά από το αποτέλεσμα και επιστρέφει **TRUE** εάν υπάρχει σειρά και **FALSE** εάν δεν υπάρχει άλλη σειρά.

**string odbc\_error** ( [resource \$connection\_id] )

- Επιστρέφει τον κωδικό του τελευταίου λάθους.

**int odbc\_num\_fields** ( resource \$result\_id )

- Επιστρέφει τον αριθμό των πεδίων.

**int odbc\_num\_rows** ( resource \$result\_id )

- Επιστρέφει τον αριθμό των γραμμών του αποτελέσματος και -1 σε περίπτωση λάθους.

**bool odbc\_free\_result** (resource \$result)

- Ελευθερώνει τους πόρους του result. Επιστρέφει **TRUE** σε επιτυχία ή **FALSE** σε αποτυχία

Πλήρης λίστα συναρτήσεων στο

<http://www.php.net/manual/en/ref.uodbc.php>

# Παράδειγμα: ODBC Διασύνδεση με ΒΔ



<?

```
$oConn=odbc_connect ("FPNWIND", "", "");
```

```
$oRs=odbc_exec($oConn, "SELECT ContactName, ContactTitle  
From Customers");
```

?>

# Παράδειγμα: ODBC Διασύνδεση με ΒΔ



```
<TABLE border = 1>
  <tr><th>Name</th><th>Title</th></tr>
  <? while (odbc_fetch_into($oRs,$row)) { ?>
    <tr>
  <? for ($Index=0; $Index<=(count($row)-1); $Index=$Index+1) { ?>
    <TD VAlign=top><?=$row[$Index]; ?> </TD>
  <? } ?>
    </tr>
  <? }
    odbc_free_result($oRs); ?>
</TABLE>
```

# **ΥΛΟΠΟΙΗΣΗ ΕΛΕΓΧΟΥ ΠΡΟΣΒΑΣΗΣ**

# Υλοποίηση ελέγχου πρόσβασης



## Please Log In

<b>Username</b>	<input type="text" value="user"/>
<b>Password</b>	<input type="password" value="••••"/>
<input type="button" value="Log In"/>	

**Here it is!**

I bet you are glad you can see this secret page.

**Go Away!**

You are not authorized to view this resource.

# auth.php



```
<?php
if (!isset($_POST["username"]) && !isset($_POST["password"])) {
//Visitor needs to enter a name and password
?>
<h1>Please Log In</h1>
<form method = post action = "auth.php ">
<!-- Εμφάνισε τη φόρμα Log In -->
</form>
<?php
}
else if($_POST["username"]=="user" && $_POST["password"]=="pass") {
// visitor's name and password combination are correct
    echo '<h1>Here it is!</h1>';
    echo 'I bet you are glad you can see this secret page.';
}
else {
// visitor's name and password combination are not correct
    echo '<h1>Go Away!</h1>';
    echo 'You are not authorized to view this resource.';
}
?>
```



# Προβλήματα απλού μηχανισμού πρόσβασης



Το προηγούμενο script παρουσιάζει τα παρακάτω προβλήματα :

1. Έχει ένα μοναδικό username/password που είναι γραμμένα μέσα στο script
  2. Αποθηκεύει το password μέσα στο script σαν απλό κείμενο (δεν το βλέπει βέβαια ο χρήστης, αλλά το βλέπει ο διαχειριστής του web server που έχει πρόσβαση στο PHP script)
  3. Προστατεύει μόνο μια σελίδα
  4. Μεταδίδει (στο Internet) τον κωδικό πρόσβασης ως απλό κείμενο
- Λύση στα δύο πρώτα προβλήματα:
    - Έλεγχος ταυτότητας με συνδυασμό PHP-MySQL (θα αποθηκεύσω πολλά username/password σε ΒΔ, το password κρυπτογραφημένο)
  - Λύση στο τρίτο πρόβλημα:
    - Χρήση συνόδων (sessions)
  - Λύση στα τέταρτο πρόβλημα:
    - Μετάδοση του password σε κρυπτογραφημένη μορφή (με SSL)

# Δημιουργία ΒΔ χρηστών



```
CREATE DATABASE test;
```

```
USE test;
```

```
CREATE TABLE IF NOT EXISTS users (  
    username varchar(50) NOT NULL,  
    password varchar(50) NOT NULL  
);
```

```
INSERT INTO users (username, password) VALUES  
('user', 'test');
```

Για κρυπτογραφημένο password χρησιμοποιούμε την SQL  
συνάρτηση **PASSWORD()**

```
INSERT INTO users (username, password) VALUES  
('user', PASSWORD('test'));
```

