**UTILIZAREA PHP CU MYSQL pas cu pas:**

**1. PHP și MySQL: introducere**

**2. Cum să începem**

**3. Extensiile PHP MySQL: MySQLi și PDO**

**4. Cum să ne conectăm la MySQL cu PHP**

**5. Interogări SQL**

**6. Cum se creează tabele**

**7. Cum se citește, se inserează și se editează rânduri**

**8. Tratarea erorilor și excepții**

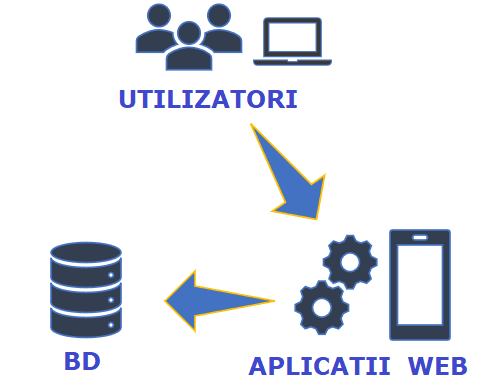
**9. Securitate SQL**

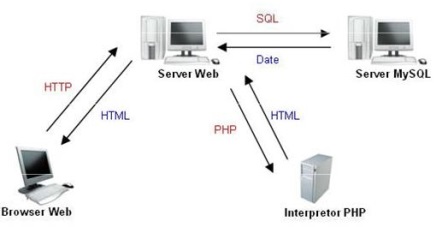
**10. Concluzie**

**1. PHP ȘI MYSQL: INTRODUCERE**

**Deci, ce MySQL? De ce aveți nevoie de el? Totul se rezumă la un fapt simplu:**

**Fiecare web site dinamic trebuie să stocheze unele date.**

****

****

**Un web site de exemplu de comerț electronic, e-commerce, trebuie să stocheze lista produselor și comenzile clienților săi, un blog trebuie să stocheze conținutul postărilor pe el, etc…**

**Limbajele de programare Back-End precum *PHP* și *Python* nu pot „stoca” nicio informație, cu excepția unor informații temporare mici, salvate în Sesiuni.**

**Astfel, aplicațiile web au nevoie de un spațiu de stocare în care să se păstreze toate aceste date și de unde să le citească atunci când este nevoie.**

**Pentru aceasta sunt folosite bazele de date precum MySQL.**

**Majoritatea aplicațiilor web utilizează BD relaționale precum MySQL, PostreSQL și Oracle.**

**Utilitar, o bază de date relațională este o colecție de tabele legate între ele. Datele sunt stocate în aceste tabele în rânduri/înscrieri.**

**MySQL este cea mai utilizată bază de date pentru site-uri web dinamice (bazele de date se mai numesc SGBD, sisteme de gestionare a bazelor de date).**

**În ciuda lipsei unor caracteristici avansate, MySQL a câștigat multă popularitate pentru licența open source și performanța ridicată pentru operațiuni simple.**

**Multe site-uri web populare precum WordPress blogurile și Wikipedia, ce se bazează pe MySQL (sau unele dintre ramificările sale, cum ar fi MariaDB).**

**ÎNTREBAREA INTREBĂRILOR ESTE: „CUM SE POAT FOLOSI O BAZĂ DE DATE MYSQL ÎN APLICAȚIA CONCRETĂ SCRISĂ ÎN PHP?”**

[**https://alexwebdevelop.com/how-to-learn-php/#php-in-practice**](https://alexwebdevelop.com/how-to-learn-php/#php-in-practice)



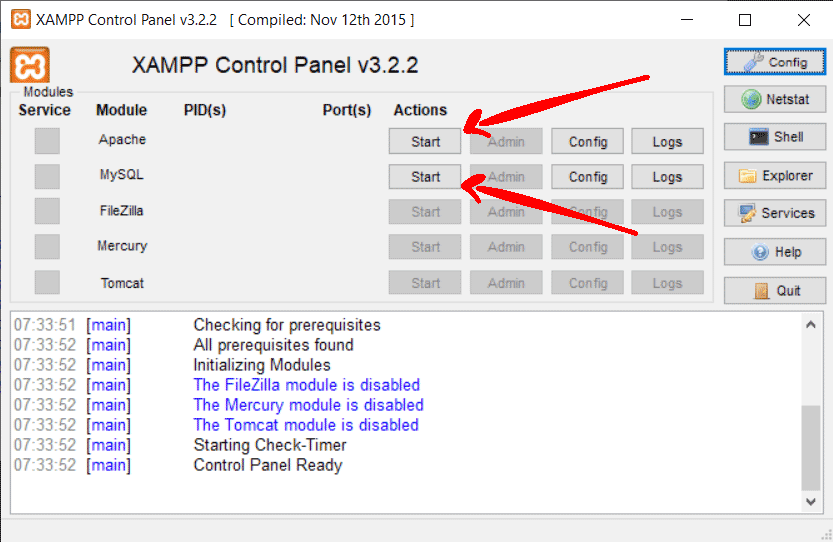
**CUM INCEPEM?**

**PASUL 1.**

**Simplu! Instalam, de exemplu,** XAMPP - unul dintre cele mai populare **medii de dezvoltare PHP! Acest pachet are integrat in el 2 servere locale – Apache (web-server) si serverul BD, MySql(MariaDB).** Pachetul XAMPP open source a fost creat pentru a fi incredibil de ușor de instalat și de utilizat.

