

Curs 8

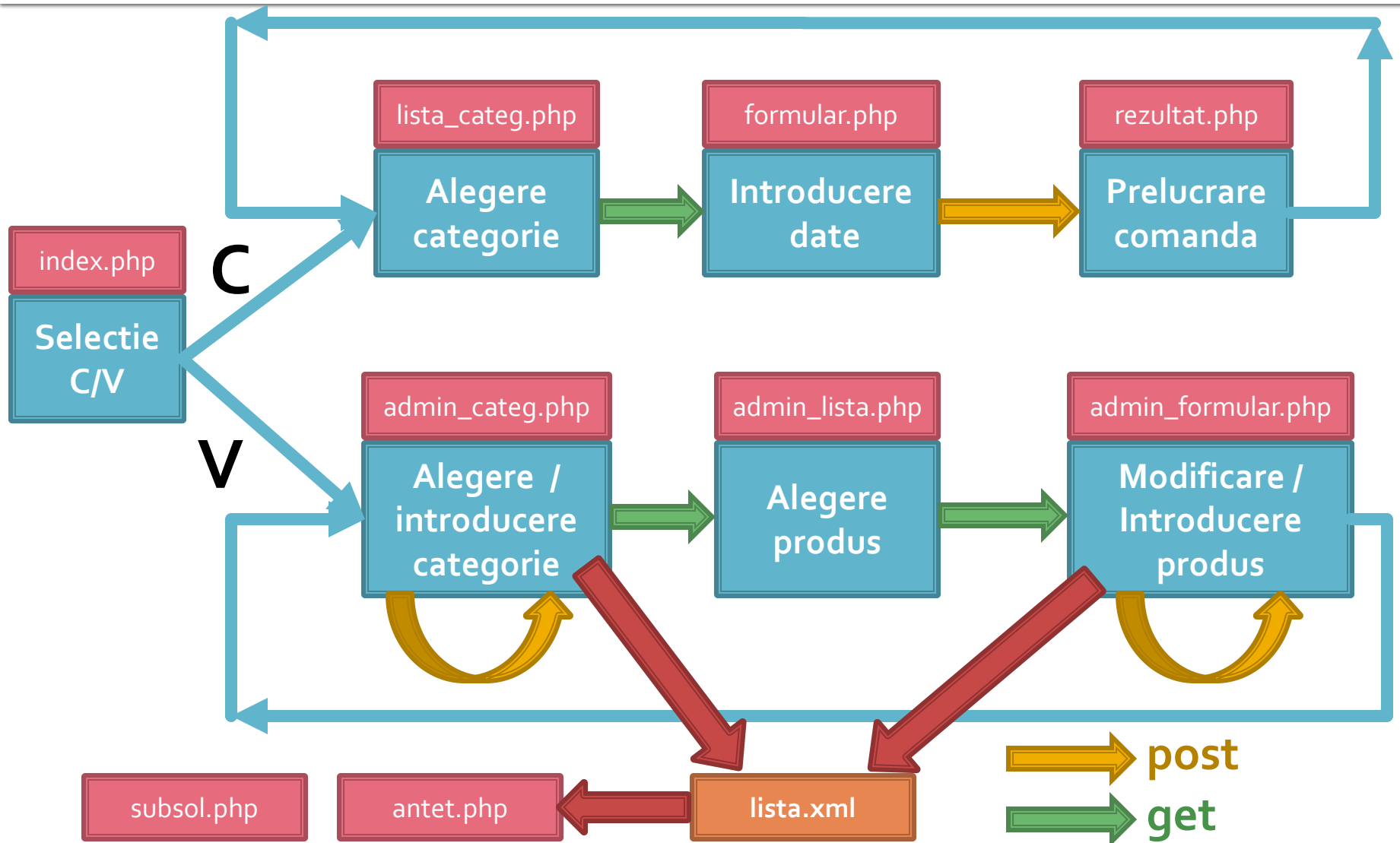
2012/2013

Tehnici moderne de proiectare a aplicatiilor web

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-un fisier **XML**
 - se preia comanda si se calculeaza suma totala
 - **se creaza paginile prin care vanzatorul poate modifica preturile, produsele, categoriile**

Plan aplicatie



Rezultat (comparator)

Categorii Produse

Alegeti categoria:

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

Trimite

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	Papetarie	3
2	Instrumente	3
3	Audio-video	3
4	Calculatoare	3
5	Jucarii	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	modifica
2	Desktop	calculator mare	1000	5	modifica
3	Imprimanta	prn	200	2	modifica
-	Produs nou				adauga

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

Date stocate XML

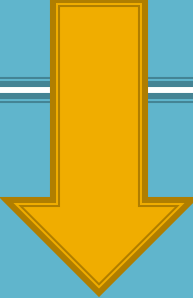
- Codul aplicatiei ramane in mare parte acelasi
- Se modifica doar citirea valorilor pentru popularea matricii \$produse ("antet.php")

```
<?xml version="1.0" encoding="utf-8"?>
<produse user="magazin" password="parola">
  <categorie nume="papetarie">
    <produs>
      <desc>mai multe pagini scrise legate</desc>
      <nume>carte</nume>
      <pret>100</pret>
      <cant>0</cant>
    </produs>
  </categorie>
  <categorie nume="instrumente">
    <produs>
      <desc>loc de depozitat instrumente de scris</desc>
      <nume>penar</nume>
      <pret>150</pret>
      <cant>0</cant>
    </produs>
  </categorie>
```

Citire date XML – antet.php

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

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



Adaugare categorii

- admin_categ.php
 - Fisier unic pentru colectare SI prelucrare date

Categorii Produse

Alegeti categoria:

Nr.	Categorie	Total Produse
1	Papetarie	3
2	Instrumente	3
3	Audio-video	3
4	Calculatoare	3
5	Jucarii	2

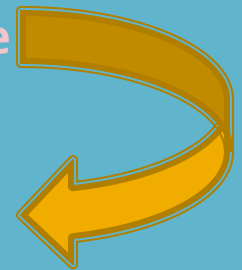
Total produse: 14

Categorie noua de produse:

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

.....//listare categorii, inclusiv cea noua, daca e cazul

```
<form action="<?php echo $_SERVER['PHP_SELF'];?>" method="post">
<input name="nou" type="text" value="" size="50" maxlength="50" />
<input name="c_nou" type="submit" value="Trimite" /></p></form>
```



Lista produse

- transmisie date prin get
 - creare link-uri personalizate
- urlencode/urlencode
 - prevazut pentru situatia utilizarii unor caractere care nu pot fi cuprinse in link-uri
 - cel mai frecvent ar putea sa apara spatii sau diacritice

Lista produse in categoria Calculatoare

Nr.	Produs	Descriere	Pret	Cantitate	Actiuni
1	Laptop	calculator mic	2000	2	modifica
2	Desktop	calculator mare	1000	5	modifica
3	Imprimanta	prn	200	2	modifica
-	Produs nou				adauga

```
<?php $categ=urldecode($_GET['c']); ?>
```

.....

```
<td><a href="admin_formular.php?c=<?php echo urlencode($categ);?>
&p=<?php echo urlencode($prod);?>">modifica</a></td>
```

.....

```
td><a href="admin_formular.php?c=<?php echo urlencode($categ);?>">
adauga</a></td>
```

Adaugare produs

- același cod pentru produs existent sau nou

```
if (isset($_GET['p']))
    {
    $prod=urldecode($_GET['p']);
    $descriere=$produse[$categ][$prod]["descr"];
    $pret=$produse[$categ][$prod]["pret"];
    $cantitate=$produse[$categ][$prod]["cant"];
    ?><input name="prod_ant" type="hidden" value="<?php echo $prod;?>" /><?php
    }
else
    {
    $prod="";
    $descriere="";
    $pret="";
    $cantitate="";
    }
```

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"/>
<input type="button" value="Trimite"/>	

Adaugare produs

```
if (isset($_POST["prod_ant"]))
    {//trebuie sters produsul anterior inlocuit
    unset($produse[$_POST['categ']][$_POST['prod_ant']]);
    }

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

//codul care urmeaza poate fi folosit initial pentru a face conversia spre XML din alte formate
$xml = new SimpleXMLElement("<?xml version='1.0' encoding='utf-8'>\n<produse user='magazin'
password='parola'>\n</produse>");//elementul root
foreach ($produse as $categ => $lista_categ)
    {
    //adaugare categorie noua
    $categ_nou = $xml->addChild("categorie");
    $categ_nou->addAttribute("nume", $categ);
    foreach ($lista_categ as $prod => $detalii)
        {
        //adaugare descendent tip produs in categorie
        $produs_nou = $categ_nou->addChild("produs");
        //adaugare descendenti detalii in produs
        $produs_nou->addChild("nume", $prod); $produs_nou->addChild("desc", $detalii["descr"]);
        $produs_nou->addChild("pret", $detalii["pret"]); $produs_nou->
        >addChild("cant", $detalii["cant"]);
        }
    }
$xml->asXML("lista.xml");
```

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"/>
<input type="button" value="Trimite"/>	

MySql

Laborator 7

Laborator 7

- Sa se continue magazinul virtual cu:
 - lista de produse si preturi se citeste dintr-o baza de date **MySQL**
 - se realizeaza structura bazei de date MySQL necesara
 - **se creaza o pagina prin care vanzatorul poate modifica preturile si produsele**

