Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Костровец Лариса Борисовна
Должность: директор
Дата подписания: 02.10.2026 00:12:42
Уникальный программный ключ:
6882606104c36dbde41c4ab93a65382136a292d6

Приложение 4
к образовательной программе

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.О.02.06 Базы данных
(индекс, наименование дисциплины в соответствии с учебным планом)

09.03.03 Прикладная информатика
(код, наименование направления подготовки/специальности)

Прикладная информатика в управлении корпоративными информационными системами
(наименование образовательной программы)

Очная форма обучения
(форма обучения)

Год набора – 2026
Донецк

Автор(ы)-составитель(и) РПД:

Литвак Елена Геннадиевна, доцент кафедры информационных технологий

Заведующий кафедрой:

Брадул Наталья Валерьевна, канд. физ.-мат. наук, заведующий кафедрой информационных технологий

Рабочая программа дисциплины Б1.О.02.06 Базы данных одобрена на заседании кафедры информационных технологий факультета государственной службы и управления Донецкого филиала РАНХиГС.

Протокол № 7 от «05» марта 2026 г.

СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

2. Объем и место дисциплины в структуре образовательной программы

3. Содержание и структура дисциплины

4. Типы оценочных материалов, показатели и критерии их оценивания

5. Формы аттестации, типовые оценочные материалы для текущего контроля успеваемости обучающихся, критерии и шкалы оценивания по контрольным точкам

6. Формы промежуточной аттестации, критерии и шкала оценивания, типовые оценочные материалы по дисциплине

7. Методические материалы по освоению дисциплины

8. Учебная литература и ресурсы информационно-телекоммуникационной сети «Интернет»

9. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Дисциплина Б1.О.02.06 Базы данных обеспечивает формирование у обучающихся следующих общепрофессиональных компетенций*:

ОТФ/ТФ и реквизиты ПС *(при наличии)* **	**Код компетенции ****	**Наименование Компетенции ****	**Код индикатора достижения компетенций ****	**Наименование индикатора достижения компетенций ****	**Образовательный результат ****
-	ОПК-7	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.2	Разрабатывает алгоритмы и программы с использованием баз данных пригодные для практического применения	**Знает** принципы построения и функционирования баз данных. **Умеет** разрабатывать и использовать базы данных в программных приложениях.

** Дисциплина может формировать компетенцию полностью или частично.*
***Должно соответствовать Приложению 1 к образовательной программе*

2. Объем и место дисциплины в структуре образовательной программы

Общий объем дисциплины:

7,00 з.е., 252 ак.час

Контактная работа обучающихся с преподавателем по видам учебных занятий: 146 ак. час на контактную работу с преподавателем, из них 54 ак. час на лекции и 72 ак. час на практические занятия. 79 ак. час на самостоятельную работу обучающихся.

Б1.О.02.06. Базы данных реализуется на 3-м и 4-м семестре 2-го курса после изучения дисциплин:

- Информационные системы и технологии.

.

3. Содержание и структура дисциплины

3.1. Структура дисциплины

Очная форма обучения

№ п/п	Наименование тем и (или) разделов	ВСЕГО	Объем дисциплины, ак.час												Форма текущего контроля успеваемости, промежуточной аттестации
			Контактная работа обучающихся с преподавателем по видам учебных занятий									Самостоятельная работа			
			Период теоретического обучения						Период промежуточной аттестации (сессия)						
			Занятия лекционного типа		Занятия семинарского типа		ИК	КСР	КЭ	Каттэк	Контроль	СРкр	СРэк	СР	
			Л	ВЛ	ЛР	ПЗ									
РАЗДЕЛ 1. Введение в теорию баз данных															
Тема 1	Математические основы построения реляционных БД. Определение и терминология. Математические отношения. Ключи. Целостность	11	2	0	0	4	0	0	0	0		0	0	5	Контрольные вопросы, практические занятия, КТ1

Тема 2	Разработка модели данных на основе нормализации. Цель нормализации. Избыточность данных и аномалии обработки. Нормальные формы 1НФ и 2НФ	11	2	0	0	4	0	0	0	0		0	0	5	Контрольные вопросы, практические занятия, КТ1
Тема 3	3 НФ, составной ключ	11	2			4								5	Контрольные вопросы, практические занятия, КТ1
РАЗДЕЛ 2. ОСНОВЫ ЯЗЫКА SQL															
Тема 4	Синтаксис SQL-операторов CREATE TABLE, INSERT, DELETE, UPDATE.	11	2	0	0	4	0	0	0	0		0	0	5	Контрольные вопросы, практические занятия, КТ2
Тема 5	Синтаксис оператора SELECT. Выборка данных по критериям	11	2	0	0	4	0	0	0	0		0	0	5	Контрольные вопросы, практические занятия, КТ2

