PHP –
Hypertext Preprocessor (II)

Funcționalitățile PHP

- PHP generază pagini cu un conținut dinamic.
- PHP creează, deschide, citește, scrie și șterge pagini pe un server.
- PHP colectează datele din formulare.
- PHP poate modifica o conținutul unei baze de date.

Utilizarea fișierelor incluse

Funcțiile PHP permit accesul la programe PHP scrise anterior, create într-un alt fișier extern.

- Funcția **require("nume_fisier");**

Când este încărcat un script PHP care conține o instrucțiune **require**, conținutul fișierului specificat este inserat și executat în script, înlocuind instrucțiunea **require**.
Ex.

```
<?php
require("antet.php");
?>
```

- O altă funcție, similară instrucțiunii **require** este funcția **include()**.
- Spre deosebire de funcția **require()** care introduce datele din fișierul extern intocmai cum sunt scrise, funcția **include()** este o instrucțiune executabilă ce determină evaluarea scriptului PHP din fișierul extern și codul acestuia este executat ca și cum ar fi apărut în textul scriptului unde este inclus.
  Sintaxa functiei **include()** este

  include("nume_fisier.php");

Lucrul cu fişiere

- **Fcţii ptr obţinerea atributelor unui fişier** fct_atribute_fis.pdf
- **Deschiderea unui fişier: fopen("nume_fisier", "mod")**

r - Permite doar citirea fișierului

r+ - Citire și scriere de la inceputul fișierului

w - Creează fișierul dacă nu există și suprascrie datele existente

w+ - Citire și scriere; la scriere , creează fișierul dacă nu există și suprascrie datele existente

a - Adăugare; Creează fișierul dacă nu există și adaugă datele noi la sfârșitul fișierului existent

a+ - Citire sau scriere; la scriere, creează fișierul dacă nu există și adaugă datele noi la sfârșitul fișierului existent

x - Doar scriere; Creează fișierul dacă nu există și generează un averisment dacă acesta există

x+ - Citire și scriere; Creează fișierul dacă nu există și generează un averisment dacă acesta există

b - Deschide fișierul in mod binar

Ex. **$fh = fopen("carte.txt", "r");**

Verificarea finalizării unei operaţii cu un fişier

Ex.

```
<?php
$fh = fopen("carte.txt", "rb");
if (!$fh) {
    echo "Nu a fost deschis fisierul carte.txt.";
}
?>
```

- **Închiderea unui fişier**
  fclose(identificator_fisier)

Operaţii asupra fişierelor

- **Citirea dintr-un fişier**

fread(identificator_fisier, lungime)

- **Citirea unei linii de text**

fgets(identificator_fisier, lungime)

- **Citirea linie cu linie a unui întreg fişier**

feof(identificator_fisier)

- **Afişarea conţinutului unui fişier**

readfile(“nume fisier")

- **Navigarea printr-un fişier**

rewind(identificator_fisier)

fseek(identificator_fisier, offset)

ftell(identificator_fisier)

- **Scrierea într-un fişier**
- **fwrite(identificator_fisier, date);**

```php
<?php
$nume = "carte.txt";
$fh = fopen($nume, "rb");
if (!$fh) {
    echo "Fisierul carte.txt nu a putut fi deschis";
}
while (!feof($fh)) {
    $s = fgets($fh, 256);
    echo "<br /> $s";
}
fclose($fh);
?>
```

```php
<?php
$nume = "carte.txt";
$fh = fopen($nume, "ab");
if (!$fh) {
    echo "Nu a fost deschis fisierul carte.txt.";
}
else {
    $ok = fwrite($fh, "\n Aceste date  s-au adaugat in fisierul carte.txt\n");
    echo "<br /> Rezultatul scris este: $ok";
     fclose($fh);
}
?>
```

Alte operaţii asupra fişierelor

- **Copierea unui fişier** **copy (sursa, destinatiea)**
- **Modificarea numelui unui fişier**
  rename(nume_vechi, nume_nou)
- **Ştergerea unui fişier** **unlink(nume_fisier)**

- *Lista completă cu funcțiile PHP pentru lucru cu sistemul de fișiere și directoare*

http://www.php.net/manual/en/ref.filesystem.php

Crearea formularelor HTML şi recepţionarea datelor de la un formular

- **<form> </form>**
- **utilizarea metodelor GET şi POST**

- datele din formular sunt preluate de scriptul PHP prin:

