

Funcții PHP pentru operare cu MySQL

resource `mysql_connect` - realizează conectarea la un server MySQL;

- funcția necesită trei parametri uzuali: nume server MySQL/ adresă IP server, nume utilizator MySQL și parola de acces;
- returnează valoarea logică FALSE în cazul în care a eșuat conectarea la server-ul de baze de date, respectiv un identificator resursă de acces la conexiunea cu server-ul MySQL în cazul în care conectarea a reușit.

resource `mysql_pconnect` - realizează o conexiune permanentă la un server MySQL (parametrii fiind aceiași ca la funcția anterior);

- inițial, caută o conexiune permanentă deja existentă, iar în cazul în care nu găsește, creează una;
- nu închide conexiunea la final de script;
- returnează valoarea logică FALSE în cazul în care a eșuat conectarea la server, respectiv un identificator resursă de acces la conexiunea cu MySQL în cazul în care conectarea a reușit.

bool `mysql_close` - primește ca parametru un identificator de acces la o conexiune cu un server MySQL și realizează închiderea acesteia. În cazul în care parametrul nu este specificat, se va închide conexiunea curentă;

- returnează valoarea logică TRUE în cazul în care închiderea conexiunii s-a realizat cu succes, respectiv valoarea logică FALSE în caz contrar;
- funcția este opțională în cazul în care conexiunea nu este permanentă, aceasta închizându-se automat la final de script.

bool `mysql_select_db` - selectează/deschide o baza de date curentă (existând deja o conexiune cu un server MySQL) pentru interogările/operațiile SQL care vor urma;

- funcția are doi parametri: primul este de tip șir de caractere, reprezentând numele bazei de date deschise/selectate, iar cel de-al doilea parametru reprezintă identificatorul resursă de acces la conexiunea cu serverul MySQL;
- returnează valoarea logică TRUE în cazul în care s-a reușit selectarea bazei de date, respectiv valoarea logică FALSE în cazul în care baza de date nu poate fi deschisă (nu există, utilizatorul nu are drepturi de acces la aceasta, etc).

bool `mysql_create_db` - permite crearea unei noi baza de date MySQL;

- funcția are doi parametri: primul este de tip șir de caractere, reprezentând numele bazei de date care va fi creată, iar cel de-al doilea parametru reprezintă identificatorul aferent conexiunii cu MySQL-ul;
- returnează valoarea logică TRUE în cazul în care a reușit să creeze baza de date, respective valoarea logică FALSE în caz contrar.

bool `mysql_drop_db` - permite ștergerea unei baze de date MySQL;

- funcția are doi parametri: primul este de tip șir de caractere, reprezentând numele bazei de date care va fi ștearsă, iar cel de-al doilea parametru reprezintă identificatorul aferent conexiunii cu MySQL-ul;
- returnează valoarea logică TRUE în cazul în care a reușit să șteargă baza de date, respectiv valoarea logică FALSE în caz contrar.

resource `mysql_query` - trimite/pasează o comandă SQL spre serverului MySQL (sub forma unui *string* de caractere);

- returnează, în caz de succes, un identificator resursă care va fi folosit în continuare ca parametru în apelurile unor eventuale funcții, (realizând extragerea rezultatului interogării), iar în caz contrar, returnează valoarea logică FALSE.

array `mysql_fetch_row` - returnează o înregistrare/linie extrasă dintr-o tabelă interogată (sau joncțiune de tabele), sub forma unui șir (*array* cu primul indice de element 0).

- presupunând \$row numele *array*-ului returnat, elementele acestuia pot fi referite pa baza indicelui lor: \$row[0], \$row[1], ...

array `mysql_fetch_assoc` - aceeași funcționalitate ca și `mysql_fetch_row`, dar elementele șirului returnat sunt referite nu prin indici, ci prin nume asociate lor, identice cu numele coloanelor referite prin interogare: `$row[coloana1]`, `$row[coloana2]`, ...

array `mysql_fetch_array` - aceeași funcționalitate ca și `mysql_fetch_row` / `mysql_fetch_assoc`, în varianta implicită (fără parametri opționali), permițând ambele moduri de referire a elementele șirului returnat: fie prin indice, fie prin numele asociat (numele coloanei).
- deci se poate face o referire echivalentă a unui element: `$row[0]` sau `$row[coloana1]`

object `mysql_fetch_object` - aceeași funcționalitate ca și `mysql_fetch_row`, returnând însă un tip *object*, datele putând fi referite ca proprietăți ale acestuia:
`$row->coloana1`,
`$row->coloana2`, ...

int `mysql_num_rows` - returnează numărul de linii obținute în urma executării unei comenzi SQL de interogare SELECT.

int `mysql_num_fields` - returnează numărul de coloane (câmpuri) aferente setului de date obținut ca urmare a execuției unei comenzi SQL SELECT.

resource `mysql_list_fields` - permite oferirea de informații asupra unei anumite tabele (joncțiuni de tabele), returnând ca rezultat un identificator resursă, care poate fi folosit de către următoarele funcții conexe:
`mysql_field_name`, `mysql_field_len`,
`mysql_field_type`.