Тема 6	Оператор join. Левое и правое соединение. Full join	11	2	0	0	4	0	0	0	0		0	0	5	Контрольные вопросы, практические занятия, КТ2
РАЗДЕЛ 3. СЛОЖНЫЕ ТИПЫ ЗАПРОСОВ															
Тема 7	Групповые операции в запросах, сортировка, вычисления в запросах	11	2	0	0	4	0	0	0	0	0	0	0	5	Контрольные вопросы, практические занятия, КТ3
Тема 8	Использование подзапросов. Встроенные функции (day(), month(), year() и др.)	11	2	0	0	4	0	0	0	0	0	0	0	5	Контрольные вопросы, практические занятия, КТ3
Тема 9	Создание и использование представлений	11	2	0	0	4	0	0	0	0	0	0	0	5	Контрольные вопросы, практические занятия, КТ3
Промежуточная аттестация										**9**					**Зачет**
Итого за семестр		**108**	**18**	**0**	**0**	**36**	**0**		**0**	**9**	**0**	**0**	**0**	**45**	
РАЗДЕЛ 4. Процедуры и функции в СУБД															
Тема 10	Язык создания хранимых процедур и	10	4	0	0	4	0	0	0	0		0	0	2	Контрольные вопросы, практические

	функций. Передача параметров в хранимые процедуры и функции														занятия, КТ4
Тема 11	Создание триггеров	12	4	0	0	4	0	0	0	0		0	0	4	Контрольные вопросы, практические занятия, КТ4
Тема 12	Работа с транзакциями и типами блокировок	12	4	0	0	4	0	0	0	0		0	0	4	Контрольные вопросы, практические занятия, КТ4
РАЗДЕЛ 5. ОСНОВЫ ВЕБ-ТЕХНОЛОГИЙ															
Тема 13	Определение, основные понятия: веб-сервер, серверный язык программирования.	12	4	0	0	4	0	0	0	0		0	0	4	Контрольные вопросы, практические занятия, КТ5
Тема 14	Инсталляция Apache+PHP+MySql на различные операционные системы	12	4	0	0	4	0	0	0	0		0	0	4	Контрольные вопросы, практические занятия, КТ5
		12	4	0	0	4	0	0	0	0		0	0	4	Контрольные

Тема 15	Введение в язык PHP														вопросы, практические занятия, КТ5
РАЗДЕЛ 6. РАБОТА С БАЗАМИ ДАННЫХ ИЗ ЯЗЫКА PHP															
Тема 16	Подключение к базе. Получение данных из базы; Передача параметров через адресную строку. Метод GET	12	4	0	0	4	0	0	0	0	0	0	0	4	Контрольные вопросы, практические занятия, КТ6
Тема 17	Добавление данных в базу. Форма регистрации пользователей. Метод POST	12	4	0	0	4	0	0	0	0	0	0	0	4	Контрольные вопросы, практические занятия, КТ6
Тема 18	Создание формы для редактирования данных	12	4	0	0	4	0	0	0	0	0	0	0	4	Контрольные вопросы, практические занятия, КТ6
Промежуточная аттестация									**2**	**9**		**9**	**18**		**Экзамен**

Итого за семестр	**144**	**36**	**0**	**0**	**36**	**0**		**2**	**9**		**9**	**18**	**34**	
Итого	**252**	**54**			**72**			**2**	**18**	**9**	**18**	**18**	**79**	

Используемые сокращения:

Л – лекции - занятия, предусматривающие преимущественную передачу учебной информации обучающимся педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях,).

ВЛ – видео лекции.

ЛР – лабораторные работы.

ПЗ – практические занятия (за исключением лабораторных работ).

ИК – индивидуальные консультации.

КСР – контроль самостоятельной работы

КЭ – консультации перед экзаменом

Каттэк – контактная работа на аттестацию в период экзаменационных сессий

Контроль - контактная работа на аттестацию в период экзаменационных сессий для заочной формы обучения

СРкр – самостоятельная работа на подготовку курсовой работы/ курсового проекта.

