



Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

**Кафедра экономико-математических методов и информационных технологий**

# **БАЗЫ ДАННЫХ**

**Методические указания  
к лабораторной работе № 1 для студентов направления подготовки  
080500 «Бизнес-информатика»**

Ульяновск  
2017

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

## Лабораторная работа №1 (18 часов) Разработка информационной системы для работы с базой данных (PHP, Apache, MySQL)

**Цель работы:** Приобретение навыков доступа к базам данных в сети Интернет, используя возможности PHP. Задачами лабораторной работы являются овладение навыками создания и заполнения таблиц баз данных, создания представлений, триггеров и хранимых процедур, освоение программных технологий доступа к базам данных MySQL с помощью серверных сценариев PHP.

**Задание:** Спроектировать базу данных формата MySQL. Разработать PHP-скрипт обработки данных.

**Формат результата:** Работающий скрипт и заполненная база данных.

### Теоретическая часть


**СУБД MySQL.** Первоначально сервер MySQL разрабатывался для управления большими базами данных с целью обеспечить более высокую скорость работы по сравнению с существующими на тот момент аналогами. И вот уже в течение нескольких лет данный сервер успешно используется в условиях промышленной эксплуатации с высокими требованиями. Несмотря на то, что MySQL постоянно совершенствуется, он уже сегодня обеспечивает широкий спектр полезных функций. Благодаря своей доступности, скорости и безопасности MySQL очень хорошо подходит для доступа к базам данных по Интернету.

**GPL.** MySQL – это программное обеспечение с открытым кодом. Это означает, что применять и модифицировать его может любой желающий. Такое ПО можно получать по Интернету и использовать бесплатно. При этом каждый пользователь может изучить исходный код и изменить его в соответствии со своими потребностями. Использование программного обеспечения MySQL регламентируется лицензией GPL (GNU General Public License), в которой указано, что можно и чего нельзя делать с этим программным обеспечением в различных ситуациях. MySQL можно загрузить с веб-сайта <http://www.mysql.com/>.

**Учетные записи MySQL.** У MySQL есть собственный интерфейс для организации взаимодействия с клиентами, с помощью которого можно перемещать данные и изменять параметры баз данных. Чтобы иметь возможность работать с базой данных, необходимы учетная запись и пароль. Каждый сервер MySQL может содержать несколько баз данных, где группируются таблицы. Если MySQL установлен на локальном компьютере, то по умолчанию именем пользователя является root.

**Форматы таблиц MySQL.** В MySQL можно было выбирать из семи основных форматов таблиц, наиболее распространенными из которых являются ISAM и InnoDB.

Принятым по умолчанию типом таблиц в MySQL является MyISAM. Если попытаться воспользоваться таблицей, которая не была активизирована или добавлена при компиляции, MySQL вместо нее создаст таблицу типа MyISAM. Это очень полезная функция, когда необходимо произвести копирование таблиц с одного SQL-сервера на другой, а серверы поддерживают различные типы таблиц (например, при копировании таблиц на подчиненный компьютер, который оптимизирован для быстрой работы без использования транзакционных таблиц).

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

Таблицы InnoDB в MySQL снабжены обработчиком таблиц, обеспечивающим безопасные транзакции (уровня ACID) с возможностями фиксации транзакции, отката и восстановления после сбоя. Для таблиц InnoDB осуществляется блокировка на уровне строки, а также используется метод чтения без блокировок в команде SELECT. Перечисленные функции позволяют улучшить взаимную совместимость и повысить производительность в многопользовательском режиме. В InnoDB нет необходимости в расширении блокировки, так как блоки строк в InnoDB занимают очень мало места. Важно, что для таблиц InnoDB поддерживаются ограничивающие условия FOREIGN KEY.

**Инструментарий phpMyAdmin.** Инструмент phpMyAdmin позволяет администрировать MySQL с помощью обычного браузера. Все, что требуется для работы с этим инструментом, – это веб-сервер с установленным PHP и база данных MySQL, которую нужно администрировать. Инструмент администрирования позволяет увидеть параметры настройки базы данных и имеющиеся в ней объекты (например, таблицы), а также добавлять новые таблицы при помощи графического интерфейса. С помощью phpMyAdmin можно создавать новые базы данных и таблицы, запускать запросы и просматривать статистику работы сервера.

