

LUCRUL CU SABLONUL MVC

De la PHP la MVC PHP

SCOPUL:

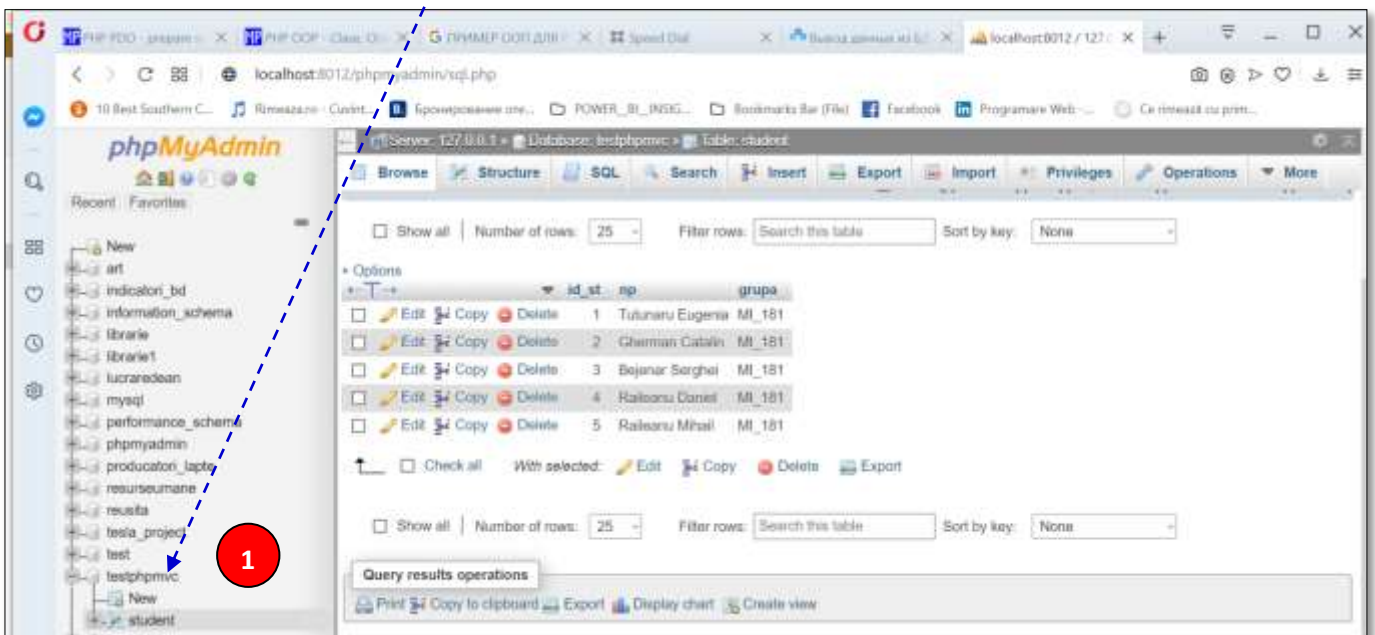
1. Trecerea de la programa PHP, lucrarea de an, la dezvoltarea ei in paradigma MVC PHP.

Astfel, pentru lucrarea de an la MBD, prima parte a ei, urmeaza sa fie realizata lucrarea de an din semestrul precedent efectuata in PHP/HTML/CSS/MYSQL, efectuata in abordarea procedurala, in paradigma MVC, utilizind framework-ul Codeigniter.

2. Prezentarea lucrării de an in MVC PHP utilizeind framework-ul Codeigniter /vezi Lab 1_4/.

In exemplul ce urmeaza este prezentat modul de trecere de la codul PHP procedural, la MVC PHP.

Baza de date, [testphpmvc](#), se prezinta dupa cum urmeaza



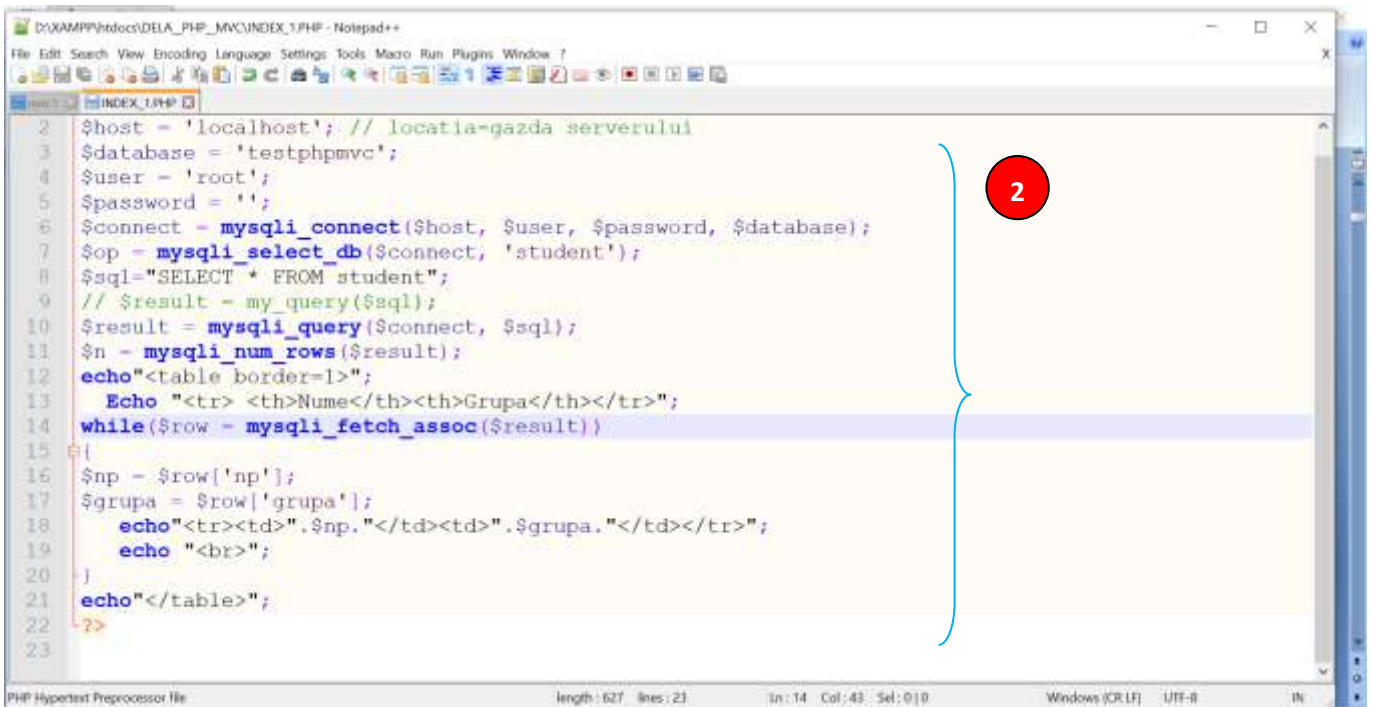
The screenshot shows the phpMyAdmin interface. The left sidebar displays a tree view of databases, with 'testphpmvc' highlighted by a red circle containing the number '1'. A dashed blue arrow points from this circle to the 'student' table in the main window. The main window displays the 'student' table with the following data:

id	st_nu	grupa
1	Tolunaru Eugenia	MI_181
2	Cherman Catalin	MI_181
3	Bojanar Serghei	MI_181
4	Haltoru Daniel	MI_181
5	Raleanu Mihail	MI_181