СРэк – самостоятельная работа на подготовку к экзамену.

СР – самостоятельная работа в семестре на подготовку к учебным занятиям.

3.2. Содержание дисциплины

РАЗДЕЛ 1. ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

Тема 1. Математические основы построения реляционных БД. Определение и терминология. Математические отношения. Ключи. Целостность. ОПК-7.2

Содержание лекции: Понятие предметной области и сущности. Математическое определение отношения как подмножества декартова произведения доменов. Основные термины реляционной модели: отношение (таблица), кортеж (строка), атрибут (столбец), домен (допустимые значения). Определение первичного ключа (Primary Key): уникальная идентификация каждой записи, свойство NOT NULL и уникальность. Определение внешнего ключа (Foreign Key): обеспечение ссылочной целостности между таблицами. Правила целостности данных: целостность сущностей (первичный ключ не может быть пустым) и ссылочная целостность (внешний ключ либо NULL, либо соответствует существующему первичному ключу).

Практическая подготовка: Выделение сущностей и атрибутов в описании предметной области (например, «Университет», «Магазин», «Библиотека»). Построение диаграммы «сущность-связь» (ER-диаграммы) с обозначением первичных и внешних ключей. Определение типов связей (1:1, 1:M, M:N) на конкретных примерах.

Тема 2. Разработка модели данных на основе нормализации. Цель нормализации. Избыточность данных и аномалии обработки. Нормальные формы 1НФ и 2НФ. ОПК-7.2

Содержание лекции: Цель нормализации – устранение избыточности данных и аномалий обработки (аномалии вставки, обновления, удаления). Определение избыточности: многократное дублирование одних и тех же данных в разных строках. Аномалии: невозможность добавить информацию без наличия другой информации (вставка), рассогласование данных при изменении (обновление), потеря данных при удалении одной строки (удаление). Правила приведения к первой нормальной форме (1НФ): атомарность всех атрибутов, отсутствие повторяющихся групп, уникальность строк (наличие первичного ключа). Правила приведения ко второй нормальной форме (2НФ): таблица должна быть в 1НФ, и все неключевые атрибуты должны зависеть от полного первичного ключа (устранение частичной функциональной зависимости).

Практическая подготовка: Анализ ненормализованной таблицы («Платежи», «Заказы») на предмет избыточности и аномалий. Пошаговое преобразование таблицы из 0НФ в 1НФ. Разбор случаев с составным первичным ключом и преобразование из 1НФ в 2НФ путем выделения зависимых атрибутов в отдельные таблицы.

Тема 3. 3 НФ, составной ключ. ОПК-7.2

Содержание лекции: Определение третьей нормальной формы (3НФ): таблица находится во 2НФ, и в ней отсутствуют транзитивные зависимости – неключевые атрибуты не должны зависеть от других неключевых атрибутов. Понятие составного первичного ключа (Composite Key): ключ, состоящий из двух и более столбцов, совместно обеспечивающих уникальность строки. Типичное применение составного ключа: таблицы-связки для реализации связи «многие-ко-многим». Правило устранения транзитивной зависимости: вынесение атрибутов, описывающих другую сущность, в отдельную таблицу и связывание через внешний ключ.

Практическая подготовка: Выявление транзитивных зависимостей в таблицах, находящихся во 2НФ (на примере: «Студенты» → «Группы» → «Факультеты»). Проектирование схемы БД в 3НФ для заданной предметной области. Разбор примера таблицы-связки с составным ключом (например, «Студент_Дисциплина» с полями «ID студента» и «ID дисциплины»).

РАЗДЕЛ 2. ОСНОВЫ ЯЗЫКА SQL

Тема 4. Синтаксис SQL-операторов CREATE TABLE, INSERT, DELETE, UPDATE. ОПК-7.2

Содержание лекции: Язык определения данных (DDL) – оператор CREATE TABLE: синтаксис задания имени таблицы, столбцов, типов данных (INT, VARCHAR(n), DATE, DECIMAL), ограничений целостности (PRIMARY KEY, FOREIGN KEY, NOT NULL, UNIQUE, CHECK). Язык манипуляции данными (DML): синтаксис INSERT INTO (добавление одной строки – VALUES, добавление из другого запроса – SELECT). Синтаксис UPDATE: SET, WHERE. Синтаксис DELETE FROM: WHERE. Правило: отсутствие WHERE приводит к удалению/обновлению всех строк. Каскадные операции ON DELETE CASCADE / ON DELETE SET NULL при создании внешнего ключа.

