Главное управление образования Гомельского облисполкома
Учреждение образования
«Гомельский государственный машиностроительный колледж»

Учебная дисциплина
«Веб-программирование для мобильных устройств»

Инструкция
по выполнению лабораторной работы №25
«Установка соединения с БД MySQL. Передача запроса к БД и обработка результата»

Гомель

2021

Составитель: _____ Гаврилова В.М.

Обсуждено и одобрено на заседании цикловой комиссии «Программируемые мобильные системы»

Протокол от «05» октября 2021 № 2

_____ В.М. Гаврилова

Лабораторная работа №25

Тема работы: «Установка соединения с БД MySQL. Передача запроса к БД и обработка результата»

1. Цель работы

Научиться устанавливать соединение с БД MySQL, выполнять передачу запросов и обработку результата посредством PHP-сценария.

2. Задание

Создать базу данных согласно варианту, состоящую минимум из двух связанных таблиц. Создать 5 записей в таблицах базы данных. Разработайте web-интерфейс для работы с базой данных. Создайте php-файл, который может добавлять, удалять/изменять значения в имеющейся базе данных.

№ варианта	Название базы данных	Наименование полей таблиц базы данных
1	Ведомость для начисления зарплаты	Цех, участок, отдел Ф.И.О. Объем выполненной работы Наименование произведенной продукции Расценки на единицу продукции Коэффициент, учитывающий стаж работника Месяц Год
2	Список оборудования	Кафедра Материально ответственное лицо Наименование оборудования Год выпуска оборудования (ГВО) Количество оборудования Стоимость единицы (СТЕ) Срок службы оборудования (ССО) Общая стоимость
3	Библиотечный каталог	УДК (универсальный десятичный код) Отрасль знаний Ф.И.О. автора Наименование издания Тип издания Стоимость Количество Дата поступления
4	Поезда	Номер поезда Время отправления Категория (скорый, пассажирский, местного сообщения) Маршрут следования Длина пути следования в километрах (ДЛ) Стоимость билета для всего пути следования (СТП)

5	Великие люди	1. Ф.И.О. 2. Область деятельности (ученый, полководец, общ. деятель, поэт, художник и т.д.) 3. Год рождения 4. Страна 5. Город 6. Продолжительность жизни
6	Записная книжка-еженедельник	Дата и время мероприятия; Место проведения мероприятия; День недели; Мероприятие Ф.И.О. контактного лица Адрес контактного лица Характеристика контактного лица Телефон контактного лица
7	Справочник автоинспектора	Марка автомобиля Мощность двигателя Габариты автомобиля Номер Цвет Год выпуска Пробег в км Дата прохождения техосмотра
8	Памятка преподавателю	1. Название дисциплины 2. Фамилия преподавателя 3. Название кафедры 4. Объем лекций, в часах 5. Объем лабораторных занятий, в часах 6. Вид контроля - зачет, экзамен 7. Количество студентов
9	Нефтебаза	Типовое название емкости (марка) для хранения топлива Номинальный объем емкости Вес емкости Материал емкости Фактический объем топлива Вид топлива Дата измерения
10	Фотоальбом	Страна (название) Столица Площадь, в млн. кв. км Население, в млн. чел. Континент Население столицы
11	Записная книжка продавца	1. Дата поступления 2. Вид товара 3. Количество 4. Цена за единицу 5. Общая стоимость 6. Дата поступления

12	Памятка агроному-химику	Название культуры Дата посадки Название удобрения Норма внесения на 100 м кв. Дата внесения (по месяцам)
13	Памятка Шерлоку Холмсу	1. Фамилия субъекта 2. Год рождения 3. Вид правонарушения 4. Дата нарушения 5. Сумма штрафа 6. Место совершения правонарушения (район)
14	Протокол технического эксперимента	1. Название эксперимента 2. Дата проведения эксперимента 3. Вид используемого оборудования 4. Время начала эксперимента 5. Время окончания эксперимента 6. Стоимость проведения эксперимента
15	Справочник кинолога	1. Порода собаки 2. Кличка собаки 3. Отец (кличка) 4. Мать (кличка) 5. Дата рождения 6. Рост
16	Памятка туристу	1. Название маршрута 2. Район маршрута 3. Вид туризма 4. Категория (от 1 до 6) 5. Стоимость проезда 6. Вес снаряжения на одного участника 7. Количество участников
17	Памятка дачнику-Овощеводу	1. Вид овощей 2. Название сорта 3. Площадь посева 4. Дата посадки 5. Дата уборки урожая 6. Урожайность - в кг на кв.м.
18	Памятка преподавателю	1. Название дисциплины 2. Фамилия преподавателя 3. Название кафедры 4. Объем лекций, в часах 5. Объем лабораторных занятий, в часах 6. Вид контроля - зачет, экзамен 7. Количество студентов
19	Дневник метеонаблюдений	Дата Пора суток (день, ночь, утро, вечер) Температура Давление Облачность (ясно, слабая, сильная, дождь) Направление ветра (азимут)

