

Лекция 8

Работа с БД в РНР

Зачем нужна база данных?

База данных – это инструмент, который позволяет хранить и обрабатывать контент сайта, т.е. все медийные материалы, которые показываются пользователю в браузере.

С помощью СУБД осуществляется выборка, добавление, удаление и изменение данных. Также с использованием базы данных с легкостью решаются такие задачи как поиск по сайту, разбиение на страницы, регистрация и авторизация пользователей. Именно поэтому в основные инструменты веб-разработчика включается СУБД.

С чего начать?

Прежде чем начать создание таблиц, следует определиться со структурой базы данных. На листе бумаги написать названия таблиц и поля, которые потребуются для хранения данных, а также проставить связи между элементами. И только затем уже можно переходить собственно к созданию таблиц и манипулированию данными.

Для этой цели мы будем использовать phpMyAdmin в силу ее распространенности, простоте и доступности. Взять phpMyAdmin можно с официального сайта www.phpmyadmin.net или использовать ту, что установилась с web-сервером OpenServer.

Работа с БД

Пусть БД будет состоять из двух таблиц (одна для лекционного материала, а вторая — для материала лабораторных работ) со следующей структурой:

Таблица Lectures

lecture_id	topic	name	content

Таблица Labworks

labwork_id	lecture_id	name	content

Работа с БД

Далее необходимо создать связь. Для организации связи предполагается использовать поля **lecture_id** обеих таблиц. **ВАЖНО**: заполняем таблицы только после создания всех связей.

Таблица Lectures

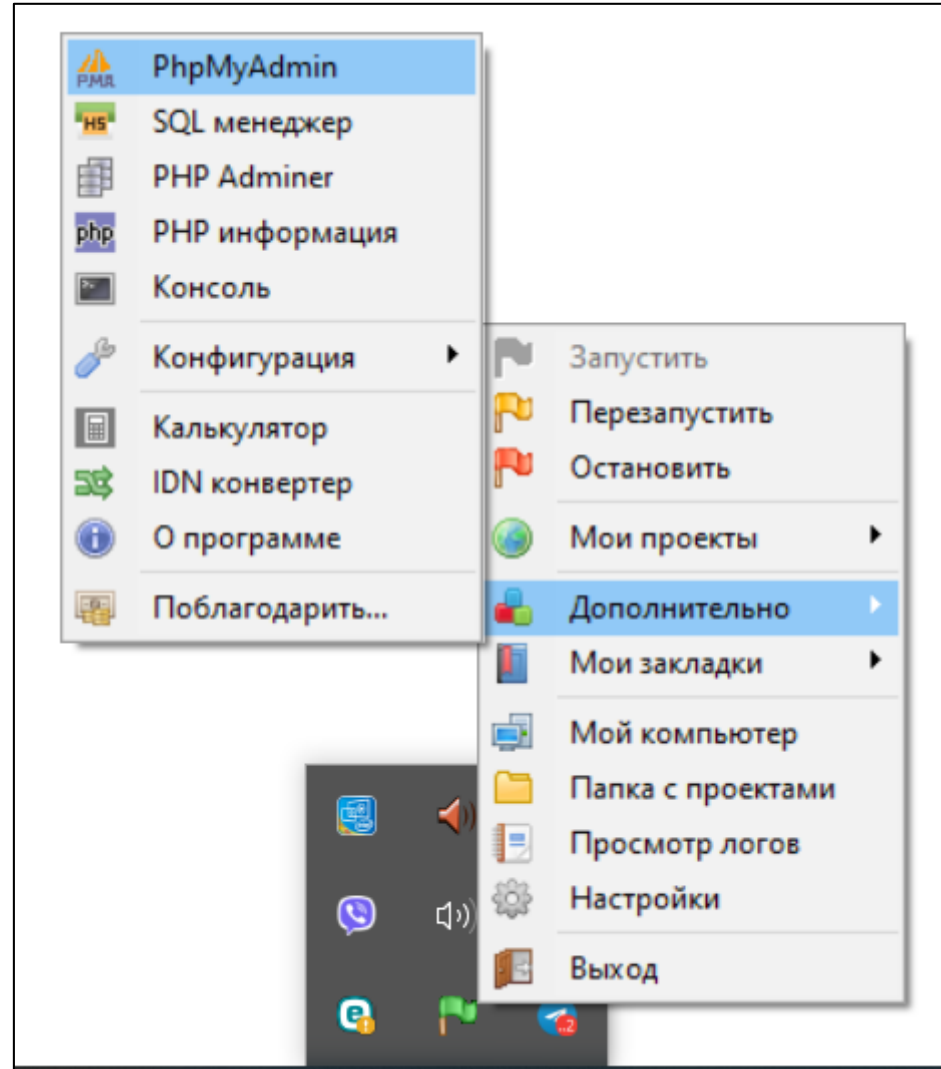
lecture_id	topic	name	content

Таблица Labworks

labwork_id	lecture_id	name	content

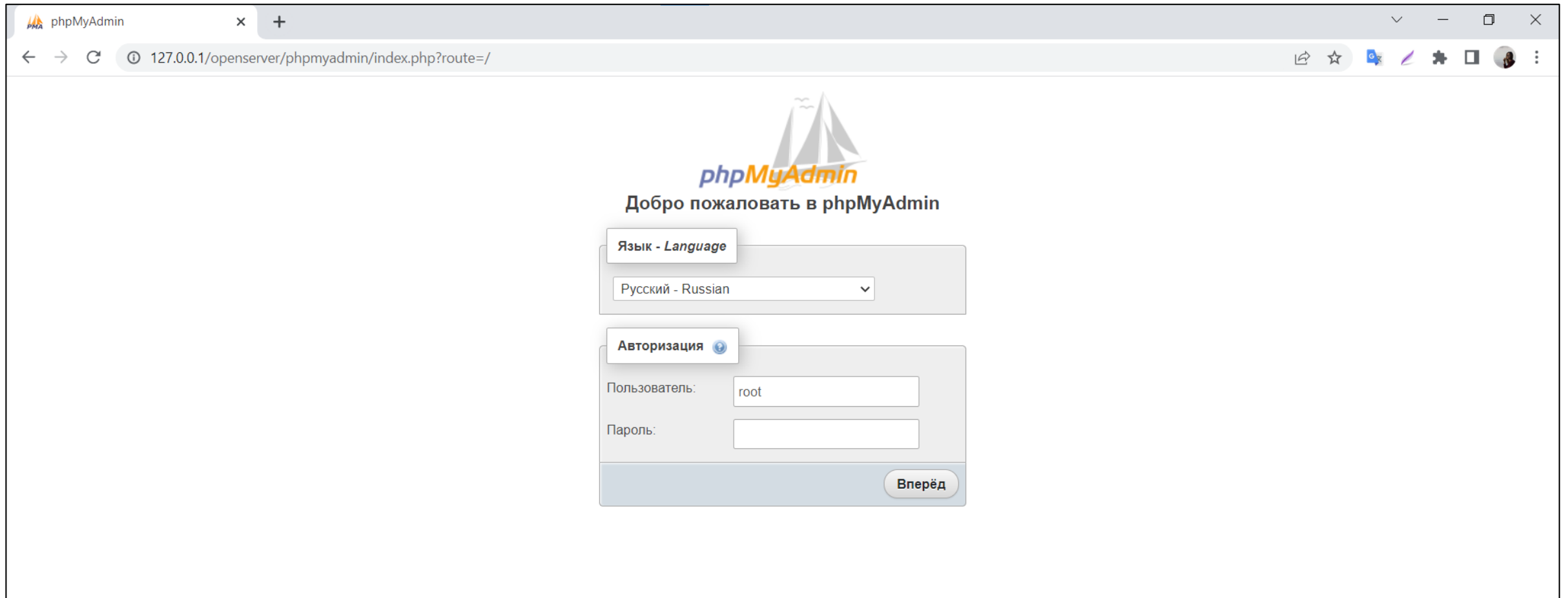
Создание таблиц

Запуск phpMyAdmin происходит следующим образом:



Создание таблиц

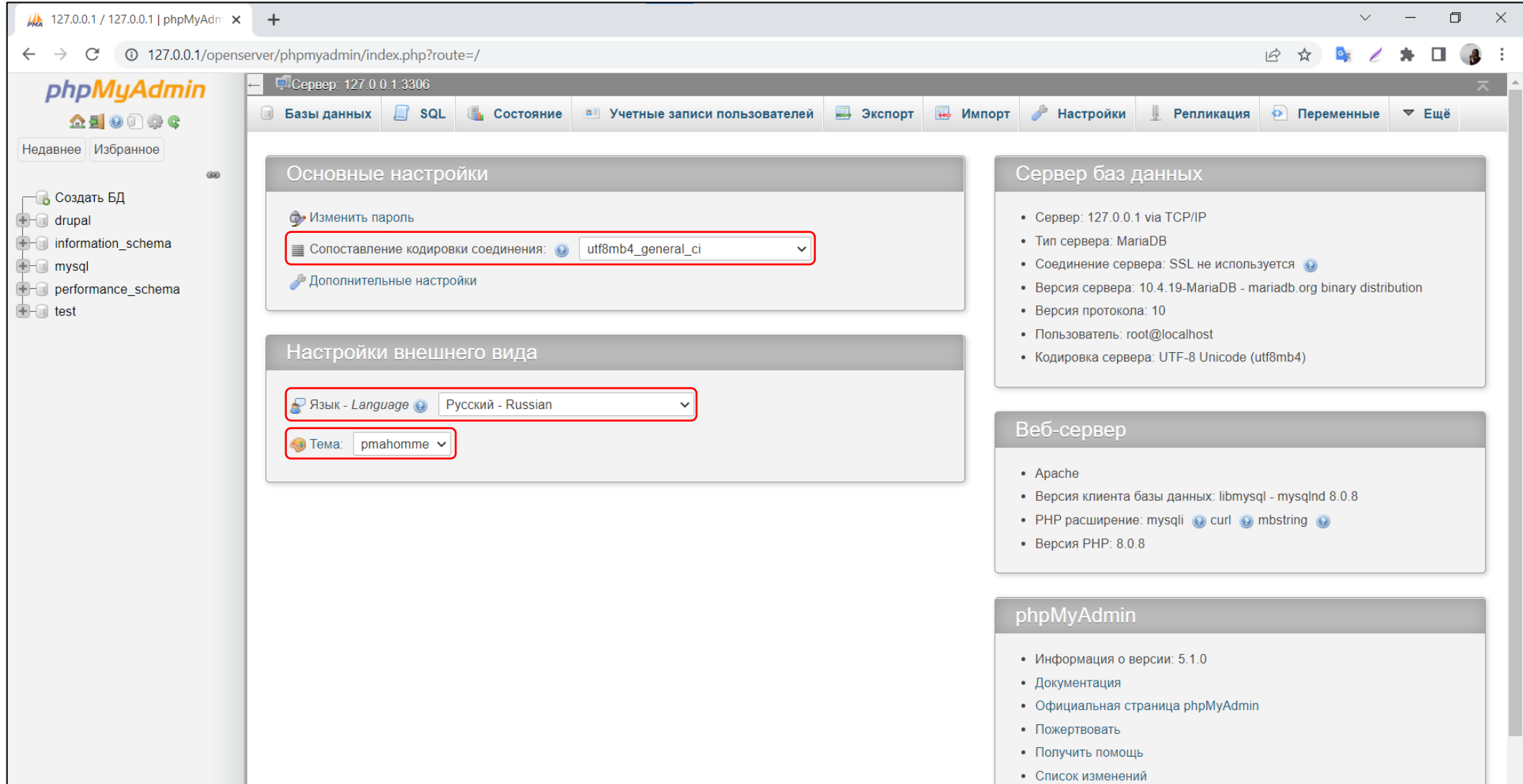
Появится окно с вводом пользователя и пароля. Обычно пользователь – **root**, пароль – **оставляем поле ввода пустым**. Можно выбрать язык.