**Ограничения в MySQL.** Важной особенностью (и недостатком) MySQL является отсутствие поддержки ограничений уровня столбцов таблиц (например, даже такое простое ограничение, как проверка принадлежности числа заданному диапазону). При этом реализация ограничений с помощью программного кода (запросов check) возможна, но не действенна: написанные запросы будут без проблем выполнены СУБД, но проигнорированы при работе с таблицами.

Рассмотренные ранее СУБД Access и SQL Server решали задачу создания ограничений целостности с помощью соответствующих ограничений (типа Check), доступных как в графическом, так и программном (с помощью непосредственного ввода кода) режимах. Те же самые задачи в MySQL решаются намного сложнее и требуют написания соответствующих программных триггеров.

**Триггеры в MySQL.** Поддержка для триггеров включена, начиная с MySQL 5.0.2. Триггер представляет собой именованный объект базы данных, который связан с таблицей, и он будет активизирован, когда специфическое событие происходит для таблицы.

В общем виде программный код создания триггера имеет следующий вид:

```
CREATE TRIGGER trigger_name trigger_time trigger_event
ON tbl_name FOR EACH ROW trigger_stmt
```

В приведенном фрагменте параметр trigger\_time задает время действия. Это может быть BEFORE или AFTER, чтобы задать, что триггер активизируется прежде или после инструкции, которая активизировала это.

Следующий параметр – trigger\_event – указывает вид инструкции, которая активизирует триггер. Здесь trigger\_event может быть одним из следующего:

– INSERT: всякий раз, когда новая строка вставлена в таблицу. Например, через команды INSERT, LOAD DATA или REPLACE.

– UPDATE: всякий раз, когда строка изменяется. Например, через инструкцию UPDATE.

– DELETE: всякий раз, когда строка удалена из таблицы. Например, через инструкции DELETE и REPLACE.

Важно, что не может быть двух триггеров для данной таблицы, которые имеют те же самые время действия и событие. Например, не может быть два триггера BEFORE UPDATE для таблицы, но возможны BEFORE UPDATE и BEFORE INSERT или BEFORE UPDATE и AFTER UPDATE.

Следующий параметр – `trigger_stmt` – задает инструкцию, которая будет выполнена, когда триггер активизируется. Если нужно выполнить много инструкций, используется операторная конструкция `BEGIN ... END`. Это также дает возможность использовать те же самые инструкции, которые являются допустимыми внутри сохраненных подпрограмм.

Схема взаимодействия серверного php-приложения с базой данных MySQL по приведенному алгоритму проиллюстрирована на рис. 1.