20	Справочник по транзисторам	1. Тип транзистора 2. Рабочее напряжение 3. Допустимый ток 4. Коэффициент усиления 5. Стоимость 6. Количество штук
21	Справочник по оборудованию	1. Наименование 2. Количество 3. Страна-изготовитель 4. Стоимость 5. Вес, в кг 6. Дата поставки 6. Занимаемый объем, в куб. дм
22	Великие люди	1. Ф.И.О. 2. Область деятельности (ученый, полководец, общ. деятель, поэт, художник и т.д.) 3. Год рождения 4. Страна 5. Город 6. Продолжительность жизни
23	Памятка любителю музыки	1. Название группы 2. Год образования 3. Лидер группы 4. Страна 5. Название диска 6. Тираж диска
24	Великие даты	1. Дата 2. Вид события (война, революция, образование государства и т.д.) 3. Страна 4. Город 5. Фамилия видного деятеля 6. Примерное число жертв
25	Памятка альпинисту	1. Название вершины 2. Год покорения 3. Высота 4. Страна расположения 5. Даты восхождений 6. Фамилия руководителя группы

3. Оснащение работы

Технические средства обучения:

- IBM – совместимый компьютер;

Электронные средства обучения:

- веб-браузер (Opera, Google Chrome, Mozilla Firefox и др.);
- html-редакторы (Notepad++, Sublime Text и др.);
- web-сервер, интерпретатор PHP.

4. Основные теоретические сведения

MySQL – это реляционная база данных**.** Важная особенность реляционных систем, их отличие от одноуровневых баз данных – возможность располагать данные в нескольких таблицах. Взаимосвязанные данные можно хранить в отдельных таблицах и объединять по ключу, общему для обеих таблиц. Ключ – это *отношение (relation)* между таблицами. Выбор *первичного ключа (primary key)* – наиболее важное решение, принимаемое при разработке новой базы данных. Самое главное, что следует понимать, – вы должны гарантировать уникальность выбранного ключа. Если есть вероятность того, что значение некоторого атрибута может совпасть у двух записей, то его нельзя использовать в качестве первичного ключа. Если таблица содержит ключевые поля из другой таблицы, то между ними образуется связь – взаимоотношением *внешнего ключа (foreign key)*, например «начальник-подчиненный» или «покупатель-покупка».

Нормализация

Представление о взаимоотношениях данных и наиболее эффективном способе их организации называется ***нормализацией***. Нормализация заключается в разделении данных на основе логических взаимоотношений с целью минимизировать дублирование данных. Повторяющиеся данные понапрасну расходуют дисковое пространство сервера и затрудняют их обслуживание. При внесении изменений в повторяющиеся данные есть риск пропустить какие-то из них, что может привести к возникновению несогласованностей в базе данных.

С другой стороны, лучшее – враг хорошего: когда данные хранятся по частям в отдельных таблицах, это может потребовать слишком больших накладных расходов на их извлечение, да и запросы могут получаться чересчур замысловатыми. Главная цель – найти золотую середину.

Формы нормализации

Чтобы нормализовать базу данных, начнем с самых главных правил и будем продвигаться вперед шаг за шагом. Процесс нормализации состоит из трех этапов, называемых *формами*. Первый этап, который называется приведением к первой нормальной форме, должен быть выполнен перед приведением базы данных ко второй нормальной форме. Аналогично, невозможно привести базу данных к третьей нормальной форме, минуя вторую. Процесс нормализации приводит структуру данных в соответствие с тремя нормальными формами.

Первая нормальная форма

Необходимо, чтобы приведенная к первой нормальной форме база данных соответствовала трем требованиям. Ни одна таблица не должна иметь повторяющихся столбцов, содержащих одинаковые по смыслу значения, и все столбцы должны содержать единственное значение. Обязательно должен быть определен первичный ключ, который уникальным образом описывал бы каждую строку. Это может быть один столбец или комбинация из нескольких столбцов, в зависимости от того, сколько потребуется столбцов для обеспечения уникальной идентификации строк.

