

МЕТОДИЧЕСКИЙ КОМПЛЕКС ПО РАБОТЕ С БАЗАМИ ДАННЫХ НА ПРИМЕРЕ СОХРАНЕНИЯ БИОРАЗНООБРАЗИЯ В КЫРГЫЗСТАНЕ171

Статья посвящена актуальной на сегодняшний день проблеме экологии и внедрения WEB-технологий в учебный процесс

БЕРИЛИШТЕР БАЗАСЫНЫН НЕГИЗИНДЕ КЫРГЫЗСТАНДАГЫ БИОЛОГИЯЛЫК ТҮРЛҮҮЛӨРДҮ САКТООНУН МЕТОДИКАЛЫК КОМПЛЕКСИ.

Бүгүнкү күндөгү экологиялык проблемалар жана WEB-технологиясын окуу процессине киргизүү.

METHODICAL COMPLEX FOR WORKING WITH DATABASES ON THE EXAMPLE OF BIODIVERSITY IN KYRGYZSTAN

The article is devoted to the present problem of ecology and implementation of WEB-technologies in the educational process

Начинаем работу с MySQL. В этом комплексе вы должно знать, что такое php MyAdmin и уметь создавать с его помощью базу данных и таблицы в ней:

1. Для чего нужна база данных,
2. Чем отличаются термины «БД» и «СУБД»,
3. Что такое реляционная база данных и из чего она состоит,
4. Познакомитесь с языком баз данных SQL,
5. Научитесь добавлять данные,
6. Удалять данные,
7. Изменять данные
8. Выбирать данные из базы,
9. Какими средствами располагает PHP для работы с MySQL и научитесь ими пользоваться.

Для чего нужна база данных

База данных – неотъемлемая часть современного веб-приложения. В ней хранится вся информация, необходимая для работы сайта. Исключение составляют файлы: изображения, документы. Они обычно и хранятся в виде файлов в предназначенном для них каталоге на сервере. Однако существуют возможность и такую информацию хранить в базе данных, но ее используют редко. Из базы данных можно выбирать информацию с помощью специального языка запросов. Операции выборки происходят очень быстро благодаря специфике внутреннего устройства хранилища данных. Использование базы данных позволяет не беспокоиться о совместном доступе к данным (что обязательно произойдет, если ваш сайт посещают, хотя бы 100 человек в день), все функции синхронизации хранилище берет на себя.

Отличие «БД» от «СУБД»

БД – база данных. Под этим термином понимается информация, которую вы храните. СУБД – система управления базой данных. Это программа, которая предоставляет доступ внешним приложениям к базе данных, обеспечивает ее работу.

Популярные СУБД: Oracle, Microsoft SQL Server, MySQL, Sybase, PostgreSQL. Сайты PHP чаще всего работают в связке с MySQL, именно эту СУБД мы будем

рассматривать в текущем уроке. База данных проектируется и создается для каждого конкретного проекта, СУБД же выбирается из небольшого списка стандартных средств. Фраза «я использую на сайте базу данных MySQL» неверная, правильно сказать «я использую на сайте СУБД MySQL».

Реляционная база данных

Реляционная база данных состоит из таблиц. Таблица имеет ограниченное количество столбцов (обычно небольшое) и сколь угодно много строк. Существуют и другие типы баз данных (не реляционные), но используются крайне редко. Так что про них можно и вовсе не знать. Такую популярность реляционная модель получила ввиду своей простоты. Ее можно использовать практически для любых задач! Слово «реляционная» происходит от английского «relation» («отношение»). Таблицы, из которых состоит база данных, как правило, связаны друг с другом, то есть находятся «в отношении». Предположим, нам нужно хранить информацию о сотрудниках и отделах (это самый распространенный пример). Приведем пример реляционной базы данных, решающих эту задачу. Нам потребуются две таблицы: сотрудники и отделы, назовем их `emps` и `depts` соответственно.

Столбцы таблицы отделов (depts):

- `id_dept` (первичный ключ)
- `name` (название отдела)

Столбцы таблицы сотрудников (emps):

- `id_emp` (первичный ключ)
- `id_dept` (внешний ключ, ссылающийся на таблицу `depts`)
- `first_name` (имя)
- `middle_name` (отчество)
- `last_name` (фамилия)

Таблицы реляционной базы данных характерны тем, что каждая должна содержать первичный ключ. *Первичный ключ* – это столбец с уникальными значениями, то есть по значению этого столбца можно однозначно идентифицировать строку таблицы.