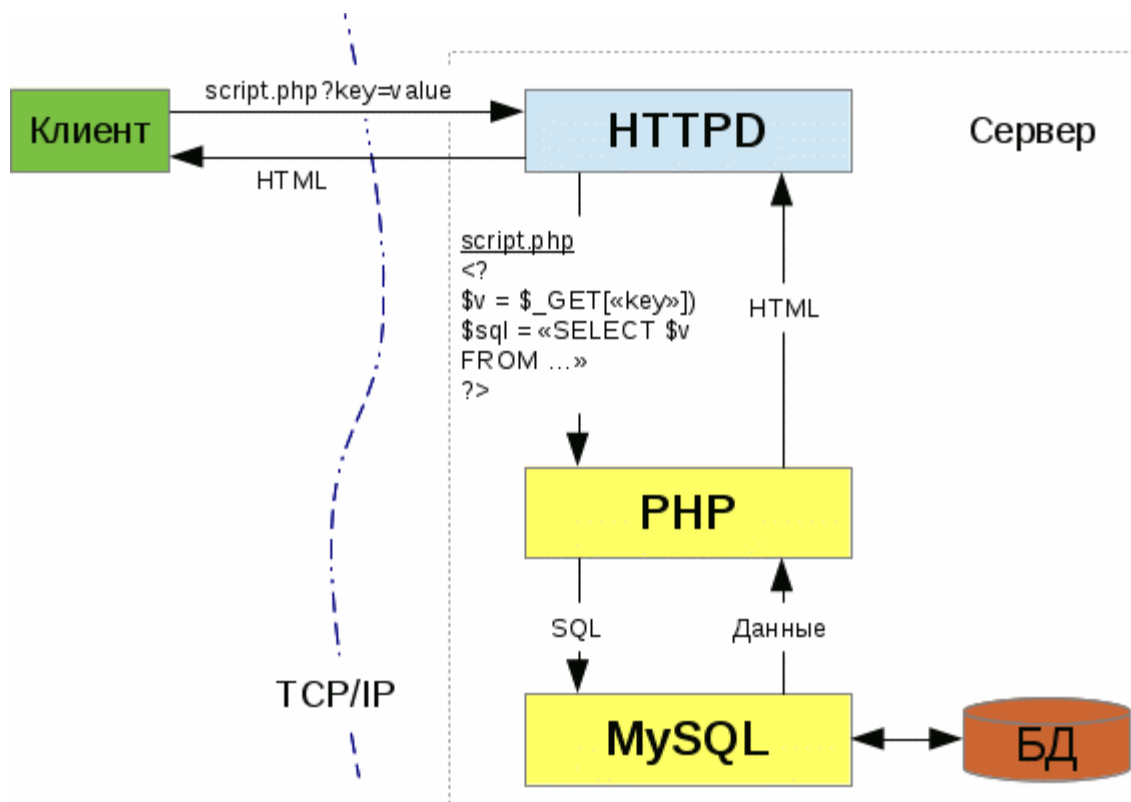



Рис. 1. Взаимодействие php-скрипта и БД MySQL

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

**MySQL и PHP.** Язык PHP (PHP: Hypertext Preprocessor) представляет собой язык сценариев, которые внедряются в страницы HTML для исполнения на стороне сервера. Как правило, для отделения PHP-кода от кода HTML используются символы `<?php код ?>`, реже – `<? код ?>` и `<script language="php">код</script>`, а также инструкции «в стиле ASP» – `<% код %>`.

Web-страница PHP имеет расширение `.php` и состоит из трех разделов:

- директивы страницы – используются для настройки и определяют, как должна обрабатываться страница. Например, так можно задать подключение внешних файлов.
- код – программный код, реализующий выполняемые на сервере операции.
- разметка страницы – это HTML-код страницы, включающий тег `body` и его содержимое.

Поддержка PHP обеспечивается многими серверами, но традиционно такие Web-приложения функционируют на Web-сервере Apache.

Приложения PHP, использующие для хранения информации базу данных, как правило, работают с системой управления базами данных MySQL. Для работы с базами данных PHP располагает достаточно широким набором функций (табл. 2) – от установки соединения с базой данных до извлечения отдельных значений, полученных в результате выполнения запроса.

Таблица 2

Функции взаимодействия с базами данных в PHP-приложениях (СУБД MySQL)

Название класса	Описание класса
<code>mysql_connect</code>	Соединение с источником данных
<code>mysql_select_db</code>	Выбор базы данных для дальнейшей работы
<code>mysql_query</code>	Отправка SQL-запросов серверу
<code>mysql_fetch_array</code>	Представление результата запроса в виде ассоциативного массива
<code>mysql_result</code>	Доступ к отдельному полю записи результата запроса

## А. Создание и заполнение базы данных

### 1. Запустить СУБД MySQL:

Рабочий стол | ярлык Start Denwer ► окно запуска Denwer; дождаться закрытия окна

### 2. Открыть среду PHPMyAdmin:


Запустить браузер ► адресная строка ← `http://localhost` ► окн. «Ура, заработало!» | выбр. <http://localhost/Tools/phpMyAdmin> ► окн. phpMyAdmin; Server: localhost

### 3. Создать новую базу данных *Education*:

ст. MySQL localhost | Create database: | ввести *Education* | Collation ← `utf8_unicode_ci` | кн. Create ► окн. Server: localhost; Database: Education | Database Education has been created.

