

РОСЖЕЛДОР
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Ростовский государственный университет путей сообщения»
(ФГБОУ ВО РГУПС)

Хусаинов В.Р.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ
СТУДЕНТОВ ПО ДИСЦИПЛИНЕ

МДК.02.01 «РАЗРАБОТКА СЕРВЕРНЫХ ПРИЛОЖЕНИЙ С БАЗАМИ ДАННЫХ»

для специальности
09.02.09 Веб-разработка

г. Ростов-на-Дону
2025 г.

СОДЕРЖАНИЕ

1 САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТОВ	3
2 СТРУКТУРА ДИСЦИПЛИНЫ.....	3
3 РАСПРЕДЕЛЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО РАЗДЕЛАМ.....	7
4 КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ ДЛЯ САМОПРОВЕРКИ.....	8
5 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ.....	10
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	14

1 САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТОВ

Самостоятельная работа студентов является важнейшей составной частью образовательного процесса и способом активного, целенаправленного приобретения студентом новых знаний и умений без непосредственного участия преподавателя, но под его методическим руководством.

В рамках данной дисциплины самостоятельная работа направлена на глубокое изучение теории реляционных баз данных и приобретение практических навыков проектирования схем БД, написания эффективных SQL-запросов и их интеграции в веб-приложения на языке PHP.

Целями самостоятельной работы являются:

- систематизация и закрепление полученных теоретических знаний по проектированию и нормализации баз данных;
- углубление знаний в области языка SQL (DML, DDL) и оптимизации запросов;
- формирование умений работать с современными СУБД (MySQL/MariaDB) и средствами администрирования (phpMyAdmin, консоль);
- развитие навыков обеспечения безопасности данных (защита от SQL-инъекций) и целостности транзакций.

Организационные мероприятия, обеспечивающие нормальное функционирование самостоятельной работы студента:

Самостоятельная работа должна быть конкретной по своей предметной направленности (разработка схемы БД под задачу).

Самостоятельная работа должна сопровождаться эффективным, непрерывным контролем (проверка ER-диаграмм, анализ кода запросов).

Виды самостоятельной работы по дисциплине:

Изучение теоретического материала: работа с документацией MySQL, учебниками по теории баз данных.

Практическое моделирование: создание ER-диаграмм, проектирование схем данных для различных предметных областей.

Выполнение лабораторных работ: написание SQL-скриптов, создание PHP-кода для работы с БД через PDO.

Разработка индивидуального проекта: создание полнофункциональной информационной системы с веб-интерфейсом.

2 СТРУКТУРА ДИСЦИПЛИНЫ

Введение. Дисциплина «Разработка серверных приложений с базами данных» реализуется в учебном плане среднего профессионального образования по специальности 09.02.09 «Веб-разработка».

Место дисциплины в структуре образовательной программы

Дисциплина является логическим продолжением курсов «Основы веб-разработки и верстки» и «Фронтенд-разработка». Она закладывает фундамент для последующего изучения работы с базами данных и проектирования сложных информационных систем.

Дисциплина отнесена к профессиональному циклу Образовательной программы, реализуется в рамках профессионального модуля ПМ.04 «Разработка веб-приложений на стороне сервера».

Дисциплина реализуется в пятом и шестом семестре.

Форма контроля – зачет, экзамен.

Целью освоения дисциплины является теоретическая и практическая подготовка студентов в области проектирования, создания и администрирования реляционных баз данных MySQL, а также интеграции их с веб-приложениями на языке PHP для реализации полноценного управления данными.

Задачи освоения дисциплины:

Изучение принципы проектирования реляционных баз данных и нормализации.

Освоение языка SQL для создания структуры БД (DDL) и манипулирования данными (DML).

Получение навыков программного взаимодействия с БД через PHP (библиотека PDO).

Изучение методов оптимизации запросов и обеспечения производительности.

Освоение технологий защиты данных и обеспечения безопасности веб-приложений.

Требования к результатам освоения дисциплины:

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

Знать:

- основные объекты базы данных (таблицы, представления, индексы);
- типы данных MySQL и критерии их выбора;
- принципы реляционной целостности и нормализации (1НФ, 2НФ, 3НФ);
- синтаксис SQL-запросов (SELECT, INSERT, UPDATE, DELETE, JOIN);

– методы защиты от SQL-инъекций (подготовленные запросы);

– принципы работы транзакций (ACID) и индексов.

Уметь:

– проектировать структуру базы данных для веб-приложения;

- писать сложные SQL-запросы с объединением таблиц и агрегацией данных;
- использовать расширение PDO для безопасного подключения к БД из PHP;
- реализовывать CRUD-интерфейс (создание, чтение, обновление, удаление);
- анализировать и оптимизировать медленные запросы.