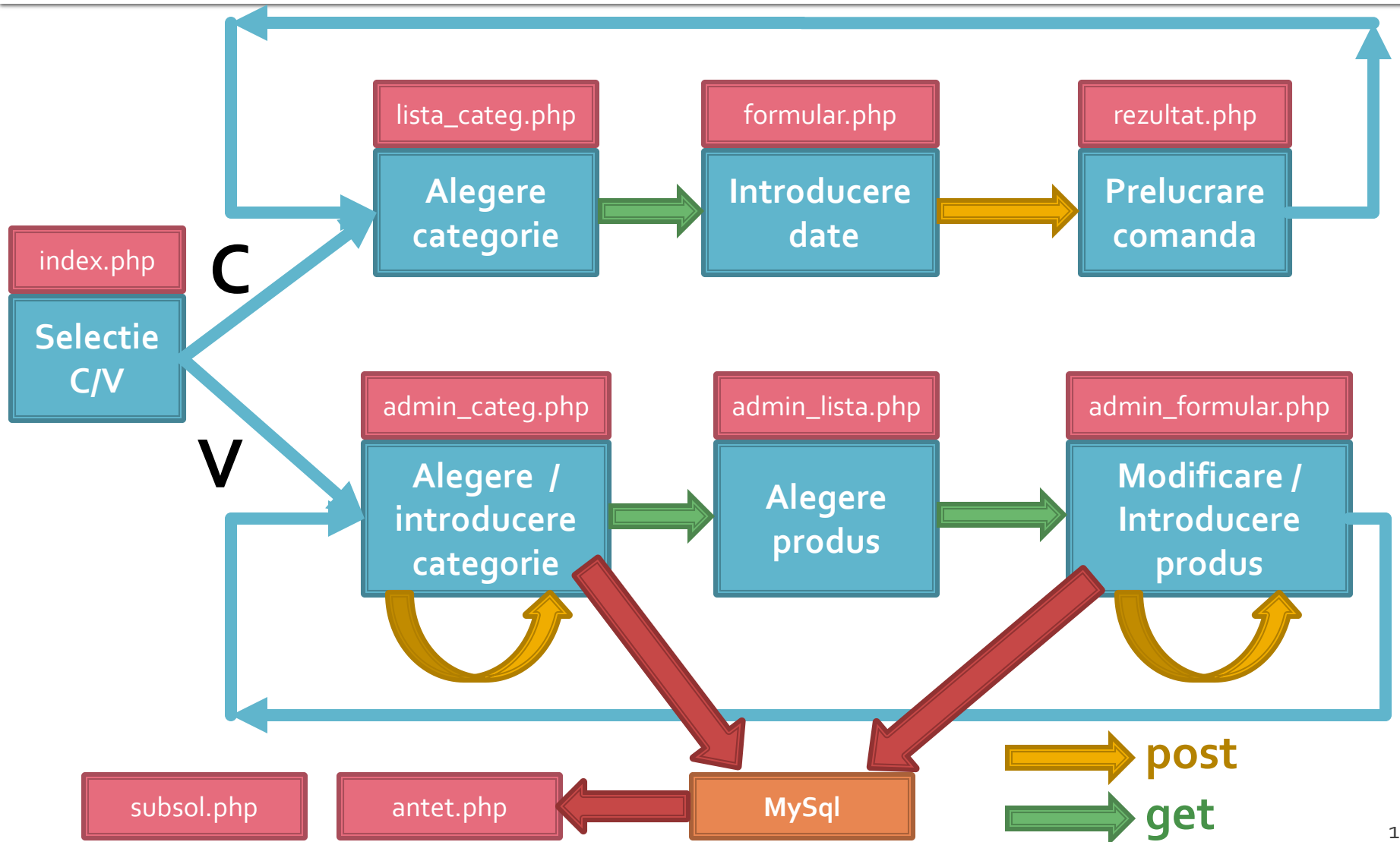
Laborator 7

- exemplu de structura baza de date

Id_gr autoincrement	Nume varchar(50)	Descriere varchar(250)
1	papetarie	...
2	instrumente	...

Id_pr autoincrement	Id_gr integer	Nume varchar(50)	Descriere varchar(250)	Pret float	Cantitate integer
1	1	carte	...	150.0	5
2	1	caiet	...	50.1	6
3	2	stilou	...	23	2

Plan aplicatie



MySql

Accesul la metode externe de stocare eficiente a datelor

MySQL vs. XML

- XML - eXtensible Markup Language
- XML isi atinge limitările atunci cand:
 - cantitatea de date este mare
 - prelucrarile datelor sunt complexe
- In general XML citește in intregime fisierul care contine datele
 - memoria necesara script-urilor PHP poate crește pana in punctul atingerii ineficientei
- Prelucrarile trebuie facute in PHP
 - PHP este limbaj interpretat deci ineficient pentru prelucrari masive de date

MySql – Recapitulare rapida

Relatii in Bazele de date

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

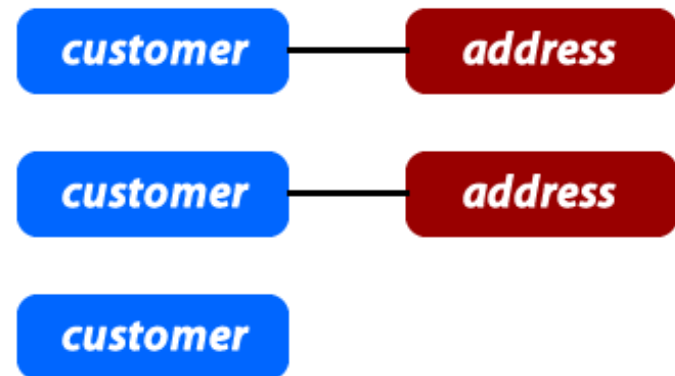
One to One

- 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

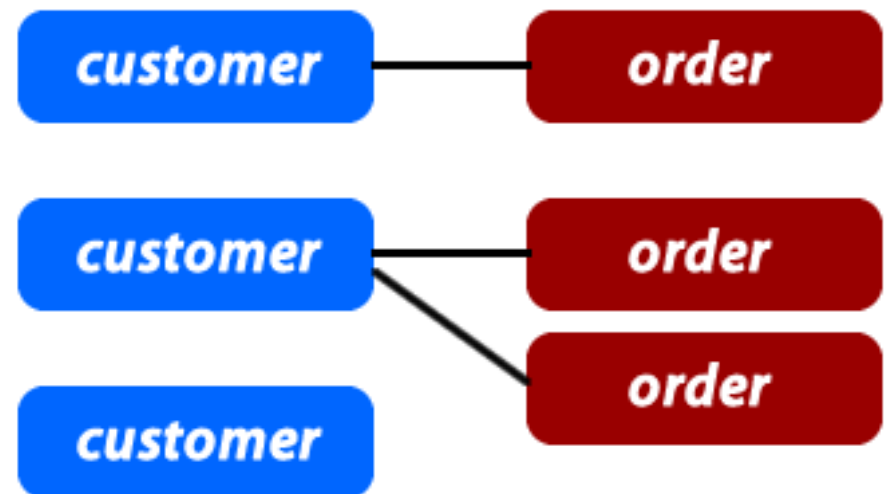
- O linie dintr-un tabel (row), identificata prin cheia primara, poate avea: **nici una, una sau mai multe linii corespondente** in celalalt tabel. In acesta o linie poate fi legata cu o **singura** linie din tabelul primar.
- Analogie cu relatii parinte/copil:
 - fiecare om are o singura mama
 - fiecare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



