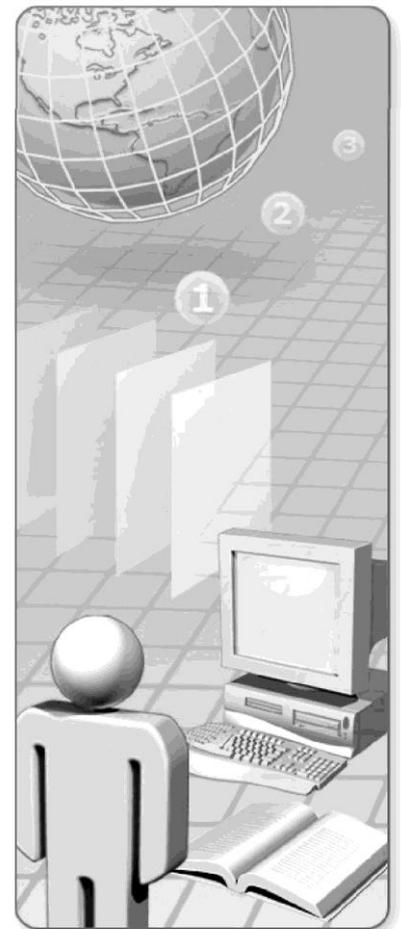


PHP. Уровень 2. Разработка web-сайтов и взаимодействие с MySQL

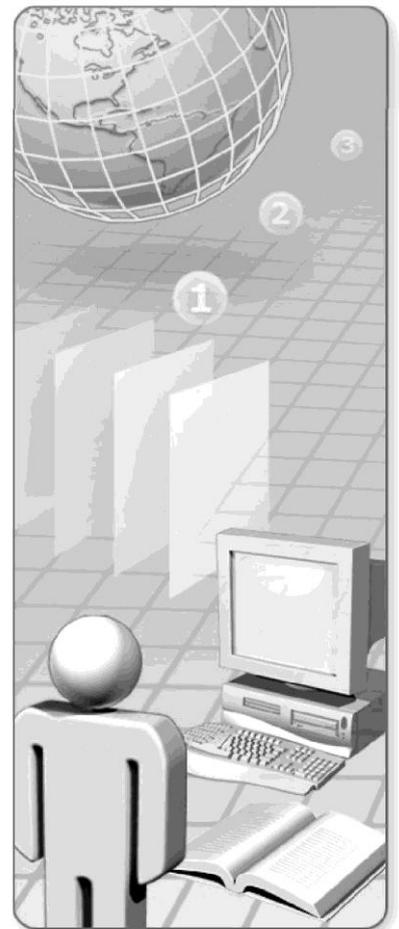


v.08.2009

Содержание курса

Модуль 1: Работа с протоколом HTTP: cookie, заголовки ответа сервера	7
Модуль 2: Сессии. Операции с файлами и директориями. Работа с почтой	21
Модуль 3: Основы работы с базами данных. Сервер баз данных MySQL	35
Модуль 4: Использование сервера баз данных MySQL в приложениях PHP	51
Модуль 5: Практическое использование сервера баз данных MySQL: создание интернет-магазина	61

Подготовка к работе



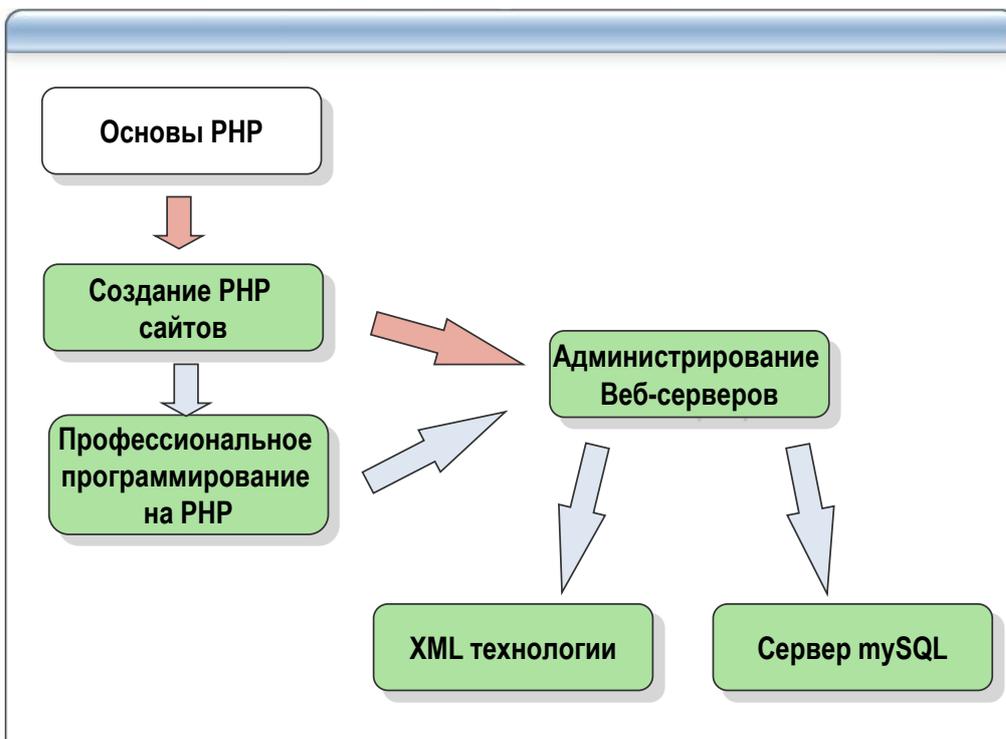
Предварительная подготовка

- Уверенная работа с компьютером
- Успешное окончание курса «PHP. Уровень 1. Основы создания сайтов»

ИЛИ

- Уметь устанавливать и настраивать сервер Apache 1.3 – 2.x и проверять его работу
- Уметь устанавливать PHP как модуль сервера Apache
- Знать язык программирования PHP на уровне программы курса "Основы создания PHP сайтов для начинающих"
- Уметь дать определения переменной, функции, управляющей конструкции
- Уметь создавать простые сценарии (скрипты) PHP 5.x

Схемы курсов



Создание виртуальной машины для Веб курса

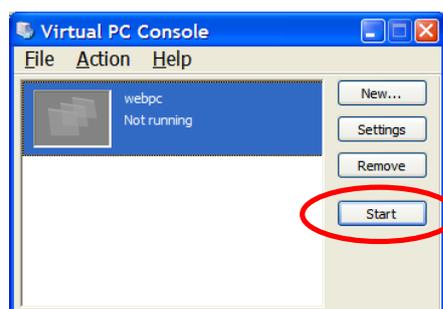
- Используется Microsoft Virtual PC
- Создается новая копия базового образа VM
- После курса машина удаляется



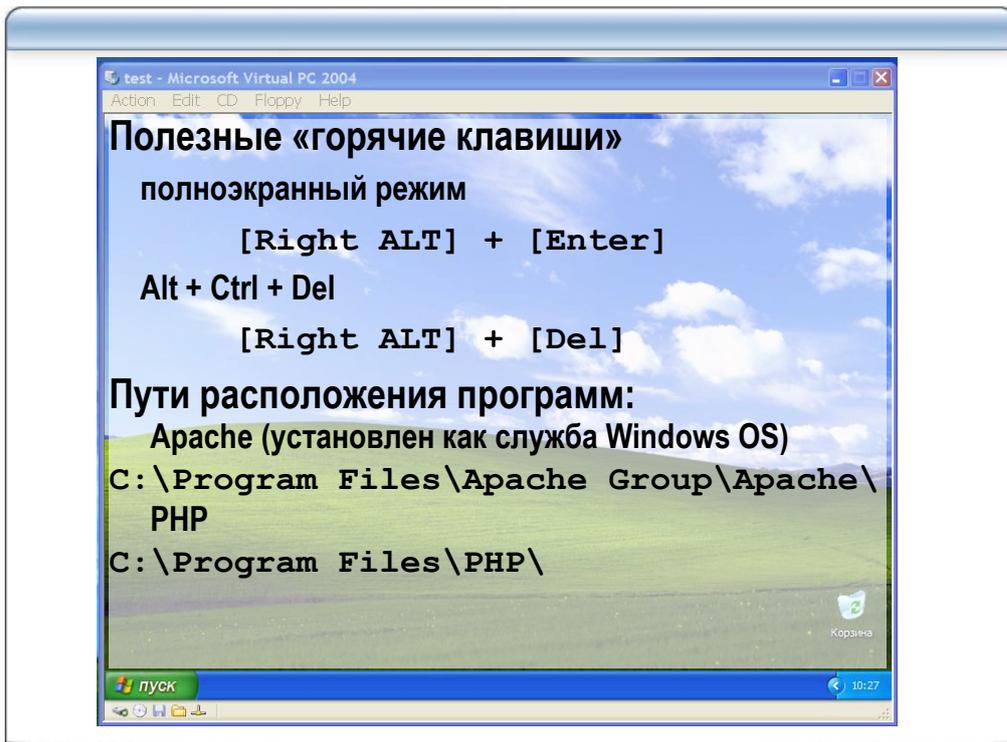
<http://www.microsoft.com/windows/virtualpc/default.mspx>

Создание и запуск виртуальной машины

- Создайте приложение Microsoft Virtual PC
Пуск → Все программы → Создать виртуальную машину
- Откройте приложение Microsoft Virtual PC
Пуск → Все программы → Microsoft Virtual PC



Виртуальная машина работает!

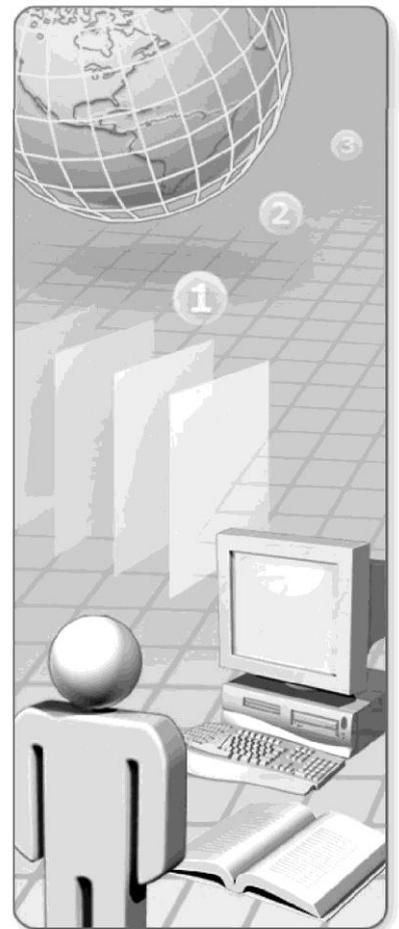


Оборудование рабочего места

1. Откройте в проводнике виртуальной машины папку
C:\Program files\Apache Group\Apache\htdocs
2. Удалите ВСЕ файлы в этой директории
3. Установите в привод CD слушателя и откройте проводником этот диск
4. В папке CD_drive:\work найдите файл work.exe
5. Скопируйте файл work.exe на Рабочий стол виртуальной машины
6. Запустите файл work.exe (самораспаковывающийся архив)
7. Нажмите кнопку «Извлечь»
8. Дождитесь распаковки архива в папку htdocs
9. **Запустите сервер Apache!**
10. Проверьте работу сервера