The screenshot shows the phpMyAdmin login interface in a web browser. The browser's address bar displays the URL `127.0.0.1/openserver/phpmyadmin/index.php?route=/`. The page features the phpMyAdmin logo, which includes a sailboat icon and the text "phpMyAdmin". Below the logo, the text "Добро пожаловать в phpMyAdmin" (Welcome to phpMyAdmin) is displayed. The interface contains two main sections: "Язык - Language" (Language) and "Авторизация" (Authentication). The "Язык - Language" section has a dropdown menu currently set to "Русский - Russian". The "Авторизация" section includes input fields for "Пользователь:" (Username) and "Пароль:" (Password). The "Пользователь:" field contains the text "root", and the "Пароль:" field is empty. At the bottom right of the authentication section is a button labeled "Вперёд" (Next).

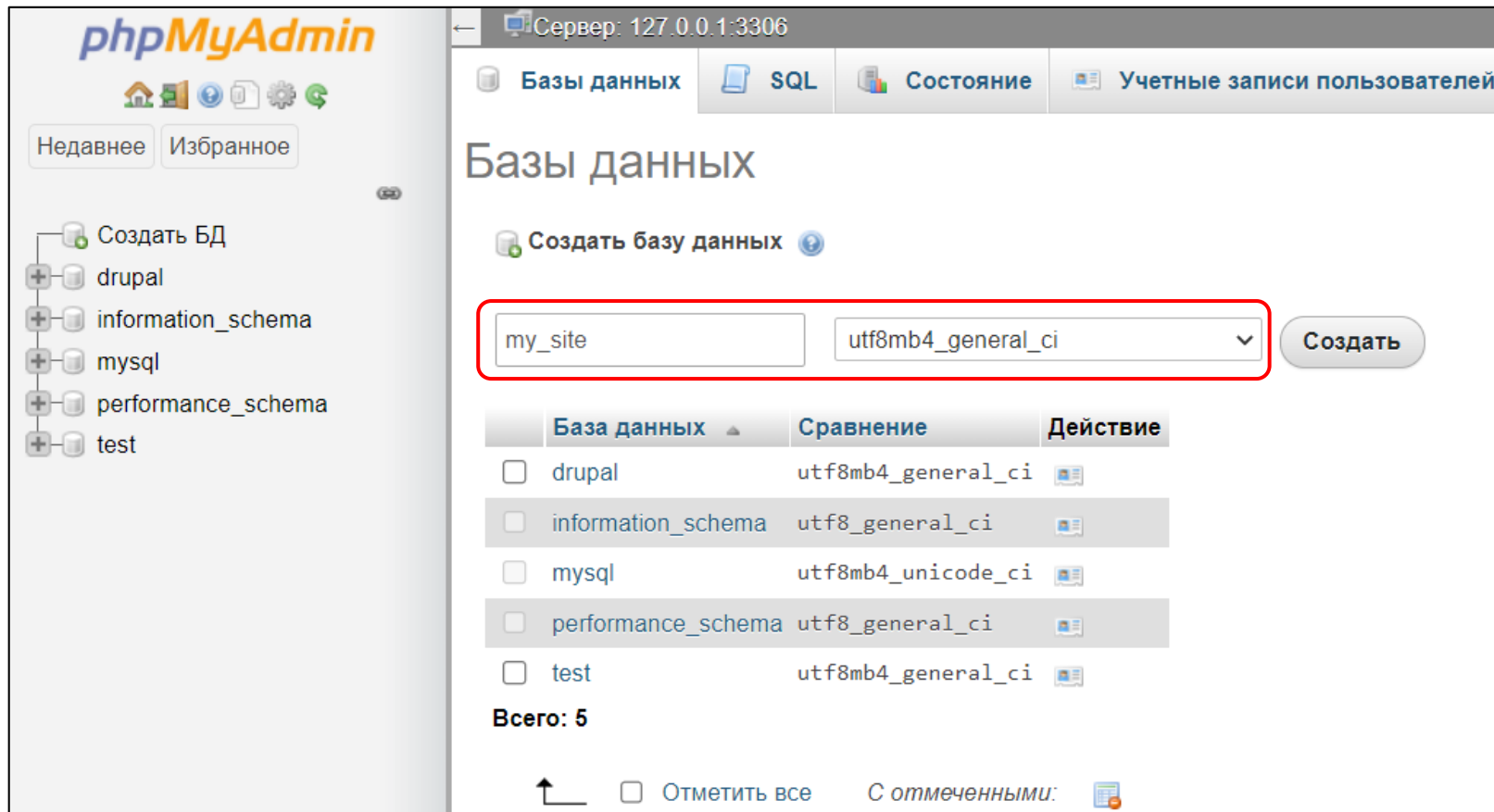
Создание таблиц

На первой странице содержится основная информация об mysql, здесь же можно переключить интерфейс на русский язык, определить кодировку символов, сменить тему.



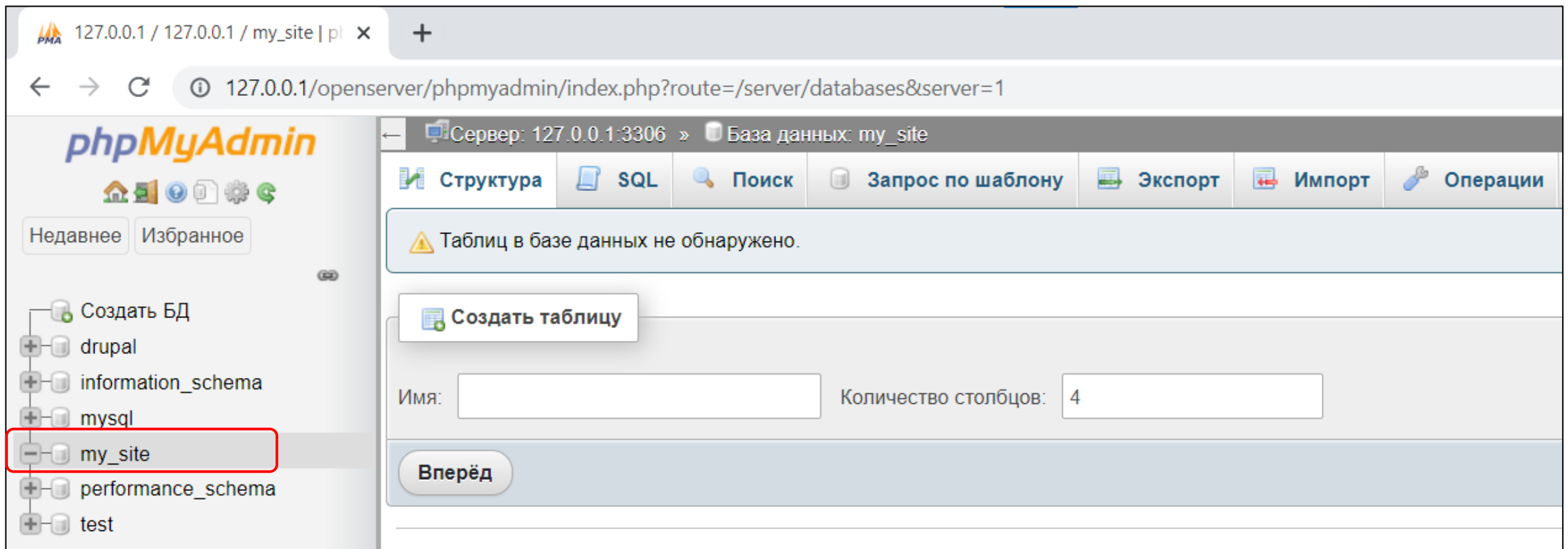
Создание таблиц

Для создания БД выбираем пункт «Создать БД», после чего вводим имя базы данных и нажимаем кнопку «Создать». В результате БД с указанным именем будет создана. Обратить внимание на кодировку.



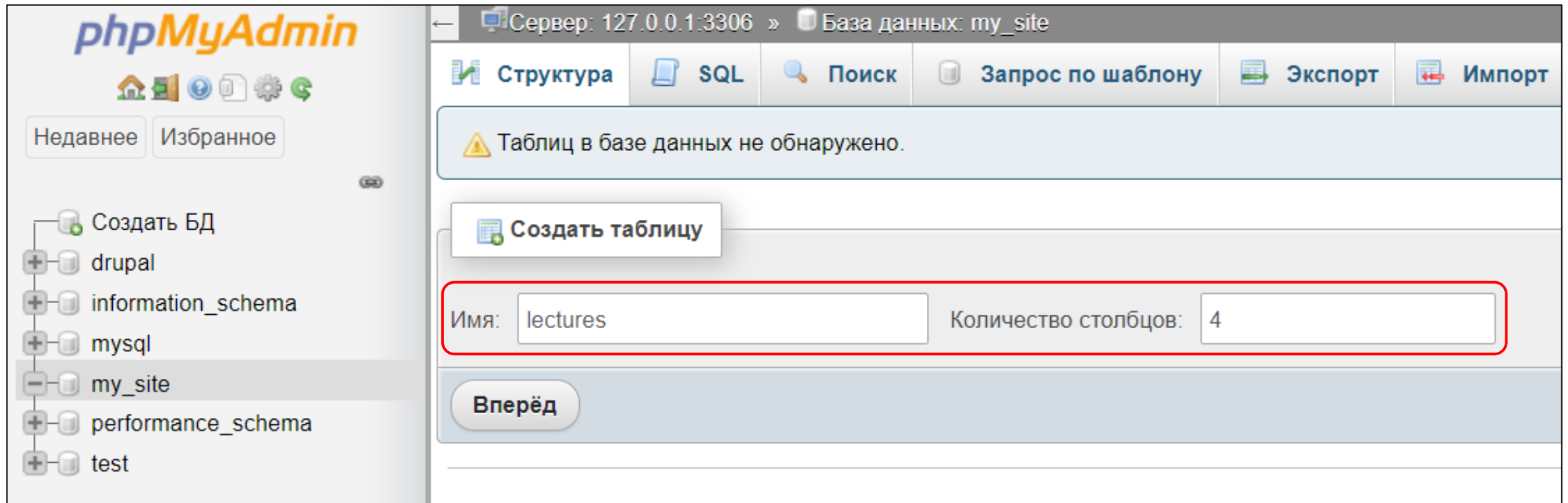
Создание таблиц

Если база уже создана, ее следует выбрать из списка в левом фрейме. БД с именем mysql и некоторые другие создаются автоматически для внутренних целей, поэтому они будут присутствовать в списке баз.



Создание таблиц

Теперь необходимо создать таблицы в соответствии со спроектированной структурой БД. После создания БД автоматически появится запрос с именем новой таблицы и количеством полей. Вводим имя таблицы (например, *lectures*) и число 4. Нажимаем «Вперед».



The screenshot shows the phpMyAdmin web interface. The top navigation bar includes links for 'Структура' (Structure), 'SQL', 'Поиск' (Search), 'Запрос по шаблону' (Query by template), 'Экспорт' (Export), and 'Импорт' (Import). The main content area displays a message: 'Таблиц в базе данных не обнаружено.' (Tables in the database not found). Below this, a 'Создать таблицу' (Create table) button is visible. The form for creating a table is shown with two input fields: 'Имя:' (Name) containing 'lectures' and 'Количество столбцов:' (Number of columns) containing '4'. A red rectangle highlights these two input fields. At the bottom of the form, there is a 'Вперёд' (Next) button. The left sidebar shows a tree view of databases, with 'my_site' selected.

Количество полей можно в любой момент изменить, поэтому, если точное значение не известно, целесообразно поставить меньшее число.

Создание таблиц

После ввода имени, появится таблица для редактирования полей. Введите имена полей, укажите их тип, длину поля, для первого поля задайте атрибут **UNSIGNED**, который означает, что числа будут только положительными.