Вторая нормальная форма

Как уже отмечалось, первая нормальная форма снижает избыточность данных в строке. Вторая нормальная форма ликвидирует избыточность данных в столбцах. Нормальные формы получаются последовательно. Для приведения ко второй нормальной форме необходимо, чтобы таблицы уже соответствовали требованиям первой.

Чтобы привести таблицу базы данных ко второй нормальной форме, нужно определить, какие из ее столбцов содержат одни и те же данные для нескольких строк. Такие столбцы нужно поместить в отдельную таблицу, связав ее с первоначальной по ключу. Другими словами, нужно отыскать поля, не зависящие от первичного ключа.

Третья нормальная форма

Если вы завершили приведение к первой и второй нормальным формам, возможно, вам не потребуется ничего больше делать с базой данных, чтобы привести ее к третьей нормальной форме. Для приведения к третьей нормальной форме нужно просмотреть таблицы и выделить данные, которые не зависят от первичного ключа, но зависят от других значений.

Третья нормальная форма требует, чтобы каждая такая часть данных была выделена в отдельную таблицу.

Типы связей

Взаимоотношения или связи, в базах данных подразделяются на следующие категории:

- связи "один-к-одному" - каждому элементу соответствует один и только один другой элемент;
- связи "один-ко-многим" - каждый ключ из одной таблицы может встречаться несколько раз в другой таблице. Это наиболее распространенный тип связи;
- связи "многие-ко-многим" - возникает между двумя таблицами, когда в каждой из них может присутствовать несколько ключей другой таблицы.

Создание базы данных MySQL

Создадим базу данных «Компьютерные технологии (komp_texn)» с одной таблицей «Продукт (product)».

product / продукт
product_ID / первичный ключ
make / производитель
model / модель
type / тип

Как вы только с этими этапами закончите, переходим к непосредственному созданию реляционной базы данных в локальном доступе – phpMyAdmin.

Сначала нужно запустить Denwer, предполагается, что вы его уже установили, нажав двойным щелчком мыши по ярлыку на рабочем столе «***StartDenwer»***. После чего запустите любой браузер, и в адресной строке введите http://localhost, и в данной странице перейдите по http://localhost/Tools/phpMyAdmin - ***«Проверка MySQL и phpMyAdmin»***.

В появившейся странице, в верхнем меню выберите вкладку «***Базы данных»,*** введите имя создаваемой базы – «*Komp_texnika*» (см. рис.1).

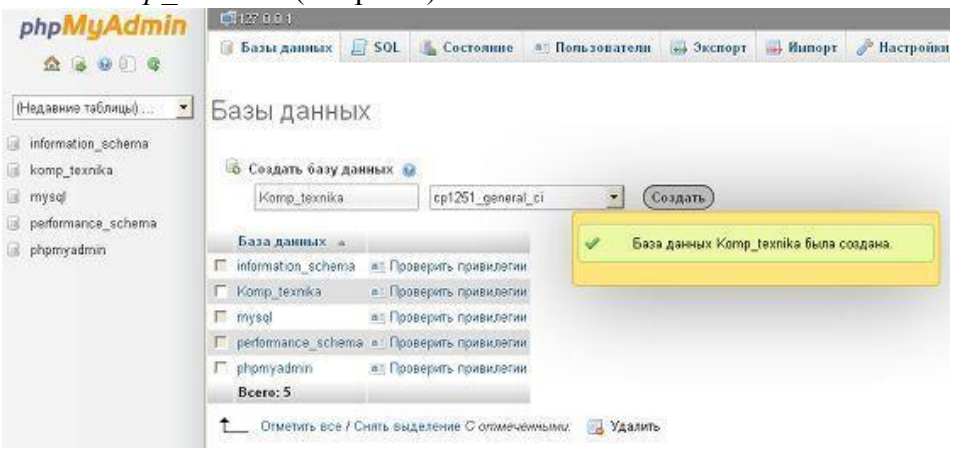

Рисунок 1

После чего переходим к созданию таблицы базы данных, описывая ее название и сколько выделяете характеристик, для этой сущности – *«Product», 4* - столбца (см. рис.2).