Работа с протоколом HTTP: cookie, заголовки ответа сервера

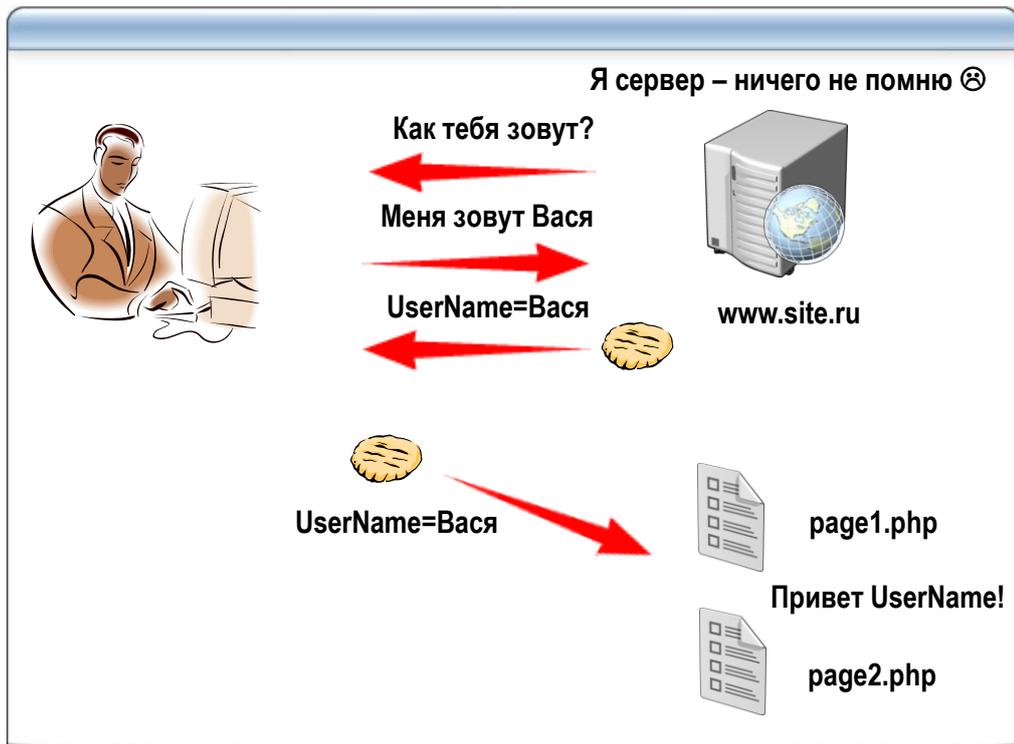
Модуль 1



Содержание

- Сессионные и постоянные куки
- Запись, чтение и удаление куки
- Перезапросы страниц (Заголовок Refresh)
- Переходы на другие страницы (Переадресация и заголовок Location)
- Установка типа содержимого (Заголовок Content-Type)
- Запрет и разрешение кэширования (заголовки Cache-Control и Expires)
- Хэширование

Cookie: как это работает?



Cookie: первый запрос страницы

```
GET /folder/index.php?name=Vasya HTTP/1.1 ↵
Host: www.specialist.ru ↵
Accept: */* ↵
Accept-Language: ru ↵
Referrer: http://yandex.ru/yandsearch?text=Rehc ↵
User-Agent: Mozilla 4.0 (compatible; MSIE 6.1,...) ↵
↵

HTTP/1.1 200 OK ↵
Server: Microsoft IIS 6 ↵
Content-Type: text/html ↵
Content-Length: 16345 ↵
Last-Modified: Sun, 03 Jul 2005 18:00:00 GMT ↵
Set-Cookie: userName=Vasya ↵
↵
```

Cookie: другие запросы страниц

```
GET /folder/index.php HTTP/1.1 ↵
Host: www.specialist.ru ↵
Accept: */* ↵
Accept-Language: ru ↵
Referrer: http://yandex.ru/yandsearch?text=Rehc ↵
User-Agent: Mozilla 4.0 (compatible; MSIE 6.1,...) ↵
Cookie: userName=Vasya ↵
↵

HTTP/1.1 200 OK ↵
Server: Microsoft IIS 6 ↵
Content-Type: text/html ↵
Content-Length: 16345 ↵
Last-Modified: Sun, 03 Jul 2005 18:00:00 GMT ↵
↵
```

Cookie: параметры

```
int setcookie (string name [, string value [,  
    int expire [, string path [, string domain  
    [, int secure]]]])
```

1. Имя куки. Только латинские буквы, цифры, символ подчеркивания и дефис. Все другие символы будут преобразованы в символ подчеркивания
2. Значение параметра
3. Дата истечения срока годности
4. Путь, который определяет, в какой части домена может использоваться данный файл cookie
5. Домен
6. Указание, что данные cookie должны передаваться только через безопасное соединение HTTPS.

Cookie: создание

```
setcookie ("TestCookie", $value);

setcookie ("TestCookie", $value,time()+3600);
/* период действия - 1 час */
setcookie ("TestCookie", $value,time()+3600,
    "/docs/", ".site.com", 1);

//Ошибка. Произведен вывод до установки cookie
<?php
    echo "Hello!";
    $color = "red";

    setcookie("BG", $color, time()+3600);
?>
```

Cookie: чтение

```
echo $_COOKIE["TestCookie"];

//Пример создания массива из разных cookies
while(list($name,$value) = each($_COOKIE)){

    // для последовательности значений
    $array[]=$value;

    // для ассоциативного массива
    $array[$name]=$value;
}
```

Cookie: массивы и cookie

```
//Создаем массив
$array = array(
    "name"=>"John",
    "login"=>"root",
    "pass"=>"p@ssw0rd");

// Упаковываем массив в строку
$str = serialize($array);

//Сохраняем массив в cookie
setcookie('user',$str, time() + 3600);

//Считываем строку и переводим в массив
$array = unserialize($_COOKIE['user']);
```

Cookie: удаление

```
setcookie("TestCookie")

setcookie("TestCookie", "")

setcookie ("TestCookie", "", time() - 3600);
```

Лабораторная работа

Использование cookie

Откройте файл `mod1\cookie.php`

Задача

- При первом запросе страницы `cookie.php` пользователем, выводить фразу «**Добро пожаловать!**»
- При повторных запросах `cookie.php` пользователем:
 - Выводить количество посещений. Например: «**Вы зашли на страницу 5 раз**»
 - Указывать дату и время последнего посещения. Например: «**Последнее посещение: 20-08-2009 16:34:45**»

Задание 1

1. Инициализируйте переменную для подсчета количества посещений
2. Если соответствующие данные передавались через куки, сохраняйте их в эту переменную
3. Нарастите счетчик посещений
4. Инициализируйте переменную для хранения значения последнего посещения страницы
5. Если соответствующие данные передавались из куки, отфильтруйте их и сохраните в эту переменную
6. Установите соответствующие куки

Задание 2

1. Выводите информацию о количестве посещений и дате последнего посещения

Запрос HEAD

```
HEAD /folder/index.html HTTP/1.1 ↵
Host: www.specialist.ru ↵
Accept: */* ↵
Accept-Language: ru ↵
Referer: http://yandex.ru/yandsearch?text=Rehc ↵
User-Agent: Mozilla 4.0 (compatible; MSIE 6.1,...) ↵
↵

HTTP/1.1 200 OK ↵
Server: Microsoft IIS 6 ↵
Content-Type: text/html ↵
Content-Length: 16345 ↵
Last-Modified: Sun, 03 Jul 2005 18:00:00 GMT
```

Header: location

```
header("Location: http://www.site.ru");
exit; /* При необходимости,
убедитесь, что последующий код не
выполняется после перенаправления */

//Проверка на отправленные заголовки
if (!headers_sent()) {
    header('Location:
                http://www.site.ru');
    exit;
}
```

Header: refresh

```
//Перегружаем страницу
header("Refresh: 3");
//Перегружаем страницу с редиректом
header("Refresh: 3;
                url=http://site.ru");
```

Header: content-type

```
header("Content-type: text/xml");

header("Content-type:
      text/html; charset=windows-1251");

//Отдаем файл пользователю
header("Content-type: text/plain");

header("Content-Disposition: attachment;
      filename=\"myfile.txt\");

/* ... ВЫВОД СОДЕРЖИМОГО ФАЙЛА ... */
```

Header: Cache-Control и Expires

<http://ru.wikipedia.org> советует:

```
header("Expires: Mon, 26 Jul 1997 05:00:00 GMT"); // disable IE
header("Last-Modified: " . gmdate("D, d M Y H:i:s") . " GMT");
header("Cache-Control: no-cache, must-revalidate");
header("Pragma: no-cache");
```

Как надо:

запрет

```
header("Cache-Control: no-store");
```

или

```
header("Cache-Control: no-store, no-cache, must-  
revalidate");
```

```
header("Expires: " . date("r"));
```

разрешение

```
header("Cache-Control: public");
```

```
header("Expires: " . date("r", time()+3600));
```

Header: Set-Cookie

```
header("Set-Cookie: name=John;  
        expires=Wed, 19 Sep 02 14:39:58 GMT");
```

Хэширование: md5

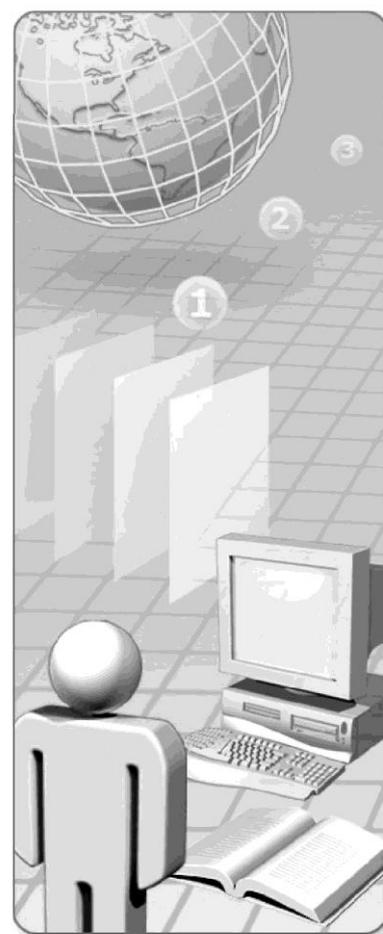
```
md5(str string)  
  
md5("888");  
//0a113ef6b61820daa5611c870ed8d5ee  
md5("Vasya");  
//96932f68a34ac08a6c92ed8db20d2ee3  
md5("MeGaPa$$w0rd");  
//bfb5a5275a34cf74cdfbdea0cf9c421
```

Выводы

- Сессионные и постоянные куки
- Запись, чтение и удаление куки
- Перезапросы страниц (Заголовок Refresh)
- Переходы на другие страницы (Переадресация и заголовок Location)
- Установка типа содержимого (Заголовок Content-Type)
- Запрет и разрешение кэширования (заголовки Cache-Control и Expires)
- Хэширование

