ЛАБОРАТОРНАЯ РАБОТА №6
ВЗАИМОДЕЙСТВИЕ PHP И MYSQL

Цель работы: изучить основные команды PHP для соединения с сервером MySQL, отправки SQL-запросов на сервер базы данных и обработки результатов SQL-запросов.

Ход выполнения лабораторной работы должен быть отражен в отчете. Отчет должен содержать титульный лист, номера заданий, коды программ, скриншот с результатом выполнения программы.

6.1 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

SQL может применяться в прикладных программах двумя способами: в виде встроенного SQL и интерфейса программирования приложений (Application Program Interface, API). Первый способ напоминает использование PHP – инструкции SQL размещаются среди кода прикладной программы. В настоящий момент такой стиль не поддерживают ни MySQL, ни PHP. Второй подход заключается в том, что программа взаимодействует с СУБД посредством совокупности функций. Именно такой подход используется при взаимодействии PHP и MySQL.

6.1.1 Функции управления соединением с сервером MySQL

Для установки соединения с сервером MySQL используется функция mysql_connect(). Синтаксис функции:

resource **mysql_connect** ([string server [, string username [, string password]]])

Эта функция устанавливает соединение с сервером **server** MySQL и возвращает дескриптор соединения с базой данных, по которому все другие функции, принимающие этот дескриптор в качестве аргумента, будут однозначно определять выбранную базу данных. Вторым и третьим аргументами этой функции являются имя пользователя **username** и его пароль **password** соответственно (пример 6.1).

Пример 6.1

```
<?php
$dblocation = "localhost"; // Имя сервера
$dbuser = "root";          // Имя пользователя
$dbpasswd = "";            // Пароль
$dbcnx = @mysql_connect($dblocation,$dbuser,$dbpasswd);
if (!$dbcnx) // Если дескриптор равен 0, соединение не установлено
{
  echo("<P>В настоящий момент сервер базы данных не доступен, поэтому
        корректное отображение страницы невозможно.</P>");
```

```
  exit();
}?>
```

В примере 6.1 переменные **$dblocation**, **$dbuser** и **$dbpasswd** хранят имя сервера, имя пользователя и пароль и, как правило, прописываются в отдельном файле (к примеру, **config.php**), который потом вставляется в каждый PHP-файл, имеющий код для работы с MySQL.

Для закрытия открытого ранее соединения можно воспользоваться функцией mysql_close(). Открытое соединение закрывается автоматически по завершении работы сценария. Синтаксис функции:

bool **mysql_close** ([resource conn_id])

Эта функция разрывает соединение с сервером MySQL и возвращает true при успешном выполнении операции и false в противном случае. Функция принимает в качестве аргумента дескриптор соединения с базой данных, возвращаемый функцией mysql_connect() (пример 6.2).

Пример 6.2

```
<? // устанавливаем соединение с базой данных
$dbcnx = @mysql_connect($dblocation,$dbuser,$dbpasswd);
if (!$dbcnx)
{
  // Выводим предупреждение
  echo ("<P>В настоящий момент сервер базы данных не доступен, поэтому
  корректное отображение страницы невозможно.</P>");
  // Завершаем работу в случае неудачи
  exit();
}
  // Выполняем запросы
if(mysql_close($dbcnx)) // разрываем соединение
{
  echo("Соединение с базой данных прекращено");
}
else
{
  echo("Не удалось завершить соединение");
} ?>
```

6.1.2 Функции состояния и диагностики ошибок

Функции mysql_errno() и mysql_error() возвращают числовой код и информационное сообщение об ошибке в работе связанных с MySQL функций PHP. Однако эти функции не могут использоваться для получения информации о результатах безуспешного завершения вызовов функций mysql_connect(), поскольку идентификатор связи становится доступным только после успешного установления соединения.

Синтаксис функции:

int **mysql_errno** ([resource conn_id);

Функция возвращает код ошибки для связанной с заданным соединением MySQL функции, которая последней возвращала значение состояния. Нулевой код указывает на отсутствие ошибки.