# login\_process\_db



```
<?php
    @ $db = mysql_pconnect("localhost", "guest", "guest");
    if (!$db) {
        echo "Error: Could not connect to database. Please try again
later.";
        exit;
    }
    mysql_select_db("test");
    $query = "select * from users where username='" . $_POST
['username'] . "' and password = PASSWORD('" . $_POST['pass'] .
"')" ;
    // echo $query;
    $result = mysql_query($query);
    if (!$result) {
        echo 'Cannot run query!...';
        exit;
    }
    $num_results = mysql_num_rows($result);
    if ($num_results > 0) {
        echo "<h2>Congratulations! You 've logged in successfully!...
</h2>";
    } else {
        echo "<H2>PLEASE TRY AGAIN</H2>";
        echo "<a href=loginform_db.htm>Back</a>";
    }
?>
```



- Χρήση συναρτήσεων `addslashes` / `stripslashes` για το input του χρήστη
  - Αποτρέπουν προβλήματα στην SQL εντολών
  - Αυξάνουν την ασφάλεια
- Κρυπτογράφηση κωδικών πρόσβασης:
  - Χρήση της συνάρτησης `password` στο επίπεδο SQL
  - Χρήση της συνάρτησης `crypt` στο επίπεδο της PHP

# SQL injection



```
<?php
$query = "SELECT * FROM members WHERE
username = \"$_POST['username']\" AND
password = \"$_POST['password']\"";
?>
```



'admin--'



```
SELECT * FROM members WHERE username = admin -- AND password = hacker
```

```
<?php
$query = "UPDATE usertable SET pwd='$pwd'
WHERE uid='$uid'";
?>
```



' OR uid LIKE '%admin%'



```
"UPDATE usertable SET pwd='...' WHERE uid=" OR uid LIKE '%admin%'";
```

# SQL injection



```
<?php
$query = „SELECT * FROM products WHERE id
= $_GET['id']“;
?>
```



```
'10; DROP
products --'
```



```
SELECT * FROM products WHERE id = 10; DROP products --
```

```
<?php
$query = "SELECT id, name, inserted, size
FROM products
WHERE size = '$size'";
$result = odbc_exec($conn, $query);
?>
```



```
'
union select '1',
concat(uname||'-'||passwd) as
name, '1971-01-01', '0' from
usertable;
--
```



```
SELECT id, name, inserted, size FROM products
WHERE size=" union select '1', concat(uname || '-' || passwd) as name, '1971-01-01',
'0' from usertable;
--
```

# Τρόποι Χειρισμού



Ποτέ δεν συνδεόμαστε σαν superuser ή database owner.

- Πάντα customized users με limited προνόμια

Χρήση statements με bound variables

- MySQLi
- PDO

Χρήση συναρτήσεων που κάνουν validate την είσοδο

- Π.χ. `is_numeric()`,

Χρήση συναρτήσεων escape

- [mysql\\_real\\_escape\\_string\(\)](#),  
[sqlite\\_escape\\_string\(\)](#),

Δεν αποκαλύπτουμε το σχήμα της βάσης ή πληροφορίες πινάκων

- Χρήση stored procedures, views, κτλ.

# MySQLi (i=improved) - Open Connection



00

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username,
$password);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn-
>connect_error);
}
echo "Connected successfully";
?>
```

Procedural

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = mysqli_connect($servername,
$username, $password);

// Check connection
if (!$conn) {
    die("Connection failed: " .
mysqli_connect_error());
}
echo "Connected successfully";
?>
```



# MySQLi - Close Connection



OO

```
$conn->close();
```

Procedural

```
mysqli_close($conn);
```

# MySQLi – Create Database



```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$conn = new mysqli($servername, $username,
$password);
if ($conn->connect_error) {
    die("Connection failed: " . $conn-
>connect_error);
}
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . $conn->error;
}
$conn->close();
?>
```

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$conn = mysqli_connect($servername,
$username, $password);
if (!$conn) {
    die("Connection failed: " .
mysqli_connect_error());
}
$sql = "CREATE DATABASE myDB";
if (mysqli_query($conn, $sql)) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " .
mysqli_error($conn);
}
mysqli_close($conn);
?>
```

# MySQLi – Create Table



```
$sql = "CREATE TABLE MyGuests (  
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY  
KEY,  
firstname VARCHAR(30) NOT NULL,  
lastname VARCHAR(30) NOT NULL,  
email VARCHAR(50),  
reg_date TIMESTAMP  
)";
```

