1. Tehnologiile AJAX. Modificarea partială (incrementală) și asincronă a paginilor Web.

AJAX, Ajax (din engleză. Asynchronous Javascript și XML- „JavaScript și Asynchronous XML”) - o metodă de construire a interfețelor interactive de utilizator pentru aplicații web care implică schimbul de date „de fundal” între browser și serverul web. Prin urmare, pagina web nu poate fi reîncărcată complet atunci când datele sunt actualizate, iar aplicația web devine mai rapidă și mai ușor de utilizat. În rusă, se pronunță uneori ca „ajax” în transliterare. Acronimul AJAX nu are un alfabet chirilic definit.

În modelul clasic de aplicații web:

* Utilizatorul vizitează o pagină web și face clic pe un element al acesteia.
* [Browserul](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) formează și trimite o [cerere către](https://ru.wikipedia.org/wiki/HTTP)[server](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) .
* Ca răspuns, serverul generează o pagină web complet nouă și o trimite browserului etc., după care browserul reîncarcă complet întreaga pagină.

Când utilizați AJAX:

* Utilizatorul vizitează o pagină web și face clic pe un element al acesteia.
* [Scriptul](https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82) (în [JavaScript](https://ru.wikipedia.org/wiki/JavaScript) ) determină ce informații sunt necesare pentru reîmprospătarea paginii.
* [Browserul](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) trimite o cerere corespunzătoare către [server](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)) .
* Serverul returnează doar partea din document la care a venit solicitarea.
* [Scriptul](https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82) face modificări ținând cont de informațiile primite (fără a reîncărca complet pagina).

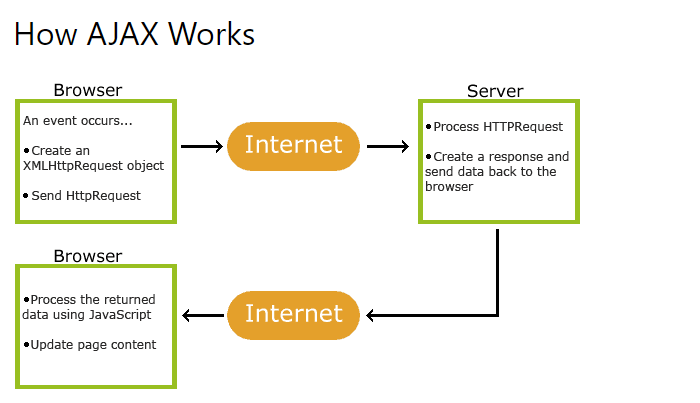
AJAX nu este o tehnologie independentă, ci un concept de utilizare a mai multor tehnologii conexe. AJAX se bazează pe două principii principale:

* utilizarea tehnologiei accesului dinamic la [server](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)) „din mers”, fără a reîncărca întreaga pagină complet, de exemplu, folosind [XMLHttpRequest](https://ru.wikipedia.org/wiki/XMLHttpRequest) (obiect principal);
* prin crearea dinamică a cadrelor copil ;
* prin crearea dinamică a etichetei <script> .
* prin crearea dinamică a etichetei <img>, așa cum este implementat în Google Analytics.
* utilizarea [DHTML](https://ru.wikipedia.org/wiki/DHTML) pentru a schimba dinamic conținutul paginii;

Acțiunile sunt convertite în operațiunea interfeței cu elemente [DOM](https://ru.wikipedia.org/wiki/Document_Object_Model) ( [engl.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Document Object Model* ), prin care variază datele procesate disponibile utilizatorului, prezentându-le astfel. De asemenea, gestionează mișcările și clicurile mouse-ului, precum și apăsările de tastă. Foi de stil în cascadă sau CSS ( [Eng.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Are Cascading the Style Sheets* ), oferă un aspect consistent și simplifică elementele de aplicație care fac apel la obiectele DOM. Obiectul XMLHttpRequest (sau mecanisme similare) este utilizat pentru interacțiune asincronă cu serverul, procesarea cererilor utilizatorilor și încărcarea datelor necesare în proces.

Trei dintre aceste patru tehnologii - [CSS](https://ru.wikipedia.org/wiki/CSS) , DOM și [JavaScript](https://ru.wikipedia.org/wiki/JavaScript)  - alcătuiesc [DHTML](https://ru.wikipedia.org/wiki/DHTML) ( [Eng.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *The Dynamic the HTML* ). Potrivit unor experți (cărți),  instrumentele DHTML, care au apărut în [1997](https://ru.wikipedia.org/wiki/1997_%D0%B3%D0%BE%D0%B4) , au arătat o mare promisiune, dar nu le-au îndeplinit.