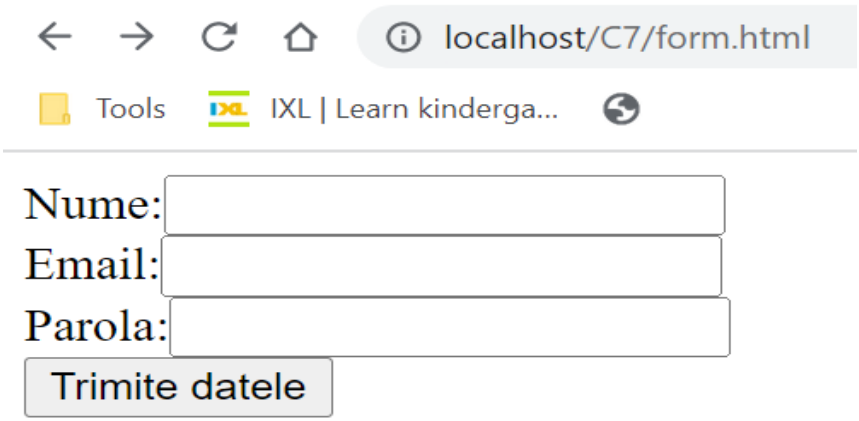
$_POST['nume'] - dacă este folosită **method="post"**

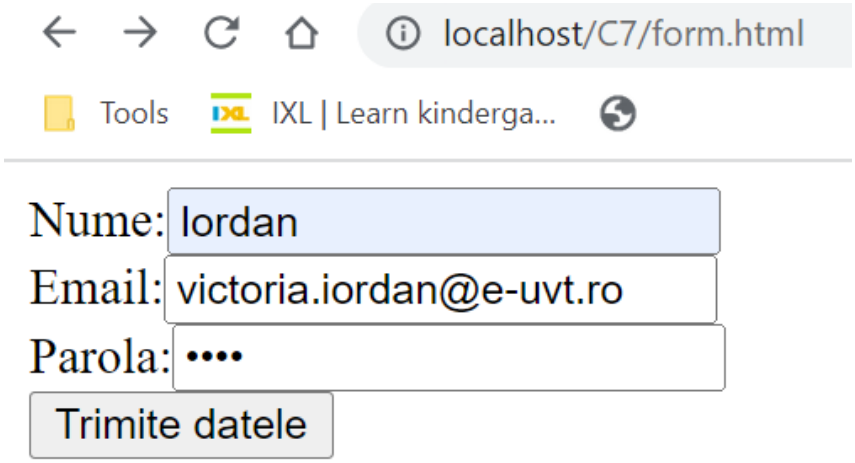
$_GET['nume'] - dacă este folosită **method="get"**

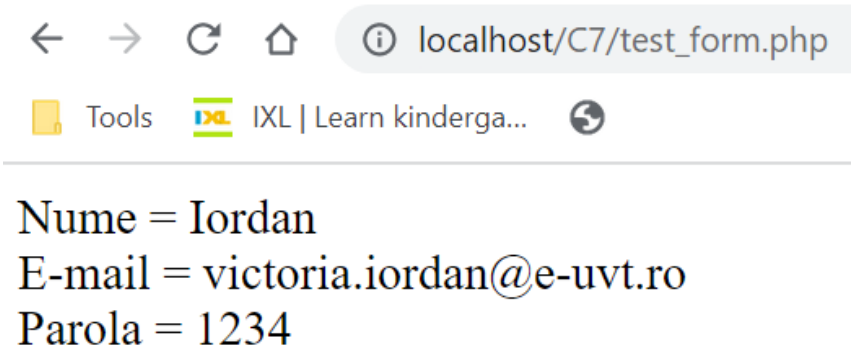
- unde "**nume** este valoarea atributului **name** al elementului din formularul HTML
- Metoda GET trimite toate informațiile adunate ca parte a adresei URL; aceste informații sunt vizibile pentru utilizator.
- Metoda POST transmite informația intr-o manieră invizibilă pentru utilizator și poate transmite o cantitate mai mare de date decât GET.

```
<!doctype html>
<html> <head>
<meta charset="utf-8" />
<title> Test-Form </title> </head>
<body>
<form action="test_form.php" method="POST">
Nume:<input type="text" name="nume" />
<br />Email:<input type="text" name="email" />
<br />Parola:<input type="password" name="parola" />
<br /><input type="submit" name="submit" value="Trimite datele" />
</form>
</body> </html>
```

```
<?php
$nume = $_POST['nume'];
$email = $_POST['email'];
$parola = $_POST['parola'];
echo "Nume = $nume";
echo "<br />E-mail = $email";
echo "<br />Parola = $parola";
?>
```




