

Curs 7

2022/2023

# Programarea aplicațiilor web

- Programarea aplicațiilor web
  - An V RC
    - 1.5C/1L/1P

# Program

- An V
  - Saptamana 1
    - Luni 17-20 Curs (Intro/HTML/CSS)
  - Saptamanile 2-8
    - Luni 16(17)-18 Curs
    - Luni 18-20 Laborator
  - Saptamanile 9-14
    - Luni 16(17)-18 Curs
    - Luni 18-20 Proiect

# Orar

- <https://orar.etti.tuiasi.ro/> : C->16(17)-18, L/P -> 18



FACULTATEA DE ELECTRONICA, TELECOMUNICATII SI TEHNOLOGIA INFORMATIEI

55RC

ETTI\_

	1 8:00 - 8:50	2 9:00 - 9:50	3 10:00 - 10:50	4 11:00 - 11:50	5 12:00 - 12:50	6 13:00 - 13:50	7 14:00 - 14:50	8 15:00 - 15:50	9 16:00 - 16:50	10 17:00 - 17:50	11 18:00 - 18:50	12 19:00 - 19:50
L										Damian R. PAW (C) 2.13 TC (R)	Damian R. PAW (L) 2.13 TC (R)	
Ma								C1	Scripcariu L. RCALSC (C)		Scripcariu L. RCALSC (L) 2.13 TC (R)	
Mi								Casian-Botez I. Etic (C) P6 (Amf.)	Casian-Botez I. Etic (S) P6 (Amf.)		Trifina L. TEFO (L) 3.25 TTI (L) Alecsandrescu I. POO (L) CI5(Corp C)	
J									Sirbu A. POO (C) P2 (Amf.)		Trifina L. TEFO (L) 3.25 TTI (L)	
V											Trifina L. TEFO (C) 3.25 TTI (L)	

# Nota

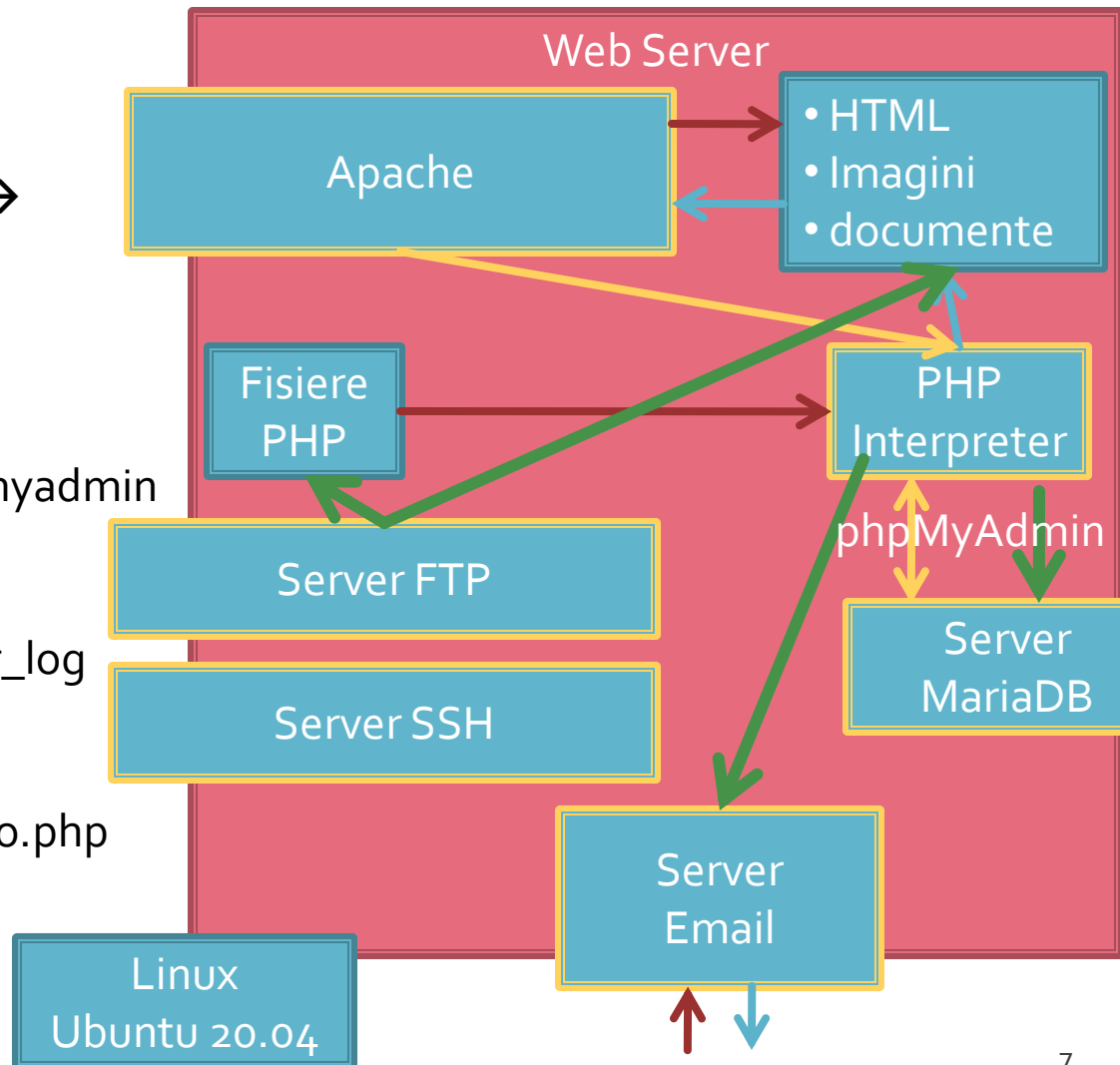
- An V
  - 33% E
  - 66% Aplicatii
    - 33% L
    - 33% P

# Nota

- An V
  - 33% E
  - 33% L
  - 33% P
- Laborator - Prezenta
  - 1pz = 1p ( $p > 5 \leftrightarrow$  Curs)
- Examen
  - Prezenta la curs: 3pz = 0.5p
  - Asemanator cu materia de proiect
- Activitate suplimentara
  - Dupa terminarea activitatii la laborator
  - +2 **1**p la E/L

# Utilizzare LAMP

1. login → **paw**:masteretti
2. ifconfig → 192.168.30.5
3. putty.exe → 192.168.30.5 → SSH → **paw**:masteretti (remote login)
4. [alte comenzi linux dorite]
5. FTP → Winscp → SFTP → student:masterrc@192.168.30.5
6. MySql → http://192.168.30.5/phpmyadmin → **root**:masteretti
7. Apache Error Log →
  - 7a. putty → nano /var/log/httpd/error\_log
  - 7b. http://192.168.30.5/logfile.php (nonstandard)
8. PHP info → http://192.168.30.5/info.php



# Tema bonus

- logfile.php
  - Afiseaza log Apache (erori php majore)
- ~~1p 2p~~ **suplimentar** la laborator/examen
- Modificare logfile.php pentru a afisa **toate** erorile PHP
  - php.ini – activare erori ✓
  - php.ini – locatie erori ✓
  - logfile.php – afisare log PHP ✓
  - drepturi de acces la fisiere (Linux)



# Server referinta LAMP

- Linux, doua variante
  - Centos 7.1
    - PHP 5.4.16
    - MariaDB 5.5.44 / root:masterrc
    - Apache 2.4.6
    - **root**/student:masterrc
  - **Ubuntu** 20.04
    - PHP 7.4.3
    - MariaDB 10.3.31 / root:masteretti
    - Apache 2.4.41
    - **paw**/student:masteretti
    - necesar suplimentar pentru **acces FTP paw**:
      - **sudo usermod -a -G upload paw**
      - **sudo chmod -R 775 /var/www**

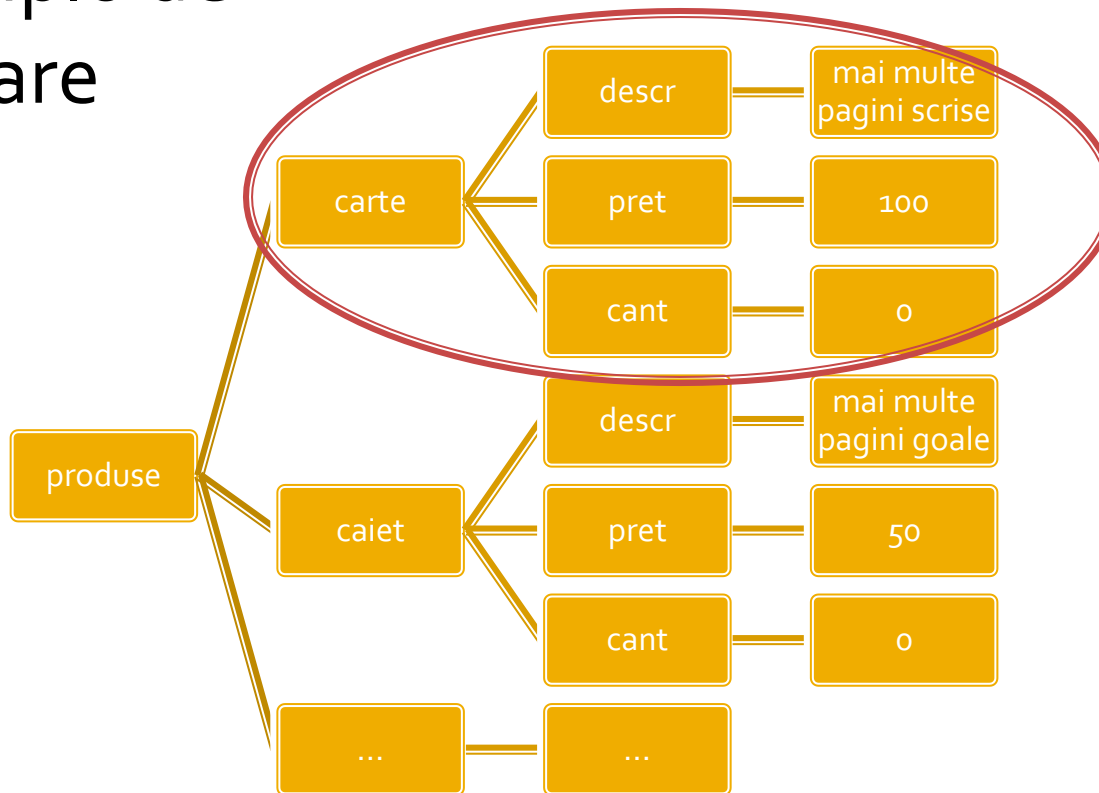
# Laborator 5

# Laborator 5

- Sa se continue magazinul virtual cu:
  - produsele sunt grupate pe **categorii** de produse
  - sa prezinte utilizatorului o lista de categorii de produse pentru a alege
  - sa prezinte utilizatorului o lista de produse si preturi **in categoria** aleasa la pasul anterior
  - lista de produse si preturi se citeste dintr-un **fisier text**
  - se preia comanda si se calculeaza suma totala
- Optional
  - se creaza o pagina prin care vanzatorul poate **modifica** preturile si produsele
  - fisierul care contine date in format **XML**

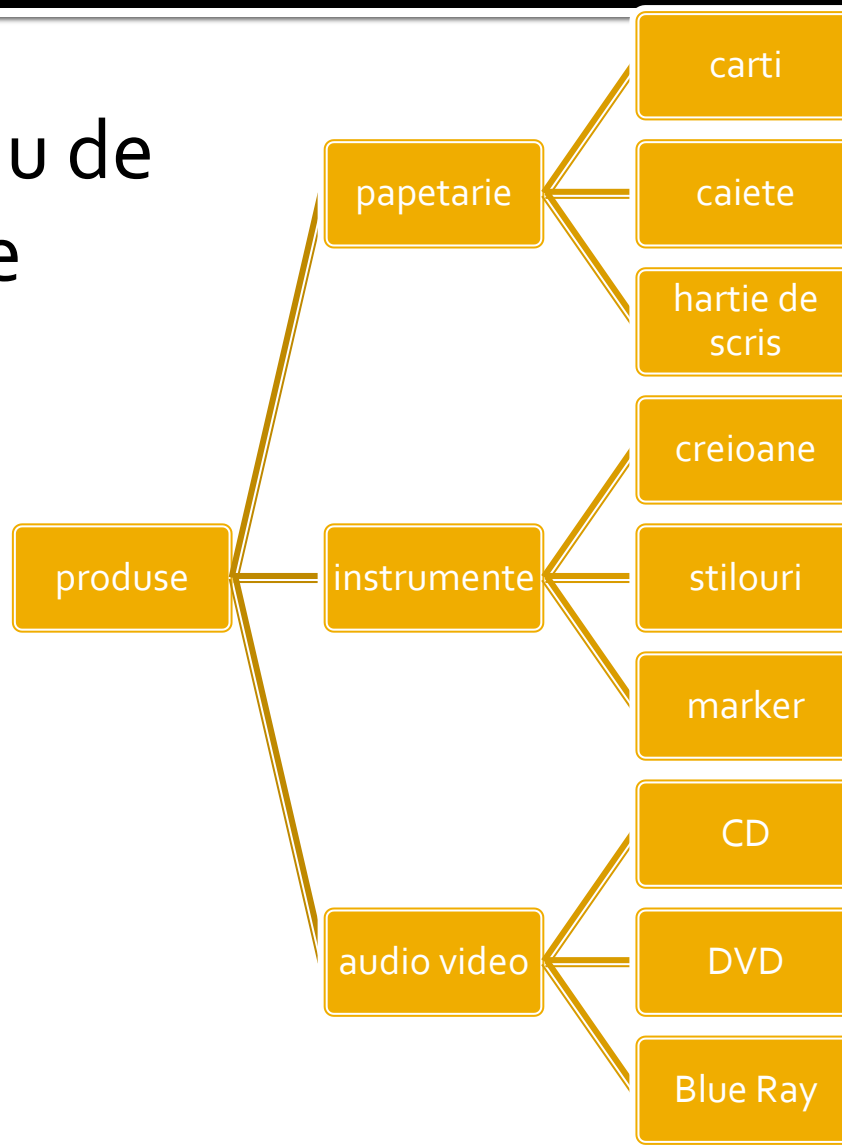
# Laborator 4 – Tablou produse

- exemplu de grupare



# Laborator 5 – Tablou produse

- exemplu de grupare



# Rezultat

## Categorii Produse

Alegeti categoria:

Nr.	Categorie	Total Produse
1	<a href="#">Papetarie</a>	3
2	<a href="#">Instrumente</a>	3
3	<a href="#">Audio-video</a>	3
4	<a href="#">Calculatoare</a>	3
5	<a href="#">Jucarii</a>	2

Total produse: 14

## Magazin online Firma X SRL

### Realizati comanda

Nr.	Produs	Pret	Cantitate
1	Carti	100	<input type="text" value="1"/>
2	Caiete	50	<input type="text" value="2"/>
3	Penare	150	<input type="text" value="1"/>
4	Stilouri	125	<input type="text" value="0"/>
5	Creioane	25	<input type="text" value="0"/>

## Magazin online Firma X SRL

### Rezultate comanda

Pret total (fara TVA): 350

Pret total (cu TVA): 416.5

Comanda receptionata la data: 17/03/2010 ora 08:24

PHP

# Artificii de programare utilizate

# Template

antet.php

```
<html>
<head>
<title>Magazin online Firma X
SRL</title>
</head>
<body bgcolor="#CCFFFF"><?php
define('PRET_CARTE',100);
define('PRET_CALET',50);
define('PRET_PENAR',150);
define('PRET_STILOU',125);
define('PRET_CREION',25);
//orice cod comun PHP
?><table width="600" border="0"
align="center">
<tr><td></td></tr>
<tr><td height="600" valign="top"
bgcolor="#FFFFCC">
<h1>Magazin online Firma X SRL</h1>
```

subsol.php

```
</td></tr>
</table>
</body>
</html>
```

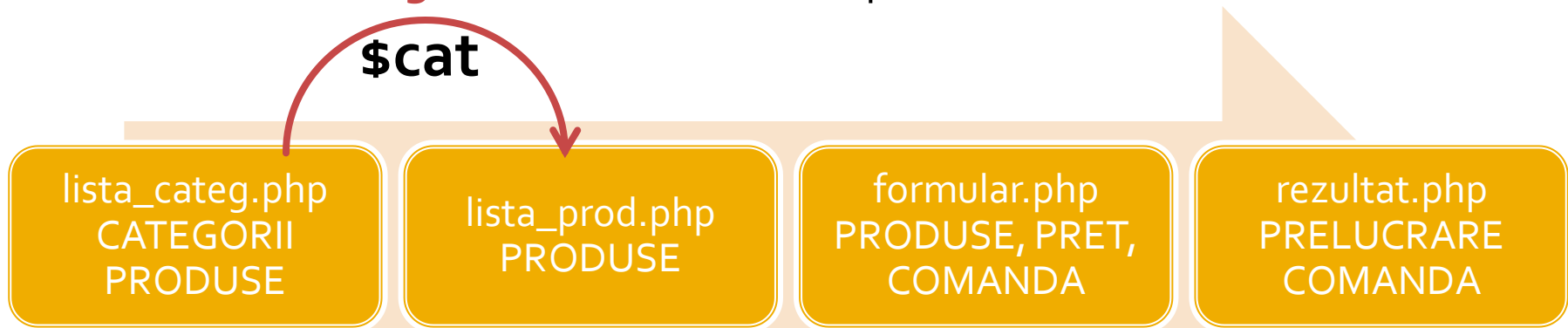
\*.php

```
<?php require('antet.php');?>
<h2>Lista Produe</h2>
<table border="1">
...
</table>
<?php require('subsol.php');?>
```



# Link purtator de informatie

- **get** datele sunt atasate adresei documentului de procesare :  
results.php?prob=81&an=2009
- se poate simula realizarea unei forme (**get**) prin scrierea corespunzatoare a link-urilor
- in **lista\_categ.php**
  - `<a href="lista_prod.php?categ=<?php echo $cat;?>"> <?php echo $cat;?> </a>`
- are efect in **lista\_prod.php**
  - `$_GET['categ']="valoarea $cat corespunzatoare"`



# Elemente de control – foreach

- `foreach (array_expression as $key => $value) statement`
- `foreach (array_expression as $value) statement`
- iterarea prin **fiecare** element al tabloului
- la fiecare element variabila declarata in instructiune variabila locala **\$key** ofera acces la cheia curenta iar variabila locala **\$value** ofera acces la valoarea asociata
- `foreach()` lucreaza cu o **copie** a tabloului deci tabloul original nu va fi modificat prin schimbarea continutului variabilelor `$key` si `$value`

# Fisier text

- Fisier text pentru stocarea informatiilor

carte	mai multe pagini scrise legate	100	0	
caiet	mai multe pagini goale legate	75	0	
penar	loc de depozitat instrumente de scris	150	0	
stilou	instrument de scris albastru	125	0	
papetarie	carte	mai multe pagini scrise legate	100	0
papetarie	caiet	mai multe pagini goale legate	75	0
instrumente	penar	loc de depozitat instrumente de scris	150	0
instrumente	stilou	instrument de scris albastru	125	0

# Cod

## ■ Extragere valori din text


```
$vect=file("produse.txt");  
foreach ($vect as $linie)  
{  
    $valori=explode("\t",$linie,4);  
    $produse[$valori[0]]=array ("descr" => $valori[1], "pret" => $valori[2], "cant" => $valori[3]);  
}
```

```
$vect=file("produse.txt");  
foreach ($vect as $linie)  
{  
    $valori=explode("\t",$linie,5);  
    $produse[$valori[0]] [$valori[1]]=array ("descr" => $valori[2], "pret" => $valori[3], "cant" =>  
$valori[4]);  
}
```

# Utilizare date

## ■ Utilizare, cu doua bucle foreach

```
$index=1;
foreach ($produse as $prod => $detalii) //primul indice in $produse imi da produsul
{
    <tr><td><?php echo $index;?></td><td><?php echo ucfirst(strtolower($prod));?></td><td><?php echo
    $detalii['descr'];?></td><td align="center"><?php echo $detalii['pret'];?></td></tr>
    <?php $index++;
}
```



```
$index=1;
foreach ($produse as $categ => $lista_categ) //primul indice in $produse imi da categoria
    foreach ($lista_categ as $prod => $detalii) //al doilea indice in $produse imi da produsul
        //din categoria stabilita cu primul indice
    {
        <tr><td><?php echo $index;?></td><td><?php echo $categ;?></td><td><?php echo
        ucfirst(strtolower($prod));?></td><td><?php echo $detalii['descr'];?></td><td align="center"><?php
        echo $detalii['pret'];?></td></tr>
        <?php $index++;
    }
```

# Laborator 6

# Plan aplicatie – Cumparator

- Pe masura ce aplicatia paraseste un fir liniar de executie este necesara introducerea unui plan (graf) al aplicatiei
- Cumparator
  - citirea fisierului text (XML) se realizeaza in antet.php, comun pentru toate fisierele

lista\_categ.php  
ALEGERE CATEGORIE

formular.php  
INTRODUCERE DATE

rezultat.php  
PRELUCRARE  
COMANDA

# Rezultat (comparator)

**Categorii Produse**

Alegeti categoria:

Nr.	Categorie	Total Produse
1	<a href="#">Papetarie</a>	3
2	<a href="#">Instrumente</a>	3
3	<a href="#">Audio-video</a>	3
4	<a href="#">Calculatoare</a>	3
5	<a href="#">Jucarii</a>	2