**PASUL 2.**

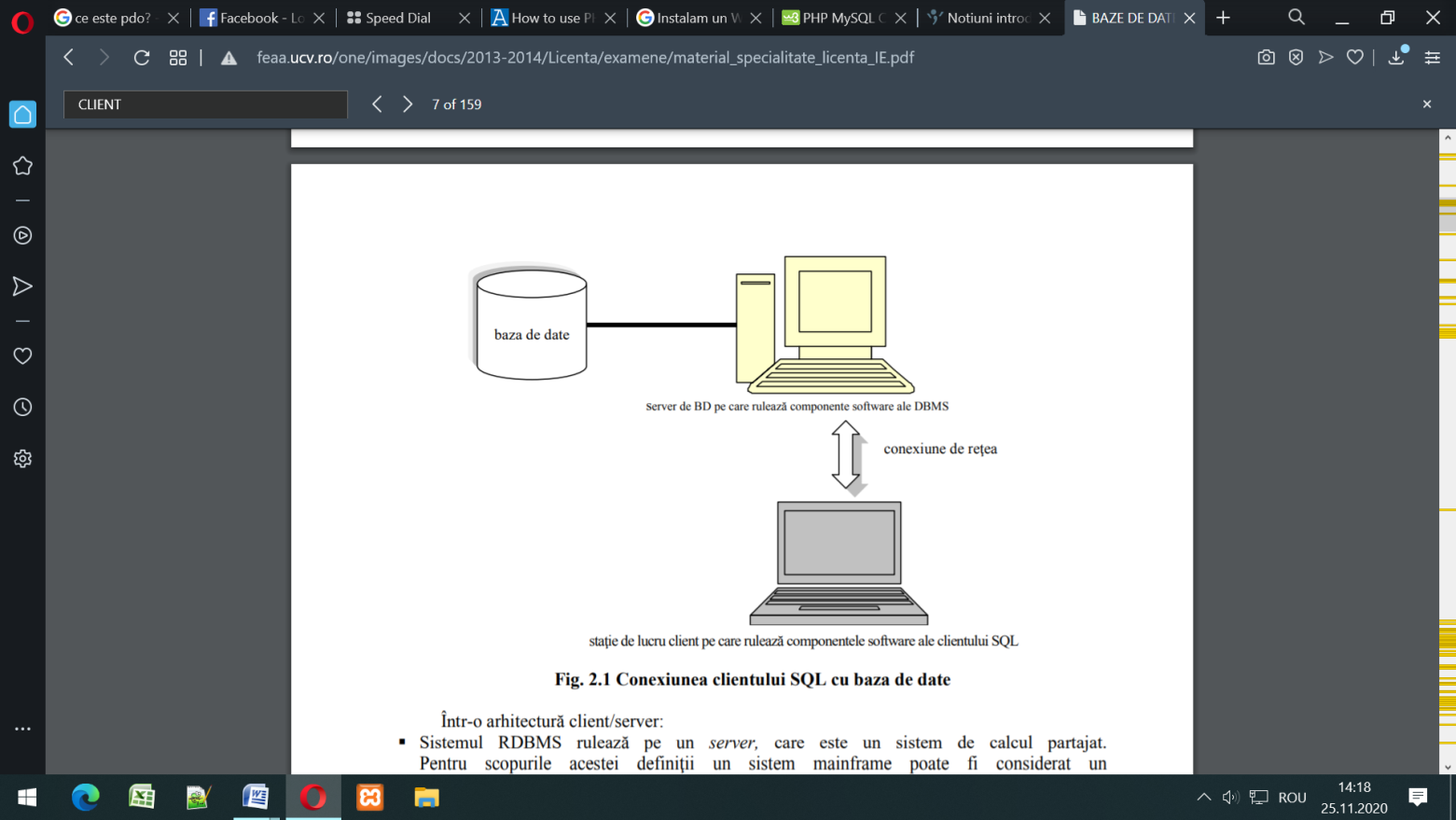
Cu ajutorul acestui mediu de lucru, putem studia PHP, MySQL separat si in interacțiune PHP cu MySQL. **Pentru a realiza acest deziderat este necesar sa lansăm panoul XAMPP, startînd cele 2 servere Apache si MySQL**