Codul **PHP/HTML/CSS/MYSQL** – procedural, de procesare a datelor, si anume:

sa se prezinte lista studentilor grupei MI-181, din tabelul "STUDENT" a BD testphpmvc

este prezentat prin scriptul **Index_1.php** după cum urmează:

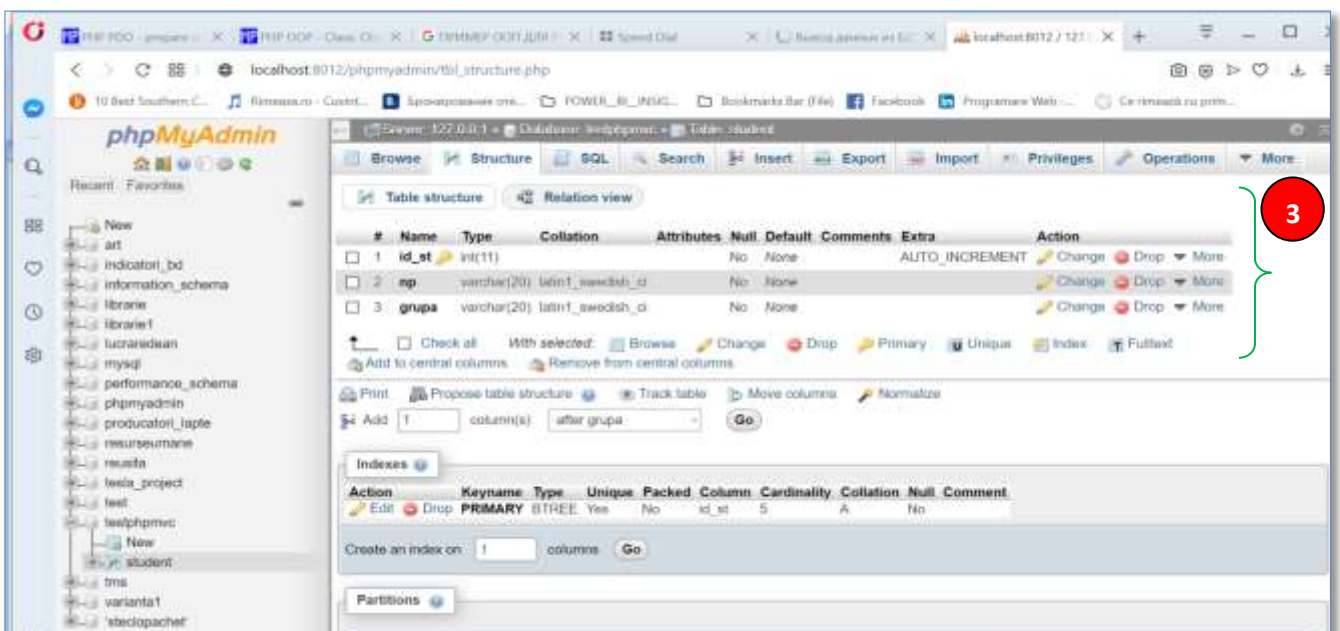


```

2 $host = 'localhost'; // locatia-gazda serverului
3 $database = 'testphpmvc';
4 $user = 'root';
5 $password = '';
6 $connect = mysqli_connect($host, $user, $password, $database);
7 $op = mysqli_select_db($connect, 'student');
8 $sql="SELECT * FROM student";
9 // $result = my_query($sql);
10 $result = mysqli_query($connect, $sql);
11 $n = mysqli_num_rows($result);
12 echo"<table border=1>";
13 Echo "<tr> <th>Nume</th><th>Grupa</th></tr>";
14 while($row = mysqli_fetch_assoc($result))
15 {
16 $np = $row['np'];
17 $grupa = $row['grupa'];
18 echo"<tr><td>".$np."</td><td>".$grupa."</td></tr>";
19 echo "<br>";
20 }
21 echo"</table>";
22 ?>
23

```

Structura BD "**testphpmvc**" si tabelul ei unic "**student**" sunt prezentate aici,

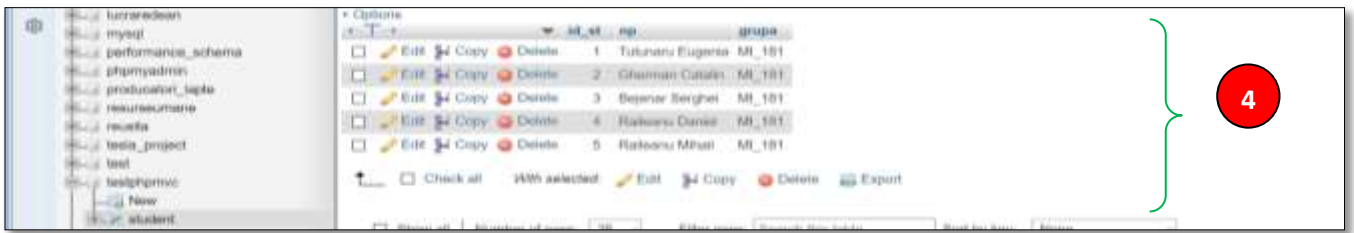


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_st	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	np	varchar(20)	latin1_swedish_ci		No	None			Change Drop More
3	grupa	varchar(20)	latin1_swedish_ci		No	None			Change Drop More

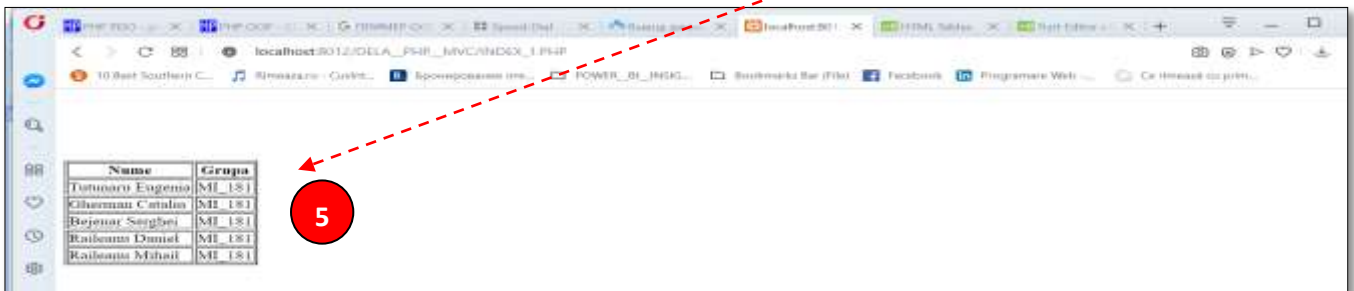
Indexes

Action	Keyname	Type	Unique	Packed	Columns	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id_st	5	A	No	