Fragmente de text simplu, cod [HTML](https://ru.wikipedia.org/wiki/HTML) , [JSON](https://ru.wikipedia.org/wiki/JSON) sau [XML](https://ru.wikipedia.org/wiki/XML) pot fi utilizate ca [format de](https://ru.wikipedia.org/wiki/XML) transfer de date .



Ajax Response

function loadDoc() {  
  var xhttp = new XMLHttpRequest();  
  xhttp.onreadystatechange = function() {  
    if (this.readyState == 4 && this.status == 200) {  
      document.getElementById("demo").innerHTML =  
      this.responseText;  
    }  
  };  
  xhttp.open("GET", "ajax\_info.txt", true);  
  xhttp.send();  
}

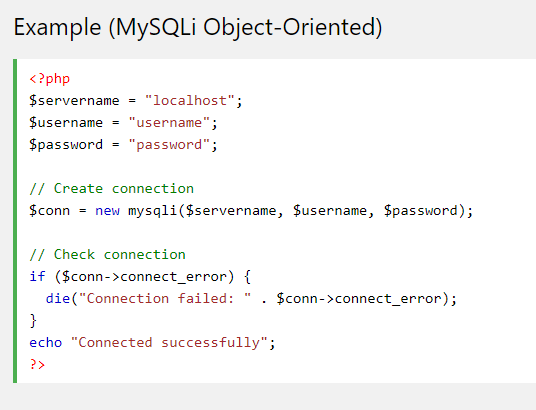
Ajax Database

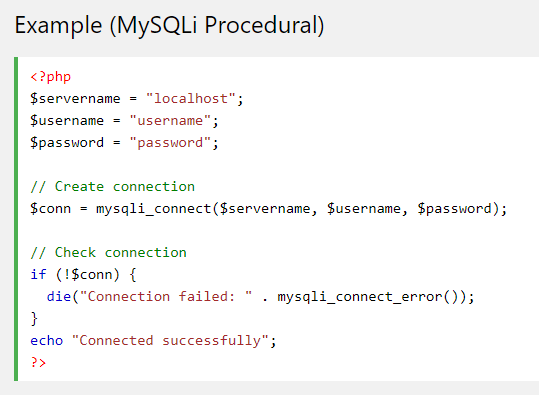
<?php  
$mysqli = new mysqli("servername", "username", "password", "dbname");  
if($mysqli->connect\_error) {  
  exit('Could not connect');  
}  
  
$sql = "SELECT customerid, companyname, contactname, address, city, postalcode, country  
FROM customers WHERE customerid = ?";  
  
$stmt = $mysqli->prepare($sql);  
$stmt->bind\_param("s", $\_GET['q']);  
$stmt->execute();  
$stmt->store\_result();  
$stmt->bind\_result($cid, $cname, $name, $adr, $city, $pcode, $country);  
$stmt->fetch();  
$stmt->close();  
echo "<table>";  
echo "<tr>";  
echo "<th>CustomerID</th>";  
echo "<td>" . $cid . "</td>";  
echo "<th>CompanyName</th>";  
echo "<td>" . $cname . "</td>";  
echo "<th>ContactName</th>";  
echo "<td>" . $name . "</td>";  
echo "<th>Address</th>";  
echo "<td>" . $adr . "</td>";  
echo "<th>City</th>";  
echo "<td>" . $city . "</td>";  
echo "<th>PostalCode</th>";  
echo "<td>" . $pcode . "</td>";  
echo "<th>Country</th>";  
echo "<td>" . $country . "</td>";  
echo "</tr>";  
echo "</table>";  
?>

2. Metode de accesare la baze de date. Interacţiunea cu HTML. Tipuri de date. Realizarea obiectelor dinamice. Interacţiunea cu HTML. Serverul My SQL.

