

**РОСЖЕЛДОР**

**Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
"Ростовский государственный университет путей сообщения"  
(ФГБОУ ВО РГУПС)**

---

УТВЕРЖДАЮ  
Первый проректор:  
М.А. Кравченко

Кафедра "Вычислительная техника и автоматизированные системы управления"

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ  
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ (МЕЖДИСЦИПЛИНАРНОМУ КУРСУ)**

**МДК.04.02 Разработка серверных приложений с базами данных**

**по Учебному плану**

специальности среднего профессионального образования  
09.02.09 Веб-разработка

Квалификация специалиста среднего звена "Разработчик веб-приложений"

Ростов-на-Дону  
2025



## Содержание

1. Результаты обучения дисциплины (модуля).....	4
2. Компетенции, формируемые в процессе изучения дисциплины (модуля).....	5
3. Курсовая работа.....	7
4. Оценочные средства для оценки успеваемости студентов.....	10
5. Шкала оценивания в зависимости от уровня сформированности компетенций .....	14

## 1. Результаты обучения дисциплины (модуля)

В результате освоения дисциплины (модуля) обучающийся должен овладеть следующими результатами:

Код и наименование компетенции выпускника	Формулировка требований к степени сформированности компетенции
ПК 4.1 Администрировать среды и платформы разработки информационных ресурсов	<b>Умения:</b>
	Устанавливать, настраивать и управлять локальной средой разработки (OpenServer) для отладки PHP-приложений с БД MySQL;
	Создавать базы данных, проектировать и формировать таблицы с учетом типов данных, первичных и внешних ключей через SQL и phpMyAdmin;
	Выполнять базовые операции администрирования БД: создание резервных копий (дампов) с помощью mysqldump или phpMyAdmin и их восстановление;
	Настраивать подключение PHP-скриптов к базе данных через расширение PDO.
	<b>Знания:</b>
	Архитектуру локального серверного окружения (веб-сервер, интерпретатор PHP, СУБД MySQL, система администрирования phpMyAdmin) в рамках сборки Denwer;
	Основные объекты базы данных (БД, таблицы, столбцы, индексы, связи) и принципы их создания;
ПК 4.2 Создавать программный код на стороне сервера в соответствии с техническим заданием (спецификацией) с использованием языков программирования,	Методы управления структурой базы данных с использованием языка DDL (CREATE, ALTER, DROP) и графического интерфейса phpMyAdmin;
	Принципы организации и хранения проектных файлов, настройки виртуальных хостов.
	<b>Умения:</b>
	«Читать» и интерпретировать техническое задание в виде структуры базы данных и набора требуемых функций сайта (формы, списки, личные кабинеты);
	Проектировать реляционную структуру базы данных для решения поставленной задачи (выделять сущности, определять типы данных и связи);
	Писать SQL-запросы для выборки, вставки, обновления и удаления данных, в том числе с использованием объединений таблиц (JOIN), условий (WHERE),

библиотек и фреймворков	сортировки (ORDER BY) и агрегатных функций
	<b>Знания:</b>
	<p>Синтаксис и базовые конструкции языка PHP для серверной разработки;</p> <p>Архитектурный шаблон взаимодействия «клиент (браузер) – сервер (PHP) – база данных (MySQL)»;</p> <p>Язык структурированных запросов (SQL) для операций манипуляции данными (DML): SELECT, INSERT, UPDATE, DELETE;</p> <p>Принципы построения защищенных веб-приложений: защита от SQL-инъекций (подготовленные выражения PDO), валидация и экранирование пользовательского ввода;</p> <p>Основы реализации бизнес-логики типовых веб-модулей: аутентификации (хеширование паролей), CRUD-операций, пагинации, простых отчетов</p>
ПК 4.3 Осуществлять отладку программного кода на стороне сервера на уровне программных модулей, межмодульных взаимодействий и взаимодействий с окружением	<b>Умения:</b>
	<p>Включать и анализировать вывод ошибок PHP и MySQL для локализации проблем;</p> <p>Применять метод «отладки выводом» для проверки значений переменных, массивов \$_GET/\$_POST и результатов SQL-запросов на промежуточных этапах работы скрипта;</p> <p>Проверять корректность и эффективность SQL-запросов с помощью EXPLAIN;</p> <p>Использовать транзакции для обеспечения целостности данных при отладке комплексных операций (например, оформление заказа).</p>
	<b>Знания:</b> <p>Классификацию ошибок в веб-разработке: синтаксические ошибки PHP, ошибки выполнения (исключения), логические ошибки в SQL, ошибки подключения к БД;</p> <p>Методы и инструменты диагностики: встроенные сообщения об ошибках PHP и MySQL, использование var_dump()/print_r() для вывода отладочной информации, анализ журналов ошибок;</p> <p>Принципы использования оператора EXPLAIN для анализа производительности и корректности SQL-запросов;</p> <p>Методы диагностики проблем взаимодействия: проверка подключения к БД, анализ передаваемых параметров (ID, данные форм), проверка состояния сессий.</p>

