

## Utilizarea bazelor de date folosind PHP

- Conectare la serverul MySQL folosind functiile PHP
- Detectare erori aparute in lucrul cu bazele de date
- Creare baze de date si tabele folosind functii PHP
- Adaugare date in tabel folosind PHP

Limbajul PHP a devenit atat de popular datorita faptului ca acesta contine un suport de functii pentru lucrul cu bazele de date.

### 1. Conectarea la serverul MySQL

Pentru a utiliza o baza de date MySQL, avem nevoie de o conexiune la serverul de baze de date. Aceasta conexiune va fi utilizata ca punct de acces pentru comenzi mysql.

Pentru a va conecta la un server MySQL, utilizati functia **mysql\_connect()**, a carei sintaxa este urmatoarea:

**mysql\_connect(*nume\_gazda*, *nume\_utilizator*, *parola*)**

Unde "nume\_gazda" este numele gazdei, serverul pe care ruleaza serviciul MySQL, "nume\_utilizator" este numele utilizatorului care se conecteaza la serverul MySQL, iar "parola" este parola MySQL asociata utilizatorului.

In cazul in care serverul MySQL ruleaza pe calculatorul personal, parametrul "nume gazda" este in general "localhost". In alte cazuri, daca serverul MySQL ruleaza pe un alt computer, va trebui sa se specifice adresa computerului pe care ruleaza serverul MySQL.

Functia returneaza FALSE în caz de eroare; în caz contrar, returneaza o valoare denumita identificator de legatura, care serveste ca instrument de manipulare pentru accesul la serverul MySQL.

Exemplu de utilizare a functiei **mysql\_connect()**:

```
<?php
$mysql=mysql_connect("localhost","root","");
if(!$mysql){
    echo "Nu s-a realizat conectarea la MySQL";
    exit;
}
echo "Conexiune la MySQL <br />";
?>
```

In acest exemplu, serverul MySQL ruleaza pe "localhost" ca si modulul PHP, utilizatorul este "root" iar parola "". Dupa executia acestui script, daca nu se reuseste conectarea la MySQL, va aparea mesajul "Nu s-a realizat conexiunea la MySQL", in caz de reusita, apare mesajul "Conexiune la MySQL"

În mod prestabilit, funcția **mysql\_connect()** se conectează la serviciul MySQL prin intermediul portului 8888, portul MySQL standard. Dacă doriți să obțineți accesul la un server MySQL care rulează pe un port non-standard, puteți atașa un caracter două puncte și numărul portului dorit la argumentul care conține numele gazdei; de exemplu, "localhost:8888".

## 2. Încheierea conexiunii la serverul MySQL

Pentru terminarea unei conexiuni la serverul MySQL se folosește instrucțiunea: **mysql\_close()**. Această funcție închide conexiunea curentă. Funcția returnează TRUE în caz de reușită; în caz contrar, returnează FALSE. În general, nu este necesară invocarea funcției **mysql\_close()**, deoarece PHP închide automat conexiunile deschise cu bazele de date atunci când un script își încheie execuția.

Exemplu de conectare la serverul MySQL și închidere a conexiunii folosind funcția **mysql\_close()**:

```
<?php
$mysql=mysql_connect("localhost","root","");
if(!$mysql){
    echo "Nu s-a realizat conectarea la MySQL";
    exit;
}
// Aici adăugați datele necesare lucrului cu serverul MySQL
mysql_close();
?>
```

## 3. Detectarea apariției erorilor

Biblioteca MySQL din PHP furnizează două funcții de verificare a erorilor:

**mysql\_errno()** și **mysql\_error()**

Fiecare funcție returnează un rezultat care reflectă eroarea, dacă aceasta există.

Nici una din cele două funcții nu necesită argumente. Funcția **mysql\_errno()** returnează un cod numeric de eroare, în timp ce funcția **mysql\_error()** returnează un șir care descrie eroarea. Dacă nu s-a produs nici o eroare, codul numeric al erorii este zero și descrierea are ca valoare un șir vid.

Informațiile de eroare sunt disponibile numai dacă este activă o conexiune cu serverul MySQL. Prin urmare, întâi trebuie să se reușească conectarea la MySQL.