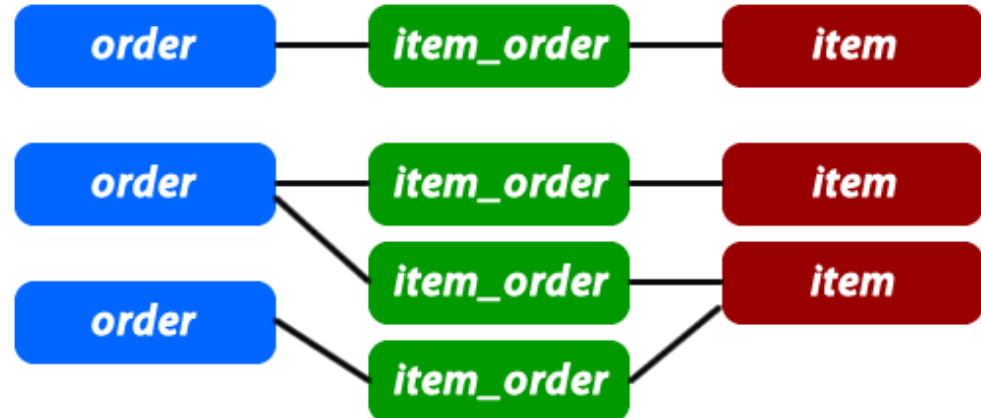
Many to Many

- 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

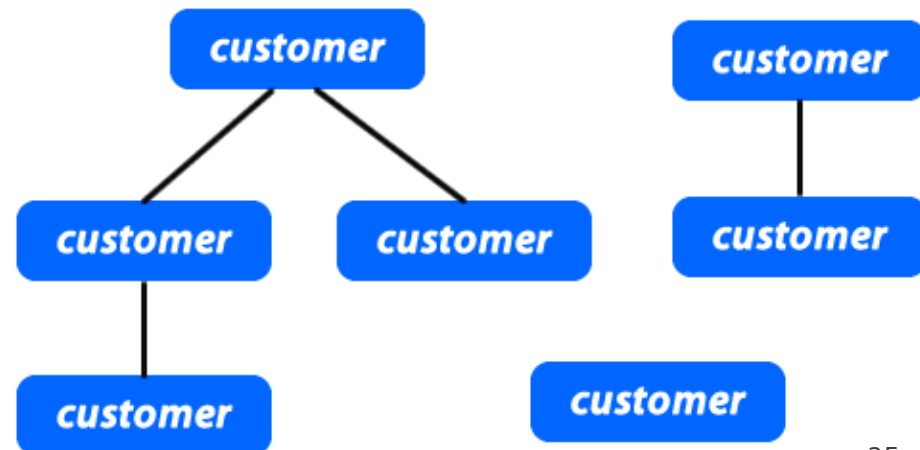
ORDERS				
order_id	customer_id	order_date	amount	
555	101	12/24/09	\$156.78	
556	102	12/25/09	\$99.99	
ITEMS				
item_id	item_name	item_description		
201	Tickle Me Elmo	It wants to be tickled		
202	District 9 DVD	Awesome sci-fi movie		
203	Batarang	It is very sharp		
ITEMS_ORDERS				
order_id	item_id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

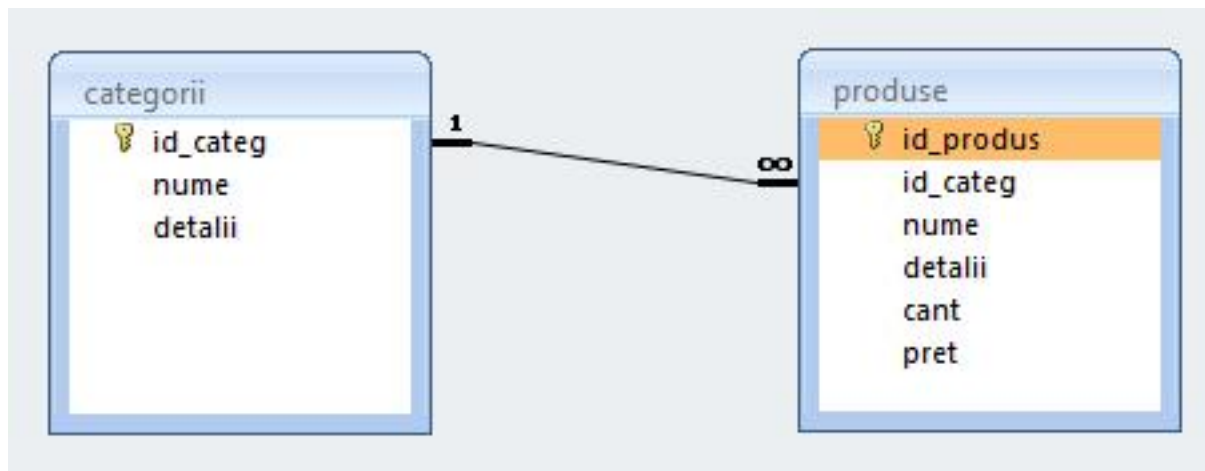
- Respectarea formelor normale ale bazelor de date aduce nenumarate avantaje
- Efectul secundar este dat de necesitatea separarii datelor intre mai multe tabele
- In exemplul utilizat avem doua concepte diferite din punct de vedere logic
 - produs
 - categorie de produs