Синтаксис функции:

string **mysql_error** ([resource conn_id]) ;

Функция возвращает строку с сообщением об ошибке для связанной с заданным соединением MySQL функции, которая последней возвращала значение состояния. Пустая строка свидетельствует об отсутствии ошибки.

6.1.3 Функции построения и выполнения запросов

Функции построения и выполнения запросов используются для создания и отсылки запросов на сервер MySQL. Строки запроса должны состоять из одного оператора SQL и не должны заканчиваться символом «точка с запятой» (";") или последовательностью "\g". Окончания «;» и «\g» являются соглашениями клиентской программы MySQL и не используются при выводе запросов через PHP.

Выборка указанной базы данных, для того чтобы сделать ее текущей базой данных для заданного соединения, выполняется функцией mysql_select_db(). Синтаксис функции:

bool **mysql_select_db**(string db_name [, resource conn_id]);

Таблицы, на которые будут делаться ссылки в последующих запросах, по умолчанию будут принадлежать этой базе данных, если не будет указана другая база данных. Функция возвращает значение «true» при успешном завершении и «false», если появилась ошибка.

Пример 6.3

Переменные $dblocation, $dbuser и $dbpasswd хранят имя сервера, имя пользователя и пароль и, как правило, прописываются в отдельном файле (к примеру, config.php) вместе с функциями для соединения и выбора базы данных, который потом вставляется в каждый PHP-файл, где имеется код для работы с MySQL.

```
<?php
$dblocation = "localhost";
$dbname = "softtime";
$dbuser = "root";
$dbpasswd = "";
$dbcnx = @mysql_connect($dblocation,$dbuser,$dbpasswd);
if (!$dbcnx)
{
  echo( "<P>В настоящий момент сервер базы данных не доступен, поэтому
        корректное отображение страницы невозможно.</P>" );
  exit();
}
```

```
if (!@mysql_select_db($dbname, $dbcnx))
{
  echo( "<P>В настоящий момент база данных не доступна, поэтому
        корректное отображение страницы невозможно.</P>" );
  exit(); } ?>
```

Задание 6.1. *Создать файл* ***config.php*** *для подключения к базе данных* ***bookby*** *(созданной в лабораторной работе №5).*

Для отправки строки к серверу MySQL для заданного соединения используется функция mysql_query(). Синтаксис функции:

int **mysql_query**(string query [, resource conn_id]) ;

Для операторов DELETE, INSERT, REPLACE и UPDATE функция mysql_query () возвращает значение «true» в случае успешного выполнения и значение «false», если имела место ошибка. После успешного выполнения запроса можно вызвать функцию mysql_affected_rows () , чтобы узнать число измененных строк.

Для операторов SELECT функция mysql_query () возвращает положительный идентификатор результирующего набора в случае успешного выполнения и значение «false», если имела место ошибка. Возвращаемый после успешного выполнения запроса идентификатор результата может использоваться самыми разными функциями обработки результирующих наборов, считывающими аргумент result_id. Для освобождения занятых результирующим набором ресурсов системы этот идентификатор необходимо передать функции mysql_free_result ().

6.1.4 Функции обработки результирующих наборов

Описанные в данном разделе функции используются для выборки результатов выполнения запросов, а также для предоставления информации о результатах, например количества измененных строк.

Синтаксис функции:

array **mysql_fetch_array** (resource result_id [, int result_type]) ;

Функция возвращает следующую строку данного результирующего набора в виде массива. Возвращает значение «false», если строк больше не осталось. Возвращаемый массив содержит значения, идентифицируемые как по числовым индексам столбцов, так и по ключам имен столбцов. Другими словами, доступ к значению каждого столбца можно получить как по числовому индексу, так и по названию столбца. Учитывается регистр символов при написании ассоциативных индексов, поэтому их следует записывать в том же регистре, что и имена столбцов в запросе.

Предположим, например, что отправлен следующий запрос:

SELECT last_name, first_name FROM president

Если строки извлекаются из результирующего набора в массив с именем $row, доступ к его элементам можно получить следующим образом:

$row [0] или $row ["last_name"] Содержит значение столбца last_name.

$row [1] или $row ["first_name"] Содержит значение столбца first_name.