## 2. Компетенции, формируемые в процессе изучения дисциплины (модуля)

Индекс и Наименование компетенции	Признаки проявления компетенции в соответствии с уровнем формирования в процессе освоения дисциплины
<p>ПК 4.1 Администрировать среды и платформы разработки информационных ресурсов</p> <p>ПК 4.2 Создавать программный код на стороне сервера в соответствии с техническим заданием (спецификацией) с использованием языков программирования, библиотек и фреймворков</p> <p>ПК 4.3 Осуществлять отладку программного кода на стороне сервера на уровне программных модулей, межмодульных взаимодействий и взаимодействий с окружением</p>	<p><b>недостаточный</b> уровень: Компетенции не сформированы. Знания отсутствуют, умения и навыки не сформированы.</p> <p><b>пороговый</b> уровень: Компетенции сформированы. Сформированы базовые структуры знаний. Умения фрагментарны и носят репродуктивный характер. Демонстрируется низкий уровень самостоятельности практического навыка.</p> <p><b>продвинутый</b> уровень: Компетенции сформированы. Знания обширные, системные. Умения носят репродуктивный характер, применяются к решению типовых заданий. Демонстрируется достаточный уровень самостоятельности устойчивого практического навыка.</p> <p><b>высокий</b> уровень: Компетенции сформированы. Знания твердые, аргументированные, всесторонние. Умения успешно применяются к решению как типовых, так и нестандартных творческих заданий. Демонстрируется высокий уровень самостоятельности, высокая адаптивность практического навыка.</p>

### 3. Курсовая работа

Курсовая работа на тему: «Разработка информационной системы управления контентом (CMS) для сайта-портфолио или блога с разграничением прав доступа (по вариантам).

Варианты:

Тема 1. Разработка системы личного кабинета пользователя

Суть: Создание модуля с регистрацией, авторизацией, профилем, сменой пароля и восстановлением доступа. Внутри кабинета — история действий/заказов, настройки.

Стек: PHP, MySQL, сессии, безопасность (хэши паролей, SQL-инъекции, CSRF-токены), HTML/CSS.

Что оценивается: Полноценный цикл работы с пользователем, безопасность, чистота кода

Тема 2. Система управления контентом (CMS) для простого сайта-визитки

Суть: Мини-CMS для управления страницами «О компании», «Услуги», «Контакты» и блогом (статьи). Админ-панель для редактирования текстов и загрузки изображений.

Стек: PHP, MySQL, CRUD, WYSIWYG-редактор (например, TinyMCE) или Markdown, базовое администрирование.

Что оценивается: Архитектура разделения админки и фронта, работа с файлами, простота использования.

Тема 3. Онлайн-каталог товаров с корзиной (без оплаты)

Суть: Каталог товаров с категориями, фильтрами и поиском. Функционал добавления в корзину, оформления «заказа» с сохранением в БД.

Стек: PHP, MySQL, сессии (для корзины), пагинация, фильтрация ('WHERE', 'LIKE', диапазоны цен).

Что оценивается: Работа со сложными SQL-запросами, организация сессий, юзабилити.

Тема 4. Система обратной связи и тикетов (служба поддержки)

Суть: Публичная форма обращения и личный кабинет, где пользователь видит статус своих обращений. Админ-панель для ответов и управления тикетами (открыт/в работе/закрыт).

Стек: PHP, MySQL, уведомления на email ('mail()') или PHPMailer), статусы, роли пользователей.

Что оценивается: Логика workflow, взаимодействие между клиентом и админом, отправка почты.

Тема 5. REST API для мобильного приложения «Список дел» (To-Do List)

Суть: Backend-часть, предоставляющая API для создания, чтения, обновления и удаления задач. Каждая задача привязана к пользователю.