Сессии. Операции с файлами и директориями. Работа с почтой

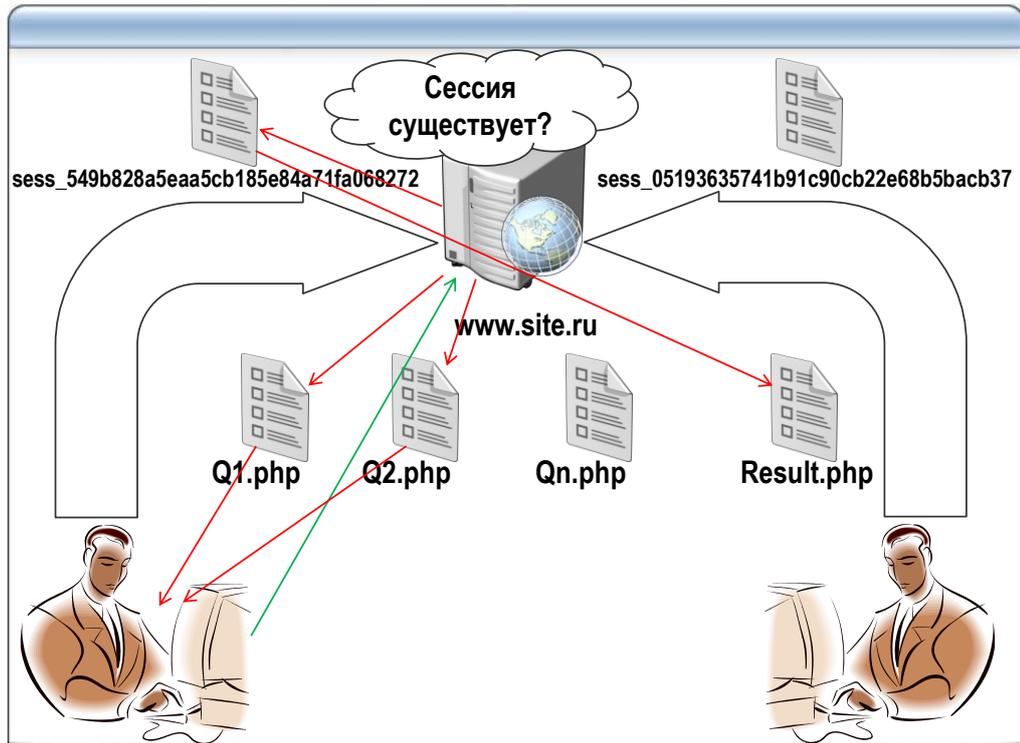
Модуль 2



Содержание

- Сессии: создание, запись, чтение, удаление
- Функции RHP для работы с файлами
- Манипуляции с файлами: копирование, переименование, удаление
- Работа с курсором
- Прямая работа с файлами
- Работа с директориями: создание, чтение, удаление
- Загрузка файлов на сервер по протоколу HTTP
- Работа с почтой

Сессии: как это работает?



Сессии: создание и использование

```
//Начало или продолжение сессии
session_start();

$_SESSION //В этом массиве всё и хранится

$_SESSION["user"] = "John";
echo $_SESSION["user"];

//Удаление
unset($_SESSION["user"]);
session_destroy();

session_id();// id сессии
session_name();// Имя сессии
```

Лабораторная работа

Использование сессий

Откройте файл `mod2\session\savepage.inc.php`

Задача

Пользователь запрашивает страницу `page1.php` и хаотично переходит со страницы на страницу, используя навигационное меню. Необходимо сохранять список посещённых страниц (добавляя их последовательно) и выводить этот список внизу после подзаголовка «Список посещённых страниц»

Задание 1

1. Создайте в сессии массив для хранения всех посещенных страниц и сохраните в качестве его очередного элемента путь к текущей странице. Вместо массива можно создать строку с уникальным разделителем и последовательно её дополнять

Откройте файл `mod2\session\visited.inc.php`

Задание 1

1. В случае сохранения данных
 - в массив, проверьте, существует ли он в сессии
 - в строку, преобразуйте её в массив
2. Выводите в цикле список всех посещенных пользователем страниц.

Страница 1

Меню

- [Страница 1](#)
- [Страница 2](#)
- [Страница 3](#)

Список посещенных страниц

1. `/test/session/page1.php`
2. `/test/session/page3.php`
3. `/test/session/page2.php`
4. `/test/session/page2.php`
5. `/test/session/page1.php`

Получение сведений о файлах

```
//Существует ли файл?  
file_exists("test.txt")  
  
//Узнаем размер файла  
filesize("test.txt");  
  
//Дата последнего обращения к файлу  
filemtime("test.txt");//date("d M Y", $mtime);  
  
//Дата изменения файла  
filemtime("test.txt");//date("d M Y", $mtime);  
  
//Дата создания файла (Windows)  
filectime("test.txt");//date("d M Y", $ctime);
```

Файлы: режимы работы

```
int fopen(string filename, string mode)
```

r – открыть файл только для чтения;
r+ – открыть файл для чтения и записи;
w – открыть файл только для записи. Если он существует, то текущее содержимое файла уничтожается. Текущая позиция устанавливается в начало;
w+ – открыть файл для чтения и для записи. Если он существует, то текущее содержимое файла уничтожается. Текущая позиция устанавливается в начало;
a – открыть файл для записи. Текущая позиция устанавливается в конец файла;
a+ – открыть файл для чтения и записи. Текущая позиция устанавливается в конец файла;
b – обрабатывать бинарный файл. Этот флаг необходим при работе с бинарными файлами в ОС Windows.

Файлы: открытие и закрытие

```
$f = fopen("test.txt", "w+") or die("Ошибка");
```

//Примеры
\$f = fopen("http://www.you_domain/test.txt", "r");
\$f= fopen("http://ftp.you_domain/test.txt", "r");

//Закрываем
fclose(\$f)

Файлы: чтение

```
//Читаем файл
fread(int f, int length)

//Читаем первые 10 символов
$str = fread($f, 10);
echo $str;
//Читаем следующие 12 символов
$str = fread($f, 12);
echo $str;

//Прочитать строку из файла
fgets(int f[, int length])

//Прочитать строку из файла и отбросить HTML-теги
fgetss(int f, int length [, string allowable])

//Считывает символ из файла
fgetc(int f)
```

Файлы: запись

```
fwrite(int f, string ws [, int
length])

fputs // Тоже, что и fwrite

//Пишем в конкретную позицию
fread($f, 7);
fwrite($f, "Наш текст");
```

Файлы: манипуляции с курсором

```
//Установка курсора
int fseek(int f, int offset [, int whence])

offset — количество символов, на которые нужно передвинуться.
whence:
SEEK_SET — движение начинается с начала файла;
SEEK_CUR — движение идет от текущей позиции;
SEEK_END — движение идет от конца файла.

//Читаем последние 10 знаков
fseek($f, -10, SEEK_END);
$s = fread($f, 10);
//Узнаем текущую позицию
$pos = ftell($f);
rewind($f) //сброс курсора

bool feof($f) //конец файла
```

Файлы: прямая работа с данными

```
//Получаем содержимое файла в виде массива
array file(string filename)

//Еще один вариант прямой работы с данными

//Чтение
file_get_contents(string filename)

//Запись
file_put_contents(string filename, mixed data[,
int flag]); //FILE_APPEND

//Если записать массив,
$array = array("I", "love", "you");
file_put_contents("test.txt", $array);
//то получим "Iloveyou"
```

Файлы: управление

```
//Копирование файла
copy(string source, string
destination);

//Переименование файла
rename(str oldname, str newname);

//Удаление файла
unlink(string filename);
```

Лабораторная работа

Работа с файлами

Откройте файл `mod2\file.php`

Задача

Написать «Гостевую книгу» с сохранением данных переданных пользователем в текстовый файл. Данные должны храниться в виде строки по принципу: одна строка – один пользователь

Задание 1

1. Установите константу для хранения имени файла
2. Проверьте, отправлялась ли форма и корректно ли отправлены данные из формы
3. В случае, если форма была отправлена, отфильтруйте полученные значения
4. Сформируйте строку для записи в файл
5. Откройте соединение с файлом и запишите в него сформированную строку.
6. Выполните перезапрос текущей страницы (чтобы избавиться от данных, отправленных методом POST)

Задание 2

1. Проверьте, существует ли файл с информацией о пользователях
2. Если файл существует, получите все содержимое файла в виде массива строк
3. В цикле выведите все строки данного файла с порядковым номером строки.
Последний пользователь должен показываться в списке первым
4. После этого выведите размер файла в байтах

Заполните форму

Имя:
Фамилия:

1 John Smith
2 Mike Lord
3 Ivan Petrov
4 Vasya Pupkin
Размер файла: 50 байт

Директории: работа и манипуляции

```
//Создание директории
mkdir(string dirname[, int mode])//!0777
//Удаление директории
rmdir(string dirname)//только пустая!
//Открываем директорию
$dir = opendir(string dirname)
//Читаем директорию
$name = readdir($dir)
//Закрываем директорию
closedir($dir)
//Это файл?
is_file(name)
//Это директория?
is_dir(name)

//Сканируем директорию
scandir(string dirname [, int order])
```

Файлы: загрузка на сервер

```
//Настройки PHP.ini

file_uploads (on|off)
upload_tmp_dir
upload_max_filesize (default = 2 Mb)

//Простая загрузка

<form enctype="multipart/form-data"
action="upload.php" method="POST">
<input type="hidden" name="MAX_FILE_SIZE"
value="51200">
<input type="file" name="userfile">
<input type="submit" value="Send">
</form>
```

Файлы: разбор на сервере

```
//Принимаем данные
$tmp = $_FILES['userfile']['tmp_name'];
$name = $_FILES['userfile']['name'];

//Перемещаем файл
move_uploaded_file($tmp, name);

//Что в массиве $_FILES
$_FILES['userfile']['name']
$_FILES['userfile']['tmp_name']
$_FILES['userfile']['size']
$_FILES['userfile']['type']
$_FILES['userfile']['error']
```

Функции работы с почтой

```
//Настройки в PHP.ini
[mail function]
; For Win32 only.
SMTP = localhost
; For Win32 only. (Только для Windows)
sendmail_from = me@localhost.com'

bool mail (string to,
           string subject,
           string message
           [, string additional_headers]
)
//Простое письмо
mail("john@smith.com",
     "Тема письма",
     "Строка 1\nСтрока 2\nСтрока 3");
```

Функции работы с почтой

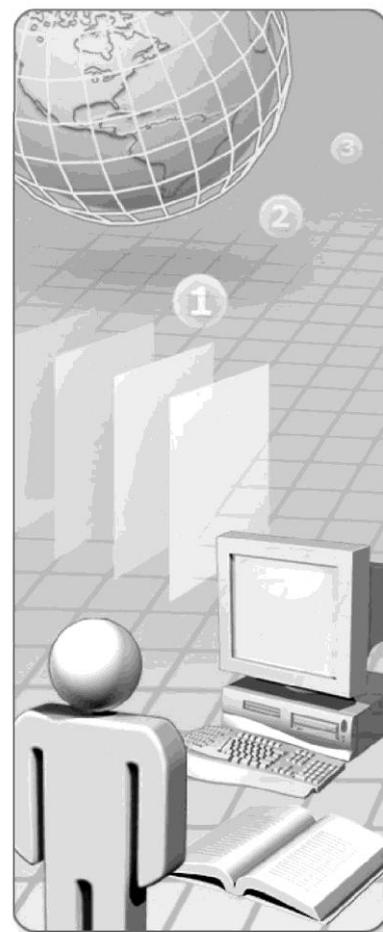
```
//Письмо с дополнительными заголовками
mail("john@smith.com",
     "Тема", $message,
     "From: webmaster@$SERVER_NAME\r\n"
     ."Reply-To: webmaster@$SERVER_NAME\r\n"
     ."X-Mailer: PHP/" . phpversion()
);
//Можно установить заголовки
$headers = "MIME-Version: 1.0\r\n";
$headers .= "Content-type: text/html;
charset=iso-8859-1\r\n";
$headers .= "From: Birthday Reminder
<birthday@example.com>\r\n";
$headers .= "Cc:
birthdayarchive@example.com\r\n";
$headers .= "Bcc: birthdaycheck@example.com\r\n";
```

Выводы

- Сессии
- Функции работы с файлами
- Функции работы с директориями
- Загрузка файлов на сервер
- Работа с почтой

Основы работы с базами данных. Сервер баз данных MySQL

Модуль 3



Содержание

- **Общие сведения о базах данных**
- **Общие сведения о SQL**
- **Обзорная установка сервера баз данных MySQL**
- **Работа с утилитами сервера баз данных MySQL**

Проектирование базы данных

Проектирование БД – создание эффективной структуры данных, обеспечивающее хранение требуемой информации

- Данные хранятся в таблицах, состоящих из столбцов и строк;
- На пересечении каждого столбца и строчки стоит в точности одно значение;
- У каждого столбца есть своё имя, которое служит его названием, и все значения в одном столбце имеют один тип.
- Запросы к базе данных возвращают результат в виде таблиц, которые тоже могут выступать как объект запросов.