Параметр result_type может иметь значения MYSQL_ASSOC (возвращаются значения только по индексам имен), MYSQL_NUM (возвращаются значения только по числовым индексам) или MYSQL_BOTH (возвращаются значения по индексам обоих типов). Если этот аргумент отсутствует, его значение по умолчанию устанавливается равным MYSQL_BOTH.

Вызов mysql_fetch_array () с параметром result_type, имеющим значение MYSQL_ASSOC или MYSQL_NUM, эквивалентно вызову mysql_fetch_assoc () или mysql_f etch_row ().

Пример 6.4

Из таблицы authors базы данных forums выбираются все записи, содержащие информацию об авторах, и выводятся на экран в табличной форме. Результат работы скрипта показан на рисунке 6.1.

```
<?php
include "config.php";
$ath = mysql_query("select * from authors;");
if($ath)
{
 // Определяем таблицу и заголовок
 echo "<table border=1>";
 echo "<tr><td>имя</td><td>пароль</td><td>e-mail</td><td>url</td></tr>";
 // Так как запрос возвращает несколько строк, применяем цикл
 while($author = mysql_fetch_array($ath))
 {
  echo "<tr><td>".$author['name']." </td><td>".$author['passw']."
    </td><td>".$author['email']." </td><td>".
  $author['url']."] </td></tr>";
 }
 echo "</table>";
}
else
{
 echo "<p><b>Error: ".mysql_error()."</b><p>";
 exit();
}
?>
```

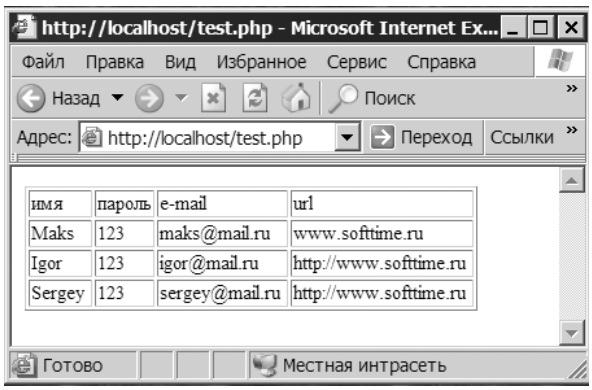

Рисунок 6.1 – Результат выполнения примера 6.4

Задание 6.2. *Создать форму для поиска книг в базе данных bookby* ***(poisk.php).*** *На странице должно быть поле со списком, в котором пользователь может выбрать критерий поиска (по ISBN, или по автору, или по заголовку книги), текстовое поле для ввода искомого значения и кнопка отправки данных. Например:*

```
<HTML> <HEAD> <TITLE>Поиск книги</TITLE> </HEAD>
<BODY>
<H1>Поиск книги в каталоге</H1>
<FORM ACTION="results.php" METHOD="POST">
    Выберите параметр поиска:<BR>
    <SELECT NAME="searchtype">
        <OPTION VALUE="author">Автор
        < OPTION VALUE ="title">Заголовок
        < OPTION VALUE ="isbn">ISBN
    </SELECT>
    <BR>
    Введите параметры поиска:<BR>
    <INPUT NAME="searchterm" TYPE="text">
    <BR>
    <INPUT TYPE="submit" VALUE="Поиск">
</FORM>
</BODY>
</HTML>
```