Стек: PHP (без фронтенда, только `echo json_encode()`), MySQL, аутентификация по API-токену (в заголовке запроса), REST-маршрутизация.

Что оценивается: Чистота и документированность API, безопасность (токены), корректность HTTP-методов и кодов ответа.

#### Тема 6. Система голосования или опросов

Суть: Создание опросов с разными типами вопросов (один вариант, несколько). Защита от накруток (по IP или сессии). Визуализация результатов в виде диаграмм (можно с помощью библиотеки Chart.js).

Стек: PHP, MySQL, JavaScript (для отрисовки графиков), защита от дублирования голосов.

Что оценивается: Логика подсчета голосов, безопасность, интерактивное отображение данных.

#### Тема 7. Сервис сокращения ссылок (аналог TinyURL)

Суть: Пользователь вводит длинный URL, получает короткую уникальную ссылку. Переход по короткой ссылке делает редирект на исходный адрес. Статистика переходов.

Стек: PHP, MySQL, генерация короткого кода, редирект (`header('Location: ...')`), подсчет кликов.

Что оценивается: Алгоритм генерации коротких ссылок, обработка редиректов, эффективность.

#### Тема 8. Простой агрегатор новостей (RSS-ридер)

Суть: Админ добавляет RSS-ленты. Система периодически (или по запросу) парсит их, сохраняет новые статьи в БД и отображает в единой ленте с возможностью фильтрации по источнику.

Стек: PHP, MySQL, парсинг XML (`simplexml_load_file()`), планировщик (можно ручной запуск через крон или кнопку в админке).

Что оценивается: Работа с внешними данными (XML), структура БД для хранения статей, интерфейс управления источниками.

#### Тема 9. Система рейтинга и комментариев для статей

Суть: Модуль, который можно подключить к существующему блогу. Пользователи могут оставлять комментарии (с поддержкой ответов) и ставить оценку статье (звезды). Вывод среднего рейтинга.



Стек: PHP, MySQL, AJAX (для добавления комментариев/оценок без перезагрузки), вложенность комментариев.

Что оценивается: Организация древовидной структуры комментариев, расчет рейтинга, UX с использованием AJAX.

Тема 10. Планировщик задач (Task Scheduler) с уведомлениями

Суть: Веб-интерфейс для создания задач с дедлайном и описанием. Фоновый скрипт (CLI), который запускается по `cron` каждую минуту, проверяет БД на предстоящие задачи и отправляет уведомления (email или в систему).

Стек: PHP (CLI + Web), MySQL, планировщик cron, отправка email.

Что оценивается: Работа PHP в CLI-режиме, взаимодействие веб-интерфейса и фонового процесса, организация планирования.

#### 4. Оценочные средства для оценки успеваемости студентов

1

##### **!BK**

Расширение PHP, которое предоставляет легкий и последовательный интерфейс для доступа к различным СУБД (MySQL, PostgreSQL и др.) через подготовленные выражения – это ...

!

PDO (PHP Data Objects)

2

##### **!BK**

Язык, используемый для создания, изменения и управления структурой базы данных и её объектами (таблицы, индексы) – ...

!

DDL (Data Definition Language)

3

##### **!BV**

Основное назначение оператора **WHERE** в SQL-запросе – это ...

!+

задание условий для фильтрации строк, выбираемых из таблицы

!-

задание условий для сортировки строк, выбираемых из таблицы

!-

задание имени таблицы, из которой нужно выбрать данные

!\*-

объединение данных из нескольких таблиц

4

##### **!BV**

Что такое SQL-инъекция?

!+

Вид атаки на веб-приложение, при котором злоумышленник вставляет или «инжектирует» вредоносный SQL-код в параметры запроса

!-

Ошибка в синтаксисе SQL-запроса, вызывающая исключение

!-

Метод оптимизации SQL-запросов с помощью вложенных SELECT

!\*-

Автоматическое добавление данных в базу через форму

5

##### **!BV**

Какая команда SQL используется для выборки данных из таблицы?

!+

SELECT

!-

GET

!-

FIND

!\*-

REQUEST

6

**!BK**

Ключ в таблице, однозначно идентифицирующий каждую её запись – ...

!

Первичный ключ (PRIMARY KEY)

7

**!BV**

Для чего используется SQL-оператор JOIN?