**Metode de acces la baze de date cu PHP**

* MySQLi (object-oriented)
* MySQLi (procedural)
* PDO







**Dynamic HTML** , sau **DHTML** , este o colecție de tehnologii utilizate împreună pentru a crea [site-uri web](https://en.wikipedia.org/wiki/Website) interactive și animate utilizând o combinație a unui [limbaj de marcare](https://en.wikipedia.org/wiki/Markup_language) static (cum ar fi [HTML](https://en.wikipedia.org/wiki/HTML) ), un limbaj de [script pentru partea clientului](https://en.wikipedia.org/wiki/Client-side_scripting) (cum ar fi [JavaScript](https://en.wikipedia.org/wiki/JavaScript) ), un limbajul de definiție a prezentării (cum ar fi [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) ) și [Modelul de obiect document](https://en.wikipedia.org/wiki/Document_Object_Model) (DOM).  Aplicația DHTML a fost introdusă de [Microsoft](https://en.wikipedia.org/wiki/Microsoft) odată cu lansarea [Internet Explorer 4](https://en.wikipedia.org/wiki/Internet_Explorer_4) în 1997. Astăzi, trimiterile la [JavaScript discret](https://en.wikipedia.org/wiki/Unobtrusive_JavaScript) codarea (DOM Scripting) a înlocuit utilizarea termenului DHTML.

DHTML permite limbajelor de scriptare să schimbe [variabilele](https://en.wikipedia.org/wiki/Variable_(programming)) în limbajul de definiție al unei pagini web, care la rândul său afectează aspectul și funcția conținutului paginii HTML „statice” altfel, după ce pagina a fost încărcată complet și în timpul procesului de vizualizare. Astfel, caracteristica dinamică a DHTML este modul în care funcționează în timp ce o pagină este vizualizată, nu în capacitatea sa de a genera o pagină unică cu fiecare încărcare a paginii.

În schimb, o [pagină web dinamică](https://en.wikipedia.org/wiki/Dynamic_web_page) este un concept mai larg, care acoperă orice pagină web generată diferit pentru fiecare utilizator, apariția încărcării sau valori variabile specifice. Aceasta include paginile create de scripturi de pe partea de client și cele create de [scripturi](https://en.wikipedia.org/wiki/Server-side_scripting) de pe [server](https://en.wikipedia.org/wiki/Server-side_scripting) (cum ar fi [PHP](https://en.wikipedia.org/wiki/PHP) , [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) , [JSP](https://en.wikipedia.org/wiki/JavaServer_Pages) sau [ASP.NET](https://en.wikipedia.org/wiki/ASP.NET) ) în care serverul web generează conținut înainte de a-l trimite clientului.

Exemplu de adăugare a unui bloc de text

<!DOCTYPE html>

<**html**>

<**head**>

<**meta** charset="utf-8">

<**title**>Using a DOM function</**title**>

<**style**>

**a** { **background-color**: #eee; }

**a**:hover { **background**: #ff0; }

#**toggleMe** { **background**: #cfc; **display**: **none**; **margin**: 30px 0; **padding**: 1em; }

</**style**>

</**head**>

<**body**>

<**h1**>Using a DOM function</**h1**>

<**h2**><**a** id="showhide" href="#">Show paragraph</**a**></**h2**>

<**p** id="toggleMe">This is the paragraph that is only displayed on request.</**p**>

<**p**>The general flow of the document continues.</**p**>

<**script**>

**function** changeDisplayState(id) {

**var** d = document.getElementById('showhide'),

e = document.getElementById(id);

**if** (e.style.display === 'none' || e.style.display === '') {

e.style.display = 'block';

d.innerHTML = 'Hide paragraph';

} **else** {

e.style.display = 'none';

d.innerHTML = 'Show paragraph';

}

}

document.getElementById('showhide').addEventListener('click', **function** (e) {

e.preventDefault();

changeDisplayState('toggleMe');

});

</**script**>

</**body**>

</**html**>

Definirea corectă a câmpurilor dintr-un tabel este importantă pentru optimizarea generală a bazei de date. Ar trebui să utilizați doar tipul și dimensiunea câmpului pe care trebuie să îl utilizați cu adevărat. De exemplu, nu definiți un câmp lat de 10 caractere, dacă știți că veți folosi doar 2 caractere. Acest tip de câmpuri (sau coloane) sunt denumite și tipuri de date, după **tipul de date pe care** le veți stoca în acele câmpuri.

MySQL folosește mai multe tipuri de date diferite împărțite în trei categorii -

* Numeric
* Data si ora
* Tipuri de șiruri.

Să le discutăm acum în detaliu.

## **Tipuri de date numerice**

MySQL folosește toate tipurile de date numerice standard ANSI SQL, așa că, dacă veniți la MySQL dintr-un alt sistem de baze de date, aceste definiții vă vor părea familiare.