Каждый сотрудник работает в каком-то отделе. Таблица `emps` ссылается на таблицу `depts`

с помощью столбца `id_dept`. Столбец `id_dept` в данном случае называется внешним ключом. Пусть в нашей фирме два отдела: «Бухгалтерия» и «Маркетинг». В бухгалтерии работают Иванов Иван Иванович, Петров Петр Петрович и Сидорова Елена Николаевна. Маркетингом занимаются Ушаков Павел Павлович и Ефремов Илья Викторович.

Таблица `depts`:

```
id_dept name
1 Бухгалтерия
2 Маркетинг
```

Таблица `emps`:

```
id_emp id_dept first_name middle_name last_name
1 1 Иван Иванович Иванов
2 1 Петров Петр Петрович
3 2 Ушаков Павел Павлович
4 2 Ефремов Илья Викторович
5 1 Сидорова Елена Николаевна
```

Язык SQL

SQL (Structured Query Language - «язык структурированных запросов») - универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных. Структурными запросы называются потому, что в общем случае могут содержать вложенные подзапросы. Проще говоря, SQL – это язык, на котором можно общаться с базой данных. Любую операцию, от создания таблицы до выборки данных, возможно, осуществить только посредством запроса на языке SQL. Запрос, как правило, отправляется внешней программой к СУБД. Та в свою очередь выполняет его, производя необходимые операции над базой данных, и возвращает результат.

Запросы делятся на два вида:

- DDL
- DML

К DDL (Data Definition Language – «язык описания данных») относятся запросы, меняющие структуру базы данных. Например, создание таблицы, удаление таблицы, добавление столбца к существующей таблице. К DML (Data Modification Language – «язык модификации данных») относятся запросы, меняющие содержимое базы данных, то есть операции над строками таблиц. Сюда относится вставка, удаление, изменение и выборка строк. Вернемся к базе данных отделов и сотрудников.

С помощью такого запроса можно создать таблицу depts:

```
CREATE TABLE depts
(
  id_dept INT NOT NULL,
  name VARCHAR(32) NOT NULL,
  PRIMARY KEY (id_dept)
)
```

А таким запросом - таблицу emps:

```
CREATE TABLE emps
(
  id_emp INT NOT NULL,
  id_dept INT NOT NULL,
  first_name VARCHAR(32) NOT NULL,
  middle_name VARCHAR(32),
  last_name VARCHAR(32) NOT NULL,
  PRIMARY KEY (id_emp),
  FOREIGN KEY (id_dept) REFERENCES depts (id_dept)
)
```

Обычно DML операции приходится производить один раз – при создании базы данных. Их чаще выполняет не вручную, а с помощью специальных утилит. Для MySQL такой утилитой является phpMyAdmin. С помощью ее графического интерфейса можно создать таблицы, не задумываясь о синтаксисе оператора CREATE TABLE. Скрипт, как правило, работает со строками таблиц, не меняя структуру базы данных, то есть производит только DML операции: вставку, удаление, изменение и выборку строк.

Вставка строк

Для вставки строк в языке SQL служит оператор INSERT. Вот так можно наполнить базу данных сотрудников и отделов:

```

INSERT INTO depts (id_dept, name) VALUES ('1', 'Бухгалтерия');
INSERT INTO depts (id_dept, name) VALUES ('2', 'Маркетинг');

INSERT INTO emps (id_emp, id_dept, last_name, first_name, middle_name)
VALUES ('1', '1', 'Иванов', 'Иван', 'Иванович');

INSERT INTO emps (id_emp, id_dept, last_name, first_name, middle_name)
VALUES ('2', '1', 'Петров', 'Петр', 'Петрович');

INSERT INTO emps (id_emp, id_dept, last_name, first_name, middle_name)
VALUES ('3', '2', 'Ушаков', 'Павел', 'Павлович');

INSERT INTO emps (id_emp, id_dept, last_name, first_name, middle_name)
VALUES ('4', '2', 'Ефремов', 'Илья', 'Викторович');

INSERT INTO emps (id_emp, id_dept, last_name, first_name, middle_name)
VALUES ('5', '1', 'Сидорова', 'Елена', 'Николаевна');

```

PHP за 3 недели © prog-school.ru Страница 5

Удаление строк

Предположим, руководство решило уволить всех сотрудников отдела маркетинга. В этом

случае поможет оператор DELETE, удаляющий строки из таблицы:

```

DELETE FROM emps
WHERE id_dept = '2'

```

Изменение строк

Елена Николаевна вышла замуж за Ивана Ивановича и поменяла фамилию. Для изменения строк таблицы служит оператор UPDATE. Воспользуемся им:

```

UPDATE emps
SET last_name = 'Иванова'
WHERE id_emp = '5'

```

Выборка строк

Для выборки строк служит оператор SELECT. С его помощью можно составлять сложнейшие запросы, выбирающие данные сразу из множества таблиц. В рамках текущего урока рассмотрим лишь самые простые примеры.