Практическая подготовка: Написание скриптов создания 3-4 связанных таблиц с правильными типами данных и ограничениями. Выполнение операций вставки (INSERT) тестовых данных. Написание запросов UPDATE и DELETE с разными условиями WHERE. Анализ результатов каскадного удаления.

Тема 5. Синтаксис оператора SELECT. Выборка данных по критериям. ОПК-7.2

Содержание лекции: Базовый синтаксис SELECT: SELECT (список столбцов или * для всех), FROM (имя таблицы), WHERE (условие отбора). Операторы сравнения (=, <>, >, <, >=, <=). Логические операторы (AND, OR, NOT). Специальные операторы: BETWEEN (диапазон), IN (перечисление), LIKE (шаблоны с % и _), IS NULL (проверка на пустое значение). Определение

порядка выполнения запроса: FROM → WHERE → SELECT → ORDER BY → LIMIT. Назначение ORDER BY (ASC, DESC) и LIMIT.

Практическая подготовка: Составление запросов выборки по различным критериям на учебной БД. Составление сложных условий с комбинированием AND/OR. Использование LIKE для поиска по части строки (например, фамилии на «Ива»). Сортировка и ограничение количества результатов.

Тема 6. Оператор join. Левое и правое соединение. Full join. ОПК-7.2

Содержание лекции: Понятие соединения таблиц (JOIN) – объединение данных из двух и более таблиц по условию связи. Внутреннее соединение INNER JOIN: возвращаются только строки с совпадением в обеих таблицах. Внешние соединения: LEFT JOIN (все строки из левой таблицы, NULL для правой при отсутствии соответствия), RIGHT JOIN (все строки из правой таблицы, NULL для левой), FULL JOIN (все строки из обеих таблиц, NULL на месте отсутствующих соответствий). Синтаксис: FROM таблица1 LEFT JOIN таблица2 ON условие. Различие между JOIN и WHERE для связывания таблиц.

Практическая подготовка: Решение задач на построение запросов с INNER JOIN (например, «вывести список заказов с именами клиентов»). Решение задач на LEFT JOIN (например, «найти клиентов, у которых нет ни одного заказа»). Сравнение результатов при замене LEFT JOIN на RIGHT JOIN.

РАЗДЕЛ 3. СЛОЖНЫЕ ТИПЫ ЗАПРОСОВ

Тема 7. Групповые операции в запросах, сортировка, вычисления в запросах. ОПК-7.2

Содержание лекции: Агрегатные функции: COUNT (количество строк), SUM (сумма), AVG (среднее), MAX (максимум), MIN (минимум). Синтаксис GROUP BY – группировка строк по значениям одного или нескольких столбцов. Правило: все неагрегированные столбцы в SELECT должны присутствовать в GROUP BY. Фильтрация групп с помощью HAVING (в отличие от WHERE, который фильтрует строки ДО группировки). Вычисляемые столбцы в SELECT: арифметические выражения, конкатенация строк, вызовы функций. Комбинирование GROUP BY с ORDER BY и LIMIT.

Практическая подготовка: Составление запросов для подсчета статистики (например, «количество студентов по группам», «сумма продаж по каждому товару»). Решение задач с HAVING (например, «вывести группы, где средний балл > 4.5»). Сравнение WHERE и HAVING на одном и том же условии. Расчет производных показателей прямо в SELECT.

Тема 8. Использование подзапросов. Встроенные функции (day(), month(), year() и др.). ОПК-7.2

Содержание лекции: Определение подзапроса (вложенного SELECT).

Типы подзапросов: скалярные (возвращают одно значение), столбцовые (возвращают один столбец), табличные (возвращают таблицу). Операторы для работы с подзапросами: IN (проверка вхождения), EXISTS (проверка существования), ANY/SOME, ALL. Встроенные функции СУБД: дата-время (YEAR(), MONTH(), DAY(), NOW(), DATE_ADD(), DATEDIFF()), строковые (CONCAT(), UPPER(), LOWER(), SUBSTRING(), LENGTH()), числовые (ROUND(), CEIL(), FLOOR(), ABS()). Функции преобразования типов (CAST, CONVERT).