The screenshot shows the phpMyAdmin interface for creating a new table named 'lectures' in the 'my_site' database. The 'Структура' (Structure) tab is active. The table has four columns: 'lecture_id' (INT, UNSIGNED, PRIMARY), 'topic' (VARCHAR, 255), 'name' (VARCHAR, 255), and 'content' (TEXT). The 'UNSIGNED' attribute for the first column is highlighted with a red box. Below the column list, the 'Структура' section shows the table's encoding as 'utf8mb4_general_ci' and the storage engine as 'InnoDB'. At the bottom, there are buttons for 'Предпросмотр SQL' (Preview SQL) and 'Сохранить' (Save).

Имя	Тип	Длина/значения	По умолчанию	Сравнение	Атрибуты	Null	Индекс	А.!	Комментарии
lecture_id	INT		Нет		UNSIGNED	<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
topic	VARCHAR	255	Нет			<input type="checkbox"/>	---	<input type="checkbox"/>	
name	VARCHAR	255	Нет			<input type="checkbox"/>	---	<input type="checkbox"/>	
content	TEXT		Нет			<input type="checkbox"/>	---	<input type="checkbox"/>	

Структура

Комментарии к таблице: Сравнение: utf8mb4_general_ci Тип таблиц: InnoDB Соединение:

Определение разделов (PARTITION):

Критерий: (Выражение или перечень)

Разделы:

Предпросмотр SQL Сохранить

Создание таблиц

Обязательно надо задать для первого поля `lecture_id` первичный ключ (**PRIMARY**) и, чтобы не изменять значение поля самим, — параметр **auto_increment (A_I)**. Для удобства можно добавить к таблице комментарий.

The screenshot shows the phpMyAdmin interface for creating a new table named 'lectures' in the 'my_site' database. The 'Structure' tab is active, and the 'lecture_id' field is configured as the primary key with the 'A_I' (auto-increment) checkbox checked. The 'name' field is highlighted with a red box, and the 'PRIMARY' index is also highlighted with a red box. The 'utf8mb4_general_ci' collation is selected for the table.

Имя	Тип	Длина/значения	По умолчанию	Сравнение	Атрибуты	Null	Индекс	A_I	Комментарии
lecture_id	INT		Нет		UNSIGNED	<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
topic	VARCHAR	255	Нет			<input type="checkbox"/>	---	<input type="checkbox"/>	
name	VARCHAR	255	Нет			<input type="checkbox"/>	---	<input type="checkbox"/>	
content	TEXT		Нет			<input type="checkbox"/>	---	<input type="checkbox"/>	

Структура

Комментарии к таблице: Сравнение: utf8mb4_general_ci Тип таблиц: InnoDB Соединение:

Определение разделов (PARTITION):

Критерий: (Выражение или перечень)

Разделы:

Предпросмотр SQL Сохранить

Создание таблиц

Обязательно следите за Сравнением, кодировка должна быть такая же, как и в самой БД, в типе таблиц выбираем **InnoDB** (иногда устанавливают тип MyISAM – не будет возможности создавать связи). Если нужно добавить строку, то нажимаем «Вперед».

The screenshot shows the phpMyAdmin interface for creating a new table named 'lectures' in the 'my_site' database. The 'Вставить' (Insert) tab is selected, and a red box highlights the 'Имя таблицы' (Table name) field containing 'lectures', the 'Добавить' (Add) button, a field for '1' field(s), and the 'Вперед' (Next) button.

Имя	Тип	Длина/Значения	По умолчанию	Сравнение	Атрибуты	Null	Индекс	А	Комментарии
lecture_id	INT		Нет		UNSIGNED	<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
topic	VARCHAR	255	Нет			<input type="checkbox"/>	---	<input type="checkbox"/>	
name	VARCHAR	255	Нет			<input type="checkbox"/>	---	<input type="checkbox"/>	
content	TEXT		Нет			<input type="checkbox"/>	---	<input type="checkbox"/>	

Below the table structure, the 'Сравнение' (Collation) is set to 'utf8mb4_general_ci' and the 'Тип таблиц' (Table type) is set to 'InnoDB' (highlighted in the dropdown menu). The 'Соединение' (Foreign key) field is empty.

At the bottom, there are buttons for 'Предпросмотр SQL' (Preview SQL) and 'Сохранить' (Save).

Создание таблиц

Для создания таблицы нажимаем «Сохранить».

The screenshot shows the phpMyAdmin interface for creating a new table named 'lectures' in the 'my_site' database. The interface includes a sidebar with navigation links and a main content area with various tabs and form fields.

Navigation and Tabs:

- Server: 127.0.0.1:3306 » Database: my_site » Table: lectures
- Tabs: Обзор, Структура, SQL, Поиск, Вставить, Экспорт, Импорт, Привилегии, Операции, Триггеры

Table Structure Form:

Имя таблицы: lectures Добавить: 1 поле(я) Вперёд

Имя	Тип	Длина/Значения	По умолчанию	Сравнение	Атрибуты	Null	Индекс	A_I	Комментарии
lecture_id	INT		Нет		UNSIGNED	<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
topic	VARCHAR	255	Нет			<input type="checkbox"/>	---	<input type="checkbox"/>	
name	VARCHAR	255	Нет			<input type="checkbox"/>	---	<input type="checkbox"/>	
content	TEXT		Нет			<input type="checkbox"/>	---	<input type="checkbox"/>	

Структура

Комментарии к таблице: Сравнение: utf8mb4_general_ci Тип таблиц: InnoDB Соединение:

Определение разделов (PARTITION):

Критерий: (Выражение или перечень)

Разделы:

Buttons: Предпросмотр SQL, **Сохранить**

Создание таблиц

Если все сделано правильно, таблица lectures будет создана. Чтобы создать еще одну таблицу, нажимаем «Новая».

phpMyAdmin

Сервер: 127.0.0.1:3306 » База данных: my_site » Таблица: lectures

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии

Структура таблицы Связи

	#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/>	1	lecture_id	int(10)		UNSIGNED	Нет	Нет
<input type="checkbox"/>	2	topic	varchar(255)	utf8mb4_general_ci		Нет	Нет
<input type="checkbox"/>	3	name	varchar(255)	utf8mb4_general_ci		Нет	Нет
<input type="checkbox"/>	4	content	text	utf8mb4_general_ci		Нет	Нет

↑ ☐ Отметить все С отмеченными:

Печать Анализ структуры таблицы ? Переместить поля Нормировать

Добавить 1 поле(я) после content

Основные настройки полей таблицы

Опишем назначение основных настроек и атрибутов полей.

Имя — имя поля. Нельзя использовать зарезервированное слово. То есть нельзя давать полю, например, имя "index".

Тип. Типы полей подробно будут рассмотрено позже.

Длина/значения. Здесь можно указать предельные значения (длины) переменных. Часто используется данная возможность для экономии ресурсов. Например, для пароля я ставлю тип VARCHAR и максимальное количество символов - 32. Тогда как, по умолчанию, VARCHAR позволяет хранить до 255 символов.

По умолчанию. Полезная опция, позволяющая задать значение поля по умолчанию.

Сравнение (Кодировка). Разумеется, данная настройка актуальна только для строковых типов. Если будете хранить в этом поле строки из русского и латинского алфавита, а также и другие символы, то используйте (utf8mb4_general_ci).

Создание таблиц

Атрибуты. Относится только к числам. Если число заведомо положительное, то включите опцию "**UNSIGNED**", что позволит расширить положительный диапазон значений выбранного типа. Также есть опция "**UNSIGNED ZEROFILL**", которая редко используется, но делает она следующее: заполняет нулями все неиспользованные старшие разряды. То есть, допустим, у Вас число 4, однако, храниться будет: "0000000004", где количество нулей определяется свободными разрядами выбранного типа.

Checkbox Null. Эта опция позволяет включить возможность значения "null". "null" - это не 0, и не пустая строка. Это пустой объект.

Index. Тут возможны несколько вариантов.

Первичный ключ (**PRIMARY**). Как правило, этим ключом назначают поле ID. Означает этот ключ, что значение поля уникально для всей таблицы, более того, по этому полю создаётся индекс.

Индекс (**INDEX**). Собственно, создавать или нет для этого поля индекс (будет необходим для организации связей).

Уникальное (**UNIQUE**). Если поставить эту опцию, то значение данного поля должно быть уникальным.

ПолнТекст (**FULLTEXT**) означает, создавать полнотекстовый индекс или нет.

Создание таблиц

Checkbox **A_I**. Позволяет задать "auto_increment". Это очень важная возможность, которую целесообразно включать практически в каждой таблице. Эта опция позволяет при добавлении новой записи автоматически увеличивать на 1 значение поля. Идеальный вариант для ID (уникального идентификатора), который должен быть уникален для каждой записи. И чтобы не думать об уникальности этого поля, достаточно просто включить опцию "auto_increment".

Комментарии. Это поле можете заполнять на своё усмотрение.

Создание таблиц

Точно так же создаем и вторую (**labworks**) (остальные) таблицы БД.

The screenshot shows the phpMyAdmin interface for creating a new table named 'labworks' in the 'my_site' database. The interface includes a sidebar with navigation options like 'Создать БД', 'drupal', 'information_schema', 'mysql', 'my_site', 'Новая', 'lectures', 'performance_schema', and 'test'. The main area displays the table structure configuration for 'labworks'.

Сервер: 127.0.0.1:3306 » База данных: my_site

Имя таблицы: labworks Добавить 1 поле(я) Вперёд

Имя	Тип	Длина/значения	По умолчанию	Сравнение	Атрибуты	Null	Индекс	А	Комментарии
labwork_id	INT		Нет		UNSIGNED	<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
lecture_id	INT		Нет		UNSIGNED	<input type="checkbox"/>	---	<input type="checkbox"/>	
name	VARCHAR	255	Нет			<input type="checkbox"/>	---	<input type="checkbox"/>	
content	TEXT		Нет			<input type="checkbox"/>	---	<input type="checkbox"/>	

Структура

Комментарии к таблице: Сравнение: Тип таблиц: Соединение:

Лабораторные работы utf8mb4_general_ci InnoDB

Определение разделов (PARTITION):

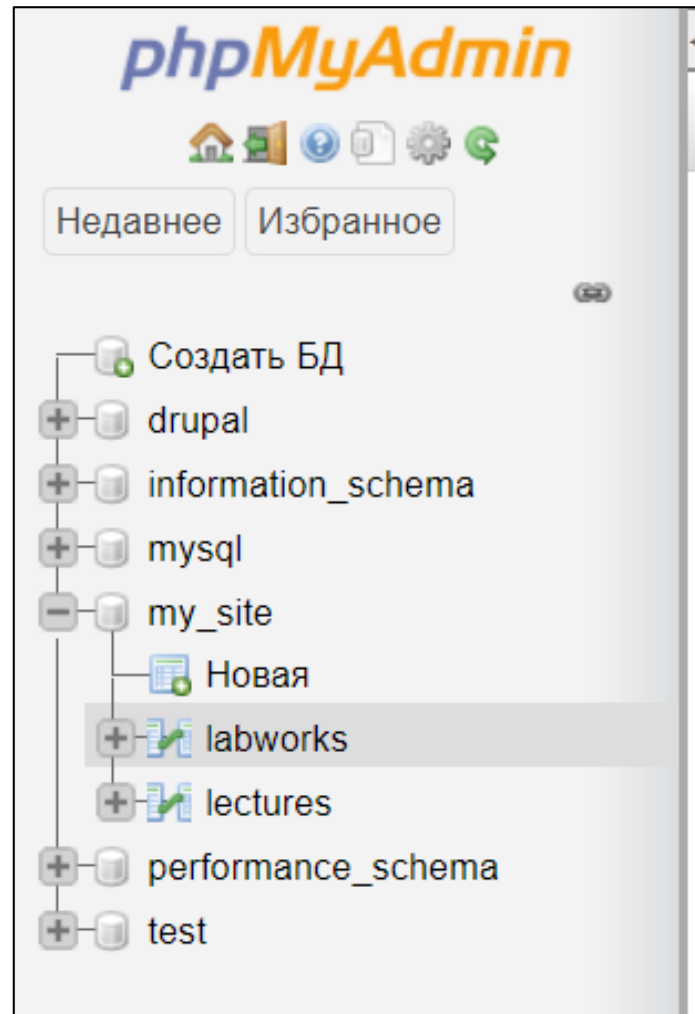
Критерий: (Выражение или перечень)

Разделы:

Предпросмотр SQL Сохранить

Создание таблиц

Теперь в нашей базе данных содержится 2 таблицы.