Требования к структуре базы данных

"Хорошая структура"

- Максимально упрощает взаимодействие с БД
- Гарантирует непротиворечивость данных
- Выжимает максимум производительности из системы

"Плохая структура"

- Приводит к непониманию результатов выполнения запросов
- Повышает риск введения в БД противоречивой информации
- Порождает избыточные данные
- Усложняет выполнение изменений структуры созданных ранее и уже заполненных данных таблицы.

Таблицы

Таблица находится в первой нормальной форме, если каждый её атрибут атомарен и все строки различны. Под выражением «атрибут атомарен» понимается, что атрибут может содержать только одно значение.

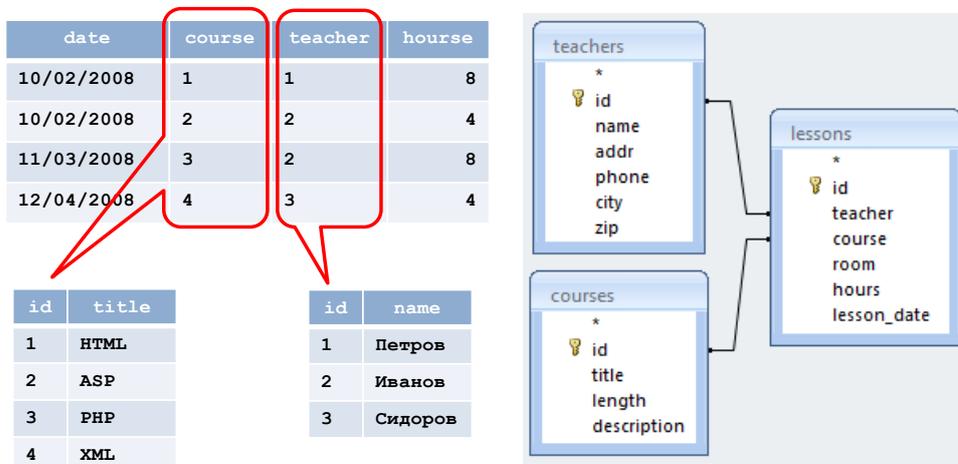
Колонки (поля)
↓

date	course	teacher	hourse
10/02/2008	HTML	Петров	8
10/02/2008	ASP	Иванов	4
11/03/2008	PHP	Иванов	8
12/04/2008	XML	Сидоров	4

←
Ряды (записи)

2 нормальная форма

Таблица находится во второй нормальной форме, если она находится в первой нормальной форме, и при этом любой её атрибут, не входящий в состав первичного ключа, функционально полно зависит от первичного ключа.



SQL – язык манипулирования данными (обзорно)

- SQL (*англ.* Structured Query Language — язык структурированных запросов) — универсальный язык, применяемый для создания, модификации и управления данными в реляционных базах данных
- Язык SQL делится на три части:
 - Операторы определения данных (Data Definition Language, DDL)
 - Операторы манипуляции данными (Data Manipulation Language, DML)
 - Операторы определения доступа к данным (Data Control Language, DCL)
- <http://ru.wikipedia.org/wiki/SQL>

Запрос SELECT

- `SELECT name, addr, city
FROM teachers
ORDER BY name`
- `SELECT title
FROM courses
WHERE length > 30`
- `SELECT *
FROM courses
WHERE length > 30
AND title LIKE 'Web%'`

SELECT: объединение таблиц

id	name	code	id	tid	course
1	Иванов	IVAN	1	1	PHP
2	Петров	PETR	2	1	XML

```
SELECT t.name, t.code, l.course
FROM teachers t
INNER JOIN lessons l ON t.id = l.tid
```

name	code	course
Иванов	IVAN	PHP
Иванов	IVAN	XML

```
SELECT t.name, t.code, l.course
FROM teachers t
LEFT OUTER JOIN lessons l ON t.id = l.tid
```

name	code	course
Иванов	IVAN	PHP
Иванов	IVAN	XML
Петров	PETR	NULL

SELECT: объединение таблиц

- ```
SELECT DISTINCT teachers.name
FROM teachers INNER JOIN
 (lessons INNER JOIN courses
 ON lessons.course = courses.id)
 ON teachers.id = lessons.teacher
WHERE courses.title LIKE 'Web%'
ORDER BY teachers.name
```

## Запрос INSERT

- `INSERT INTO courses  
VALUES (Null, 'Java2', '...', 40)`
- `INSERT INTO courses (title, length)  
VALUES ('Java2', 40)`

## Запрос DELETE

- `DELETE FROM lessons  
WHERE lessons.date = '2005-06-11'`

## Запрос UPDATE

```
● UPDATE teachers
 SET
 zarplata = zarplata * 2,
 premia = premia * 10
 WHERE name LIKE 'Иванов%'
 OR name LIKE 'Петров%'
 OR name LIKE 'Сидоров%'

● UPDATE teachers
 SET
 zarplata = zarplata * 2,
 premia = premia * 10
 WHERE name IN
 ('Иванов', 'Петров', 'Сидоров')
```

## Создание базы данных и таблицы (MySQL)

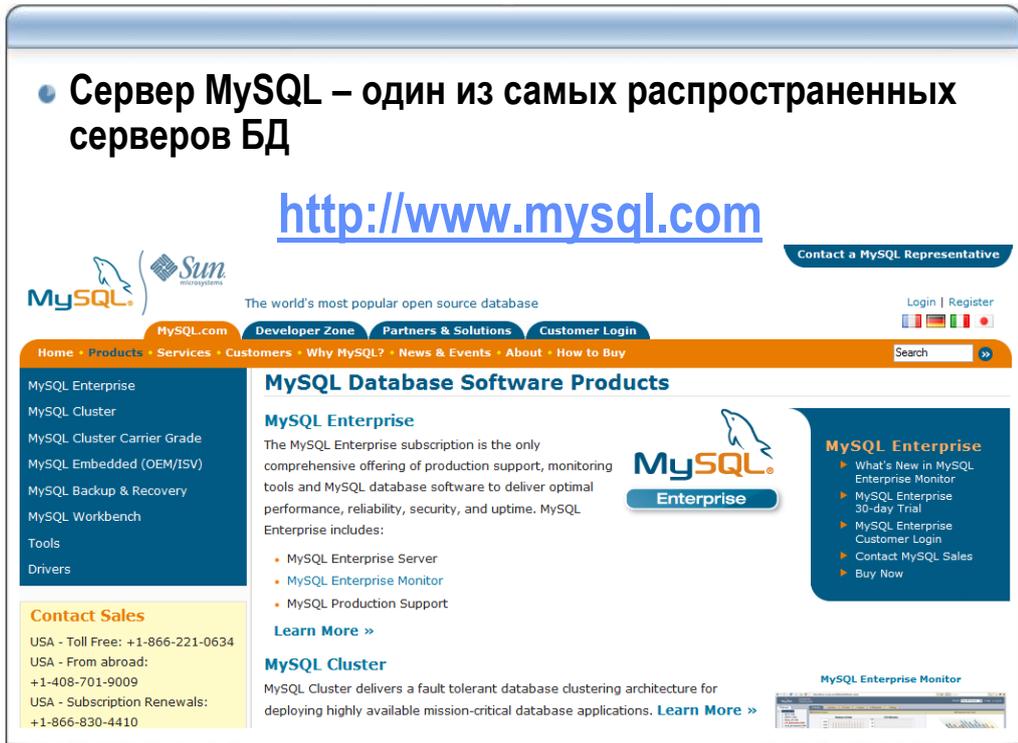
```
CREATE DATABASE имя_БД;

CREATE TABLE news (
 id int(4) NOT NULL auto_increment,
 title varchar(255) NOT NULL default '',
 description varchar(255) NOT NULL default
 '',
 content text,
 author varchar(50) NOT NULL default '',
 pubdate timestamp NOT NULL default '',
 PRIMARY KEY (id)
);
```

## Назначение и описание сервера MySQL 5

● **Сервер MySQL – один из самых распространенных серверов БД**

<http://www.mysql.com>



**MySQL Database Software Products**

**MySQL Enterprise**

The MySQL Enterprise subscription is the only comprehensive offering of production support, monitoring tools and MySQL database software to deliver optimal performance, reliability, security, and uptime. MySQL Enterprise includes:

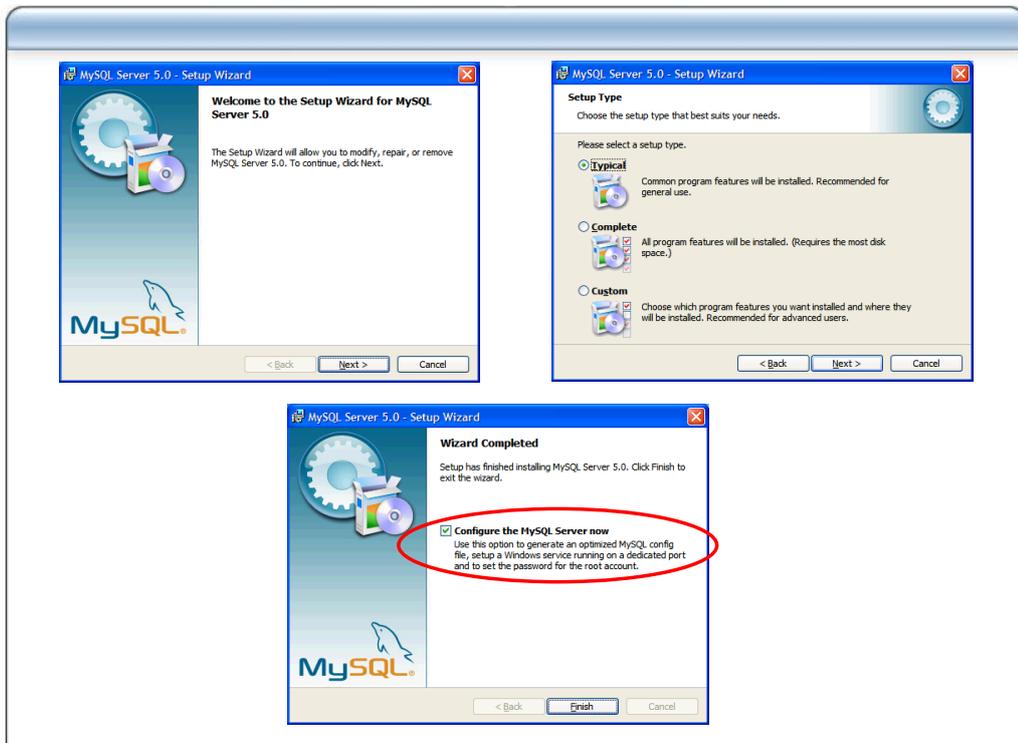
- MySQL Enterprise Server
- MySQL Enterprise Monitor
- MySQL Production Support

[Learn More >>](#)

**MySQL Cluster**

MySQL Cluster delivers a fault tolerant database clustering architecture for deploying highly available mission-critical database applications. [Learn More >>](#)

## Установка сервера MySQL



**Welcome to the Setup Wizard for MySQL Server 5.0**

The Setup Wizard will allow you to modify, repair, or remove MySQL Server 5.0. To continue, click Next.