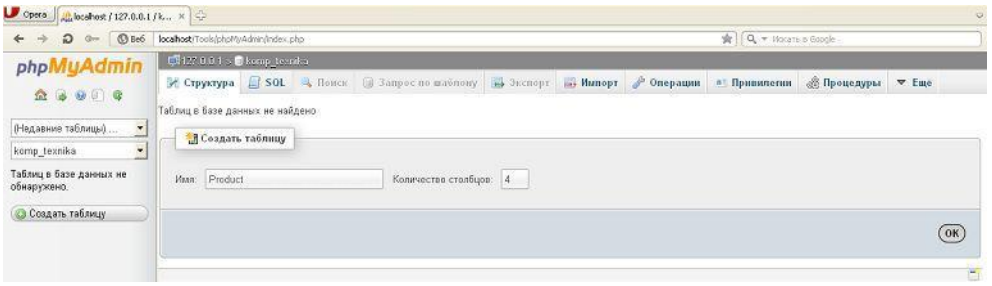

Рисунок 2

Следующим действием, будет описание характеристик/свойств, и указание их типов в соответствующих диапазонах (см. рис.3).

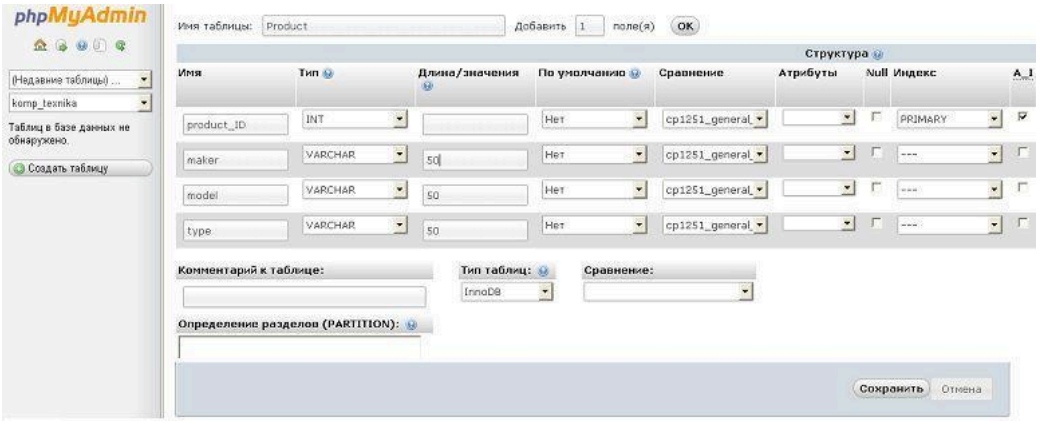

Рисунок 3

Теперь в нашей базе данных имеется пустая таблица *Product*, которую нужно заполнить, для этого в верхнем меню выберите вкладку «***Вставить***» и перед вами откроется страница для заполнения данных полей (см. рис.4).

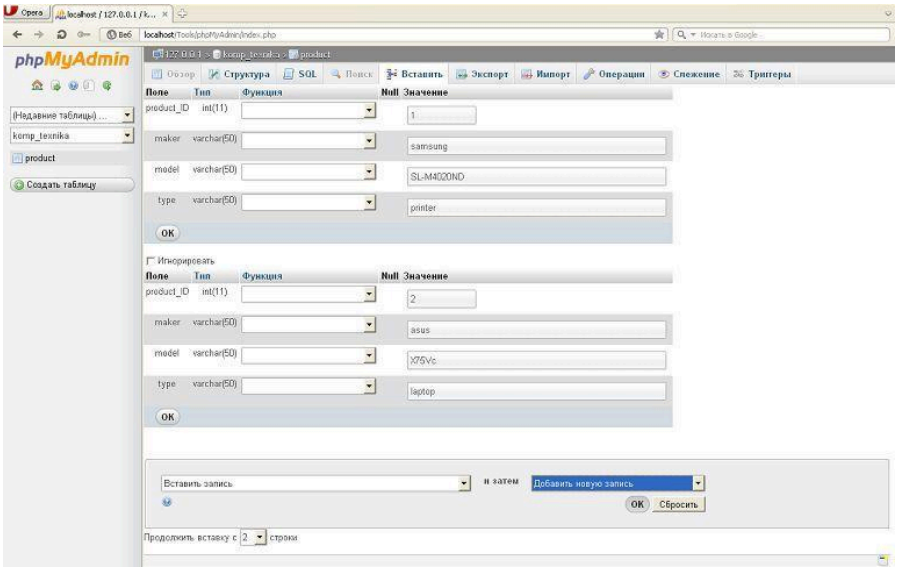

Рисунок 4

Хочу обратить ваше внимание, на то что, в таблицу можно вставить сначала только две записи. Для добавления еще значения, нужно в нижней части страницы в выпадающем списке выбрать пункт ***«Добавить новую запись»*** и нажать ***ОК***.