Практическая подготовка: Составление запросов с подзапросами в WHERE (оператор IN, EXISTS). Использование скалярного подзапроса в SELECT для вычисления дополнительных полей. Извлечение года, месяца из даты для группировки (агрегация продаж по месяцам). Форматирование строк с помощью CONCAT для вывода ФИО одной строкой.

Тема 9. Создание и использование представлений. ОПК-7.2

Содержание лекции: Определение представления (VIEW) – сохраненный именованный запрос, виртуальная таблица, не хранящая данные физически. Цели создания представлений: упрощение сложных запросов (скрытие многотабличных JOIN), обеспечение безопасности данных (предоставление доступа только к части столбцов или строк), создание абстрактного уровня доступа к БД. Синтаксис CREATE VIEW имя AS SELECT...; DROP VIEW. Ограничения обновляемых представлений: представление не должно содержать GROUP BY, DISTINCT, агрегатных функций, UNION, подзапросов в SELECT.

Практическая подготовка: Создание представления, объединяющего 3-4 таблицы с вычисляемыми полями. Запрос к созданному представлению как к обычной таблице. Попытка обновления данных через представление и анализ ошибок. Модификация и удаление представления.

РАЗДЕЛ 4. ПРОЦЕДУРЫ И ФУНКЦИИ В СУБД

Тема 10. Язык создания хранимых процедур и функций. Передача параметров в хранимые процедуры и функции. ОПК-7.2

Содержание лекции: Определение хранимой процедуры – сохраняемого на сервере набора SQL-инструкций. Отличие хранимой функции: функция обязательно возвращает скалярное значение и может использоваться в SQL-выражении. Типы параметров: IN (входной, по умолчанию), OUT (выходной, возвращает значение), INOUT (комбинированный). Синтаксис CREATE PROCEDURE имя (IN параметр тип) BEGIN ... END. Синтаксис CREATE FUNCTION имя (параметры) RETURNS тип DETERMINISTIC ... Вызов процедуры через CALL, функции – внутри SELECT. Преимущества: снижение сетевого трафика, инкапсуляция бизнес-логики, повышение безопасности.

Практическая подготовка: Создание хранимой процедуры с входным параметром (например, поиск сотрудников по отделу). Создание скалярной

функции (например, расчет возраста по дате рождения). Использование функции внутри запроса SELECT. Создание процедуры с OUT-параметром для возврата статистики (количество записей, сумма).

Тема 11. Создание триггеров. ОПК-7.2

Содержание лекции: Определение триггера – автоматически выполняемой процедуры при наступлении события с таблицей (INSERT, UPDATE, DELETE). Время срабатывания: BEFORE (до выполнения операции, для валидации или модификации вставляемых/обновляемых данных), AFTER (после операции, для логирования, обновления итоговых таблиц), INSTEAD OF (для представлений – заменяет стандартную операцию). Уровни срабатывания: FOR EACH ROW (для каждой затронутой строки) – наиболее распространенный. Доступ к значениям: OLD (старое значение для UPDATE и DELETE), NEW (новое значение для INSERT и UPDATE). Синтаксис CREATE TRIGGER имя BEFORE INSERT ON таблица FOR EACH ROW BEGIN ... END.

Практическая подготовка: Создание триггера BEFORE INSERT для автоматической установки даты создания записи (NOW()). Создание триггера AFTER DELETE для логирования удаленных записей в таблицу аудита. Создание триггера BEFORE UPDATE для проверки допустимости нового значения. Анализ производительности и рисков (каскадные срабатывания).

Тема 12. Работа с транзакциями и типами блокировок. ОПК-7.2

Содержание лекции: Свойства ACID: атомарность (All or Nothing), согласованность (сохранение целостности БД), изоляция (независимость параллельных транзакций), долговечность (сохранение после сбоя). Команды управления транзакциями: START TRANSACTION / BEGIN, COMMIT (фиксация изменений), ROLLBACK (откат). Точки сохранения (SAVEPOINT). Уровни изоляции транзакций: READ UNCOMMITTED (грязное чтение), READ COMMITTED (неповторяемое чтение), REPEATABLE READ (фантомное чтение), SERIALIZABLE (полная изоляция). Типы блокировок: разделяющие (Shared Lock, для чтения) и исключительные (Exclusive Lock, для записи). Понятие взаимоблокировки (deadlock).