Următoarea listă prezintă tipurile de date numerice obișnuite și descrierile acestora -

* **INT** - Un număr întreg de dimensiuni normale care poate fi semnat sau nesemnat. Dacă este semnat, intervalul admisibil este cuprins între -2147483648 și 2147483647. Dacă nu este semnat, intervalul admis este de la 0 la 4294967295. Puteți specifica o lățime de până la 11 cifre.
* **TINYINT** - Un număr întreg foarte mic care poate fi semnat sau nesemnat. Dacă este semnat, intervalul admis este de la -128 la 127. Dacă nu este semnat, intervalul admis este de la 0 la 255. Puteți specifica o lățime de până la 4 cifre.
* **SMALLINT** - Un număr întreg mic care poate fi semnat sau nesemnat. Dacă este semnat, intervalul permis este de la -32768 la 32767. Dacă nu este semnat, intervalul permis este de la 0 la 65535. Puteți specifica o lățime de până la 5 cifre.
* **MEDIUMINT** - Un număr întreg de dimensiuni medii care poate fi semnat sau nesemnat. Dacă este semnat, intervalul permis este de la -8388608 la 8388607. Dacă nu este semnat, intervalul permis este de la 0 la 16777215. Puteți specifica o lățime de până la 9 cifre.
* **BIGINT** - Un număr întreg mare care poate fi semnat sau nesemnat. Dacă este semnat, intervalul permis este de la -9223372036854775808 la 9223372036854775807. Dacă nu este semnat, intervalul admisibil este de la 0 la 18446744073709551615. Puteți specifica o lățime de până la 20 de cifre.
* **FLOAT (M, D)** - Un număr în virgulă mobilă care nu poate fi nesemnat. Puteți defini lungimea afișajului (M) și numărul de zecimale (D). Acest lucru nu este necesar și va fi implicit la 10,2, unde 2 este numărul de zecimale și 10 este numărul total de cifre (inclusiv zecimale). Precizia zecimală poate merge în 24 de locuri pentru un FLOAT.
* **DOUBLE (M, D)** - Un număr cu virgulă mobilă de dublă precizie care nu poate fi nesemnat. Puteți defini lungimea afișajului (M) și numărul de zecimale (D). Acest lucru nu este obligatoriu și va fi implicit la 16,4, unde 4 este numărul de zecimale. Precizia zecimală poate merge la 53 de locuri pentru un DOUBLE. REAL este un sinonim pentru DUBLU.
* **DECIMAL (M, D)** - Un număr în virgulă mobilă despachetat care nu poate fi nesemnat. În zecimale neambalate, fiecare zecimală corespunde unui octet. Este necesară definirea lungimii afișajului (M) și a numărului de zecimale (D). NUMERIC este un sinonim pentru DECIMAL.

## **Tipuri de dată și oră**

Datele și ora MySQL sunt următoarele:

* **DATE** - O dată în format AAAA-LL-ZZ, între 01-01-01 și 9999-12-31. De exemplu, 30 decembrie lea , 1973 va fi stocat ca 1973-12-30.
* **DATETIME** - O combinație de dată și oră în AAAA-LL-ZZ HH: MM: format SS, între 1000-01-01 00:00:00 și 9999-12-31 23:59:59. De exemplu, 3:30 după - amiaza pe 30 decembrie lea , 1973 ar fi stocate ca 1973-12-30 15:30:00.
* **TIMESTAMP** - un marcaj de timp între miezul nopții, 01 ianuarie st , 1970 și cândva în 2037. Acest lucru arata ca formatul DATETIME anterior, numai fără cratime între numere; 3:30 după - amiaza pe 30 decembrie lea , 1973 ar fi stocate ca 19731230153000 (YYYYMMDDHHMMSS).
* **TIME** - Stochează ora într-un format HH: MM: SS.
* **AN (M)** - Stochează un an într-un format din 2 sau 4 cifre. Dacă lungimea este specificată ca 2 (de exemplu YEAR (2)), YEAR poate fi între 1970 și 2069 (70-69). Dacă lungimea este specificată ca 4, atunci YEAR poate fi între 1901 și 2155. Lungimea implicită este 4.

## **Tipuri de șiruri**

Deși tipurile numerice și de date sunt distractive, majoritatea datelor pe care le veți stoca vor fi în format șir. Această listă descrie tipurile de date obișnuite ale șirurilor în MySQL.