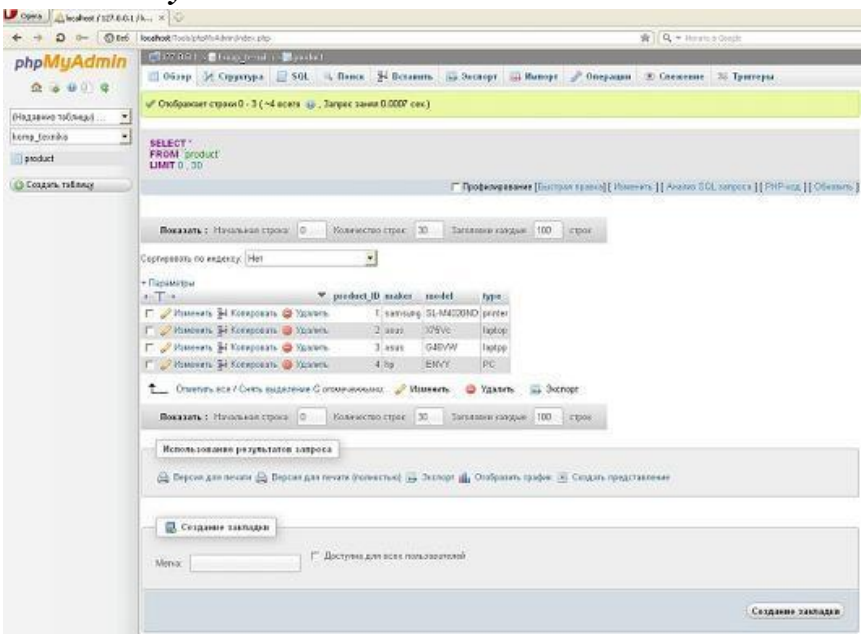

Рисунок 5

Для работы с базой данных необходимо:

1. Установить соединение с сервером:

mysql_connect (server_name, username, password);

2. Выбрать базу данных, которая будет активной:

mysql_select_db (database_name);

3. Выполнить запрос к базе данных

mysql_query (query);

4. Если запросом является выбор информации с БД, то результат необходимо перевести в массив, например:

```
$q = mysql_query(запрос) or die('сообщение об ошибке');
while($row = mysql_fetch_array($q) {
echo "$row['имя_поля']"; }
```

Пусть у нас создана база данных «school», включающая таблицу «user». Таблица содержит пять полей:

```
id_user (счетчик)
    user_name (текст)
    user_login (текст)
    user_password (текст)
    user_e-mail (текст)
    user_info (текст большого объема)
```

Создадим php -файл, который назовем user.php. Он должен иметь следующую структуру:

1. код для подключения к базе данных;
2. код для удаления записи в базе данных;
3. форма для ввода значений в базу данных;
4. код для добавления записи в базе данных;
5. код для вывода значений на экран;

Подключение базы данных

Для того, чтобы добавлять информацию в базу данных необходимо вначале страницы добавить следующие строчки php -кода:

```
<?
   $db = mysql_connect("localhost", "root");
   mysql_select_db("school", $db);
?>
```

Первая строка с помощью функции mysql _ connect () создает объектную переменную $db. Функция имеет два параметра: имя «хоста», где располагается база данных, а второй имя пользователя, имеющего право на соединение с базой (root – это пользователь создаваемый системой и имеющий максимальные права). Во второй строке происходит подключение к конкретной базе данных.

Приведенные строчки будут корректно работать на локальном компьютере, если у вас установлен Denwer, но при попытке перенести ваш сервис на удаленный сервер произойдет ошибка, ведь там база будет располагаться в другом месте, права пользователя также настраивает администратор и он вам не даст прав администратора, значит, эти две строки придется менять. Причем менять во всех php -файлах, использующих подключение к базе данных. Имеет смысл вынести эти четыре строчки в отдельный файл, назвать его connect.php а в основном php -файле сделать ссылку:

```
<? include ("connect.php");  ?>
```

Это не ошибка, для файла, хранящего подключение к базе данных лучше использовать расширение php. Все дело в том, что если вы используете расширение .inc, браузер будет воспринимать его как текстовый файл и если злоумышленник наберет в строке адреса connect.inc

он получит имя базы данных и пароли к ней. Содержимое php — файла, будет скрыто, сервер в ответ на запрос сгенерирует пустую html — страницу.