Практическая подготовка: Выполнение группы операторов INSERT/UPDATE внутри транзакции с последующим COMMIT (успех) и ROLLBACK (откат). Моделирование сбоя между операциями. Создание ситуации взаимоблокировки в двух сессиях и анализ механизмов ее разрешения СУБД. Изменение уровня изоляции сессии и наблюдение за поведением параллельных запросов.

РАЗДЕЛ 5. ОСНОВЫ ВЕБ-ТЕХНОЛОГИЙ

Тема 13. Определение, основные понятия: веб-сервер, серверный язык программирования. ОПК-7.2

Классическая архитектура динамических веб-приложений: Apache + PHP

+ MySQL (так называемый «золотой стандарт» для небольших и средних проектов). Apache – это веб-сервер, который принимает HTTP-запросы от браузера, отдаёт статические файлы (HTML, CSS, изображения) и, при обнаружении файлов с расширением .php, передаёт их на обработку модулю PHP. PHP – это серверный язык программирования, встроенный в Apache как модуль (mod_php) либо работающий через CGI/FPM; он выполняется на стороне сервера, генерирует динамический HTML-контент и, что самое важное, содержит встроенные функции (mysqli, PDO) для подключения и работы с MySQL. MySQL – это реляционная СУБД, которая хранит все данные и выполяет SQL-запросы, переданные из PHP. Студент должен понимать полный цикл обработки запроса: браузер → Apache (принимает запрос) → PHP (интерпретирует код) → MySQL (выполняет SELECT/INSERT/UPDATE/DELETE) → PHP (формирует HTML-ответ) → Apache (отправляет ответ) → браузер. Также знать различие между клиентским кодом (JavaScript выполняется в браузере) и серверным кодом (PHP выполняется на сервере, пользователь его не видит).

Практическая подготовка:

Запуск локального стека (OpenServer / XAMPP) с активными сервисами Apache и MySQL. Создание в корневой директории веб-сервера (DocumentRoot) файла test.php с кодом <?php phpinfo(); ?> и проверка, что в выводе присутствуют разделы «Apache», «mysqlnd» (MySQL Native Driver) и «mysqli». Написание простейшего скрипта, который через mysqli_connect() подключается к MySQL (сервер localhost, пользователь root) и выполняет запрос SELECT VERSION() для вывода версии сервера БД. Анализ заголовков HTTP-ответа с помощью инструментов разработчика в браузере (вкладка Network), чтобы убедиться, что Apache отправляет корректные заголовки Content-Type: text/html. Схематическое рисование на доске/в тетради последовательности вызовов: браузер → Apache → PHP → MySQL → PHP → Apache → браузер с указанием протоколов (HTTP, TCP/IP, сокеты между PHP и MySQL).

Тема 14. Инсталляция Apache+PHP+MySql на различные операционные системы. ОПК-7.2

Содержание лекции: Понятие веб-стека (набора программ) для локальной разработки. Готовые сборки: XAMPP (кроссплатформенный), OpenServer (Windows), MAMP (macOS), LAMP (Linux: Linux, Apache, MySQL, PHP). Раздельная установка на Linux через пакетный менеджер (apt, yum, pacman). Основные конфигурационные файлы: php.ini (настройки PHP: error_reporting, display_errors, upload_max_filesize), my.cnf / my.ini (настройки MySQL: порт, путь к данным, буферы), httpd.conf (Apache: виртуальные хосты, модули). Понятие корневой директории сервера (DocumentRoot) и виртуальных хостов для размещения нескольких проектов.

Практическая подготовка: Установка XAMPP/OpenServer на учебный компьютер. Запуск веб-сервера и сервера БД из панели управления. Проверка работоспособности через создание файла phpinfo() в корневой директории. Настройка простого виртуального хоста (свое доменное имя, например,

test.local). Подключение к MySQL через phpMyAdmin (браузерный инструмент).

