

# Curs TWSS

## Introducere în MySQL

- Sistem de management al bazelor de date pentru servere web
- SQL = Structured Query Language
- **Bază de date:** colecție de informații stocate și organizate

```
SELECT title FROM publications WHERE author = 'Charles Dickens';
```

- Orice bază de date poate conține unul sau mai multe tabele
- **Tabel:** colecție de înregistrări (linii)
- **Linie:** colecție de coloane (câmpuri) care conțin efectiv informația
- **Coloană:** numele unui câmp dintr-o linie

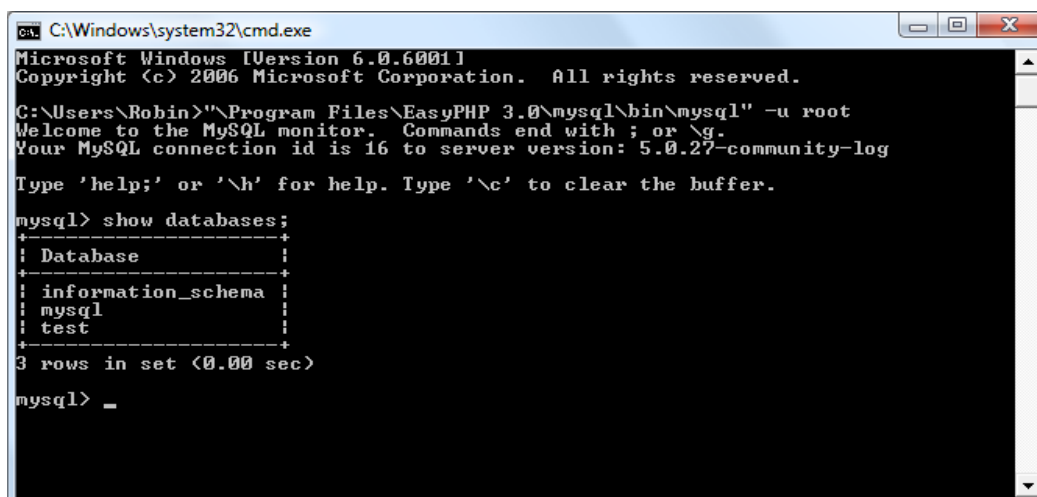
Author	Title	Type	Year
Mark Twain	The Adventures of Tom Sawyer	Fiction	1876
Jane Austen	Pride and Prejudice	Fiction	1811
Charles Darwin	The Origin of Species	Non-fiction	1856
Charles Dickens	The Old Curiosity Shop	Fiction	1841
William Shakespeare	Romeo and Juliet	Play	1594

## Interacțiunea cu MySQL:

- Linia de comandă
- Interfață web (phpMyAdmin)
- Limbaj de programare (PHP)

## Linia de comandă

- `mysql -u username -p`
- `SHOW databases`



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\Robin>"\Program Files\EasyPHP 3.0\mysql\bin\mysql" -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16 to server version: 5.0.27-community-log
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| test      |
+-----+
3 rows in set (0.00 sec)