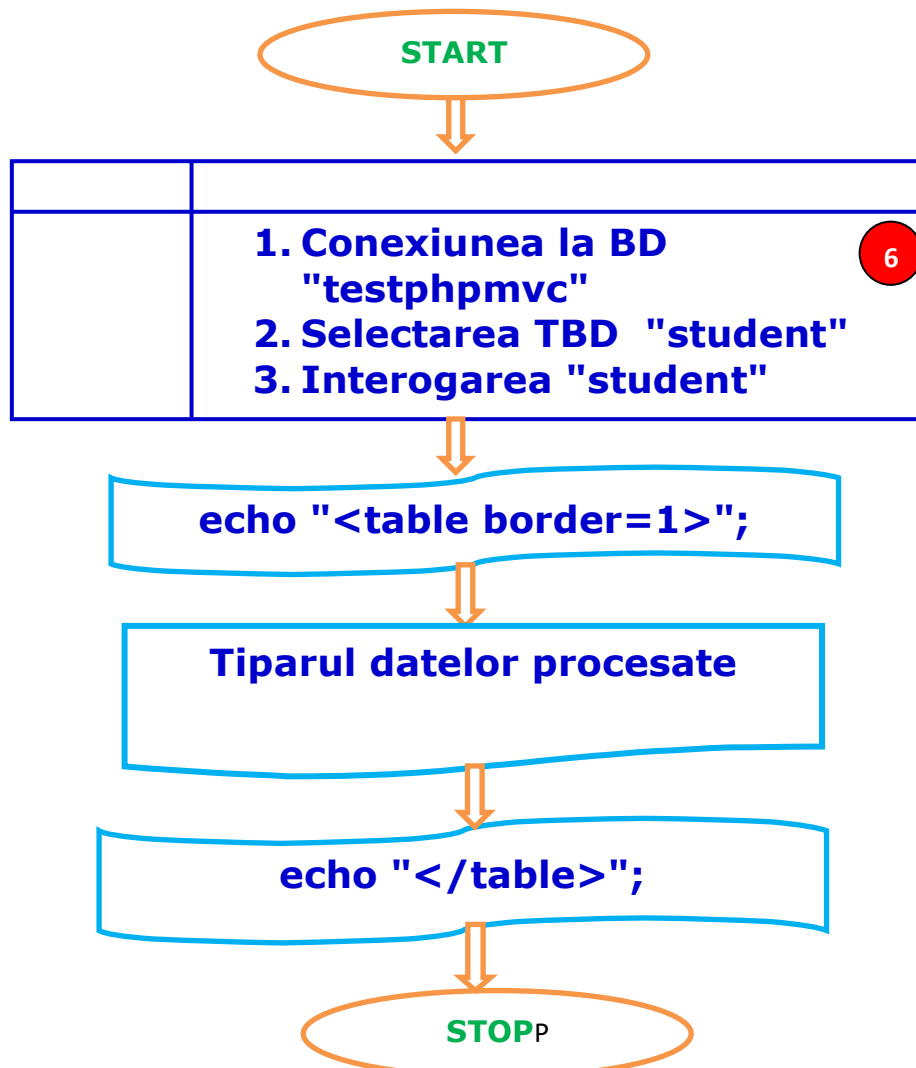
iar datele, sunt prezentate mai jos dupa cum urmeaza



Rezultatul lucrului acestui script "liniar" este



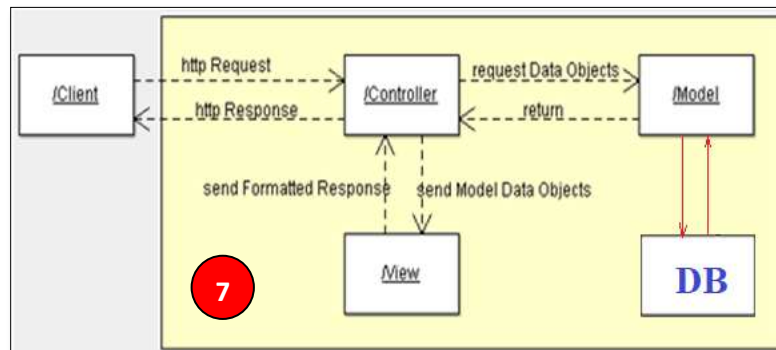
Schea bloc a aplicatiei PHP /liniara/



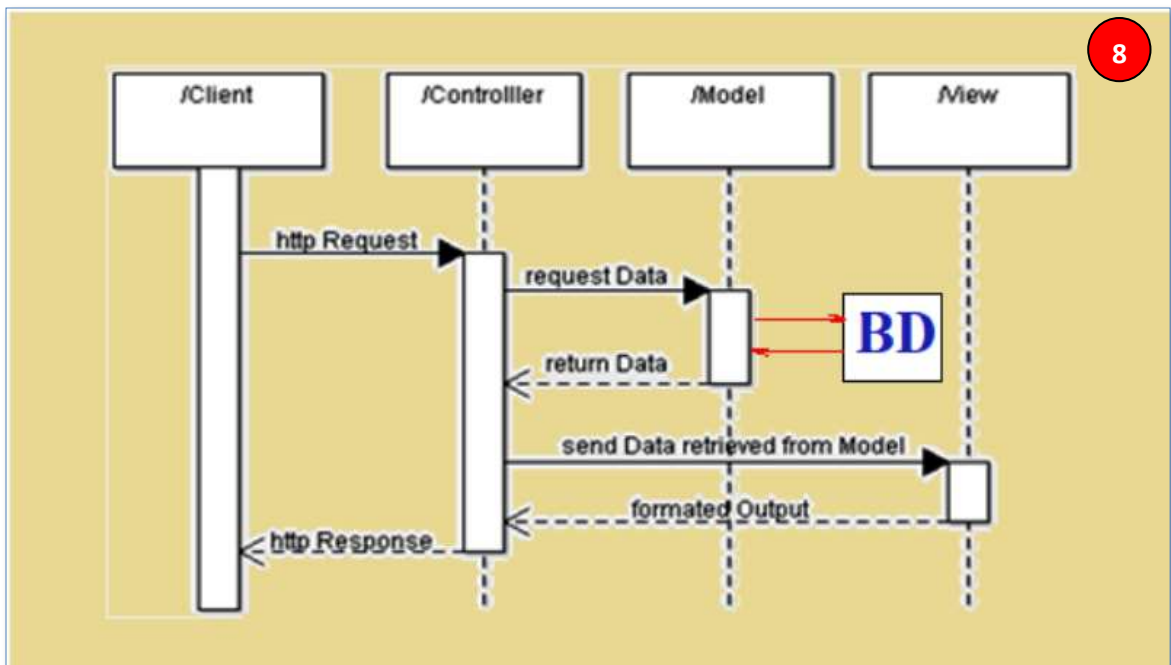
SARCINA

- 1) Cum să prezentăm codul scriptului inițial prezentat în PHP – “procedural”, în paradigma/modelul MVC, dacă el este structurat în componente **view.html**, **model.php** și **controller.php**?
- 2) Cum să prezentăm codul PHP în modelul MVC, folosind paradigma programării orientate pe obiecte (POO) în PHP?

