

ELEMENTE CRUD

Când lucrăm cu baze de date în PHP, PHP 5 și versiunile ulterioare pot funcționa cu baza de date MySQL folosind: **MySQLi** ("I" – improved perfectionat) și **PDO** (PHP DATA OBJECTS). Ce alegem?

Fiindcă există diferențe între ele ce tin de

1. suportul bazei de date,
2. de stabilitate și
3. problemele de performanță.

mysql – este vechi și nu este în uz (pina la versiunea PHP 4)(din 2012);

mysqli – recomandabil pentru a fi utilizat php; (obiectual și procedural!!)

pdo – este ceva mai încet ca mysqli, dar are anumite avantaje

	PDO	MySQLi
Suport BD	12 tipuri diferite	Doar MySQL
API	POO	POO+component procedurală

Voi prezenta în continuare cele trei moduri de a lucra cu PHP și MySQL:

https://www.w3schools.com/php/php_mysql_create.asp

- MySQLi (obiect orientat)
- MySQLi (procedural)
- DOP

Instalarea MySQLi Pentru Linux și Windows: Extensia MySQLi este instalată automat în majoritatea cazurilor, când este instalat pachetul **php5, mysql in sus**.

CONEXIUNEA LA MYSQL / ÎNAINTE DE A PUTEA ACCESA DATELE DIN BAZA DE DATE MYSQL, TREBUIE SĂ NE PUTEAM CONECTA LA SERVER/

Conectare la BD MYSQL utilizând MySQLi (obiect orientat)

```
<?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";
?>
```

Conectare la BD MYSQL utilizând MySQLi (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";
?>
```

Conectare la BD MYSQL utilizând PDO

```
<?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();
}
?>
```

INCHIDEREA CONEXIUNII SE REALIZEAZĂ CU AJUTORUL COMENZILOR RESPECTIVE

MySQLi Object-Oriented:

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

MySQLi Procedural:

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

PDO:

```
$conn = null;
```

CREATE

CREAREA BD MYSQL UTILIZÂND MYSQLI (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());
}
// Create database
$sql = "CREATE DATABASE myDB";
if (mysqli_query($conn, $sql))
{
    echo "Database created successfully";
}
else
{
    echo "Error creating database: " . mysqli_error($conn);
}
mysqli_close($conn);
?>

```

CREAȚI UN TABEL MYSQL FOLOSIND MYSQLI ȘI PDO

Instrucțiunea CREATE TABLE este utilizată pentru a crea un tabel în MySQL.

Vom crea un tabel numit „MyDB”, cu cinci coloane: „id”, „prenume”, „prenume”, „e-mail” și „reg_date”:

Crearea BD MYSQL utilizând MySQLi (procedural)

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
// sql to create table
$sql = "CREATE TABLE MyDB (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
)";
if (mysqli_query($conn, $sql))
{
    echo "Table MyGuests created successfully";
}
else
{
    echo "Error creating table: " . mysqli_error($conn);
}

```

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

```
?>
```

INSERAREA DATELOR IN TABELUL BD MYSQL UTILIZÂND MYSQLI (PROCEDURAL)

Introduceți data în MySQL folosind MySQLi și PDO

După ce au fost create o bază de dată și un tabel, putem începe să adăugăm date în ele.

Iată câteva reguli de sintaxă de urmat:

- Interogarea SQL **trebuie** să fie **inclusă în ghilimele** în PHP
- Valorile șirului din interogarea SQL **trebuie** să fie **incluse în ghilimele**
- Valorile numerice **nu trebuie incluse în ghilimele**
- Cuvântul NULL **nu trebuie inclusă în ghilimele**

Instrucțiunea INSERT INTO este utilizată pentru a adăuga înregistrări noi la un tabel MySQL:

Formatul

```
INSERT INTO table_name (column1, column2, column3,...)
```

```
VALUES (value1, value2, value3,...)
```

Anterior am creat un tabel gol numit „MyDB” cu cinci coloane: „id”, „prenume”, „prenume”, „e-mail” și „reg_date”.

Notă: Dacă o coloană este **AUTO_INCREMENT** (cum ar fi coloană „id”), nu este necesar să fie specificat în interogarea SQL; MySQL va adăuga automat informații.

Următorul exemplu inserează o nouă înregistrare în tabelul „MyDB”:

INSERAREA DATELOR IN TABELUL BD MYSQL UTILIZÂND MYSQLI (PROCEDURAL)

```
<?php
```

```
$servername = "localhost";
```

```
$username = "username";
```

```
$password = "password";
```

```
$dbname = "myDB";
```

```
// Create connection
```

```
$conn = mysqli_connect($servername, $username, $password, $dbname);
```

```
// Check connection
```

```
if (!$conn)
```

```
{
```

```
    die("Connection failed: " . mysqli_connect_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_close($conn);
```

```
?>
```

INTRODUCEREA MAI MULTOR ÎNREGISTRĂRI ÎN MYSQL FOLOSIND MYSQLI(PROCEDURAL)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn)
{
die("Connection failed: " . mysqli_connect_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_close($conn);
?>
```

SELECTAREA DATELOR DINTR-O BAZĂ DE DATE MYSQL FOLOSIND MYSQLI(PROCEDURAL)

Instrucțiunea SELECT este utilizată pentru a selecta date dintr-unul sau mai multe tabele:

```
SELECT column_name(s) FROM table_name
```

sau putem folosi caracterul * pentru a selecta TOATE coloanele dintr-un tabel:

```
SELECT * FROM table_name
```

Mai întâi, se configurează o interogare SQL care selectează coloanele **id**, **firstname** și **lastname** din tabelul MyBD. Următoarea linie de cod execută interogarea și pune rezultatele rezultate într-o variabilă numită **\$result**.