Exemplu de folosire a funcțiilor respective pentru a verifica modul de lucru a funcției **mysql\_select\_db()** (aceasta selectează baza de date pentru lucru):

```
<?php
$mysql=mysql_connect("localhost","root","");
if(!$mysql){
    echo "Nu s-a realizat conectarea la MySQL";
    exit;
}
echo "Conexiune la MySQL <br />";
mysql_select_db("lucrubd");
```

```

if (mysql_error()) {
    echo "<br />". mysql_errno(). " : ". mysql_error();
}
?>

```

Daca se reuseste conexiunea la serverul MySQL si baza de date cu numele "lucru db" este inexistentă, rezultatul afisat de exemplul de mai sus va fi urmatorul:

Conexiune la MySQL

1049 : Unknown database 'lucru db'

#### **4. Crearea si selectarea unei baze de date.**

O functie importanta in lucrul cu bazele de date este **mysql\_query()**, care are urmatoarea forma:

**mysql\_query("interogare")**

*sau*

**mysql\_query("interogare", "id\_con")**

- "interogare" este un sir ce contine comenzile SQL care urmeaza a fi executate (*in PHP, comenzile SQL nu trebuie sa se încheie cu un caracter punct si virgula*).

- "id\_con" este identificatorul de conectare returnat de functia *mysql\_connect()*, daca acesta este omis se foloseste ultima legatura deschisa cu aceasta functie.

Functia **mysql\_query()** returneaza TRUE daca serverul a reusit sa execute interogarea; în caz contrar, returneaza FALSE.

- **Important: pentru ca sa puteti lucra cu baze de date in PHP trebuie sa cunoasteti comenzile SQL necesare.**

Pentru a crea o noua baza de date folosim comanda SQL **"CREATE DATABASE numeBazaDeDate;"**, pe care o putem lansa de la o consola MySQL, sau o putem apela de la un script PHP prin intermediul functiei **mysql\_query()**.

Exemplu prin care creem baza de date "lucru db":

```

<?php
$mysql=mysql_connect("localhost","root","");
if(!$mysql){
    echo "Nu s-a realizat conectarea la MySQL";
    exit;
}
echo "Conexiune la MySQL <br />";
$createdb = mysql_query("CREATE DATABASE lucru db");
if ($createdb)
    echo "Baza de date lucru db a fost creata <br />";
else
    echo "<br />". mysql_errno(). " : ". mysql_error();
mysql_close();
?>

```

- Dupa executarea acestui script, daca totul este corect, va fi creata baza de date "lucru db" si va fi afisat mesajul "Baza de date lucru db a fost creata".

Dupa ce programul a obtinut o conexiune cu serverul MySQL, înainte de a lucra cu tabelele dintr-o baza de date

MySQL, programul trebuie sa selecteze baza de date respectiva. Pentru aceasta, folositi functia **mysql\_select\_db()**, care are urmatoarea forma:

- **mysql\_select\_db(baza\_de\_date)**

Unde "baza\_de\_date" este numele bazei de date la care urmeaza a se obtine acces. Functia returneaza TRUE daca poate obtine accesul la baza de date, respectiv FALSE în caz contrar. Iata un exemplu de selectare a bazei de date "lucru db":

```
<?php
$mysql=mysql_connect("localhost","root","");
if(!$mysql){
    echo "Nu s-a realizat conectarea la MySQL";
    exit;
}
echo "Conexiune la MySQL <br />";

$selectdb = mysql_select_db('lucru db');
if ($selectdb)
    echo "Baza de date lucru db a fost selectata <br />";
else
    echo "<br />". mysql_errno(). " : ". mysql_error();
mysql_close();
?>
```

## **5. Crearea tabelelor**

Dupa conectare si selectarea bazei de date putem crea tabele in cadrul bazei de date selectate. Pentru aceasta folosim limbajul SQL **CREATE TABLE numeTabel (coloana1, coloana2, ...)**; impreuna cu functia **mysql\_query()**.

In exemplul urmat cream tabelul cu numele "carti" care contine urmatoarele campuri:

Nume camp	Tip
id	Identificator unic, Not Null, cheie primara
nume	Char(30)
autor	Char(25)
gen	Char(10)
data_intrare	Date
pret	Decimal(12,2)

In stanga, la "Nume Camp", sunt trecute coloanele tabelului, iar in dreapta tipul acestora, coloana cu nume "id" fiind cheia primara.