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

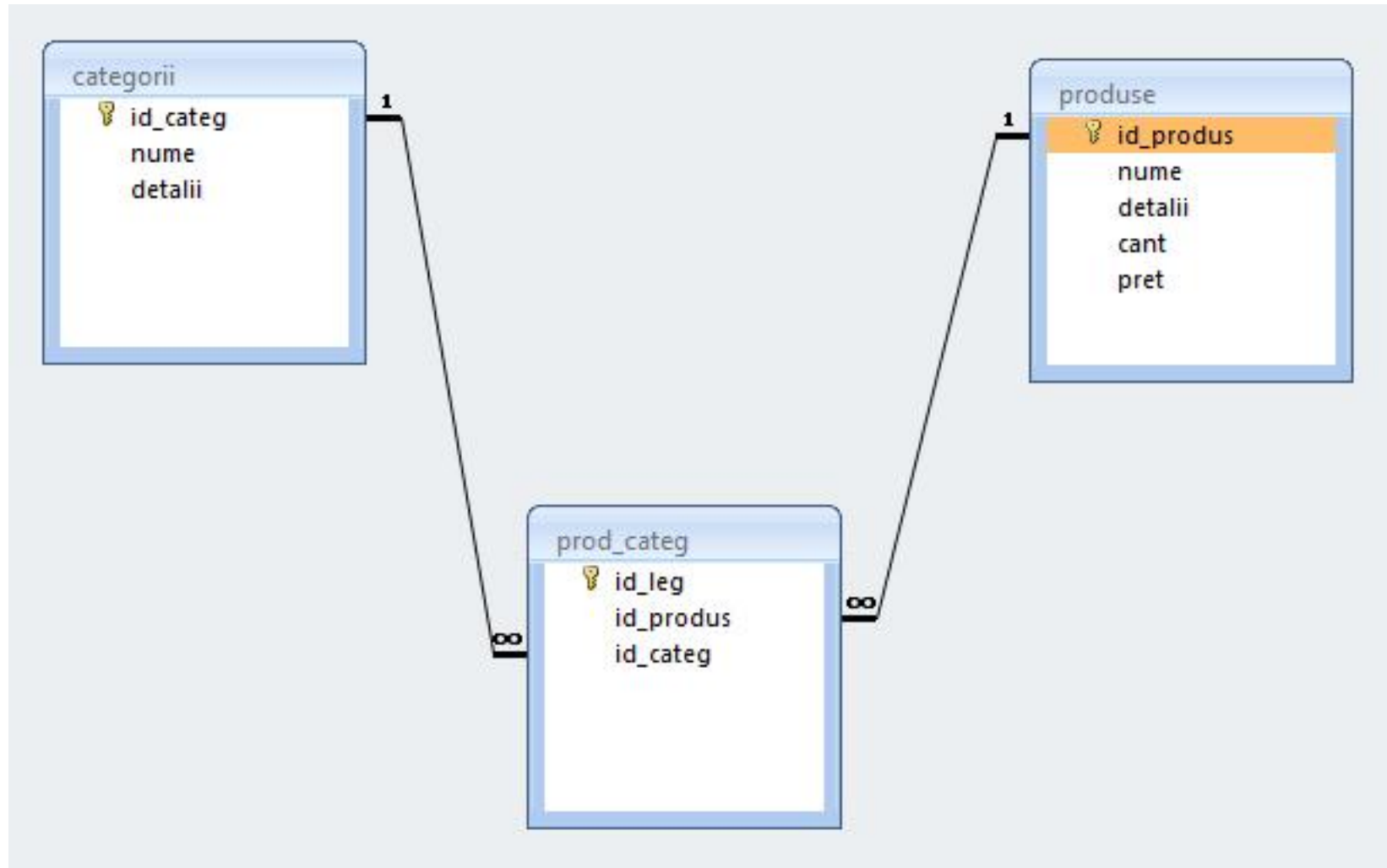
- Legaturile implementata
 - One to Many
 - in tabelul "produse" apare cheia externa (foreign key): "id_categ"



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" într-un întreg pe 64 biti cu tabela asociată

MySql/PHP

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`
 - conectare la server-ul MySQL
 - resource `mysql_connect` ([string server [, string username [, string password [, bool new_link [, int client_flags]]]])
 - tipic: `mysql_connect($host, $user, $pass)`
 - tipic: `$host="localhost"`
- `mysql_pconnect` – persistent pentru reutilizarea conexiunilor

Funcții PHP de acces MySQL

- `mysql_select_db`
 - selectarea bazei de date dorita
 - bool `mysql_select_db` (string `database_name` [, resource `link_identifier`])
 - resursa este obtinuta in urma unui apel anterior la `mysql_connect` sau `mysql_pconnect`
- `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

Funcții PHP de acces MySQL

- `mysql_num_rows`
 - indica numărul de linii returnate SELECT de o interogare sau SHOW
 - int `mysql_num_rows` (resource result)
- `mysql_affected_rows`
 - indica numărul de linii afectate de o interogare INSERT, UPDATE, REPLACE sau DELETE
 - int `mysql_affected_rows` ([resource link_identifier])
- `mysql_insert_id`
 - returnează valoarea unei eventuale coloane autoincrement generate de o interogare INSERT precedentă
 - int `mysql_insert_id` ([resource link_identifier])

Funcții PHP de acces MySQL

Parcurgerea resurselor rezultat

- `mysql_fetch_assoc`
 - returnează o **matrice asociativă** corespunzătoare liniei de la indexul intern (indecsi de tip șir corespunzatori denumirii coloanelor – field – din tabelul de date) și incrementează indexul intern sau **false** dacă nu mai sunt linii
 - array `mysql_fetch_assoc` (resource result)
- `mysql_fetch_row`
 - returnează o matrice cu indecsi întregi
 - array `mysql_fetch_row` (resource result)

Funcții PHP de acces MySQL

Parcurgerea resurselor rezultat

- `mysql_fetch_array`
 - grupează funcționalitatea `mysql_fetch_assoc` și `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 indicată
 - bool `mysql_data_seek` (resource result, int row_number)