Total produse: 14

**Magazin online Firma X SRL**

Finalizati comanda

Nr.	Produs	Pret	Cantitate
1	Carti	100	<input type="text" value="1"/>
2	Caiete	50	<input type="text" value="2"/>
3	Penare	150	<input type="text" value="1"/>
4	Stilouri	125	<input type="text" value="0"/>
5	Creioane	25	<input type="text" value="0"/>

**Magazin online Firma X SRL**

**Rezultate comanda**

Pret total (fara TVA): 350

Pret total (cu TVA): 416.5

Comanda receptionata la data: 17/03/2010 ora 08:24

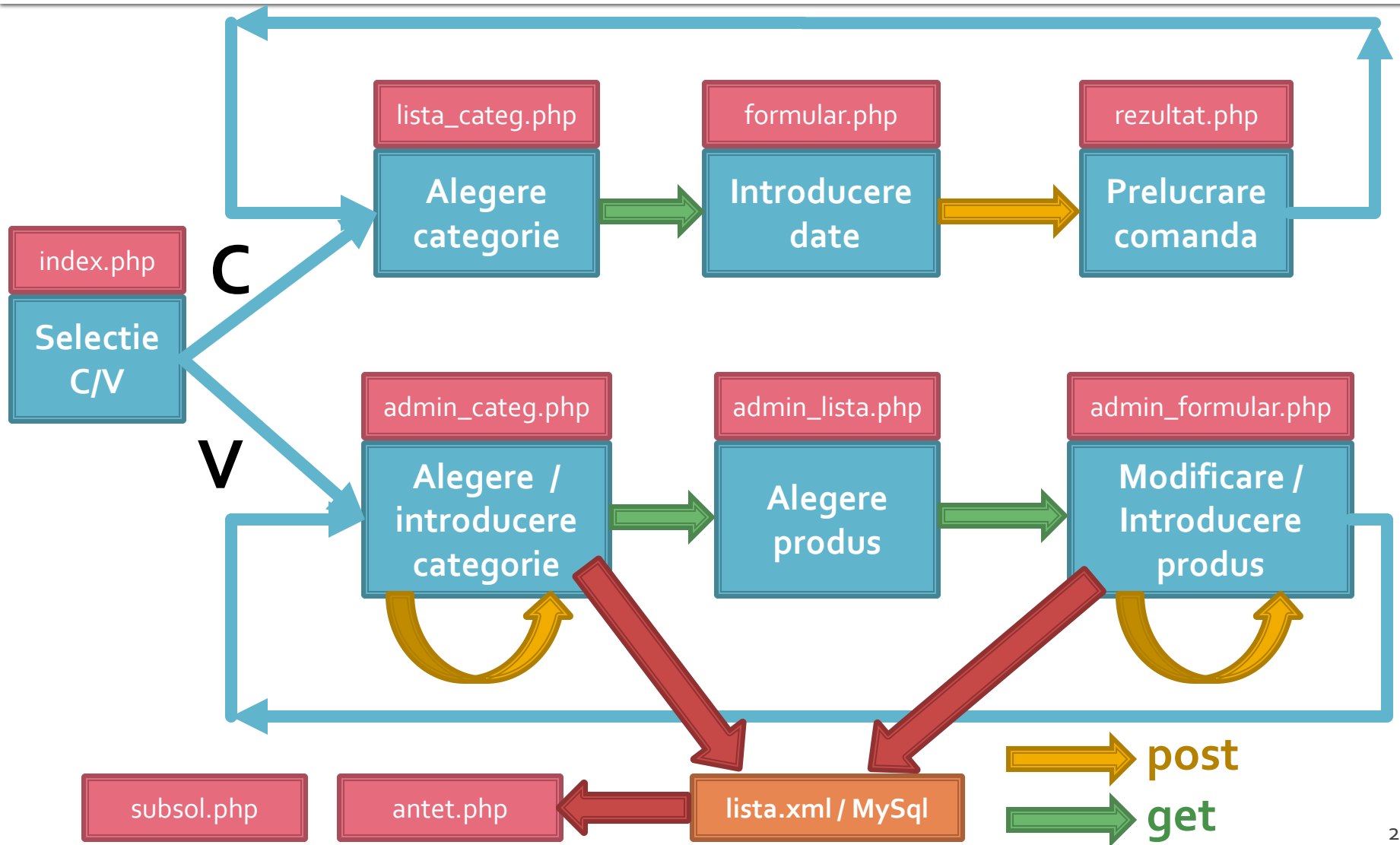




# Plan aplicatie – Vanzator

- Aparitia aplicatiei pentru vanzator
  - introduce un fir paralel de executie cu necesitatea alegerii initiale: cumparator/vanzator
  - aduce posibilitatea scrierii fisierului XML
  - diverse operatii de scriere
    - introducere categorie de produse
    - introducere produs nou intr-o categorie existenta
    - modificare produs existent
  - modificarea fisierului implica 2 actiuni:
    - colectare date
    - prelucrare

# Plan aplicatie (Proiect !!)



# Rezultat (vanzator)

**Magazin** Firma X

[Inceput](#) | [Inapoi](#)

## Magazin online Firma X SRL

Alegeti:

- [Cumparator](#)
- [Vanzator](#)

### Categorii Produse

Alegeti categoria:

Nr.	Categorie	Total Produse
1	<a href="#">Papetarie</a>	3
2	<a href="#">Instrumente</a>	3
3	<a href="#">Audio-video</a>	3
4	<a href="#">Calculatoare</a>	3
5	<a href="#">Jucarii</a>	2

Total produse: 14

Categorie noua de produse:

### Lista produse in categoria Calculatoare

Nr.	Produs	Descriere	Pret	Cantitate	Actiuni
1	Laptop	calculator mic	2000	2	<a href="#">modifica</a>
2	Desktop	calculator mare	1000	5	<a href="#">modifica</a>
3	Imprimanta	prn	200	2	<a href="#">modifica</a>
-	Produs nou				<a href="#">adauga</a>

### Produs in categoria Calculatoare

Produs	<input type="text" value="laptop"/>
Descriere	<input type="text" value="calculator mic"/>
Pret	<input type="text" value="2000"/>
Cantitate	<input type="text" value="2"/>



# Laborator 6(7)

- Sa se continue magazinul virtual cu:
  - produsele sunt grupate pe categorii de produse
  - sa prezinte utilizatorului o lista de grupe de produse pentru a alege
  - sa prezinte utilizatorului o lista de produse si preturi in grupa aleasa
  - lista de produse si preturi se citeste dintr-o baza de date **MySQL**
  - se preia comanda si se calculeaza suma totala
  - **se creaza paginile prin care vanzatorul poate modifica preturile, produsele, categoriile**

PHP

# Artificii de programare utilizate

# Rezultat (vanzator)

**Magazin** Firma X

[Inceput](#) | [Inapoi](#)

## Magazin online Firma X SRL

Alegeti:

- [Cumparator](#)
- [Vanzator](#)

### Categorii Produse

Alegeti categoria:

Nr.	Categorie	Total Produse
1	<a href="#">Papetarie</a>	3
2	<a href="#">Instrumente</a>	3
3	<a href="#">Audio-video</a>	3
4	<a href="#">Calculatoare</a>	3
5	<a href="#">Jucarii</a>	2

Total produse: 14

Categorie noua de produse:

### Lista produse in categoria Calculatoare

Nr.	Produs	Descriere	Pret	Cantitate	Actiuni
1	Laptop	calculator mic	2000	2	<a href="#">modifica</a>
2	Desktop	calculator mare	1000	5	<a href="#">modifica</a>
3	Imprimanta	prn	200	2	<a href="#">modifica</a>
-	Produs nou				<a href="#">adauga</a>

### Produs in categoria Calculatoare

Produs	<input type="text" value="laptop"/>
Descriere	<input type="text" value="calculator mic"/>
Pret	<input type="text" value="2000"/>
Cantitate	<input type="text" value="2"/>

 post

 get

# Fisier unic pentru colectare SI prelucrare date

- De multe ori se prefera aceasta varianta
- Permite pastrarea unitara a tuturor operatiilor pentru indeplinirea unei actiuni
  - acces mai simplu
  - usurinta la programare
  - evitarea erorilor: File does not exist: D:/Server/...
- Acelasi fisier e folosit initial pentru a colecta date si apoi, daca se detecteaza prezenta acestora, pentru prelucrarea lor

# Fisier unic pentru colectare SI prelucrare date

- Fisierul de receptie pentru <form> va fi fisierul curent
- se recomanda utilizarea variabilei globale `$_SERVER['SCRIPT_NAME']`
  - flexibilitate la redenumirea fisierelor
- alternativ `$_SERVER['PHP_SELF']` nu este recomandata
  - probleme de securitate
- Sectiunea de colectare date se afiseaza numai in absenta datelor


```
<form action="<?php echo $_SERVER['SCRIPT_NAME'];?>" method="post">  
<p><input name="date_ok" type="submit" value="Trimite" /></p>  
</form>
```



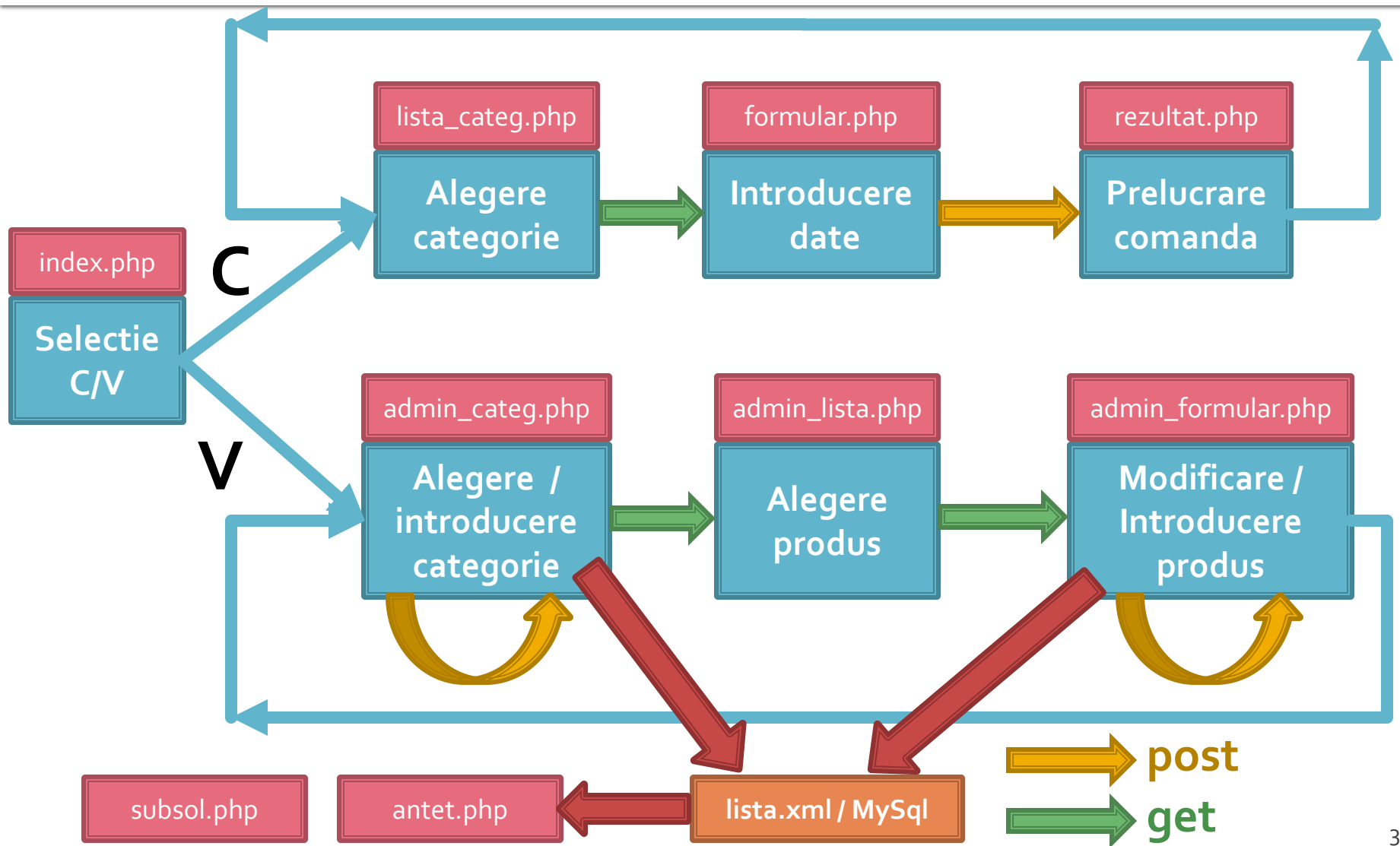
# Fisier unic pentru colectare SI prelucrare date

- Detectia existentei datelor se face prin verificarea existentei ( isset(\$variabila) ) valorilor introduse
  - eventual pentru un plus de protectie se poate verifica si continutul lor

```
if (isset($_POST["date_ok "]))
{ //date trimise
  if ($_POST["date_ok "]=="Trimite" )
  { //date trimise de fisierul curent
    //prelucrare
  }
}
else
{
  //colectare date
  <form action="<?php echo $_SERVER['SCRIPT_NAME '];?>" method="post">
  <p><input name="date_ok" type="submit" value="Trimite" /></p></form>
}
```



# Plan aplicatie



Stocare eficienta a datelor

# MySQL

# MySql

- Baza de date – instrument pentru stocarea si manipularea informatiei eficient si efectiv
  - datele sunt protejate de corupere sau pierderi accidentale
  - nu se utilizeaza mai multe resurse decat minimul necesar
  - datele pot fi accesate cu performanta acceptabila
- Baze de date relationale
  - model relational (matematic eficient) – Codd ~1970

# DBMS, RDBMS

- DBMS – database management system aplicatii incluse in baza de date pentru accesul la informatii
- RDBMS – Relational DBMS. Majoritatea sistemelor de baze de date tind la aceasta titulatura
  - ~300 de reguli trebuie respectate
  - nici un sistem actual nu implementeaza total aceste reguli

# Relatii

- Toate sistemele de baze de date sunt caracterizate de:
  - toate informatiile sunt reprezentate intr-o aranjare ordonata **bidimensionala** numita **relatie**
  - toate valorile (attribute) stocate sunt scalare (in orice celula din tabel se stocheaza **o singura** valoare)
  - toate operatiile se aplica asupra unei intregi relatii si rezulta o intreaga relatie
- Terminologii (**MySQL**)
  - tabel – **table** / recordset / **result set**
  - linie – record / **row**
  - coloana – field / **column**

# Chei

- Din toate combinatiile de coloane care pot fi utilizate pentru identificarea unica a unei linii se alege **macar** una utilizata intern de RDBMS pentru ordonarea datelor – **cheie primara**
  - Celelalte chei candidate devin **chei alternative** si pot fi folosite pentru eficientizarea prelucrarilor (crearea de “index” dupa aceste chei)
- In cazul in care nu exista o combinatie de coloane utilizabila ca si cheie cu utilitate practica se introduce artificial o cheie, cu numere intregi incrementate automat de DBMS (autoincrement)
  - de multe ori este recomandata o astfel de actiune, numerele intregi fiind mult mai usor de controlat, ordonat, cautat decat alte tipuri de date
  - cheile de tip autoincrement nu e **nevoie** sa contina informatie

MySql

# Relatii in Bazele de date



# Relatii in Bazele de date

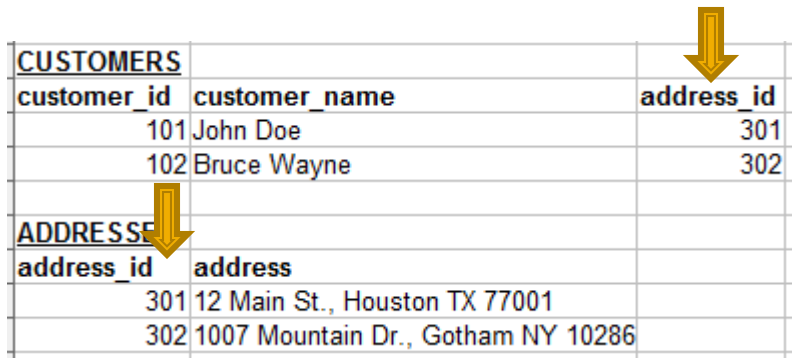
- Legaturile intre tabele pot fi
  - One to One
  - One to Many
  - Many to Many
    - Unare (auto referinta)

# One to One

- Fiecare tabel poate avea corespondenta **o singura linie (row) sau nici una** de cealalta parte a relatiei
- echivalent cu o relatie “bijectiva”
- analogie cu casatorie:
  - o persoana poate fi casatorita sau nu
  - daca este casatorita va fi casatorita cu o singura persoana din tabelul cu persoane de sex opus
  - persoana respectiva va fi caracterizata de aceeasi relatie “one to one” – primeste simultan un singur corespondent in tabelul initial

# One to One

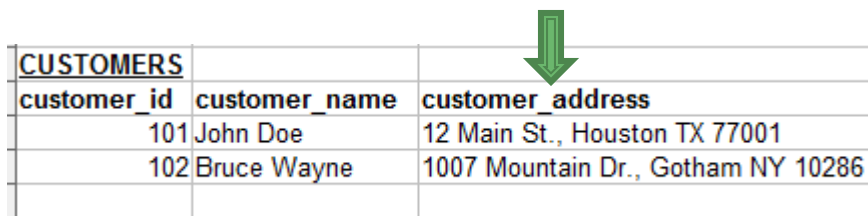
- de multe ori legaturile "one to one" se bazeaza pe reguli externe
- de obicei se poate realiza usor si eficient gruparea ambelor tabele in unul singur



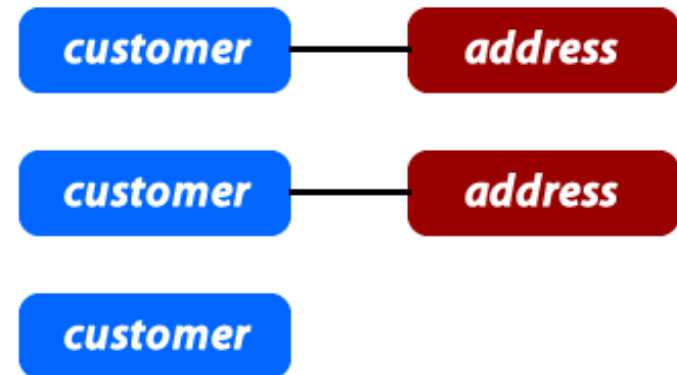
CUSTOMERS		
customer_id	customer_name	address_id
101	John Doe	301
102	Bruce Wayne	302

ADDRESSES	
address_id	address
301	12 Main St., Houston TX 77001
302	1007 Mountain Dr., Gotham NY 10286



CUSTOMERS		
customer_id	customer_name	customer_address
101	John Doe	12 Main St., Houston TX 77001
102	Bruce Wayne	1007 Mountain Dr., Gotham NY 10286




# One to Many


- O linie dintr-un tabel (row), identificata prin cheia primara, poate avea: **nici una, una sau mai multe linii corespondente** in celalalt tabel. In acesta o linie poate fi legata cu o **singura** linie din tabelul primar.
- Analogie cu relatii parinte/copil:
  - fiecare om are o singura mama
  - fiecare femeie poate avea nici unul, unul sau mai multi copii

# One to Many, Many to One

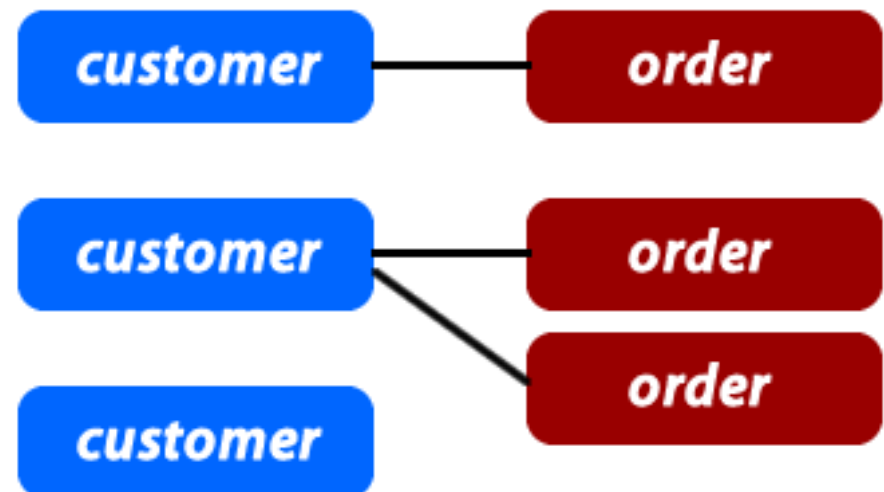
- de obicei aceste legaturi se implementeaza prin introducerea cheii primare din tabelul **One** in calitate de coloana in tabelul **Many** (cheie externa – foreign key)



CUSTOMERS	
customer_id	customer_name
101	John Doe
102	Bruce Wayne



ORDERS			
order_id	customer_id	order_date	amount
555	101	12/24/09	\$156.78
556	102	12/25/09	\$99.99
557	101	12/26/09	\$75.00