**Setup Type**

Choose the setup type that best suits your needs.

Please select a setup type.

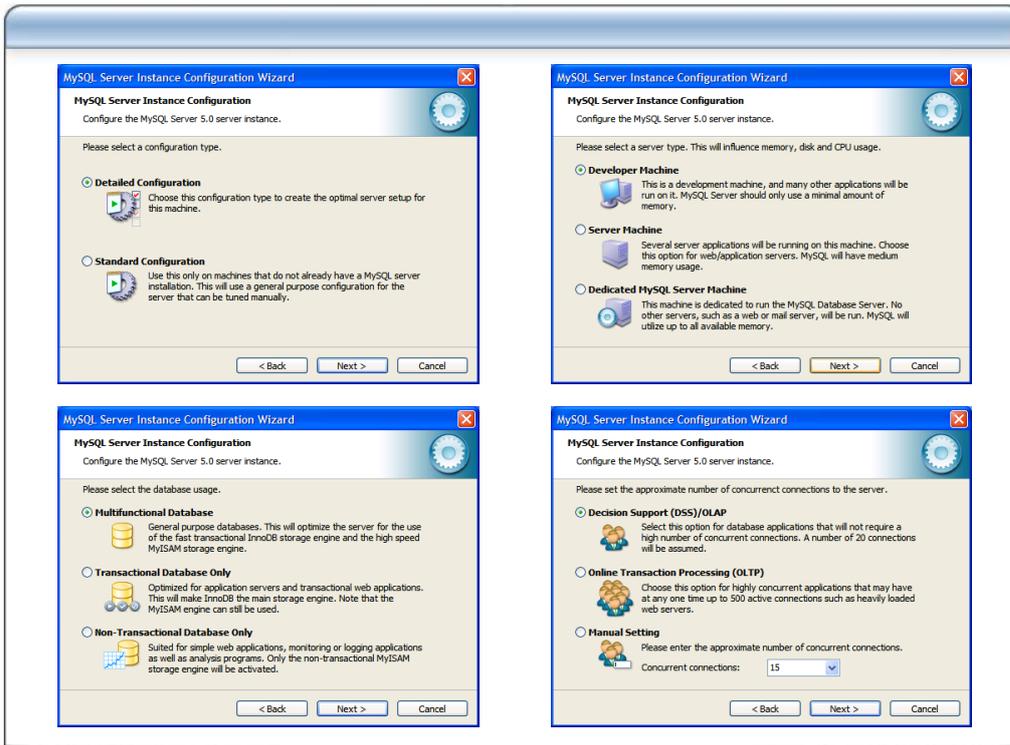
- Typical**  
Common program features will be installed. Recommended for general use.
- Complete**  
All program features will be installed. (Requires the most disk space.)
- Custom**  
Choose which program features you want installed and where they will be installed. Recommended for advanced users.

**Wizard Completed**

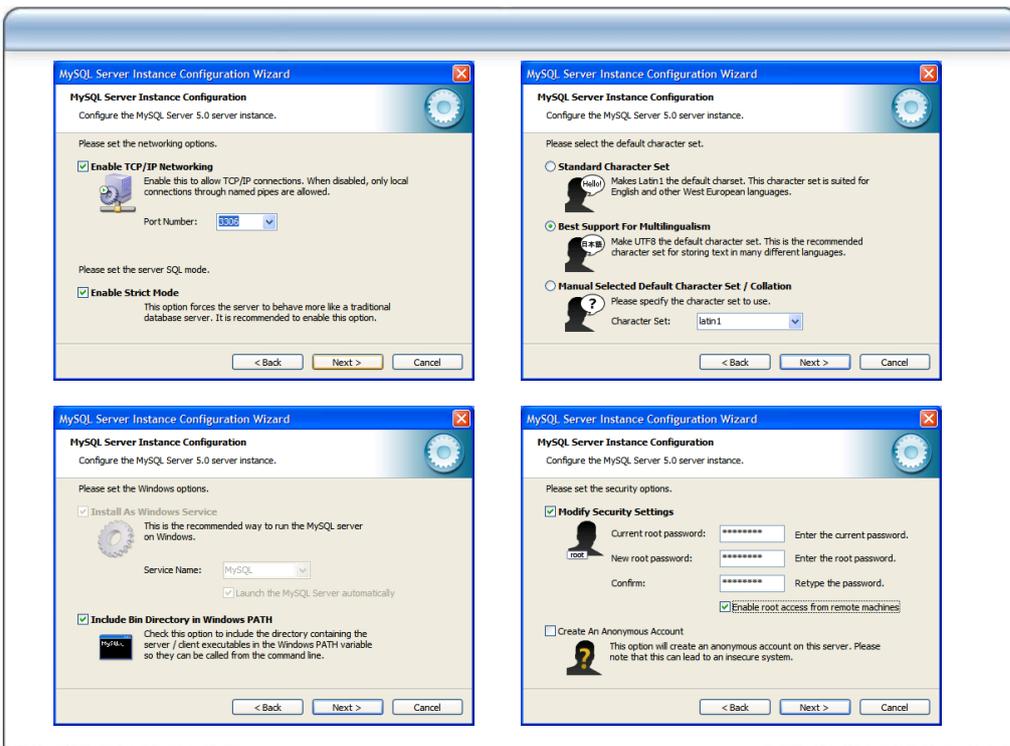
Setup has finished installing MySQL Server 5.0. Click Finish to exit the wizard.

**Configure the MySQL Server now**  
Use this option to generate an optimized MySQL config file, setup a Windows service running on a dedicated port and to set the password for the root account.