### 4. В базе данных с помощью конструктора таблиц создать структуру таблицы *Students* (справочник студентов):

- создать новую таблицу *Students* с 11 полями:

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

окн. Server: localhost; Database: Education | Create new table on database Education: | Name ← *Students*; Number of fields ← 11 | кн. Go ► окн. Server: localhost; Database: Education; Table: *Students*;

– задать поле идентификатора студента (счетчик, первичный ключ):

ст. Field ← *stud\_ID*; Type ← *int*; уст. *A\_I*; Index ← *Primary*; перейти на след. строку;

– задать поле номера зачетной книжки (уникальное, 6 цифр):

ст. Field ← *no\_zk*; Type ← *varchar*; Length / Values<sup>1</sup> ← 6;  
Index ← *Unique*; перейти на след. строку;

– задать поля фамилии, имени, отчества (строковые):

ст. Field ← *fam*; Type ← *varchar*; Length / Values<sup>1</sup> ← 20; перейти на след. строку;

аналогично – для имени (*im*) и отчества (*ot*);

– создать уникальный индекс *ФИО* из полей *fam*, *im*, *ot*:

ст. Field *fam* ← Index ← *Index*; перейти на след. строку;

ст. Field *im* ← Index ← *Index*; перейти на след. строку;

ст. Field *ot* ← Index ← *Index*;

– задать поле кода специальности (до 7 букв):

ст. Field ← *spec*; Type ← *varchar*; Length / Values<sup>1</sup> ← 7; перейти на след. строку;

– задать поле курса (одна цифра от 1 до 6):

ст. Field ← *kurs*; Type ← *varchar*; Length / Values<sup>1</sup> ← 1; перейти на след. строку;

– задать поле номера группы (две цифры):

ст. Field ← *gr*; Type ← *varchar*; Length / Values<sup>1</sup> ← 2; перейти на след. строку;

– задать поле даты рождения:

ст. Field ← *data\_r*; Type ← *date*; перейти на след. строку;

– задать поле биографии (мемо):

ст. Field ← *biogr*; Type ← *longtext*; уст. *Null*; перейти на след. строку;

– задать поле фотографии (объект):

ст. Field ← *foto*; Type ← *blob*; уст. *Null*; перейти на след. строку;

кн. Save ► окн. Table `Education`.`Students` has been created.

3. Аналогичным образом создать структуру таблицы *Subjects* (справочник предметов):


– создать новую таблицу базы данных *Education*:

пан. phpMyAdmin | Database ← *Education (1)*;

пан. Create new table on database Education: | Name ← *Subjects*; Number of fields ← 5 |

кн. Go ► окн. Server: localhost; Database: Education; Table: *Subjects*.

– задать поле идентификатора предмета (счетчик, первичный ключ):

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

ст. *Field* ← *predm\_ID*; Тип ← *int*; уст. *A\_I*; Index ← *Primary*; перейти на след. строку;

– аналогичным образом создать остальные поля таблицы:

- *name* (название предмета (строковое));
- *cycle* (цикл дисциплин, к которым относится предмет (строковое));
- *hours* (количество часов (числовое));
- *dep* (название кафедры, на которой ведется преподавание предмета (строковое));

– сохранить структуру таблицы:

кн. Save ► окн. Table `Education`.`Subjects` has been created.

4. Аналогичным образом создать структуру таблицы *Uspev* (успеваемость студентов по предметам) с полями:

- *stud* (идентификатор студента (тип такой же, как в таблице *Студенты*, но не счетчик));
- *predm* (идентификатор предмета (тип такой же, как в таблице *Предметы*, но не счетчик));
- *ocenka* (оценка (символ));
- *data* (дата (дата)).

– ввести составной первичный ключ (поля идентификаторов студента и предмета):

Поле *stud* | Index ← *Primary*; перейти на след. строку;

Поле *predm* | Index ← *Primary*;

– сохранить структуру таблицы:

кн. Save ► окн. Table `Education`.`Uspev` has been created.