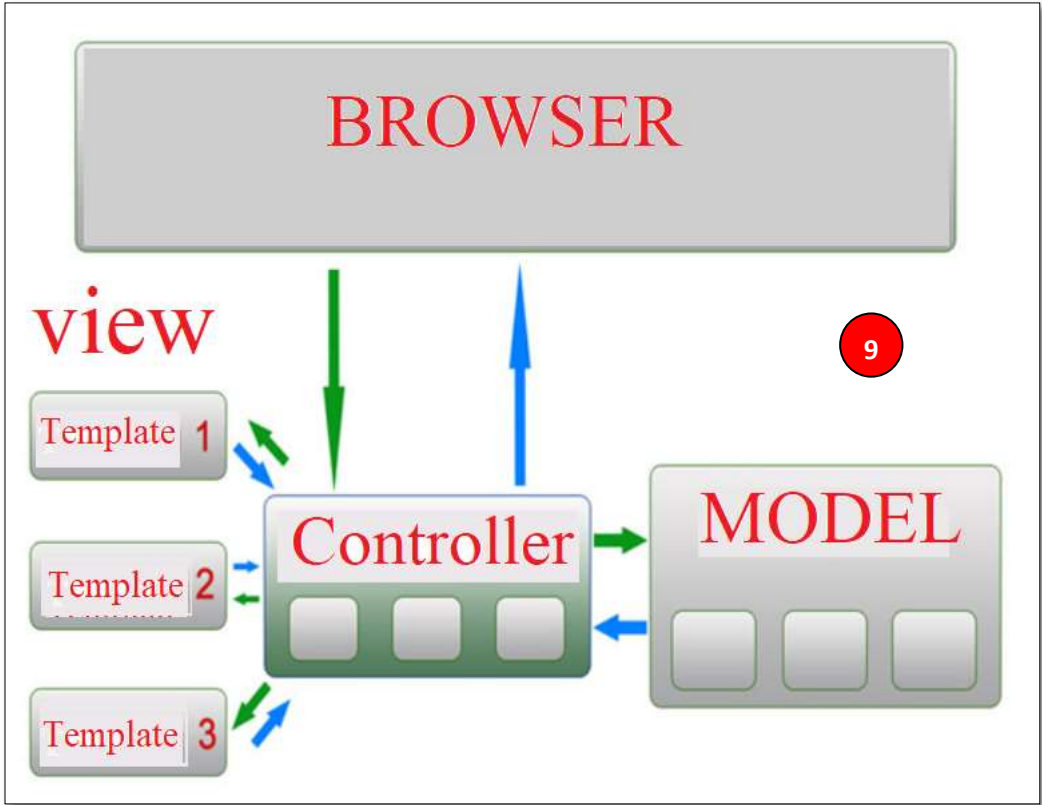
Reamintim paradigma MVC poate fi prezentată schematic după cum urmează



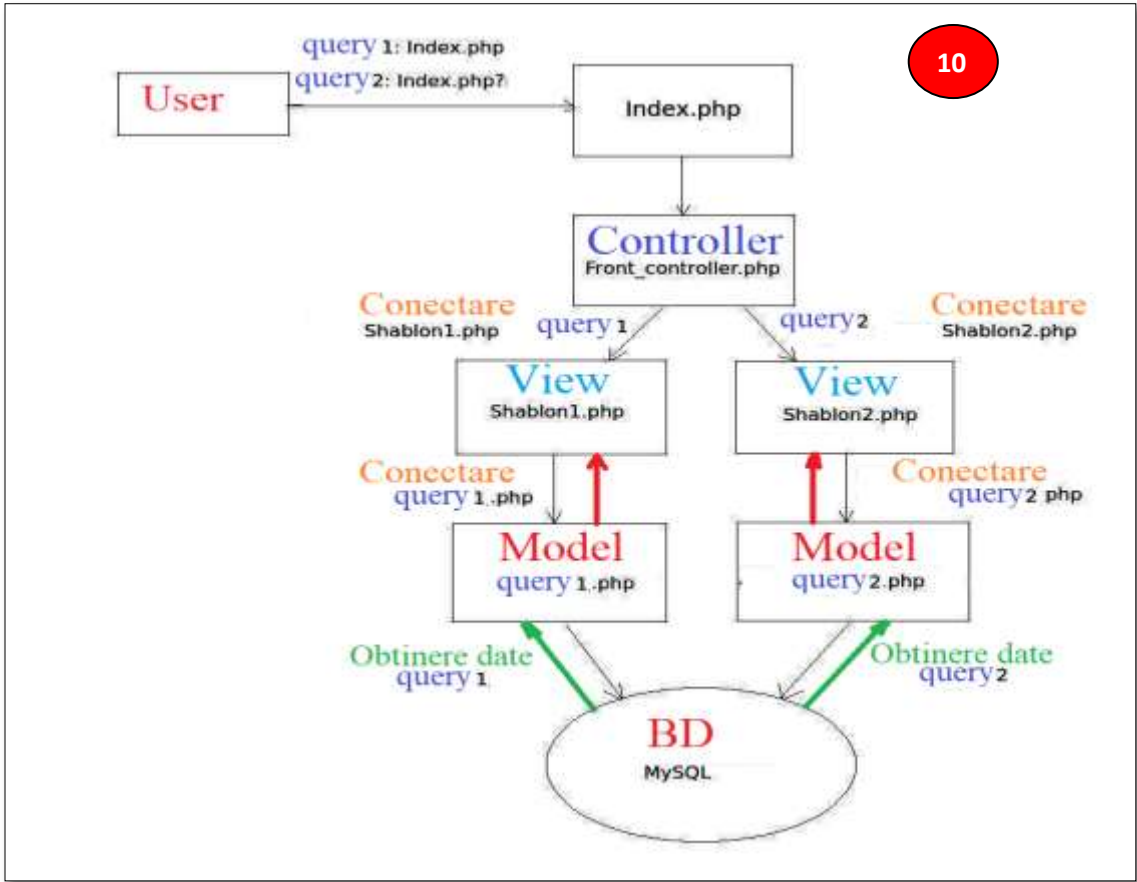
Sau prin următoarea diagramă de secvență MVC, în care putem observa fluxul acțiunilor în timpul unei solicitări prin **http** de la browser:



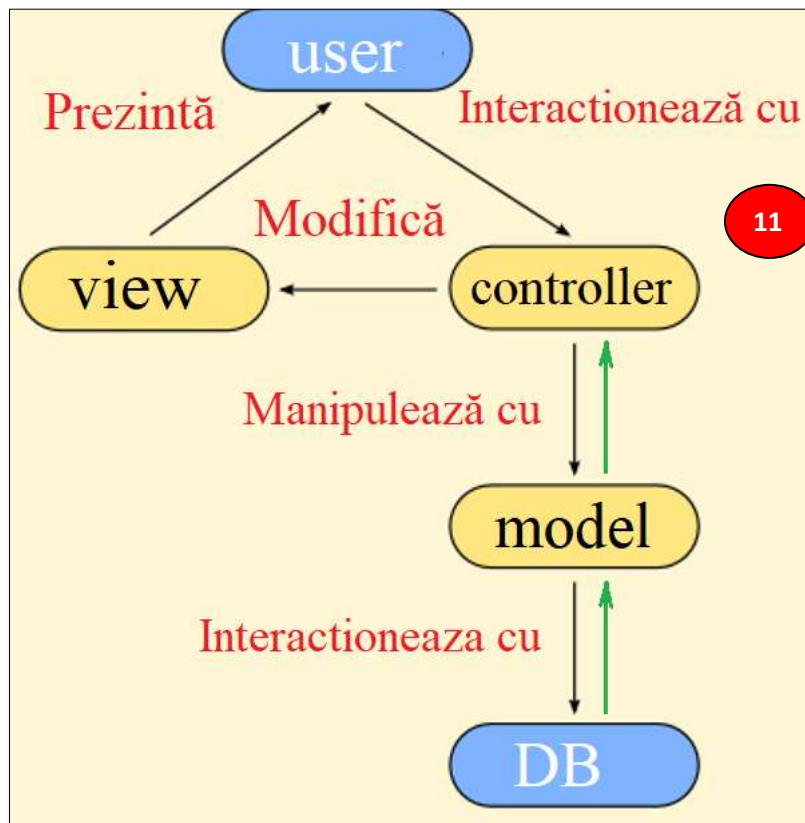
Care mai poate fi prezentată și după cum urmează



Ori in limbajul interogarilor, atunci cind sunt 2 interogari, putem schematic prezenta

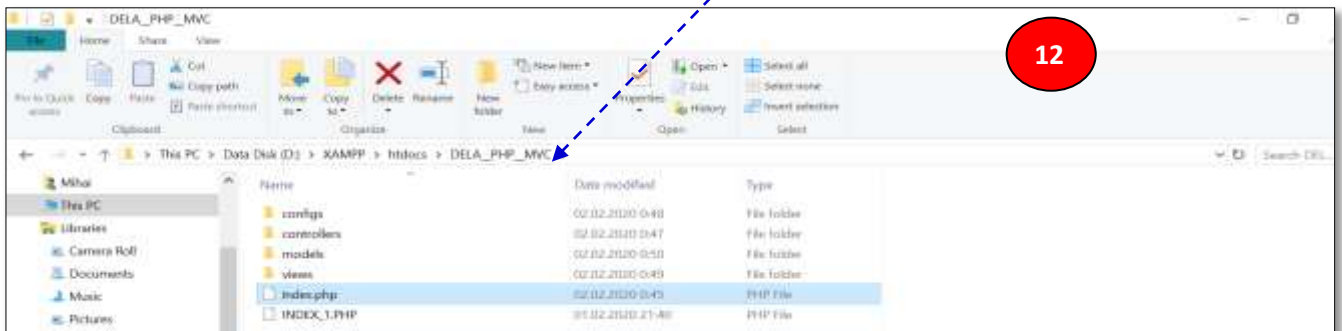


In continuare, pentru cazul nostru vom utiliza următoarea schema



In cazul nostru iată soluția MVC

Structura mapei DELA__PHP__MVC /pentru Xampp/



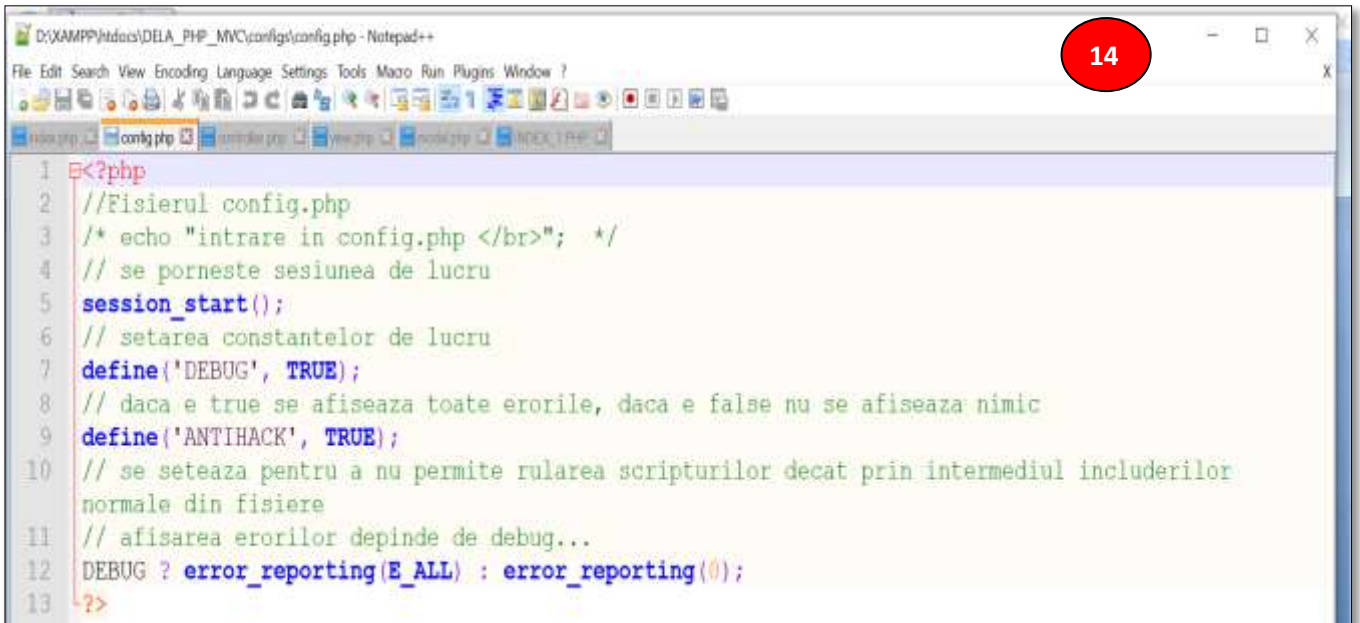
CU FISIERELE IN FIECARE DIN MAPE DUPA CUM URMEAZA:

Fisierul index.php - uneori i se mai zice front-controller

```

1 <?php
2 //Fisierul index.php - uneori i se mai zice front-controller
3 include_once( "configs/config.php" );
4 include_once 'controllers/controller.php';
5 include_once( "views/view.php" );
6 ?>
  