```
if ($conn->query($sql) === TRUE) {  
    echo "Table MyGuests created successfully";  
} else {  
    echo "Error creating table: " . $conn->error;  
}
```

```
$sql = "CREATE TABLE MyGuests (  
id INT(6) UNSIGNED AUTO_INCREMENT  
PRIMARY KEY,  
firstname VARCHAR(30) NOT NULL,  
lastname VARCHAR(30) NOT NULL,  
email VARCHAR(50),  
reg_date TIMESTAMP  
)";
```

```
if (mysqli_query($conn, $sql)) {  
    echo "Table MyGuests created  
successfully";  
} else {  
    echo "Error creating table: " .  
mysqli_error($conn);  
}
```

# MySQLi – Insert



```
$sql = "INSERT INTO MyGuests (firstname,  
lastname, email)  
VALUES ('John', 'Doe', 'john@example.com')";
```

```
if ($conn->query($sql) === TRUE) {  
    echo "New record created successfully";  
} else {  
    echo "Error: " . $sql . "<br>" . $conn->error;  
}
```

```
$sql = "INSERT INTO MyGuests  
(firstname, lastname, email)  
VALUES ('John', 'Doe',  
'john@example.com')";
```

```
if (mysqli_query($conn, $sql)) {  
    echo "New record created  
successfully";  
} else {  
    echo "Error: " . $sql . "<br>" .  
mysqli_error($conn);  
}
```

# MySQLi – Insert multiple



```
$sql = "INSERT INTO MyGuests (firstname,
lastname, email)
VALUES ('John', 'Doe', 'john@example.com');";
$sql .= "INSERT INTO MyGuests (firstname,
lastname, email)
VALUES ('Mary', 'Moe', 'mary@example.com');";
$sql .= "INSERT INTO MyGuests (firstname,
lastname, email)
VALUES ('Julie', 'Dooley', 'julie@example.com');";

if ($conn->multi_query($sql) === TRUE) {
    echo "New records created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
```

```
$sql = "INSERT INTO MyGuests
(firstname, lastname, email)
VALUES ('John', 'Doe',
'john@example.com');";
$sql .= "INSERT INTO MyGuests
(firstname, lastname, email)
VALUES ('Mary', 'Moe',
'mary@example.com');";
$sql .= "INSERT INTO MyGuests
(firstname, lastname, email)
VALUES ('Julie', 'Dooley',
'julie@example.com');";

if (mysqli_multi_query($conn, $sql)) {
    echo "New records created
successfully";
} else {
    echo "Error: " . $sql . "<br>" .
mysqli_error($conn);
}
```

# MySQLi – Insert με prepared statements



```
$stmt = $conn->prepare("INSERT INTO MyGuests (firstname,  
lastname, email) VALUES (?, ?, ?)");  
$stmt->bind_param("sss", $firstname, $lastname, $email);
```

```
$firstname = "John";  
$lastname = "Doe";  
$email = "john@example.com";  
$stmt->execute();
```

```
$firstname = "Mary";  
$lastname = "Moe";  
$email = "mary@example.com";  
$stmt->execute();
```

```
$firstname = "Julie";  
$lastname = "Dooley";  
$email = "julie@example.com";  
$stmt->execute();
```

```
echo "New records created successfully";  
$stmt->close();
```

deprecated

# MySQLi – Select



```
$sql = "SELECT id, firstname, lastname FROM
MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " .
$row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
```

```
$sql = "SELECT id, firstname, lastname
FROM MyGuests";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row =
mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " .
$row["firstname"]. " " .
$row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
```

# MySQLi – Delete



id	firstname	lastname	email
1	John	Doe	john@example.com
2	Mary	Moe	mary@example.com
3	Julie	Dooley	julie@example.com

```
$sql = "DELETE FROM MyGuests WHERE id=3";  
  
if ($conn->query($sql) === TRUE) {  
    echo "Record deleted successfully";  
} else {  
    echo "Error deleting record: " . $conn->error;  
}
```

```
$sql = "DELETE FROM MyGuests WHERE  
id=3";  
  
if (mysqli_query($conn, $sql)) {  
    echo "Record deleted successfully";  
} else {  
    echo "Error deleting record: " .  
    mysqli_error($conn);  
}
```



# MySQLi – Update



id	firstname	lastname	email
1	John	Doe	john@example.com
2	Mary	Moe	mary@example.com