!+

Для объединения строк из двух или более таблиц на основе связанного столбца между ними

!-

Для объединения нескольких баз данных в одну

!-

Для создания связи между таблицами на уровне схемы БД

!\*\_-

Для выборки уникальных значений из столбца

8

**!BV**

Что делает функция password\_hash() в PHP?

!+

Создает хеш пароля, используя стойкий односторонний алгоритм

!-

Проверяет соответствие введенного пароля хешу в базе данных

!-

Шифрует пароль симметричным алгоритмом

!\*\_-

Сохраняет пароль в открытом виде в специальной таблице

9

**!BK**

Принцип, гарантирующий, что транзакция будет выполнена либо полностью, либо не выполнена вовсе – ...

!

Атомарность (Atomicity)

10

**!BV**

Какой HTTP-метод обычно используется в веб-форме для передачи данных, которые изменяют состояние на сервере (например, добавление новой записи)?

!+

POST

!-

GET

!-

PUT

!\*\_-

HEAD

### **Перечень вопросов для устного опроса:**

1) Взаимодействие PHP и MySQL: принципы подключения к базе данных через PDO, выполнение запросов и обработка результатов.

- 2) Проектирование структуры базы данных: создание таблиц, выбор типов данных, определение первичных и внешних ключей.
- 3) Основы языка SQL для веб-разработки: операции SELECT, INSERT, UPDATE, DELETE; использование условий WHERE и сортировки ORDER BY.
- 4) Объединение таблиц (JOIN): назначение и использование INNER JOIN для получения связанных данных из нескольких таблиц.
- 5) Защита от SQL-инъекций: механизм работы подготовленных выражений (Prepared Statements) в PDO.

#### **Перечень вопросов для самоподготовки:**

- 1) Реляционные связи в базах данных: виды связей (один-к-одному, один-ко-многим, многие-ко-многим) и их реализация с помощью первичных и внешних ключей;
- 2) Оптимизация SQL-запросов: использование индексов, анализ плана выполнения запроса с помощью EXPLAIN;
- 3) Транзакции в MySQL: понятие ACID, использование BEGIN, COMMIT, ROLLBACK для обеспечения целостности данных;
- 4) Пагинация данных: принципы реализации постраничного вывода на уровне SQL (LIMIT, OFFSET) и на уровне приложения;
- 5) Нормализация баз данных: цели, аномалии данных, первая, вторая и третья нормальные формы (1NF, 2NF, 3NF);
- 6) Полнотекстовый поиск в MySQL: создание и использование FULLTEXT-индексов, операторы MATCH() и AGAINST();
- 7) Резервное копирование и восстановление БД: утилита mysqldump, создание и импорт дампа базы данных.

#### **Перечень контрольных вопросов к зачету:**

##### **Знать:**

- 1) Архитектура веб-приложения «клиент-сервер-БД» и роль СУБД MySQL;
- 2) Основные объекты базы данных: базы, таблицы, столбцы, первичный ключ;
- 3) Типы данных в MySQL: числовые, строковые, дата и время;
- 4) Язык определения данных (DDL): команды CREATE, ALTER, DROP;
- 5) Язык манипулирования данными (DML): команды INSERT, SELECT, UPDATE, DELETE;
- 6) Основы реляционных связей: связи «один-ко-многим», внешний ключ (FOREIGN KEY);
- 7) Объединение таблиц (JOIN): назначение и использование INNER JOIN;
- 8) Принципы подключения PHP к MySQL с использованием расширения PDO;
- 9) Методы защиты от SQL-инъекций: подготовленные выражения (Prepared Statements);
- 10) Основы аутентификации пользователей: хеширование паролей (password\_hash, password\_verify).

##### **Уметь:**

- 1) Установка и настройка локального серверного окружения (Denwer);
- 2) Создание базы данных и таблиц через интерфейс phpMyAdmin и SQL-запросы;
- 3) Написание SQL-запросов на выборку данных с условиями, сортировкой и ограничениями (WHERE, ORDER BY, LIMIT);

- 4) Проектирование и создание связанных таблиц с использованием первичных и внешних ключей;
- 5) Настройка подключения PHP-приложения к базе данных через PDO;
- 6) Выполнение SELECT-запросов из PHP и вывод результатов в виде HTML-страницы;
- 7) Обработка HTML-форм: получение данных, валидация, выполнение защищённого INSERT-запроса;
- 8) Реализация страницы детального просмотра записи на основе параметра из GET-запроса;
- 9) Создание страницы и формы для обновления данных (UPDATE) в базе данных;
- 10) Реализация базовой регистрации и аутентификации пользователя с хешированием пароля.

