ОСНОВЫ PHP

<?php

echo "Привет, Мир! <br> ";

echo '<h1> Привет, Мир! </h1>';

echo "<hr>"; //разделительная линия

/\* многострочные комменты\*/

# однострочные #

// автоопределение типа

//переменные и операции над ними

$x = 3.0;

$str = "hello";

$a = 12e-4;

$x1 = $x+5;

$x2 = $x1 + $x; // - \* / % остаток

$x++; // --

$x+=1; // \*= /= -=

$a==1; //сравнение

$a!=1; //сравнение

$a<>1; //сравнение

//вывод текста и переменной

// экранирование символов \"

// строка "..." может содержать переменные

// строка '...' не может содержать переменные

// ' '.' ' - конкатенация строк

echo 'x='.$x.'<br>'; //

echo "x= {$x} <br>";

print("Интерполированная строка х = {$x}");

echo "<hr>";

$str1 = " <br>первая строка

вторая строка

третья строка<br> ";

echo nl2br($str1);

//посмотреть тип переменной и значение

echo "<hr>";

echo var\_dump($x)."<br>";

echo var\_dump($str).'<br>' ;

echo var\_dump($a);

echo "<hr>";

//условный оператор

//проверка существования переменной

if (isset($a)) echo "{$a} <br>";

else {echo "не существует <br>";}

if (!isset($a1)) echo " нет переменной<br>";

echo "<hr>";

//тернарный условный оператор

$b=1;

echo $b>1?"More then":"Less then".'<br>';

$c=$b>1?3:2;

//логические операторы

// и - and

// или - or ||

// не - !

//массивы и циклы

$ar = array(1,2,3,4,5);

$fruit = array('apple','orange','lime');

$fruit[]='cherry'; // добавление в конец

unset($ar[4]); //удаление элемента

$N = count($ar); // количество элементов

$N1 = sizeof($ar); // количество элементов

//ассоциативные массива

$customer=array('name'=>'Ben', "age"=>"35", "sex"=>"m");

$customer["tel"]='234'; // добавление в конец

//циклы для обхода массивов

foreach ($fruit as $it)

{

echo "{$it} <br>";

}

foreach ($customer as $key=>$value)

{

echo "{$key} = {$value} <br>";

}

for ($i=0; $i<sizeof($ar); $i++)

echo ' '.$ar[$i];

echo '<br>';

$j=1;

while ($j<5)

{

echo $j;

$j++;

}

echo '<br>';

//do {...} while (условие)

// многомерные массивы

$custom = array();

$custom[]=array('name'=>'Ben', "age"=>"35", "sex"=>"m");

$custom[]=array('name'=>'Juan', "age"=>"30", "sex"=>"m");

$custom[]=array('name'=>'Selena', "age"=>"20", "sex"=>"f");

// $custom[1]['name']; обращение к элементу

foreach ($custom as $cus)

{

foreach ($cus as $key=>$val)

echo "{$key} = {$val} ";

echo '<br>';

}

//функции

Функция может возвращать некоторое значение - число, строку и т.д., то есть некоторый результат. Для возвращения значения в функции применяется оператор **return**, после которого указывается возвращаемое значение.

function F($a, $b)

{ //global $c;

$a++;

echo "Функция выводит переменные ".$a .' '. $b.'<br>';

return $a+$b;

}

echo F(12,10);

?>

<?php

function displayInfo($name, $age)

{

    echo "<div>Имя: $name <br />Возраст: $age</div><hr>";

}

displayInfo("Tom", 36);

displayInfo("Bob", 39);

displayInfo("Sam", 28);

?>

### Необязательные параметры

Выше при определении функции мы были обязаны передать для всех параметров функции значения. Например, если функция определяет два параметра, соответственно нам надо передать в ее вызов два значения. Однако, PHP позволяет сделать параметры необязательными. Такие параметры имеют значение по умолчанию, которое применяется, если при вызове функции не передано никакого значения.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | <?php  function displayInfo($name, $age = 18)  {      echo "<div>Имя: $name <br />Возраст: $age</div><hr>";  }    displayInfo("Tom", 36);  displayInfo("Sam"); |