Все сотрудники:

```

SELECT *
FROM emps

```

Сотрудники отдела бухгалтерии:

```

SELECT *
FROM emps
WHERE id_dept = '1'

```

Сотрудники отдела бухгалтерии, отсортированные по фамилии, имени, отчеству:

```

SELECT *
FROM emps
WHERE id_dept = '1'
ORDER BY last_name, first_name, middle_name

```

Как зовут сотрудника №1?

```

SELECT last_name, first_name, middle_name
  FROM emps
 WHERE id_emp = '1'
Сколько всего в фирме работает человек?
SELECT count(*)
  FROM emps;

```

Средства PHP для работы с MySQL

```

int mysql_connect (
[ string $server = ini_get("mysql.default_host") [,
string $username = ini_get("mysql.default_user") [,
string $password = ini_get("mysql.default_password") [,
bool $new_link = false [,
int $client_flags = 0]]]])

```

Перед тем, как работать с базой данных, необходимо установить с ней соединение. Для этого служит функция `mysql_connect()`. Результат выполнения функции – дескриптор соединения, который пригодится, только если вы собираетесь работать сразу с несколькими подключениями. В большинстве случаев это не требуется, и результат выполнения функции проверяют лишь на неравенство `false` (что означает, что подключение прошло успешно). Функция принимает множество параметров и все они необязательны. Подробнее о их назначении можно прочитать в документации. Чаще всего функция вызывается с тремя параметрами:

```
mysql_connect($server, $username, $password);
```

Если вы тестируете сайт на локальном компьютере и у вас установлен пакет Денвер, то параметры должны быть следующими (если вы, конечно, не меняли конфигурацию):

```

$server = 'localhost';
$username = 'root';
$password = "";

```

Когда сайт работает на удаленном сервере, параметр `$server` остается «localhost», `$username` и `$password` определяются при создании пользователя базы данных.

```
int mysql_select_db(string $dbname [, int $link_identifier])
```

До того как послать первый запрос серверу MySQL, необходимо указать, с какой базой данных мы собираемся работать. Для этого и предназначена функция `mysql_select_db()`. Она уведомляет PHP, что в дальнейших операциях с соединением `$link_identifier` (или с последним открытым соединением, если указанный параметр не задан) будет использоваться база данных `$dbname`. Функция возвращает `true` в случае успеха, иначе `false`.

```
int mysql_query(string $query [, int $link_identifier])
```

Запрос к базе данных. Текст запроса формулируется на языке SQL. Для запросов на выборку данных функция возвращает идентификатор результата в случае успеха и `false` в случае ошибки. Для запросов, не подразумевающих результат (`INSERT`, `UPDATE`, `DELETE`) функция в случае успеха возвращает `true`.

```
array mysql_fetch_array(int $result)
```

Функция извлекает очередную строку результата выборки данных. В качестве параметра принимает идентификатор, полученной вызовом функции `mysql_query()`. Возвращает массив, состоящий из значений каждого столбца текущей строки. В качестве ключа – порядковый номер столбца (начиная с нуля). Когда извлекать больше нечего, функция возвращает `false`.

```
$server = 'localhost';
```

```

$username = 'root';
$password = "";
$dbname = 'lesson8';
mysql_connect($server, $username, $password);
mysql_select_db($dbname);
$result = mysql_query('SELECT id_dept, name FROM depts');
while ($row = mysql_fetch_array($result))
{
    echo '<li>';
    echo '<a href="dept.php?id_dept=' . $row[0] . ">';
    echo $row[1];
    echo '</a>';
    echo '</li>';
}

```

```

array mysql_fetch_assoc(int $result)

```

Функция аналогична mysql_fetch_array(), но возвращает ассоциативный массив, в котором в качестве ключа выступает имя столбца.

```

$result = mysql_query('SELECT id_dept, name FROM depts');