localhost/C7/form.html
Tools
IXL | Learn kinderga...
Nume: Iordan
Email: victoria.iordan@e-uvt.ro
Parola: ••••
Trimite datele


localhost/C7/test_form.php
Tools
IXL | Learn kinderga...
Nume = Iordan
E-mail = victoria.iordan@e-uvt.ro
Parola = 1234

- Pentru a prelua şi folosi datele **$_GET['nume']**

Ex.

- **$var1 = $_GET['nume1']**
- **$var2 = $_GET['nume2']**

```
<form action="test_form_get.php" method="GET">
.....
$nume = $_GET['nume'];
$email = $_GET['email'];
$parola = $_GET['parola'];
.....
```

- **Trimiterea de date unui script prin adresa URL**

http://adresa/fisier.php?nume1=valoare1&nume2=valoare2

PHP şi MySQL

- **Bazele de date** sunt colecţii de date, aranjate într-o anumită formă

- Operaţii :
- Crearea bazei de date;
- Inserarea datelor in baza de date;
- Ștergerea datelor din baza de date;
- Adăugarea sau modificarea datelor;

Baze de date RELAŢIONALE

- înmagazinează datele în tabele care se pot lega logic după valorile anumitor coloane
- Informatiile sunt organizate in tabele
- Inregistrare / câmp
- bazate pe teoria matematică a relaţiilor, proiectarea bazelor de date poate fi tratată algoritmic
- utilizează un limbaj standardizat de interogare a bazei de date numit SQL (Structured Query Language)

- Pentru implementarea unei baze de date este nevoie de un sistem de gestiune a bazelor de date (SGBD).
- Scopul unui sistem de gestiune al unei baze de date este acela de a oferi un mediu care să fie convenabil, dar şi eficient pentru a putea fi folosit la:

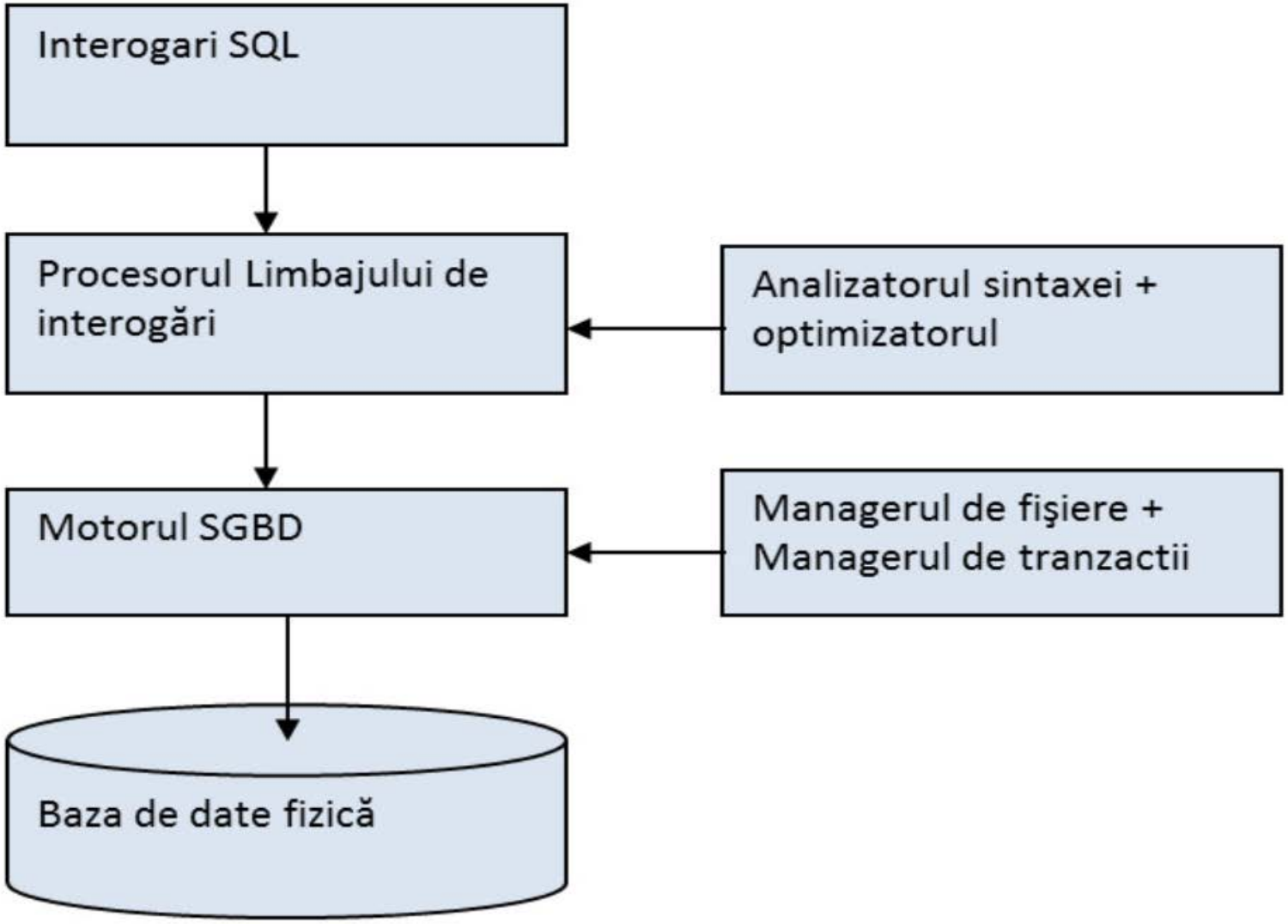
 extragerea informaţiilor din baza de date;

 înmagazinarea datelor în baza de date.

- Microsoft Access;
- Visual Foxpro;
- MySQL;
- Oracle;

- Administrarea sistemului MySQL se poate face fie din linia de comandă, fie folosind aplicaţia PHPMyAdmin.

- localhost/phpmyadmin/


Interogari SQL
Procesorul Limbajului de interogări
Analizatorul sintaxei + optimizatorul
Motorul SGBD
Managerul de fişiere + Managerul de tranzactii
Baza de date fizică

Crearea unei baze de date MySQL

- **CREATE DATABASE nume_db;**
- **USE nume_db;**

Ex.
CREATE DATABASE test;

- Comenzile uzuale:

1. **DDL – Data Definition Language**

- CREATE – creează o bază de date şi/sau un tabel
- DROP - şterge o bază de date şi/sau un tabel
- ALTER - modifică structura unui tabel după ce acesta a fost creat cu instrucțiunea CREATE TABLE

2. **DML – Data Manipulation Language**

- INSERT – adaugă înregistrări(linii) într-un tabel
- DELETE - şterge înregistrări(linii) dintr-un tabel
- UPDATE – modifică înregistrările dintr-un tabel
- SELECT – selectează înregistrările dintr-un tabel

Limbajul de definire a datelor (DDL) gestionează structura datelor

CREATE TABLE tabel (

câmp1 tip1,

câmp2 tip2, …

PRIMARY KEY (cheie1, cheie2, ...));

Ex.

```
CREATE TABLE carte (carteid CHAR(10),
                    titlu VARCHAR(255),
                    pret decimal(5,2));
```

Obs. Cheia primară este o coloană sau o combinație de coloane care definesc în mod unic un rând într-un tabel al unei baze de date relaționale. O tabelă poate avea cel mult o cheie primară. Cheia primară impune constrângerea implicită NOT NULL.

- Tipuri de date folosite în MySQL(o parte din ele):

- Int – număr întreg
- Char – secţiune cu lungime fixă de max. 255 caractere
- Varchar – secţiune variabilă de max 255 caractere
- Float – număr real mic
- Double – număr real mare
- Text – şir de maxim 65535 caractere
- Date – data im format an-luna-zi
- Time – ora in format oră-minut-secundă

- atribute opţionale ale unui câmp:

- **NOT NULL** - Fiecare rând trebuie să conţină o valoare a coloanei asociate; valorile nule nu sunt permise
- **DEFAULT valoare** - Dacă nu este dată o valoare a coloanei asociate, se va presupune valoarea specificată.
- **AUTO INCREMENT** - MySQL va repartiza în mod automat un număr de serie ca valoare a coloanei asociate.
- **PRIMARY KEY** - Coloana asociată este cheia primară a tabelului care o conţine.