Изменение таблиц

Всегда есть возможность изменить существующую таблицу — добавить новые поля или изменить существующие. Для этой цели выберите закладку «Структура», где доступны следующие варианты. Чтобы изменить существующие поля, необходимо отметить возле них **checkbox** и нажать на значок карандаша. Откроется уже знакомая схема, где модифицируются параметры поля таблицы.

Обзор

Структура

SQL

Поиск

Вставить

Экспорт

Импорт

Привилегии

Структура таблицы

Связи

	#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/>	1	labwork_id	int(10)		UNSIGNED	Нет	Нет
<input type="checkbox"/>	2	lecture_id	int(10)		UNSIGNED	Нет	Нет
<input checked="" type="checkbox"/>	3	name	varchar(255)	utf8mb4_general_ci		Нет	Нет
<input checked="" type="checkbox"/>	4	content	text	utf8mb4_general_ci		Нет	Нет

↑

Отметить все

С отмеченными:

Печать

Анализ структуры таблицы

?

Переместить поля

Нормировать

Добавить

1

поле(я)

после content

Вперёд

Изменение таблиц

Откроется уже знакомая схема, где можно изменить только выбранные поля таблицы.

Сервер: 127.0.0.1:3306 » База данных: my_site » Таблица: labworks "Лабораторные работы"

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Имя	Тип	Длина/Значения	По умолчанию	Сравнение	Атрибуты	Null	A_I	Комментарии
name	VARCHAR	255	Нет	utf8mb4_general_		<input type="checkbox"/>	<input type="checkbox"/>	
content	TEXT		Нет	utf8mb4_general_		<input type="checkbox"/>	<input type="checkbox"/>	

Структура

Предпросмотр SQL Сохранить

Изменение таблиц

Добавление нового поля производится при выборе той же закладки «Структура», где следует выбрать количество добавляемых полей и после какого поля они будут размещаться.

Структура таблицы

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии
<input type="checkbox"/>	1 labwork_id	int(10)		UNSIGNED	Нет	Нет	
<input type="checkbox"/>	2 lecture_id	int(10)		UNSIGNED	Нет	Нет	
<input type="checkbox"/>	3 name	varchar(255)	utf8mb4_general_ci		Нет	Нет	
<input type="checkbox"/>	4 content	text	utf8mb4_general_ci		Нет	Нет	

↑ ☐ Отметить все С отмеченными: [иконки]

Печать Анализ структуры таблицы [иконка] Переместить поля [иконка] Нормировать [иконка]

Добавить 1 поле(я) [выпадающий список]

- после content
- в начале таблицы
- после labwork_id
- после lecture_id
- после name
- после content

Вперёд

Индексы

Действие	Имя индекса	Тип	Упакован	Столбец	Уникальных элементов
----------	-------------	-----	----------	---------	----------------------

Создание связей между таблицами

Почему же связи удобно держать в самой базе данных? Ведь эту задачу может решать и само приложение. Все дело в ограничениях и действиях при изменении, которые можно наложить на связи.

Например, можно запретить удалять лекцию, если с ней связана хотя бы одна лабораторная работа. Или удалить все лабораторные работы, если удалена тема лекции, с которой они связаны. В любом случае, с помощью связей повышается отказоустойчивость и надежность приложения.

Для организации связей движок таблиц должен быть **InnoDB**. Только он поддерживает внешние ключи (foreign key).

Создание связей между таблицами

Чтобы создать внешний ключ, выбираем таблицу, в которую вы хотите добавить внешний ключ, на вкладке «Структура» нажимаем на «Связи». Выбираем поле для связи с таблицей и действия при изменении записей в таблицах. Нажимаем «Сохранить». Если все сделано верно, внешний ключ будет создан.

The screenshot shows the 'Связи' (Relationships) tab in a database management tool. The main section is titled 'Ограничения внешнего ключа' (Foreign Key Constraints). It contains a table with columns: 'Действия' (Actions), 'Свойства ограничения' (Constraint Properties), 'Столбец' (Column), and 'Ограничение внешнего ключа (INNODB)' (Foreign Key Constraint (INNODB)). The 'Ограничение внешнего ключа (INNODB)' section is expanded, showing a table with columns: 'База данных' (Database), 'Таблица' (Table), and 'Столбец' (Column). The 'Столбец' column is set to 'lecture_id'. The 'База данных' column is set to 'my_site'. The 'Таблица' column is set to 'lectures'. The 'Столбец' column is set to 'lecture_id'. Below the table, there are buttons for 'Предпросмотр SQL' (Preview SQL) and 'Сохранить' (Save).

Действия	Свойства ограничения	Столбец	Ограничение внешнего ключа (INNODB)		
			База данных	Таблица	Столбец
	Ограничения внешнего кл ON DELETE RESTRICT ON UPDATE CASCADE	lecture_id + Добавить столбец	my_site	lectures	lecture_id

+ Добавить ограничение

Предпросмотр SQL Сохранить

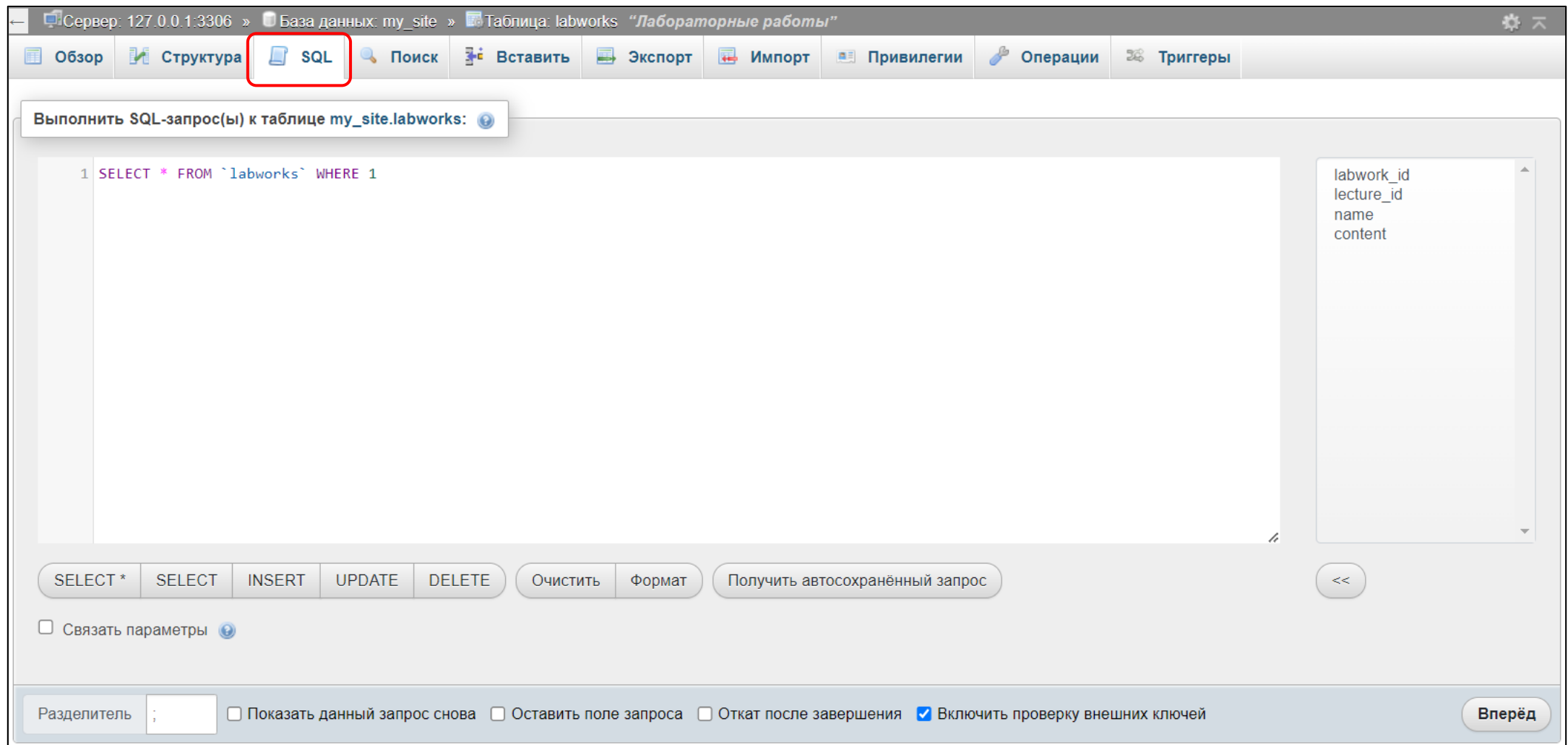
Создание связей между таблицами

Изменять таблицы, создавать связи и так далее можно делать через SQL-запросы. Соответственно через SQL-запрос для создания внешнего ключа будет выглядеть следующим образом:

```
ALTER TABLE `labworks` ADD FOREIGN KEY (`lecture_id`) REFERENCES `lectures`(`lecture_id`) ON DELETE RESTRICT ON UPDATE CASCADE;
```

Создание связей между таблицами

Для работы с SQL-запросами нужно открыть вкладку SQL



Создание связей между таблицами

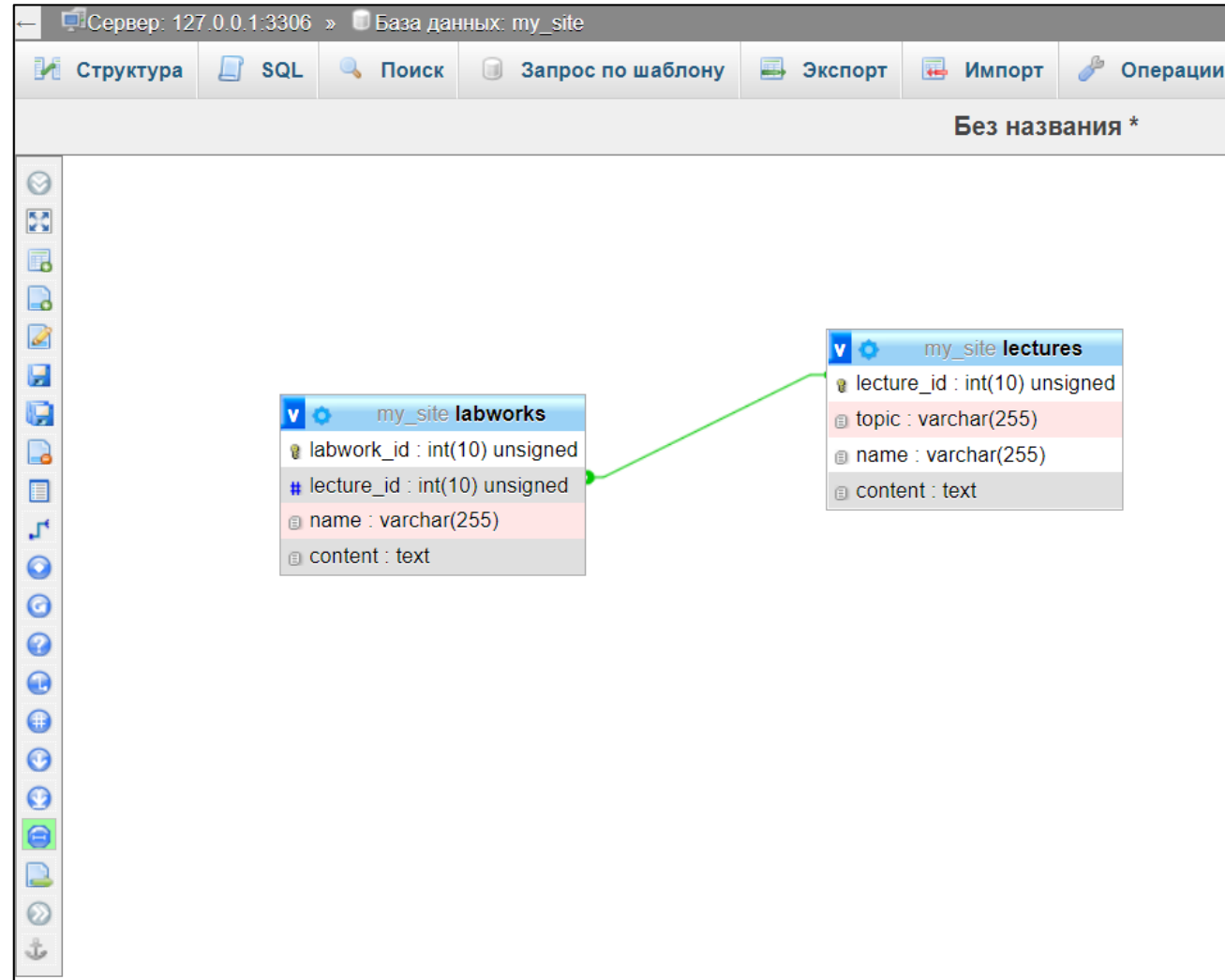
Посмотреть созданную связь можно через «Еще» -> «Дизайнер»