```
$sql = "UPDATE MyGuests SET lastname='Doe'
WHERE id=2";

if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . $conn->error;
}
```

```
$sql = "UPDATE MyGuests SET
lastname='Doe' WHERE id=2";

if (mysqli_query($conn, $sql)) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " .
mysqli_error($conn);
}
```

# PDO – Open, Close Connection



```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
    $conn = new PDO("mysql:host=$servername;dbname=myDB",
$username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
    echo "Connected successfully";
}
catch(PDOException $e)
{
    echo "Connection failed: " . $e->getMessage();
}
?>
```

```
$conn = null;
```

# PDO – Create Database



```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
try {
    $conn = new PDO("mysql:host=$servername;dbname=myDB", $username,
$password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "CREATE DATABASE myDBPDO";
    // use exec() because no results are returned
    $conn->exec($sql);
    echo "Database created successfully<br>";
}
catch(PDOException $e)
{
    echo $sql . "<br>" . $e->getMessage();
}
$conn = null;
?>
```

# PDO – Create Tables



```
$sql = "CREATE TABLE MyGuests (  
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
firstname VARCHAR(30) NOT NULL,  
lastname VARCHAR(30) NOT NULL,  
email VARCHAR(50),  
reg_date TIMESTAMP  
)";  
// use exec() because no results are returned  
$conn->exec($sql);  
echo "Table MyGuests created successfully";
```

# PDO – Insert



```
$sql = "INSERT INTO MyGuests (firstname, lastname, email)  
VALUES ('John', 'Doe', 'john@example.com')";
```

```
$conn->exec($sql);  
echo "New record created successfully";
```

# PDO – Insert Multiple



```
// begin the transaction
$conn->beginTransaction();

$conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')");
$conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Mary', 'Moe', 'mary@example.com')");
$conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Julie', 'Dooley', 'julie@example.com')");

// commit the transaction
$conn->commit();
echo "New records created successfully";
}
catch(PDOException $e)
{
// roll back the transaction if something failed
$conn->rollback();
echo "Error: " . $e->getMessage();
```

# PDO – Prepare Statements



```
$stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email)
VALUES (:firstname, :lastname, :email)");
$stmt->bindParam(':firstname', $firstname);
$stmt->bindParam(':lastname', $lastname);
$stmt->bindParam(':email', $email);

$firstname = "John";
$lastname = "Doe";
$email = "john@example.com";
$stmt->execute();

$firstname = "Mary";
$lastname = "Moe";
$email = "mary@example.com";
$stmt->execute();

$firstname = "Julie";
$lastname = "Dooley";
$email = "julie@example.com";
$stmt->execute();
echo "New records created successfully";
}
```

# PDO – Select, Delete, Update



```
$stmt = $conn->prepare("SELECT id, firstname, lastname FROM MyGuests");  
$stmt->execute();  
$result = $stmt->setFetchMode(PDO::FETCH_ASSOC);  
foreach(new TableRows(new RecursiveArrayIterator($stmt->fetchAll())) as $k=>$v) {  
    echo $v;
```

```
// sql to delete a record  
$sql = "DELETE FROM MyGuests WHERE id=3";  
$conn->exec($sql);  
echo "Record deleted successfully";
```

```
$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";  
$stmt = $conn->prepare($sql);  
$stmt->execute();  
echo $stmt->rowCount() . " records UPDATED successfully";
```



# **ΧΕΙΡΙΣΜΟΣ ΗΜΕΡΟΜΗΝΙΩΝ**

# Λήψη ημερομηνίας & ώρας από την PHP



- Συνάρτηση `date()` επιστρέφει την ημερομηνία ως μορφοποιημένο string