string `mysql_field_name` - returnează succesiv numele coloanelor tablei.

int `mysql_field_len` - returnează succesiv dimensiunea coloanelor tablei.

string `mysql_field_type` - returnează succesiv tipul coloanelor tablei.

int `mysql_affected_rows` - returnează numărul de linii afectate de o comandă SQL anterioară de tip UPDATE, INSERT, DELETE.

int `mysql_errno` - permite identificarea erorii care s-a produs în timpul execuției de către serverul MySQL a unei comenzi SQL;
- returnează codul numeric al erorii în cazul în care a apărut o eroare în timpul executării unei comenzii SQL, respectiv 0 în caz contrar.

string `mysql_error` - permite identificarea erorii care s-a produs în timpul executării unei comenzi SQL;
- returnează un mesaj în care eroarea este descrisă explicit în cazul în care a apărut o eroare în timpul executării unei comenzii SQL, respective un șir vid în caz contrar.

tip_variabil `mysql_result` - preia valoarea unei celule din setul de date, aferente unei anumite coloane dintr-un rând al unui cursor, rezultat ca urmare a unei comenzi SQL SELECT.

bool `mysql_free_result` - realizează eliberarea zonei de memorie ocupată de cursorul indicat ca argument, rezultat ca urmare a unei comenzi SQL SELECT;
- returnează valoarea logică TRUE în caz de succes, respectiv valoarea logică FALSE în caz contrar.

Observatie

Cele mai folosite funcții pentru lucrul cu baze de date MySQL sunt următoarele:

- `mysql_connect` - permite conectarea la un server MySQL;

- mysql_select_db - permite selecția bazei de date MySQL;
- mysql_query - execută o comandă SQL;
- mysql_fetch_row - returnează o linie din tabelă ca șir;
- mysql_num_fields - returnează numărul de coloane(câmpuri) dintr-un set de rezultate creat ca urmare a unei comenzi SELECT;
- mysql_num_rows - returnează numărul de linii obținute în urma executării unei comenzi SQL SELECT;
- mysql_list_fields - oferă informații asupra unei anumite tabele, returnând ca rezultat un identificator, care poate fi folosit de către următoarele funcții: mysql_field_name, mysql_field_len, mysql_field_type;
- mysql_field_name - returnează succesiv numele câmpurilor tablei;
- mysql_field_type - returnează succesiv tipul câmpurilor tablei;
- mysql_field_len - returnează succesiv lungimea câmpurilor tablei;
- mysql_close - închide o conexiune cu baza de date

Aplicatii

1. Determinarea adresei IP si a browser-ului utilizatorului

Exista situatii in care se doreste obtinerea unor informatii despre vizitatorii unui site. Astfel se poate afla adresa IP a vizitatorului, browser-ului utilizat de acesta sau sistemul de operare pe care acesta il utilizeaza. Toate acestea pot fi stocate in baze de date.

In aplicatia care urmeaza se va determina adresa IP si browser-ul utilizatorului.

\$_SERVER este o variabila predefinita de tip masiv asociat. Cu ajutorul acesteia , scriptul PHP poate accesa o multime de informatii. Pentru a afla adresa IP a vizitatorului se foloseste \$_SERVER["REMOTE_ADDR"] si pentru a afla informatii despre browser se foloseste \$_SERVER["HTTP_USER_AGENT"]. In acest din urma caz, daca sirul obtinut contine subsirul MSIE 11, atunci browser-ul utilizat este Microsoft Internet Explorer versiunea 11, daca contine subsirul Firefox, atunci browser-ul este de tip Firefox, etc.

Scriptul urmatoar ofera informatii legate de browser-ul utilizatorului si adresa IP a acestuia.

```
<?php
$brw=$_SERVER["HTTP_USER_AGENT"];
echo "Informatia generala: ".$brw."<br>";

if(strpos($brw,"MSIE"))
    $brw="Microsoft Internet Explorer";
else
    if(strpos($brw,"MSIE"))
        $brw="Microsoft Internet Explorer";
    else
        if(strpos($brw,"MSIE"))
            $brw="Microsoft Internet Explorer";
        else
            if(strpos($brw,"Netscape"))
                $brw="Netscape";
            else
                if(strpos($brw,"Firefox"))
                    $brw="Firefox";
            else
```

```

        if(strstr($brw,"Mozilla"))
            $brw="Mozilla";
        else
            if(strstr($brw,"Opera"))
                $brw="Opera";
            else
                $brw="Browser necunoscut";
echo " Browser ".$brw."<br>";

$ip=$_SERVER["REMOTE_ADDR"];
echo " Adresa IP ".$ip;
?>

```

2. Cum se creeaza un forum?

Pentru a crea un forum vom folosi:

1. O baza de date – pe care o vom denumi forum si un table pe care il com denumi mesaje, cate un mesaj pe un rand. Fiecare mesaj va contine: nr. de ordine, ID-ul celui care a trimis mesajul, titlul mesajului si textul acestuia. Scriptul urmatior numit f1.php creeaza baza de date si tabelul.

```

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

//creare baza

$resultat=mysql_query("create database forum;");
if(!$resultat){
    echo "Nu s-a reusit crearea bazei de date";
    exit;
}

//creare tabel
$baza=mysql_select_db("forum");
$interogare="create table mesaje(nr int unique key auto_increment, id
char(20),titlu char(100), textul longtext);";
$resultat=mysql_query($interogare);
if(!$resultat)
    echo mysql_errno().": ".mysql_error();
?>