5. Задать ограничения для столбцов таблицы *Students* в форме триггера *students\_constraints*, срабатывающего при попытке вставки новой записи в таблицу:

– создать новый SQL-запрос для таблицы *Students*:

пан. phpMyAdmin | Database: ← *Education (1)*; выбр. *Students*; вкл. SQL ►

окн. Run SQL query/queries on database Education | удалить текст «SELECT \* FROM `Students` WHERE 1»;

– ввести SQL-запрос на удаление триггера *students\_constraints\_insert* в случае его существования (для возможности редактирования текста триггера):

```
DROP TRIGGER IF EXISTS `students_constraints_insert` ;
```


– ввести «заголовочный» код создания триггера:

```
delimiter ||
```

```
CREATE TRIGGER `students_constraints_insert` BEFORE INSERT ON `Students`  
FOR EACH  
ROW BEGIN
```

– ввести код, проверяющий, что значение поля *kurs* лежит в пределах от 1 до 6 и заменяющий его на 0 в противном случае:

```
IF not(NEW.kurs >=1 and NEW.kurs <= 6) THEN
```

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

```
SET NEW.kurs = 0;
END IF;
```

– ввести код, проверяющий, что значение поля *spec* вводится русскими буквами и заменяющий его на 0 в противном случае:

```
IF not(NEW.spec >= 'А' AND NEW.spec <= 'я') THEN
SET NEW.spec = 0;
END IF;
```

– ввести код, проверяющий, что значение поля *gr* лежит в пределах от 1 до 99 и заменяющий его на 0 в противном случае:

```
IF not(NEW.gr >='01' AND NEW.gr <= '99') THEN
SET NEW.gr = 0;
END IF;
```

– ввести завершающий код триггера:

```
END ||
delimiter ;
```

6. Аналогичным образом запрограммировать триггер *students\_constraints\_update*, срабатывающий при попытке изменения записи в таблице.

7. Для таблиц *Subjects* и *Uspev* написать по два триггера, проверяющие целостность данных при вставке и изменении данных.

8. Установить связи между таблицами и указать правила ссылочной целостности:

– изменить тип таблицы *Student*:

пан. phpMyAdmin | Database: ← *Education (1)*; выбр. *Students*; вкл. Operations ►  
Storage Engine ← *InnoDB*;

аналогично для всех остальных таблиц.

– указать поле *stud* в качестве внешнего ключа таблицы *Students*, ссылающегося на первичный ключ таблицы *Students*:

пан. phpMyAdmin | Database: ← *Education (3)*; выбр. *Uspev*; вкл. Operations ►  
выбр. Relation View ► ф. Links to | стр. *stud*; Foreign key (InnoDB) ←  
'Education`.`Students`; onDelete ← cascade; onUpdate ← cascade;


аналогичным образом задать внешний ключ, ссылающийся на первичный ключ таблицы *Subjects*.

8. Наполнить базу данных сведениями о студентах (не менее 5), предметах (не менее 3) и оценках (не менее 10):

пан. phpMyAdmin | Database: ← *Education (3)*; выбр. таблицу; вкл. Insert ► заполнить столбец *Value*.

Освоить приемы изменения и удаления полей и записей. Проверить работоспособность ограничений значений полей, уникальности и др., предусмотренные при задании структуры



Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

базы данных. Проверить работоспособность ссылочной целостности, удаляя, изменяя и вставляя данные.

## Б. Представления и хранимые процедуры

1. Создать просмотр для вывода кратких сведений о студентах (идентификатор, номер зачетки, фамилия и инициалы, идентификатор группы):

– создать новый SQL-запрос для таблицы *Students*:

пан. phpMyAdmin | Database: ← *Education (3)*; выбр. *Students*; вкл. SQL ►  
окн. Run SQL query/queries on database Education | удалить текст «SELECT \* FROM `Students` WHERE 1»;

– ввести программный код запроса на извлечение данных:

```
SELECT `stud_ID`,`no_zk`,CONCAT(fam,' ',SUBSTRING(im,1),',',SUBSTRING(ot,1),')
AS FIO, CONCAT(spec,'-',kurs,gr) AS Grupa FROM `Students` WHERE 1
```

– выполнить запрос

окн. Run SQL query/queries on database Education | кн. Go ► результаты на экране;

– сформировать представление по результатам запроса:

ф. Query results operations | CREATE VIEW | VIEW name ← *Students\_info* | кн. Go.