mysql> _
```

### Reguli:

- ; la finalul fiecărei comenzi
- \c + Enter – anularea unei comenzi

MySQL prompt	Descriere
mysql>	Stare de așteptare comandă
->	Stare de așteptare pentru următoarea linie din comandă
'>	Stare de așteptare pentru a continua completarea șirului de caractere care a început cu '
">	Stare de așteptare pentru a continua completarea șirului de caractere care a început cu "
/*>	Stare de așteptare pentru a continua completarea șirului de caractere care a început cu /*

### Comenzi MySQL

Comandă	Parametru	Descriere
ALTER	DATABASE, TABLE	
BACKUP	TABLE	
\c		
CREATE	DATABASE, TABLE	
DELETE	Expresie cu TABLE & ROW	
DESCRIBE	TABLE	Describe coloanele tabelului
DROP	DATABASE, TABLE	Șterge baza de date sau tabelul
EXIT (CTRL + C)		
GRANT	User detalii	Schimbă privilegiile utilizatorului
HELP (\h, \?)	Item	
ALTER	DATABASE, TABLE	
BACKUP	TABLE	
\c		
CREATE	DATABASE, TABLE	
DELETE	Expresie cu TABLE & ROW	
DESCRIBE	TABLE	Describe coloanele tabelului
DROP	DATABASE, TABLE	Șterge baza de date sau tabelul
EXIT (CTRL + C)		
GRANT	User detalii	Schimbă privilegiile utilizatorului
HELP (\h, \?)	Item	
INSERT	Expresie	
LOCK	TABLE	
QUIT		=EXIT
RENAME	TABLE	
SHOW	Item	
SOURCE	Filename	Execută comenzile din fișier
STATUS (\s)		Afișează statusul curent
TRUNCATE	TABLE	Golește tabelul
UNLOCK	TABLE	
UPDATE	Expresie	
USE	DATABASE	

<b>Creează baza de date</b>
CREATE DATABASE publications;
USE publications;
<b>Creează utilizatori</b>
GRANT PRIVILEGES ON database.object TO 'username@hostname' IDENTIFIED BY 'password';
<i>database.object = (*.*, database.*, database.object)</i>
GRANT ALL ON publications.* TO 'jim' IDENTIFIED BY 'mypasswd';
<b>Creează tabel</b>
CREATE TABLE classics ( author VARCHAR(128), title VARCHAR(128), type VARCHAR(16), year CHAR(4));
DESCRIBE classics;

## Tipuri de date

Tip	Număr de octeți	Exemplu
CHAR (n)	Exact n (<=255)	Char (5) Hello – 5 octeți Char (57) New York – 57 octeți
VARCHAR (n)	Până la n (<=65535)	VARCHAR (100) Greetings – 9 octeți VARCHAR (7) Morning – 7 octeți
BINARY (n) BYTE (n)	Exact n (<=2YY)	
VARBINARY (n)	Până la n (<=6YY3Y)	
TINYTEXT (n)	Până la n (<=2YY)	Șiruri de caractere
TEXT (n)	Până la n (<=6YY3Y)	
MEDIUMTEXT (n)	Până la n (<=1677721Y)	
LONGTEXT (n)	Până la n (<=429496729Y)	
TINYBLOB (n)	Până la n (<=2YY)	Date binare
BLOB (n)	Până la n (<=6YY3Y)	
MEDIUMBLOB (n)	Până la n (<=1677721Y)	
LOBLOB (n)	Până la n (<=429496729Y)	
TINYINT	1 octet [-128, 127] [0, 2YY]	
SMALLINT	2 octeți	
MEDIUMINT	3 octeți	
INT, INTEGER	4 octeți	
BIGINT	8 octeți	
FLOAT	4 octeți	
DOUBLE, REAL	8 octeți	
DATETIME		'0000-00-00 00:00:00'

DATE	'0000-00-00'
TIMESTAMP	'0000-00-00 00:00:00' = DATETIME (doar pentru anii 1970-2037)
TIME	'00:00:00'
YEAR	0000

Caracteristici ale coloanelor
INT UNSIGNED
NOT NULL
AUTO_INCREMENT
KEY

## Operațiuni cu tabele

```
CREATE TABLE classics (
  author VARCHAR(128),
  title VARCHAR(128),
  type VARCHAR(16),
  year CHAR(4),
  id INT UNSIGNED NOT NULL AUTO_INCREMENT KEY)
```

### Adăugarea informațiilor în tabele

```
INSERT INTO classics(author, title, type, year)
VALUES('Mark Twain','The Adventures of Tom Sawyer','Fiction','1876');
INSERT INTO classics(author, title, type, year)
VALUES('Jane Austen','Pride and Prejudice','Fiction','1811');
INSERT INTO classics(author, title, type, year)
VALUES('Charles Darwin','The Origin of Species','Non-Fiction','1856');
INSERT INTO classics(author, title, type, year)
VALUES('Charles Dickens','The Old Curiosity Shop','Fiction','1841');
INSERT INTO classics(author, title, type, year)
VALUES('William Shakespeare','Romeo and Juliet','Play','1594');
```

### Redenumirea unui tabel