Apoi, **function num_rows()** verifică dacă există mai multe de zero rânduri returnate.

Dacă sunt returnate mai mult de zero rânduri, funcția **fetch_assoc()** pune toate rezultatele într-o matrice asociativă pe care o putem parcurge. Bucla **while()** în ciclu prezintă rezultatul ce setează ieșirile date din **id**, **firstname** și **lastname**.

URMĂTORUL EXEMPLU ARATĂ FELUL CA EXEMPLU DE MAI SUS, ÎN MODUL PROCEDURAL MYSQLI:

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn)
{
    die("Connection failed: " . mysqli_connect_error());
}
$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"]. " " . $row["lastname"]. "<br>";
    }
}
else
{
    echo "0 results";
}
mysqli_close($conn);
?>

```

1. Variabila **\$result** conține setul de rezultate. Este un obiect **mysqli_result**.
2. Pentru a parcurge toate rândurile din acest set de rezultate, trebuie să utilizăm o funcție **mysqli_fetch_***, cum ar fi **mysqli_fetch_assoc ()** folosită în exemplu.
3. Aceste funcții obțin rândul curent din setul de rezultate și îl returnează într-o **PHP** matrice/array standart sau într-un obiect PHP standard.
4. În exemplu, **mysqli_fetch_assoc ()** returnează o matrice asociativă/**associative array** în care indicile matricei este numele coloanei (nume și preț) și valorile sunt valorile coloanei.
5. După fiecare iterație, indicatorul intern al setului de rezultate se deplasează cu 1 rând, astfel încât data următoare **mysqli_fetch_assoc ()** va citi următorul rând, până când nu mai rămân mai multe rânduri.
6. **İată ce se întâmplă în bucla while din exemplul de mai sus:**
7. **mysqli_fetch_assoc ()** citește primul rând din **\$result** și returnează o matrice/array asociativă în interiorul **\$row** Comanda **echo** tipărește:
8. **"id: " . \$row["id"]. " - Name: " . \$row["firstname"]. " " . \$row["lastname"]**.
9. La următoarea iterație while, se citește următorul row/rând și se imprimă comanda **echo**:
10. **"id: " . \$row["id"]. " - Name: " . \$row["firstname"]. " " . \$row["lastname"]**.
11. Când nu mai există rânduri/rows, **mysqli_fetch_assoc ()** returnează **false** și bucla while se oprește
12. Dacă interogarea nu returnează niciun rând/row, bucla while nu rulează nici măcar o dată.

Comanda de SELECT/SELECȚIE A DATELELOR DINTR-O BAZĂ DE DATĂ MYSQL (MYSQLI PROCEDURAL)

Clauza ORDER BY este utilizată pentru a sorta setul de rezultate în ordine crescătoare sau descendentă. Clauza ORDER BY sortează înregistrările în ordine crescătoare în mod implicit. Pentru a sorta înregistrări în ordine descrescătoare, utilizați cuvântul cheie DESC.

Formatul

SELECT column_name(s) FROM table_name ORDER BY column_name(s) ASC|DESC

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn)
{
    die("Connection failed: " . mysqli_connect_error());
}
$sql = "SELECT id, firstname, lastname FROM MyGuests ORDER BY lastname";
$result = mysqli_query($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"]. " " . $row["lastname"]. "<br>";
    }
}
else
{
    echo "0 results";
}
mysqli_close($conn);
?>
```

ȘTERGEREA DATELOR DINTR-UN TABEL MYSQL FOLOSIND MYSQLI PROCEDURAL

Instrucțiunea DELETE este utilizată pentru a șterge înregistrările dintr-un tabel:

**DELETE FROM table_name
WHERE some_column = some_value**

Observați clauza WHERE din sintaxa DELETE: Clauza WHERE specifică înregistrare sau înregistrări care ar trebui șterse. Dacă omiteți clauza WHERE, toate înregistrările vor fi șterse!

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn)
{
    die("Connection failed: " . mysqli_connect_error());
}
// sql to delete a record
$sql = "DELETE FROM MyDB WHERE id=3";
if (mysqli_query($conn, $sql))
{
    echo "Record deleted successfully";
}
else
{
    echo "Error deleting record: " . mysqli_error($conn);
}
mysqli_close($conn);
?>

```

UPDATE

ACTUALIZARE DATE DINTR-UN TABEL MYSQL FOLOSIND MYSQLI PROCEDURAL

Instrucțiunea UPDATE este utilizată pentru actualizarea înregistrărilor existente într-un tabel:

```

UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value

```

Observați clauza WHERE din sintaxa UPDATE: Clauza WHERE specifică înregistrare sau înregistrări care ar trebui actualizate. Dacă omiteți clauza WHERE, toate înregistrările vor fi actualizate!

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn)
{
    die("Connection failed: " . mysqli_connect_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);  
}  
mysqli_close($conn);  
?>
```