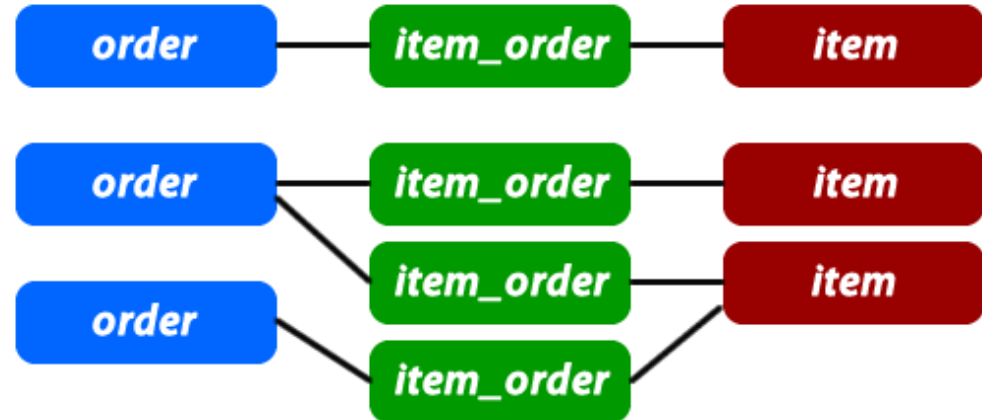
# Many to Many

- Fiecare linie (row) din **ambele tabele** implicate in legatura poate fi legat cu **oricate (niciuna, una sau mai multe) linii** din tabelul corespondent.
- Analogie cu relatii de rudenie (veri de exemplu), tabel 1 – barbati, tabel 2 – femei :
  - fiecare barbat poate fi ruda cu una sau mai multe femei
  - la randul ei fiecare femeie poate fi ruda cu unul sau mai multi barbati

# Many to Many

- de obicei aceste legaturi se implementeaza prin introducerea unui tabel **suplimentar** (numit tabel **asociat** sau de **legatura**) care sa memoreze legaturile

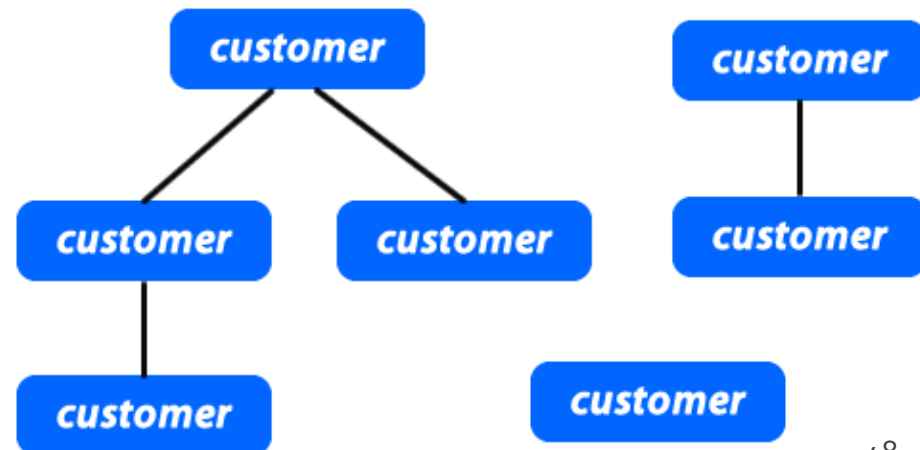
ORDERS				
order_id	customer_id	order_date	amount	
555	101	12/24/09	\$156.78	
556	102	12/25/09	\$99.99	
ITEMS				
item_id	item_name	item_description		
201	Tickle Me Elmo	It wants to be tickled		
202	District 9 DVD	Awesome sci-fi movie		
203	Batarang	It is very sharp		
ITEMS_ORDERS				
order_id	item_id			
555	201			
555	202			
556	202			
556	203			



# Self Referencing (unare)

- Un caz particular de legatura "one to many" in care legatura e in interiorul aceluiasi tabel
- rezolvarea este similara, introducerea unei coloane suplimentara, cu referinta la cheia primara din tabel
- analogie cu relatii parinte copil cand ambele persoane se regasesc in acelasi tabel

CUSTOMERS		
customer_id	customer_name	referrer_customer_id
101	John Doe	0
102	Bruce Wayne	101
103	James Smith	101





# Relatii in Bazele de date

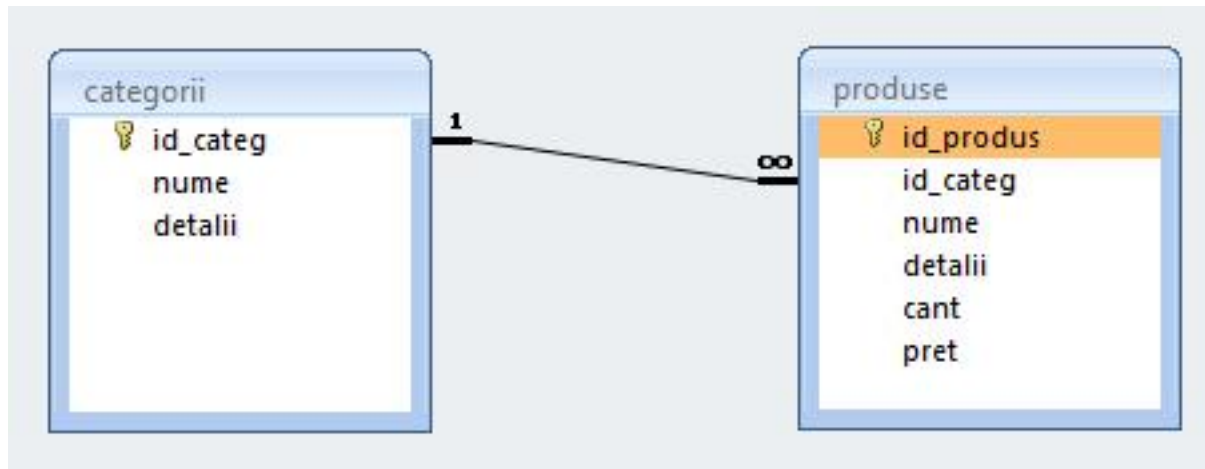
- Respectarea formelor normale ale bazelor de date aduce nenumarate avantaje
- Efectul secundar este dat de necesitatea separarii datelor intre mai multe tabele
- In exemplul utilizat avem doua concepte diferite din punct de vedere logic
  - produs
  - categorie de produs

# Relatii in Bazele de date

- In exemplul utilizat avem doua concepte diferite din punct de vedere logic
  - produs
  - categorie de produs
- Cele doua tabele nu sunt independente
- Intre ele exista o legatura data de functionalitatea dorita pentru aplicatie: **un produs va apartine unei anumite categorii de produse**

# Relatii in Bazele de date

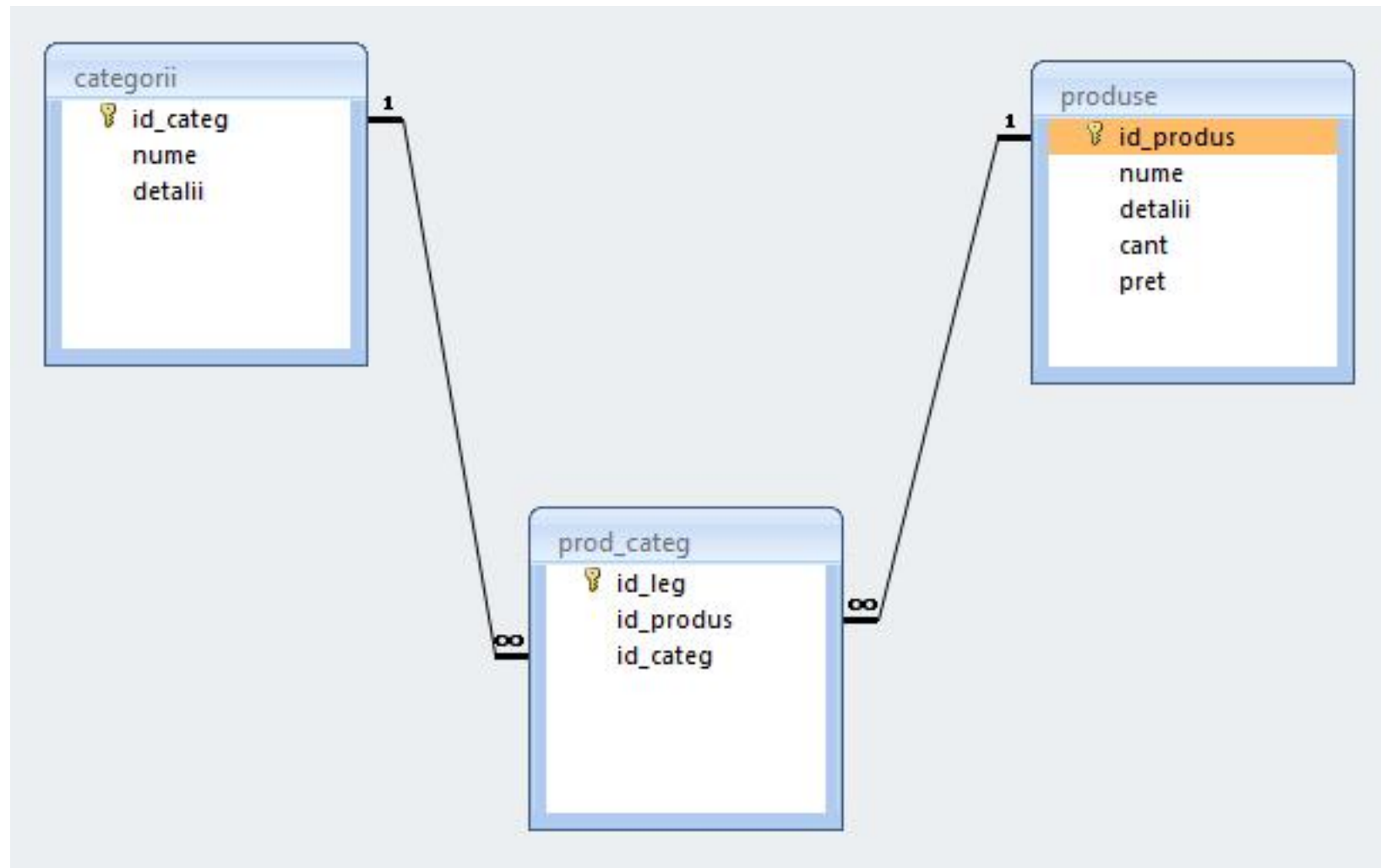
- Legaturile implementata
  - One to Many
  - in tabelul "produse" apare cheia externa (foreign key): "id\_categ"



# Relatii in Bazele de date

- Daca se doreste o situatie cand un produs poate apartine **mai multor categorii** (o carte cu CD poate fi inclusa si in "papetarie" si in "audio-video")
  - relatia devine de tipul **Many to Many**
  - e necesara introducerea unui tabel de legatura cu coloanele "id\_leg" (cheie primara), "id\_categorie" si "id\_produs" (chei externe)

# Relatii in Bazele de date



# Relatii

- Nu trebuie evitate relatiile
  - Many to Many
  - One to Many
- Prelucrarea cade in sarcina server-ului de baze de date (RDBMS)
  - JOIN – **esential** in aplicatii cu baze de date

# MySql – eficienta

- eficienta unei aplicatii web
  - 100% - **toate prelucrarile "mutate" in RDBMS**
  - PHP **doar** afisarea datelor
- eficienta unei aplicatii MySql
  - 25% **alegerea corecta a tipurilor de date**
  - 25% **crearea indecsilor necesari in aplicatii**
  - 25% **normalizarea corecta a bazei de date**
  - 20% **cresterea complexitatii interogarilor pentru a "muta" prelucrarile pe server-ul de baze de date**
  - 5% **scrierea corecta a interogarilor**

MySql

# Tipuri de date



# MySql – tipuri de date

- numeric
  - intregi
    - BIT (implicit 1 bit)
    - TINYINT (implicit 8 biti)
    - SMALLINT (implicit 16 biti)
    - INTEGER (implicit 32biti)
    - BIGINT (implicit 64biti)
  - real
    - FLOAT
    - DOUBLE
    - DECIMAL – fixed point

# MySql – tipuri de date

- data/timp
  - DATE ('YYYY-MM-DD')
    - '1000-01-01' pana la '9999-12-31'
  - DATETIME ('YYYY-MM-DD HH:MM:SS')
    - '1000-01-01 00:00:00' pana la '9999-12-31 23:59:59'
  - TIMESTAMP ('YYYY-MM-DD HH:MM:SS')
    - '1970-01-01 00:00:00' pana la partial 2037

# MySql – tipuri de date

- sir
  - CHAR (M)
    - sir de lungime constanta M,  $M < 255$
  - VARCHAR (M)
    - sir de lungime variabila, maxim M,  $M < 255$  ( $M < 65535$ )
- cantitati mari de date
  - TEXT
    - au alocat un set de caractere, operatiile tin cont de acesta
  - BLOB
    - sir de octeti, operatiile tin cont de valoarea numerica
  - TINYBLOB/TINYTEXT, BLOB/TEXT, MEDIUMBLOB/MEDIUMTEXT, LARGEBLOB/LARGETEXT
    - date  $2^8-1$ ,  $2^{16}-1$ ,  $2^{24}-1$ ,  $2^{32}-1 = 4\text{GB}$

# MySQL – tipuri de date

- enumerare

- ENUM('val1','val2',...)

- una singura din cele maxim 65535 valori distincte posibile

- SET('val1','val2',...)

- niciuna sau mai multe din cele maxim 64 valori distincte
    - echivalent cu "setare de biti" intr-un intreg pe 64 biti cu tabela asociata

# Metode de stocare

# Metode de stocare

- Metoda de stocare a datelor nu e o caracteristica a server-ului ci a fiecarui tabel in parte
- Exemplu ulterior CREATE: "ENGINE = InnoDB"
- MySql suporta diferite metode de stocare, fiecare cu avantajele/dezavantajele sale
- Implicit se foloseste metoda MyISAM, dar la instalarea server-ului (laborator 1) o anumita selectie poate schimba valoarea implicita in InnoDB
- **Alegerea metodei de stocare potrivita are implicatii majore asupra performantei aplicatiei**

# Metode de stocare

- MyISAM
- InnoDB
- Memory
- Merge
- Archive
- Federated
- NDBCLUSTER
- CSV
- Blackhole
- Example

# Metode de stocare

## ■ MyISAM

- metoda de stocare implicita in MySql
- performanta ridicata (resurse ocupate si viteza)
- posibilitatea cautarii in intregul text (index FULLTEXT)
- blocare acces la nivel de tabel
- nu accepta tranzactii
- nu accepta FOREIGN KEY
  - probleme relative la integritatea datelor

## ■ InnoDB

## ■ Memory



# Metode de stocare

- **MyISAM**

- **InnoDB**

- devine metoda de stocare implicita in MySql daca la instalare se alege model tranzactional
- performanta medie (resurse ocupate si viteza)
- blocare acces la nivel de linie
- **nu** accepta index FULLTEXT
  - incepand cu MySql 5.6.4 este introdus index FULLTEXT
- **accepta** tranzactii
- **accepta** FOREIGN KEY
  - probleme mai putine la integritatea datelor prin constrangeri intre tabele

- **Memory**

# Metode de stocare

- MyISAM
- InnoDB
- **Memory**
  - metoda de stocare recomandata pentru tabele temporare
  - performanta maxima (viteza – datele sunt stocate in RAM)
    - la oprirea server-ului datele se pierd, tabelul este pastrat dar va fi fara nici o linie
  - nu accepta tipuri de date mari (BLOB, TEXT) – maxim 255 octeti
  - nu accepta index FULLTEXT
  - nu accepta tranzactii
  - nu accepta FOREIGN KEY
    - probleme relative la integritatea datelor

# Acces la server-ul MySql din PHP

# Acces la server-ul MySQL din PHP

- Bibliotecile corespunzatoare trebuie activate in php.ini – vezi laboratorul 1.
  - mysql
  - mysqli (improved accesul la functionalitati ulterioare MySQL 4.1)
- O baza de date existenta poate fi accesata daca exista un utilizator cunoscut in PHP cu drepturi de acces corespunzatoare – vezi laboratorul 1.
- O baza de date poate fi creata si din PHP dar nu e metoda recomandata daca nu e necesara
  - cod dificil de implementat pentru o singura utilizare
  - necesita existenta unui utilizatori cu drepturi mai mari pentru crearea bazei de date si alocarea de drepturi unui utilizator restrans

# Funcții PHP de acces MySQL

- `mysql_connect`
  - realizarea unei conexiuni cu server-ul MySQL
  - resource `mysql_connect` ( string server, string user, string password )
  - rezultatul
    - succes – resursa (conexiune, link\_identifier)
    - eșec – false
- `mysql_select_db`
  - selectează `baza de date` de pe server cu care se va lucra în continuare
  - bool `mysql_select_db` (string database [, resource link\_identifier])

# Funcții PHP de acces MySQL

- `mysql_query`
  - trimiterea unei interogari SQL spre server
  - resource `mysql_query` ( string query [, resource link\_identifier] )
  - rezultatul
    - SELECT, SHOW, DESCRIBE sau EXPLAIN – resursa (tabel)
    - UPDATE, DELETE, DROP, etc – true/false
- `mysql_fetch_assoc`
  - returneaza o **matrice asociativa** corespunzatoare liniei de la indexul intern (indecsi de tip sir corespunzatori denumirii coloanelor – field – din tabelul de date) si incrementeaza indexul intern sau **false** daca nu mai sunt linii
  - array `mysql_fetch_assoc` ( resource result )

# Functii PHP de acces MySql

## Parcurgerea resurselor rezultat

- `mysql_fetch_assoc`
  - returneaza o **matrice asociativa** corespunzatoare liniei de la indexul intern (indecsi de tip sir corespunzatori denumirii coloanelor – field – din tabelul de date) si incrementeaza indexul intern sau **false** daca nu mai sunt linii
  - array **`mysql_fetch_assoc`** ( resource result )
- `mysql_fetch_row`
  - returneaza o matrice cu indecsi intregi
  - array `mysql_fetch_row` ( resource result )

# Functii PHP de acces MySql

## Parcurgerea resurselor rezultat

- `mysql_fetch_array`
  - grupeaza functionalitatea `mysql_fetch_assoc` si `mysql_fetch_row`
  - array `mysql_fetch_array` ( resource result [, int result\_type] )
  - MYSQL\_ASSOC, MYSQL\_NUM, MYSQL\_BOTH (implicit)
- `mysql_data_seek`
  - muta indexul intern la valoarea indicata
  - bool `mysql_data_seek` ( resource result, int row\_number )



# Resurse MySQL

- Resursele reprezinta o combinatie intre
  - date structurate (valori + structura) rezultate in urma unor interogari SQL
  - functii de acces la aceste date/structuri
- Analogie cu POO
  - o "clasa speciala" creata in urma interogarii cu functii predefinite de acces la datele respective

# Resurse MySQL

## Structura

Index intern	Col 1 (tip date)	Col 2 (tip date)	....
1			
2			
...			

## Date

Index intern	Col 1	Col 2	....
1	Val 11	Val 12	...
2	Val 21	Val 22	...
...	...	...	...