```
ALTER TABLE classics RENAME pre1900;
ALTER TABLE pre1900 RENAME classics;
```

### Schimbarea tipului de date

```
ALTER TABLE classics MODIFY year SMALLINT;
```

### Adăugarea unei noi coloane

```
ALTER TABLE classics ADD pages SMALLINT UNSIGNED;
```

### Redenumirea unei coloane

```
ALTER TABLE classics CHANGE type category VARCHAR(16);
```

### Ștergerea unei coloane

```
ALTER TABLE classics DROP pages;
```

### Ștergerea unui tabel

```
DROP TABLE classics;
```

### Indexarea tabelelor

Momentan tabelul *classics* funcționează, și poate fi interogată fără probleme prin MySQL. Dar în cazul în care începe să crească, și ajungem la sute sau mii de înregistrări, accesul la el o să fie tot mai greoaie și lentă, fiindcă MySQL-ul trebuie să caute în fiecare înregistrare la o comandă de interogare. Asta ar arăta parcă în bibliotecă ar trebui să luăm în mână fiecare carte atunci când căutăm ceva. Exact ca și în biblioteci, unde există fișe a cărților, și în bazele de date putem crea index-uri pentru a le accesa mai ușor.

La exemplul de față titlul *The Adventures of Tom Sawyer* o să fie salvat: *The Adventures of To*.

```
ALTER TABLE classics ADD INDEX(author(20));  
ALTER TABLE classics ADD INDEX(title(20));  
ALTER TABLE classics ADD INDEX(category(4));  
ALTER TABLE classics ADD INDEX(year);  
DESCRIBE classics;
```

### Comenzi echivalente

```
ALTER TABLE classics ADD INDEX(author(20));  
CREATE INDEX author ON classics (author(20));
```

```
CREATE TABLE classics (  
  author VARCHAR(128),  
  title VARCHAR(128),  
  category VARCHAR(16),  
  year SMALLINT,  
  INDEX(author(20)),  
  INDEX(title(20)),  
  INDEX(category(4)),  
  INDEX(year))
```

### SELECT

```
SELECT author,title FROM classics;  
SELECT title,isbn FROM classics;  
SELECT COUNT(*) FROM classics;  
SELECT author FROM classics;  
SELECT DISTINCT author FROM classics;
```

### DELETE

```
DELETE FROM classics WHERE title='Little Dorrit';
```

## WHERE

```
SELECT author,title FROM classics WHERE author="Mark Twain";
SELECT author,title FROM classics WHERE isbn="9781598184891 ";
SELECT author,title FROM classics WHERE author LIKE "Charles%";
SELECT author,title FROM classics WHERE title LIKE "%Species";
SELECT author,title FROM classics WHERE title LIKE "%and%";
```

## LIMIT

```
SELECT author,title FROM classics LIMIT 3;
SELECT author,title FROM classics LIMIT 1,2;
SELECT author,title FROM classics LIMIT 3,1;
```

- Primul returnează primele trei înregistrări din tabel.
- Al doilea comandă returnează două înregistrări din tabel, începând de la poziția 1 (trece peste prima înregistrare)
- Al treilea returnează o singură rând din tabel, cel de pe poziția 3 (al patrulea rând)

## UPDATE

```
UPDATE classics SET author='Mark Twain (Samuel Langhorne Clemens)'
WHERE author='Mark Twain';
UPDATE classics SET category='Classic Fiction'
WHERE category='Fiction';
```

## ORDER BY

```
SELECT author,title FROM classics ORDER BY author;
SELECT author,title FROM classics ORDER BY title DESC;
SELECT author,title,year FROM classics ORDER BY author, year DESC;
SELECT author,title,year FROM classics ORDER BY author ASC, year DESC;
```

## GROUP BY

