**Funcții procedurale MySQLi**

[Extensia îmbunătățită MySQL](https://translate.google.com/translate?hl=ro&prev=_t&sl=en&tl=ro&u=http://www.phpknowhow.com/mysql/php-mysql-connections/%23mysql-improved-extension#mysql-improved-extension) ( MySQLi ) oferă o interfață procedurală, precum și o interfață orientată obiect. In continuare vom analiza unele dintre [funcții](https://translate.google.com/translate?hl=ro&prev=_t&sl=en&tl=ro&u=http://www.phpknowhow.com/basics/functions/%23built-in-functions" \l "built-in-functions)  procedurale comune MySQLi.

**mysqli\_ connect ()**

Această funcție este utilizată pentru conectarea la MySQL . Înainte de a efectua orice operațiune în baza de date, trebuie să ne conectăm la MySQL . La succes, această funcție returnează un identificator de legătură pe care îl putem utiliza în alte funcții MySQLi . În caz de eșec, va genera o eroare.

In următorul exemplu este prezentat modul în care un utilizator numit **root**cu parola **root123**trebuie să se conecteze la o bază de date numită **company\_db**la **localhost**. Utilizatorul **root**ar trebui să aibă privilegii de a accesa**company\_db**.

**$link = mysqli\_connect (' localhost ', 'root', 'root123', ' company\_db ');**

Dacă portul MySQL este diferit de cel implicit (3306), trebuie să dați numărul portului ca al cincilea parametru.

**$link = mysqli\_connect ('localhost', 'root', 'root123', 'company\_db', '3800');**

**eroare mysqli\_connect\_()**

**Dacă mysqli\_connect( )**oferă o eroare la eșec atunci funcția **mysqli\_connect\_error()**stochează eroarea în ultimul apel efectuat către **mysqli\_connect()**. Dacă nu există nicio eroare, aceasta returnează NULL.

Pentru a încerca o **eroare mysqli\_connect( )**, oprim serverul MySQL și apelăm la **mysqli\_connect()**. Dacă am [activat erorile PHP](https://translate.google.com/translate?hl=ro&prev=_t&sl=en&tl=ro&u=http://www.phpknowhow.com/configuration/php-ini-error-settings/) , vom vedea o eroare care include informații precum cele de mai jos. **mysqli\_connect\_error( )**va returna același mesaj.

**Can’t connect to MySQL server on ‘localhost’**

**Notă:** În practică, nu este bine să afișăm mesaje de eroare ca acestea utilizatorilor aplicației dvs. PHP (acestea pot conține date sensibile și pot părea prea tehnice).

Prin urmare, putem utiliza [operatorul de suspendare a erorilor](https://translate.google.com/translate?hl=ro&prev=_t&sl=en&tl=ro&u=http://www.phpknowhow.com/basics/operators/%23error-suppression-operator" \l "error-suppression-operator) în fața **mysqli\_connect( )**pentru a opri aruncarea erorilor și utilizați **mysqli\_connect\_error()**pentru a înregistra eroarea pentru depanarea ca mai jos.

**<? php**

**$link = @ mysqli\_ connect ( ' localhost ', 'root', 'root123', ' company\_db ');**

**if ( mysqli\_connect\_error ())**

**{**

**$logMessage = ' Eroare MySQL : ' . mysqli\_connect\_ error ( );**

**// Apelați loggerul aici.**

**die ( „Nu s-a putut conecta la baza de date”);**

**}**

**// Restul codului urmează aici**

**?>**

**Notă**

**Operatorul @ suprimă mesajele de eroare create de următorul cod. În acest caz special, o eroare de conectare nu va duce la o eroare înregistrată (sau afișată), dar cel mai probabil va fi prinsă mai departe**

**mysqli\_select\_ db ()**

Pentru a schimba baza de date utilizată, putem utiliza funcția **mysqli\_select\_ db ( )**. Pentru exemplu, să admitem că user **root**are și privilegii si pentru o altă bază de date numită **company\_new\_db,**atunci putem schimba baza de date după cum urmează.

**<?php**

**$ link = @ mysqli\_connect (' localhost ', 'robin', 'robin123', ' company\_db ');**

**/ / Operațiuni pe „ company\_db ”**

**MySQLi \_select\_db ($ link, ' company\_new\_db ');**

**/ / Operațiuni pe „ company\_new\_db ”**

**?>**

Vom avea nevoie de această funcție numai dacă aplicația noastră PHP lucrează odată cu mai multe baze de date.