2. Аналогичным образом создать представление для вывода сведений об успеваемости студентов из таблицы успеваемости с указанием сведений о студенте из запроса *Students\_info* и сведений о предмете из таблицы предметов. Результат должен содержать следующие поля: ФИО студента, Группа, Предмет, Дата, Оценка. Назвать представление как *Students\_uspev*. Программный код запроса:

```
SELECT `no_zk`,`FIO`,`Grupa`,`ocenka`,`data` FROM `students_info`,`uspev`,`subjects`
WHERE `students_info`.`stud_ID`=`uspev`.`stud` AND
`subjects`.`predm_ID`=`uspev`.`predm`
```


3. Создать хранимую процедуру для вывода кратких сведений о студенте (идентификатор, фамилия и инициалы, идентификатор группы) по номеру его зачетной книжки:

– создать новый SQL-запрос для базы данных *Education*:

пан. phpMyAdmin | Database: ← *Education (3)*; вкл. SQL ►  
окн. Run SQL query/queries on database Education;

– ввести программный код хранимой процедуры:

```
DELIMITER $$
CREATE PROCEDURE studs(IN IN_ZK VARCHAR(6),OUT V_ID INT, OUT V_ZK
VARCHAR(6),OUT V_FIO VARCHAR(25),OUT V_GR VARCHAR(25))
BEGIN
SELECT stud_id, no_zk, concat(fam,' ',substring(im,1),',',substring(ot,1),'),
concat(spec,'-',kurs,gr)
into V_ID, V_ZK, V_FIO, V_GR
FROM Students
```

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

```
WHERE `no_zk` = IN_ZK;
END;
$$
DELIMITER ;
```

– проверить работоспособность хранимой процедуры:

пан. phpMyAdmin | Database: ← *Education* (3); вкл. SQL ►  
окн. Run SQL query/queries on database Education; ввести код:  
CALL studs('номер зачетной книжки',@V\_ID,@V\_ZK,@V\_FIO,@V\_GR);  
SELECT @V\_ID,@V\_ZK,@V\_FIO,@V\_GR;  
кн. Go ► Результат на экране.

## **В. PHP и MySQL**

1. Запустить программную оболочку Denwer:

Рабочий стол | ярлык Start Denwer ► окно запуска Denwer; дождаться закрытия окна

2. Проверить корректность работы Denwer:

Запустить браузер ► адресная строка ← http://localhost ► окн. «Ура, заработало!»

3. Создать виртуальную директорию для хранения файлов веб-приложения:

Мой компьютер | Локальный диск D | WebServers | denwer | www | denwer |  
создать папку под именем lab3\_группа (например, lab3\_asoi338).  
При наличии уже существующей папки с таким именем добавить постфикс «2»,  
например, lab3\_asoi338\_2.

4. Создать файл index.php начальной страницы веб-приложения:


Директория lab3\_группа | Создать текстовый документ |  
Переименовать файл как index.php

5. Открыть файл index.php с помощью редактора Notepad++:

Директория lab3\_группа | Файл index.php | пр. кн. мыши;  
выбр. Edit with Notepad++ ► окно редактора Notepad++

6. Ввести HTML-разметку страницы index.php:

```
<HTML>
<HEAD><TITLE>Лабораторная работа 3</TITLE></HEAD>
<BODY>
  <H2>Информация о студентах</H2>
  <FORM id="form" method="POST" action="index.php">
    <TABLE border="1" width="60%">
      <TR>
        <TH width="10%">Код</TH>
        <TH width="20%">Зачетная книжка</TH>
        <TH width="40%">ФИО</TH>
        <TH width="30%">Группа</TH>
      </TR>
    <TR align="center">
```

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

```

        <TD width="10%">Значение кода</TD>
        <TD width="20%">Значение зачетки</TD>
        <TD width="40%">Значение ФИО</TD>
        <TD width="30%">Значение группы</TD>
    </TR>
</TABLE>
<BR/> Номер зачетной книжки: <input name="zk" type="text"/>
<input type="submit" value="Запрос"/>
</FORM>
</BODY>
</HTML>

```

#### 7. Проверить работоспособность созданной страницы:

Браузер со стартовой страницей Denwer | строка адреса «<http://localhost/denwer/>» | дополнить названием виртуальной директории (см. п. 3), например, как [http://localhost/denwer/lab3\\_asoi338](http://localhost/denwer/lab3_asoi338) ► страница на экране