* **CHAR (M)** - Un șir de lungime fixă ​​cuprins între 1 și 255 de caractere (de exemplu CHAR (5)), căptușit la dreapta cu spații la lungimea specificată atunci când este stocat. Definirea unei lungimi nu este necesară, dar valoarea implicită este 1.
* **VARCHAR (M)** - Un șir de lungime variabilă cuprins între 1 și 255 de caractere. De exemplu, VARCHAR (25). Trebuie să definiți o lungime atunci când creați un câmp VARCHAR.
* **BLOB sau TEXT** - Un câmp cu o lungime maximă de 65535 de caractere. BLOB-urile sunt „obiecte mari binare” și sunt utilizate pentru a stoca cantități mari de date binare, cum ar fi imagini sau alte tipuri de fișiere. Câmpurile definite ca TEXT conțin, de asemenea, cantități mari de date. Diferența dintre cele două este că sortările și comparațiile datelor stocate sunt **sensibile** la **majuscule** și **minuscule** pe BLOB-uri și **nu** sunt **sensibile** la **majuscule și minuscule** în câmpurile TEXT. Nu specificați o lungime cu BLOB sau TEXT.
* **TINYBLOB sau TINYTEXT** - O coloană BLOB sau TEXT cu o lungime maximă de 255 de caractere. Nu specificați o lungime cu TINYBLOB sau TINYTEXT.
* **MEDIUMBLOB sau MEDIUMTEXT** - O coloană BLOB sau TEXT cu o lungime maximă de 16777215 caractere. Nu specificați o lungime cu MEDIUMBLOB sau MEDIUMTEXT.
* **LONGBLOB sau LONGTEXT** - O coloană BLOB sau TEXT cu o lungime maximă de 4294967295 caractere. Nu specificați o lungime cu LONGBLOB sau LONGTEXT.
* **ENUM** - O enumerare, care este un termen elegant pentru listă. Când definiți un ENUM, creați o listă de elemente din care trebuie selectată valoarea (sau poate fi NULĂ). De exemplu, dacă doriți ca câmpul dvs. să conțină „A” sau „B” sau „C”, veți defini ENUM ca ENUM („A”, „B”, „C”) și numai acele valori (sau NULL) ar putea popula vreodată acel câmp.

**MySQL** este un [sistem de gestiune a bazelor de date](https://ro.wikipedia.org/wiki/Sistem_de_gestiune_a_bazelor_de_date) [relaționale](https://ro.wikipedia.org/w/index.php?title=Baze_de_date_rela%C8%9Bionale&action=edit&redlink=1), produs de compania suedeza [MySQL AB](https://ro.wikipedia.org/wiki/MySQL_AB) și distribuit sub [Licența Publică Generală GNU](https://ro.wikipedia.org/wiki/Licen%C8%9Ba_Public%C4%83_General%C4%83_GNU). Este cel mai popular [SGBD](https://ro.wikipedia.org/wiki/SGBD) [open-source](https://ro.wikipedia.org/wiki/Open-source) la ora actuală[[6]](https://ro.wikipedia.org/wiki/MySQL#cite_note-6), fiind o componentă cheie a stivei [LAMP](https://ro.wikipedia.org/wiki/LAMP) ([Linux](https://ro.wikipedia.org/wiki/Linux), [Apache](https://ro.wikipedia.org/wiki/Apache), MySQL, [PHP](https://ro.wikipedia.org/wiki/PHP)).

Deși este folosit foarte des împreună cu [limbajul de programare](https://ro.wikipedia.org/wiki/Limbaj_de_programare) [PHP](https://ro.wikipedia.org/wiki/PHP), cu MySQL se pot construi aplicații în orice limbaj major. Există multe scheme API disponibile pentru MySQL ce permit scrierea aplicațiilor în numeroase limbaje de programare pentru accesarea bazelor de date MySQL, cum ar fi: C, C++, C#, Java, [Perl](https://ro.wikipedia.org/wiki/Perl), [PHP](https://ro.wikipedia.org/wiki/PHP), [Python](https://ro.wikipedia.org/wiki/Python), FreeBasic, etc., fiecare dintre acestea folosind un tip specific API. O interfață de tip ODBC denumită MyODBC permite altor limbaje de programare ce folosesc această interfață, să interacționeze cu bazele de date MySQL cum ar fi [ASP](https://ro.wikipedia.org/wiki/ASP) sau [Visual Basic](https://ro.wikipedia.org/wiki/Visual_Basic). În sprijinul acestor limbaje de programare, unele companii produc componente de tip COM/COM+ sau .NET (pentru Windows) prin intermediul cărora respectivele limbaje să poată folosi acest [SGBD](https://ro.wikipedia.org/wiki/SGBD) mult mai ușor decât prin intermediul sistemului ODBC. Aceste componente pot fi gratuite (ca de exemplu MyVBQL) sau comerciale.