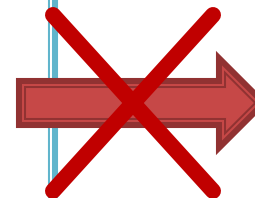
Functii de acces la structura



Functii de acces la date



~~Acces direct~~



# Resurse MySQL

- Functiile de acces la structura sunt rareori utilizate
  - majoritatea aplicatiilor sunt concepute pe structura fixa, si cunosc structura datelor primite
  - exceptie: aplicatii generale, ex.: PhpMyAdmin
- Majoritatea functiilor de acces la date sunt caracterizate de acces secvential
  - se citesc in intregime valorile stocate pe o linie
  - simultan se avanseaza indexul intern pe urmatoarea pozitie, pregatindu-se urmatoarea citire

# Resurse MySQL

- Functiile sunt optimizate pentru utilizarea lor intr-o structura de control **do {} while()**, sau **while() {}** de control
  - returneaza FALSE cand "s-a ajuns la capat"
- tipic se realizeaza o citire (mysql\_fetch\_assoc) urmata de o bucla **do {} while()**
  - pentru a se putea introduce cod de detectie probleme rulat o singura data

# Exemplu de utilizare

```
$hostname = "localhost";  
$database = "world";  
$username = "web";  
$password = "ceva";  
$conex= mysql_connect($hostname, $username, $password);  
mysql_select_db($database, $ conex);
```

```
$query = "SELECT `Code`, `Name`, `Population` FROM `country` AS c ";  
$result = mysql_query($ query, $ conex) or die(mysql_error());  
$row_result = mysql_fetch_assoc($ result );  
$totalRows_result = mysql_num_rows($ result );
```

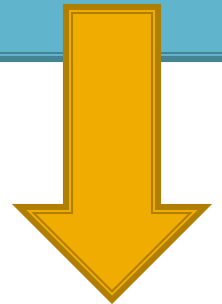
# Exemplu de utilizare

```
<?php
do {?>
<tr>
    <td><?php echo $index; ?>&nbsp;</td>
    <td><?php echo $ row_result ['Code']; ?>&nbsp;</td>
    <td><?php echo $ row_result ['Name']; ?>&nbsp;</td>
    <td><?php echo $ row_result ['Population']; ?>&nbsp;</td>
</tr>
<?php
    $index++;
}
while ($ row_result = mysql_fetch_assoc($ result )); ?>
```

# Modificari laborator cu date stocate text

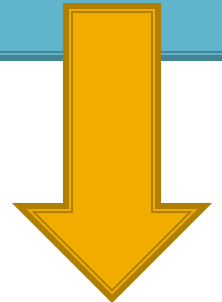
- Codul aplicatiei ramane in mare parte acelasi
- Se modifica doar citirea valorilor pentru popularea matricii \$produse ("antet.php")

```
$vect=file("produse.txt");  
foreach ($vect as $linie)  
{  
    $valori=explode("\t",$linie,5);  
    $produse[$valori[0]] [$valori[1]]=array ("descr" => $valori[2], "pret" => $valori[3], "cant" =>  
$valori[4]);  
}
```



# Modificari laborator cu date stocate XML

```
$xml = simplexml_load_file("lista.xml");
if ($xml)
{
    foreach ($xml->categorie as $categorie)
    {
        $produce[(string)$categorie["nume"]]=array();
        foreach ($categorie->produs as $prod_cur)
        {
            $produce[(string)$categorie["nume"]][(string)$prod_cur->nume]=array
            ("descr" => (string)$prod_cur->desc, "pret" => (string)$prod_cur->pret,
            "cant" => (string)$prod_cur->cant);
        }
    }
}
```





# Modificari laborator cu date stocate

## MySQL

```
$hostname = "localhost";
$database = "tmpaw";
$username = "web";
$password = "test";
$conex= mysql_connect($hostname, $username, $password);
mysql_select_db($database, $conex);
$query = "SELECT * FROM `categorii` AS c";
$result_c = mysql_query($query, $conex) or die(mysql_error());
$row_result_c = mysql_fetch_assoc($result_c);
$totalRows_result_c = mysql_num_rows($result_c);
do {
    $query = "SELECT * FROM `produse` AS p WHERE `id_categ` = ".$row_result_c['id_categ'];
    $result_p = mysql_query($query, $conex) or die(mysql_error());
    $row_result_p = mysql_fetch_assoc($result_p);
    $totalRows_result_p = mysql_num_rows($result_p);
    $produse[$row_result_c['nume']] = array();
    do {
        $produse[$row_result_c['nume']][$row_result_p['nume']] = array ("descr" =>
$row_result_p['detalii'], "pret" => $row_result_p['pret'], "cant" => $row_result_p['cant']);
    }
    while ($row_result_p = mysql_fetch_assoc($result_p));
}
while ($row_result_c = mysql_fetch_assoc($result_c));
```

# Optimizare

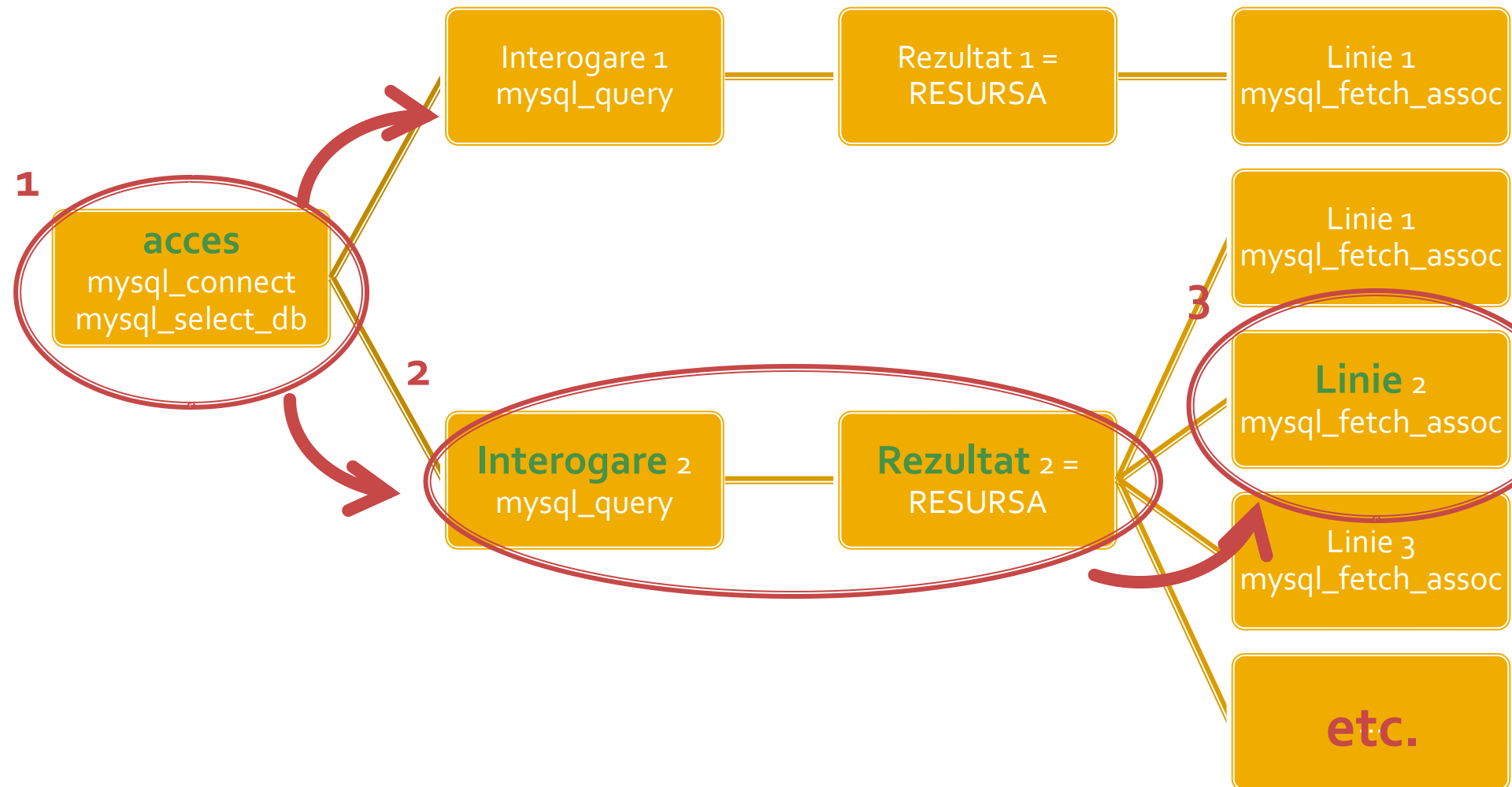
- o singura interogare SQL, unirea tabelelor (JOIN) lasata in baza server-ului MySql

```
$hostname = "localhost";
$database = "tmpaw";
$username = "web";
$password = "test";
$conex= mysql_connect($hostname, $username, $password);
mysql_select_db($database, $conex);

$query = "SELECT p.*, c.`nume` AS `nume_categ` FROM `produse` AS p
        LEFT JOIN `categorii` AS c ON (c.`id_categ` = p.`id_categ`)";
$result = mysql_query($query, $conex) or die(mysql_error());
$row_result = mysql_fetch_assoc($result);
$totalRows_result = mysql_num_rows($result);

do{
    $produse[$row_result['nume_categ']][$row_result['nume']]=array ("descr" => $row_result['detalii'], "pret"
=> $row_result['pret'], "cant" => $row_result['cant']);
}
while ($row_result = mysql_fetch_assoc($result));
```

# Functii de acces la server-ul MySQL



!! IMPORTANT

**PHP > 5.5**

# PHP 5.5

- Incapand cu versiunea 5.5 a PHP extensia mysql este declarata **depreciata**
  - orice utilizare a unei functii genereaza eroare de tip **E\_DEPRECATED**
  - se preconizeaza ca in PHP > 6 aceasta extensie va fi eliminata total (**realizat**)
- Alternativele de utilizare sunt
  - extensia **mysqli** (MySQL Improved)
  - extensia **PDO** (PHP Data Objects)

# Extensia mysql

- Inafara securitatii sporite ofera acces la facilitatile curente ale server-ului MySQL
  - accesul la interogari predefinite (Prepared Statements) (viteza, securitate)
    - server side
    - client side
  - proceduri stocate pe server (viteza, securitate)
  - interogari multiple
  - tranzactii (integritate)

# Extensia mysqli

- Doua modalitati de utilizare
  - procedurala (similar mysql)
  - POO (similar PDO)
- Utilizarea procedurala (aproape) similara cu utilizarea extensiei originale mysql
  - tranzitie facila
  - tranzitie cu mici diferente de parametri

# mysqli – Procedural

```
<?php
$mysqli = mysqli_connect("example.com", "user", "password", "database");
$res = mysqli_query($mysqli, "SELECT 'Please do not use the mysql extension ' AS _msg FROM DUAL");
$row = mysqli_fetch_assoc($res);
echo $row['_msg'];

$mysql = mysql_connect("example.com", "user", "password");
mysql_select_db("test");
$res = mysql_query("SELECT ' for new developments.' AS _msg FROM DUAL", $mysql);
$row = mysql_fetch_assoc($res);
echo $row['_msg'];
?>
```

- toate functiile mysql au un echivalent mysqli
- majoritatea functiilor au aceeasi parametri in aceeasi ordine
- sunt totusi functii cu mici diferente (Ex: **mysqli\_connect**, **mysqli\_query**)



# mysqli – Programare orientata object

```
<?php
$var = new mysqli("example.com", "user", "password", "database");
$res = $var->query ($mysqli, "SELECT 'Please do not use the mysql extension ' AS _msg FROM DUAL");
$row = $res->fetch_assoc();
echo $row['_msg'];

$mysqli = mysqli_connect("example.com", "user", "password");
mysqli_select_db("test");
$res = mysqli_query("SELECT ' for new developments.' AS _msg FROM DUAL", $mysqli);
$row = mysqli_fetch_assoc($res);
echo $row['_msg'];
?>
```

# Resurse MySQL – mysql

## Structura

Index intern	Col 1 (tip date)	Col 2 (tip date)	....
1			
2			
...			

## Date

Index intern	Col 1	Col 2	....
1	Val 11	Val 12	...
2	Val 21	Val 22	...
...	...	...	...

## Metode

Constructor	query	fetch_assoc	....
-------------	-------	-------------	------

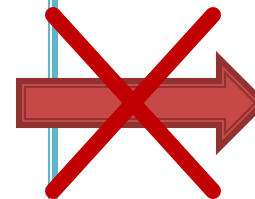
Functii de acces la structura



Functii de acces la date



Acces direct



Metode atasate resursei



# Conversia la mysqli (obligatorie)

## ■ exemplul anterior

```
$hostname = "localhost";
$database = "tmpaw";
$username = "web";
$password = "test";
$conex= mysql_connect($hostname, $username, $password);
mysql_select_db($database, $conex);

$query = "SELECT p.*, c.`nume` AS `nume_categ` FROM `produse` AS p
        LEFT JOIN `categorii` AS c ON (c.`id_categ` = p.`id_categ`)";
$result = mysql_query($query, $conex) or die(mysql_error());
$row_result = mysql_fetch_assoc($result);
$totalRows_result = mysql_num_rows($result);

do{
    $produse[$row_result['nume_categ']][$row_result['nume']]=array ("descr" => $row_result['detalii'], "pret"
=> $row_result['pret'], "cant" => $row_result['cant']);
}
while ($row_result = mysql_fetch_assoc($result));
```



# mysqli (Procedural)


```
//$conex= mysql_connect($hostname, $username, $password);
//mysql_select_db($database, $conex);
$conex = mysqli_connect($hostname, $username, $password, $database);

$query = "SELECT p.*, c.`nume` AS `nume_categ` FROM `produse` AS p
        LEFT JOIN `categorii` AS c ON (c.`id_categ` = p.`id_categ`)";
//$result = mysql_query($query, $conex) or die(mysql_error());
$result = mysqli_query($conex, $query);

//$row_result = mysql_fetch_assoc($result);
$row_result = mysqli_fetch_assoc($result);

//$totalRows_result = mysql_num_rows($result);
$totalRows_result = mysqli_num_rows($result);

do {
    $produse[$row_result['nume_categ']][$row_result['nume']] = array ("descr" => $row_result['detalii'], "pret"
=> $row_result['pret'], "cant" => $row_result['cant']);
}
//while ($row_result = mysql_fetch_assoc($result));
while ($row_result = mysqli_fetch_assoc($result));
```



# mysqli (POO)

```
//$conex= mysql_connect($hostname, $username, $password);
//mysql_select_db($database, $conex);
//$conex = mysqli_connect($hostname, $username, $password, $database);
$conex = new mysqli($hostname, $username, $password, $database);

$query = "SELECT p.*, c.`nume` AS `nume_categ` FROM `produse` AS p
        LEFT JOIN `categorii` AS c ON (c.`id_categ` = p.`id_categ`)";
//$result = mysql_query($query, $conex) or die(mysql_error());
//$result = mysqli_query($conex, $query);
$result = $conex->query( $query );

//$row_result = mysql_fetch_assoc($result);
//$row_result = mysqli_fetch_assoc($result);
$row_result = $result->fetch_assoc();

//$totalRows_result = mysql_num_rows($result);
//$totalRows_result = mysqli_num_rows($result);
$totalRows_result = $result->num_rows;

do {
    $produse[$row_result['nume_categ']][$row_result['nume']]=array ("descr" => $row_result['detalii'], "pret"
=> $row_result['pret'], "cant" => $row_result['cant']);
}
//while ($row_result = mysql_fetch_assoc($result));
while ($row_result = $result->fetch_assoc();)
```

# Laborator 6

# Plan aplicatie – Cumparator

- Pe masura ce aplicatia paraseste un fir liniar de executie este necesara introducerea unui plan (graf) al aplicatiei
- Cumparator
  - citirea fisierului text (XML) se realizeaza in antet.php, comun pentru toate fisierele

lista\_categ.php  
ALEGERE CATEGORIE

formular.php  
INTRODUCERE DATE

rezultat.php  
PRELUCRARE  
COMANDA

# Rezultat (comparator)

**Categorii Produse**

Alegeti categoria:

Nr.	Categorie	Total Produse
1	<a href="#">Papetarie</a>	3
2	<a href="#">Instrumente</a>	3
3	<a href="#">Audio-video</a>	3
4	<a href="#">Calculatoare</a>	3
5	<a href="#">Jucarii</a>	2

Total produse: 14

**Magazin online Firma X SRL**

Finalizati comanda

Nr.	Produs	Pret	Cantitate
1	Carti	100	<input type="text" value="1"/>
2	Caiete	50	<input type="text" value="2"/>
3	Penare	150	<input type="text" value="1"/>
4	Stilouri	125	<input type="text" value="0"/>
5	Creioane	25	<input type="text" value="0"/>

**Magazin online Firma X SRL**

**Rezultate comanda**

Pret total (fara TVA): 350

Pret total (cu TVA): 416.5

Comanda receptionata la data: 17/03/2010 ora 08:24

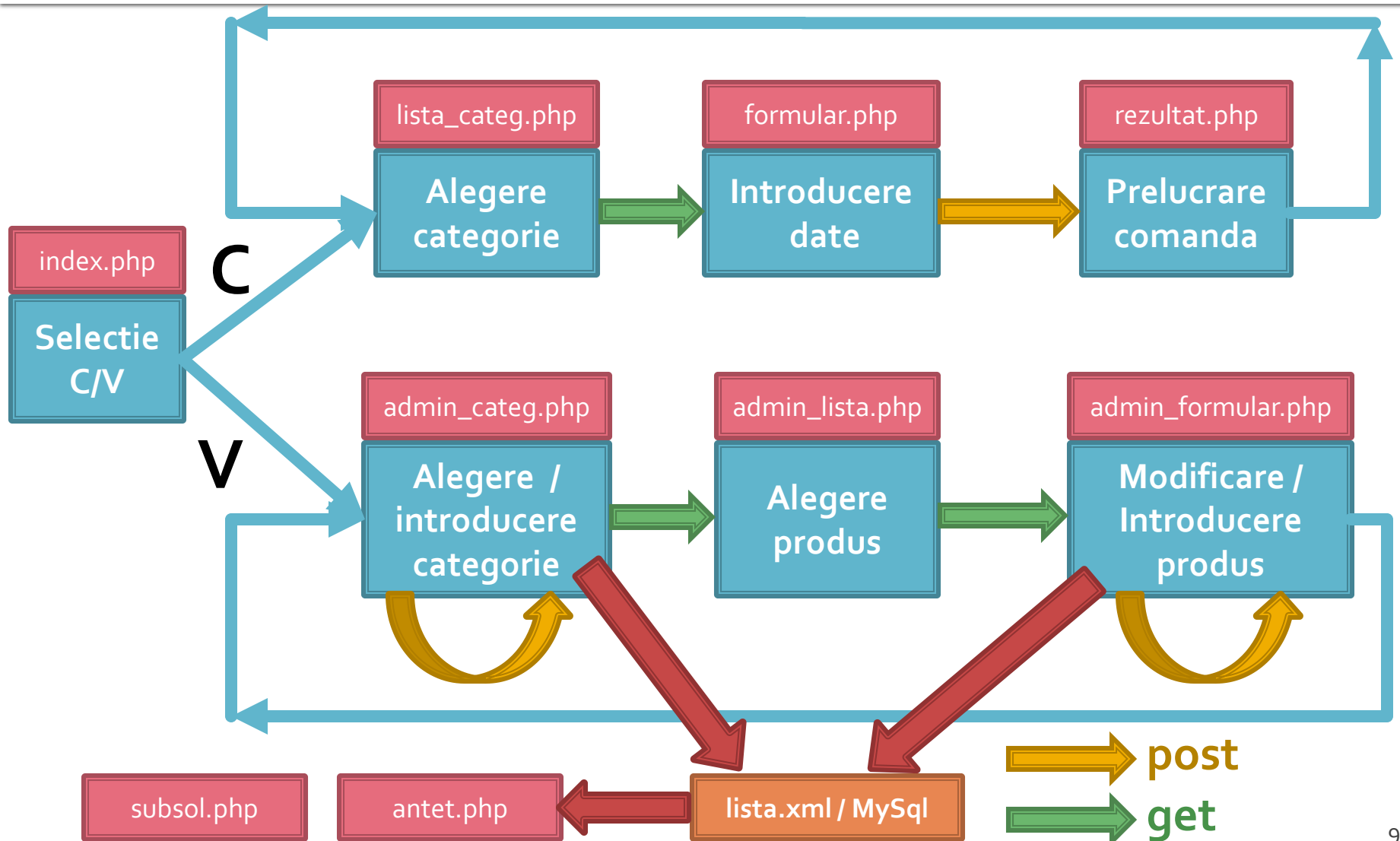




# Plan aplicatie – Vanzator

- Aparitia aplicatiei pentru vanzator
  - introduce un fir paralel de executie cu necesitatea alegerii initiale: cumparator/vanzator
  - aduce posibilitatea scrierii fisierului XML
  - diverse operatii de scriere
    - introducere categorie de produse
    - introducere produs nou intr-o categorie existenta
    - modificare produs existent
  - modificarea fisierului implica 2 actiuni:
    - colectare date
    - prelucrare

# Plan aplicatie (Proiect !!)



# Rezultat (vanzator)

**Magazin** Firma X

[Inceput](#) | [Inapoi](#)

## Magazin online Firma X SRL

Alegeti:

- [Cumparator](#)
- [Vanzator](#)

### Categorii Produse

Alegeti categoria:

Nr.	Categorie	Total Produse
1	<a href="#">Papetarie</a>	3
2	<a href="#">Instrumente</a>	3
3	<a href="#">Audio-video</a>	3
4	<a href="#">Calculatoare</a>	3
5	<a href="#">Jucarii</a>	2

Total produse: 14

Categorie noua de produse:

### Lista produse in categoria Calculatoare

Nr.	Produs	Descriere	Pret	Cantitate	Actiuni
1	Laptop	calculator mic	2000	2	<a href="#">modifica</a>
2	Desktop	calculator mare	1000	5	<a href="#">modifica</a>
3	Imprimanta	prn	200	2	<a href="#">modifica</a>
-	Produs nou				<a href="#">adauga</a>

### Produs in categoria Calculatoare

Produs	<input type="text" value="laptop"/>
Descriere	<input type="text" value="calculator mic"/>
Pret	<input type="text" value="2000"/>
Cantitate	<input type="text" value="2"/>



# Laborator 6

- Sa se continue magazinul virtual cu:
  - produsele sunt grupate pe categorii de produse
  - sa prezinte utilizatorului o lista de grupe de produse pentru a alege
  - sa prezinte utilizatorului o lista de produse si preturi in grupa aleasa
  - lista de produse si preturi se citeste dintr-o baza de date **MySQL**
  - se preia comanda si se calculeaza suma totala
  - **se creaza paginile prin care vanzatorul poate modifica preturile, produsele, categoriile**