Владеть:

- навыками работы с инструментами администрирования (phpMyAdmin, Workbench);
- технологией создания безопасной системы аутентификации пользователей;
- методами резервного копирования и восстановления баз данных.

В результате освоения учебной дисциплины обучающийся должен обладать профессиональными компетенциями:

- ПК 4.1. Администрировать среды и платформы разработки информационных ресурсов.
- ПК 4.2. Создавать программный код на стороне сервера в соответствии с техническим заданием (спецификацией) с использованием языков программирования, библиотек и фреймворков.
- ПК 4.3. Осуществлять отладку программного кода на стороне сервера на уровне программных модулей, межмодульных взаимодействий и взаимодействий с окружением.

Содержание дисциплины

Семестр № 5

1. Введение в архитектуру и инструменты веб-приложений с базой данных

Клиент-серверная архитектура в контексте PHP и БД: роль веб-сервера (Apache/Nginx), интерпретатора PHP и СУБД (MySQL). Понятие "полного стека" (LAMP/LEMP). Жизненный цикл запроса: от браузера до БД и обратно. Инструменты разработчика: IDE (PhpStorm, VSCode), системы контроля версий (Git, базовые команды), административные инструменты для БД (phpMyAdmin, Adminer, CLI MySQL). Локальная среда разработки: сборки (OpenServer, Denwer) vs виртуальные среды (Docker).

2. Основы языка SQL для разработки серверной части
 Структура SQL: DML (манипуляция) и DDL (определение). Ключевые операции: выборка данных (SELECT с WHERE, ORDER BY, LIMIT), вставка (INSERT), обновление (UPDATE), удаление (DELETE). Фильтрация и работа с условиями. Агрегирующие функции (COUNT, SUM, AVG) и группировка

(GROUP BY). Создание и модификация структур: работа с таблицами (CREATE TABLE, ALTER TABLE), типами данных (INT, VARCHAR, TEXT, DATETIME), ключами (PRIMARY KEY, FOREIGN KEY). Важность индексов для производительности.

3. Интеграция PHP и MySQL. Безопасность базовых операций
Подключение к базе данных из PHP: расширения (mysqli, PDO), установка соединения, обработка ошибок подключения. Выполнение запросов: mysqli_query, PDO::query, PDO::exec. Защита от SQL-инъекций: подготовленные выражения (mysqli_prepare/bind_param, PDO::prepare/bindValue) как единственный надежный способ. Чтение результатов: работа с результирующими наборами (mysqli_fetch_assoc, PDOStatement::fetch). Правильное закрытие соединений и освобождение ресурсов.

Семестр № 6

4. Реализация бизнес-логики и управление состоянием приложения
Организация кода: разделение логики (работа с БД) и представления (HTML). Паттерн "Транзакционный сценарий". Управление сессиями в PHP (session_start, \$_SESSION): аутентификация и авторизация пользователя, корзина покупок. Работа с cookies (setcookie). Реализация типовых бизнес-сценариев: регистрация и валидация данных, многошаговые формы, фильтрация и пагинация списков (LIMIT ... OFFSET). Основы роутинга (через .htaccess или простой фронт-контроллер).

5. Проектирование сложных структур и оптимизация взаимодействия с БД

Нормализация базы данных (1NF, 2NF, 3NF) для избежания аномалий. Связи между таблицами: один-ко-многим (JOIN), многие-ко-многим (через связующую таблицу). Использование операторов JOIN (INNER, LEFT) в SQL для объединения данных. Проблема N+1 запроса и стратегии ее решения: агрегирующие запросы, жадная загрузка (Eager Loading). Оптимизация запросов: анализ с помощью EXPLAIN, использование индексов, кэширование результатов (на уровне приложения с помощью Memcached/Redis или запросов).

6. Безопасность, администрирование и современные тенденции

Всесторонняя безопасность: защита от XSS (экранирование вывода, htmlspecialchars), CSRF (использование токенов), безопасная загрузка файлов. Работа с паролями: хеширование (password_hash, password_verify). Резервное копирование и восстановление БД (mysqldump). Базовое администрирование MySQL: создание пользователей, назначение привилегий (GRANT). Обзор современных подходов: использование ORM (Doctrine, Eloquent) как альтернатива чистому SQL, паттерн Active Record vs Data Mapper, тенденции к использованию независимых слоев доступа к данным.

3 РАСПРЕДЕЛЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО РАЗДЕЛАМ

Цель методических рекомендаций: помочь студентам при самостоятельном изучении учебной программы с использованием лекционных материалов и рекомендуемой учебно-методической литературы.