**PASUL 3.**

**Conectarea la o BD. Cum să conectăm în continuare BD pe serverul MySQL?**

**Pentru aceasta avem nevoie de un CLIENT SQL? Prezentăm schematic arhitectura client/server.**

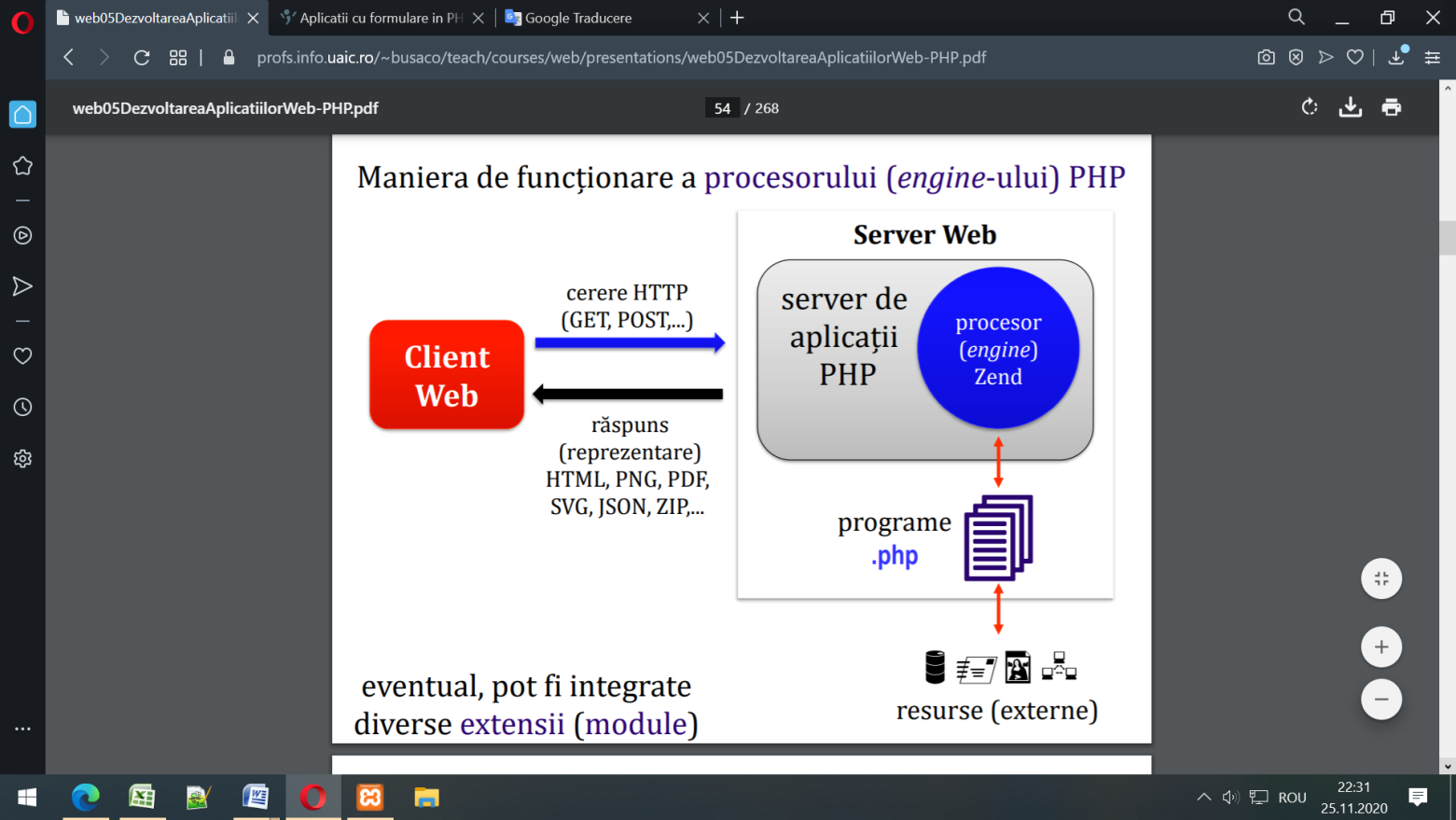
****

**Într-o arhitectură client/server:**

* SGBD rulează pe un **SERVER**, care este un sistem de calcul partajat/impartit.
* Fişierele care compun baza de date din punct de vedere fizic sunt stocate pe discuri conectate la serverul de baze de date.
* Utilizatorii care au acces la baza de date folosesc staţii de lucru, numite ***CLIENŢI***. Clientul trebuie să aibă o conexiune de reţea la baza de date, care poate fi o reţea privată, instalată acasă sau la birou, ori o reţea publica, precum Internet.
* Componentele software furnizate de producătorul SGBD rulează pe staţiile de lucru ale clienţilor pentru a oferi utilizatorilor posibilitatea să introducă instrucţiuni SQL, să le transmită SGBD în vederea prelucrării şi să vadă rezultatele returnate de SGBD. În general, acest software se numeşte ***CLIENT SQL***.

De reţinut că: nimic nu ne opreşte să instalăm clientul SQL pe acelaşi calculator cu sistemul SGBD. De fapt, mulţi dezvoltatori care utilizează sisteme SGBD precum MySQL, Microsoft SQL Server şi Oracle fac în mod obişnuit acest lucru, deoarece este foarte convenabil să aibă întregul mediu de dezvoltare pe un singur calculator, cum ar fi un laptop.

Totuşi, în momentul în care este necesar accesul partajat al mai multor utilizatori, este mult mai convenabil şi mai eficient să aveţi o singură copie a sistemului SGBD pe un server partajat şi să aveţi numai ***clientul SQL*** ***instalat pe staţia de lucru a fiecărui client***.



În funcţie de interfaţa cu utilizatorul de pe staţia de lucru client, clienţii SQL sunt clasificaţi în trei categorii:

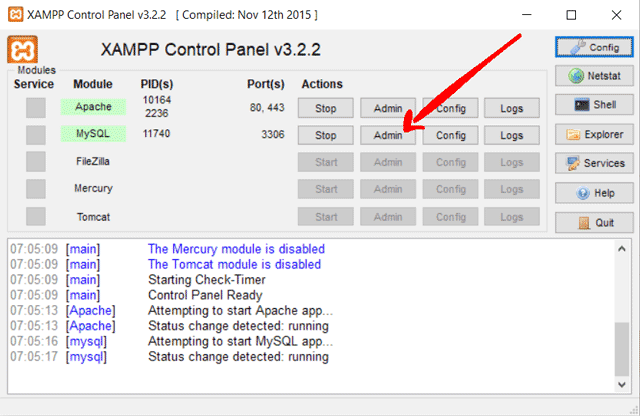
1. ***în linia de comandă(prompter/subsol):*** o interfaţă în linia de comandă se bazează exclusiv pe intrări şi ieşiri de tip text, cu comenzile introduse de la tastatură şi răspunsurile afişate ca mesaje de tip text. ***Principalul avantaj al interfeţelor în linia de comandă este că pot fi rulate pe APROAPE ORICE SISTEM DE OPERARE***.
2. ***grafici:*** o interfaţă grafică cu utilizatorul (GUI - graphical user interface) rulează sub un tip oarecare de sistem bazat pe ferestre, cum ar fi Mac OS sau Microsoft Windows, şi afişează datele sau opţiunile comenzilor folosind elemente grafice.
3. ***bazaţi pe web:*** **o interfaţă bazată pe web rulează pe serverul de baze de date, folosind un browser web de pe staţia de lucru client pentru a interacţiona cu utilizatorul bazei de date.**

În cele din urmă, vom utiliza ***aplicația web, PhpMyadmin***, una dintre cele mai populare aplicații. Este ușor de utilizat și îl putem rula din browser ca pe orice altă aplicație web.