```

```

while ($row = mysql_fetch_assoc($result))
{
    echo '<li>';
    echo '<a href="dept.php?id_dept=' . $row['id_dept'] . ">';
    echo $row['name'];
    echo '</a>';
    echo '</li>';
}

```

```

int mysql_num_rows(int $result)

```

Функция возвращает число строк, содержащееся в результате выборки данных.

```

$result = mysql_query('SELECT id_dept, name FROM depts');
$count = mysql_num_rows($result);

```

```

for ($i = 0; $i < $count; $i++)
{
    $row = mysql_fetch_array($result);
    echo '<li>';

    echo '<a href="dept.php?id_dept=' . $row['id_dept'] . ">';
    echo $row['name'];
    echo '</a>';
    echo '</li>';
}

```

```

int mysql_affected_rows([resource $link_identifier])

```

Функция возвращает число строк, затронутых последним запросом INSERT, UPDATE или DELETE.

```

mysql_query("DELETE FROM emps WHERE id_dept=2");

```

```

$count = mysql_affected_rows();

```

```

echo 'Уволены все сотрудники из отдела маркетинга. Их было $count чел.';

```

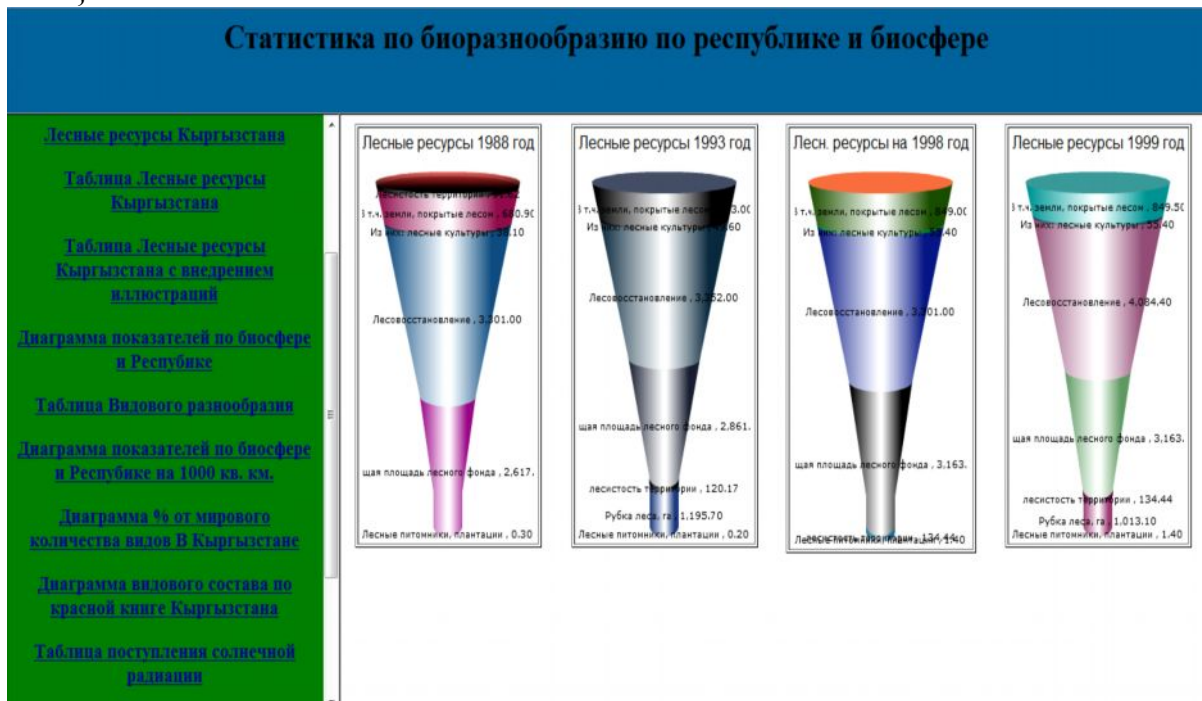
```
int mysql_errno([ int $link_identifier])
string mysql_error([ int $link_identifier])
```

Если в процессе работы с MySQL возникают ошибки, то сообщение об ошибке и ее номер можно получить с помощью этих двух функций. Первая возвращает номер последней зарегистрированной ошибки. Вторая - строку, содержащую текст сообщения об ошибке. Ее удобно применять в отладочных целях.

```
$result = mysql_query($query_text);

if ($result == false)
{
    $err_code = mysql_errno();
    $err_text = mysql_error();

    die("Ошибка MySQL #$err_code: $err_text" .
        "<br/>" .
        "при выполнении SQL запроса: query_text");
}
```