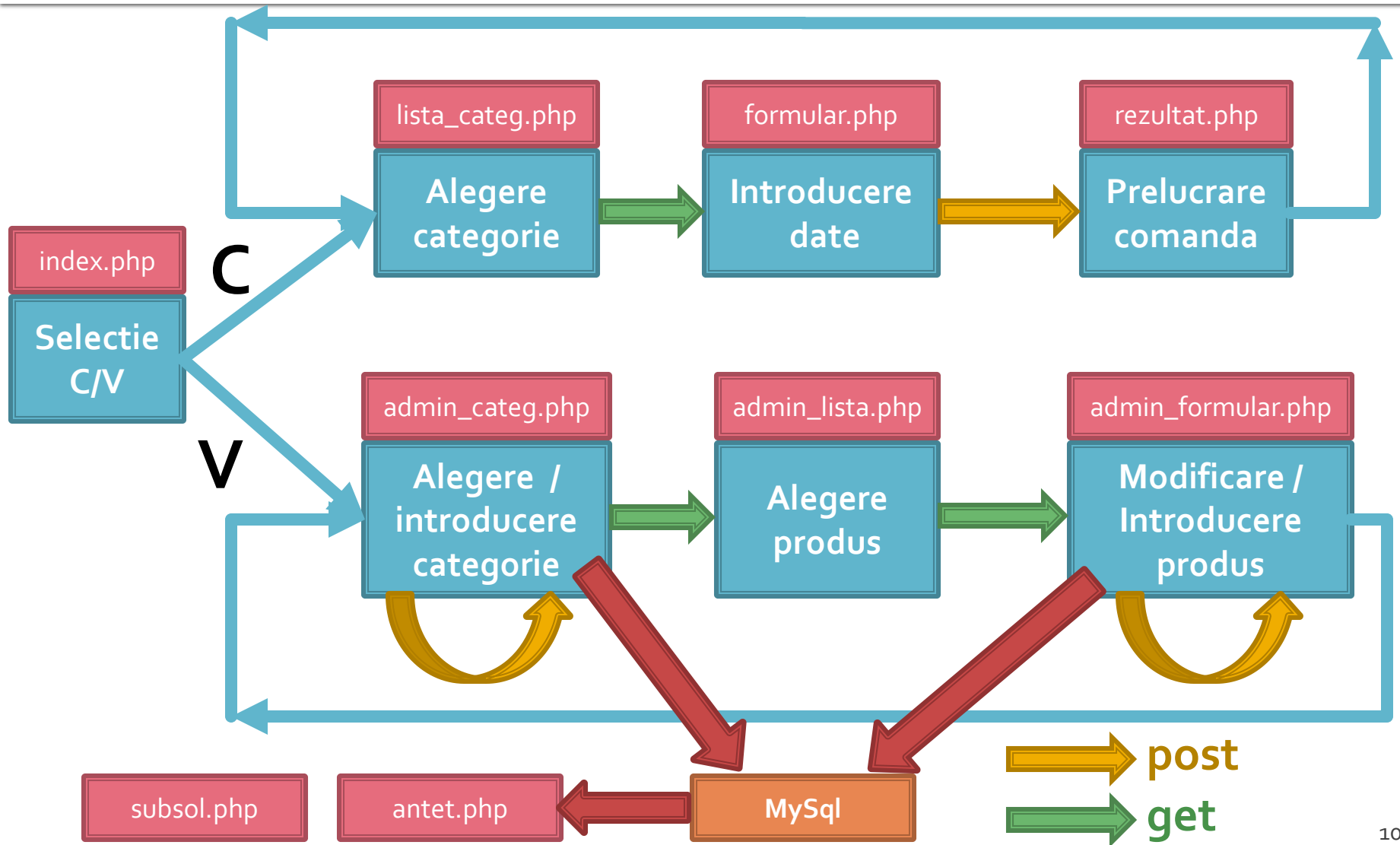
# Laborator 6/7 – Mod de lucru

- Se continua lucrul la aplicatie (L5)
- Se recomanda laboratorul **asincron** – S2
- Se poate folosi fisierul cu surse cpypaste.txt  
(site-<http://rf-opto.etti.tuiasi.ro>)

# Laborator 6/7 – Mod de lucru

- Se ia o decizie relativ la relatia dintre produse si categorii (S50-S53)
  - One to Many
  - Many to Many
- Se creaza cele 2(3) tabele corespunzatoare
- Se populeaza cu date
- Se actualizeaza planul aplicatiei pentru a corespunde cu aplicatia proprie
  - nume de fisiere, tipuri de transfer a datelor

# Plan aplicatie



# Rezultat (comparator)

**Categorii Produse**

Alegeti categoria:

Nr.	Categorie	Total Produse
1	<a href="#">Papetarie</a>	3
2	<a href="#">Instrumente</a>	3
3	<a href="#">Audio-video</a>	3
4	<a href="#">Calculatoare</a>	3
5	<a href="#">Jucarii</a>	2

Total produse: 14

**Magazin online Firma X SRL**

Finalizati comanda

Nr.	Produs	Pret	Cantitate
1	Carti	100	<input type="text" value="1"/>
2	Caiete	50	<input type="text" value="2"/>
3	Penare	150	<input type="text" value="1"/>
4	Stilouri	125	<input type="text" value="0"/>
5	Creioane	25	<input type="text" value="0"/>

**Magazin online Firma X SRL**

**Rezultate comanda**

Pret total (fara TVA): 350

Pret total (cu TVA): 416.5

Comanda receptionata la data: 17/03/2010 ora 08:24





# Rezultat (vanzator)

**Magazin** Firma X

[Inceput](#) | [Inapoi](#)

## Magazin online Firma X SRL

Alegeti:

- [Cumparator](#)
- [Vanzator](#)

### Categorii Produse

Alegeti categoria:

Nr.	Categorie	Total Produse
1	<a href="#">Papetarie</a>	3
2	<a href="#">Instrumente</a>	3
3	<a href="#">Audio-video</a>	3
4	<a href="#">Calculatoare</a>	3
5	<a href="#">Jucarii</a>	2

Total produse: 14

Categorie noua de produse:

### Lista produse in categoria Calculatoare

Nr.	Produs	Descriere	Pret	Cantitate	Actiuni
1	Laptop	calculator mic	2000	2	<a href="#">modifica</a>
2	Desktop	calculator mare	1000	5	<a href="#">modifica</a>
3	Imprimanta	prn	200	2	<a href="#">modifica</a>
-	Produs nou				<a href="#">adauga</a>

### Produs in categoria Calculatoare

Produs	<input type="text" value="laptop"/>
Descriere	<input type="text" value="calculator mic"/>
Pret	<input type="text" value="2000"/>
Cantitate	<input type="text" value="2"/>



# Laborator 6/7 – Mod de lucru

- Se creaza firul de executie paralel pentru vanzator
  - fisierele pentru cumparator reprezinta o buna cale de pornire (Save As, Copy/Paste) pentru 2 din cele 3 fisiere
- Se lucreaza cat mai mult la conversia text -> MySQL
  - activitatea se continua la laboratorul 7

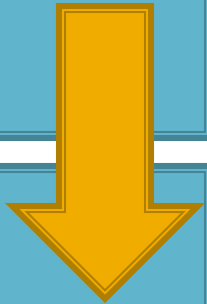
# Plan aplicatie – vanzator

- Deoarece citirea datelor se face in fisierul antet.php (modificat anterior) vor aparea modificari doar la nivelul scrierii datelor noi introduse
- Fisiere
  - admin\_lista.php – nemodificat
  - admin\_categ.php – scrie categorii noi in baza de date: se incuieste cod XML cu cod MySql
  - admin\_formular.php – scrie produse noi / corectii in baza de date: se incuieste cod XML cu cod MySql

# admin\_categ.php

```
if (isset($_POST["c_nou"]))
    {
        //categorii noua introdusa
        $categ_nou=$xml->addChild("categorii");
        $categ_nou->addAttribute("nume", $_POST["nou"]);
        $xml->asXML("lista.xml"); // salvare fisier
        $produse[$_POST["nou"]]=array(); // update matrice produse
        echo "<p>Categorii ".$_POST["nou"]." adaugata!</p>";
    }
```

```
if (isset($_POST["c_nou"]))
    {
        //categorii noua introdusa
        $query = "INSERT INTO `categorii` (`nume`, `detalii`) VALUES ('
".$_POST["nou_nume"]."`, '".$_POST["nou_desc"]."')";
        echo $query; //util in perioada de testare
        $result = mysql_query($query, $conex) or die(mysql_error());
        $record=mysql_insert_id(); //obtinerea id-ului nou
        $produse[$_POST["nou_nume"]]=array(); // update matrice produse
        echo "<p>Categorii ".$_POST["nou_nume"]." adaugata! Are id = ".$record."</p>";
    }
```



# admin\_categ.php



## Magazin online Firma X SRL

INSERT INTO `categorii` (`nume`, `detalii`) VALUES ('jucarii', 'pentru copii')

Categoria jucarii adaugata! Are id = 4

# admin\_formular.php

- Pentru inlocuire/adaugare produs apare o tratare diferita a celor doua situatii:
  - Adaugarea de produs face apel la interogarea SQL `INSERT INTO `produse` ...`
  - Modificarea unui produs existent va face apel la interogarea SQL `UPDATE `produse` SET ...`

# admin\_formular.php

```
if (isset($_POST["prod_ant"]))//exista deja acest produs anterior?
    //exista deja acest produs UPDATE
    unset($produse[$_POST['categ']][$_POST['prod_ant']]);//trebuie sters produsul anterior inlocuit
    $query = "UPDATE `produse` SET `nume`='".$_POST["prod"]."', `detalii`='".$_POST["descriere"]."',
`cant`='".$_POST["cantitate"]."', `pret`='".$_POST["pret"]."' WHERE `nume`='".$_POST["prod_ant"].'";
    echo $query;//util in perioada de testare
    $result = mysql_query($query, $conex) or die(mysql_error());
    echo "<p>Produsul '".$_POST["prod"]."' modificat in categoria '".$_POST['categ']."'!</p>";
}

else
    //NU exista acest produs INSERT
    $query = "INSERT INTO `produse` (`nume`, `detalii`, `pret`, `cant`, `id_categ`) VALUES
('".$_POST["prod"]."', '".$_POST["descriere"]."', '".$_POST["pret"]."', '".$_POST["cantitate"]."',
(SELECT `id_categ` FROM categorii WHERE `nume` = '".$_POST['categ'].')";
    echo $query;//util in perioada de testare
    $result = mysql_query($query, $conex) or die(mysql_error());
    $record=mysql_insert_id();//obtinerea id-ului nou
    echo "<p>Produsul '".$_POST["prod"]."' adaugat in categoria '".$_POST['categ']."'! Are id =
".$_record."</p>";
}

$produse[$_POST['categ']][$_POST['prod']]=array("descr" => $_POST['descriere'], "pret" => $_POST['pret'], "cant" =>
$_POST['cantitate']);
```

# Faza de verificare/depanare

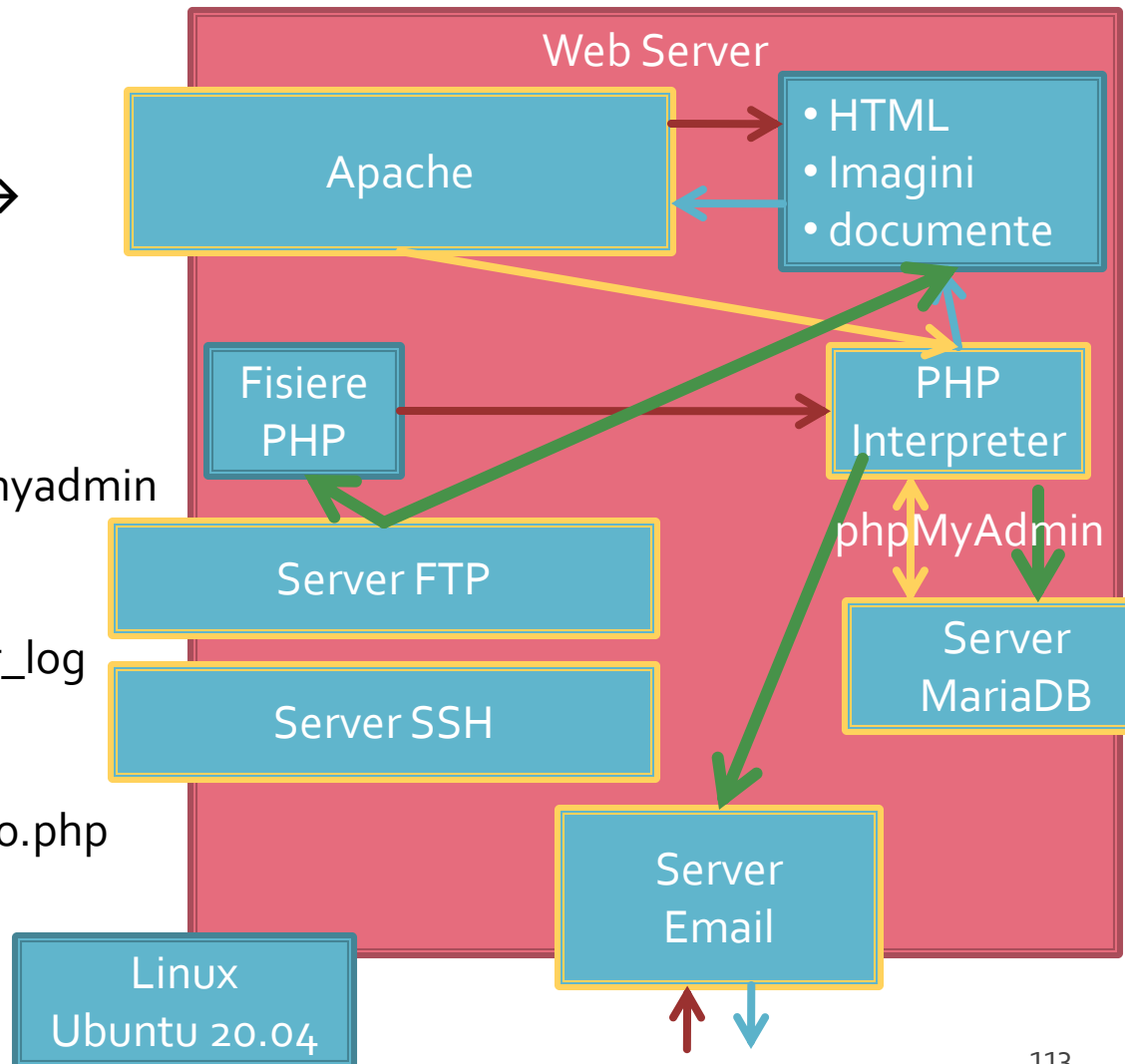
- Se recomanda utilizarea posibilitatii vizualizarii matricilor
  - In fisierul care receptioneaza datele
  - temporar pina la definitivarea codului
- utilizarea de cod "verbose" (manual) in etapele initiale de scriere a surselor PHP poate fi extinsa si la alte tipuri de date
  - singura (aproape) metoda de depanare(debug) in PHP
  - `<p>temp <?php echo "a=";echo $a; ?> </p>`

```
echo "<pre>";  
print_r($_POST);  
echo "</pre>";
```



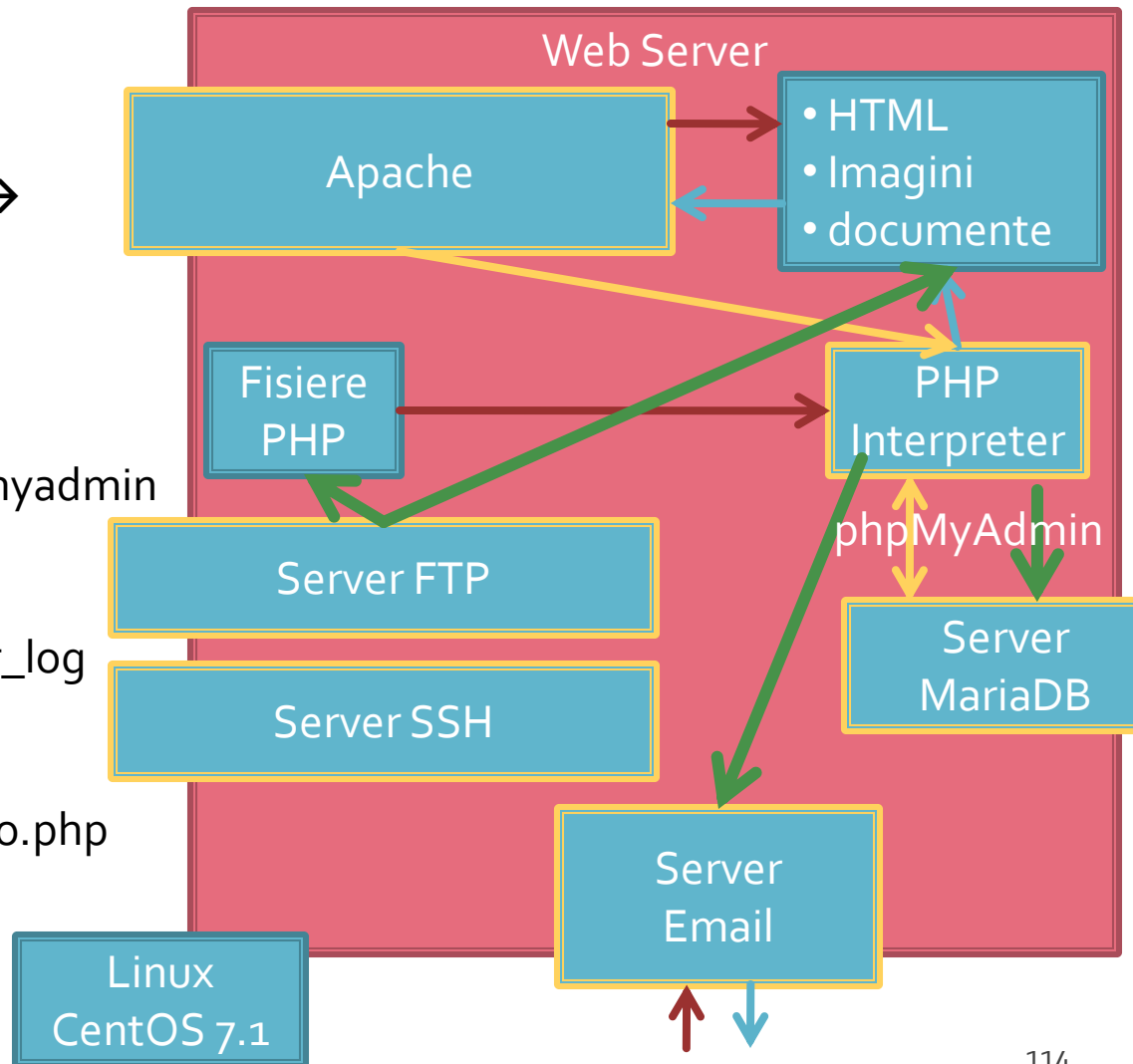
# Utilizzare LAMP

1. login → **paw**:masteretti
2. ifconfig → 192.168.30.5
3. putty.exe → 192.168.30.5 → SSH → **paw**:masteretti (remote login)
4. [alte comenzi linux dorite]
5. FTP → Winscp → SFTP → student:masterrc@192.168.30.5
6. MySql → http://192.168.30.5/phpmyadmin → **root**:masteretti
7. Apache Error Log →
  - 7a. putty → nano /var/log/httpd/error\_log
  - 7b. http://192.168.30.5/logfile.php (nonstandard)
8. PHP info → http://192.168.30.5/info.php

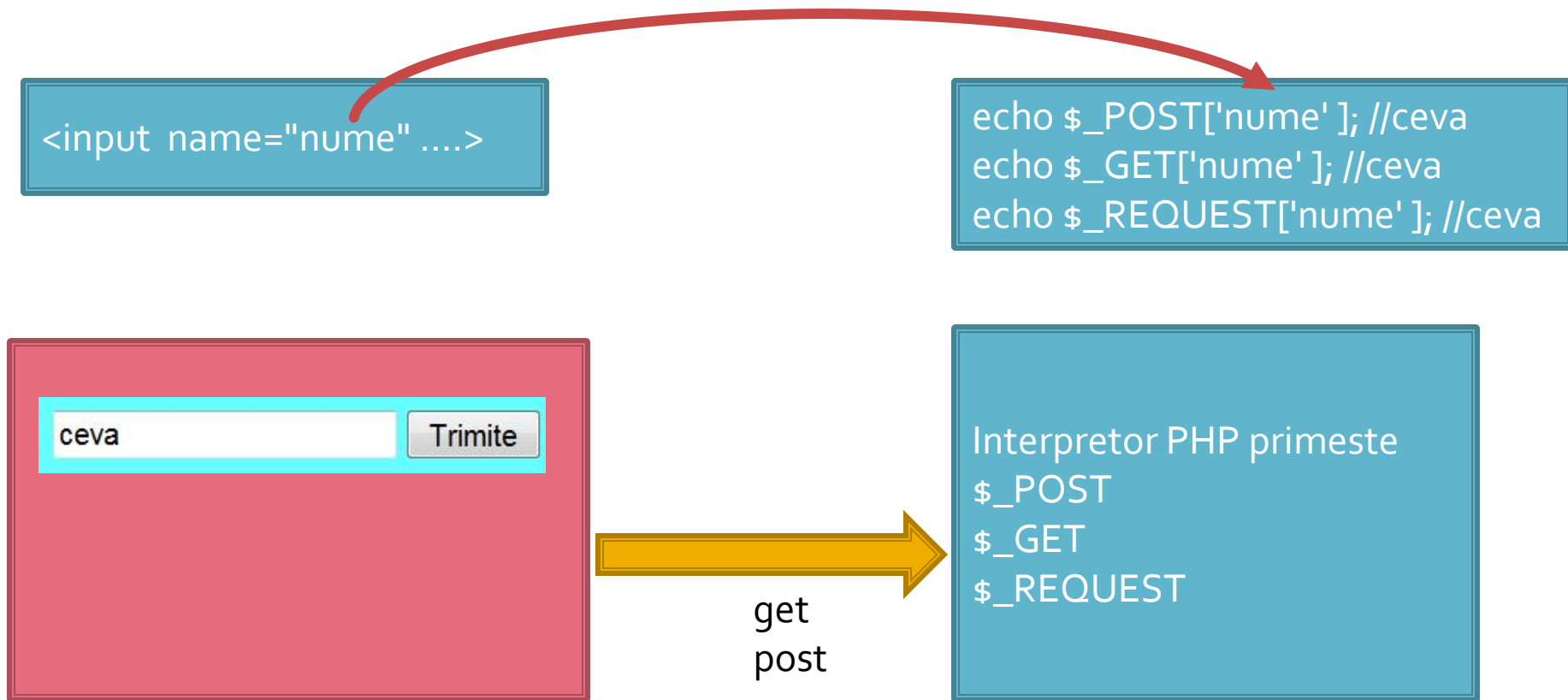


# Utilizare LAMP

1. login → root:masterrc
2. ifconfig → 192.168.30.5
3. putty.exe → 192.168.30.5 → SSH → root:masterrc (remote login)
4. [alte comenzi linux dorite]
5. FTP → Winscp → SFTP → student:masterrc@192.168.30.5
6. MySql → http://192.168.30.5/phpmyadmin → root:masterrc
7. Apache Error Log →
  - 7a. putty → nano /var/log/httpd/error\_log
  - 7b. http://192.168.30.5/logfile.php (nonstandard)
8. PHP info → http://192.168.30.5/info.php