Создание формы для ввода данных

Для того, чтобы мы могли заполнять данные необходимо создать форму.

```
<H2>Персональная информация:</H2>
<code><form action="user.php" method="get">
<p>имя: <input name="name" size="50" type="text"></p>
<p>логин: <input name="login" size="50" type="text"></p>
<p>пароль: <input name="password" size="50" type="password"></p>
<p>e-mail: <input name="e_mail" size="50" type="text"></p>
<p>информация: <textarea name="info" rows="4" cols="40"></textarea></p>
<p><input name="add" type="submit" value="добавить"><input name="b2" type="reset" value="очистить"></p>
</form>
```

Как видно из кода, данные вводятся в пять полей и при нажатии на кнопку «добавить» переменные формы передаются в тот же самый файл user.php.

Добавление данных в таблицу

```
<?
if (isset($_GET['add'])){
$name = $_GET['name'];
$login = $_GET['login'];
$password = $_GET['password'];
$e_mail = $_GET['name'];
$info = $_GET['info'];
$sql_add = "INSERT INTO users VALUES (null, '$name', '$login', '$password', '$e_mail', '$info')";
mysql_query($sql_add);
print("<p>Спасибо, вы зарегистрированы в базе данных</p>");
}
?>
```

Переменная «add» будет передана в файл user.php вместе с другими переменными и если это произойдет, выполняется четыре действия:

- Из формы будут извлечены все переменные (в приведенном фрагменте они передавались методом GET).
- Создается строковая переменная $sql_add, содержащая sql-запрос. В нем вместо значений полей используются имена извлеченных переменных, в которых хранится имя пользователя, логин и т.д… Каждая переменная взята в апострофы.
- С помощью функции mysql_query() выполняется sql-запрос. Аргументом этой функции является строка с запросом.
- Печатается сообщение о добавлении нового пользователя.

Вывод данных из таблицы на экран

Передоложим, что мы хотим вывести на экран список зарегистрированных пользователей. При этом нам не нужно, выводить персональную информацию, логин, пароль, достаточно только имени и электронного адреса. Для вывода данных можно использовать следующий код

```
<h2>Зарегистрированные пользователи:</h2>
<?
$sql_select  =  "SELECT id_user, user_name, user_email FROM user ORDER BY binary (lower(user_name))";
$result  = mysql_query($sql_select);
$num = mysql_num_rows($result);
print("<P>Всего пользователей: $num.</p>");
while($row = mysql_fetch_array($result)){
print("<p>$row[user_name] - ");
print("<a href=\"user.php?act=delete&id_user= $row[id_user] \">удалить</a>, ");
print("<a href=\"mailto:$row[user_email]\">написать письмо</a></p>");
}
?>
```

Приведенный фрагмент самый сложный. Для более легкого восприятия мы разбили его на три части.

В первой части создается текстовая переменная $sql_select, в которой формируется sql-запрос на выборку данных с сортировкой по полю «user_name». Обратите внимание, в sql-запросе перечислены не все, а только три поля, которые нас интересуют в выборке. После функция mysql_query() выполняет этот запрос, и результат заносит в переменную $result. Можно представить себе, что в этой переменной хранится не одно значение, а таблица значений!

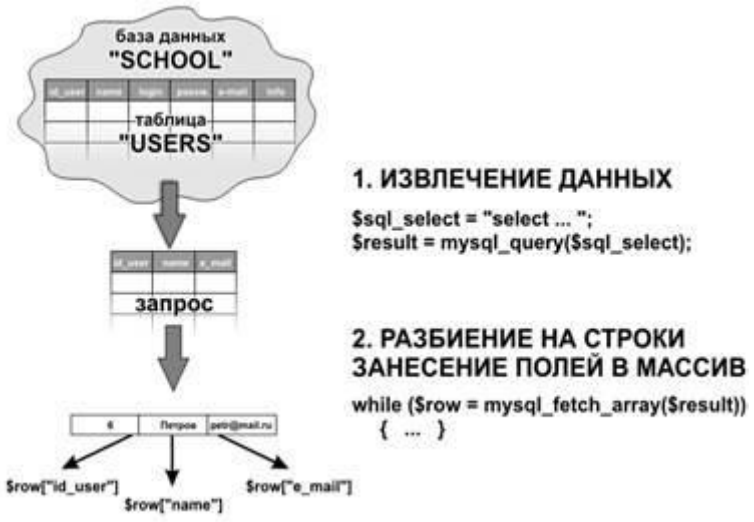

Рисунок 1 - Схема извлечения данных из таблицы.