Реальный пример работы с базой данных в табличном виде:

Программный код данного примера:

```
<table border = 0 ><tr>
<td align=center>Лесные ресурсы 1993 год </td>
</tr>
<tr>
<td align=center> </td>
</tr>
</table>
<!-- START Script Block for Chart FactoryMax -->
<div id="FactoryMaxDiv" align="center">
    Chart.
</div>
<script type="text/javascript">
    //Instantiate the Chart
```

```

var chart_FactoryMax = new FusionCharts("FCF_Funnel.swf", "FactoryMax",
"190", "350", "0", "0");
chart_FactoryMax.setTransparent("false");

//Provide entire XML data using dataXML method
chart_FactoryMax.setDataXML("<graph caption='Поступл. солн. Радиации в%'
subCaption='ккал / кв. см' pieSliceDepth='30' showBorder='2' showNames='1'
formatNumberScale='0' numberSuffix='' ><set name='Лесные питомники, плантации
'value='0.2'color='199108244' /><set name='Рубка леса, га
'value='1195.7'color='22264180' /><set name='лесистость территории
'value='120.175'color='246223217' /><set name='Общая площадь лесного фонда
'value='2861.3'color='10222839' /><set name='Лесовосстановление
'value='3352'color='10153245' /><set name='Из них: лесные культуры
'value='49.6'color='140202230' /><set name='В т.ч. земли, покрытые лесом
'value='843'color='226102145' /></graph>")
//Finally, render the chart.
chart_FactoryMax.render("FactoryMaxDiv");
</script>
<!-- END Script Block for Chart FactoryMax -->
</td></td>
</tr>
</table>

```

Работа с PHPMyAdmin с данными по базе в табличном виде:

30 row(s) starting from record # 0
 in horizontal mode and repeat headers after 100 cells
 Sort by key: None
 + Options

	id_product	id_catalog	address	price	table1	hide	number
<input type="checkbox"/>	1	86	В т.ч. земли, покрытые лесом	680.9	rezalt_less1988	show	3
<input type="checkbox"/>	2	86	Общая площадь лесного фонда	2617.8	rezalt_less1988	show	2
<input type="checkbox"/>	3	86	Из них: лесные культуры	38.1	rezalt_less1988	show	4
<input type="checkbox"/>	4	86	Лесные питомники, плантации	0.3	rezalt_less1988	show	5
<input type="checkbox"/>	5	86	Лесистость территории	91.623	rezalt_less1988	show	9
<input type="checkbox"/>	6	86	Лесовосстановление	3301	rezalt_less1988	show	7

30 row(s) starting from record # 0
 in horizontal mode and repeat headers after 100 cells

Резюме

Тема баз данных слишком обширна, однако собранные в документе знания помогут начать изучение этой области и использовать базы данных в ваших проектах уже сейчас!

Требования:

1. На странице просмотра фотографии полного размера под картинкой должно быть указано число ее просмотров.
2. На странице просмотра галереи список фотографий должен быть отсортирован по популярности. В начале списка должны находиться наиболее популярные фотографии.

Популярность определяется числом кликов по фотографии (просмотров увеличенного изображения).

Комментарии:

- при загрузке фотографии ей следует выделять уникальный идентификатор
- помимо идентификатора в базе следует хранить тип картинки («gif», «jpg» и т.д.) и число кликов по ней
- уменьшенные изображения следует хранить в отдельной папке с именами вида «идентификатор.тип»
- изображения полного размера также следует хранить в отдельной папке с именами вида «идентификатор.тип»

ЗАКЛЮЧЕНИЕ

Используя методические указания данного учебного комплекса можно уверенно добиваться усвоения учебного материала WEB- технологий на практических примерах сохранения природы как основного источника жизнедеятельности всего живого. Давая учащимся возможность проявить себя в роли эксперта в той или иной ситуации по результатам работы с базой данных.

Литература:

1. Т.В. Фомина Экология в школе Бишкек 2001 г.211 стр.
2. PHP,MySQL,XML программирование для Интернета 2-издание Санкт-Петербург 2008 г.321 стр.
3. Никитин А. и Ляпин Д. - PHP за 3 недели 2010 видео курс