Задание 6.3. *Вывести на экран результаты поиска results.php по критериям, введенным в форму, созданную в задании 6.2. После нажатия кнопки в форме в переменной searchtype будет храниться имя столбца, по которому будет осуществляться поиск в таблице books, а искомое значение будет храниться в переменной searchterm. Пример файла results.php:*

```
<HTML>
 <HEAD> <TITLE>Результаты поиска книги</TITLE> </HEAD>
<BODY>
<H1> Результаты поиска книги </H1>
```

```
<?
trim($_POST["searchterm"]) ; //убираем лишние пробелы по краям слова
if (!$_POST["searchtype"] || !$_POST["searchterm"])
{
  echo "Вы не ввели критерии поиска. Вернитесь назад и попробуйте еще раз";
  exit();
}
// добавляет символы косой черты перед символами,
// которые могут быть интерпретированы как управляющие
$_POST["searchtype"] = addslashes($_POST["searchtype"]);
$_POST["searchterm"] = addslashes($_POST["searchterm"]);
include "config.php"; // соединяемся с сервером MySQL и выбираем БД
// составляем запрос
$query = "select * from books where ".$_POST["searchtype"]." like
          '%".$_POST["searchterm"]."%'";
$result = mysql_query($query); // выполняем запрос
// определяем количество строк в запросе
$num_results = mysql_num_rows ($result);
echo "<P>Количество найденных книг: ".$num_results."</P>";
for  ($i=0; $i <$num_results; $i++) {
      $row = mysql_fetch_array($result); // функция берет каждую строку
//из списка результата и возвращает ее в виде ассоциативного массива
      echo "<P><STRONG>". ($i+1) .". Название: ";
      echo $row["title"];
      echo "</STRONG><BR>Автор: " ;
      echo $row["author"];
      echo "<BR>ISBN: ";
      echo $row["isbn"];
      echo "<BR>Цена: ";
      echo $row["price"];
      echo "</P>";
}
?>
</BODY> </HTML>
```

Функция **mysql_fetch_assoc** () возвращает следующую строку данного результирующего набора в виде ассоциативного массива. Синтаксис функции:

array **mysql_fetch_assoc** (resource result_id);

Функция возвращает значение «false», если строк больше не осталось. Значения столбцов доступны с помощью ассоциативных имен, соответствующих именам столбцов, выбранных запросом.

Вызов mysql_fetch_assoc () подобен вызову mysql_fetch_array() со вторым аргументом MYSQL_ASSOC. Значения столбцов с цифровыми индексам не возвращаются.

Функция **mysql_fetch_row**() возвращает следующую строку заданного результирующего набора в виде массива либо значение «false», если строк больше не осталось. Синтаксис функции:

array **mysql_fetch_row** (resource result_id) ;

Доступ к значениям столбцов можно получать через элементы массива, используя индексы столбцов в диапазоне от 0 до mysql_num_fields ()-1.

Вызов функции mysql_fetch_row() аналогичен вызову функции mysql_fetch_array() со вторым аргументом, равным MYSQL_NUM. Значения столбцов с ассоциативными индексами не возвращаются.

Функция **mysql_affected_rows** () возвращает количество строк, измененных последним запросом DELETE, INSERT, REPLACE или UPDATE в заданном соединении. Синтаксис функции:

int **mysql_affected_rows** ([resource conn_id]) ;

Функция mysql_affected_rows () возвращает значение 0, если ни одна строка изменена не была, и значение -1, если имела место ошибка.

Вызванная после оператора SELECT функция mysql_affected_rows () возвращает число строк, полученных в результате выборки. Хотя обычно для операторов SELECT используется функция mysql_num_rows ().

Функция **mysql_num_fields** () возвращает число столбцов в заданном результирующем наборе. Синтаксис функции:

int **mysql_num_fields** (resource result_id) ;

Функция **mysql_num_rows** () возвращает число строк в заданном результирующем наборе. Синтаксис функции:

int **mysql_num_rows** (resource result_id) ;

Функция **mysql_insert_id** () возвращает значение AUTO_INCREMENT, сгенерированное последним запросом заданного соединения. Синтаксис функции:

int **mysql_insert_id** ([resource conn_id]) ;

Если в процессе всего соединения такое значение создано не было, эта функция возвращает нуль. В общем, функцию mysql_insert_id () следует вызывать сразу после запроса, который сгенерирует значение AUTO_INCREMENT. Если между этим запросом и вызовом функции будет выполнен промежуточный запрос, возвращаемое функцией значение может быть сброшено до нуля.