Номер раздела данной дисциплины	Наименование тем, вопросов, вынесенных для самостоятельного изучения
Семестр № 5	
Тема 1. Принципы проектирования интерфейсов администратора (Admin Panel) для работы с данными.	Основные задачи админки: CRUD-операции, модерация, аналитика. Ключевые элементы: таблицы с сортировкой и фильтрацией, сложные формы, дашборды. Принципы: безопасность (роли и логи), удобство (пагинация, массовые операции). Кастомизация vs готовые решения.
Тема 2. Автоматизация рутинных операций: SQL-скрипты и консольные утилиты для администрирования БД.	Автоматизация задач администрирования: SQL-скрипты для чистки/обновления данных, дампа БД (mysqldump). Консольные скрипты на PHP (CLI) для импорта CSV, отправки отчетов. Запуск по расписанию через cron.
Тема 3. Альтернативные методы аутентификации и основы безопасности сессий.	Защита сессий PHP: регенерация ID, привязка к атрибутам. Механизм "Запомнить меня". Альтернативы: аутентификация по токенам (JWT для API), двухфакторная аутентификация (2FA/TOTP).
Семестр № 6	
Тема 4. Введение в систему контроля версий Git для совместной разработки проекта с БД.	Базовый workflow Git: add, commit, push, pull. Что не коммитить (пароли, дампы БД). Работа с БД в команде: миграции и сиды для версионирования структуры и тестовых данных. Ветвление.
Тема 5. Проектирование и реализация простого REST API для мобильного клиента.	Принципы REST: ресурсы, HTTP-методы, статусы. Реализация на PHP: парсинг входного JSON, отдача JSON-ответов. Аутентификация через токен в заголовке. Базовые эндпоинты (список, элемент, создание). Пагинация.
Тема 6. Стратегии кэширования данных для повышения	Цели и уровни кэширования (БД, приложение, HTTP). Инструменты: Memcached, Redis. Паттерны в PHP: "сначала

производительности веб-приложения.

кэш" (Cache-Aside), инвалидация по TTL или событию. Кэширование фрагментов HTML и HTTP-заголовки (Cache-Control).

4 КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ ДЛЯ САМОПРОВЕРКИ

1. Виды и формы самостоятельной работы студентов по дисциплине «Разработка серверных приложений с базами данных»:

- изучение документации MySQL и PHP PDO;
- проектирование ER-диаграмм баз данных;
- выполнение лабораторных работ по написанию SQL-запросов и PHP-кода;
- подготовка к зачету и экзамену.

2. Методические рекомендации:

Используйте phpMyAdmin или консоль MySQL для тестирования «сырых» SQL-запросов перед внедрением их в PHP-код.

При проектировании БД всегда рисуйте схему связей таблиц на бумаге или в редакторе.

Оформляйте отчеты по лабораторным работам со скриншотами структуры таблиц и результатами выборки.

3. Подготовка к аттестации:

Убедитесь в понимании разницы между DDL (создание структуры) и DML (работа с данными).

Повторите синтаксис JOIN и агрегатных функций.

Перечень информационных технологий

Для проведения лекционных и лабораторных занятий рекомендуется использовать программное обеспечение:

1. **Операционная система:** Windows, macOS или Linux.
2. **Пакет офисных программ:** Microsoft Office, LibreOffice или аналоги (для оформления отчетов).
3. **Веб-браузеры:** Google Chrome, Mozilla Firefox или Microsoft Edge (для тестирования веб-интерфейсов).
4. **Интегрированная среда разработки (IDE) или редактор кода:** Visual Studio Code, PhpStorm (рекомендуется для PHP), Sublime Text.
5. **Локальный серверный комплекс (WAMP/LAMP):** OpenServer Panel, XAMPP или Docker (должен включать веб-сервер Apache/Nginx, интерпретатор PHP и СУБД MySQL/MariaDB).

6. **Средства администрирования баз данных:** phpMyAdmin (обычно входит в состав локального сервера), MySQL Workbench, DBeaver или HeidiSQL (для визуального проектирования и управления БД).