?>

### Переменное количество параметров

В PHP функция может принимать переменное количество параметров. Для этого у функции определяется один параметр, перед которым указывается оператор **...** (три точки). Такой параметр рассматривается как массив:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  1  1  1  1 | <?php  function sum(...$numbers)  {      $result = 0;      foreach($numbers as $number) {          $result += $number;      }      echo "<p>Сумма: $result</p>";  }  sum(1, 2, 3);  sum(2, 3);  sum(4, 5, 8, 10);  ?> |

При обращении к подобной функции мы можем передавать в нее различное количество значений. Результат:

Сумма: 6

Сумма: 5

Сумма: 27

Если функция должна принимать и другие параметры, то параметр, который представляет переменное количество значений, указывается в конце после остальных параметров.

Например, определим функцию, которая принимает имя студента и неопределенное количество его баллов успеваемости и выводит средний балл:

|  |  |
| --- | --- |
|  | function getAverageScore($name, ...$scores)  {      $result = 0.0;      foreach($scores as $score) {          $result += $score;      }      $result = $result / count($scores);      echo "<p>$name: $result</p>";  }  getAverageScore("Tom", 5, 5, 4, 5);  getAverageScore("Bob", 4, 3, 4, 4, 4); |

<html>

<head>

<title>test</title>

<meta charset="utf-8" />

</head>

<body>

</body>

</html>

ОСНОВНЫЕ ФУНКЦИИ PHP

ДЛЯ РАБОТЫ С MySQL

PHP поддерживает работу с базой данных MySQL.  
Специальные встроенные функции для работы с MySQL позволяют просто и эффективно работать с этой СУБД: выполнять любые запросы, читать и записывать данные, обрабатывать ошибки.

Сценарий, который подключается к БД, выполняет запрос и показывает результат, будет состоять всего из нескольких строк. Для работы с MySQL не надо ничего дополнительно устанавливать и настраивать; всё необходимое уже доступно вместе со стандартной поставкой PHP.

**Что такое mysqli?**

mysqli (MySQL Improved) — это расширение PHP, которое добавляет в язык полную поддержку баз данных MySQL. Это расширение поддерживает множество возможностей современных версий MySQL.

**Как выглядит работа с базой данных**

Типичный процесс работы с СУБД в PHP-сценарии состоит из нескольких шагов:

1. Установить подключение к серверу СУБД, передав необходимые параметры: адрес, логин, пароль.
2. Убедиться, что подключение прошло успешно: сервер СУБД доступен, логин и пароль верные и так далее.
3. Сформировать правильный SQL запрос (например, на чтение данных из таблицы).
4. Убедиться, что запрос был выполнен успешно.
5. Получить результат от СУБД в виде массива из записей.
6. Использовать полученные записи в своём сценарии (например, показать их в виде таблицы).

**Функция mysqli connect: соединение с MySQL**

Перед началом работы с данными внутри MySQL, нужно открыть соединение с сервером СУБД.

В PHP это делается с помощью стандартной функции mysqli\_connect(). Функция возвращает результат — ресурс соединения. Данный ресурс используется для всех следующих операций с MySQL.

Но чтобы выполнить соединение с сервером, необходимо знать как минимум три параметра:

* Адрес сервера СУБД;
* Логин;
* Пароль.

Если вы следовали стандартной процедуре установки MySQL или используете OpenServer, то адресом сервера будет localhost, логином — root. При использовании OpenServer пароль для подключения — это пустая строка ‘’, а при самостоятельной установке MySQL пароль вы задавали в одном из шагов мастера установки.

Базовый синтаксис функции mysqli\_connect():

mysqli\_connect(<адрес сервера>, <имя пользователя>, <пароль>, <имя базы данных>);

**Проверка соединения**

*Первое, что нужно сделать после соединения с СУБД — это выполнить проверку, что оно было успешным.*

Эта проверка нужна, чтобы исключить ошибку при подключении к БД. Неверные параметры подключения, неправильная настройка или высокая нагрузка заставит MySQL отвеграть новые подключения. Все эти ситуации приведут к невозможности соединения, поэтому программист должен проверить успешность подключения к серверу, прежде чем выполнять следующие действия.