Exemplu de utilizare

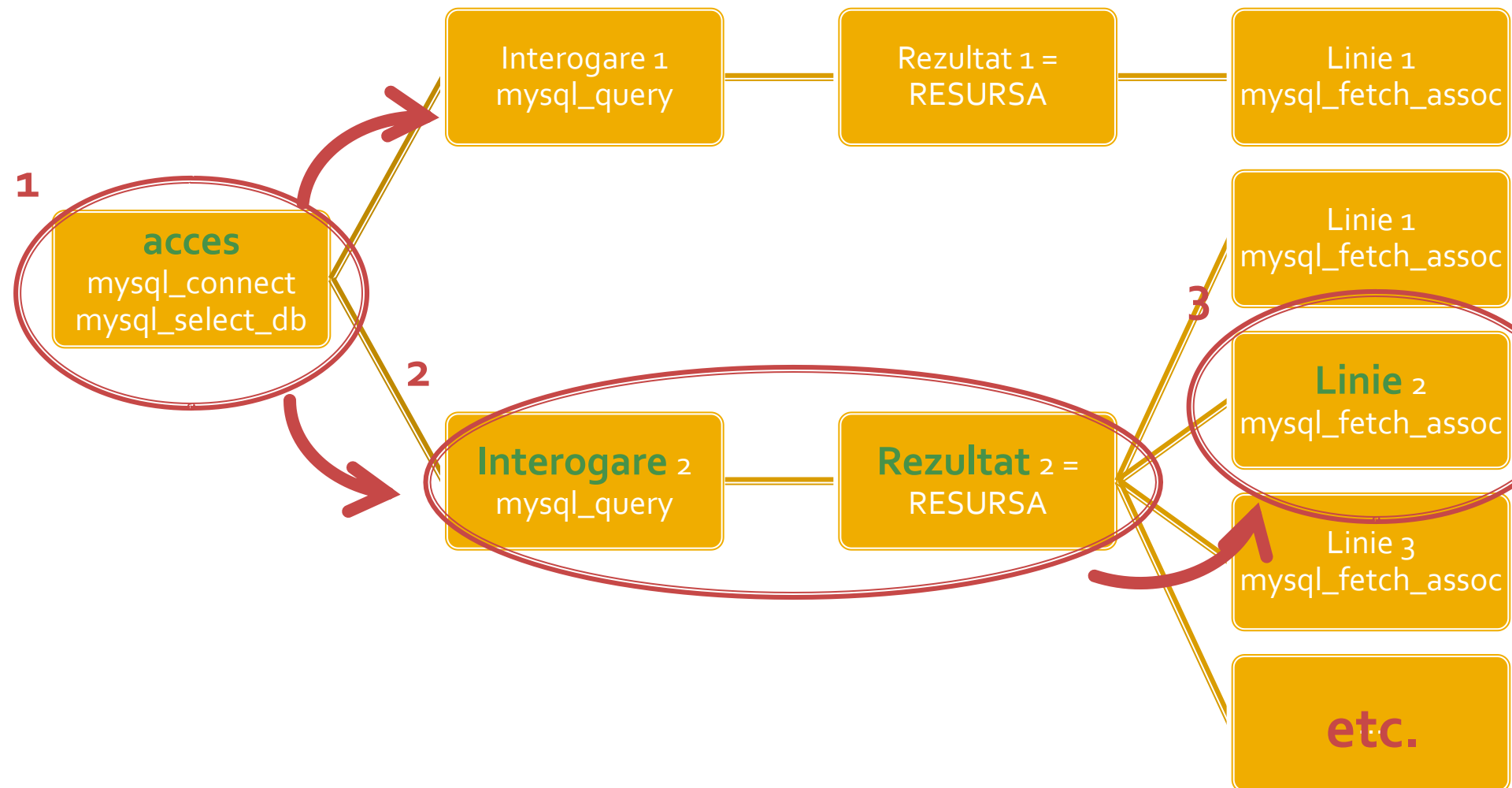
```
$hostname = "localhost";  
$database = "world";  
$username = "web";  
$password = "ceva";  
$conex= mysql_connect($hostname, $username, $password);  
mysql_select_db($database, $conex);
```

```
$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 )); ?>
```

Funcții de acces la server-ul MySQL



MySql

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

The screenshot shows the MySQL Table Editor interface for a table named 'categorii' in the 'tmpaw' database. The table has three columns: 'id_categ' (INT(10), UNSIGNED, ZEROFILL, NULL), 'nume' (VARCHAR(45), BINARY, NULL), and 'detalii' (VARCHAR(150), BINARY, NULL). A primary index is defined on the 'id_categ' column, with index settings: Index Name: PRIMARY, Index Kind: PRIMARY, and Index Type: BTREE.

Table Name: categorii Database: tmpaw Comment: InnoDB free: 11264 kB

Columns and Indices Table Options Advanced Options

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
id_categ	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	
nume	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		
detalii	VARCHAR(150)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	NULL	

Indices Foreign Keys Column Details

PRIMARY

Index Settings

Index Name: PRIMARY

Index Kind: PRIMARY

Index Type: BTREE

Index Columns (Use Drag'n'Drop)

id_categ

Apply Changes Discard Changes Close

Tabel Prognose

The screenshot shows the MySQL Table Editor window for a table named 'produse' in the 'tmpaw' database. The table is currently empty. The editor displays the table's structure with columns and their properties.


Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
id_produ	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	
id_categ	INT(10)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		
nume	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> BINARY		
detalii	VARCHAR(150)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	NULL	
cant	INT(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	
pret	FLOAT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	

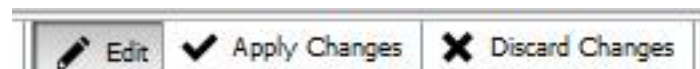
The 'Indices' tab is active, showing a PRIMARY index. The index settings are:

- Index Name: PRIMARY
- Index Kind: PRIMARY
- Index Type: BTREE
- Index Columns: id_produ

Buttons at the bottom include 'Apply Changes', 'Discard Changes', and 'Close'.

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



Introducere date initiale

The screenshot shows the MySQL Query Browser interface. The main window displays the following table data:

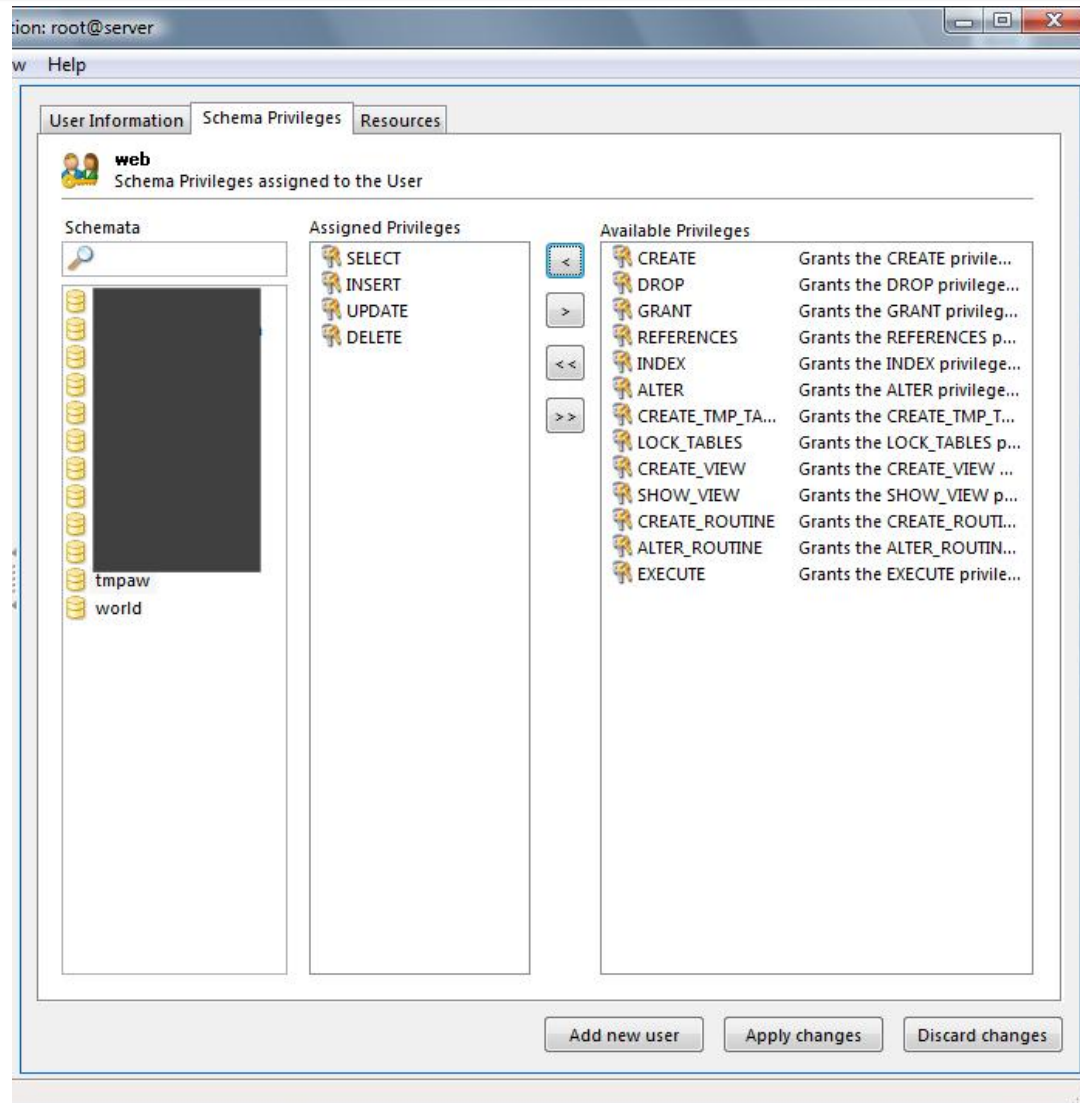
id_produ	id_categ	nume	detalii	cant	pret
1	1	carte	mai multe pagini scrise legate	0	100
2	1	caiet	mai multe pagini goale legate	0	75
3	1	hartie scris	mai multe pagini goale NElegate	0	50
4	2	penar	loc de depozitat instrumente de scris	0	150
5	2	stilou	instrument de scris albastru	0	125
6	2	creion	instrument 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

The interface includes a menu bar (File, Edit, View, Query, Script, Tools, Window, Help), a toolbar with various icons, and a right-hand sidebar with 'Schemata', 'Syntax', 'Functions', 'Params', and 'Trx' sections. The 'Schemata' section shows a tree view of the database structure, including 'tmpaw', 'categorii', 'produse', and 'world'.

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 aloca drepturile “SELECT” + “INSERT” + “UPDATE” asupra bazei de date create (sau mai multe daca aplicatia o cere)

Drepturi de acces



Backup

The screenshot shows the MySQL Administrator interface for configuring a backup project. The window title is "MySQL Administrator - Connection: root@server". The main area is titled "Backup Project" and has three tabs: "Backup Project", "Advanced Options", and "Schedule".

General

Project Name: Name for this backup project.

Schemata

The Schemata list on the left includes: school, tmpaw, and world. The tmpaw schema is selected and highlighted in blue.


Backup Content

Data directory	Obj...	Rows	Data ...	Last update
<input checked="" type="checkbox"/> tmpaw				
<input checked="" type="checkbox"/> categorii	Inno...	3	16384	
<input checked="" type="checkbox"/> produse	Inno...	9	16384	

At the bottom of the window, there are three buttons: "New Project", "Save Project", and "Execute Backup Now".

Yellow arrows indicate the workflow: one arrow points from the "Backup" icon in the left sidebar to the "Backup Project" tab; another arrow points from the "tmpaw" schema in the Schemata list to the "Backup Content" table; and a third arrow points from the "Execute Backup Now" button to the right side of the window.