Контрольные вопросы

1. Что такое База Данных (БД) и Система Управления Базами Данных (СУБД)?
2. Перечислите основные объекты реляционной БД (таблица, строка, столбец, ключ).
3. Какие типы данных существуют в MySQL для хранения целых чисел, строк и дат?
4. Что такое Первичный ключ (PRIMARY KEY)? Каковы его свойства?
5. Что такое Автоинкремент (AUTO_INCREMENT)?
6. Как создать новую базу данных и таблицу с помощью SQL?
7. Как добавить новую запись в таблицу (команда INSERT)?
8. Как выбрать данные из таблицы (команда SELECT)?
9. Как отфильтровать выборку данных (оператор WHERE)?
10. Как отсортировать полученные данные (ORDER BY)?
11. Как ограничить количество выбираемых записей (LIMIT)?
12. Что такое Внешний ключ (FOREIGN KEY)? Зачем он нужен?
13. Опишите виды связей между таблицами (1:1, 1:N, N:M).
14. Что такое целостность данных? Каскадное удаление (ON DELETE CASCADE).
15. Для чего используется оператор JOIN?
16. В чем разница между INNER JOIN и LEFT JOIN?
17. Как обновить данные в таблице (UPDATE)? Почему важен WHERE?
18. Как удалить запись (DELETE)? Что такое "мягкое удаление"?
19. Что такое PDO? Преимущества перед старым расширением mysql.
20. Как подключиться к базе данных через PDO в PHP?
21. В чем опасность подстановки переменных напрямую в SQL-запрос?
22. Что такое SQL-инъекция? Приведите пример.
23. Как работают Подготовленные запросы (Prepared Statements)?
24. Как получить ID последней вставленной записи (lastInsertId)?
25. Как безопасно хранить пароли пользователей в БД?
26. Что такое Нормализация? Опишите требования 1, 2 и 3 нормальных форм.
27. Что такое денормализация и когда она применяется?
28. Что такое Транзакция? Расшифруйте аббревиатуру ACID.
29. Как начать, подтвердить и отменить транзакцию (START, COMMIT, ROLLBACK)?
30. Что такое Индекс? Зачем он нужен и как влияет на скорость записи?

31. Как проанализировать эффективность запроса (EXPLAIN)?
32. Что такое Группировка (GROUP BY)?
33. Какие агрегатные функции вы знаете (COUNT, SUM, AVG, MAX, MIN)?
34. Как реализовать постраничную навигацию (пагинацию) средствами SQL?
35. Как организовать хранение иерархических данных (деревьев) в реляционной БД?
36. Что такое Полнотекстовый поиск? Отличия от LIKE.
37. Как защитить веб-приложение от XSS и CSRF атак при работе с БД?
38. Зачем нужно резервное копирование? Утилита mysqldump.
39. Что такое Миграции базы данных?
40. Что такое ORM (Object-Relational Mapping)?
41. В чем отличие SQL и NoSQL баз данных?
42. Как реализовать связь "Многие-ко-многим" в MySQL?
43. Как работает кэширование запросов?
44. Что такое представление (VIEW) в базе данных?
45. Как оптимизировать запрос с JOIN?

5 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Индивидуальное задание выполняется в течение двух семестров и представляет собой разработку комплексной Информационной Системы (ИС) с веб-интерфейсом.

Цель работы: Проектирование и реализация реляционной базы данных и создание веб-приложения на PHP, обеспечивающего управление данными этой БД.

Требования к проекту:

1. База данных (MySQL):
 - Минимум 4-5 связанных таблиц (сущностей).
 - Наличие связей «Один-ко-многим» (1:N) и «Многие-ко-многим» (N:M).
 - Правильное использование типов данных и индексов.
 - Соблюдение правил нормализации (минимум до 3НФ).
2. Функционал приложения (PHP):
 - Аутентификация: Регистрация, вход, выход, разграничение прав (Администратор / Пользователь).
 - CRUD: Полный цикл управления данными (Создание, Чтение, Обновление, Удаление) для основных сущностей через веб-интерфейс.

- Поиск и Фильтрация: Возможность поиска записей и фильтрации по параметрам.
 - Пагинация: Постраничный вывод больших списков данных.
 - Безопасность: Использование подготовленных запросов (PDO), хеширование паролей, защита от XSS/CSRF.
3. Интерфейс:
- Понятный и удобный пользовательский интерфейс (можно использовать CSS-фреймворки, например Bootstrap).