Соединение с MySQL устанавливается один раз в сценарии, а затем используется при всех запросах к БД.

Результатом выполнения функции mysqli\_connect() будет значение специального типа — ресурс.

Если подключение к MySQL не удалось, то функция mysqli\_connect() вместо ресурса вернет логическое значение типа «ложь» — false.

Хорошей практикой будет всегда проверять значение результата выполнения этой функции и сравнивать его с ложью.

Соединение с MySQL и проверка на ошибки:

<?php

$db = mysqli\_connect("localhost", "root", "");

if ($db == false){

print("Ошибка: Невозможно подключиться к MySQL " . mysqli\_connect\_error());

}

else {

print("Соединение установлено успешно");

}

Функция mysqli\_connect\_error() просто возвращает текстовое описание последней ошибки MySQL.

Подключение к базе удобнее вынести в отдельный файл, например, connect.php и далее подключать его в основных php файлах

содержимое файла Connect.php

<?php

$host = "localhost";

$username = "root";

$password = "";

$database = "ychet";

$db = new mysqli($host, $username, $password, $database);

// или так

// $db = mysqli\_connect($host, $username, $password, $database);

if ($db == false)

{

print("Ошибка: Невозможно подключиться к MySQL " . mysqli\_connect\_error());

}

else

{

print("Соединение установлено успешно");

}

?>

Содержимое файла \*.php в котором идет обращение к базе (как подключить файл connect.php)

<?php

require 'connect.php';

/////дальнейшая работа с базой через переменную $db

?>

или так можно подключить любой блок

<?php include "menu.php"; ?>

**Установка кодировки**

Первым делом после установки соединения крайне желательно явно задать кодировку, которая будет использоваться при обмене данными с MySQL. Если этого не сделать, то вместо записей со значениями, написанными кириллицей, можно получить последовательность из знаков вопроса: ‘?????????????????’.  
Вызови эту функцию сразу после успешной установки соединения:

mysqli\_set\_charset($con, "utf8");

ОСНОВНЫЕ ФУНКЦИИ ДЛЯ РАБОТЫ С БД

MySQLi предоставляет два способа взаимодействия с базами данных: объектно-ориентированный и процедурный.

1. Для подключения к базе

$db = mysqli\_connect("localhost", "root", ""); //процедурный стиль

if (!$db) {

die("Connection failed: " . mysqli\_connect\_error());

}

или

$db = new mysqli("адрес\_сервера", "имя\_пользователя", "пароль"); // объектно-ориентированный стиль

if ($db->connect\_error)

{

die("Connection failed: " . $db->connect\_error);

}

2. Для выполнения запросов

Процедурный способ обращения

$sql = "INSERT INTO Users (name, age) VALUES

('Sam', 41),

('Bob', 29),

('Alice', 32)";

$result = mysqli\_query($db, $sql);

Если выполнение запроса прошло успешно, то функция mysqli\_query() возвращает true, если неудачно, то возвращается false.

ОО стиль

$result = $db->query($sql);

3. Функция mysqli\_insert\_id после срабатывания операции вставки в таблицу возвращает идентификатор последней добавленной записи.

$city\_id = mysqli\_insert\_id($db);

4. Количество полученных строк

mysqli\_num\_rows($result);

$rowsCount = $result->num\_rows;

5. Выщипление строки из таблицы результата запроса

$row = mysqli\_fetch\_array($result) //процедурный стиль

$row = $result->fetch\_array() //ОО стиль

6. После окончания работы с полученным набором строк мы можем очистить отведенную для него память с помощью функции mysqli\_free\_result(), в которую передается полученный набор строк

mysqli\_free\_result($result);

$result->free();

7. Закрыть соединение с базой

mysqli\_close($db);

$db->close();

ПРИМЕР

1. Cоздадим в PHPMyAdmin базу

CREATE TABLE Users (id INTEGER AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(30), age INTEGER);