```
SELECT category,COUNT(author) FROM classics GROUP BY category;
```

## Accesarea MySQL din PHP



login.php

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

## Conectarea la o bază de date

### 1. Conectarea la o bază de date MySQLi – procedural

```
<?php
require_once 'login.php';
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully"; ?>
```

### 1. Conectarea la o bază de date MySQLi – object oriented

```
<?php
require_once 'login.php';
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";?>
```

## Citire date din baza de date

2. Formarea query de interogare a bazei de date
3. Aplicarea query-ului asupra bazei de date
4. Afișarea rezultatelor pe pagina web MySQLi – procedural

```
<?php
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"];
        echo "Name: " . $row["firstname"]. "<br>";
    }
} else {
    echo "0 results";
}
?>
```

2. Formarea query de interogare a bazei de date
3. Aplicarea query-ului asupra bazei de date
4. Afișarea rezultatelor pe pagina web MySQLi – object oriented

```
<?php
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"];
        echo "Name: " . $row["firstname"]. "<br>";
    }
} else {
    echo "0 results";
}
?>
```

## Înserare date în baza de date

2. Formarea query de interogare a bazei de date
3. Aplicarea query-ului asupra bazei de date
4. Afișarea rezultatelor pe pagina web MySQLi – procedural

```
<?php
$sql = "INSERT INTO MyGuests (firstname, lastname, email) VALUES ('John', 'Doe', 'john@example.com')";
if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}
?>
```



2. Formarea query de interogare a bazei de date
3. Aplicarea query-ului asupra bazei de date
4. Afișarea rezultatelor pe pagina web MySQLi – object oriented

```
<?php
$sql = "INSERT INTO MyGuests (firstname, lastname, email) VALUES ('John', 'Doe',
'john@example.com')";
if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
?>
```

#### Ștergere date din baza de date

2. Formarea query de interogare a bazei de date
3. Aplicarea query-ului asupra bazei de date
4. Afișarea rezultatelor pe pagina web MySQLi – procedural

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

2. Formarea query de interogare a bazei de date
3. Aplicarea query-ului asupra bazei de date
4. Afișarea rezultatelor pe pagina web MySQLi – object oriented

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

#### Actualizare date în baza de date

2. Formarea query de interogare a bazei de date
3. Aplicarea query-ului asupra bazei de date
4. Afișarea rezultatelor pe pagina web MySQLi – procedural

```
<?php
$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";
if (mysqli_query($conn, $sql)) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . mysqli_error($conn);
}
?>
```

2. Formarea query de interogare a bazei de date
3. Aplicarea query-ului asupra bazei de date
4. Afișarea rezultatelor pe pagina web MySQLi – object oriented

```
<?php
$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";
if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . $conn->error;
}
?>
```

#### Deconectare de la baza de date

5. Deconectarea de la MySQL  
MySQLi – procedural

```
<?php
mysqli_close($conn);
?>
```

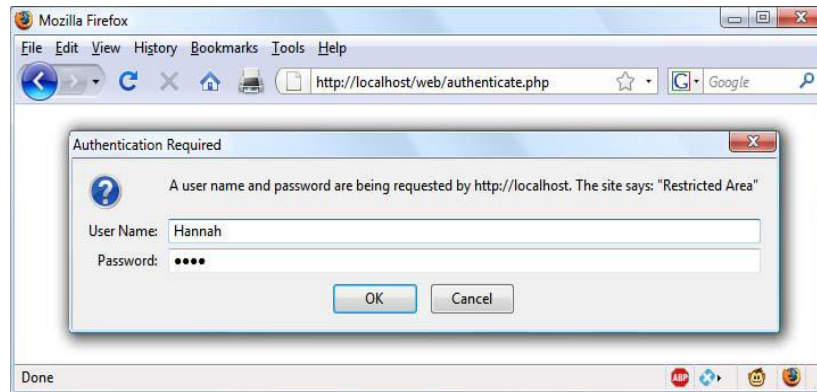
5. Deconectarea de la MySQL  
MySQLi – object oriented