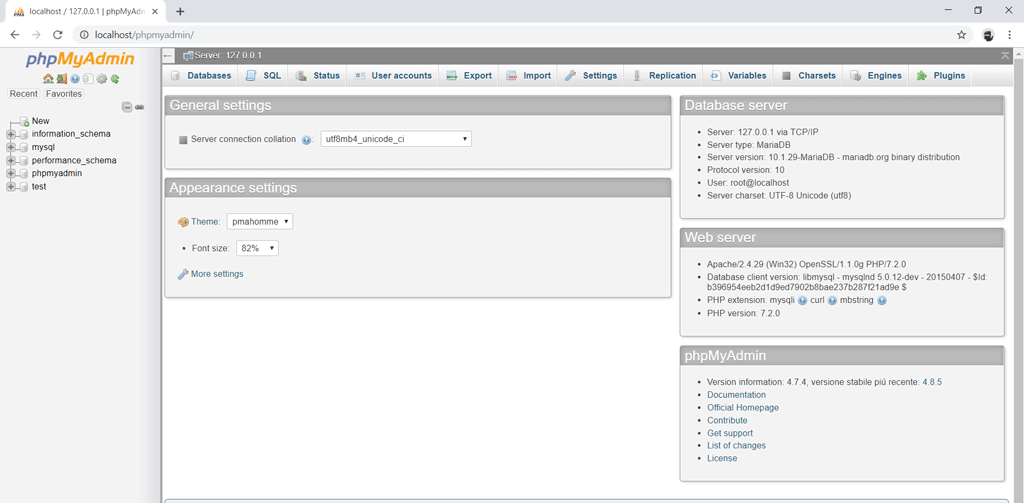
Toate mediile de dezvoltare PHP includ în mod implicit phpMyAdmin. Pentru a-l accesa, de obicei trebuie să deschideți următoarea adresă URL:

http: // localhost / phpmyadmin /

Dacă utilizăm XAMPP, putem ajunge acolo dând clic pe butonul MySQL Admin din panoul de control:

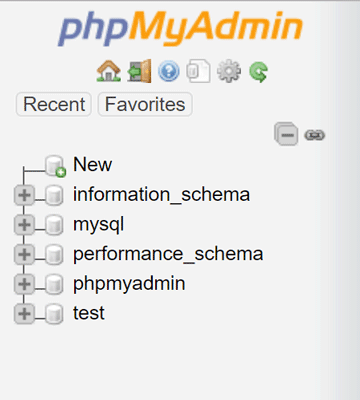


În funcție de configurația phpMyAdmin va permite să accesăm direct baza de date sau va cere **numele de utilizator și parola**. În acest caz urmează să verificăm documentația mediului de dezvoltare (puteți încerca să utilizați root / root, un cont implicit comun).



**CUM FUNCȚIONEAZĂ O BAZĂ DE DATE?**

MySQL gestionează **un set de baze de date sau *scheme*.** Noi am vorbit despre termenul schemă, deoarece baza de date poate fi ușor confundată cu întreaga instalare. Fiecare dintre aceste scheme conține unul sau mai multe tabele. Un tabel are una sau mai multe coloane (sau câmpuri). În cele din urmă, tabelele conțin date sub formă de rânduri/inscrieri. În partea stângă a interfeței phpMyAdmin puteți vedea lista schemelor disponibile la moment pe instalarea MySQL:



De obicei, ***fiecare aplicație web*** își folosește **propria schemă(BD)** pentru a stoca datele sale. De exemplu, phpMyAdmin însuși stochează propriile date în schema numită phpmyadmin (o puteți vedea în captura de ecran de mai sus).

Sa cream

1. Propria schema(BD) “**myDBSchema**”
2. Proiul cont cu numele de user “**myUserDB**” si parola “**myPassBD**”
3. Alocăm privilegii de lucru cu schema(BD) creată.

**Cum se realizează acești pași vedeti in fisierul atasat**

##### **Step 1: create your new schema**

1. Click on New on the left side, above the schema list
2. In the Database name field, insert the name of your database (for example, “**myDBSchema**”)
3. Click on the Create button

##### **Step 2: create your new account**

1. Click on the User accounts button in the upper section
2. Click on Add user account (below the list of already existing users)
3. In the User name row, make sure Use text field: is selected and input **myUserDB** in the text field
4. In the Host name row, select local from the drop-down menu
5. In the Password field, make sure Use text field: is selected and input **myPassBD**in the text field
6. Enter **myPassBD** again in the Re-type: field
7. Scroll down the page and click Go (bottom right)
8. A message with You have added a new user. will appear

##### **Step 3: set the account privileges**

1. Click on the User accounts button in the upper section
2. Find the row with myUser and click on Edit privileges
3. Click on Database in the upper section, next to Global (do not click on the Databases main tab)
4. Select mySchema from the list and click the Go button (bottom right)
5. Select the Check all checkbox to select all privileges
6. Click on the Go button (bottom right)
7. A message with You have updated the privileges for ‘myUser’@’localhost’. will appear

Sau aici

<https://dev.mysql.com/doc/refman/8.0/en/create-user.html>

**DOAR CU TITLUL DE INFORMARE: PHP, ÎNSEAMNĂ "PHP: HYPERTEXT PREPROCESSOR" A FOST LANSAT PRIN 1995**

**PHP 5**

**Pe 13 iulie 2004** a fost lansat PHP 5, cu Zend Engine II, ce a adus și o orientare obiect mai pronunțată și suportînd mai multe caracteristici ale acestui tip de programare. PHP 5 aduce mai multe noutăți față de versiunea 4:

* Suport îmbunătățit pentru OOP
* **INTRODUCE EXTENSIA PDO - PHP DATA OBJECTS**, care definește o modalitate facilă și consistentă de accesare a diferitelor baze de date
* Imbunătățiri de performanță

**La sfîrșitul lui 2007 doar versiunea 5.x** mai era întreținută, deoarece în data de 13 iulie 2007 (exact la 3 ani după lansarea PHP5), PHP Group a anunțat că **PHP4 va fi scos din uz pe 31 decembrie 2007** , deși prognozează că anumite upgrade-uri de securitate se vor oferi pînă pe 8 august.