```
echo date("jS F Y"); → "18th May 2003"
```

```
echo date("d/m/y, h:s A"); → "18/05/03, 12:41 PM"
```

# Κωδικοί μορφοποίησης



Κωδικός	Περιγραφή
A	Πρωί ή απόγευμα, AM ή PM
d	Ημέρα του μήνα, 01 έως 31
D	Συντομευμένη μορφή ημέρας, Mon έως Sun
F	Μήνας του έτους με πλήρη μορφή κειμένου, January ως December
g	Ώρα της ημέρας σε 12ωρη μορφή, χωρίς αρχικά μηδενικά, 1 ως 12
G	Ώρα της ημέρας σε 24ωρη μορφή, χωρίς αρχικά μηδενικά, 0 ως 23
h	Ώρα της ημέρας σε 12ωρη μορφή, με αρχικά μηδενικά, 01 ως 12
H	Ώρα της ημέρας σε 24ωρη μορφή, χωρίς αρχικά μηδενικά, 00 ως 23

# Κωδικοί μορφοποίησης



Κωδικός	Περιγραφή
i	Λεπτά της ώρας με αρχικά μηδενικά, 00 ως 59
j	Ημέρα του μήνα, χωρίς αρχικά μηδενικά, 1 έως 31
l	Ημέρα της εβδομάδας με πλήρη μορφή κειμένου, Monday ως Sunday
m	Μήνας του έτους με αρχικά μηδενικά, 01 ως 12
M	Συντομευμένη μορφή μήνα, Jan ως Dec
O	Διαφορά από την ώρα Greenwich (GMT) σε ώρες, π.χ. +02:00
s	Δευτερόλεπτα με αρχικά μηδενικά, 00 ως 59
t	Αριθμός ημερών του μήνα, 28 ως 31
U	Δευτερόλεπτα από την Unix Epoch (January 1 1970 00:00:00 GMT)
Y	Πλήρης αριθμητική αναπαράσταση του έτους, π.χ. 1999 ή 2003
y	Αριθμητική αναπαράσταση του έτους με δύο ψηφία, π.χ. 99 ή 03

## Χρησιμοποιώντας τις σφραγίδες χρόνου του UNIX



- Η PHP δίνει την δυνατότητα μετατροπής ημερομηνίας/ώρας σε σφραγίδα χρόνου
- Συχνά, τα υπολογιστικά συστήματα μετρούν το χρόνο με βάση τις «Σφραγίδες Χρόνου του UNIX», δηλαδή τον αριθμό δευτερολέπτων από την 1/1/1970, 00:00:00 GMT
- Μετατροπή ημερομηνίας/ώρας σε σφραγίδα χρόνου:  
`int mktime (int hour, int minute, int second, int month, int day, int year);`
- Σφραγίδα χρόνου τρέχουσας ημερομηνίας:  
`$timestamp = mktime();` ή `$timestamp = date("U");`

# Προσδιορισμός ημερομηνίας



- Έπειτα χρησιμοποιούμε την `getdate()` για μορφοποίηση.  
`array getdate (int timestamp)`

Κλειδί	Τιμή
seconds	Δευτερόλεπτα, 0 ως 59
minutes	Λεπτά, 0 ως 59
hours	Ώρες, 0 ως 23
mday	Ημέρα του μήνα, 1 ως 31
wday	Ημέρα της εβδομάδας, 0 (Sunday) ως 6 (Saturday)
mon	Μήνας, 1 ως 12
year	Έτος (με 4 ψηφία), π.χ. 1999 ή 2003
weekday	Ημέρα εβδομάδας, πλήρης μορφή κειμένου, Sunday ως Saturday
month	Μήνας, πλήρης μορφή κειμένου, January ως December

```
$today = getdate();  
$month = $today['month'];  
$mday = $today['mday'];  
$year = $today['year'];  
echo "$month $mday, $year";
```

# Μετατροπή ημερομηνιών μεταξύ PHP-MySQL



- Οι ημερομηνίες/ώρες στην MySQL ανακαλούνται διαφορετικά (η MySQL περιμένει πρώτα το έτος)
  - π.χ. η 29η Αυγούστου 2003 δίνεται ως 2003-08-29 ή 03-08-29
- Για να βάλουμε ημερομηνίες στην MySQL από PHP:
  - Χρησιμοποιείται η `date()`, π.χ. `date("Y-m-d")`
- Για να εμφανίσουμε μέσω PHP ημερομηνίες που ανακλήθηκαν από τη MySQL:
  - Χρησιμοποιούνται οι συναρτήσεις της MySQL:
  - `SELECT DATE_FORMAT ( date_column, '%m %d %Y ') FROM tablename;`
  - `SELECT UNIX_TIMESTAMP ( date_column) FROM tablename;`

# Υπολογισμοί με ημερομηνίες



<?

```
// Η ημερομηνία γενεθλίων
```

```
$day = 4;
```

```
$month = 9;
```

```
$year = 1975;
```

```
// Η σφραγίδα χρόνου για την ημ/νία γενεθλίων
```

```
$bdayunix = mktime (0, 0, 0, $month, $day, $year);
```

```
$nowunix = mktime(); // Η σημερινή σφραγίδα χρόνου
```

```
$ageunix = $nowunix - $bdayunix; // Πόσος χρόνος πέρασε;
```

```
// Μετατροπή από δευτερόλεπτα σε χρόνια
```

```
$age = floor($ageunix / (365 * 24 * 60 * 60));
```

```
echo "Age is $age"; // Θα εκτυπώσει την ηλικία σε έτη
```

?>