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);
```

Indicatii examinare

Teme de proiect

- La toate temele **1p** din nota este obtinut de indeplinirea functionalitatii cerute.
- La toate temele forma paginii prezinta importanta (dependentă de dificultatea temei)

Teme de nota 10

- Necesitatea conlucrării între 2 studenți cu două teme “pereche” (a și b). Fiecare tema se alocă **doar** unei perechi de studenți simultan.
- **Necesitatea** investigării posibilităților de îmbunătățire
- Baza de date cu care se lucrează să conțină minim **100** de înregistrări.
- Tema se **sustine** în săptămâna a 14-a
- Sustinerea este precedată obligatoriu de introducerea datelor într-o bază de date vidă (restaurare **back-up**) și copierea codului pe server-ul de referință
- Forma paginii este importantă

Teme de nota 10+

- Cerinte
 - Necesitatea conlucrării între 3 studenți cu trei teme "pereche". Fiecare tema se alocă **doar** unei echipe de 3 studenți simultan.
 - **Necesitatea** investigării posibilităților de îmbunătățire
 - Baza de date cu care se lucrează să conțină minim **200** de înregistrări.
 - Necesitatea utilizării Javascript în aplicație (aplicație liberă dar cu efect **tehnic** nu estetic)
 - Tema se predă/trimite cu macar 1 zi înaintea susținerii ei
 - Tema se **sustine** în săptămâna a 14-a
 - Susținerea e precedată obligatoriu de introducerea datelor într-o bază de date vidă (restaurare **back-up**) și copierea codului pe server-ul de referință
 - Forma paginii trebuie să respecte cerințele "**F shape pattern**".
- Avantaje
 - **1p bonus** la nota finală (dacă tema e îndeplinită)
 - Acceptarea absentelor la laborator

Exemplu

- 1. Galerie de imagini in care imaginile sunt ordonate dupa categorii.

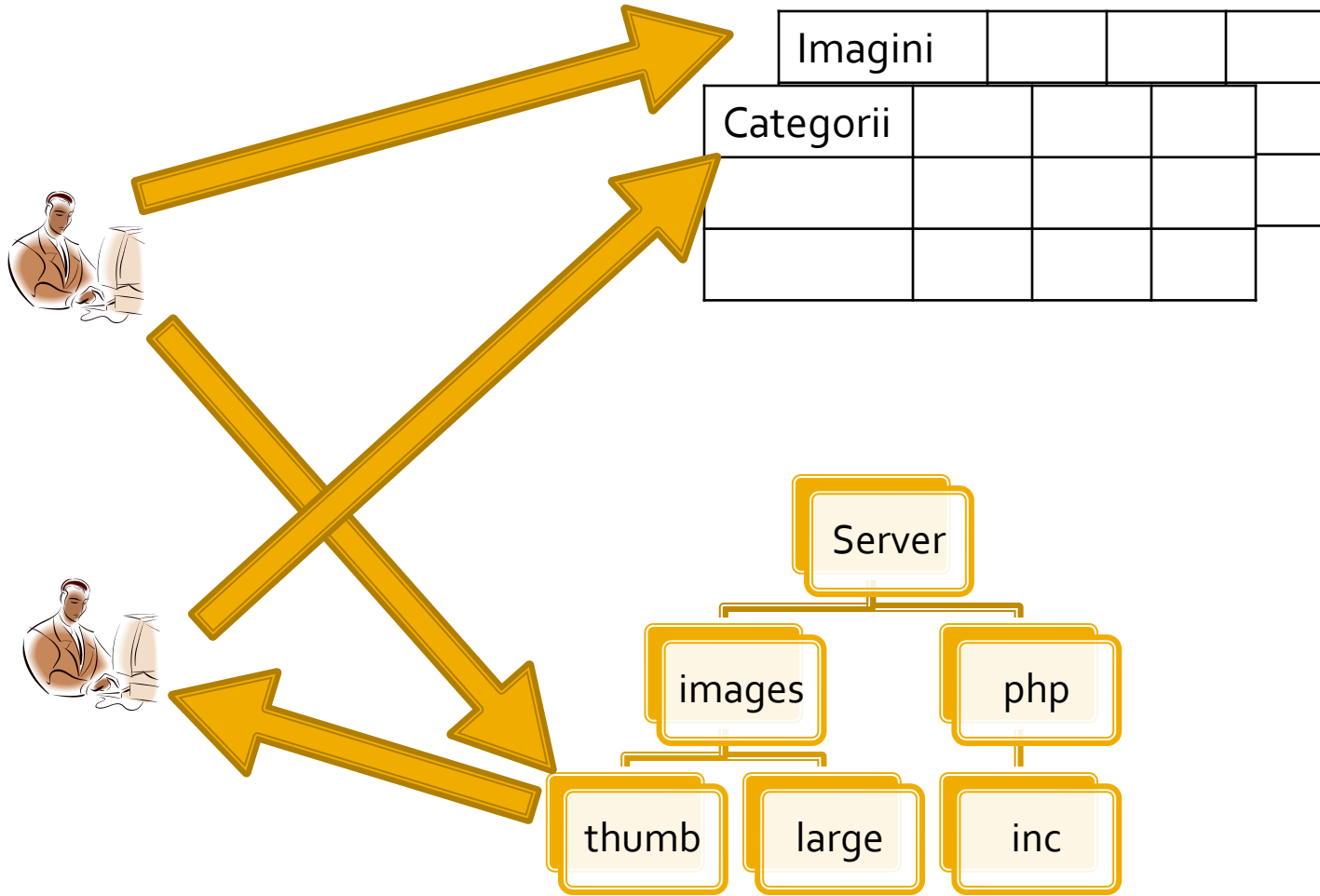


a. aplicatia pentru adaugarea de categorii si afisare a imaginilor (cu alegerea prealabila a categoriei si afisarea listei de imagini format mic)



b. aplicatia pentru adaugare de imaginilor (cu alegerea prealabila a categoriei si generarea prealabila a imaginii format mic)

Exemplu



Teme de proiect

- Functionalitate
 - La toate temele **1p** din nota este obtinut de indeplinirea functionalitatii cerute.
 - orice tehnologie, orice metoda, "sa faca ceea ce trebuie"
- Forma paginii prezinta importanta
 - dependenta de dificultatea temei
- Initiativa
 - **Necesitatea** investigarii posibilitatilor de imbunatatire
- Cooperare
 - Necesitatea conlucrarii intre 2/3 studenti cu teme "pereche"

Notare

- 1p – functionalitate
 - cadrul didactic va incerca sa foloseasca aplicatia respectiva. Daca “pe dinafara e vopsit gardul” se obtine 1p
- 1p – mutarea site-ului (restaurare backup + setare server) pe un server de referinta
 - server-ul de referinta va fi masina virtuala utilizata la laborator (inclusiv aplicatiile cu pricina)
 - sa va pregatiti pentru situatia in care pe acel server exista si alte baze de date care nu trebuie distruse
 - fiecare student isi pune sursele in directorul propriu, in radacina server-ului. Daca tema depinde de anumite fisiere ale colegului, le cereti inainte
- 1p – cunoasterea codului
 - raspunsul la intrebari de genul: “unde ai facut aceasta”
- Teme “de nota 10”
 - 1p – initiativa. Investigarea posibilitatilor de imbunatatire
 - 1p – intrebari legate de cooperarea cu colegul
 - 1p – explicatii relativ la functionarea unei anumite secvente de cod

Examen

- probleme
- fiecare student are subiect propriu
- toate materialele permise
- tehnica de calcul **nu** este necesara dar este permisa

Examen

- Oricare din temele de proiect (sau asemenea) poate constitui una din problemele de examen
 - se va cere realizarea planului / structurii logice a aplicatiei
- Se poate cere scrierea unui cod pentru realizarea anumitor operatii, fara necesitatea corectitudinii tehnice absolute (";", nume corect al functiilor, parametri functie etc.)
- Se poate cere interpretarea unui cod php/MySql cu identificarea efectului

Contact

- Laboratorul de microunde si optoelectronica
- <http://rf-opto.etti.tuiasi.ro>
- rdamian@etti.tuiasi.ro