**PHP 6**

**La versiunea PHP 6** s-a lucrat mai bine de 5 ani cu rezultate modeste și după tot acest efort s-a renunțat la această versiune în 2010. Acest efort considerabil nu a rămas fără urmări iar diferite caracteristici se regăsesc în versiunile PHP 5.3 și PHP 5.4. Versiunile PHP 5.3 și PHP 5.4 au venit cu diferite caracteristici foarte impresionante și din acest motiv mulți programatori considera că versiunea PHP 5.4 ar fi trebuit să devină versiunea PHP 6.0 a limbajului însă nu a fost cazul.

Din cauza că diferite caracteristici ce trebuiau implementate în versiunea PHP 6 au fost implementate în PHP 5.3 și PHP 5.4 și întreaga documentație cu tot ce urma să vină în PHP 6 a fost publica timp de 5 ani atunci s-a luat decizia să se renunțe la versiunea PHP 6 și să fie omisă pentru a nu exista nici-o confuzie în legătură cu noile caracteristici ale limbajului PHP.

**PHP 7**

**PHP 7 reprezintă singura lansare a acestui limbaj din ultimii 11 ani** și înainte de a fi lansat a stat o perioada de 2 ani în care a fost dezvoltat. PHP 7 vine cu o nouă versiune a motorului ZEND, ceea ce îl face să fie de doua ori mai rapid folosind mai puțină memorie. PHP 7 vine cu un suport considerabil pentru platformele ce rulează pe 64 de biți.

**PHP 7 a fost lansat în data de 3 Decembrie 2015.**

PHP 7 este de doua ori mai rapid decât orice versiune anterioară. PHP 7 convertește mult mai rapid liniile noastre de cod în format binar și folosește mult mai putină memorie pentru acest proces. Pentru a înțelege mai bine uitați-vă la următoarele date ce reprezintă cererile pe care PHP 5.6 și PHP 7.0 le poate procesa într-o secundă pe diferite platforme.

**PHP MYSQLI INTRODUCERE**

Când dorim să ne connectăm la baza de date MySQL din PHP avem la dispoziție trei modalități: folosind extensiile mysql sau mysqli sau clasa PDO ***(PHP Database Object - bază-de-date obiect orientata în PHP )***.

**1. Extensia**mysql

Este cea mai veche dintre cele trei și este în curs de a fi scoasă din uz în PHP 5.5.0,(pe XAMPP avem PHP versiunea 7.4.9) pe lângă aceasta, ultima ei revizuire a fost făcută prin 2002 și deci ultimele funcționalități din MySQL nu pot fi accesate prin intermediul ei. Nu este recomandat ca cei care învață PHP acum să o studieze - poate numai din motive istorice și/sau dacă urmează să se ocupe cu cod PHP învechit.

**2. Extensia**mysqli

mysqli adică **MySQLImproved** - este varianta îmbunătățită a extensiei mysql. Are o IPA (API) atât **procedurală cât și orientată pe obiecte**. ***Conține 100 de funcții*** (față de 48, cât conține extensia mysql). Era metoda recomandată de conectare la baza de date până la apariția extensiei PDO.

**3. Extensia/clasa**PDO

Este cea mai nouă - și cea recomandată la momentul actual - modalitate de conectare la baza de date. Un mare avantaj față de mysqli este faptul că suportă ***12 drivere de baze* de date** (nu numai pentru MySQL, cum suportă mysqli). Aceasta înseamnă că dacă la un moment dat trebuie să schimbi baza de date (să zicem că vrem/trebuie să folosim PostgreSQL) modificările pe care trebuie să le aducem codului sunt minime, pe când dacă ai folosit mysqli trebuie să modificăm întreg codul care are legătură cu baza de date.

<http://www.w3bai.com/ro/php/php_ref_mysqli.html>

**PHP MySQLi = PHP MySQL îmbunătățită!**

**Funcțiile mysqli** vă permite să acceseze serverele de baze de date MySQL.

**Notă: Extensia MySQLi este proiectată pentru a lucra cu MySQL versiunea 4.1.13 sau mai nou.**

<http://www.w3bai.com/ro/php/php_superglobals.html>

**Variabile globale!!**

**Instalare / Configurația la rulare**

Pentru funcțiile mysqli să fie disponibile, trebuie să compilați PHP cu suport pentru extensia MySQLi.

Extensia MySQLi a fost introdus cu PHP versiunea 5.0.0. MySQL Driverul nativ a fost inclus în versiunea PHP 5.3.0.