```

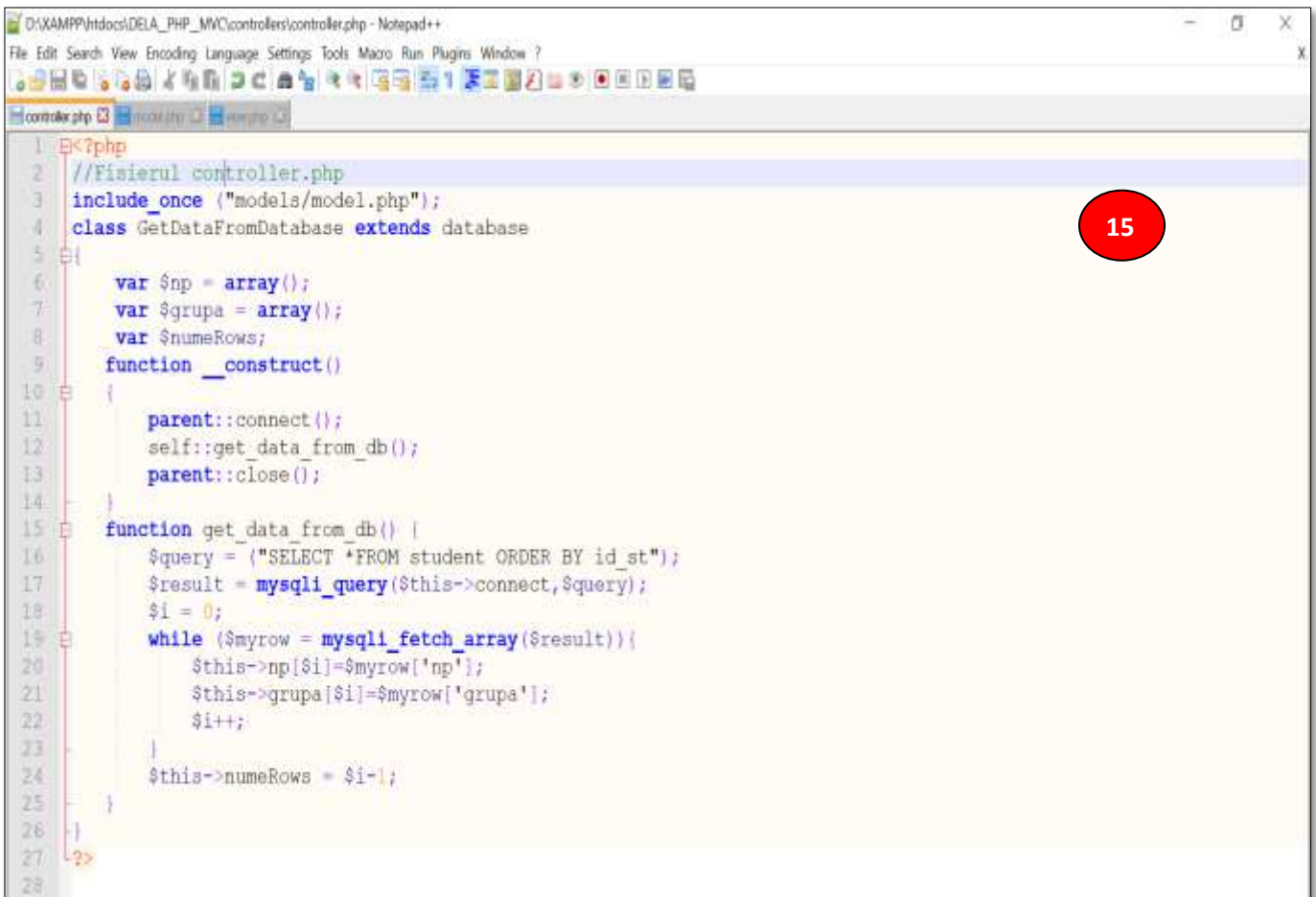
Fisierul config.php



```

1 <?php
2 //Fisierul config.php
3 /* echo "intrare in config.php </br>"; */
4 // se porneste sesiunea de lucru
5 session_start();
6 // setarea constantelor de lucru
7 define('DEBUG', TRUE);
8 // daca e true se afiseaza toate erorile, daca e false nu se afiseaza nimic
9 define('ANTIHack', TRUE);
10 // se seteaza pentru a nu permite rulara scripturilor decat prin intermediul includerilor
    normale din fisiere
11 // afisarea erorilor depinde de debug...
12 DEBUG ? error_reporting(E_ALL) : error_reporting(0);
13 ?>
  
```

Fisierul controller.php



```

1 <?php
2 //Fisierul controller.php
3 include_once ("models/model.php");
4 class GetDataFromDatabase extends database
5 {
6     var $np = array();
7     var $grupa = array();
8     var $numeRows;
9     function __construct()
10    {
11        parent::connect();
12        self::get_data_from_db();
13        parent::close();
14    }
15    function get_data_from_db() {
16        $query = ("SELECT *FROM student ORDER BY id_st");
17        $result = mysqli_query($this->connect,$query);
18        $i = 0;
19        while ($myrow = mysqli_fetch_array($result)){
20            $this->np[$i]=$myrow['np'];
21            $this->grupa[$i]=$myrow['grupa'];
22            $i++;
23        }
24        $this->numeRows = $i-1;
25    }
26 }
27 ?>
28
  
```

Fisierul model.php

```

D:\XAMPP\htdocs\DELA_PHP_MVC\models\model.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
model.php
1 <?php
2 //Fisierul model.php
3 class database
4 {
5     var $connect;
6     var $selectDatabase;
7
8     public function connect()
9     {
10         $server = "localhost";
11         $user = "root";
12         $password = "";
13         $db = "testphpmvc";
14         $this->connect = mysqli_connect($server, $user, $password,$db);
15         $this->selectDatabase = mysqli_select_db($this->connect,'student');
16     }
17
18     function close()
19     {
20         mysqli_close($this->connect);
21     }
22 }
23 ?>
24