**mysqli\_ close ()**

Putem utiliza această funcție pentru a închide o conexiune MySQL . Ea reintoarce TRUE dacă actiunea este cu succes și FALS invers.

**<?php**

**$ link = @ mysqli\_connect (' localhost ', 'root', 'root123', ' company\_db ');**

**/ / Operațiile MySQL merg aici.**

**MySQLi \_close ($ link);**

**?>**

PHP va închide conexiunile deschise și va elibera resursele la sfârșitul scriptului PHP. Dar, practică bună recomandă să utilizăm în mod explicit  **mysqli\_close( )**la sfârșitul operațiunilor MySQL pentru a elibera resurse imediat.

**interogare mysqli\_ ()**

Această funcție este utilizată pentru executarea interogărilor MySQL . Ea reintoarce FALS în caz de esec. Pentru interogările SELECT, SHOW, DESCRIBE și EXPLAIN (dacă rezultatul este o relatie la ieșire), returnează un set de rezultate MySQL (resursă) care poate fi utilizat în funcții precum **mysqli\_fetch\_array( )**.

Pentru toate celelalte interogări precum INSERT, **UPDATE și DELETE**, acesta returnează TRUE în caz de succes.

**<?php**

**$Link = @ mysqli\_connect (' localhost ', 'root', 'root123', ' company\_db ');**

**$query = "SELECT \* FROM employee ";**

**if ( mysqli\_query ($ link, $ query))**

**{**

**/ / Iterează și afișează rezultatul**

**}**

**else**

**{**

**/ / Afișare eroare**

**}**

**mysqli \_close ($ link);**

**?>**

**mysqli\_fetch\_ array ()**

Această funcție este utilizată pentru citirea datelor dintr-un set de rezultate MySQL (returnat de o **interogare mysqli\_query( )**). Ea citește și returnează un rând de date ca o [matrice](https://translate.google.com/translate?hl=ro&prev=_t&sl=en&tl=ro&u=http://www.phpknowhow.com/basics/arrays/)/array și apoi mută indicatorul pe rândul următor. Când nu mai sunt rânduri de returnat, acesta returnează NULL. Datorită acestui comportament, este adesea folosit cu un [While Loop](http://www.phpknowhow.com/basics/loops/#while-loop) dupa cum se arată in exemplul de mai jos.

**while ($row = mysqli\_fetch\_array ($ result))**

**{**

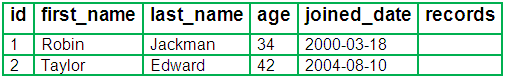
**/ \* Până când există date, $row va fi o matrice/array.**

**\* La sfârșit , $row devine NUL terminând bucla.**

**\* /**

**}**

Să presupunem că tabelul următor al **angajaților**este disponibil în baza noastră de date **company\_db**.



Mai jos este modul în care am prelua/fetch **id** -ul, **first\_name** și **last\_name** din acest tabel.

**<?php**

**$link = @mysqli\_connect('localhost', 'root', 'root123', 'company\_db');**

**$query = "SELECT `id`, `first\_name`, `last\_name` FROM `employee`";**

**$result = mysqli\_query($link, $query);**

**while ($row = mysqli\_fetch\_array($result))**

**{**

**echo $row[0] . ': ' . $row[1] . ' ' . $row[2];**

**echo '<br />';**

**}**

**mysqli\_free\_result($result);**

**mysqli\_close($link);**

**?>**

**Nota:** Funcția **mysqli\_free\_result ()** eliberează memoria asociată cu rezultatul.

Când [rulați](https://translate.google.com/translate?hl=ro&prev=_t&sl=en&tl=ro&u=http://www.phpknowhow.com/basics/running-php-files/) , codul de mai sus browserul web va afișa următorul conținut în  **1: Robin Jackman  
2: Taylor Edward**Aditional față de [matrice/array indexată](https://translate.google.com/translate?hl=ro&prev=_t&sl=en&tl=ro&u=http://www.phpknowhow.com/basics/arrays/%23indexed-vs-associative-arrays" \l "indexed-vs-associative-arrays) , **mysqli\_fetch\_ matrice ( )**returnează, de asemenea, o matrice asociată în care cheile/indecsii corespund numelui de coloane corespunzătoare ale tabelului. Deci, următorul segment de cod va produce aceeași ieșire.   
**while ($row = mysqli\_fetch\_array($result))**

**{**

**echo $row['id'] . ': ' . $row['first\_name'] . ' ' . $row['last\_name'];**

**echo '<br />';**

**}**

Putem limita ce matricea/array-ul s[ reintoarcem ca rezultat.