Ex.

```
CREATE TABLE carte (
    carteid CHAR(10) PRIMARY KEY,
    titlu VARCHAR(255) NOT NULL,
    pret DECIMAL(5,2) DEFAULT 50.00);
```

- **ALTER TABLE** modifică structura unui tabel existent prin redenumirea/adăugarea/schimbarea/ștergerea structurii unei coloane sau index:

```
ALTER TABLE tabel RENAME TO nume_nou_tabel;
ALTER TABLE table_name ADD ( câmp1 def1, col2 def2, ... );
ALTER TABLE table_name MODIFY (câmp1 tip1, câmp2 tip2, ... );
ALTER TABLE tabel DROP COLUMN câmp;
ALTER TABLE tabel RENAME COLUMN nume_vechi TO nume_nou;
```

- **TRUNCATE TABLE** – Șterge toate articolele unui tabel:

```
TRUNCATE TABLE tabel;
```

- **DROP TABLE** – Șterge tabelul:

```
DROP TABLE tabel;
```

Accesul la datele dintr-o bază de date, interogările SQL

- **DML – Data Manipulation Language**

SELECT * FROM tabel;

SELECT col1, col2, … FROM tabel [WHERE cond];

INSERT INTO tabel VALUES (val1, val2, …);

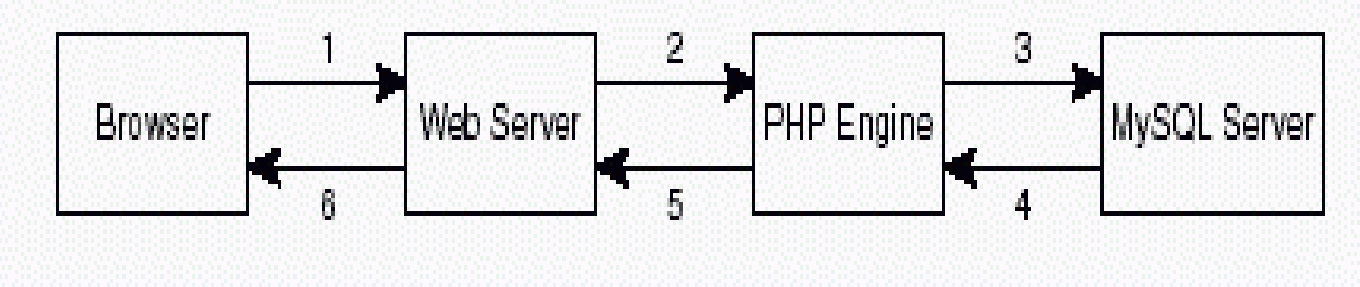
INSERT INTO tabel (col1, col2, …) VALUES (val1, val2, …);

UPDATE tabel SET col1=val1, col2=val2 WHERE conditie;

DELETE FROM tabel WHERE conditie;

- **Sortarea şi gruparea**
- **ORDER BY valoare**
- **GROUP BY coloana-sortare**
- **HAVING conditie**

- Un browser web nu poate decât să afişeze pagini HTML. Prin PHP se poate realiza conexiunea la MySQL, se pot trimite interogări şi se poate afişa rezultatul interogărilor în format HTML, format înţeles de un browser web. Principiul de funcţionare este cel descris în figura următoare:



1. **Browser-ul unui utilizator face o cerere HTTP pentru o anumită pagină (spre exemplu se caută o anumită carte pe un site prin intermediul fișierului results.php).**
2. **Server-ul Web primește cererea pentru results.php, caută fișierul și il pasează motorului PHP pentru a fi procesat.**
3. **Motorul PHP incepe procesarea și găsește o comandă de conectare la o baza de date și o interogare după un anumit criteriu. PHP deschide o conexiune cu server-ul MySQL și ii trimite interogarea.**
4. **Server-ul MySQL primește interogarea și o procesează și trimite rezultatul (o listă de cărți, in cazul nostru) PHP-ului.**
5. **PHP-ul termină de rulat script-ul (de obicei are loc o formatare a rezultatelor interogării in HTML) și returnează fișierul HTML rezultat server-ului Web**
6. **Server-ul Web trimite HTML-ul inapoi către browser de unde utilizatorul poate vedea lista cărților pe care le-a cerut**