```
<?php
$conn->close();
?>
```

## Autentificare și managementul sesiunii în PHP

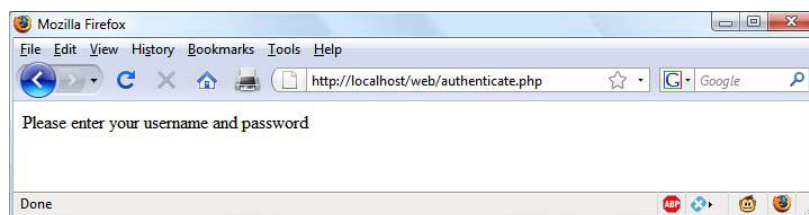
### Autentificare HTTP

- Se utilizează serverul web pentru a administra lista utilizatorilor și a parolelor



### Autentificare HTTP

```
<?php
if (isset($_SERVER['PHP_AUTH_USER']) && isset($_SERVER['PHP_AUTH_PW']))
{
    echo "Welcome User: " . $_SERVER['PHP_AUTH_USER'] . " Password: " .
    $_SERVER['PHP_AUTH_PW'];
}
else
{
    header('WWW-Authenticate: Basic realm="Restricted Section"');
    header('HTTP/1.0 401 Unauthorized');
    die("Please enter your username and password");
}
?>
```



### Autentificare HTTP cu verificarea credențialelor

```
<?php
$username = 'admin';
$password = 'letmein';
if (isset($_SERVER['PHP_AUTH_USER']) && isset($_SERVER['PHP_AUTH_PW']))
{
    if ($_SERVER['PHP_AUTH_USER'] == $username && $_SERVER['PHP_AUTH_PW'] ==
    $password)
        echo "You are now logged in";
    else die("Invalid username / password combination");
}
```

```

}
else
{
    header('WWW-Authenticate: Basic realm="Restricted Section"');
    header('HTTP/1.0 401 Unauthorized');
    die ("Please enter your username and password");
}
?>

```

Salvarea numelor de utilizator și a parolelor:

- Criptare cu md5: 32 caractere hexazecimale
- Criptare cu sha1: 40 caractere hexazecimale

```

$token = md5('mypassword');
34819d7beeabb9260a5c854bc85b3e44

```

```

$token = md5('saltstringmypassword');
$token = md5('hqb%$tmypasswordcg*I');

```

Adăugarea utilizatorilor într-o bază de date

```

<?php
$salt1 = "qm&h*";
$salt2 = "pg!@";
$forename = 'Bill';
$surname = 'Smith';
$username = 'bsmith';
$password = 'mysecret';
$token = md5("$salt1$password$salt2");
add_user($forename, $surname, $username, $token);
$forename = 'Pauline';
$surname = 'Jones';
$username = 'pjones';
$password = 'acrobat';
$token = md5("$salt1$password$salt2");
add_user($forename, $surname, $username, $token);
function add_user($fn, $sn, $un, $pw)
{
    $query = "INSERT INTO users VALUES('$fn', '$sn', '$un', '$pw')";
    $result = mysql_query($query);
    if (!$result) die ("Database access failed: " . mysql_error());
}
?>

```

Autentificare HTTP folosind MySQL

```

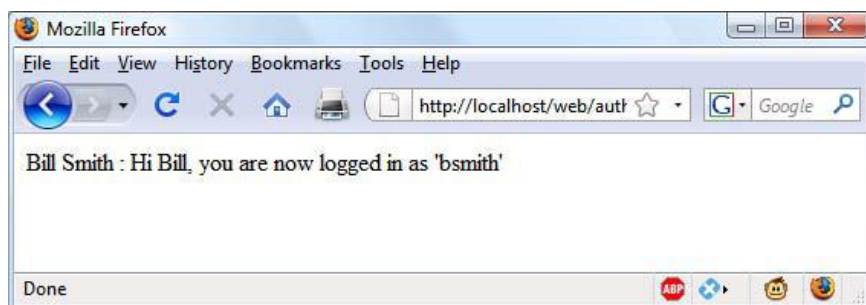
if (isset($_SERVER['PHP_AUTH_USER']) && isset($_SERVER['PHP_AUTH_PW']))
{

```