**// Returns only an Indexed array**

**mysqli\_fetch\_array($result, MYSQLI\_NUM);**

**// Returns only an Associated array**

**mysqli\_fetch\_array($result, MYSQLI\_ASSOC);**

PHP oferă două funcții care produc aceleași rezultate pe care le obțineți prin trecerea constantelor la **mysqli\_fetch\_ array( )**.

**// Same as mysqli\_fetch\_array($result, MYSQLI\_NUM)**

**mysqli\_fetch\_row($result);**

**// Same as mysqli\_fetch\_array($result, MYSQLI\_ASSOC)**

**mysqli\_fetch\_assoc($result);**

**mysqli\_free\_ result ()**

Imediat după ce ați utilizat un set de rezultate, puteți elibera memoria utilizată pentru acesta, după cum urmează.

**mysqli \_free\_result ($ result);**

**mysqli\_num\_ rows ()**

**mysqli\_num\_ rows( )**returnează numărul de rânduri dintr-un set de rezultate. Folosind-o, putem realize o acțiune diferită atunci când setul de rezultate este gol.

**if (mysqli\_num\_rows($result) > 0)**

**{**

**// Proceed with the $result**

**}**

**else**

**{**

**// Show an error message**

**}**

**mysqli\_affected\_ rows()**

Această funcție oferă informații despre ultima interogare MySQL executată. Pentru INSERT, UPDATE, REPLACE și DELETE, acesta oferă numărul de rânduri afectate. Pentru SELECT, returnează numărul de rânduri din setul de rezultate ca **mysqli\_num\_rows( )**.

Pentru exemplificare, urmează o interogare UPDATE pentru actualizarea numelui de familie al lui Taylor în tabelul **employee**. ***Cu condiția ca câmpul de identificare să fie unic***, știm că un singur rând ar trebui să fie afectat de această interogare.

**$query = "UPDATE `employee` SET `last\_name` = 'Adams' WHERE `id` = 2";**

**mysqli\_query($link, $query);**

**if (mysqli\_affected\_rows($link) == 1)**

**{**

**// Rest of the code**

**}**

**else**

**{**

**// Show an error message**

**}**

**eroare mysqli\_ ()**

Dacă a existat o eroare în ultima interogare MySQL , această funcție va returna eroarea. Dacă nu a existat nicio eroare, acesta va returna un șir gol.

**if (!mysqli\_query($link, $query))**

**{**

**$logMessage = 'MySQL Error: ' . mysqli\_error($link);**

**// Call your logger here.**

**die('There was an error in the query');**

**}**

**mysqli\_real\_escape\_ string ()**

Unele caractere precum apostroful (‘) unic au o semnificație specială în instrucțiunile SQL. De exemplu, ghilimelul unic este folosit pentru împachetarea șirurilor. Deci, dacă instrucțiunea dvs. SQL conține aceste caractere speciale, trebuie să le salvati prin utilizarea **mysqli\_real\_escape\_string( )**înainte de a trimite interogarea la **mysqli\_query()**.

Următorul apel către **mysqli\_query( )**returnează FALS, deoarece (‘) unic din O'Neil nu a fost escăpat.

**$query = "SELECT `id` FROM `employee` WHERE `last\_name` = 'O'Neil'";**

**mysqli\_query($link, $query);**

În urma apelului la **mysqli\_query()** s-ar returna un set de rezultate adecvat (cu condiția ca un angajat să existe cu numele O'Neil), deoarece numele este scăpat pentru prima dată prin **mysqli\_real\_escape\_string()**.

**$name = mysqli\_real\_escape\_string($link, "O'Neil");**

**$query = "SELECT `id` FROM `employee` WHERE `last\_name` = '$name'";**

**mysqli\_query($link, $query);**

Dacă instrucțiunile dvs. SQL sunt construite pe baza [[user inputs](https://translate.google.com/translate?hl=ro&prev=_t&sl=en&tl=ro&u=http://www.phpknowhow.com/get-and-post/understanding-get-and-post/)](http://www.phpknowhow.com/get-and-post/understanding-get-and-post/)[,](https://translate.google.com/translate?hl=ro&prev=_t&sl=en&tl=ro&u=http://www.phpknowhow.com/get-and-post/understanding-get-and-post/) cum ar fi cele de mai jos, este întotdeauna o idee bună să utilizați această funcție, deoarece intrarea utilizatorului poate conține caractere speciale.