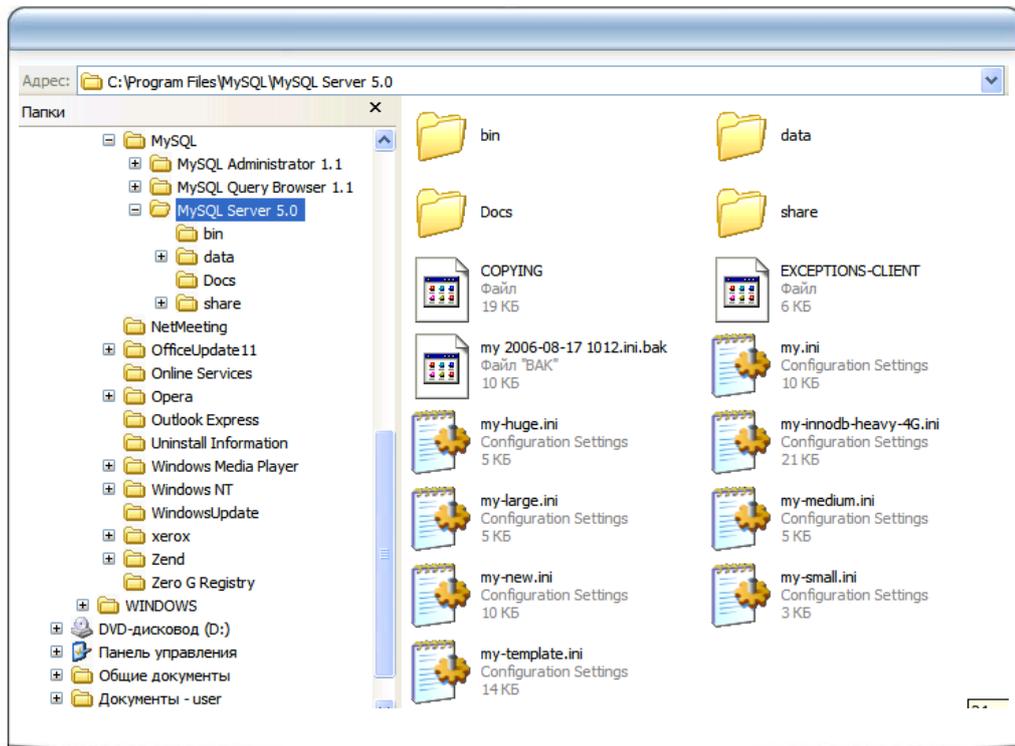
## Установка сервера MySQL



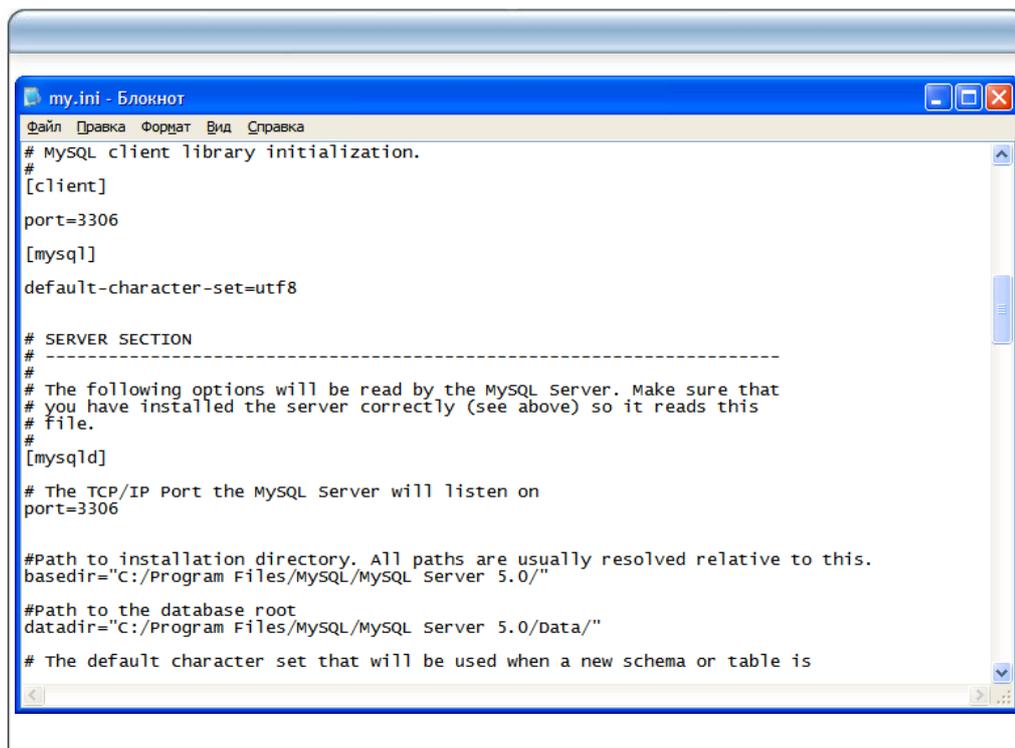
## Установка сервера MySQL



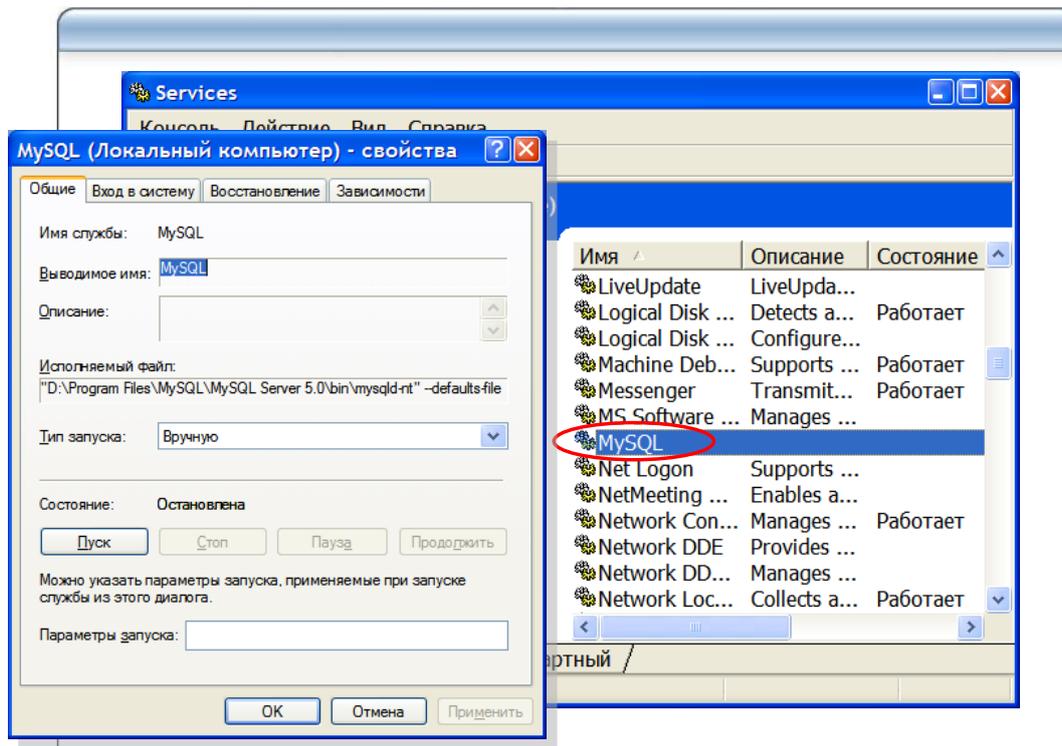
## Обзор файлов сервера MySQL 5



## Конфигурационный файл my.ini



## Служба сервера MySQL



## Клиентские программы для работы сервером

- Утилиты командной строки
  - mysql.exe
  - mysqldump.exe
- [MySQL Administrator](#)
- [MySQL Query Browser](#)
- другие, например, [EMS MySQL Manager](#)
- Веб-инструменты: [www.phpmyadmin.net](http://www.phpmyadmin.net)

## Использование утилиты mysql

- `mysql.exe -uлогин -pпароль [имя_БД]`
- `USE имя_БД;`
- `SET NAMES 'cp866';`
- `SHOW TABLES;`
- `DESCRIBE имя_таблицы;`
- `quit`

## Использование утилиты mysqldump

```
mysqldump -uroot -ppass имя_БД > dump.sql
```

```
mysql -uroot -ppass имя_БД < dump.sql
```

## Лабораторная работа

Работа с сервером MySQL из командной строки

### Внимание:

- в конце каждой команды (запроса) нажимайте клавишу «ENTER»! нажимайте клавишу «ENTER»
- внимательно смотрите на результат, появляющийся после введённой команды (запроса). В случае появления сообщения начинающегося со слова «**ERROR**», вернитесь к предыдущему пункту и повторите его – где-то Вы допустили ошибку.

### Задание 1

1. Скопируйте файл **web.sql** из директории **mod3** в директорию **C:\** (в корень диска C:)
2. Откройте окно командной строки и введите команду:  
`cd c:\`
3. Наберите команду (поменяйте **login** и **password** на ваш логин и пароль):  
`mysql -ulogin -ppassword`
4. В консоли сервера наберите запросы:  
`CREATE DATABASE web;`  
`quit`
5. Импортируйте данные из файла введя команду:  
`mysql -ulogin -ppassword web < web.sql`

**В случае успешного выполнения команды никаких сообщений не будет! Только приглашение к следующему вводу.**

### Задание 2

1. В окне командной строки наберите команды (поменяйте **login** и **password** на ваш логин и пароль):  
`mysql -ulogin -ppassword`  
`USE web;`  
`SHOW TABLES;`
2. Результат должен быть таким, как показано ниже. Если это не так, проверьте все шаги лабораторной работы или обратитесь к инструктору

```
mysql> SHOW TABLES;
+-----+
| Tables_in_web |
+-----+
| courses |
| lessons |
| teachers |
+-----+
3 rows in set (0.01 sec)
```

### Задание 3

1. Введите запрос и проверьте число записей в таблице **courses**

```
SELECT COUNT(*) FROM courses;
mysql> SELECT COUNT(*) FROM courses;
+-----+
| COUNT(*) |
+-----+
| 7 |
+-----+
1 row in set (0.01 sec)
```

2. Повторите предыдущий шаг для таблиц **lessons** и **teachers**

### Задание 4

1. Введите запрос:

```
DESCRIBE courses
mysql> DESCRIBE courses;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
id	int(11)	YES		NULL	
title	varchar(50)	YES		NULL	
length	int(11)	YES		NULL	
description	mediumtext	YES		NULL	
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.04 sec)
```

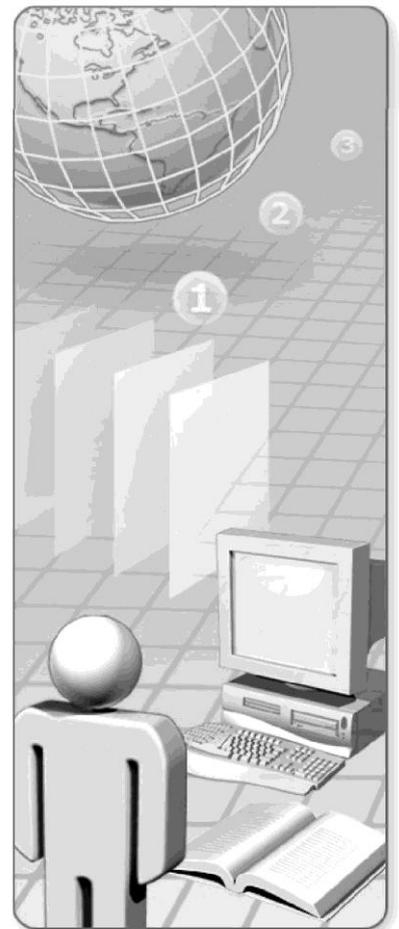
2. Повторите предыдущий шаг для таблиц **lessons** и **teachers**
3. Введите команду **quit** для выхода из консоли сервера MySQL

## Выводы

- Общие сведения о базах данных
- Общие сведения о SQL
- Установка сервера MySQL
- Утилиты сервера MySQL

# Использование сервера баз данных MySQL в приложениях PHP

## Модуль 4



## Содержание

- **Функции PHP для работы с сервером баз данных MySQL**
- **Подключение к серверу баз данных MySQL**
- **Выборка и обработка результатов**
- **Модули mysql и mysqli**
- **Эффективная работа с сервером MySQL**
- **Использование SQL View**
- **Использование подготовленных запросов**

## Алгоритм работы с сервером баз данных

1. Устанавливаем соединение с сервером БД
2. Выбираем базу данных для работы
3. Пошлaем запрос
  - ✓ При необходимости (SELECT), работаем с выбранными данными
4. Закрываем соединение

Подключение необходимых расширений  
в `php.ini`

- `php_pdo.dll`
- `php_pdo_mysql.dll`
- `php_mysql.dll`

## Функции PHP для работы с сервером MySQL

```
$conn = mysql_connect("host", "login", "pass")

mysql_close([$conn])

$conn = @mysql_connect("localhost", "root", "1234")
or die("Ошибка!");

//
//
//

mysql_close([$conn]);
```

## Функции PHP для работы с сервером MySQL

```

mysql_select_db(string db, [$conn]);
mysql_errno ([$conn]);
mysql_error ([$conn]);

$conn = @mysql_connect("localhost", "root",
"1234") or die("Ошибка!");

mysql_select_db("news");

if(mysql_errno() > 0){
 echo mysql_errno(). ": ". mysql_error();
}

mysql_close ([$conn]);

```

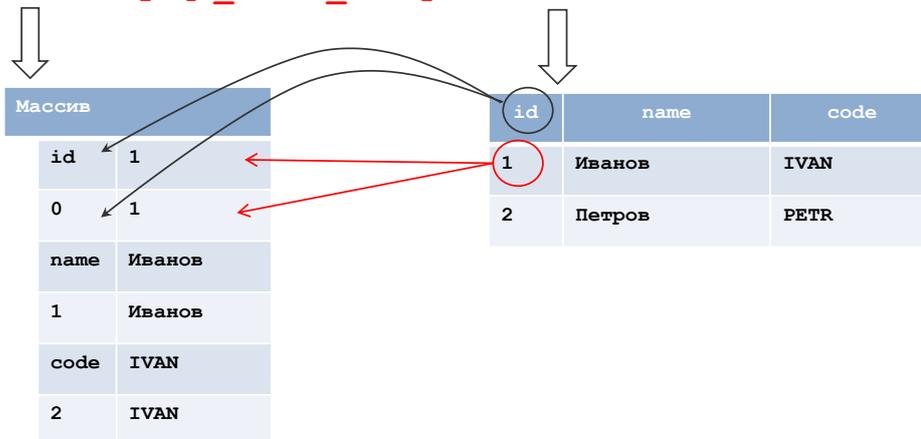
## Конвертируем результат запроса в массив

```

$result = mysql_query("SELECT * FROM teachers");
var_dump($result); ➔ resource(3, mysql result)

$row = mysql_fetch_array($result);

```



## Функции PHP для работы с сервером MySQL

```
$result = mysql_query(string query[, $conn])

$row = mysql_fetch_array($result[, type])

//По умолчанию
mysql_fetch_array($result, MYSQL_BOTH)

//Индексированный массив
mysql_fetch_array($result, MYSQL_NUM) //Что и
mysql_fetch_row($result)

//Ассоциативный массив
mysql_fetch_array($result, MYSQL_ASSOC) //Что и
mysql_fetch_assoc($result)
```

## Функции PHP для работы с сервером MySQL

```
//Точечная выборка
mysql_result($result, int row, string field)

mysql_num_rows($result) //Количество записей
mysql_num_fields($result) //Кол-во полей
mysql_field_name($result, int field) //Имя поля

$result = mysql_query("SELECT * FROM news");
echo mysql_num_rows($result);

mysql_affected_rows([$conn]) //Кол-во изменений

mysql_query("DELETE FROM news WHERE pubDate =
'2005-06-11'");
echo mysql_affected_rows();

mysql_insert_id([$conn]) //id последней записи
```

## Практическая работа

Гостевая книга с хранением данных на сервере MySQL

### Создание базы данных

1. Откройте в текстовом редакторе файл **mod4\createdb.php**
2. Поменяйте значения констант **DB\_LOGIN** и **DB\_PASSWORD** на те, которые используются на актуальном сервере MySQL
3. Сохраните файл
4. Запустите файл **mod4/createdb.php** в браузере.  
В случае успешного выполнения скрипта, появится фраза:  
«Структура базы данных успешно создана!»
5. Откройте в текстовом редакторе файл **mod4\gbook.php** и выполните следующие задания:

### Задание 1

#### Приём данных от пользователя и вставка новой записи в таблицу

1. Подключитесь к серверу MySQL
2. Выберите активную Базу Данных «**gbook**»
3. Проверьте, была ли корректным образом отправлена форма
4. Если она была отправлена: отфильтруйте полученные данные
5. Сформируйте SQL-оператор на вставку данных в таблицу **msgs** и выполните его
6. Выполните перезапрос страницы, чтобы избавиться от информации, переданной через форму

### Задание 2

#### Выборка данных из таблицы и показ результата выборки

1. Сформируйте SQL-оператор на выборку всех данных из таблицы **msgs** в обратном порядке и выполните его
2. Результат выборки сохраните в переменной.
3. Закройте соединение с сервером MySQL
4. Получите количество рядов результата выборки и выведите его на экран
5. В цикле получите очередной ряд результата выборки в виде ассоциативного массива.
6. Используя этот цикл, выведите на экран все сообщения, а также информацию об авторе каждого сообщения.  
После каждого сообщения сформируйте ссылку для удаления этой записи.
7. Информацию об идентификаторе удаляемого сообщения передавайте методом **GET**.