The screenshot shows a database management application window. The title bar indicates the server is 127.0.0.1:3306 and the database is my_site. The main menu bar includes options like Структура, SQL, Поиск, Запрос по шаблону, Экспорт, Импорт, Операции, Привилегии, Процедуры, События, Триггеры, and Ещё. The 'Ещё' (More) menu is open, showing the 'Дизайнер' (Designer) option. Below the menu, there is a 'Фильтры' (Filters) section with a search box. A table list is displayed with columns: Таблица, Действие, Строки, Тип, Сравнение, Размер, and Фрагментировано. The table list shows two tables: labworks and lectures. At the bottom, there are links for 'Печать' (Print) and 'Словарь данных' (Data Dictionary).

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
labworks	[Icons]	0	InnoDB	utf8mb4_general_ci	32.0 КиБ	-
lectures	[Icons]	0	InnoDB	utf8mb4_general_ci	16.0 КиБ	-
2 таблицы	Всего	0	InnoDB	utf8mb4_general_ci	48.0 КиБ	0 Байт

Создание связей между таблицами

Посмотреть созданную связь можно через «Еще» -> «Дизайнер»



Добавление и редактирование данных

Информацию в таблицу лучше всего заносить через административную часть сайта, которую предстоит еще написать. Но пока она не создана, предварительные данные для отладки программы можно добавлять и через phpMyAdmin. Для этого выберите необходимую таблицу (в нашем случае lectures) и нажмите на закладку «Вставить».

The screenshot shows the phpMyAdmin interface for inserting data into the 'lectures' table. The left sidebar displays a database structure tree with 'my_site' expanded, showing 'labworks' and 'lectures'. The main panel is titled 'Вставить' (Insert) and shows the table structure with columns: 'lecture_id' (int(10) unsigned), 'topic' (varchar(255)), 'name' (varchar(255)), and 'content' (text). Each column has a corresponding input field for the value to be inserted. The 'lecture_id' field is a small text box, while the others are larger text areas. The interface includes navigation tabs at the top: Обзор, Структура, SQL, Поиск, Вставить, Экспорт, Импорт, Привилегии, Операции, and Триггеры. A 'Вперед' (Next) button is located at the bottom right.

Столбец	Тип	Функция	Null	Значение
lecture_id	int(10) unsigned			<input type="text"/>
topic	varchar(255)			<input type="text"/>
name	varchar(255)			<input type="text"/>
content	text			<input type="text"/>

Добавление и редактирование данных

Заполняем поля данными. Поля, помеченные как `auto_increment` (у нас это поле `lecture_id`), заполнять не надо, записи в них будут созданы автоматически.

The screenshot shows the phpMyAdmin interface for the 'lectures' table. The left sidebar displays the database structure, with 'my_site' selected and 'lectures' highlighted. The main panel shows the table structure and a form for adding or editing records.

Столбец	Тип	Функция	Null	Значение
lecture_id	int(10) unsigned			
topic	varchar(255)			Введение в PHP
name	varchar(255)			PHP - это широко используемый язык сценариев общего назначения с открытым исходным кодом.
content	text			PHP - это язык программирования, специально разработанный для написания web-приложений (сценариев), исполняющихся на Web-сервере.

Ввод данных:

- lecture_id:
- topic:
- name:
- content:

Вперед

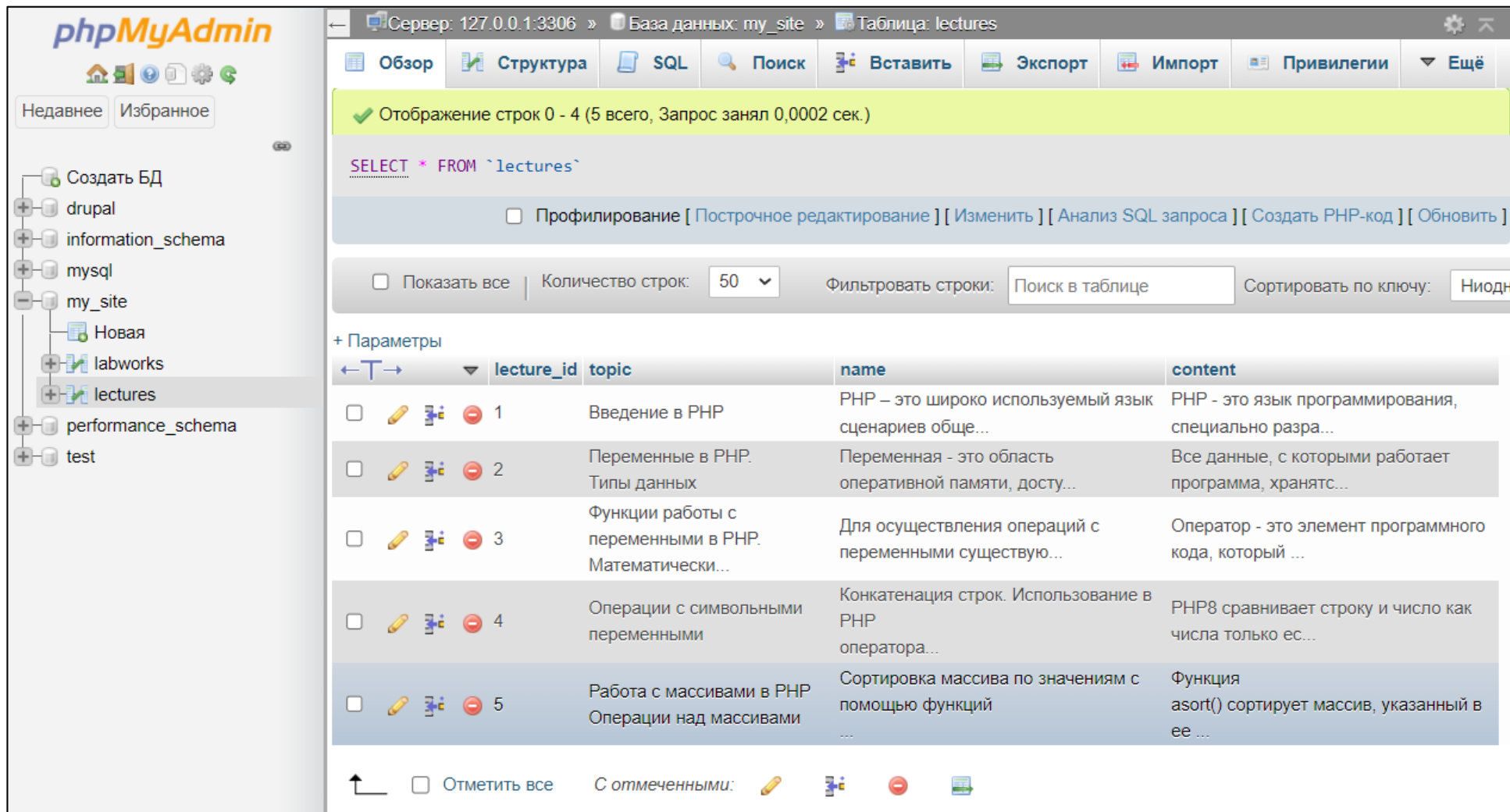
Добавление и редактирование данных

Добавлять данные можно также через SQL-запрос, например:

```
INSERT INTO `lectures` (`lecture_id`, `topic`, `name`, `content`) VALUES (NULL, 'Введение в PHP', 'PHP – это широко используемый язык\r\nсценариев общего назначения с открытым\r\nисходным кодом.', 'PHP – это язык программирования,\r\nспециально разработанный для написания\r\nweb-приложений (сценариев),\r\nисполняющихся на Web-сервере.');
```

Добавление и редактирование данных

Просмотреть существующие в таблице записи можно через закладку «Обзор». Там же можно при необходимости удалить их или отредактировать.



phpMyAdmin

Сервер: 127.0.0.1:3306 » База данных: my_site » Таблица: lectures

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Ещё
















Отображение строк 0 - 4 (5 всего, Запрос занял 0,0002 сек.)






`SELECT * FROM `lectures``

☐ Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

☐ Показать все | Количество строк: 50 | Фильтровать строки: Поиск в таблице | Сортировать по ключу: Ниодн