```

16

Fisierul view.php

```

D:\XAMPP\htdocs\DELA_PHP_MVC\views\view.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
view.php
1 <?php
2 // Fisierul view.php
3 error_reporting(E_ALL ^ E_DEPRECATED);
4 include_once 'controllers/controller.php';
5 ?>
6 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
7 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
8 <html xmlns="http://www.w3.org/1999/xhtml">
9 <head>
10 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
11 <title></title>
12 </head>
13 <body>
14 <table border="1">
15 <?php
16 $GetDataFromDatabase = new GetDataFromDatabase;
17 $i = 0;
18 echo "<table border=1>";
19 Echo "<tr> <th>Nume</th><th>Grupa</th></tr>";
20 while ($i <= $GetDataFromDatabase->numeRows) {
21     echo "<tr><td>".$GetDataFromDatabase->np[$i]."</td><td>".$GetDataFromDatabase->grupa[$i]."</td></tr>";
22     $i++;
23 }
24 echo "</table>";
25 ?>
26 </table>
27 </body>
28 </html>

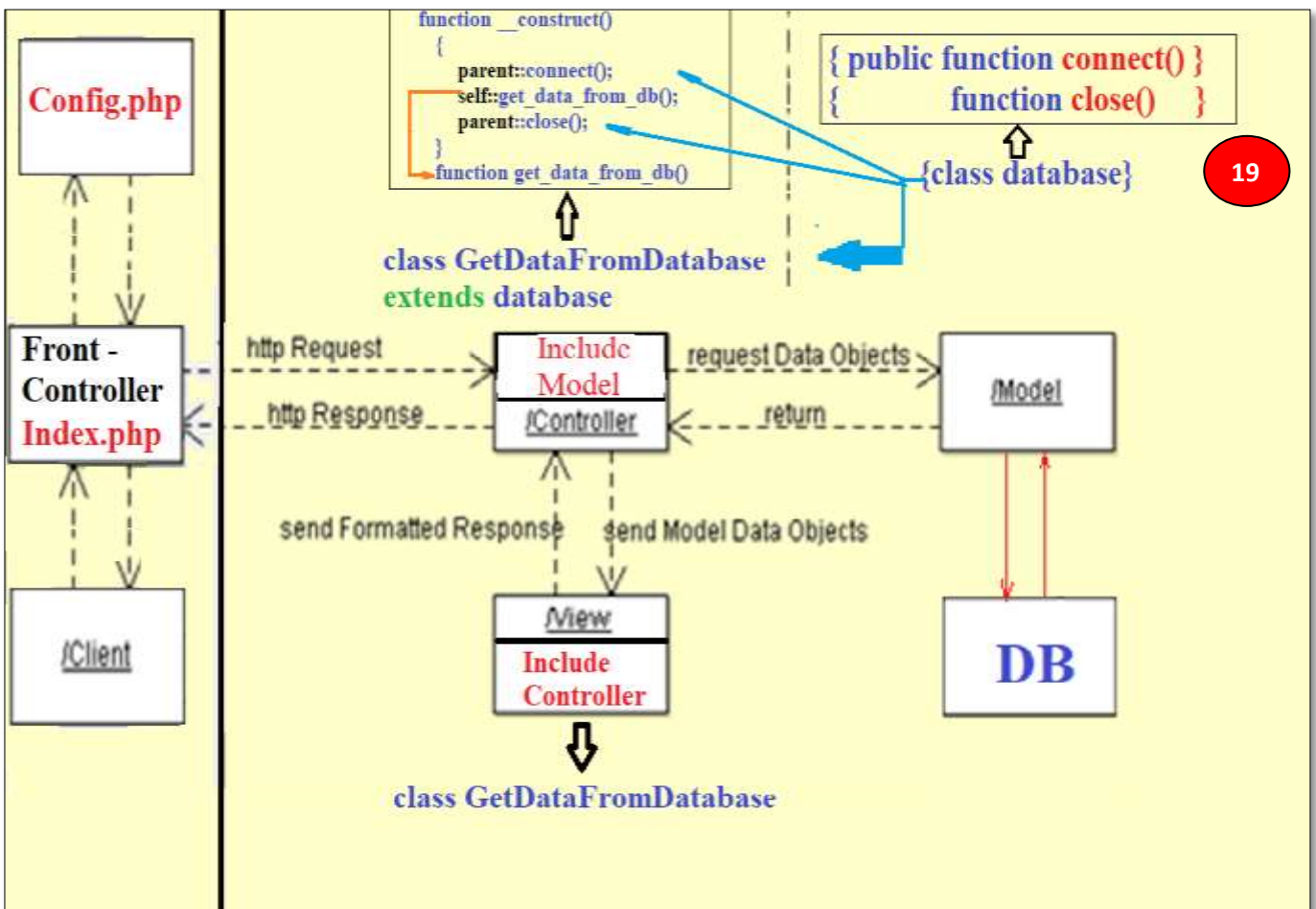
```

17

Rezultatul lansarii aplicatiei obtinute prin modelul MVC, cu ajutorul fisierului index.php, este...

Num	Grupa
Tutuianu Eugenia	MI 181
Gherman Catalin	MI 181
Bejenar Serghei	MI 181
Raileanu Daniel	MI 181
Raileanu Mihail	MI 181

Daca am utilizeze diagram de flux, atunci schematic pentru exemplul nostrum de mai sus modelul MVC poate fi prezentat dupa cum urmeaza:



Material ajutator

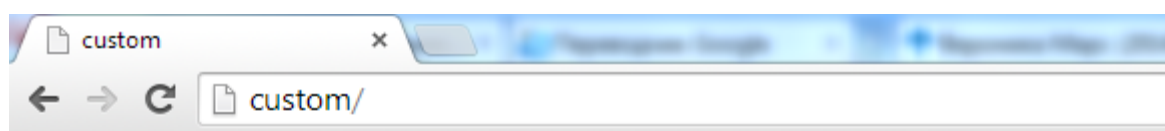
Пример использования циклов: отрисовка HTML таблицы на PHP

<http://site-on.net/create/php/10-html-table-php>

Здравствуйтесь уважаемые читатели блога **Site on!** В прошлой статье мы рассмотрели всё о **циклах** в PHP, но для закрепления полученных теоритических материалов, предлагаю выполнить практическую задачу, которая довольно часто встречается в повседневной жизни разработчика сайтов.

В наших с вами любимых CMS для отрисовки любой HTML таблицы используются циклы, по-другому никак. Например, у товара (смартфона) есть характеристики: ширина, высота, глубина, цвет, размер дисплея, объём памяти и тд. Все эти свойства обычно отрисовываются в виде HTML таблицы, это очень удобно и хорошо, ровно выглядит.

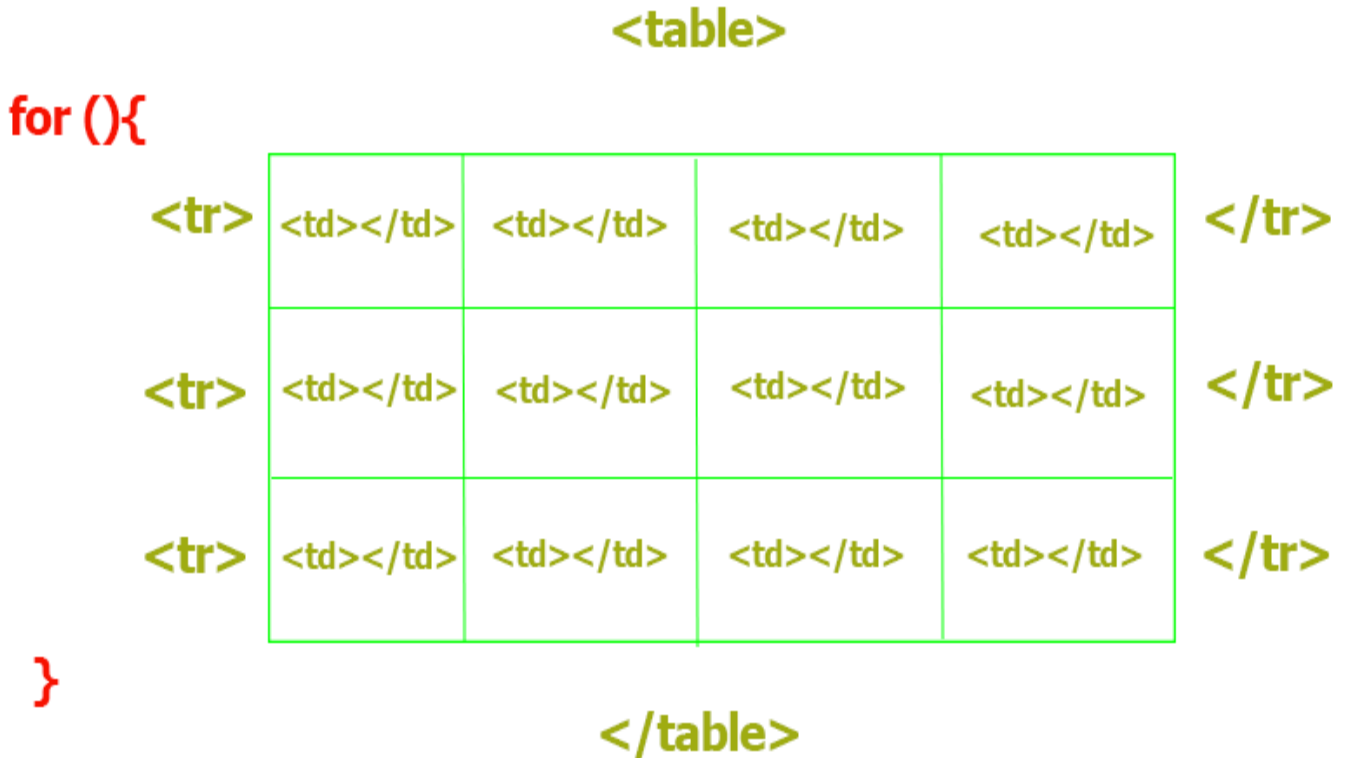
Чтобы понять, как это делается, сегодня мы решим такую простую задачу, как создание таблицы умножения:



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80
5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	102	108	114	120
7	14	21	28	35	42	49	56	63	70	77	84	91	98	105	112	119	126	133	140
8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144	152	160
9	18	27	36	45	54	63	72	81	90	99	108	117	126	135	144	153	162	171	180
10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200
11	22	33	44	55	66	77	88	99	110	121	132	143	154	165	176	187	198	209	220
12	24	36	48	60	72	84	96	108	120	132	144	156	168	180	192	204	216	228	240
13	26	39	52	65	78	91	104	117	130	143	156	169	182	195	208	221	234	247	260
14	28	42	56	70	84	98	112	126	140	154	168	182	196	210	224	238	252	266	280
15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255	270	285	300
16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	256	272	288	304	320
17	34	51	68	85	102	119	136	153	170	187	204	221	238	255	272	289	306	323	340
18	36	54	72	90	108	126	144	162	180	198	216	234	252	270	288	306	324	342	360
19	38	57	76	95	114	133	152	171	190	209	228	247	266	285	304	323	342	361	380
20	40	60	80	100	120	140	160	180	200	220	240	260	280	300	320	340	360	380	400

Проступим

Для начала нужно придумать и понять алгоритм, которому мы будем следовать. Из чего состоит HTML таблица? Давайте я нарисую:



Как видим из рисунка, тег table встречается всего 1 раз, поэтому он явно должен быть вне цикла. Повторяется у нас только строка (tr) и внутри неё ячейки, они же столбцы (td). Всё, больше у нас ничего не повторяется.

Теперь нужно решить, какой цикл лучше использовать для данной задачи. Это классический вариант и лучшим решением будет цикл for, хотя можно использовать и while, если вы к нему привыкли больше. Но нам понадобится не один цикл, а два: первый будет рисовать и считать строки (tr), а второй столбцы (td).

```
<?php
$rows = 20; // количество строк, tr
$cols = 20; // количество столбцов, td

echo '<table border="1">';

for ($tr=1; $tr<=$rows; $tr++){ // в этом цикле счётчик $tr
    // следит за количеством строк и всегда равен текущему номеру строки.
    // То есть в начале $tr=1, так как в начале у нас 1 строка, затем
    // каждый раз прибавляем единицу, пока не дойдём до заданного количества
    // $rows.
    echo '<tr>';
    for ($td=1; $td<=$cols; $td++){ // в этом цикле счётчик $td аналогичен
        // счётчику $tr.
```

```

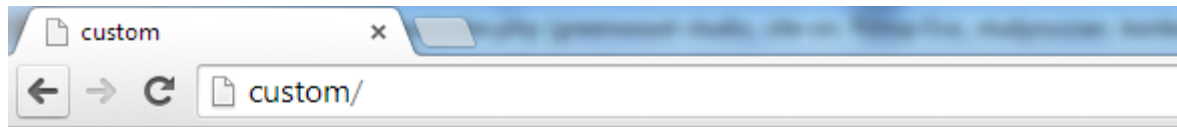
    echo '<td>'. $str*$std .'</td>';
  }
  echo '</tr>';
}

echo '</table>';

?>

```

Готово! Результат:



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80
5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	102	108	114	120
7	14	21	28	35	42	49	56	63	70	77	84	91	98	105	112	119	126	133	140
8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144	152	160
9	18	27	36	45	54	63	72	81	90	99	108	117	126	135	144	153	162	171	180
10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200
11	22	33	44	55	66	77	88	99	110	121	132	143	154	165	176	187	198	209	220
12	24	36	48	60	72	84	96	108	120	132	144	156	168	180	192	204	216	228	240
13	26	39	52	65	78	91	104	117	130	143	156	169	182	195	208	221	234	247	260
14	28	42	56	70	84	98	112	126	140	154	168	182	196	210	224	238	252	266	280
15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255	270	285	300
16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	256	272	288	304	320
17	34	51	68	85	102	119	136	153	170	187	204	221	238	255	272	289	306	323	340
18	36	54	72	90	108	126	144	162	180	198	216	234	252	270	288	306	324	342	360
19	38	57	76	95	114	133	152	171	190	209	228	247	266	285	304	323	342	361	380
20	40	60	80	100	120	140	160	180	200	220	240	260	280	300	320	340	360	380	400

В первом цикле мы отрисовываем tr, внутри него td. Этот код можно было бы назвать идеальным шаблоном отрисовки любой таблицы, но я знаю, как сделать ещё лучше. Незачем каждый раз напрягать РНР и делать вывод на экран (echo) после каждой итерации, гораздо лучше поместить всё в одно место (переменную) и сделать echo в самом конце, один единственный раз:

```
<?php
```

```
$rows = 20; // количество строк, tr
```

```

$cols = 20; // количество столбцов, td

$stable = '<table border="1">';

for ($str=1; $str<=$rows; $str++){
    $stable .= '<tr>';
    for ($std=1; $std<=$cols; $std++){
        $stable .= '<td>'. $str*$std .'</td>';
    }
    $stable .= '</tr>';
}

$stable .= '</table>';
echo $stable; // сделали эхо всего 1 раз

?>

```

Для этого мы использовали оператор `.=` присвоение через **конкатенацию**. Результат будет точно таким же. Но это ещё не всё. Нам нужно привести нашу таблицу к виду, как на самом первом рисунке в этой статье. То есть сделать первую строку и столбец полужирным и поставить зелёный фон. Такого результата можно добиться двумя способами:

- С помощью CSS3 (правильный способ);
- С помощью PHP (неправильный способ, но возьмём его, так как в этом разделе учим PHP);

Итак, делаем с помощью PHP:

```

<?php

$rows = 20; // количество строк, tr
$cols = 20; // количество столбцов, td

$stable = '<table border="1">';

for ($str=1; $str<=$rows; $str++){
    $stable .= '<tr>';
    for ($std=1; $std<=$cols; $std++){
        if ($str===1 or $std===1){
            $stable .= '<th style="color:white;background-color:green;">'. $str*$std .'</th>'; // вычислили
первую строку или столбец
        }else{
            $stable .= '<td>'. $str*$std .'</td>'; // все ячейки, кроме ячеек из первого столбца и первой
строки
        }
    }
    $stable .= '</tr>';
}

$stable .= '</table>';
echo $stable; // сделали эхо всего 1 раз

?>

```

Красота :) Можете выводить таблицу хоть 100 на 100.