Примерный перечень тем индивидуальных проектов:

1. Интернет-магазин автозапчастей: каталог, категории, производители, корзина, заказы.
2. Система управления отелем: номера, категории, клиенты, бронирования, услуги.
3. Электронный журнал успеваемости: студенты, группы, дисциплины, оценки, преподаватели.
4. Сайт агентства недвижимости: объекты, типы недвижимости, риелторы, клиенты, сделки.
5. Онлайн-аптека: лекарства, категории, производители, рецепты, заказы.
6. Форум для программистов: разделы, темы, сообщения, пользователи, рейтинги.
7. Система учета библиотечного фонда: книги, авторы, издательства, читатели, выдача книг.
8. Сервис доставки еды: рестораны, меню, блюда, курьеры, заказы.
9. Сайт кадрового агентства: вакансии, компании, резюме, соискатели, отклики.
10. Система управления проектами (Bug Tracker): проекты, задачи, статусы, приоритеты, исполнители.
11. Онлайн-кинотеатр: фильмы, жанры, актеры, режиссеры, отзывы, расписание сеансов.
12. Социальная сеть (мини): пользователи, посты, друзья, сообщения, лайки.
13. Складской учет: товары, склады, поставщики, накладные прихода/расхода.
14. Сайт фитнес-клуба: абонементы, тренеры, групповые занятия, расписание, залы.
15. Сервис аренды автомобилей: автопарк, классы авто, клиенты, договоры аренды, штрафы.
16. Справочник рецептов: блюда, ингредиенты, категории, пользователи, комментарии.
17. Система тестирования знаний: тесты, вопросы, варианты ответов, результаты, студенты.

18. Учет компьютерной техники: компьютеры, комплектующие, аудитории, сотрудники, инвентаризация.
19. Сайт ветеринарной клиники: питомцы, владельцы, врачи, приемы, диагнозы.
20. Доска объявлений: объявления, категории, регионы, пользователи, модерация.
21. Сервис записи на услуги (Салон красоты): услуги, мастера, график работы, записи клиентов.
22. Учет личных финансов (семейный бюджет): счета, категории трат, доходы, расходы, члены семьи.
23. Сайт музыкальной группы: альбомы, треки, концерты, новости, фанаты.
24. Система голосования/петиций: петиции, категории, подписи, пользователи.
25. Блог-платформа (Multi-user): блоги, посты, теги, комментарии, подписчики.
26. Сайт туристического агентства: туры, страны, отели, заказы, клиенты.
27. Система учета заявок (HelpDesk): заявки, отделы, сотрудники, статусы, приоритеты.
28. Аукцион: лоты, категории, ставки, пользователи, история торгов.
29. Сервис отзывов о компаниях: компании, рубрики, отзывы, оценки, пользователи.
30. Учет научных публикаций: статьи, журналы, авторы, конференции, цитирование.

Примерный план выполнения индивидуальной работы

1. Проектирование БД: Составление списка сущностей и атрибутов. Построение ER-диаграммы. Нормализация схемы.
2. Реализация БД: Создание таблиц и связей в MySQL (через phpMyAdmin или SQL-скрипт). Наполнение тестовыми данными.
3. Каркас приложения: Создание структуры папок, файла подключения к БД (db.php), настройка автозагрузки.
4. Аутентификация: Реализация регистрации и входа.
5. Разработка модулей: Поочередная реализация CRUD для каждой сущности (Например: сначала Категории, потом Товары).
6. Связывание данных: Реализация вывода связанных данных (например, названия категории в списке товаров) через JOIN.
7. Поиск и фильтры: Добавление форм поиска и обработки параметров запроса.

8. Тестирование и отладка: Проверка всех сценариев работы, защита от ошибок ввода.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Дюбуа, П. MySQL. Сборник рецептов / Поль Дюбуа. – 3-е изд. – Санкт-Петербург : БХВ-Петербург, 2016. – 1076 с.
2. Шварц, Б. MySQL 8.0. Оптимизация и масштабирование / Бэррон Шварц, Петр Зайцев, Вадим Ткаченко. – Москва : ДМК Пресс, 2020. – 480 с.
3. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Робин Никсон. – 5-е изд. – Санкт-Петербург : Питер, 2021. – 816 с.
4. Веллинг, Л. Разработка веб-приложений с помощью PHP и MySQL / Люк Веллинг, Лаура Томсон. – 5-е изд. – Москва : Вильямс, 2019. – 1072 с.
5. Фейерштейн, С. Построение баз данных. Язык запросов SQL / Стивен Фейерштейн. – Санкт-Петербург : Питер, 2021. – 528 с.
6. Яргер, Р. MySQL. Библиотека профессионала / Рэнди Яргер, Джордж Риз. – Москва : Лори, 2014. – 604 с.
7. Официальная документация MySQL [Электронный ресурс]. – Режим доступа: <https://dev.mysql.com/doc/> – Загл. с экрана.
8. Руководство по PHP Data Objects (PDO) [Электронный ресурс] // Официальный сайт PHP. – Режим доступа: <https://www.php.net/manual/ru/book.pdo.php> – Загл. с экрана.
9. SQL-Tutorial (W3Schools) [Электронный ресурс]. – Режим доступа: <https://www.w3schools.com/sql/> – Загл. с экрана.