```

2. In continuare avem nevoie de un scrip prin care utilizatorul scrie un mesaj si la apasarea butonului se trimite catre server mesajul. Se vor introduce datele: ID, Titlu, Mesajul. Fisierul f2.php va contine acest script. Acesta este format din etichete HTML.

```

<html>
<head>
</head>
<body>

```

```

<form action="insereaza.php" method="post">
ID<br>
<input name="id" type="text">
<br>Titlu<br>
<input name="titlu" type="text">
<br><br>Mesajul dv<br>
<textarea name="mesajul" rows="10" cols="30"></textarea>
<br>
<input type="submit" value="Adauga mesajul">
</form>
</body>
</html>

```

- Pe server, informatiile transmise de catre cel care a scris mesajul sunt preluate de catre scriptul insereaza.php. Practic, se verifica daca fiecare camp in parte este nevid, dupa care mesajul este memorat in tabel.

```

<?php
//conectare la mysql
$mysql=mysql_connect("localhost","root","");
if(!$mysql){
    echo "Nu s-a realizat conectarea la MySQL";
    exit;
}
$baza=mysql_select_db("forum");
if(!$baza){
    echo mysql_errno()." : ".mysql_error();
    exit;
}
$id=$_POST['id'];
if(strcmp($id,"")==0){
    echo "Nu ati introdus ID-ul";
    require_once("f2.php");
    exit;
}
$titlu=$_POST['titlu'];
if(strcmp($titlu,"")==0){
    echo "Nu ati introdus titlul";
    require_once("f2.php");
    exit;
}
$mesajul=$_POST['mesajul'];
if(strcmp($mesajul,"")==0){
    echo "Nu ati introdus mesajul";
    require_once("f2.php");
    exit;
}
$cerere="insert into mesaje (id,titlu,textul)
values('$id','$titlu','$mesajul')";
$rezultat=mysql_query($cerere);
if(!$rezultat)

```

```

    echo mysql_errno().":".mysql_error();
else
    echo "Mesaj inregistrat cu succes!";
?>

```

4. O persoana care doreste sa vizualizeze mesajele scrise va apela scriptul f3.php. Acesta afiseaza tabelul mesajelor cu campurile nr.curent/ID/Titlu mesaj. Pentru a-l vizualiza trebuie sa se execute clic pe titlul mesajului. Atunci cand se executa clic se va apela un alt script f4.php, care va afisa mesajul cerut. Problema este ca va trebui sa se cunosca numarul current al mesajului respectiv. Prin urmare este necesar un link care sa transmita si un parametru. Este necesar un apel de forma: f4.php?nr_mesaj. De exemplu, daca numarul mesajului este 12, vom scrie f4.php?12. In rest, script-ul f3 afiseaza datele in tr-un table.

f3.php

```

<?php
//conectare la mysql
$mysql=mysql_connect("localhost","root","");
if(!$mysql){
    echo "Nu s-a realizat conectarea la MySQL";
    exit;
}
$baza=mysql_select_db("forum");
if(!$baza){
    echo mysql_errno().":".mysql_error();
    exit;
}
$cerere="select * from mesaje";
$rezultat=mysql_query($cerere);
if(!$rezultat){
    echo mysql_errno().":".mysql_error();
    exit;
}

echo "<table border=5>";
//cap de tabel
echo "<tr> <td align='left'><strong>nr.curent<td
align='center'><strong>ID<td><strong>Titlu mesaj</tr>";
while($rand=mysql_fetch_array($rezultat)){
    echo "<tr>";
    echo "<td
align='right'>".$rand["nr"]."<td>".$rand["id"]."<td>".$rand["titlu"]."</a>";
    echo "</tr>";
}
echo "</table>";
?>

```

5. Scriptul f4.php, apelat de f3.php primeste ca parametru de intrare numarul mesajului si practice apeleaza MySQL-ul, deschide baza de date si afiseaza continutul mesajului, identificandu-l cu ajutorul unei interogari.

f4.php

```
<?php
//conectare la mysql
$nr=$_SERVER["QUERY_STRING"];
$sql=mysql_connect("localhost","root","");
if(!$sql){
    echo "Nu s-a realizat conectarea la MySQL";
    exit;
}
$baza=mysql_select_db("forum");
if(!$baza){
    echo mysql_errno()." : ".mysql_error();
    exit;
}
$cerere="select textul from mesaje where nr=$nr";
$resultat=mysql_query($cerere);
$rand=mysql_fetch_array($resultat);
echo $rand["textul"];
?>
```

Probleme de rezolvat

1. Adaugati script-ului f3.php o secventa in urma careia, la apasarea unui buton, sa se poata introduce un mesaj (practic, sa fie apelat scriptul f2.php).
2. Reproiectati aplicatia astfel incat sa existe posibilitatea unui raspuns la mesaj.