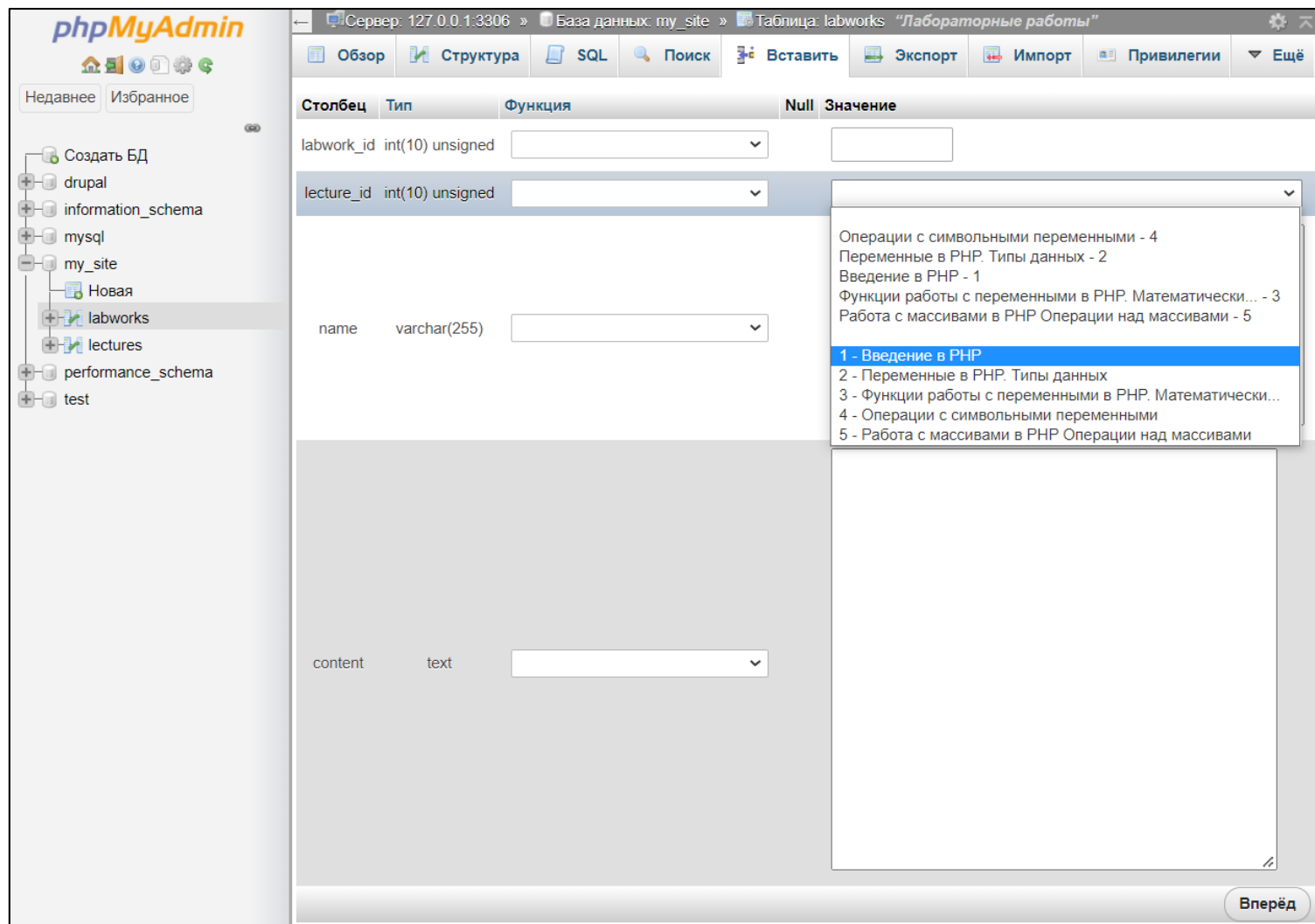
+ Параметры

	lecture_id	topic	name	content
<input type="checkbox"/>   	1	Введение в PHP	PHP – это широко используемый язык сценариев обще...	PHP - это язык программирования, специально разра...
<input type="checkbox"/>   	2	Переменные в PHP. Типы данных	Переменная - это область оперативной памяти, досту...	Все данные, с которыми работает программа, хранятс...
<input type="checkbox"/>   	3	Функции работы с переменными в PHP. Математически...	Для осуществления операций с переменными существу...	Оператор - это элемент программного кода, который ...
<input type="checkbox"/>   	4	Операции с символьными переменными	Конкатенация строк. Использование в PHP оператора...	PHP8 сравнивает строку и число как числа только ес...
<input type="checkbox"/>   	5	Работа с массивами в PHP. Операции над массивами	Сортировка массива по значениям с помощью функций ...	Функция <code>asort()</code> сортирует массив, указанный в ее ...

 ☐ Отметить все С отмеченными:    

Добавление и редактирование данных

Далее заполнить таблицу *labworks*. Так как у нас есть связь, то в поле *lecture_id* можно выбрать только те лекции, которые есть в таблице *lectures*.



Подключение к MySQL через PHP

Получение информации через БД происходит в несколько этапов.

- Посетитель запрашивает веб-страницу, указывая в браузере ее адрес (URL).
- Веб-сервер (Apache в нашем случае) определяет, что запрашивается PHP-файл и запускает его интерпретатор.
- Скрипт PHP обращается к MySQL и запрашивает требуемую информацию.
- База данных MySQL возвращает результат запроса обратно в программу PHP.
- Скрипт анализирует полученную информацию и сохраняет ее в одной или нескольких переменных. Затем текст выводится с помощью функции echo.
- Окончательно сформированный программой код HTML отправляется на веб-сервер, где он пересылается посетителю.

Подключение к MySQL через PHP

Для подключения к существующей БД используется функция *mysqli_connect ()*. Ее синтаксис в общем случае выглядит следующим образом.

```
mysqli_connect (адрес, имя пользователя, пароль, имя_БД)
```

Адрес — это IP-адрес или имя хоста компьютера, где запущена MySQL (например, localhost для локальной БД). Имя пользователя, пароль и непосредственно имя_БД необходимы для подключения к определенной базе данных.

Подключение к MySQL через PHP

Пример подключения:

```
<?php
    $host = "localhost";           // адрес сервера
    $database = "my_site";         // имя базы данных
    $user = "root";                // имя пользователя
    $password = "";               // пароль
    $link = mysqli_connect($host, $user, $password, $database)
or die("Ошибка ".mysqli_error($link));
?>
```

Пароль к локальной базе можно не указывать, поэтому он опущен. Отслеживать возникновение ошибки можно разными способами, но везде используется функция **die** – как и функция **echo**, предназначена для вывода текста, но после ее применения работа программы прерывается.

Подключение к MySQL через PHP

Чтобы пресечь возможные проблемы с кодировкой при отображении на страницах информации, рекомендуется в коде принудительно установить используемый набор символов. Пример установки набора символов *utf8mb4_general_ci* и обработка возможной ошибки представлена на следующем листинге кода.

```
if (!mysqli_set_charset($link, "utf8mb4")) {  
    echo "Ошибка при загрузке набора символов utf8mb4 ";  
    mysqli_error($link);  
    exit();  
}
```

В таблице приведены названия charset и соответствующие им кодировки, которые могут использоваться.

Charset	Description	Default collation	Maxlen
big5	Big5 Traditional Chinese	big5_chinese_ci	2
dec8	DEC West European	dec8_swedish_ci	1
cp850	DOS West European	cp850_general_ci	1
hp8	HP West European	hp8_english_ci	1
koi8r	KOI8-R Relcom Russian	koi8r_general_ci	1
latin1	cp1252 West European	latin1_swedish_ci	1
latin2	ISO 8859-2 Central European	latin2_general_ci	1
swe7	7bit Swedish	swe7_swedish_ci	1
ascii	US ASCII	ascii_general_ci	1
ujis	EUC-JP Japanese	ujis_japanese_ci	3
sjis	Shift-JIS Japanese	sjis_japanese_ci	2
hebrew	ISO 8859-8 Hebrew	hebrew_general_ci	1
tis620	TIS620 Thai	tis620_thai_ci	1
euckr	EUC-KR Korean	euckr_korean_ci	2
koi8u	KOI8-U Ukrainian	koi8u_general_ci	1
gb2312	GB2312 Simplified Chinese	gb2312_chinese_ci	2
greek	ISO 8859-7 Greek	greek_general_ci	1
cp1250	Windows Central European	cp1250_general_ci	1
gbk	GBK Simplified Chinese	gbk_chinese_ci	2
latin5	ISO 8859-9 Turkish	latin5_turkish_ci	1
armscii8	ARMSCII-8 Armenian	armscii8_general_ci	1
utf8	UTF-8 Unicode	utf8_general_ci	3
ucs2	UCS-2 Unicode	ucs2_general_ci	2
cp866	DOS Russian	cp866_general_ci	1
keybcs2	DOS Kamenicky Czech-Slovak	keybcs2_general_ci	1
macce	Mac Central European	macce_general_ci	1
macroman	Mac West European	macroman_general_ci	1
cp852	DOS Central European	cp852_general_ci	1
latin7	ISO 8859-13 Baltic	latin7_general_ci	1
utf8mb4	UTF-8 Unicode	utf8mb4_general_ci	4
cp1251	Windows Cyrillic	cp1251_general_ci	1
utf16	UTF-16 Unicode	utf16_general_ci	4
utf16le	UTF-16LE Unicode	utf16le_general_ci	4
cp1256	Windows Arabic	cp1256_general_ci	1
cp1257	Windows Baltic	cp1257_general_ci	1
utf32	UTF-32 Unicode	utf32_general_ci	4
binary	Binary pseudo charset	binary	1
geostd8	GEOSTD8 Georgian	geostd8_general_ci	1
cp932	SJIS for Windows Japanese	cp932_japanese_ci	2
eucjpms	UJIS for Windows Japanese	eucjpms_japanese_ci	3
gb18030	China National Standard GB18030	gb18030_chinese_ci	4

Подключение к MySQL через PHP

Для выборки данных из таблиц БД будет использоваться язык SQL. На PHP это будет выглядеть следующим образом.

```
$query = "SELECT * FROM lectures";  
$result = mysqli_query($link, $query) or  
die("Ошибка " . mysqli_error($link));
```

Данный код записывает в переменную **\$result** всю информацию из таблицы **lectures**. Отметим, что переменная **\$result** будет типа **object**. С ней будем работать с помощью циклов.

Подключение к MySQL через PHP

Количество записей определим с помощью функции *mysqli_num_rows(имя переменной)* , количество же столбцов будет в данном случае равно количеству полей в таблице *lectures*. Условно информация применительно к используемой нами БД в переменной **\$result** будет иметь следующий табличный вид:

1	Наименование раздела 1	Наименование лекции 1	Текст лекции 1
2	Наименование раздела 2	Наименование лекции 2	Текст лекции 2
3	Наименование раздела 3	Наименование лекции 3	Текст лекции 3
4	Наименование раздела 4	Наименование лекции 4	Текст лекции 4
5	Наименование раздела 5	Наименование лекции 5	Текст лекции 5
6	Наименование раздела 6	Наименование лекции 6	Текст лекции 6

Подключение к MySQL через PHP

Получаем данные из БД и выводим

```
if($result) // проверяем, получены ли данные из БД
{
    $rows = mysqli_num_rows($result);
    // количество полученных строк (записей)
    for ($i = 0 ; $i < $rows ; ++$i)
    {
        $row = mysqli_fetch_row($result);
        for ($j = 0 ; $j < 3 ; ++$j)
        {
            echo $row[$j] . "<br>";
        }
        echo "<br>";
    }
}
```