Задание 6.4. *Создать файл* ***show.php****, который будет выводить на экран список всех книг.*

Задание 6.5. *Создать форму для ввода новой книги в БД.*

6.2 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

1 Создать базу данных employee

```
create database employee;
use employee;

create table department
(
departmentID int not null auto_increment primary key,
name varchar(30)
) type=InnoDB;

create table employee
(
employeeID int not null auto_increment primary key,
name varchar(80),
job varchar(30),
departmentID int not null references department (departmentID)
) type=InnoDB;

create table employeeSkills
(
employeeID int not null references employee(employeeID),
skill varchar(15) not null,
primary key(employeeID, skill)
) type=InnoDB;

create table client
(
clientID int not null auto_increment primary key,
name varchar(40),
address varchar(100),
contactPerson varchar(80),
contactNumber char(12)
) type=InnoDB;

create table assignment
(
clientID int not null references client(clientID),
employeeID int not null references employee(employeeID),
workdate date not null,
hours float,
primary key(clientID, employeeID, workdate)
) type=InnoDB;
```

```
insert into department values
(42, 'Финансовый отдел'),
(128, 'Отдел проектирования'),
(null, 'Отдел кадров'),
(null, 'Отдел маркетинга');
insert from employee values
(7513, 'Нора Эдвардс', 'Программист', 128),
(9842, 'Бен Смит', 'Администратор БД' ,42),
(6651, 'Аджай Пател', 'Программист' ,128),
(9006, 'Кэнди Барнетт', 'Системный администратор' ,128);

insert into employeeskills values
(7513, 'C'),
(7513, 'Perl'),
(7513, 'Java'),
(9842, 'DB2'),
(6651, 'VB'),
(6651, 'Java'),
(9006, 'NT'),
(9006, 'Linux');

insert into client values
(Null, 'TelcoInc', '1 Collins St Melbourne', 'Fred Smith', '95551234'),
(Null, 'The Bank', '100 Bourke St Melbourne', 'Jan Tristan', '95559876');

insert into assignment values
(1, 7513, ' 2003-01-20', 8.5);
```

2 Написать скрипты, которые выполняют нижеперечисленные запросы и выводят на экран результаты их работы.

а) Выбрать из таблицы всех работников, вывести на экран их имена и их табельный номер.

```
select name, employeeID from employee;
```

б) Выбрать список всех работников, которые работают программистами:

```
select employeeID, name
from employee
where job=’Программист’;
```

в) Выбрать все рабочие дни работника с табельным номером 6651, в которые он работал более 8 ч.

```
select * from assignment
where employeeID=6651 and hourse > 8;
```

г) Подсчитать количество различных должностей в таблице работников.

```
select count(distinct job) from employee;
```

д) Подсчитать количество различных должностей в таблице работников.

```
select count(*), job
from employee
group by job;
```

е) Выбрать все записи из таблицы работников, которые единственные работают в своей должности.

```
select count(*), job
from employee
group by job
having count(*)=1;
```

ж) Выбрать все записи из таблицы работников и отсортировать их по должности в алфавитном порядке, а по имени в обратном порядке.

```
select *
from employee
order by job asc, name desc;
```

з) Выбрать первые пять записей из таблицы employeeskills

```
select *
from employeeskills
limit 5;
```

6.3 КОНТРОЛЬНЫЕ ВОПРОСЫ

1 Какие функции возвращают числовой код и информационное сообщение об ошибке?

– mysql_error();
– mysql_connect();
– mysql_errno();
– mysql_select_db();
– mysql_affected_row().

2 Какая команда позволяет установить текущую базу данных для заданного соединения?

– mysql_connect();
– mysql_select_db();
– mysql_affected_row();
– mysql_query();
– mysql_fetch_array().

3 Для чего предназначена функция mysql_fetch_array()?

– возвращает количество строк, измененных последним запросом;
– возвращает следующую строку данного результирующего набора в виде массива;
– выбирает текущую базу данных;
– отправляет строку запроса к серверу;
– возвращает сообщение об ошибке.

4 Какая команда возвращает значение AUTO_INCREMENT, сгенерированное последним запросом заданного соединения?

– mysql_select_db();
– mysql_affected_row();
– mysql_insert_id();
– mysql_free_result();
– mysql_fetch_array().

5 Какие функции возвращают следующую строку данного результирующего набора в виде ассоциативного массива?

– mysql_fetch_assoc();
– mysql_fetch_array() с параметром MYSQL_ASSOC;
– mysql_fetch_array() с параметром MYSQL_NUM;
– mysql_affected_row();
– mysql_num_rows().