Pentru detalii despre instalare, du - te la: [http://php.net/manual/en/mysqli.installation.php](https://php.net/manual/en/mysqli.installation.php)

Pentru detalii de configurare în timpul rulării, du - te la: [http://php.net/manual/en/mysqli.configuration.php](https://php.net/manual/en/mysqli.configuration.php)

**PHP Funcții 5 mysqli**

|  |  |
| --- | --- |
| **Funcţie** | **Descriere** |
| [mysqli\_affected\_rows()](http://www.w3bai.com/ro/php/func_mysqli_affected_rows.html) | Returnează numărul de rânduri afectate în operațiunea MySQL precedent |
| [mysqli\_autocommit()](http://www.w3bai.com/ro/php/func_mysqli_autocommit.html) | Activează sau dezactivează modificările bazei de date auto-comiterea |
| [mysqli\_change\_user()](http://www.w3bai.com/ro/php/func_mysqli_change_user.html) | Modifică utilizatorul conexiunii bazei de date specificată |
| [mysqli\_character\_set\_name()](http://www.w3bai.com/ro/php/func_mysqli_character_set_name.html) | Returnează caracterul setat implicit pentru conectarea bazei de date |
| [mysqli\_close()](http://www.w3bai.com/ro/php/func_mysqli_close.html) | Închide o conexiune bază de date deschisă anterior |
| [mysqli\_commit()](http://www.w3bai.com/ro/php/func_mysqli_commit.html) | Comite tranzacția curentă |
| [mysqli\_connect\_errno()](http://www.w3bai.com/ro/php/func_mysqli_connect_errno.html) | Returnează codul de eroare de la ultima eroare de conexiune |
| [mysqli\_connect\_error()](http://www.w3bai.com/ro/php/func_mysqli_connect_error.html) | Returnează descrierea erorii din ultima eroare de conexiune |
| [mysqli\_connect()](http://www.w3bai.com/ro/php/func_mysqli_connect.html) | Deschide o nouă conexiune la serverul MySQL |
| [mysqli\_data\_seek()](http://www.w3bai.com/ro/php/func_mysqli_data_seek.html) | Ajustează indicatorul rezultat la un rând arbitrar în setul rezultat |
| [mysqli\_debug()](http://www.w3bai.com/ro/php/func_mysqli_debug.html) | Efectuează operații de depanare |
| [mysqli\_dump\_debug\_info()](http://www.w3bai.com/ro/php/func_mysqli_dump_debug_info.html) | Dumps multe informații de depanare în buștean |
| [mysqli\_errno()](http://www.w3bai.com/ro/php/func_mysqli_errno.html) | Returnează ultimul cod de eroare pentru cel mai recent apel de funcție |
| [mysqli\_error\_list()](http://www.w3bai.com/ro/php/func_mysqli_error_list.html) | Returnează o listă de erori pentru cel mai recent apel de funcție |
| [mysqli\_error()](http://www.w3bai.com/ro/php/func_mysqli_error.html) | Returnează ultima descriere de eroare pentru cel mai recent apel de funcție |
| [mysqli\_fetch\_all()](http://www.w3bai.com/ro/php/func_mysqli_fetch_all.html) | Fetches toate rândurile de rezultate ca o matrice asociativă, o matrice numerică sau ambele |
| [mysqli\_fetch\_array()](http://www.w3bai.com/ro/php/func_mysqli_fetch_array.html) | Fetches un rând rezultat ca asociativă, o matrice numerică sau ambele |
| [mysqli\_fetch\_assoc()](http://www.w3bai.com/ro/php/func_mysqli_fetch_assoc.html) | Fetches un rând rezultat ca o matrice asociativă |
| [mysqli\_fetch\_field\_direct()](http://www.w3bai.com/ro/php/func_mysqli_fetch_field_direct.html) | Returnează meta-date pentru un singur câmp din setul de rezultate, ca un obiect |
| [mysqli\_fetch\_field()](http://www.w3bai.com/ro/php/func_mysqli_fetch_field.html) | Returnează următorul câmp din setul de rezultate, ca un obiect |
| [mysqli\_fetch\_fields()](http://www.w3bai.com/ro/php/func_mysqli_fetch_fields.html) | Returnează o matrice de obiecte care reprezintă câmpurile într-un set de rezultate |
| [mysqli\_fetch\_lengths()](http://www.w3bai.com/ro/php/func_mysqli_fetch_lengths.html) | Returnează lungimile coloanelor din rândul curent din setul de rezultate |
| [mysqli\_fetch\_object()](http://www.w3bai.com/ro/php/func_mysqli_fetch_object.html) | Returnează rândul curent al unui set de rezultate, ca un obiect |
| [mysqli\_fetch\_row()](http://www.w3bai.com/ro/php/func_mysqli_fetch_row.html) | Fetches un rând dintr-un rezultat-set și returnează ca o matrice recenzată |
| [mysqli\_field\_count()](http://www.w3bai.com/ro/php/func_mysqli_field_count.html) | Returnează numărul de coloane pentru cel mai recent interogare |
| [mysqli\_field\_seek()](http://www.w3bai.com/ro/php/func_mysqli_field_seek.html) | Setează cursorul câmp la câmpul dat de offset |
| [mysqli\_field\_tell()](http://www.w3bai.com/ro/php/func_mysqli_field_tell.html) | Returnează poziția cursorului câmpului |
| [mysqli\_free\_result()](http://www.w3bai.com/ro/php/func_mysqli_free_result.html) | Elibereaza memoria asociată cu un rezultat |
| [mysqli\_get\_charset()](http://www.w3bai.com/ro/php/func_mysqli_get_charset.html) | Returnează un obiect set de caractere |
| [mysqli\_get\_client\_info()](http://www.w3bai.com/ro/php/func_mysqli_get_client_info.html) | Returnează versiunea MySQL biblioteca client |
| [mysqli\_get\_client\_stats()](http://www.w3bai.com/ro/php/func_mysqli_get_client_stats.html) | statistici despre client pe returnări-proces |
| [mysqli\_get\_client\_version()](http://www.w3bai.com/ro/php/func_mysqli_get_client_version.html) | Returnează MySQL versiunea de bibliotecă client ca un întreg |