### Задание 3

#### Приём данных на удаление и удаление записи из таблицы

1. Проверьте, был ли запрос методом **GET** на удаление записи
2. Если он был: отфильтруйте полученные данные
3. Сформируйте SQL-оператор на удаление записи и выполните его
4. Выполните перезапрос страницы, чтобы избавиться от информации, переданной методом **GET**

#### Примерный вид готовой гостевой книги

## Гостевая книга

Ваше имя:

Ваш E-mail:

Сообщение:

Записей в гостевой книге: 2

---

[Ivan Petrov](#)

Privet from Ivan

[Удалить](#)

---

[John Smith](#)

Hello from John!

[Удалить](#)

## Расширение `mysqli`

### Подключение необходимых расширений

- `php_pdo.dll`
- `php_mysqli.dll`

### Особенности

- Процедурный интерфейс
- Объектно-ориентированный интерфейс
- Имеет поддержку дополнительных функций мониторинга, отлова ошибок, управления загрузкой и репликации

### Предупреждения

- Нет подключения к базе данных по умолчанию
- Нет соединения по умолчанию. Необходимо явно обращаться к соединению с сервером базы данных

## Функции PHP для работы с `mysqli`

```
$conn =
mysqli_connect('host', 'root', '1234', 'news');

$result = mysqli_query($conn, 'SELECT * FROM
articles')

while($row =
mysqli_fetch_array($result, MYSQLI_NUM)) {
 echo $row[0];
}

mysqli_free_result($result);

mysqli_close($conn);
```

## Использование SQL View

```
CREATE TABLE t
 (qty int, price int);
```

| qty | price |
|-----|-------|
| 3   | 50    |

```
CREATE VIEW v AS
 SELECT qty, price, qty * price AS value
 FROM t;

SELECT * FROM v;
```

| qty | price | value |
|-----|-------|-------|
| 3   | 50    | 150   |

## Использование подготовленных запросов

```
mysql_connect("localhost", "root", "password");
mysql_select_db("test");

mysql_query("PREPARE myinsert FROM
 'INSERT INTO
 test_table (name, price)
 VALUES (?, ?)'");

for ($i = 0; $i < 1000; $i++){

 mysql_query("SET @name = 'Товар # $i'");
 mysql_query("SET @price = " . ($i * 10));
 mysql_query("EXECUTE myinsert USING @name, @price");

}
mysql_close();
```

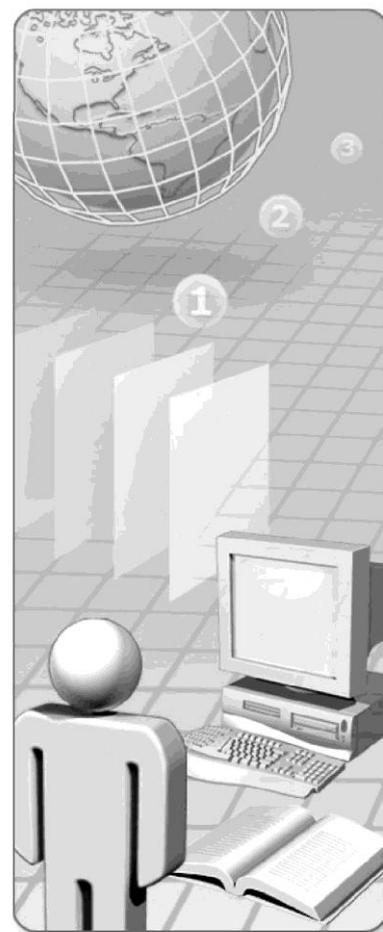
<http://dev.mysql.com/doc/refman/5.0/en/sqlps.html>

## Выводы

- **Функции PHP для работы с сервером MySQL**
- **Подключение к серверу MySQL**
- **Выборка и обработка результатов**
- **Модули mysql и mysqli**
- **Использование SQL View**
- **Использование подготовленных запросов**

# Практическое использование сервера баз данных MySQL: создание интернет-магазина

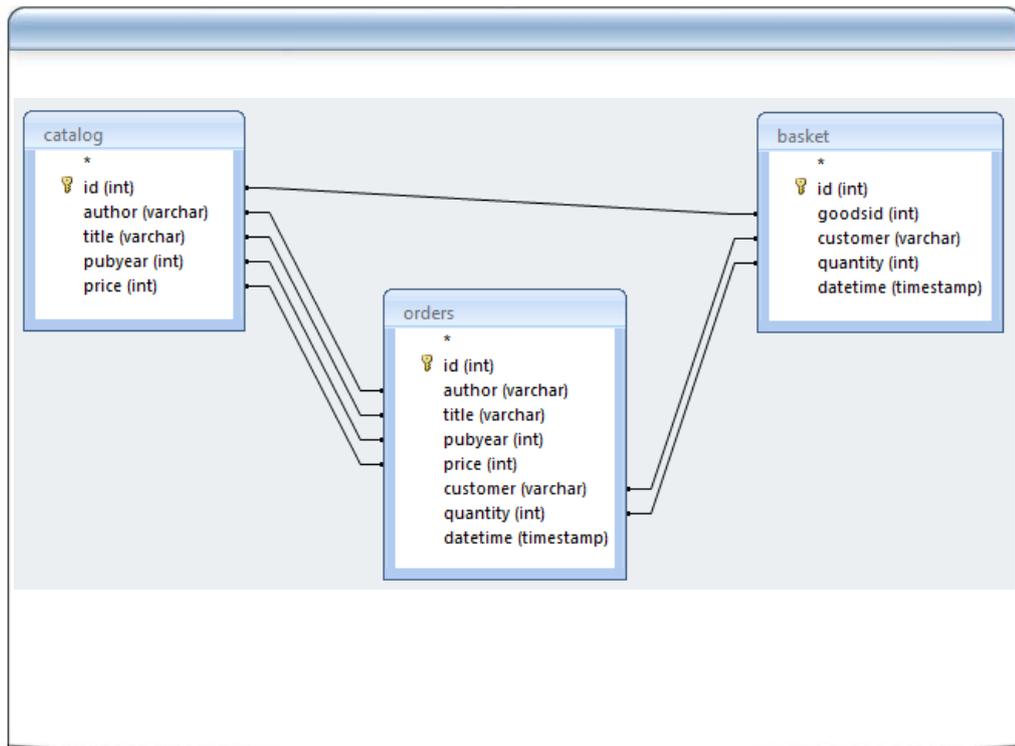
## Модуль 5



## Содержание

- **Создание базы данных**
- **Заполнение каталога товарами**
- **Выборка товаров из каталога**
- **Перемещение выбранных товаров в корзину пользователя**
- **Выборка товаров из корзины пользователя**
- **Удаление товаров из корзины пользователя**
- **Формирование заказа**
- **Выборка заказов**

## Схема электронного магазина



**Необходимые для работы файлы:**

**см. Приложение №1 на стр. 75**

**Примерные SQL-запросы используемые в практической работе:**

**см. Приложение №2 на стр. 76**

## Практическая работа №1

### Создание базы данных

1. Откройте в текстовом редакторе файл **eshop\create\_db.php**
2. Поменяйте значения констант **DB\_LOGIN** и **DB\_PASSWORD** на те, которые используются на актуальном сервере MySQL
3. Сохраните файл
4. Запустите файл **eshop/create\_db.php** в браузере.  
В случае успешного выполнения скрипта, появится фраза:  
**«Структура базы данных успешно создана!»**

## Практическая работа №2

### Соединение с сервером баз данных.

Откройте в текстовом редакторе файл `eshop\eshop_db.inc.php` и выполните следующие задания:

#### Задание 1

1. Создайте четыре константы:
  - DB\_HOST** - для хранения адреса сервера баз данных MySQL
  - DB\_LOGIN** - для хранения логина для соединения с сервером баз данных MySQL
  - DB\_PASSWORD** - для хранения пароля для соединения с сервером баз данных MySQL
  - DB\_NAME** - для хранения имени базы данных
2. Создайте константу **ORDERS\_LOG**, которая будет хранить имя файла с личными данными пользователей
3. Создайте переменную **count**, которая будет хранить количество товаров в корзине пользователя и присвойте ей значение по умолчанию
4. Установите соединение с сервером баз данных MySQL используя вышесозданные константы
5. Выберите на сервере для работы базу данных **DB\_NAME**

## Практическая работа №3

### Создание запроса на добавление товаров в каталог

Откройте в текстовом редакторе файл `eshop\eshop_lib.inc.php` и выполните следующие задания:

#### Задание 1

1. Опишите функцию `save()`, сохраняющую новый товар в таблицу `catalog`
2. Функция должна принимать следующие значения:
  - `author` – значение переданное в поле `author`
  - `title` – значение переданное в поле `title`
  - `pubyear` – значение переданное в поле `pubyear`
  - `price` – значение переданное в поле `price`

## Практическая работа №4

### Добавление товаров в каталог

Откройте в текстовом редакторе файл **eshop\save2cat.php** и выполните следующие задания:

#### Задание 1

1. Получите и отфильтруйте данные из формы

#### Задание 2

1. Вызовите функцию **save()** для сохранения нового товара в базу данных, передав необходимые параметры

#### Задание 3

1. Переадресуйте пользователя на страницу добавления нового товара (**add2cat.php**)

Запустите файл **eshop/add2cat.php** в браузере и заполните каталог

## Практическая работа №5

### Выборка товаров из таблицы catalog. Показ каталога

Откройте в текстовом редакторе файл `eshop\eshop_lib.inc.php` и выполните следующие задания:

#### Задание 2

1. Опишите функцию `selectAll()`, возвращающую все содержимое каталога товаров

Откройте в текстовом редакторе файл `eshop\catalog.php` и выполните следующие задания:

#### Задание 1

1. Выведите в этом месте строку «Товаров в корзине: » и текущее количество товаров в корзине для данного пользователя
2. Слово "корзине" оформите в виде гиперссылки на документ `basket.php`

#### Задание 2

1. С помощью функции `selectAll()` получите выборку всех товаров
2. В цикле выведите все товары на экран
3. Значение ячейки «В корзину» оформите в виде гиперссылки на документ `add2basket.php`, добавив параметр `id` с идентификатором (поле `id`) товара

## Практическая работа №6

### Получение количества товаров в корзине пользователя

Откройте в текстовом редакторе файл `eshop\eshop_db.inc.php` и выполните следующие задания:

#### Задание 2

1. Выполните SQL-оператор на выборку количества товаров в корзине данного пользователя
2. Получите результат и сохраните его в значении переменной **\$count**

## Практическая работа №7

### Добавление товара в корзину покупателя

Откройте в текстовом редакторе файл `eshop\eshop_lib.inc.php` и выполните следующие задания:

#### Задание 3

1. Опишите функцию `add2basket()`, которая будет добавлять товары в корзину пользователя
2. Функция должна принимать следующие значения:
  - `customer` – идентификатор покупателя
  - `goodsid` – идентификатор товара
  - `quantity` – количество товара
  - `datetime` – время добавления товара в корзину

Откройте в текстовом редакторе файл `eshop\add2basket.php` и выполните следующие задания:

#### Задание 1

1. Получите идентификатор конкретного покупателя
2. Получите идентификатор товара, добавляемого в корзину
3. Назначьте количество добавляемого товара равным 1
4. Получите дату добавления товара в корзину (текущую дату) в формате UNIX `timestamp`
5. Добавьте товар в корзину, вызвав функцию `add2basket()` и передав ей необходимые параметры
6. Переадресуйте пользователя на каталог товаров(`catalog.php`)

Откройте в текстовом редакторе файл `eshop\eshop_db.inc.php` и выполните следующие задания:

#### Задание 2

1. Выполните SQL-оператор на выборку количества товаров в корзине данного пользователя
2. Получите результат и сохраните его в значении переменной `count`

Запустите файл `eshop/catalog.php` в браузере и добавьте некоторое количество товара в корзину пользователя

## Практическая работа №8

### Выборка товаров из корзины. Показ корзины

Откройте в текстовом редакторе файл `eshop\eshop_lib.inc.php` и выполните следующие задания:

#### Задание 4

1. Опишите функцию `myBasket()`, которая будет возвращать всю пользовательскую корзину

Откройте в текстовом редакторе файл `eshop\basket.php` и выполните следующие задания:

#### Задание 1

1. Проверьте, есть ли товары в корзине пользователя
2. Если товаров нет, выводите строку "Корзина пуста!"
3. Если товары есть, выводите их в HTML-таблице

#### Задание 2

1. Получите все товары из корзины пользователя в виде массива
2. Создайте переменные для подсчета порядковых номеров (`i`) и общей суммы заказа (`sum`)
3. В цикле выводите все позиции из корзины на экран
4. Также, в цикле увеличивайте значение переменной `sum` на соответствующее значение(сумма текущего товара \* его количество)
5. Значение ячейки "Удалить" оформите в виде гиперссылки на документ `delete_from_basket.php`, добавив параметр `id` с `id` записи

#### Задание 3

1. Выведите общую сумму товаров в корзине

Запустите файл `eshop/basket.php` в браузере и убедитесь в наличие товаров в корзине пользователя

## Практическая работа №9

### Удаление товара из корзины покупателя

Откройте в текстовом редакторе файл **eshop\eshop\_lib.inc.php** и выполните следующие задания:

#### Задание 5

1. Опишите функцию **basketDel()**, которая будет удалять товар из корзины пользователя
2. Функция должна принимать следующие значения:  
`id` – идентификатор записи в таблице **basket**

Откройте в текстовом редакторе файл **eshop\delete\_from\_basket.php** и выполните следующие задания:

#### Задание 1

1. Получите идентификатор удаляемого товара
2. Вызовите функцию **basketDel()** для данного товара, передав необходимые параметры
3. Переадресуйте пользователя на корзину заказов(**basket.php**)

Запустите файл **eshop/basket.php** в браузере и удалите некоторое количество товара из корзины пользователя

## Практическая работа №10

### Создание заказа на покупку товаров. Заказ товаров

Откройте в текстовом редакторе файл `eshop\eshop_lib.inc.php` и выполните следующее задание:

#### Задание 6

1. Опишите функцию `resave()` для пересохранения товаров из корзины (таблица `basket`) в заказы (таблица `orders`)
2. Функция должна принимать следующие значения:  
`datetime` – дата оформления заказа
3. Для получения содержимого корзины в этой функции воспользуйтесь функцией `myBasket()`
4. Опишите в функции `resave()` SQL-оператор, который будет вставлять данные из корзины в таблицу `orders` и выполните его
5. Опишите SQL-оператор для удаления данных о корзине текущего покупателя из таблицы `basket`

Откройте в текстовом редакторе файл `eshop\saveorder.php` и выполните следующие задания:

#### Задание 1

1. Получите из формы и обработайте данные заказа

#### Задание 2

1. Получите `id пользователя` и текущие дату/время в формате `UNIX timestamp`
2. Создайте переменную `order`
3. Создайте строку из полученных данных, разделяя их символом «|» вида «`name | email | phone | address | customer_id | timestamp`». Например: "Иван Иванов|ivan@mail.ru|123-12-23|Москва, Сумской пр-д 17 кв.105|549b828a5eaa5cb185e84a71fa068272|1234567890"
4. Присвойте созданную строку переменной `order`
5. Запишите значение переменной `order` в файл `orders.log`  
**ВНИМАНИЕ:** в зависимости от того, каким образом будет производиться работа с файлом, возможно, будет необходимо проверить существование данного файла. Новые данные должны записываться в конец файла!
6. Сохраните файл

#### Задание 3

1. Вызовите функцию `resave()` для пересохранения купленных товаров из корзины в таблицу `orders`, передав необходимые параметры

Зайдите в корзину покупателя и оформите заказ

## Практическая работа № 11

### Создание файла просмотра заказов. Просмотр заказов

Откройте в текстовом редакторе файл `eshop\eshop_lib.inc.php` и выполните следующее задание:

#### Задание 7

1. Опишите функцию `getOrders()` для получения информации о заказах
2. Получите в виде массива `orders` данные о пользователях из файла `orders.log`
3. Создайте массив `allorders` для хранения информации обо всех заказах
4. В цикле `foreach` переберите все заказы
5. Внутри цикла `foreach` создайте ассоциативный массив `orderinfo` для хранения информации о каждом конкретном заказе
6. Сохраните информацию о пользователе из массива `orders(name, email, phone, address, customer, date)` в массиве `orderinfo`
7. Опишите SQL-оператор для выборки из таблицы заказов всех товаров для конкретного покупателя
8. Получите весь результат этой выборки
9. Сохраните полученный в предыдущем пункте результат как значение ключа `"goods"` в массиве `orderinfo`
10. Добавьте сформированный массив `orderinfo` в виде значения очередного ключа массива `allorders`
11. Функция `getOrders()` должна возвращать массив `allorders` с информацией о всех покупателях и сделанных ими заказах

Откройте в текстовом редакторе файл `eshop/orders.php` и выполните следующее задание:

#### Задание 1

1. Вызовите функцию `getOrders()` и сохраните результат её работы в переменную
2. Используя цикл `foreach` выведите информацию обо всех заказах в виде HTML-таблицы

Запустите файл `eshop/orders.php` в браузере и просмотрите оформленные заказы

## Приложение №1

### Необходимые для работы файлы

- **create\_db.php**  
PHP-код создания базы данных и таблиц
- **eshop\_db.inc.php**  
основные настройки сайта
- **eshop\_lib.inc.php**  
библиотека функций сайта
- **add2cat.php**  
HTML-форма для добавления товаров (без PHP-кода)
- **save2cat.php**  
PHP-код обработки HTML-формы
- **catalog.php**  
вывод списка товаров в виде HTML-таблицы
- **add2basket.php**  
PHP-код обработки данных для добавления товара в корзину (без PHP-кода)
- **basket.php**  
вывод товаров покупателя в виде HTML-таблицы
- **delete\_from\_basket.php**  
PHP-код обработки данных для удаления товара из корзины
- **orderform.php**  
HTML-форма для добавления персональных данных покупателя (без PHP-кода)
- **orders.php**  
PHP-код обработки персональных данных покупателя для сохранения в текстовый файл и перемещения товаров из корзины покупателя в таблицу orders
- **saveorder.php**  
Вывод списка заказов в виде HTML-таблицы

## Приложение №2

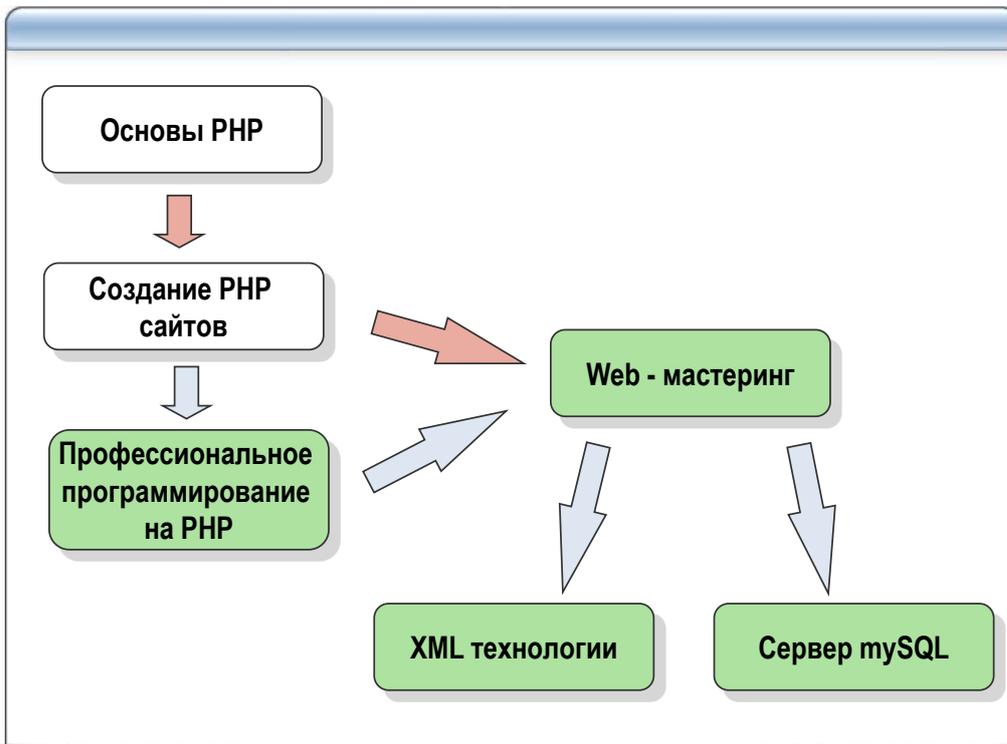
### Примерные SQL-запросы используемые в практической работе

1. `$sql = "SELECT count(*) FROM basket  
WHERE customer='".session_id()."";`
2. `$sql = "INSERT INTO catalog(author, title, pubyear, price)  
VALUES('$author', '$title', $pubyear, $price)";`
3. `$sql = "SELECT * FROM catalog";`
4. `$sql = "INSERT INTO catalog(customer, goodsid, quantity,  
datetime)  
VALUES('$customer', $goodsid, $quantity,  
$datetime)";`
5. `$sql = "SELECT * FROM catalog, basket  
WHERE customer='".session_id()."'  
AND catalog.id=basket.goodsid";`
6. `$sql = "DELETE FROM basket WHERE id = $id";`
7. `$sql = "INSERT INTO catalog(author, title, pubyear, price,  
customer, quantity, datetime)  
VALUES(  
" . $item["author"] . ",  
" . $item["title"] . ",  
" . $item["pubyear"] . ",  
" . $item["price"] . ",  
" . $item["customer"] . ",  
" . $item["quantity"] . ",  
$datetime)";`
8. `$sql = "DELETE FROM basket  
WHERE customer='".session_id()."";`
9. `$sql = "SELECT * FROM orders  
WHERE customer='".$orderinfo["customer"]."  
AND datetime='".$orderinfo["date"];"`

## Выводы

- Создание базы данных
- Каталог товаров
- Корзина пользователя
- Формирование заказа
- Выборка заказов

## Что дальше?



## У вас есть вопросы?

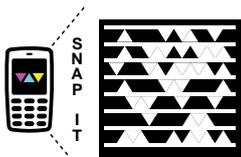
• Напишите E-mail тренеру...

• Конференция Клуб Выпускников:

<http://forum.specialist.ru/Discussions.aspx?id=9>

• Семинары и консультации

<http://www.specialist.ru/Events/Seminars>



Get the free app for your phone at <http://gettag.mobi>