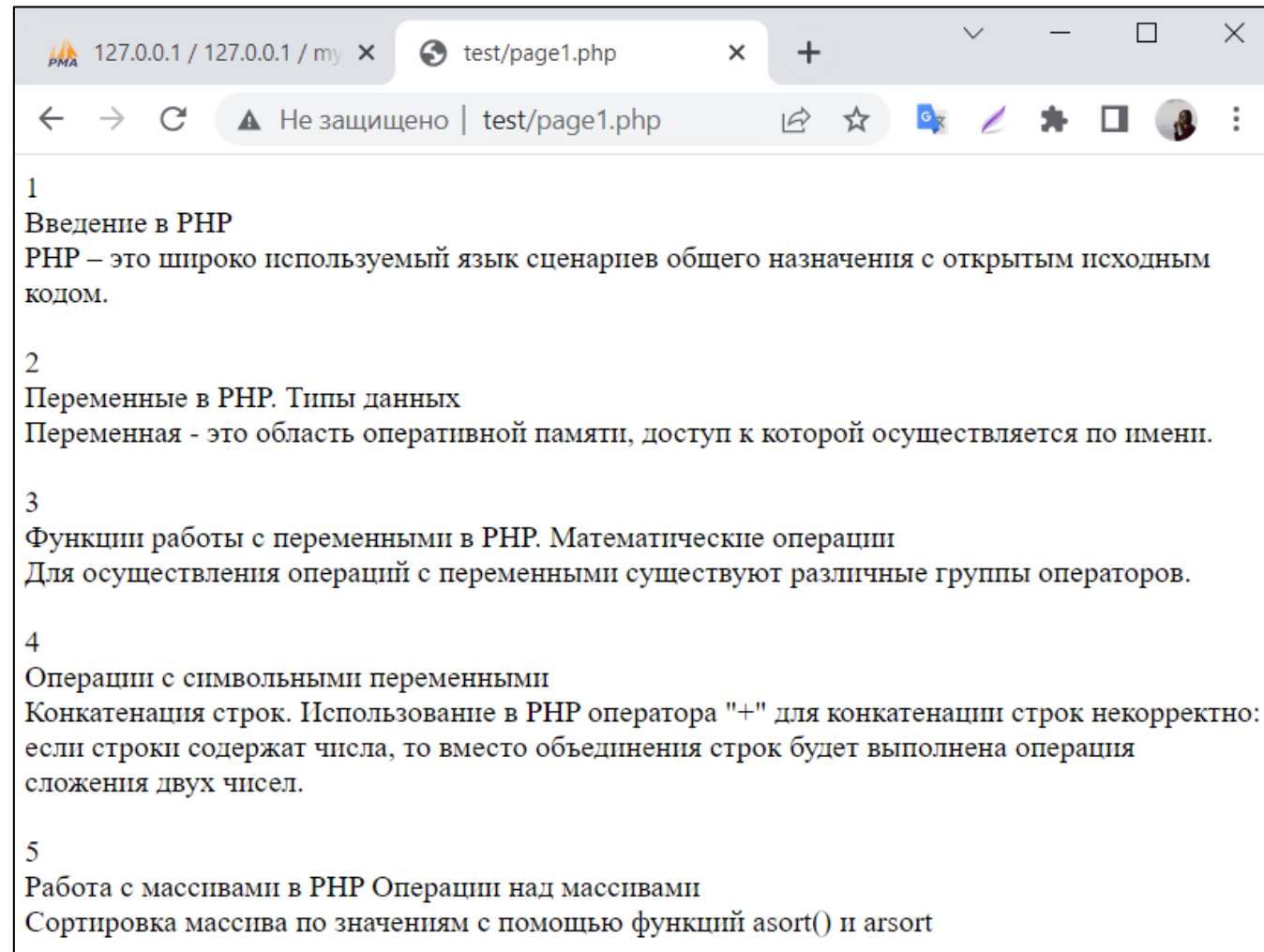
Подключение к MySQL через PHP

Можно получать данные из БД в виде ассоциативного массива

```
if ($result)
{
    $rows = mysqli_num_rows($result);
    // количество полученных строк (записей)
    for ($i = 0 ; $i < $rows ; ++$i)
    {
        $row = mysqli_fetch_assoc($result);
        echo $row['lecture_id'].".    ".$row['topic'].".
".$row['name'];
        echo "<br>";
    }
}
```

Подключение к MySQL через PHP

Результат:



Запись данных в БД

Создадим форму для добавления данных в таблицу *lectures* (файл primer.html)

```
<html>
  <head>
    <meta charset="utf-8">
    <title>Добавление данных в БД</title>
  </head>
  <body>
    <form action="page1.php" method="POST">Заполняем поля для передачи информации:
    <br><br>
    Тема раздела
    <input name="topic" type="text" value="" class="type-1"/>
    <br><br>
    Название лекции
    <input name="name" type="text" value="" class="type-1"/>
    <br><br>
    Текст лекции
    <textarea rows="10" cols="45" name="content" class="type-1"></textarea>
    <br><br>
    <input type="submit" value="Добавить лекцию"></form>
  </body>
</html>
```

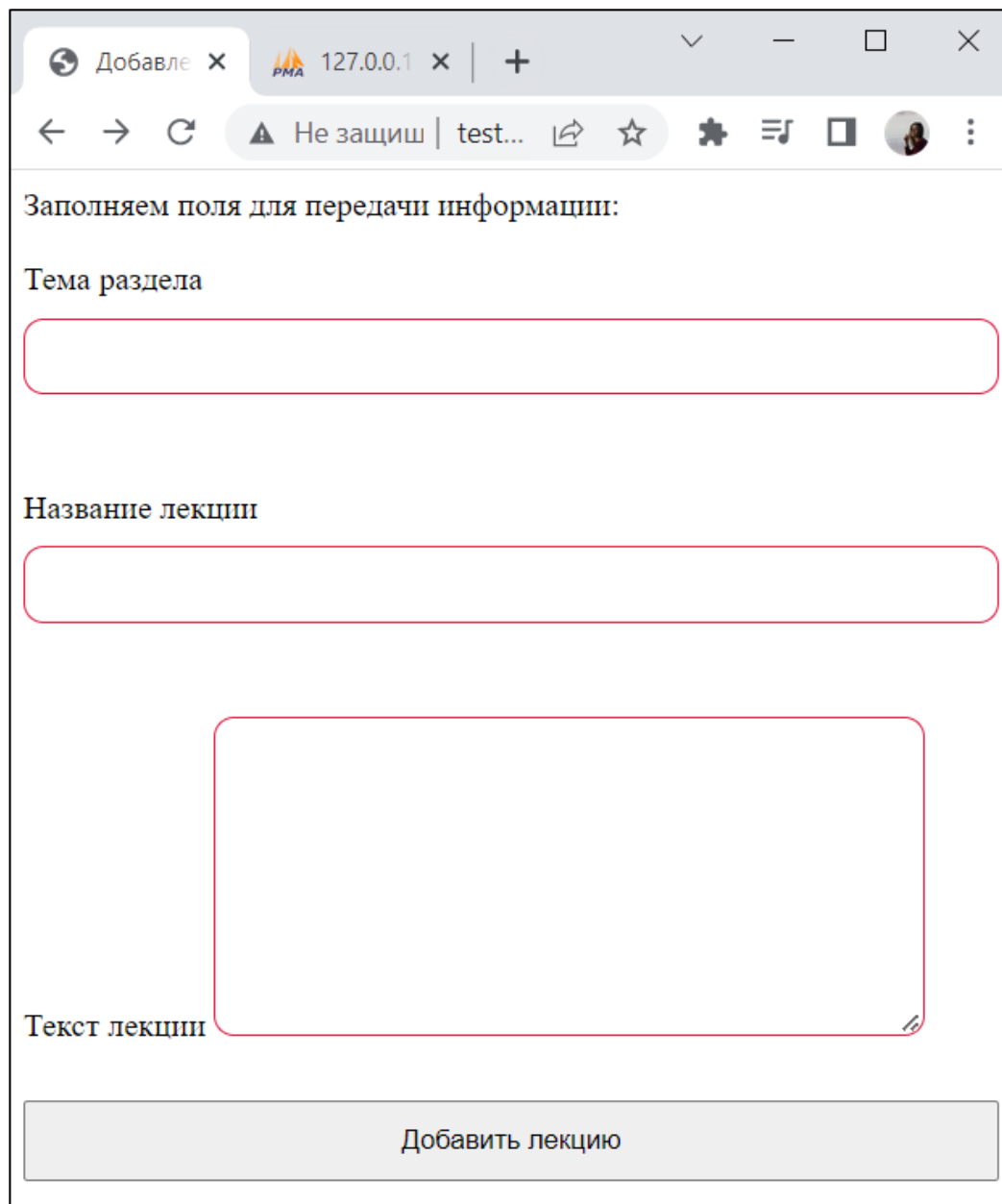
Запись данных в БД

Добавим стили

```
<style>
input {
  display: block;
  width: 100%;
  margin: 10px 0;
  padding: 10px;
}
.type-1 {
  border-radius: 10px;
  border: 1px solid #DC143C;
  transition: .3s border-color;
}
.type-1:hover {
  border: 1px solid #aaa;
}
</style>
```

Запись данных в БД

Результат



The screenshot shows a web browser window with the following elements:

- Browser Tabs:** One tab is open with the title "Добавле" (Add) and a small icon.
- Address Bar:** Displays "127.0.0.1" and a warning icon with the text "Не защищ" (Not protected).
- Form Title:** "Заполняем поля для передачи информации:" (Fill in the fields for information transfer).
- Form Fields:**
 - Тема раздела** (Section topic): A single-line text input field.
 - Название лекции** (Lecture title): A single-line text input field.
 - Текст лекции** (Lecture text): A multi-line text area with a small icon in the bottom right corner.
- Submit Button:** A wide button at the bottom labeled "Добавить лекцию" (Add lecture).

Запись данных в БД

Получаем данные из формы

```
$topic = $_POST['topic'];  
$name = $_POST['name'];  
$content = $_POST['content'];
```

Формируем запрос

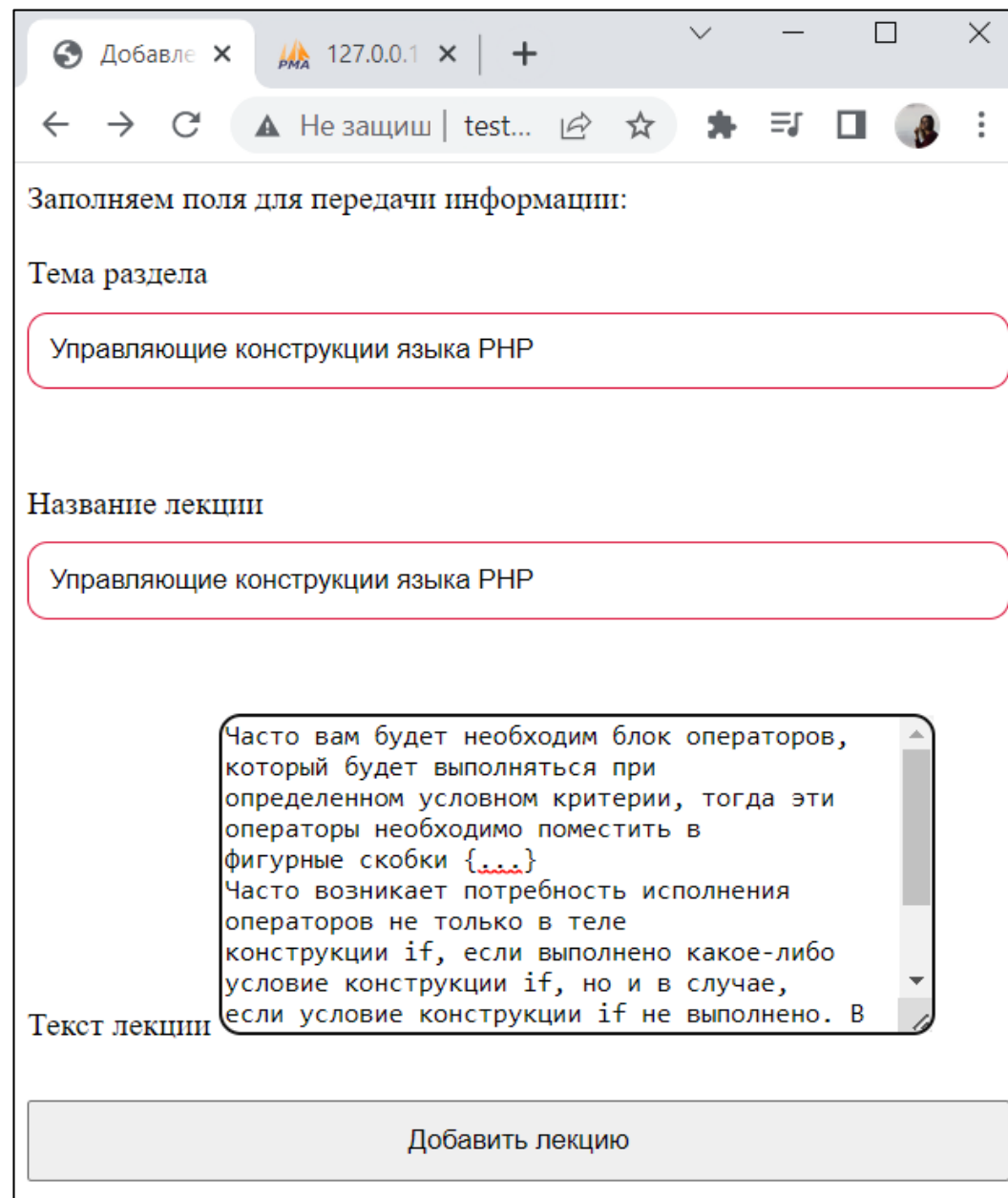
```
$query="insert into `lectures` (`lecture_id`, `topic`, `name`,  
`content`) values (null, '$topic', '$name', '$content')";
```

С помощью *mysqli_query()* выполняем запрос к базе данных. Возвращает false в случае возникновения ошибки. В случае успешного выполнения запросов, которые создают набор результатов, таких как SELECT, SHOW, DESCRIBE или EXPLAIN, *mysqli_query()* вернёт объект *mysqli_result*. Для остальных успешных запросов *mysqli_query()* вернёт true.

```
$result = mysqli_query($link, $query);
```