| [mysqli\_get\_connection\_stats()](http://www.w3bai.com/ro/php/func_mysqli_get_connection_stats.html) | statistici despre conexiunea returnează clientului |
| [mysqli\_get\_host\_info()](http://www.w3bai.com/ro/php/func_mysqli_get_host_info.html) | Returnează numele de gazdă al serverului MySQL și tipul de conexiune |
| [mysqli\_get\_proto\_info()](http://www.w3bai.com/ro/php/func_mysqli_get_proto_info.html) | Returnează versiunea de protocol MySQL |
| [mysqli\_get\_server\_info()](http://www.w3bai.com/ro/php/func_mysqli_get_server_info.html) | Returnează versiunea de server MySQL |
| [mysqli\_get\_server\_version()](http://www.w3bai.com/ro/php/func_mysqli_get_server_version.html) | Returnează versiunea de server MySQL ca un întreg |
| [mysqli\_info()](http://www.w3bai.com/ro/php/func_mysqli_info.html) | Returnează informații despre cel mai recent executat interogarea |
| [mysqli\_init()](http://www.w3bai.com/ro/php/func_mysqli_init.html) | Inițializează MySQLi și returnează o resursă pentru utilizarea cu mysqli\_real\_connect() |
| [mysqli\_insert\_id()](http://www.w3bai.com/ro/php/func_mysqli_insert_id.html) | Returnează ID-ul generat automat utilizat în ultima interogare |
| [mysqli\_kill()](http://www.w3bai.com/ro/php/func_mysqli_kill.html) | Cere serverului să omoare un fir MySQL |
| [mysqli\_more\_results()](http://www.w3bai.com/ro/php/func_mysqli_more_results.html) | Verifică dacă există mai multe rezultate de la un sistem multi interogare |
| [mysqli\_multi\_query()](http://www.w3bai.com/ro/php/func_mysqli_multi_query.html) | Îndeplinește una sau mai multe interogări, pe baza de date |
| [mysqli\_next\_result()](http://www.w3bai.com/ro/php/func_mysqli_next_result.html) | Pregătește rezultatul următorul set de mysqli\_multi\_query() |
| [mysqli\_num\_fields()](http://www.w3bai.com/ro/php/func_mysqli_num_fields.html) | Returnează numărul de câmpuri într-un set de rezultate |
| [mysqli\_num\_rows()](http://www.w3bai.com/ro/php/func_mysqli_num_rows.html) | Returnează numărul de rânduri dintr-un set de rezultate |
| [mysqli\_options()](http://www.w3bai.com/ro/php/func_mysqli_options.html) | Seturi de opțiuni suplimentare conecta și afectează comportamentul pentru o conexiune |
| [mysqli\_ping()](http://www.w3bai.com/ro/php/func_mysqli_ping.html) | Ping-uri o conexiune la server, sau încearcă să se reconecteze în cazul în care conexiunea a mers în jos |
| mysqli\_prepare() | Pregătește o declarație SQL pentru execuție |
| [mysqli\_query()](http://www.w3bai.com/ro/php/func_mysqli_query.html) | Efectuează o interogare împotriva bazei de date |
| [mysqli\_real\_connect()](http://www.w3bai.com/ro/php/func_mysqli_real_connect.html) | Deschide o nouă conexiune la serverul MySQL |
| [mysqli\_real\_escape\_string()](http://www.w3bai.com/ro/php/func_mysqli_real_escape_string.html) | Escapes caractere speciale într-un șir de caractere pentru a fi utilizat într-o instrucțiune SQL |
| mysqli\_real\_query() | Executa o interogare SQL |
| mysqli\_reap\_async\_query() | Returnează rezultatul din interogare asincronă |
| [mysqli\_refresh()](http://www.w3bai.com/ro/php/func_mysqli_refresh.html) | Reîmprospătează tabele sau cache-uri, sau resetează informațiile despre serverul de replicare |
| [mysqli\_rollback()](http://www.w3bai.com/ro/php/func_mysqli_rollback.html) | Rolls înapoi tranzacția curentă pentru baza de date |
| [mysqli\_select\_db()](http://www.w3bai.com/ro/php/func_mysqli_select_db.html) | Modifică baza de date implicită pentru conexiunea |
| [mysqli\_set\_charset()](http://www.w3bai.com/ro/php/func_mysqli_set_charset.html) | Setează implicit setul de caractere client |
| mysqli\_set\_local\_infile\_default() | Unsets utilizator handler definit pentru sarcina de comandă locală infile |
| mysqli\_set\_local\_infile\_handler() | Setați funcția de apel invers pentru LOAD comanda LOCAL INFILE |
| [mysqli\_sqlstate()](http://www.w3bai.com/ro/php/func_mysqli_sqlstate.html) | Returnează codul de eroare SQLSTATE pentru ultima operație MySQL |
| [mysqli\_ssl\_set()](http://www.w3bai.com/ro/php/func_mysqli_ssl_set.html) | Folosit pentru a stabili conexiuni securizate folosind SSL |
| [mysqli\_stat()](http://www.w3bai.com/ro/php/func_mysqli_stat.html) | Returnează starea sistemului actual |
| [mysqli\_stmt\_init()](http://www.w3bai.com/ro/php/func_mysqli_stmt_init.html) | Inițializează o declarație și returnează un obiect pentru a fi utilizat cu mysqli\_stmt\_prepare() |
| mysqli\_store\_result() | Transferă un set de rezultate din ultima interogare |
| [mysqli\_thread\_id()](http://www.w3bai.com/ro/php/func_mysqli_thread_id.html) | Returnează ID-ul firului pentru conexiunea curentă |
| [mysqli\_thread\_safe()](http://www.w3bai.com/ro/php/func_mysqli_thread_safe.html) | Returnează dacă biblioteca client este compilat ca firele de executie |
| mysqli\_use\_result() | Inițiază regăsirea unui set de rezultate din ultima interogare executată folosind mysqli\_real\_query() |
| mysqli\_warning\_count() | Returnează numărul de avertismente de la ultima interogare în conexiunea |