PHP MySQL

- PHP are trei moduri diferite prin care se poate conecta și interacționa cu baza de date MySQL :

1. extensia MySQL originală (*cu functii*),
2. MySQL Improved (*MySQLi*), sau
3. PHP Data Objects (*PDO*).

Ele nu pot fi amestecate in același script. Extensia originală MySQL nu mai este activ dezvoltată si nu este recomandată pentru proiecte PHP-MySQL noi. Documentația PHP descrie **MySQLi** ca fiind opțiunea preferată recomandată de MySQL pentru proiecte noi.

PDO (*PHP Data Objects*) este o extensie PHP pentru accesare baze de date in PHP

- **PDO** folosește caracteristicile OOP (*Programare Orientata pe Obiecte*) valabile incepând cu PHP 5.1.

PDO poate lucra cu următoarele tipuri de baze de date:

- MySQL
- PostgreSQL
- SQLite 2 & 3
- Firebird
- Informix (IBM Informix Dynamic Server)
- ODBC
- Oracle
- DBLM: FreeTDS / Sybase / MS-SQL
- IBM (IBM DB2)

- Unul din avantajele PDO este acela că se folosesc funcții similare pentru interogarea și prelucrarea bazelor de date, indiferent de tipul lor (din cele menționate mai sus).
  Script-urile care folosesc interfața PDO pentru conectare la baza de date efectuează in general următoarele operații:

1. Conectare la serverul bazei de date, prin apelare **new PDO()**, obținând un obiect pentru lucrul cu acea bază de date.
2. Aplicare de funcții specifice PDO pentru efectuarea interogărilor la baza de date.
3. Reținerea și prelucrarea datelor returnate.
4. Deconectarea de la server.

Utilizarea bazelor de date folosind PHP

- Inainte de a putea accesa informații din baza de date, trebuie creată o conexiune cu serverul MySQL.
- **Conectarea la serverul MySQL**

PDO - Pentru conectarea la un server MySQL cu PHP și MySQLi, se creează o **instanță de obiect mysqli**, prin **new mysqli()**, la care se adaugă datele de conectare.

$conn=new mysqli_connect($nume_gazda, $nume_utilizator, $parola, $bazadate, $port) ;

(port implicit: 3306, altfel “localhost:port”)

sau – stil procedural-

$conn=mysqli_connect($nume_gazda, $nume_utilizator, $parola, $bazadate, $port)

- **Incheierea conexiunii: mysqli_close()**;

Ex.

```
<?php
$conn = mysqli_connect( "localhost", "root", "")
if (!$conn) {
exit("Eroare la conectare ". mysqli_connect_error($conn));
}
// Aici adaugati datele necesare lucrului cu serverul MySQL
mysqli_close($conn);
?>
```

Creare baza de date

- CREATE DATABASE nume_bd

```
$createdb = mysqli_query($conn,"CREATE DATABASE test");
if ($createdb)
    echo "Baza de date test a fost creata <br />";
else
    echo "<br />". mysqli_errno($conn). " : ". mysqli_error($conn);
```

- **Interogări**

mysqli_query("interogare")

mysqli_query("id_con", "interogare")

- "interogare" este un şir ce conţine comenzile SQL care urmează a fi executate (*in PHP, comenzile SQL nu trebuie să se încheie cu un caracter punct şi virgula*)
- "id_con" este identificatorul de conectare returnat de funcţia *mysql_connect()*, dacă acesta este omis se foloseşte ultima legatură deschisă cu această funcţie.