Запись данных в БД

Заполняем поля формы
и нажимаем «Добавить лекцию»



Добавле x 127.0.0.1 x +

← → ↻ Не защищ | test... ☆ ⚙ ⌵ 👤 ⋮

Заполняем поля для передачи информации:

Тема раздела

Управляющие конструкции языка PHP

Название лекции

Управляющие конструкции языка PHP

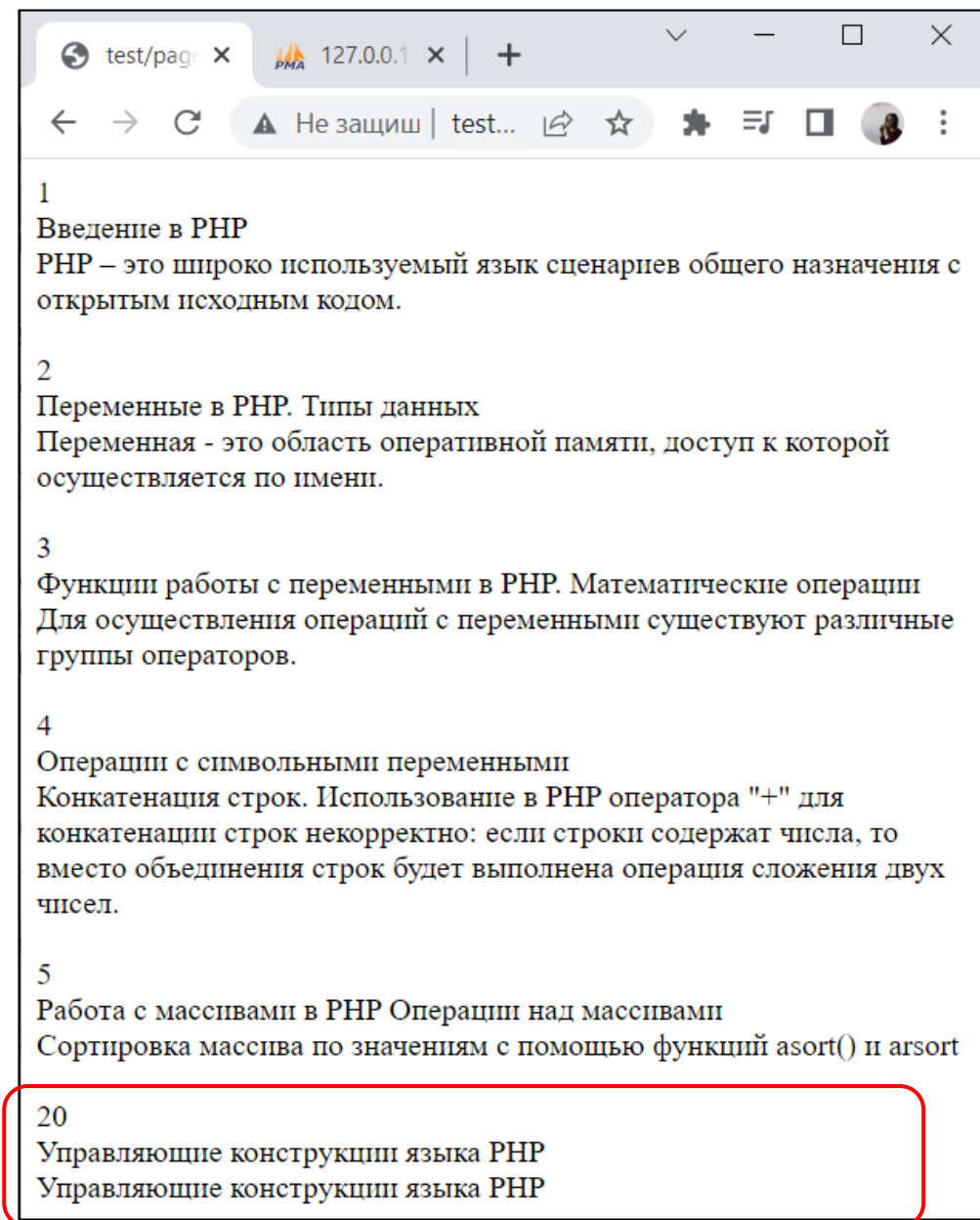
Текст лекции

Часто вам будет необходим блок операторов, который будет выполняться при определенном условном критерии, тогда эти операторы необходимо поместить в фигурные скобки {...}.
Часто возникает потребность исполнения операторов не только в теле конструкции if, если выполнено какое-либо условие конструкции if, но и в случае, если условие конструкции if не выполнено. В

Добавить лекцию

Запись данных в БД

Заполняем поля формы
и нажимаем «Добавить лекцию»



Типы данных в MySQL

MySQL предоставляет следующие типы данных, которые можно разбить на ряд групп.

Символьные типы

CHAR: представляет строку фиксированной длины.

Длина хранимой строки указывается в скобках, например, CHAR(10) - строка из десяти символов. И если в таблицу в данный столбец сохраняется строка из 6 символов (то есть меньше установленной длины в 10 символов), то строка дополняется 4 пробелами и в итоге все равно будет занимать 10 символов

VARCHAR: представляет строку переменной длины.

Длина хранимой строки также указывается в скобках, например, VARCHAR(10). Однако в отличие от CHAR, хранимая строка будет занимать именно столько места, сколько необходимо. Например, если определенная длина в 10 символов, но в столбец сохраняется строка в 6 символов, то хранимая строка так и будет занимать 6 символов плюс дополнительный байт, который хранит длину строки.

Типы данных в MySQL

MySQL предоставляет следующие типы данных, которые можно разбить на ряд групп.

Символьные типы

Начиная с MySQL 5.6, типы CHAR и VARCHAR по умолчанию используют кодировку UTF-8, которая позволяет использовать до 3 байт для хранения символа в зависимости от языка (для многих европейских языков по 1 байту на символ, для ряда восточно-европейских и ближневосточных - 2 байта, а для китайского, японского, корейского - по 3 байта на символ).

Ряд дополнительных типов данных представляют текст неопределенной длины:

TINYTEXT: представляет текст длиной до 255 байт.

TEXT: представляет текст длиной до 65 КБайт.

MEDIUMTEXT: представляет текст длиной до 16 Мбайт

LARGETEXT: представляет текст длиной до 4 ГБайт

Типы данных в MySQL

MySQL предоставляет следующие типы данных, которые можно разбить на ряд групп.

Числовые типы

TINYINT: представляет целые числа от -127 до 128, занимает 1 байт

BOOL: фактически не представляет отдельный тип, а является лишь псевдонимом для типа **TINYINT(1)** и может хранить два значения 0 и 1. Однако данный тип может также в качестве значения принимать встроенные константы **TRUE** (представляет число 1) и **FALSE** (предоставляет число 0).

Также имеет псевдоним **BOOLEAN**.

TINYINT UNSIGNED: представляет целые числа от 0 до 255, занимает 1 байт

SMALLINT: представляет целые числа от -32768 до 32767, занимает 2 байта

SMALLINT UNSIGNED: представляет целые числа от 0 до 65535, занимает 2 байта

Типы данных в MySQL

MySQL предоставляет следующие типы данных, которые можно разбить на ряд групп.

MEDIUMINT: представляет целые числа от -8388608 до 8388607, занимает 3 байта

MEDIUMINT UNSIGNED: представляет целые числа от 0 до 16777215, занимает 3 байта

INT: представляет целые числа от -2147483648 до 2147483647, занимает 4 байта

INT UNSIGNED: представляет целые числа от 0 до 4294967295, занимает 4 байта

BIGINT: представляет целые числа от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807, занимает 8 байт

BIGINT UNSIGNED: представляет целые числа от 0 до 18 446 744 073 709 551 615, занимает 8 байт

DECIMAL: хранит числа с фиксированной точностью. Данный тип может принимать два параметра precision и scale: DECIMAL(precision, scale).

Типы данных в MySQL

MySQL предоставляет следующие типы данных, которые можно разбить на ряд групп.

Типы для работы с датой и временем

DATE: хранит даты с 1 января 1000 года до 31 декабря 9999 года (с "1000-01-01" до "9999-12-31"). По умолчанию для хранения используется формат `yyyy-mm-dd`. Занимает 3 байта.

TIME: хранит время от -838:59:59 до 838:59:59. По умолчанию для хранения времени применяется формат `"hh:mm:ss"`. Занимает 3 байта.

DATETIME: объединяет время и дату, диапазон дат и времени - с 1 января 1000 года по 31 декабря 9999 года (с "1000-01-01 00:00:00" до "9999-12-31 23:59:59"). Для хранения по умолчанию используется формат `"yyyy-mm-dd hh:mm:ss"`. Занимает 8 байт

TIMESTAMP: также хранит дату и время, но в другом диапазоне: от "1970-01-01 00:00:01" UTC до "2038-01-19 03:14:07" UTC. Занимает 4 байта

YEAR: хранит год в виде 4 цифр. Диапазон доступных значений от 1901 до 2155. Занимает 1 байт.

Типы данных в MySQL

MySQL предоставляет следующие типы данных, которые можно разбить на ряд групп.

Размер данных в байтах для DECIMAL зависит от хранимого значения.

Данный тип также имеет псевдонимы NUMERIC, DEC, FIXED.

- **FLOAT**: хранит дробные числа с плавающей точкой одинарной точности от $-3.4028 * 10^{38}$ до $3.4028 * 10^{38}$, занимает 4 байта

Может принимать форму `FLOAT(M,D)`, где M - общее количество цифр, а D - количество цифр после запятой.

- **DOUBLE**: хранит дробные числа с плавающей точкой двойной точности от $-1.7976 * 10^{308}$ до $1.7976 * 10^{308}$, занимает 8 байт. Также может принимать форму `DOUBLE(M,D)`, где M - общее количество цифр, а D - количество цифр после запятой.

Типы данных в MySQL

MySQL предоставляет следующие типы данных, которые можно разбить на ряд групп.

Тип **DATE** может принимать даты в различных форматах, однако непосредственно для хранения в самой бд даты приводятся к формату "yyyy-mm-dd". Некоторые из принимаемых форматов:

- yyyy-mm-dd - 2018-05-25
- yyyy-m-dd - 2018-5-25
- yy-m-dd - 18-05-25

В таком формате двузначные числа от 00 до 69 воспринимаются как даты в диапазоне 2000-2069. А числа от 70 до 99 как диапазон чисел 1970 - 1999.

- yyyymmdd - 20180525
- yyyy.mm.dd - 2018.05.25

Типы данных в MySQL

MySQL предоставляет следующие типы данных, которые можно разбить на ряд групп.

Для времени тип **TIME** использует 24-часовой формат. Он может принимать время в различных форматах:

- hh:mi - 3:21 (хранимое значение 03:21:00)
- hh:mi:ss - 19:21:34
- hhmiss - 192134

Примеры значений для типов **DATETIME** и **TIMESTAMP**:

- 2018-05-25 19:21:34
- 2018-05-25 (хранимое значение 2018-05-25 00:00:00)

Типы данных в MySQL

MySQL предоставляет следующие типы данных, которые можно разбить на ряд групп.

Составные типы

- **ENUM**: хранит одно значение из списка допустимых значений. Занимает 1-2 байта
- **SET**: может хранить несколько значений (до 64 значений) из некоторого списка допустимых значений. Занимает 1-8 байт.

Бинарные типы

- **TINYBLOB**: хранит бинарные данные в виде строки длиной до 255 байт.
- **BLOB**: хранит бинарные данные в виде строки длиной до 65 КБ.
- **MEDIUMBLOB**: хранит бинарные данные в виде строки длиной до 16 МБ
- **LARGEBLOB**: хранит бинарные данные в виде строки длиной до 4 ГБ