*2. Добавление данных из формы HTML*

В большинстве случаев добавляемые данные будут приходить из вне, например, присылаться в запросе пользователя. Рассмотрим добавление данных, отправленных из формы HTML. Для этого определим веб-страницу form.html, на которой определим следующий код:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  1  1  1  1  1  1 | <!DOCTYPE html>  <html>  <head>  <title>test</title>  <meta charset="utf-8" />  </head>  <body>  <h3>Добавление пользователя</h3>  <form action="index.php" method="POST">      <p>Имя:      <input type="text" name="username" /></p>      <p>Возраст:      <input type="number" name="userage" /></p>      <input type="submit" value="Добавить">  </form>  //обработчик  <?php  if (isset($\_POST["username"]) && isset($\_POST["userage"]))  {  require 'connect.php'; //соединяемся с базой  $name = $db->real\_escape\_string($\_POST["username"]);  $age = $db->real\_escape\_string($\_POST["userage"]);  $sql = "INSERT INTO Users (name, age) VALUES ('$name', $age)";  if($db->query($sql))  {  echo "Данные успешно добавлены";  }  else  {  echo "Ошибка: " . $conn->error;  }  $conn->close();  }  ?>  </body>  </html>  !!!! перед добавлением данных к ним применяется метод **$conn->real\_escape\_string()**, которая принимает сохраняемое значение и экранирует в нем спецсимволы, что позволяет защитить от SQL-инъекций. |

Здесь определены два поля ввода. И по нажатию на кнопку их данные в запросе POST будут уходить скрипту index**.php**. Теперь определим сам скрипт index**.php**.

*3. Вывод данных из базы*

$sql = "SELECT \* FROM Users";

if($result = $db->query($sql)) //OO стиль

// if($result = mysqli\_query($conn, $sql)) //процедурный стиль

{

    foreach($result as $row)

{

        $userid = $row["id"];

        $username = $row["name"];

        $userage = $row["age"];

echo $userid .' '. $username .' '. $userage.'<br>';

}

}

Другой вариант вывода

$sql = "SELECT \* FROM Users";

if($result = $db->query($sql))

//if($result = mysqli\_query($conn, $sql))

{

    while($row = $result->fetch\_array())

//while($row = mysqli\_fetch\_array($result)) //процедурный стиль

{

        $userid = $row["id"];

        $username = $row["name"];

        $userage = $row["age"];

    }

}

После окончания работы с полученным набором строк мы можем очистить отведенную для него память с помощью метода free():

|  |  |
| --- | --- |
|  | $result->free(); |

Рассмотрим вариант, когда результаты выводятся в таблицу

<!DOCTYPE html>

<html>

<head>

<title>METANIT.COM</title>

<meta charset="utf-8" />

</head>

<body>

<h2>Список пользователей</h2>

<?php

require 'connect.php'; //соединяемся с базой

$sql = "SELECT \* FROM Users";

if($result = $db->query($sql)){

    $rowsCount = $result->num\_rows; // количество полученных строк

    echo "<p>Получено объектов: $rowsCount</p>";

    echo "<table><tr><th>Id</th><th>Имя</th><th>Возраст</th></tr>";

    foreach($result as $row){

        echo "<tr>";

            echo "<td>" . $row["id"] . "</td>";

            echo "<td>" . $row["name"] . "</td>";

            echo "<td>" . $row["age"] . "</td>";

        echo "</tr>";

    }

    echo "</table>";

    $result->free();

} else{

    echo "Ошибка: " . $db->error;

}

$db->close();

?>

</body>

</html>

4. Можно унифицировать процедуру добавления, удаления и редактирования данных из базы, например создать файл functions.php

в блоке **head** подключить

<head>

<?php

include "dbconnect.php";

include "functions.php";

?>

</head>

// вызвать функции в основном блоке

<body>

....

<?php

if (isset($\_POST['filt']))

{

$filt\_cat = isset($\_POST['filt\_category'])?$\_POST['filt\_category']:'%';

$filt\_price1 = $\_POST['price1'];

$filt\_price2 = $\_POST['price2'];

$sql1 = "SELECT \* FROM `prod` WHERE `category` like '$filt\_cat' and `price` BETWEEN $filt\_price1 AND $filt\_price2";

View\_all\_tab\_by\_sql($sql1, "Продукты");

}

?>

....