**$lastName = mysqli\_real\_escape\_string($link, $\_POST['lastName']);**

**$query = "SELECT `id` FROM `employee` WHERE `last\_name` = '$lastName'";**

**mysqli\_query($link, $query);**

Un alt fapt este că intrările utilizatorilor pot conține diferite incercari cum ar fi mai jos.

**Exemple #1 Stilul procedural**

**<?php  
$link = mysqli\_connect("localhost", "my\_user", "my\_password", "world");  
  
/\* check connection \*/  
if (mysqli\_connect\_errno()) {  
    printf("Connect failed: %s\n", mysqli\_connect\_error());  
    exit();  
}  
  
$query = "SELECT Name, CountryCode FROM City ORDER by ID LIMIT 3";  
$result = mysqli\_query($link, $query);  
  
/\* numeric array \*/  
$row = mysqli\_fetch\_array($result, MYSQLI\_NUM);  
printf ("%s (%s)\n", $row[0], $row[1]);  
  
/\* associative array \*/  
$row = mysqli\_fetch\_array($result, MYSQLI\_ASSOC);  
printf ("%s (%s)\n", $row["Name"], $row["CountryCode"]);  
  
/\* associative and numeric array \*/  
$row = mysqli\_fetch\_array($result, MYSQLI\_BOTH);  
printf ("%s (%s)\n", $row[0], $row["CountryCode"]);  
  
/\* free result set \*/  
mysqli\_free\_result($result);  
  
/\* close connection \*/  
mysqli\_close($link);  
?>**

**Exemple #1 Stilul procedural**

**session\_start();**

**require\_once "connect\_to\_mysql.php";**

**// where i store username and password to access my db.**

**$sqlCommand = "SELECT \* property FROM portal";**

**// dbname: portal - table: propery**

**$query = mysqli\_query($myConnection, $sqlCommand);**

**$Displayproperty = '';**

**while ($row = mysqli\_fetch\_array($query))**

**$id = $row["pid"];**

**$title = $row["ptitle"];**

**$area = $row["parea"];**

**$city = $row["pcity"];**

**$Displayproperty .= '<table width="500" border="0" cellspacing="0" cellpadding="1">**

**<tr>**

**<td>' . $id . '</td>**

**<td>' . $title . '</td>**

**<td>' . $area . '</td>**

**<td>' . $city . '</td>**

**<td><a href="forms.php?pid=' . $id . '">Upload images</a><br /></td>**

**</tr>**

**</table>';**

**Exemple #3 Stilul procedural**

**<?php  
$link = mysqli\_connect("localhost", "my\_user", "my\_password", "world");  
  
/\* check connection \*/  
if (mysqli\_connect\_errno()) {  
    printf("Connect failed: %s\n", mysqli\_connect\_error());  
    exit();  
}  
  
$query = "SELECT Name, CountryCode FROM City ORDER by ID DESC LIMIT 50,5";  
  
if ($result = mysqli\_query($link, $query)) {  
  
    /\* fetch associative array \*/  
    while ($row = mysqli\_fetch\_row($result)) {  
        printf ("%s (%s)\n", $row[0], $row[1]);  
    }  
  
    /\* free result set \*/  
    mysqli\_free\_result($result);  
}  
  
/\* close connection \*/  
mysqli\_close($link);  
?>**

**?php**

**// Setting up connection with database Geeks**

**$connection = mysqli\_connect("localhost", "root", "",**

**"Geeks");**

**// Check connection**

**if (mysqli\_connect\_errno())**

**{**

**echo "Database connection failed.";**

**}**

**// query to fetch Username and Password from**

**// the table geek**

**$query = "SELECT Username, Password FROM geek";**

**// Execute the query and store the result set**

**$result = mysqli\_query($connection, $query);**

**if ($result)**

**{**

**// it return number of rows in the table.**

**$row = mysqli\_num\_rows($result);**

**printf("Number of row in the table : " . $row);**

**// close the result.**

**mysqli\_free\_result($result);**

**}**

**// Connection close**

**mysqli\_close($connection);**

**?>**