6 Какая функция возвращает число строк в заданном результирующем наборе?

– mysql_insert_id();
– mysql_num_rows();
– mysql_affected_row();
– mysql_free_result();
– mysql_fetch_array().

7 Какая команда открывает соединение с сервером MySQL?

– mysql_select_db();
– mysql_connect();
– mysql_affected_row();
– mysql_errno();
– mysql_error().

8 Какая команда отправляет строку запроса к серверу MySQL?

– mysql_connect();
– mysql_select_db();
– mysql_affected_row();
– mysql_query();
– mysql_fetch_array().

9 Для чего предназначена функция mysql_affected_row()?

– возвращает количество строк, измененных последним запросом;
– выбирает текущую базу данных;
– отправляет строку запроса к серверу;
– возвращает сообщение об ошибке;
– возвращает количество строк в результирующем запросе.

10 В виде какого массива может вернуть функция mysql_fetch_array() следующую строку результирующего набора?

– в виде ассоциативного (по индексам имен);
– в виде индексного (по числовым индексам);
– по индексам обоих типов;

– нет верных ответов;

– эта функция не возвращает никакого массива.

11 Какие функции возвращают следующую строку данного результирующего набора в виде массива с числовыми индексами?

– mysql_fetch_row();

– mysql_fetch_array() с параметром MYSQL_NUM;

– mysql_fetch_array() с параметром MYSQL_ASSOC;

– mysql_affected_row();

– mysql_num_rows().

ЛИТЕРАТУРА

1 Васкевич, Д. Стратегия клиент/сервер. Руководство по выживанию для специалистов по реорганизации бизнеса / Д. Васкевич. – Киев : Диалектика, 1996.

2 Дилип, Н. Стандарты и протоколы Интернета / Н. Дилип. – М. : Русская редакция, 1999.

3 Дюбуа, П. MySQL / П. Дюбуа. – М. : Изд. дом «Вильямс», 2001.

4 Жданов, А. Dreamweaver 3: краткий курс / А. Жданов, Б. Карпов, М. Левченко. – СПб. : Питер, 2001.

5 Кожемякин, А. А. HTML и CSS в примерах. Создание web-страниц / А. А. Кожемякин. – М. : Альтекс-А, 2004.

6 Котеров, Д. PHP 5 / Д. Котеров, А. Костарев. – СПб. : БХВ-Петербург, 2004.

7 Кузнецов, М. В. PHP 5. Практика разработки web-сайта / М. В. Кузнецов, И. В. Симдянов. – СПб. : БХВ-Петербург, 2005.

8 Кузнецов, М. В. Самоучитель PHP 5 / М. В. Кузнецов, И. В. Симдянов. – СПб. : БХВ-Петербург, 2004.

9 Мещеряков, Е. Публикация баз данных в Интернете / Е. Мещеряков. – СПб. : БХВ, 2001.

10 Орлов, Л. В. Web-сайт без секретов / Л. В. Орлов. – М. : Новый издательский дом, 2004.

11 Томсон, Л. Разработка web-приложений на PHP и MySQL / Л. Томсон. – СПб. : ДиаСофтЮП, 2003.

12 Хоумер, А. Dynamic HTML : справочник / А. Хоумер, К. Улмен. – СПб. : Питер, 2000.

13 Шарма, В. Разработка web-серверов для электронной коммерции / В. Шарма, Р. Шарма. – М. : Изд. дом «Вильямс», 2001.

Св. план 2016, поз. 9

Учебное издание

Алексеев Виктор Федорович
Русак Татьяна Вячеславовна
Пискун Геннадий Адамович

ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

ПОСОБИЕ

Редактор *Е. И. Герман*
Компьютерная правка, оригинал-макет *А. В. Бас*

Подписано в печать 09.02.2017. Формат 60x84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 3,14. Уч.-изд. л. 6,5. Тираж 100 экз. Заказ 250.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014,
№2/113 от 07.04.2014, №3/615 от 07.04.2014.
ЛП №02330/264 от 14.04.2014.
220013, Минск, П. Бровки, 6