</body>

**functions.php**

<?php

function View\_all\_tab($tab\_name, $caption)

{ global $db;

$sql = "SELECT \* FROM `$tab\_name`";

$result = mysqli\_query($db, $sql);

$row = mysqli\_fetch\_array($result);

$col\_number = mysqli\_num\_rows($result);

for ($i=0; $i<$col\_number; $i++)

{

echo "$row[0] $row[1] $row[2] <br>";

$row = mysqli\_fetch\_array($result);

}

}

function View\_all\_tab1($tab\_name, $tab\_caption)

{ global $db;

$sql3 = "SELECT \* FROM `$tab\_name`";

$result3 = mysqli\_query($db, $sql3);

// $count = mysqli\_num\_rows($result);

$row = mysqli\_fetch\_array($result3);

$col\_number = count($row);

echo " <caption> <h3> {$tab\_caption} </h3> </caption> <table style='width:

. "."400px; font-size: 12pt; border: 0px solid #999; background-color: "

. "#ffffff; '> ";

do

{

echo "<tr >";

for ($i=0; $i < 3; $i++)

{ echo "<th>$row[$i]".'</th>';}

echo "<th><a href=\"delete.php?id=$row[0]\"> Del </a> </th><th>"

. "<a href=\"edit\_prod.php?edit\_id=$row[0]\"> Edit </a></th></tr>";

}

while ($row = mysqli\_fetch\_array($result3));

}

function View\_all\_tab\_by\_sql($sql\_f, $tab\_caption)

{ global $db;

$result3 = mysqli\_query($db, $sql\_f);

// $count = mysqli\_num\_rows($result);

// echo "<br>".$sql\_f;

$row = mysqli\_fetch\_array($result3);

$col\_number = count($row);

echo " <caption> <h5> {$tab\_caption} </h5> </caption> <table style='width:

. "."400px; font-size: 12pt; border: 0px solid #999; background-color: "

. "#ffffff; '> ";

do

{

echo "<tr >";

for ($i=0; $i < 3; $i++)

{ echo "<th>$row[$i]".'</th>';}

echo "<th><a href=\"delete.php?id=$row[0]\"> Купить</a></th></tr>";

}

while ($row = mysqli\_fetch\_array($result3));

}

//вызов

View\_all\_tab("Users", "Пользователи");

Связные таблицы. Заполнение

Создадим новую таблицу Заказы (id\_s, id\_u, date) (Sales)

поле id\_u заполняется из данных таблицы Users

<form action="sales.php" method="POST">

<br><!-- comment -->

<br>

<center>

<select name="users\_select">

<option selected disabled>Выберите поставщика</option>

<?php

$sql = "SELECT \* FROM `Users`";

$result = mysqli\_query($db, $sql);

$row = mysqli\_fetch\_array($result);

do{

printf("<option value='%s'> %s </option>", $row[0], $row[1]);

}

while ($row = mysqli\_fetch\_array($result));

?>

</select>

<input type="submit" value="search" name="search\_customer" />

</form>

<br> <br>

<?php

if (isset($\_POST['users\_select']))

{

$id\_users = $\_POST['users\_select'];

$sql = "INSERT INTO Sales (id\_u, date) VALUES ('$id\_users ', '21.04.22')";

$result = mysqli\_query($db, $sql);

?>

**ЗАГОТОВКИ К ЛЕКЦИИ**

////////////////////////////подключение базы

<?php

$host = "localhost";

$username = "root";

$password = "";

$database = "test";

$db = new mysqli($host, $username, $password, $database);

//$db = mysqli\_connect($host, $username, $password, $database);

?>

**ДЕЛАЕМ ВЫПАДАЮЩИЙ СПИСОК**

/////////////////////////заглушка выпадающего списка

<select name="custom\_select">

<option selected disabled>Выберите поставщика</option>

<option value="Чебурашка">Чебурашка</option>

<option value="Крокодил Гена">Крокодил Гена</option>

<option value="Шапокляк">Шапокляк</option>

<option value="Крыса Лариса">Крыса Лариса</option>

</select>

/////////////////////одно значение вывели

<center>

<select name="custom\_select">

<option selected disabled>Выберите поставщика</option>

<?php

$sql = "Select \* from `customers`";