- **Crearea unui tabel**

```
$sql = "CREATE TABLE `carti` (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
nume varchar(30) NOT NULL,
autor varchar(25) ,
gen varchar(10) NOT NULL,
data_intrare date,
pret decimal(12,2))";
if (mysqli_query($conn,$sql))
    echo "Tabelul carti a fost creat <br />";
else
    echo "Tabelul carti nu a putut fi creat  : ". mysqli_errno($conn). " : ". mysqli_error($conn);
```

- **Inserarea de înregistrări**

```
$sql = "INSERT INTO `carti` (nume, autor, gen, data_intrare, pret) VALUES ('Poezii', 'Lucian Blaga', 'poezie', '2021-04-06', '45')";
if (mysqli_query($conn,$sql))
    echo 'Datele au fost adaugate';
else
    echo "Datele nu au fost adaugate deoarece : ". mysqli_errno($conn). " : ". mysqli_error($conn);
```

Prelucrarea rezultatelor interogărilor SELECT

- **mysqli_fetch_assoc() (*sau* mysqli_fetch_array() cu parametru "MYSQL_ASSOC").**
  - sunt similare, adaugă datele intr-o matrice unde pentru fiecare element avem chei cu numele coloanelor, iar valorile lor sunt datele din rândul respectiv.
- **mysqli_fetch_row() - adaugă datele intr-o matrice asociativă unde pentru fiecare element avem chei cu numere consecutive (începând de la 0) care reprezintă ordinea coloanelor, iar valorile lor sunt datele din rândul respectiv**
- **mysqli_fetch_object() - aduce rezultatele rândului sub forma de obiect cu perechile $rand->coloana**

- **Afișarea unui tabel**

```
// interogare sql SELECT
$sql = "SELECT * FROM `carti`";
// executa interogarea si retine datele returnate
$result = mysqli_query($conn,$sql);
// daca $result contine cel putin un rand
if ($result) {
  // afiseaza datele din fiecare rand din $result
  while($row = mysqli_fetch_array($result)) {
            echo '<br /> id: '. $row['id']. ' - nume: '. $row['nume']. ' - autor: '.
$row['autor'];
  }
}
else {
  echo '0 rezultate';
}
```

DELETE

DELETE from nume_table WHERE conditie

Instrucțiunea DELETE este trimisă la serverul MySQL cu metoda **query()** a obiectului mysqli.

Pentru a șterge complet un tabel, se folosește DROP TABLE:

DROP TABLE nume_table

Pentru a șterge o intreagă bază de date, cu toate tabelele și informațiile din ea, se folosește DROP DATABASE:

DROP DATABASE database_name

```
…
// interogare DELETE
$sql = "DELETE FROM `carti` WHERE `nume`='Poezii'";
// executa interogarea si verifica daca exista erori
if (!mysqli_query($conn,$sql)) {
  echo ("Error: ". mysqli_error($conn));
}
$sql1 = "SELECT `id`, `nume`, `autor` FROM `carti`";
$result = mysqli_query($conn,$sql1);
while($row = mysqli_fetch_array($result)) {
    echo '<br /> id: '. $row['id']. ' - nume: '. $row['nume']. ' - autor: '.
$row['autor'];
  }
…
```

Actualizare

UPDATE nume_table

SET col1=val1,col2=val2, …

WHERE o_col=o_val

comanda UPDATE este trimisă serverului MySQL cu metoda **query()** a obiectului mysqli

```
…
// interogare sql UPDATE
$sql = "UPDATE `carti` SET `pret`='55' WHERE `nume`='Poezii'";
// executa interogarea si verifica pentru erori
if (!mysqli_query($conn,$sql)) {
  echo ("Error: ". mysqli_error($conn));
}
$sql1 = "SELECT `id`, `nume`, `autor`,`pret`  FROM `carti`";
$result = mysqli_query($conn,$sql1);
while($row = mysqli_fetch_row($result)) {
    echo '<br /> id: '. $row[0]. ' - nume: '. $row[1]. ' - autor: '. $row[2]. ' -
pret: '. $row[3];
  }
…
```

Aplicație

- **Realizarea unui formular de prelucrare a datelor, introducerea informațiilor in baza de date, validareea datelor și vizualizarea acestora**

Problema:

- Formularul va avea următoarele câmpuri: nume (tipul text), prenume (tipul text), vârsta (tipul text), email (tipul text) și comentariu (textarea).
- Cerința: Toate câmpurile sunt obligatorii, câmpurile nume și prenume trebuie să conțină numai litere, câmpul vârsta să conțină numai cifre, câmpul comentariu să aibe maxim 255 caractere, adresa emai se va valida.