**Перечень контрольных вопросов к экзамену:**

**Знать:**

- 1) Нормализация баз данных: цели, аномалии данных, первая, вторая и третья нормальные формы;
- 2) Транзакции в СУБД: понятие, свойства ACID, команды управления;
- 3) Индексы в базах данных: назначение, типы, влияние на производительность;
- 4) Агрегатные функции и группировка данных в SQL (GROUP BY, COUNT, SUM, AVG);
- 5) Принципы реализации постраничного вывода данных (пагинации) в веб-приложениях;
- 6) Механизмы хранения состояния корзины покупок: сессии и база данных;
- 7) Организация древовидных структур в реляционной БД (метод parent\_id);
- 8) Принципы полнотекстового поиска в MySQL (FULLTEXT-индексы);
- 9) Основные уязвимости веб-приложений: XSS, CSRF и методы защиты;
- 10) Методы резервного копирования и восстановления баз данных MySQL.

**Уметь:**

- 1) Проектирование и создание нормализованной схемы базы данных для заданной предметной области;
- 2) Применение транзакций для обеспечения целостности данных при выполнении связанных операций;
- 3) Создание индексов для оптимизации запросов и анализ плана выполнения с помощью EXPLAIN;
- 4) Написание SQL-запросов с использованием агрегатных функций для формирования отчётов;
- 5) Реализация пагинации данных на стороне сервера (PHP + SQL);
- 6) Разработка модуля корзины покупок с использованием сессий и переносом данных в БД при оформлении заказа;
- 7) Создание иерархического (древовидного) меню категорий на основе рекурсивного алгоритма или цикла;
- 8) Настройка и использование полнотекстового поиска по содержимому таблиц;
- 9) Реализация административного раздела с проверкой прав доступа и защитой от CSRF;

10) Создание и восстановление резервной копии базы данных с помощью утилиты mysqldump.

## 5. Шкала оценивания в зависимости от уровня сформированности компетенций

### Описание шкал оценивания компетенций

Значение оценки	Уровень освоения компетенции	Шкала оценивания (для аттестационной ведомости, зачетной книжки, документа об образовании)	Шкала оценивания (процент верных при проведении тестирования)
Балльная оценка - "удовлетворительно".	Пороговый	Оценка « <b>удовлетворительно</b> » выставляется обучающемуся, который имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения последовательности изложения программного материала и испытывает трудности в выполнении практических навыков.	От 40% до 59%
Балльная оценка - "хорошо".	Базовый	Оценка « <b>хорошо</b> » выставляется обучающемуся, твердо знающему программный материал, грамотно и по существу его излагающему, который не допускает существенных неточностей в ответе, правильно применяет теоретические положения при решении практических работ и задач, владеет необходимыми навыками и приемами их выполнения.	От 60% до 84%
Балльная оценка - "отлично".	Высокий	Оценка « <b>отлично</b> » выставляется обучающемуся, глубоко и прочно усвоившему программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагающему, в ответе которого тесно увязываются теория с практикой. При этом обучающийся не затрудняется с ответом при	От 85% до 100%

Значение оценки	Уровень освоения компетенции	Шкала оценивания (для аттестационной ведомости, зачетной книжки, документа об образовании)	Шкала оценивания (процент верных при проведении тестирования)
		видоизменении задания, показывает знакомство с литературой, правильно обосновывает ответ, владеет разносторонними навыками и приемами практического выполнения практических работ.	
Дуальная оценка - "зачтено".	Пороговый, Базовый, Высокий	Оценка « <b>зачтено</b> » выставляется обучающемуся, который имеет знания, умения и навыки, не ниже знания только основного материала, может не освоить его детали, допускать неточности, недостаточно правильные формулировки, нарушения последовательности изложения программного материала и испытывает трудности в выполнении практических навыков.	От 40% до 100%
Балльная оценка - "неудовлетворительно", Дуальная оценка - "не зачтено".	Не достигнут	Оценка « <b>неудовлетворительно, не зачтено</b> » выставляется обучающемуся, который не знает значительной части программного материала, допускает ошибки, неуверенно выполняет или не выполняет практические работы.	От 0% до 39%