De asemenea, folosim indicatorul "AUTO\_INCREMENT" pentru a preciza faptul ca MySQL va repartiza o valoare secventiala unica în coloana care serveste drept cheie primara a tabelului.

Iata si scriptul PHP pentru crearea tabelului:

```
<?php
$mysql=mysql_connect("localhost","root","");
```

```

if(!$mysql){
    echo "Nu s-a realizat conectarea la MySQL";
    exit;
}

echo "Conexiune la MySQL <br />";
$selectdb = mysql_select_db('lucrddb');
if (!$selectdb)
    echo "Baza de date lucrddb nu a putut fi selectata deoarece : ". mysql_errno(). " : ". mysql_error();
$sql = "CREATE TABLE `carti` (id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY, nume varchar(30) NOT NULL, autor
varchar(25) , gen varchar(10) NOT NULL, data_intrare date, pret decimal(12,2))";
if (mysql_query($sql, $conn))
    echo "Tabelul carti a fost creat <br />";
else
    echo "Tabelul carti nu a putut fi creat deoarece : ". mysql_errno(). " : ". mysql_error();
mysql_close();
?>

```

In exemplul de mai sus, variabila "\$sql" contine comanda SQL pentru crearea acestui tabel. Instructiunea **mysql\_query(\$sql, \$conn)**, executa aceasta comanda si in cazul in care rezultatul este TRUE se trimite la iesire sirul: "Tabelul carti a fost creat". In cazul in care instructiunea SQL nu a putut fi executata, functia **mysql\_query(\$sql)** intoarce valoarea FALSE, afisand eroarea din cauza careia tabelul nu a putut fi creat.

## 6. Adaugarea de date in tabel

Dupa ce aveti tabelul creat in baza de date MySQL puteti adauga date in el, pentru aceasta se foloseste functia **mysql\_query("interogare")** unde "*interogare*" este urmatoarea comanda:

**INSERT INTO `nume\_tabel` (coloana1, coloana2, ...) VALUES ('date1', 'date2', ...)**

- unde *nume\_tabel* este numele tabelului in care se adauga datele
- "*coloana1, coloana2, ...*" sunt numele coloanelor in care se doreste adaugarea datelor
- "*date1, date2, ...*" sunt datele care vor fi adaugate, tinandu-se cont de ordinea si numarul lor cu cea a coloanelor

Iata un exemplu, in care se foloseste tabelul "carti" creat mai sus

```

<?php
$mysql=mysql_connect("localhost","root","");
if(!$mysql){
    echo "Nu s-a realizat conectarea la MySQL";
    exit;
}

echo "Conexiune la MySQL <br />";
$selectdb = mysql_select_db('lucrddb');
if (!$selectdb)
    echo "Baza de date lucrddb nu a putut fi selectata deoarece : ". mysql_errno(). " : ". mysql_error();

$sql = "INSERT INTO `carti` (nume, autor, gen, data_intrare, pret) VALUES ('Glossa', 'Mihai Eminescu', 'poezie', '19-05-2009', 8)";
if (mysql_query($sql, $conn))

```

```
    echo 'Datele au fost adaugate';
else
    echo "Datele nu au fost adaugate deoarece : ". mysql_errno(). " : ". mysql_error();
mysql_close();
?>
```

- In acest exemplu, datele "'Glossa', 'Mihai Eminescu', 'poezie', '19-05-2009', '8'" vor fi adaugate in coloanele "nume, autor, gen, data\_intrare, pret" in aceeasi ordine in care sunt scrise si datele si coloanele.
- Daca au fost adaugate cu succes, apare mesajul "Datele au fost adaugate", in caz contrar apare mesajul de eroare.

Daca doriti sa adaugati mai multe linii in tabel, printr-o singura interogare, puteti folosi urmatorul model

```
<?php
// Aici adaugati comenzile pt. conectare si selectarea bazei de date

$sql = "INSERT INTO `nume_tabel` (coloana1, coloana2, coloana3)
VALUES
    ('date1', 'date2', 'date3'),
    ('valoare1', 'valoare2', 'valoare3'),
    ('text1', 'text2', 'text3')";
if (mysql_query($sql))
    echo 'Datele au fost adaugate';
else
    echo "Datele nu au fost adaugate deoarece : ". mysql_errno(). " : ". mysql_error();
mysql_close();
?>
```

---