# Client / Server



# Depanare

```
echo "<pre>";  
print_r($_POST);  
echo "</pre>";
```

```
<p>temp <?php echo  
"a=";echo $a; ?> </p>
```

MySQL – Server Windows 2000

# **Mini – Indrumar practic**

## **Lucru cu bazele de date**

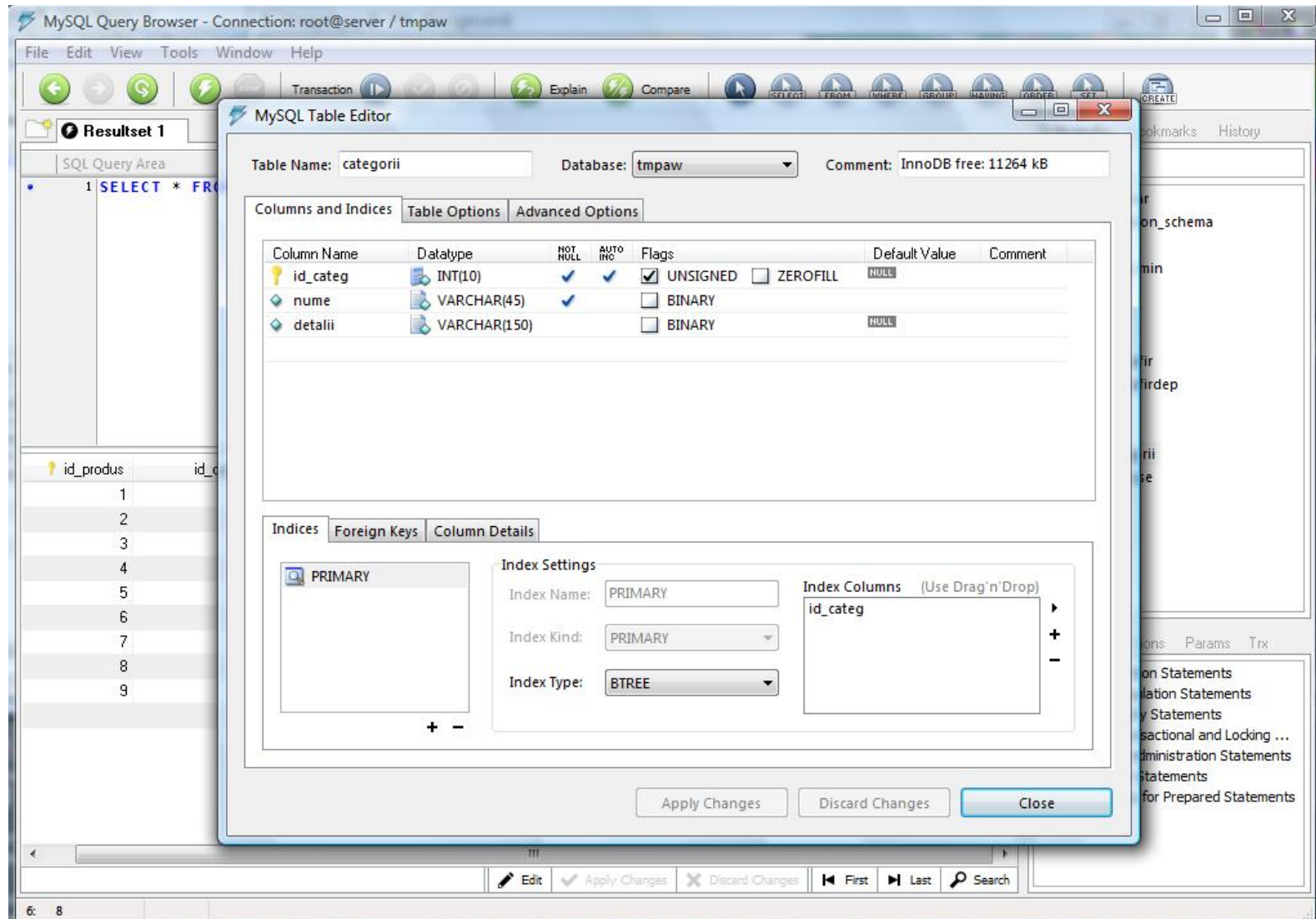
# Realizarea bazei de date

- Se recomanda utilizarea utilitarului **MySQL Query Browser** sau un altul echivalent pentru crearea scheletului de baza de date (detalii – laborator 1)
- Se initializeaza aplicatia cu drepturi depline (“root” si parola)
  - se creaza o noua baza de date:
    - in lista “Schemata” – Right click – Create New Schema
  - se activeaza ca baza de date curenta noua “schema” – Dublu click pe numele ales

# Introducere tabele

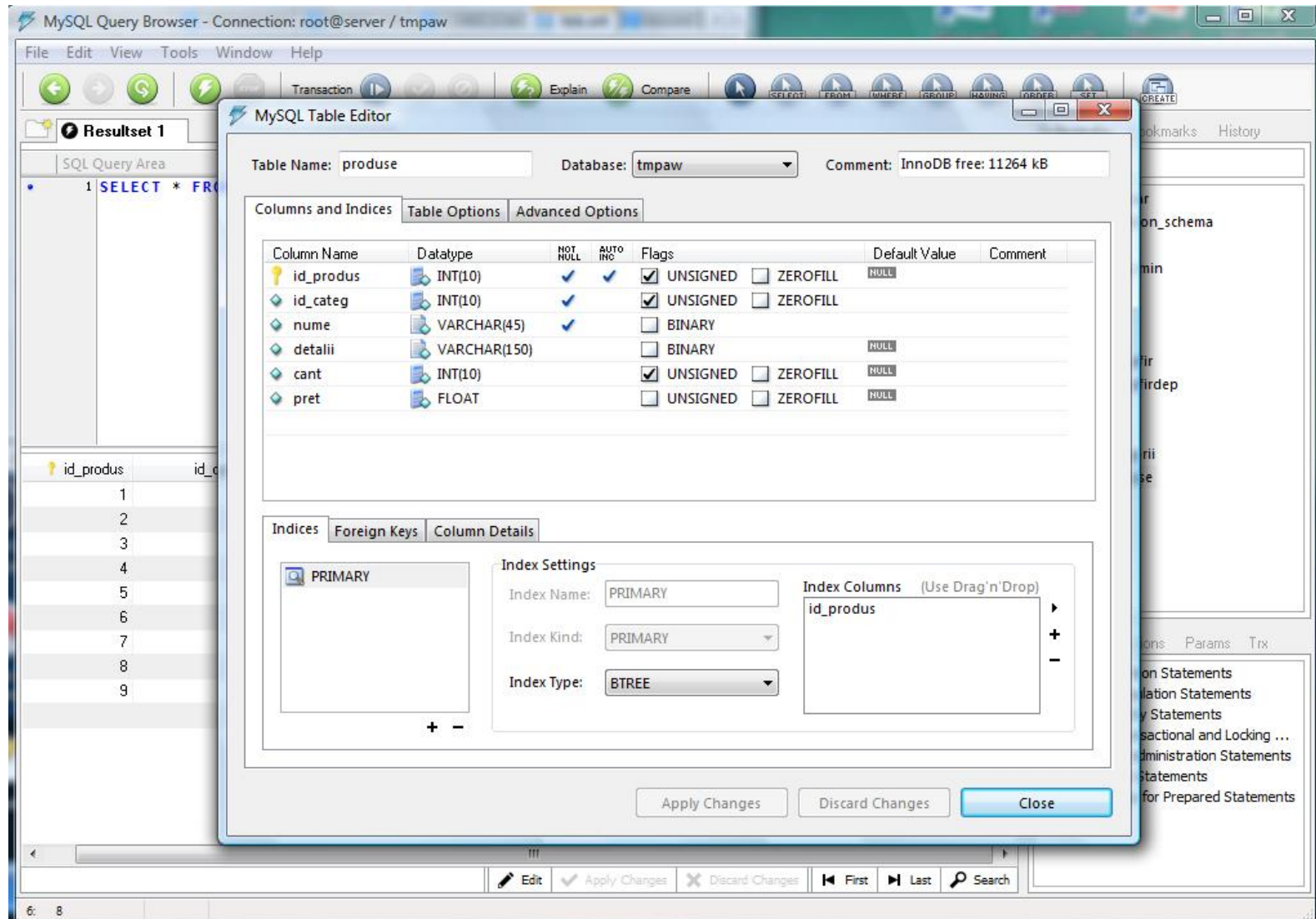
- Introducere tabel – Click dreapta pe numele bazei de date aleasa – Create New Table
- se defineste structura tabelului
  - nume coloane
  - tip de date
  - NOT NULL – daca se accepta ca acea coloana sa ramana fara date (NULL) sau nu
  - AUTOINC – daca acea coloana va fi de tip intreg si va fi incrementata automat de server (util pentru crearea cheilor primare)
  - Default value – valoarea implicita care va fi inserata daca la introducerea unei linii noi nu se mentioneaza valoare pentru acea coloana (legat de optiunea NOT NULL)

# Tabel Categorii




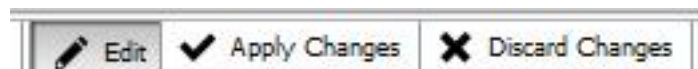


# Tabel Prognose



# Introducere date initiale

- Dublu click pe tabel → In zona "SQL Query Area" se completeaza interogarea de selectie totala
  - `SELECT * FROM produse p;`
- Executia interogarii SQL
  - Meniu → Query → Execute
  - Bara de butoane 
- Lista rezultata
  - initial vida
  - poate fi editata – butoanele "Edit", "Apply Changes", "Discard Changes" din partea de jos a listei



# Introdúcere date initiale

MySQL Query Browser - Connection: root@server / tmpaw

File Edit View Query Script Tools Window Help

Transaction Explain Compare SELECT FROM WHERE GROUP HAVING ORDER SET CREATE

Resultset 1

SQL Query Area

```
1 SELECT * FROM produse p;
```

id_produs	id_categ	nume	detalii	cant	pret
1	1	carte	mai multe pagini scrise legate	0	100
2	1	caiet	mai multe pagini goale legate	0	75
3	1	hartie scris	mai multe pagini goale NElegate	0	50
4	2	penar	loc de depozitat instrumente de scris	0	150
5	2	stilou	instrument de scris albastru	0	125
6	2	creion	instrument de scris gri	0	25
ALL	3	cd	canta	0	50
ALL	3	dvd	vizual	0	100
ALL	3	blue ray	vizual extrem	0	500

Schemata Bookmarks History

tmpaw

- categoriai
- produse
- world

Syntax Functions Params Trx

- Data Definition Statements
- Data Manipulation Statements
- MySQL Utility Statements
- MySQL Transactional and Locking ...
- Database Administration Statements
- Replication Statements
- SQL Syntax for Prepared Statements

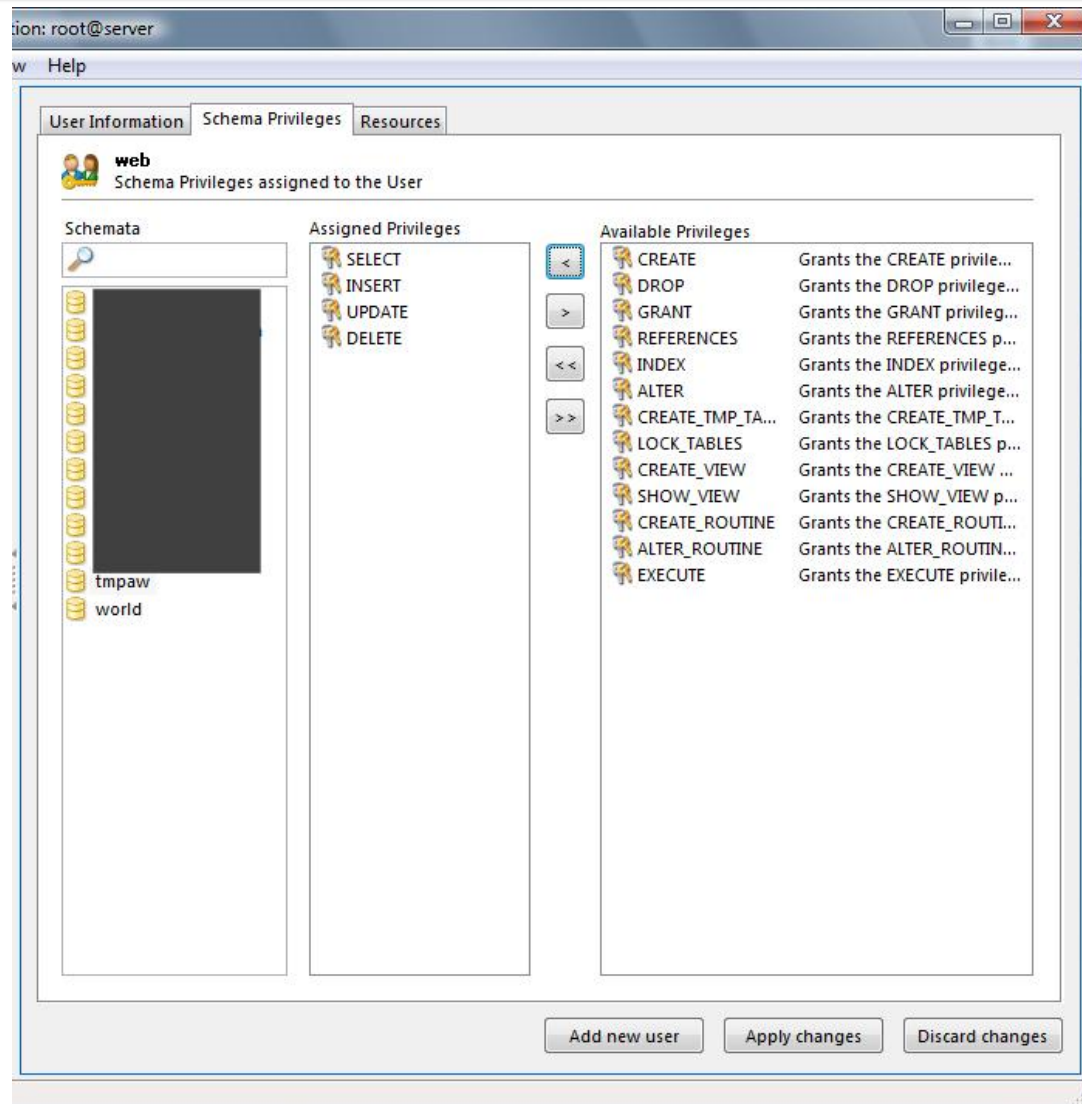
Edit Apply Changes Discard Changes First Last Search

6: 8

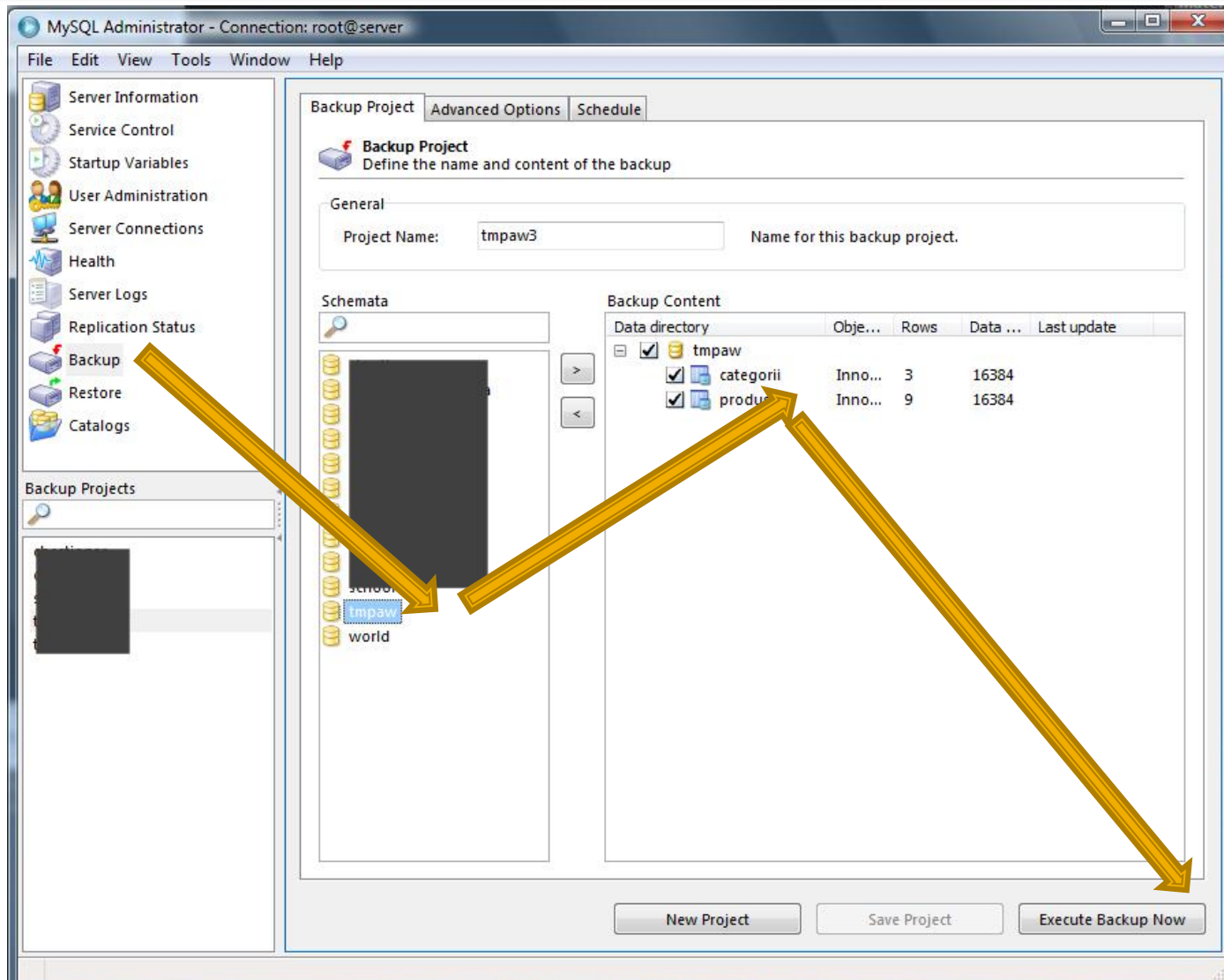
# Backup, Restore, drepturi de acces

- Se recomanda utilizarea utilitarului **MySQL Administrator** sau un altul echivalent (detalii – laborator 1)
- Se initializeaza aplicatia cu drepturi depline ("root" si parola)
- Se creaza un utilizator limitat (detalii – laborator 1)
- Se alocă drepturile "SELECT" + "INSERT" + "UPDATE" asupra bazei de date create (sau mai multe daca aplicatia o cere)


# Drepturi de acces



# Backup



# Restaurarea bazei de date

- Din **MySql Administrator**
  - Sectiunea Restore → "Open Backup File"
- Din **MySql Query Browser**
  - Meniu → File → Open Script
  - Executie script SQL
    - Meniu → Script → Execute
    - Bara de butoane 
- Scriptul SQL rezultat contine comenzile/interogariile SQL necesare pentru crearea bazei de date si popularea ei cu date

# Script SQL Backup - utilitate

- Poate fi folosit ca un model extrem de bun pentru comenzile necesare pentru crearea programatica (din PHP de exemplu) a bazei de date

```
CREATE DATABASE IF NOT EXISTS tmpaw;  
USE tmpaw;  
  
DROP TABLE IF EXISTS `categorii`;  
CREATE TABLE `categorii` (  
  `id_categ` int(10) unsigned NOT NULL auto_increment,  
  `nume` varchar(45) NOT NULL,  
  `detalii` varchar(150) default NULL,  
  PRIMARY KEY (`id_categ`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
INSERT INTO `categorii` (`id_categ`,`nume`,`detalii`) VALUES  
(1,'papetarie',NULL),  
(2,'instrumente',NULL),  
(3,'audio-video',NULL);
```