```

$un_temp = mysql_entities_fix_string($_SERVER['PHP_AUTH_USER']);
$pw_temp = mysql_entities_fix_string($_SERVER['PHP_AUTH_PW']);
$query = "SELECT * FROM users WHERE username='$un_temp'";
$result = mysql_query($query);
if (!$result)
    die("Database access failed: " . mysql_error());
elseif (mysql_num_rows($result))
{
    $row = mysql_fetch_row($result);
    $salt1 = "qm&h*";
    $salt2 = "pg!@";
    $token = md5("$salt1$pw_temp$salt2");
    if ($token == $row[3])
        echo "$row[0] $row[1] : Hi $row[0]; logged in as '$row[2]'";
    else die("Invalid username/password combination");
}
else die("Invalid username/password combination");
}
else
{
    header('WWW-Authenticate: Basic realm="Restricted Section"');
    header('HTTP/1.0 401 Unauthorized');
    die ("Please enter your username and password");
}

```



- Start sesiune: session\_start()
- Salvare variabile de sesiune: \$\_SESSION['variable'] = \$value;
- Terminarea unei sesiuni: session\_destroy();
- Timeout pentru sesiune (1 zi; implicit 1440 secunde): ini\_set('session.gc\_maxlifetime', 60 \* 60 \* 24);

```

<?php
function destroy_session_and_data()
{
    session_start();
    $_SESSION = array();
    if (session_id() != "" || isset($_COOKIE[session_name()]))
        setcookie(session_name(), "", time() - 2592000, '/');
    session_destroy();
}
?>

```

## Securitatea sesiunii

```
$_SESSION['ip'] = $_SERVER['REMOTE_ADDR'];
$_SESSION['ua'] = $_SERVER['HTTP_USER_AGENT'];
$_SESSION['check'] = md5($_SERVER['REMOTE_ADDR'].$_SERVER['HTTP_USER_AGENT']);
if ($_SESSION['ip'] != $_SERVER['REMOTE_ADDR'])
    different_user();
if ($_SESSION['ua'] != $_SERVER['HTTP_USER_AGENT'])
    different_user();
if ($_SESSION['check'] != md5($_SERVER['REMOTE_ADDR'].$_SERVER['HTTP_USER_AGENT']))
    different_user();
```

## Regenerarea sesiunii

```
<?php
session_start();
if (!isset($_SESSION['initiated']))
{
    session_regenerate_id();
    $_SESSION['initiated'] = 1;
}
if (!isset($_SESSION['count']))
    $_SESSION['count'] = 0;
else
    ++$_SESSION['count'];
echo $_SESSION['count'];
?>
```

## Cuprins

Introducere în MySQL .....	1
Interacțiunea cu MySQL:.....	1
Reguli: .....	2
Comenzi MySQL .....	2
Tipuri de date .....	3
Operațiuni cu tabele .....	4
Indexarea tabelor .....	5
SELECT .....	5
DELETE.....	5
WHERE .....	6
LIMIT .....	6
UPDATE .....	6
ORDER BY .....	6
GROUP BY.....	6
Accesarea MySQL din PHP .....	7
Conectarea la o bază de date.....	7
Citire date din baza de date .....	8
Înserare date în baza de date .....	8
Ștergere date din baza de date .....	9
Actualizare date în baza de date.....	9
Deconectare de la baza de date.....	10
Autentificare și managementul sesiunii în PHP .....	11
Adăugarea utilizatorilor într-o bază de date.....	12
Securitatea sesiunii .....	14
Regenerarea sesiunii.....	14