Во второй части с помощью функций mysql _ num _ rows () считается количество записей извлеченных из таблицы. Аргументом функции является переменная $result. Полученный результат выводится на экран.

В третьей части выбранные данные выводятся на экран. Для этого переменная $result разбивается на строки. Каждая строка содержит данные об одной записи таблицы. До тех пор пока такое разбиение возможно (используется цикл while) каждая срока с помощью функции mysql_fetch_array() разбивается на отдельные значения и каждое значение заносится в массив $row[] (рисунок 1).

Теперь мы можем выводить значения на экран. Обратите внимание, на два момента:

1. Для создания ссылки необходимо использовать тег

```
<a href="/…">…</ a>
```

.

Но у символа кавычки внутри функции print () своя работа – ограничение текстовой строки, которую необходимо выводить на экран. Поэтому в приведенном листинге перед кавычками в теге <a> используется символ «обратный слеш». Он маскирует кавычки html -тега внутри текстовой строки.

2. Действия по удалению данных из таблицы находится в том же самом файле user.php. Поэтому вместе с адресом страницы в теге передается две переменных act (тип действия) и id_user (идентификатор удаляемого пользователя). Значение идентификатора берется из массива $row[id_user].

Удаление данных из таблицы

Код для удаления данных из таблицы необходимо расположить ДО кода вывода пользователей на экран. Дело в том, что когда человек нажмет ссылку «удалить», произойдет перезагрузка файла user. php и данные должны быть удалены из таблицы раньше, чем будет выполнен sql -запрос на выборку данных.

```php
<?
   if ( isset($_GET['act']) ) {
        $id_user = $_GET['id_user'];
      $sql_delete = "DELETE FROM user WHERE id_user= $id_user";
      mysql_query($sql_delete);
  }
?>
```

Программный код выполняется в том случае, если в файл передается переменная $act. В приведенном коде формируется текстовая строка с sql -запросом. Далее этот запрос выполняется с помощью уже известной функции mysql_query().

Полученный программный код несовершенен. Перечислим главные недостатки:

- Вы можете регистрировать одного и того же пользователя многократно, данные будут заноситься в таблицу, изменяя только поле id _ user .
- Вы можете регистрировать пустых пользователей, ведь в программном коде нет проверки, а ввел ли человек обязательные поля.
- Отсутствует возможность изменять введенные данные.

5. Порядок выполнения работы

1. Создать структуру базы данных в phpMyAdmin, состоящую минимум из двух связанных таблиц.
2. Заполните таблицы базы данных значениями.
3. Разработайте web-интерфейс для работы с базой данных.
4. Продемонстрируйте добавление, редактирование и удаление данных в БД.
5. Отобразить содержимое таблиц в браузере.

6. Форма отчета о работе

Номер учебной группы _____
Фамилия, инициалы обучающегося _____
Дата выполнения работы _____
Тема работы: _____
Цель работы: _____
Задание: _____
Оснащение работы: _____
Результат выполнения работы: _____

7. Контрольные вопросы

1 Дайте понятие MySQL?
2 Дайте определение базы данных?
3 Назовите виды баз данных?
4 Что понимают под нормализацией?
5 Перечислите и охарактеризуйте формы нормализации?
6 Назовите и охарактеризуйте типы отношений в БД?
7 Дайте понятие запроса?
8 Для чего предназначен оператор SELECT?
9 Как выполнить добавление данных в БД с помощью SQL?
10 Что такое SQL?
11 Как выполнить редактирование данных в БД с помощью SQL?
12 Как выполнить удаление данных в БД с помощью SQL?

Рекомендуемая литература

1 Колисниченко, Д. PHP и MySQL. Разработка Web-приложений / Д. Колисниченко. - М.: БХВ-Петербург, 2017.

2 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript и CSS / Р. Никсон. - М.: Питер, 2017.

3 Интернет–ресурс php.net