$result = mysqli\_query($db, $sql);

$row = mysqli\_fetch\_array($result);

?>

<option value="<?php echo "Utyf"; ?>"> <?php echo "$row[1]"; ?> </option>

<option value="Крокодил Гена">Крокодил Гена</option>

<option value="Шапокляк">Шапокляк</option>

<option value="Крыса Лариса">Крыса Лариса</option>

</select>

</center>

//////////////////////// окончательный вариант выпадающего списка

<select name="custom\_select">

<option selected disabled>Выберите поставщика</option>

<?php

$sql = "Select \* from `customers`";

$result = mysqli\_query($db, $sql);

$row = mysqli\_fetch\_array($result);

do{

printf("<option value='%s' width='250px'>%s </option>",$row[0],$row[1]);

}

while($row = mysqli\_fetch\_array($result));

?>

</select>

**ДЕЛАЕМ КНОПКУ К ВЫПАДАЮЩЕМУ СПИСКУ**

<form action "customer.php" method="post">

<select name="custom\_select">

<option selected disabled>Выберите поставщика</option>

<?php

$sql = "Select \* from `customers`";

$result = mysqli\_query($db, $sql);

$row = mysqli\_fetch\_array($result);

do{

printf("<option value='%s' width='250px'>%s </option>",$row[0],$row[1]);

}

while($row = mysqli\_fetch\_array($result));

?>

</select>

<input type="submit" value="search" name="search\_customer" />

</form>

**ВЫВОДИМ ВСЕ ЗАКАЗЫ ВЫБРАННОГО ПОСТАВЩИКА**

<?php

if (isset($\_POST['custom\_select']))

{

//определим имя поставщика

$id=$\_POST['custom\_select'];

$sql1 = "SELECT \* FROM `customers` WHERE `id\_c`=$id";

$result = mysqli\_query($db, $sql1);

$cust\_name\_from\_id = mysqli\_fetch\_array($result);

echo "<caption> Заказы поставщика {$cust\_name\_from\_id['name']} </caption> <table style='width: 400px; font-size: 12pt; border: 2px solid #999; background-color: #DCDCDC; border-radius: 5px;'> ";

//запрос к таблице Заказы

$sql1 = "SELECT \* FROM `sales` WHERE `id\_c\_fk`=$id";

$result = mysqli\_query($db, $sql1);

while ($row = mysqli\_fetch\_array($result))

{

echo "<tr style='border-radius: 5px;'> <th> {$row[0]} &nbsp </th> <th> {$row[1]} </th> <th> {$row[2]} </th> <th> {$row[3]}</th> </tr>";

}

echo "</table>";

}

?>

**ВЫВОДИМ ВСЕ ПОЛЯ ТАБЛИЦЫ**

**//просто выводим поля**

function View\_all\_tab($tab\_name, $tab\_caption)

{ global $db;

$sql3 = "SELECT \* FROM `$tab\_name`";

$result3 = mysqli\_query($db, $sql3);

$row = mysqli\_fetch\_array($result3);

$col\_number = mysqli\_num\_rows($result3);

for ($i=0; $i < $col\_number; $i++)

{ echo "$row[0] $row[1] $row[2] $row[3] <br>";

$row = mysqli\_fetch\_array($result3);

}

}

//вызов из программы

<div><!-- блок просмотра всех поставщиков -->

<br>

<?php View\_all\_tab("customers", "ПОСТАВЩИКИ"); ?>

</div>

**//выведем в виде таблицы**

function View\_all\_tab($tab\_name, $tab\_caption)

{ global $db;

$sql3 = "SELECT \* FROM `$tab\_name`";

$result3 = mysqli\_query($db, $sql3);

// $count = mysqli\_num\_rows($result);

$row = mysqli\_fetch\_array($result3);

$col\_number = count($row);

echo " <caption> <h5> {$tab\_caption} </h5> </caption> <table style='width:

. "."400px; font-size: 12pt; border: 0px solid #999; background-color: "

. "#ffffff; '> ";

do

{

echo "<tr >";

for ($i=0; $i < 4; $i++)

{ echo "<th>$row[$i]".'</th>';}

echo "<th><a href=\"delete.php?id=$row[0]\"> Del </a> </th><th>"

. "<a href=\"edit.php?edit\_id=$row[0]\"> Edit </a></th></tr>";

}

while ($row = mysqli\_fetch\_array($result3));

}

**//обработчик удаления delete.php?id=$row[0]**

<?php

include "dbconnect.php";

echo $\_GET['id'];

$sql3 = "DELETE FROM `customers` WHERE `customers`.`id\_c` = ".$\_GET['id'];

$result3 = mysqli\_query($db, $sql3);

if ($result3) echo "Deleted";

/////////////////////// customers.php

<center>

<form action "customer.php" method="post">

<select name="custom\_select">

<option selected disabled>Выберите поставщика</option>

<?php

$sql = "Select \* from `customers`";

$result = mysqli\_query($db, $sql);

$row = mysqli\_fetch\_array($result);

do{

printf("<option value='%s' width='250px'>%s </option>",$row[0],$row[1]);

}

while($row = mysqli\_fetch\_array($result));

?>

</select>

<input type="submit" value="search" name="searc\_customer" />

</form>

<?php

if (isset($\_POST['custom\_select']))

{

//определим имя поставщика

$id=$\_POST['custom\_select'];

$sql1 = "SELECT \* FROM `customers` WHERE `id\_c`=$id";

$result = mysqli\_query($db, $sql1);

$cust\_name\_from\_id = mysqli\_fetch\_array($result);

echo "<caption> Заказы поставщика {$cust\_name\_from\_id['name']} "

. "</caption> <table style='width: 400px; font-size: 12pt; border: "

. "2px solid #999; background-color: #DCDCDC; "

. "border-radius: 5px;'> ";

//запрос к таблице Заказы

$sql1 = "SELECT \* FROM `sales` WHERE `id\_c\_fk`=$id";

$result = mysqli\_query($db, $sql1);

while ($row = mysqli\_fetch\_array($result))

{

echo "<tr style='border-radius: 5px;'> <th> {$row[0]} &nbsp </th> <th> {$row[1]} </th> <th> {$row[2]} </th> <th> {$row[3]}</th> </tr>";

}

echo "</table>";

}

?>

<div><!-- блок просмотра всех поставщиков -->

<br>

<?php View\_all\_tab1("customers", "ПОСТАВЩИКИ"); ?>

</div>

</center>

////////////////////functions.php

function View\_all\_tab($tab\_name, $tab\_caption)

{ global $db;

$sql3 = "SELECT \* FROM `$tab\_name`";

$result3 = mysqli\_query($db, $sql3);

$row = mysqli\_fetch\_array($result3);

$col\_number = mysqli\_num\_rows($result3);

for ($i=0; $i < $col\_number; $i++)

{ echo "$row[0] $row[1] $row[2] $row[3] <br>";

$row = mysqli\_fetch\_array($result3);

}

}

function View\_all\_tab1($tab\_name, $tab\_caption)

{ global $db;

$sql3 = "SELECT \* FROM `$tab\_name`";

$result3 = mysqli\_query($db, $sql3);

// $count = mysqli\_num\_rows($result);

$row = mysqli\_fetch\_array($result3);

$col\_number = count($row);

echo " <caption> <h5> {$tab\_caption} </h5> </caption> <table style='width:

. "."400px; font-size: 12pt; border: 0px solid #999; background-color: "

. "#ffffff; '> ";

do

{

echo "<tr >";

for ($i=0; $i < 4; $i++)

{ echo "<th>$row[$i]".'</th>';}

echo "<th><a href=\"delete.php?id=$row[0]\"> Del </a> </th><th>"

. "<a href=\"edit.php?edit\_id=$row[0]\"> Edit </a></th></tr>";

}

while ($row = mysqli\_fetch\_array($result3));

}