#### 8. Создать PHP-сценарий соединения с базой данных Education:

– создать в виртуальной директории и открыть файл connection.php:

Директория lab3\_группа | Создать текстовый документ | Переименовать файл как connection.php | пр. кн. мыши; выбр. Edit with Notepad++ ► окно редактора Notepad++

– ввести программный код сценария:

– ввести открывающий тег сценария:

```
<?php
```

– ввести программный код соединения с локальным сервером:

```
$link = @mysql_connect("localhost", "root")
or die("Невозможно соединиться с сервером");
```

– ввести программный код соединения с базой данных Education:

```
$db=@mysql_select_db("Education") or die("Нет такой базы данных");
```

– ввести закрывающий тег сценария:

```
?>
```


– сохранить сценарий.

#### 9. Дополнить файл index.php PHP-инструкциями:

– подключить сценарий соединения с базой данных:

окно редактора Notepad++ | вкл. Index.php | поставить курсор до первой строки <HTML> | ввести код:

```
<?php
include("connection.php");
```

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

?>

– проверить работоспособность сценария, обновив в браузере страницу index.php.

– выполнить запрос к представлению Students\_info базы данных:

окно редактора Notepad++ | вкл. Index.php | поставить курсор после строки include("connection.php"); | ввести код:

```
$sql = "SELECT * FROM `students_info`";
$query = mysql_query($sql);
```

– организовать цикл по строкам таблицы:

окно редактора Notepad++ | вкл. Index.php | поставить курсор перед второй строкой <TR> | ввести код:

```
<?php
    for($i=0;$i<$count;$i++)
    {
        ?>
```

поставить курсор после второй строки </TR> | ввести код:

```
<?php
    }
?>
```

– выполнить подстановку результатов запроса в строки таблицы:

окно редактора Notepad++ | вкл. Index.php | поставить курсор перед второй строкой с текстом «Значение кода» | заменить текст «Значение кода» на:

```
<?php echo mysql_result($query,$i,stud_id);?>
```

аналогичным образом заменить фрагменты текста «Значение зачетки», «Значение ФИО» и «Значение группы» на фрагменты кода:

```
<?php echo mysql_result($query,$i,no_zk);?>
<?php echo mysql_result($query,$i,FIO);?>
<?php echo mysql_result($query,$i,Gruppa);?>
```

– сохранить файл index.php.

– проверить работоспособность сценария, обновив в браузере страницу index.php.


– добавить инструкции фильтрации данных по номеру зачетной книжки:

– добавить условие, проверяющее, был ли запрос на фильтрацию:

```
окно редактора Notepad++ | вкл. Index.php | поставить курсор
перед строкой с текстом «sql = "SELECT * FROM `students_info`»;» |
ввести код:
    if(!($_POST['zk'] or $_POST['zk']==))
    {
```

– закрыть условный блок (если запроса не было, то будут показаны все записи):

```
окно редактора Notepad++ | вкл. Index.php | поставить курсор
после строки с текстом «$count = mysql_num_rows($query);» | ввести «}»
```

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

– добавить код, выполняющий запрос на фильтрацию:

окно редактора Notepad++ | вкл. Index.php | поставить курсор  
после строки с текстом «};» | ввести код

```
else
{
    $sql = "SELECT * FROM `students_info` where
    `no_zk`='".$$_POST['zk']."'";
    $query = mysql_query($sql);
    $count = mysql_num_rows($query);
}
```

– сохранить файл index.php.

– проверить работоспособность сценария, обновив в браузере страницу index.php.

## 5. Контрольные вопросы

1. Что представляет собой СУБД MySQL?
2. Каковы преимущества СУБД MySQL?
3. Как реализуются триггеры в MySQL?
4. Что представляют собой хранимые процедуры в MySQL?
5. Как реализуется взаимодействие MySQL и PHP?

## 6. Содержание и оформление отчета

Отчет должен содержать:

- титульный лист, название и цель работы;
- скриншоты результатов работы с СУБД MySQL;
- выводы по работе.