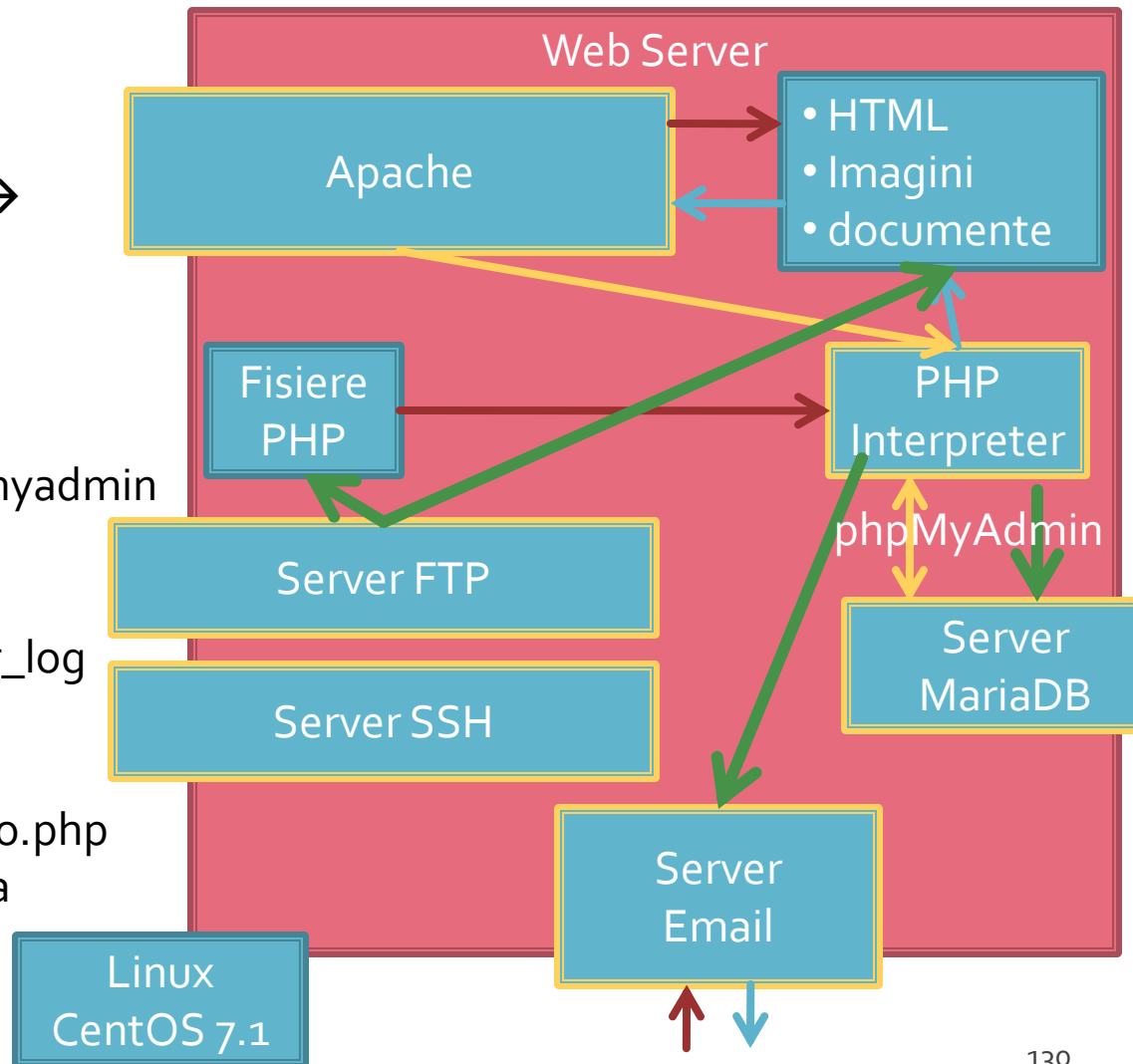
MySQL – Server Centos 7.1

# Mini – Indrumar practic

## Lucru cu bazele de date

# Utilizare LAMP

1. login → root:masterrc
2. ifconfig → 192.168.30.5
3. putty.exe → 192.168.30.5 → SSH → root:masterrc (remote login)
4. [alte comenzi linux dorite]
5. FTP → Winscp → SFTP → student:masterrc@192.168.30.5
6. MySql → http://192.168.30.5/phpmyadmin → root:masterrc
7. Apache Error Log →
  - 7a. putty → nano /var/log/httpd/error\_log
  - 7b. http://192.168.30.5/logfile.php (nonstandard)
8. PHP info → http://192.168.30.5/info.php
9. daca serviciul DHCP duce la oprirea Apache: service httpd restart



# PhpMyAdmin

- <http://192.168.30.5/phpmyadmin>
  - root
  - parola administrator **MySQL/MariaDB** (masterrc)



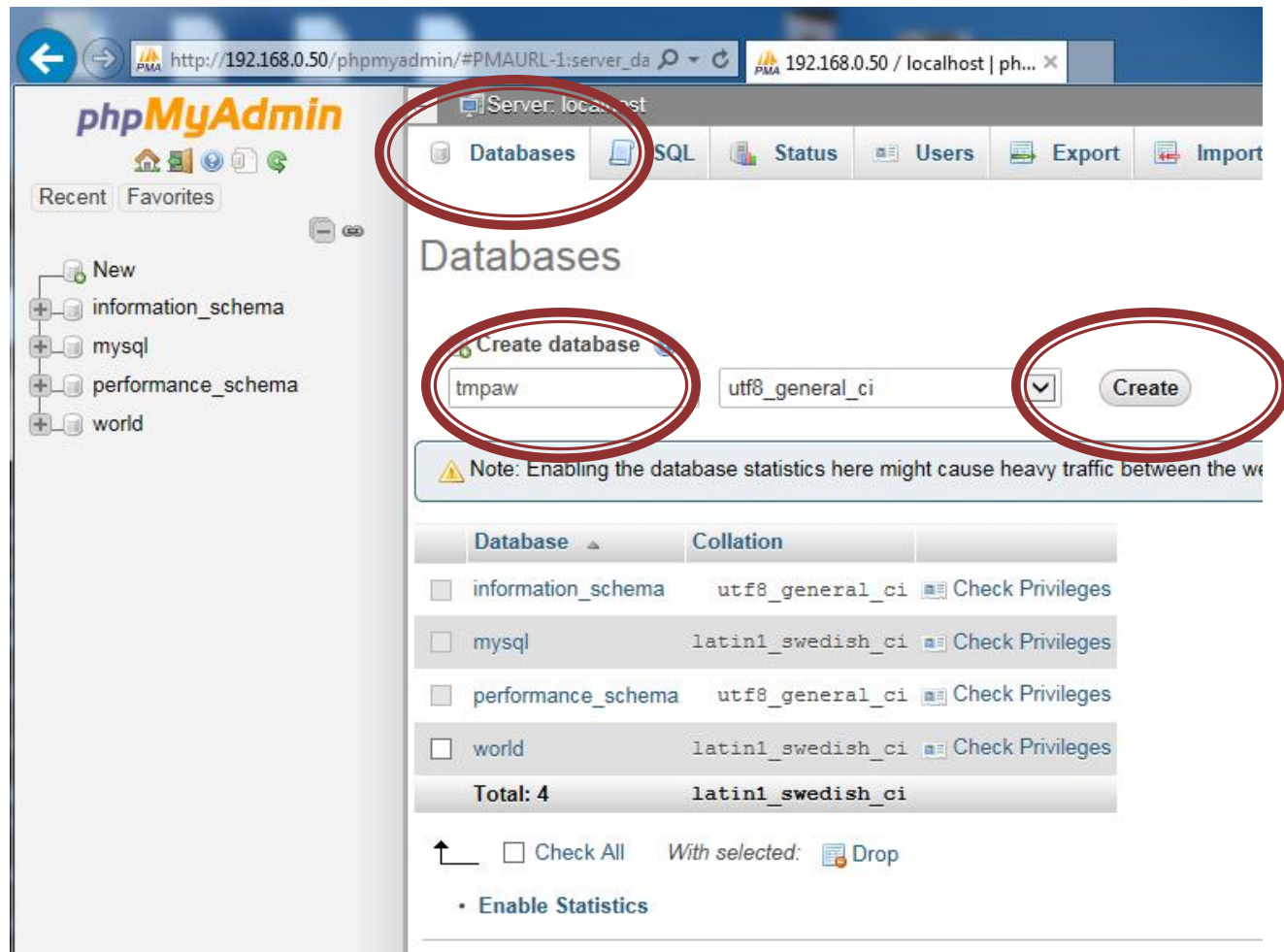
# PhpMyAdmin

The screenshot displays the PhpMyAdmin web interface in a browser window. The address bar shows the URL `http://192.168.0.50/phpmyadmin/#PMAURL-0:index.php`. The interface includes a top navigation bar with tabs for Databases, SQL, Status, Users, Export, Import, Settings, Replication, Variables, Charsets, and Engines. The left sidebar shows a tree view of databases: information\_schema, mysql, performance\_schema, and world. The main content area is divided into several panels:

- General Settings:** Includes a "Change password" link and a "Server connection collation" dropdown set to `utf8mb4_unicode_ci`.
- Appearance Settings:** Includes a "Language" dropdown set to `English`, a "Theme" dropdown set to `pmahomme`, and a "Font size" dropdown set to `82%`. A "More settings" link is also present.
- Database server:** Lists server details:
  - Server: Localhost via UNIX socket
  - Server type: MariaDB
  - Server version: 5.5.44-MariaDB - MariaDB Server
  - Protocol version: 10
  - User: root@localhost
  - Server charset: UTF-8 Unicode (utf8)
- Web server:** Lists web server details:
  - Apache/2.4.6 (CentOS) OpenSSL/1.0.1e-fips mod\_fcgid/2.3.9
  - PHP/5.4.16 mod\_python/3.5.0- Python/2.7.5
  - Database client version: libmysql - 5.5.44-MariaDB
  - PHP extension: mysqli
  - PHP version: 5.4.16
- phpMyAdmin:** Lists version information and links:
  - Version information: 4.4.15.1
  - Documentation
  - Wiki
  - Official Homepage
  - Contribute
  - Get support
  - List of changes

# Creare Baza de Date

- Databases → "nume" → Create



The screenshot shows the phpMyAdmin web interface. The 'Databases' tab is selected and circled in red. Below it, the 'Create database' form is visible, with the database name 'tmpaw' entered in the text field, 'utf8\_general\_ci' selected in the collation dropdown, and the 'Create' button circled in red. A list of existing databases is shown below the form, including 'information\_schema', 'mysql', 'performance\_schema', and 'world'. The interface also includes a left sidebar with a tree view of databases and a top navigation bar with various tools like SQL, Status, Users, Export, and Import.

phpMyAdmin

Server: localhost

Databases SQL Status Users Export Import

Databases

Create database

tmpaw utf8\_general\_ci [v] Create

Note: Enabling the database statistics here might cause heavy traffic between the we

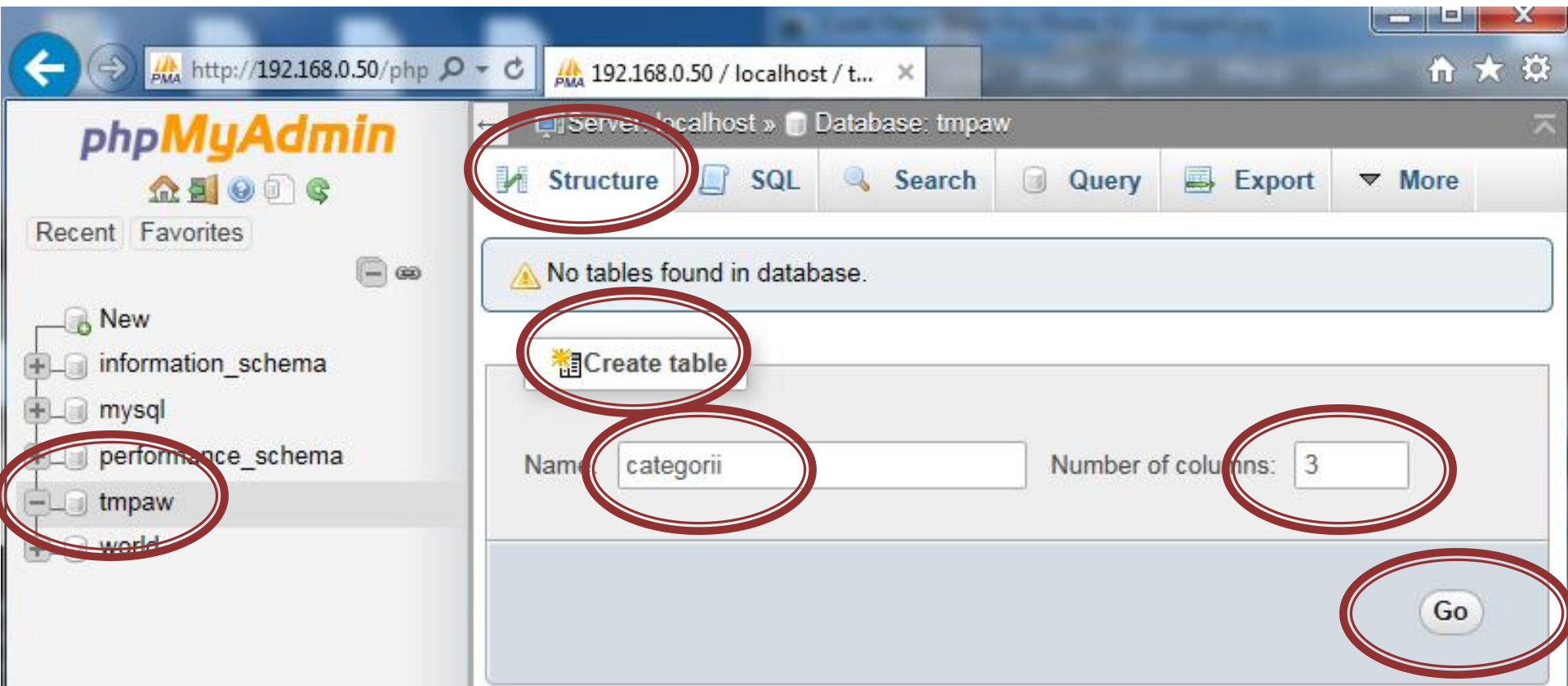
Database	Collation	
<input type="checkbox"/> information_schema	utf8_general_ci	<a href="#">Check Privileges</a>
<input type="checkbox"/> mysql	latin1_swedish_ci	<a href="#">Check Privileges</a>
<input type="checkbox"/> performance_schema	utf8_general_ci	<a href="#">Check Privileges</a>
<input type="checkbox"/> world	latin1_swedish_ci	<a href="#">Check Privileges</a>
Total: 4	latin1_swedish_ci	

↑ ☐ Check All With selected: [Drop](#)

• Enable Statistics

# Creare tabelle in baza de date

- Baza de date (in lista) → Structure → div Create Table → nume/coloane → Go



# Introducere coloane, tabel categorii

- (eventual ) Adaugare coloane / Stabilire nume
- Name / Type / Length / Default

The screenshot shows the phpMyAdmin interface for a table named 'categorii'. The table structure is as follows:

Name	Type	Length/Values	Default	Collation
id_categ	INT		None	
nume	VARCHAR	45	None	
detalii	VARCHAR	150	None	

The interface also shows the 'Table comments:' and 'Collation:' fields at the bottom.



# Introducere coloane

- (eventual) NOT NULL / Index / Auto Increment
  - in functie de “necesitatile” coloanei respective

Table name:  Add  column(s)

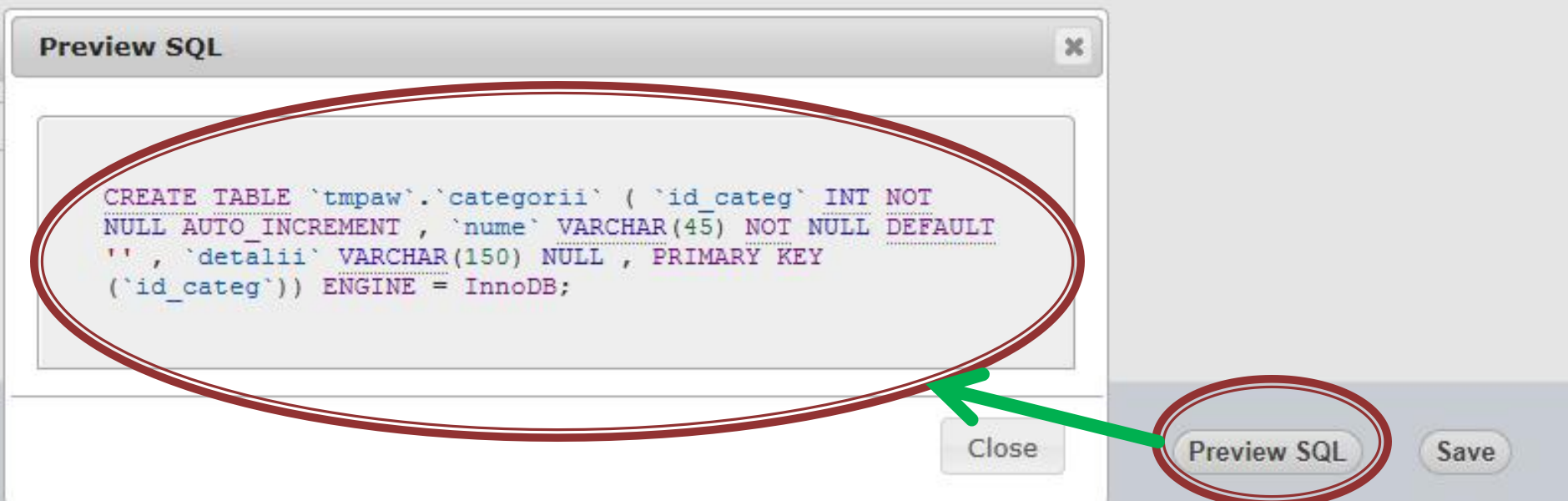
Structure

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I	Comments
id_categ	INT		None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
nume	VARCHAR	45	As defined:			<input type="checkbox"/>	---	<input type="checkbox"/>	
detalii	VARCHAR	150	None			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>	



# Preview SQL

- in aproape toate etapele in PhpMyAdmin
  - exemplu de cod SQL/schelet utilizabil (copy/paste) in aplicatia PHP
  - modificari de finete absente din interfata
    - copy → Sectiune "SQL" in interfata → paste → modificare



# Introducere coloane, tabel produse

- New → Nume → Add Columns → ...

The screenshot shows the phpMyAdmin interface. On the left, the 'Database: tmpaw' is selected in the sidebar. The main area displays the 'Add Columns' form for the table 'produse'. The 'Table name' field is 'produse', and the 'Add' button is highlighted. The 'Number of columns to add' is set to '1'. The 'Go' button is also highlighted. Below the form, the table structure is shown with columns: id\_produs, id\_categ, nume, detalii, cant, and pret. The 'Type' column for 'pret' is set to 'FLOAT'.

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I
id_produs	INT		None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
id_categ	INT		None			<input type="checkbox"/>	---	<input type="checkbox"/>
nume	VARCHAR	45	As defined:			<input type="checkbox"/>	---	<input type="checkbox"/>
detalii	VARCHAR	150	None			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>
cant	INT		None			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>
pret	FLOAT		None			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>

# Introducere date initiale (interfata)

- Tabel → Insert → Completare → Go

The screenshot shows the phpMyAdmin interface for a database named 'tmpaw'. The left sidebar shows the database structure, with 'categorii' selected under 'tmpaw'. The main area displays the 'Insert' form for the 'categorii' table. The table has three columns: 'id\_categ' (int(11)), 'nume' (varchar(45)), and 'detalii' (varchar(150)). The 'nume' field is filled with 'papetarie'. The 'Go' button is at the bottom right. Below the table, there are options to 'insert as new row' and 'Go back to previous page', and a 'Continue insertion with' dropdown set to '1' row.