Extensia **mysqli** **contine atat interfata procedurala** (set de functii) **cat si interfata orientata obiect** (clase si metode) **pentru lucrul cu MySQL din PHP**.

**Deci aceeasi functionalitate este oferita de fiecare din modalitatile de lucru alese (cu functii sau cu clase si metode), alegerea este a programatorului.**  
In continuare vom prezenta principalele functii din extensia **mysqli**, ce sunt foarte asemanatoare (ca nume si functionalitate) cu cele din extensia **mysql**, cu urmatoarele diferente de retinut:

* numele functiilor incep cu **mysqli\_**
* **mysqli\_connect()** primeste un al patrulea argument - numele bazei de date selectate astfel incat sa nu mai fie nevoie apelarea functiei ***mysqli\_select\_db()***
* variabila ce pastreaza conexiunea la baza de date (returnata de functia **mysqli\_connect()**), denumita in general **$link**, va fi necesara ca prim argument la functii ce au nevoie de o conexiune activa, cum ar fi **mysqli\_query().** In functiile din extensia **mysql** acest argument era pe a 2-a pozitie pentru **mysql\_query()** si era optional.
* **exista doua functii suplimentare pentru detectarea erorilor,** mai precis pentru detectarea erorilor la conectare. Acestea sunt **mysqli\_connect\_error()** - returneaza mesajul erorii, si **mysqli\_connect\_errno().** Celelalte functii ce returneaza ultima eroare generata de serverul MySQL sunt similare, si anume: **mysqli\_error()** si **mysqli\_errno()**.
* functia **mysqli\_query()** primeste un al treilea parametru, constanta MYSQLI\_USE\_RESULT (implicit) sau MYSQLI\_STORE\_RESULT.

1. daca este folosit MYSQLI\_STORE\_RESULT setul de rezultate este extras integral in memoria calculatorului pe care ruleaza PHP dar permite un acces rapid la inregistrari
2. daca este folosit MYSQLI\_USE\_RESULT inregistrarile din setul de rezultate sunt extrase din serverul MySQL pe masura ce sunt accesate in PHP. Acest mod de lucru este potrivit pentru seturi mari de inregistrari, caz in care informatia nu trebuie stocata integral in memorie.

Codul de conectare la baza de date folosind functia **mysqli\_connect()** ar putea arata astfel: **conectare.php**

#### PHP

**<?php**

$db\_host = 'localhost';

$db\_user = 'root';

$db\_pass = '';

$db\_name = 'mysite';

$link = @mysqli\_connect($db\_host, $db\_user, $db\_pass, $db\_name);

if (!$link)

{

die('Eroare la conectare: ' . mysqli\_connect\_error());

}

**?>**

Functia **mysqli\_connect()** va returna **FALSE in cazul esuarii conexiunii**, si un **obiect de tip mysqli in caz de succes**, sau mai simplu, o variabila ce reprezinta conexiunea la serverul MySQL.  
  
Scriptul prin care realizam un select dintr-un tabel, devine: **select.php**

#### PHP

**<?php**

include("conectare.php");

$sql = 'SELECT \* FROM utilizatori';

$result = mysqli\_query($link, $sql);

while ($d = mysqli\_fetch\_assoc($result))

{

print\_r($d);

}

**?>**

#### Nota

Restul functiilor sunt si ele asemanatoare cu cele din extensia mysql:

**mysqli\_fetch\_row(),**

**mysqli\_fetch\_array(),**

**mysqli\_num\_rows(),**

**mysqli\_affected\_rows(), ....**

**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#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');**

**mysqli\_connect\_error()**

**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 cea 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#error-suppression-operator) în locul **mysqli\_connect( )**pentru a opri prezentarea lor și a utilize in acest caz funcția  **mysqli\_connect\_error(),**pentru a înregistra eroarea pentru depanarea/prelucrarea ei ca in cele ce urmează 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-ul **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.

**mysqli\_query()**

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))**

**{**

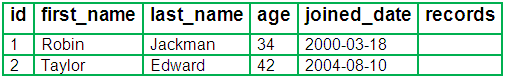
**/ \* Până când există date, $row va fi o matrice/array.**

**\* 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#indexed-vs-associative-arrays),  **mysqli\_fetch\_ array ( )**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);**

**Pe scurt:  
- mysqli\_fetch\_row() - Returneaza fiecare rand ca un Array in care cheile sunt numere consecutive (incepand de la 0), reprezentand ordinea coloanelor din setul de rezultate ($row[0], $row[1]).  
  
- mysqli\_fetch\_object() - Returneaza fiecare rand ca un obiect. Valorile coloanelor se acceseaza ca proprietati ale obiectului, prin numele lor ($row->col1, $row->col2).  
  
- mysqli\_fetch\_assoc() - Returneaza fiecare rand ca un Array in care cheile sunt numele coloanelor ($row['col1'], $row['col2']).  
  
- mysqli\_fetch\_array() poate avea unul din aceste 3 moduri:  
- fetch\_array(MYSQLI\_NUM) - Identic cu mysqli\_fetch\_row().  
- fetch\_array(MYSQLI\_ASSOC) - Identic cu mysqli\_fetch\_assoc().  
- fetch\_array(MYSQLI\_BOTH) - Produce un Array cu ambele tipuri de chei, atat numele coloanelor cat si numere consecutive ($row[0], $row[1], $row['col1'], $row['col2']).**

**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**

**}**

**mysqli\_error()**

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 utilizam 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);**

**?>**

**<?php**

**session\_start();**

**$link = mysqli\_connect("localhost", "root", "", "librarie");**

$sql = "SELECT \* FROM `carti`";

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

**$row = mysqli\_fetch\_array($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);  
?>**