Column	Type	Function	Null	Value
id_categ	int(11)			
nume	varchar(45)			papetarie
detalii	varchar(150)		<input checked="" type="checkbox"/>	

Continue insertion with  row(s)

# Vizualizare date existente

- Tabel → Browse → salt la pagina (numar de linii pe pagina)

The screenshot shows the phpMyAdmin interface. The left sidebar displays the database structure, with 'tmpaw' selected and 'categorii' highlighted. The main panel shows the 'Browse' tab for the 'categorii' table. The table structure is displayed, showing three rows of data. The 'id\_categ' column is highlighted, and the 'nume' and 'detalii' columns are also visible. The table data is as follows:

id_categ	nume	detalii
1	papetarie	NULL
2	instrumente	NULL
3	audio-video	NULL

# Introducere date initiale (SQL)

- Tabel → SQL → completare → Go

The screenshot shows the phpMyAdmin web interface. On the left sidebar, the 'produse' table under the 'tmpaw' database is selected. The top menu bar has the 'SQL' tab highlighted. The main area contains a text editor with an SQL INSERT query. Below the editor are buttons for 'SELECT \*', 'SELECT', 'INSERT', 'UPDATE', 'DELETE', 'Clear', and 'Format'. At the bottom right, there is a 'Go' button. The query text is as follows:

```
1 INSERT INTO `produse` (`id_produș`, `id_categ`, `nume`, `detalii`, `cant`, `pret`)
2 VALUES
3 (1,1,'carte','mai multe pagini scrise legate',0,100),
4 (2,1,'caiet','mai multe pagini goale legate',0,75),
5 (3,1,'hartie scris','mai multe pagini goale NElegate',0,50),
6 (4,2,'penar','loc de depozitat instrumente de scris',0,150),
7 (5,2,'stilou','instrument de scris albastru',0,125),
8 (6,2,'creion','instrument de scris gri',0,25),
9 (7,3,'cd','canta',0,50),
10 (8,3,'dvd','vizual',0,100),
11 (9,3,'blue ray','vizual extrem',0,500);|
```

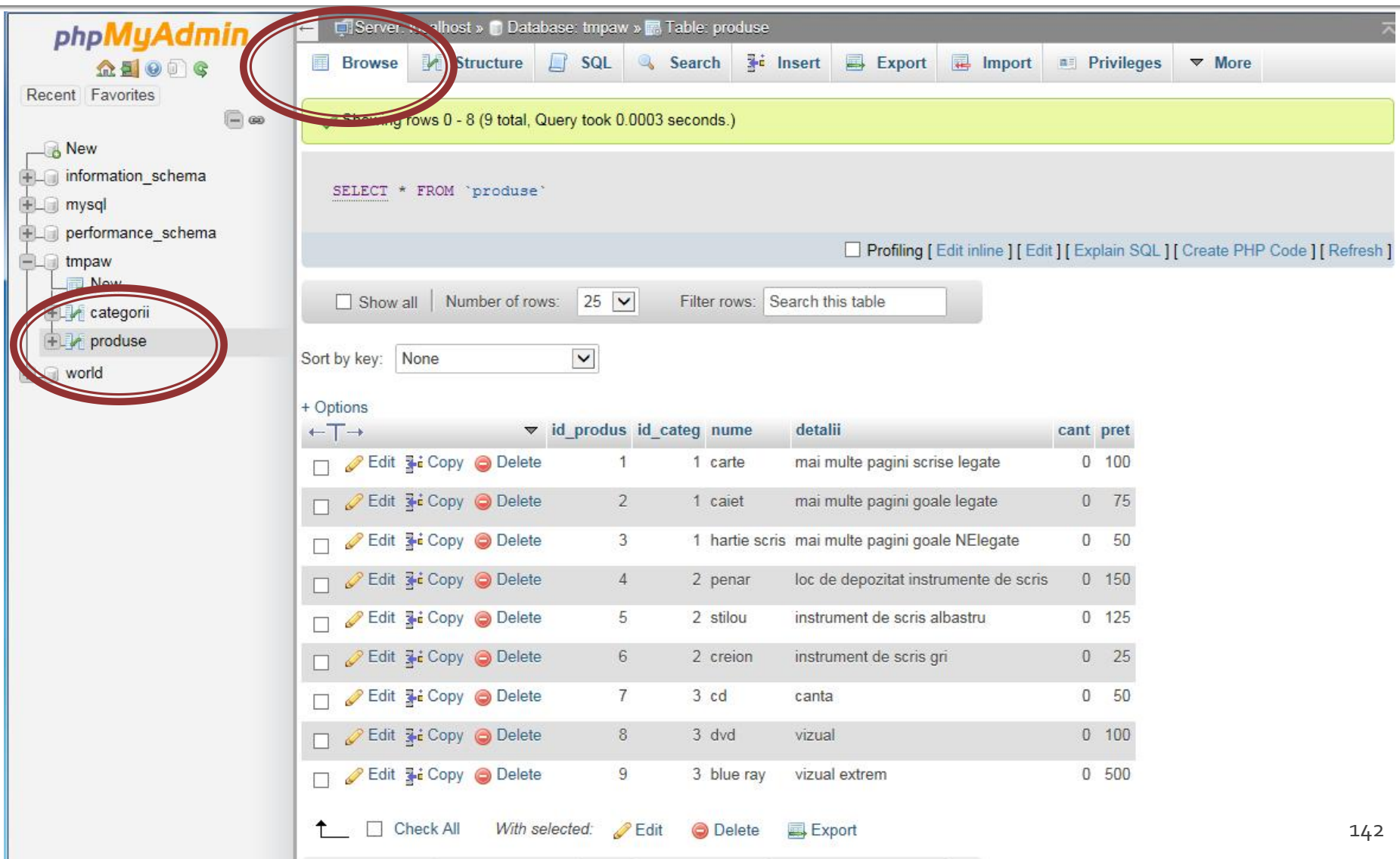
Columns listed on the right:

- id\_produș
- id\_categ
- nume
- detalii
- cant
- pret

At the bottom, the 'Go' button is circled in red.



# Tabel produse



phpMyAdmin

Server: localhost » Database: tmpaw » Table: produse

Structure SQL Search Insert Export Import Privileges More

Showing rows 0 - 8 (9 total, Query took 0.0003 seconds.)

```
SELECT * FROM `produse`
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP Code ] [ Refresh ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Sort by key: None

+ Options

				id_produs	id_categ	nume	detalii	cant	pret
<input type="checkbox"/>	Edit	Copy	Delete	1	1	carte	mai multe pagini scrise legate	0	100
<input type="checkbox"/>	Edit	Copy	Delete	2	1	caiet	mai multe pagini goale legate	0	75
<input type="checkbox"/>	Edit	Copy	Delete	3	1	hartie scris	mai multe pagini goale NElegate	0	50
<input type="checkbox"/>	Edit	Copy	Delete	4	2	penar	loc de depozitat instrumente de scris	0	150
<input type="checkbox"/>	Edit	Copy	Delete	5	2	stilou	instrument de scris albastru	0	125
<input type="checkbox"/>	Edit	Copy	Delete	6	2	creion	instrument de scris gri	0	25
<input type="checkbox"/>	Edit	Copy	Delete	7	3	cd	canta	0	50
<input type="checkbox"/>	Edit	Copy	Delete	8	3	dvd	vizual	0	100
<input type="checkbox"/>	Edit	Copy	Delete	9	3	blue ray	vizual extrem	0	500

☐ Check All With selected: Edit Delete Export

# Adaugare utilizator

- Server → Users → Add user

The screenshot shows the phpMyAdmin interface. The top navigation bar includes 'Server: localhost', 'Database: tmpaw', and 'Table: produse'. The 'Server: localhost' tab is circled in red. Below the navigation bar, the 'Users' tab is also circled in red. The 'Users overview' section displays a table of users. At the bottom, the 'New' button is circled in red, and the 'Add user' link is highlighted.

Server: localhost » Database: tmpaw » Table: produse

phpMyAdmin

Recent Favorites

New

- information\_schema
- mysql
- performance\_schema
- tmpaw
  - New
  - categoriai
  - produse
- world

Databases SQL Status Users Export Import Settings

### Users overview

	User name	Host	Password	Global privileges	Grant	Action
<input type="checkbox"/>	root	127.0.0.1	Yes	ALL PRIVILEGES	Yes	Edit Privileges Export
<input type="checkbox"/>	root	::1	Yes	ALL PRIVILEGES	Yes	Edit Privileges Export
<input type="checkbox"/>	root	localhost	Yes	ALL PRIVILEGES	Yes	Edit Privileges Export
<input type="checkbox"/>	root	tmpaw.etti	Yes	ALL PRIVILEGES	Yes	Edit Privileges Export
<input type="checkbox"/>	web	%	Yes	USAGE	No	Edit Privileges Export

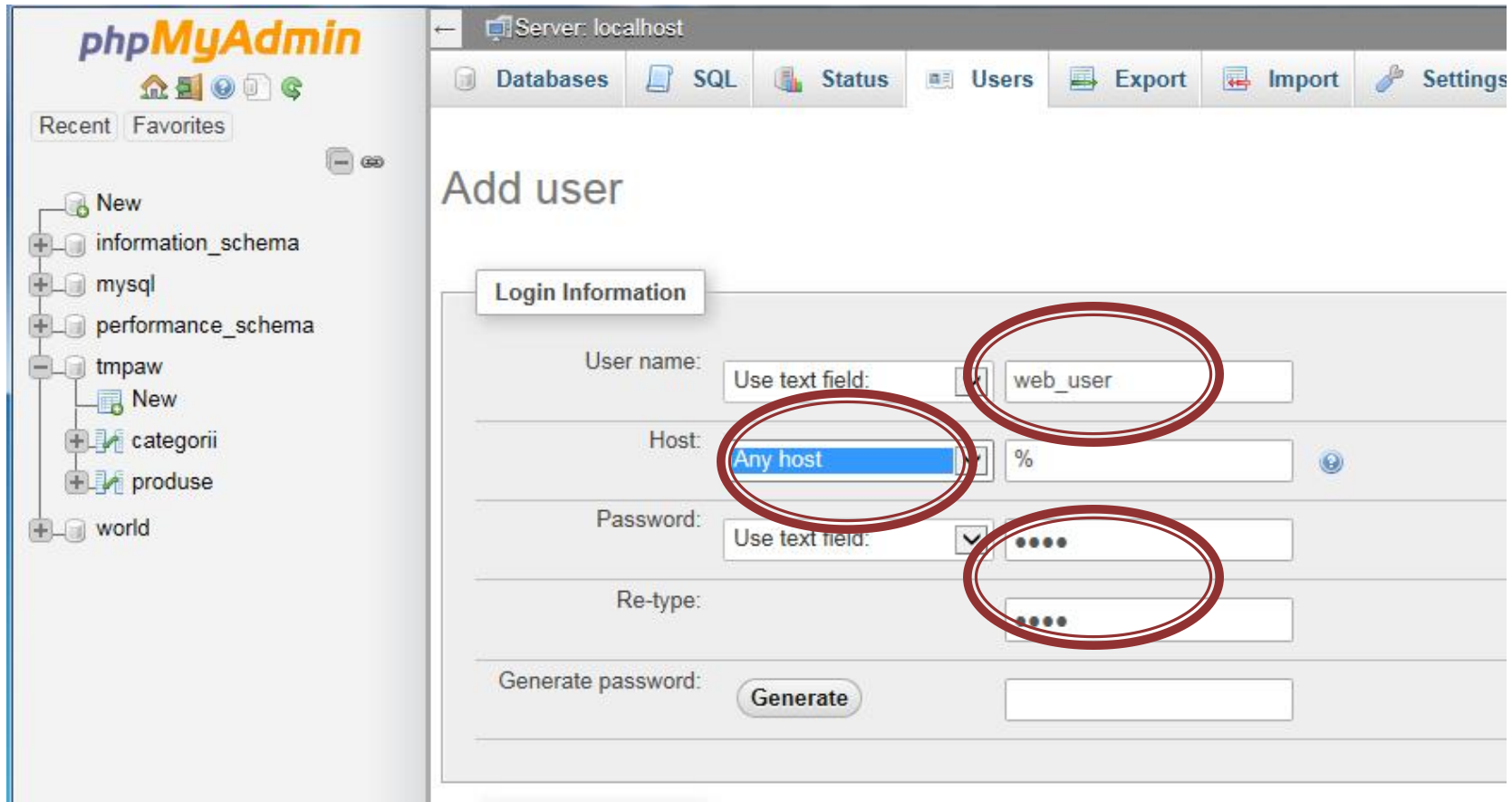
↑ ☐ Check All With selected: Export

New

Add user

# Adaugare utilizator

- Nu e recomandabil/**posibil** sa se utilizeze user-ul MySql "root" pentru aplicatii



The screenshot shows the phpMyAdmin interface for adding a new user. The left sidebar displays a tree view of databases and tables, including 'information\_schema', 'mysql', 'performance\_schema', 'tmpaw', 'categorii', 'produse', and 'world'. The main panel is titled 'Add user' and contains a 'Login Information' section. This section has four rows of input fields: 'User name:' with a text field containing 'web\_user', 'Host:' with a dropdown menu showing 'Any host', 'Password:' with a text field containing masked characters, and 'Re-type:' with another masked text field. A 'Generate password:' button is located at the bottom. Four red circles are drawn around the 'web\_user' text, the 'Any host' dropdown, and both password input fields to highlight the default or recommended values for application users.



# Drepturi de acces

- Server → Users → Edit Privileges

The screenshot shows the phpMyAdmin interface. The left sidebar contains a tree view of databases and schemas. The main content area displays the 'Users overview' page. The top navigation bar includes links for Databases, SQL, Status, Users, Export, Import, and Settings. The 'Users overview' table lists users and their privileges. The 'Edit Privileges' link for the 'web\_user' row is circled in red.

	User name	Host	Password	Global privileges	Grant	Action
<input type="checkbox"/>	root	127.0.0.1	Yes	ALL PRIVILEGES	Yes	<a href="#">Edit Privileges</a> <a href="#">Export</a>
<input type="checkbox"/>	root	:::1	Yes	ALL PRIVILEGES	Yes	<a href="#">Edit Privileges</a> <a href="#">Export</a>
<input type="checkbox"/>	root	localhost	Yes	ALL PRIVILEGES	Yes	<a href="#">Edit Privileges</a> <a href="#">Export</a>
<input type="checkbox"/>	root	tmpaw.etti	Yes	ALL PRIVILEGES	Yes	<a href="#">Edit Privileges</a> <a href="#">Export</a>
<input type="checkbox"/>	web	%	Yes	USAGE	No	<a href="#">Edit Privileges</a> <a href="#">Export</a>
<input type="checkbox"/>	web_user	%	Yes	USAGE	No	<a href="#">Edit Privileges</a> <a href="#">Export</a>

# Drepturi de acces

- Database → nume → Go

The screenshot shows the phpMyAdmin interface for a local MySQL server. The left sidebar displays a tree of databases, including 'information\_schema', 'mysql', 'performance\_schema', 'tmpaw', and 'world'. The main panel is titled 'Edit Privileges: User 'web\_user'@'%''. The 'Database' tab is selected and circled in red. Below the title, there is a section for 'Database-specific privileges' with a table showing no privileges. At the bottom, a list of databases is provided for selecting privileges, with 'mysql', 'tmpaw', and 'world' circled in red.

Server: localhost

Databases SQL Status Users Export Import Settings

Global Database Change password Login Information

Edit Privileges: User 'web\_user'@'%'

Database-specific privileges

Database	Privileges	Grant	Table-specific privileges	Action
None				

mysql  
tmpaw  
world

Add privileges on the following database(s).

# Drepturi de acces

- Se alocă drepturile SELECT + INSERT + UPDATE + DELETE asupra bazei de date create

The screenshot shows the phpMyAdmin interface for a MySQL server on localhost. The left sidebar displays a tree of databases, including 'tmpaw'. The main panel is titled 'Edit Privileges: User **'web\_user'@'%'** - Database **tmpaw**'. Below the title, there is a section for 'Database-specific privileges' with a 'Check All' button. A note states: 'Note: MySQL privilege names are expressed in English.' The privileges are organized into three columns: 'Data', 'Structure', and 'Administration'. The 'Data' column has four checked items: SELECT, INSERT, UPDATE, and DELETE. The 'Structure' column has several unchecked items: CREATE, ALTER, INDEX, DROP, CREATE TEMPORARY TABLES, and SHOW VIEW. The 'Administration' column has three unchecked items: GRANT, LOCK TABLES, and REFERENCES. Red circles highlight the 'Data' column and the user and database names in the title.

phpMyAdmin

Recent Favorites

Server: localhost

Databases SQL Status Users Export Import Settings Replication

Database Table

Edit Privileges: User **'web\_user'@'%'** - Database **tmpaw**

Database-specific privileges ☒ Check All

Note: MySQL privilege names are expressed in English.

Data	Structure	Administration
<input checked="" type="checkbox"/> SELECT	<input type="checkbox"/> CREATE	<input type="checkbox"/> GRANT
<input checked="" type="checkbox"/> INSERT	<input type="checkbox"/> ALTER	<input type="checkbox"/> LOCK TABLES
<input checked="" type="checkbox"/> UPDATE	<input type="checkbox"/> INDEX	<input type="checkbox"/> REFERENCES
<input checked="" type="checkbox"/> DELETE	<input type="checkbox"/> DROP	
	<input type="checkbox"/> CREATE TEMPORARY TABLES	
	<input type="checkbox"/> SHOW VIEW	

# Drepturi de acces, verificare

- Nume → Privileges
- Marea majoritate a aplicatiilor **nu** au nevoie de drepturi de acces la structura/administrare

The screenshot shows the phpMyAdmin interface. On the left, the 'tmpaw' database is selected in the sidebar. The main panel displays the 'Privileges' tab for the 'tmpaw' database. The table lists users and their privileges. The 'web\_user' entry is circled in red, showing 'SELECT, INSERT, UPDATE, DELETE' privileges. The 'Privileges' tab is also circled in red.

User	Host	Type	Privileges	Grant	Action
<input type="checkbox"/> root	127.0.0.1	global	ALL PRIVILEGES	Yes	Edit Privileges
<input type="checkbox"/> root	:::1	global	ALL PRIVILEGES	Yes	Edit Privileges
<input type="checkbox"/> root	localhost	global	ALL PRIVILEGES	Yes	Edit Privileges
<input type="checkbox"/> root	tmpaw.etti	global	ALL PRIVILEGES	Yes	Edit Privileges
<input type="checkbox"/> web_user	%	database-specific	SELECT, INSERT, UPDATE, DELETE	No	Edit Privileges

↑ ☐ Check All With selected: Export

# Index

- Adaugare index e esentiala pentru viteza
  - exemplu, produse grupate pe categorii, selectia produselor dintr-o categorie se face cu :
    - `SELECT * FROM `produse` WHERE `id_categ` = 1`
- Tabel → Structure → Index / Selectare + Index

The screenshot shows the phpMyAdmin interface. On the left, the database structure tree is visible, with 'produse' selected under the 'categorii' database. The main panel displays the 'Table structure' for 'produse'. The table has 6 columns: id\_produs (int(11), PRIMARY, AUTO\_INCREMENT), id\_categ (int(11), PRIMARY), nume (varchar(45), PRIMARY), detalii (varchar(150), PRIMARY), cant (int(11), PRIMARY), and pret (float, PRIMARY). The 'Index' tab is selected, showing the 'Index' button circled in green. The 'Unique' button is also circled in green. The 'Index' button is circled in red. The 'Unique' button is circled in red. The 'Index' button is circled in red. The 'Unique' button is circled in red.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id_produs	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	id_categ	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	nume	varchar(45)	utf8_general_ci		No			Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	detalii	varchar(150)	utf8_general_ci		Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values
5	cant	int(11)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values
6	pret	float			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values







# Verificare/Stergere index

- Apasare +Indexes, se deschide lista de indecsi
- Apasare -Indexes, se inchide lista de indecsi

- Indexes

Indexes ?

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
 Edit  Drop PRIMARY		BTREE	Yes	No	id_produ	9	A	No	
 Edit  Drop id_categ		BTREE	No	No	id_categ	9	A	No	

Create an index on  columns

# Backup, Restore

- Ca si in cazul Windows 2000 facilitatea de Backup realizeaza un script SQL care contine structura si datele exprimate sub forma de interogari SQL
- O deosebire intre PhpMyAdmin si aplicatiile specifice MySql (aceleasi de pe Windows 2000 sau MySql Workbench) este absenta liniilor de creare a bazei de date
  - CREATE DATABASE IF NOT EXISTS tmpaw;
  - USE tmpaw;
- La utilizarea PhpMyAdmin trebuie sa se creeze manual inaintea restaurarii baza de date

# Backup

- Nume (tabel sau baza de date) → Export

The screenshot displays the phpMyAdmin web interface. On the left sidebar, the database 'tmpaw' is selected and circled in red. The main panel shows the 'Export' tab, also circled in red. The title of the panel is 'Exporting tables from "tmpaw" database'. Under 'Export Method', the 'Quick' option is selected. Under 'Output', the 'Save on server' checkbox is checked. Under 'Format', 'SQL' is selected in the dropdown menu. A 'Go' button is at the bottom.

Server: localhost » Database: tmpaw

Structure SQL Search Query **Export** Import Operations

### Exporting tables from "tmpaw" database

**Export Method:**

- ☒ Quick - display only the minimal options
- ☐ Custom - display all possible options

**Output:**

- ☒ Save on server in the directory `/var/lib/phpMyAdmin/save/`
- ☐ Overwrite existing file(s)

**Format:**

SQL

Go



# Restore

- Se creaza in avans baza de date
- Nume → Import → Browse (alegere fisier backup)
- fisierele SQL pot fi compresate gzip, bzip2, zip

The screenshot shows the phpMyAdmin interface. On the left sidebar, the database 'tmpaw' is selected and circled in red. The main panel shows the 'Import' tab, also circled in red. The title is 'Importing into the database "tmpaw"'. Under 'File to Import:', there is a text area for the file name and a 'Browse...' button, both circled in red. Below this, there are radio buttons for 'Browse your computer' and 'Select from the web server upload directory'. The 'Character set of the file:' is set to 'utf-8'. Under 'Partial Import:', there is a checked checkbox for 'Allow the interruption of an import...' and a text input for 'Skip this number of queries...' set to '0'.

phpMyAdmin

Server: localhost » Database: tmpaw

Structure SQL Search Query Export Import Operations Privileges Routines

### Importing into the database "tmpaw"

**File to Import:**

File may be compressed (gzip, bzip2, zip) or uncompressed.  
A compressed file's name must end in `.[format].[compression]`. Example: `example.sql.gz`

☐ Browse your computer:  **Browse...** (Max: 2048KiB)

You may also drag and drop a file on any page.

☐ Select from the web server upload directory `/var/lib/phpMyAdmin/upload/`: There are no files to upload!

Character set of the file:

**Partial Import:**

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. (This might be a good way to import large file)

Skip this number of queries (for SQL) or lines (for other formats), starting from the first one:

# Contact

- Laboratorul de microunde si optoelectronica
- <http://rf-opto.etti.tuiasi.ro>
- [rdamian@etti.tuiasi.ro](mailto:rdamian@etti.tuiasi.ro)