Тема 15. Введение в язык PHP. ОПК-7.2

Содержание лекции: PHP – скриптовый язык, встраиваемый в HTML через теги <?php ... ?>. Базовый синтаксис: переменные начинаются с ,строки(вкавычках),целыеивещественныечисла,булевызначения.Суперглобальныемассивы:,строки(вкавычках),целыеивещественныечисла,булевызначения.Суперглобальныемассивы:_GET (данные из строки запроса), POST(данные изформы), _SERVER (информация о сервере и запросе), $_SESSION (данные между страницами). Основные операторы: echo, print, условный оператор if-else, циклы (for, while, foreach). Функции для работы со строками (strlen, substr, explode), массивами (count, array_push). Правила конкатенации строк (оператор .).

Практическая подготовка: Написание простейшего PHP-скрипта, выводящего "Hello, World!" и текущую дату. Создание страницы, обрабатывающей параметры GET (переданные через ?name=Иван). Создание страницы с формой и обработчиком на POST (суммирование двух чисел). Использование цикла для генерации HTML-списка из массива данных.

РАЗДЕЛ 6. РАБОТА С БАЗАМИ ДАННЫХ ИЗ ЯЗЫКА PHP

Тема 16. Подключение к базе. Получение данных из базы; Передача параметров через адресную строку. Метод GET. ОПК-7.2

Содержание лекции: Способы подключения PHP к MySQL: расширение mysqli (объектно-ориентированный и процедурный стили) и PDO (абстракция от типа БД). Алгоритм подключения: создание соединения (mysqli_connect), проверка ошибок, установка кодировки (mysqli_set_charset(utf8)), выполнение запроса (mysqli_query). Получение результата: mysqli_fetch_assoc (ассоциативный массив), mysqli_fetch_row (индексированный массив), mysqli_fetch_all. Метод GET – передача параметров через URL (строка запроса после ?). Правило экранирования данных перед подстановкой в SQL: mysqli_real_escape_string или подготовленные выражения (prepared statements) для защиты от SQL-инъекций. Пример: id=_GET['id']; sql="SELECT∗FROMusersWHEREid=*sql*="*SELECT∗FROMusersWHEREid*=id";.

Практическая подготовка: Написание PHP-скрипта подключения к БД MySQL. Создание страницы, которая принимает GET-параметр (например, id товара) и выводит подробную информацию о нем из базы. Реализация списка новостей с гиперссылками «подробнее», где ID передается через GET. Демонстрация уязвимости к SQL-инъекции в "небезопасном" коде и её закрытие через экранирование.

Тема 17. Добавление данных в базу. Форма регистрации пользователей. Метод POST. ОПК-7.2

Содержание лекции: Метод POST – передача данных в теле HTTP-запроса, используется при изменении данных (INSERT, UPDATE, DELETE) и для конфиденциальной информации (пароли, email). Структура HTML-формы с method="POST" и action="обработчик.php". Обработка формы: получение данных из суперглобального массива $_POST. Этапы обработки регистрации: 1) получение данных (логин, email, пароль); 2) валидация (проверка заполненности, корректности email, длины пароля); 3) хэширование пароля функцией password_hash(); 4) формирование запроса INSERT INTO; 5) редирект на страницу приветствия через header('Location: ...') для предотвращения повторной отправки формы.

Практическая подготовка: Создание формы регистрации (логин, email, пароль). Написание обработчика, проверяющего уникальность логина (SELECT) перед вставкой. Хэширование пароля и вставка записи в таблицу users. Реализация страницы логина (авторизации) с проверкой пароля через password_verify(). Сохранение статуса авторизации в сессии ($_SESSION).

Тема 18. Создание формы для редактирования данных. ОПК-7.2

Содержание лекции: Двухэтапный паттерн редактирования. Этап 1 – вывод формы с текущими данными: получение ID редактируемой записи из _GET (например, ?id=5), выполнение SELECT, подстановка значений извлеченной строки в атрибуты value HTML-полей. Обязательное скрытое поле \ для сохранения ID между этапами. Этап 2 – сохранение изменений: обработка POST-запроса, валидация новых данных, формирование UPDATE table SET column1='value1', column2='value2' WHERE id = {скрытое_поле}. Защита от подмены ID: проверка прав пользователя (соответствие ID сессии или роли). Обработка ошибок (отсутствие записи, конфликт обновления).

Практическая подготовка: Создание страницы редактирования профиля пользователя с предзаполненными полями (имя, email). Реализация механизма изменения пароля (поле с новым паролем, обновление только при заполнении).

4. Типы оценочных материалов, показатели и критерии оценивания

4.1. Оценочные материалы по дисциплине Б1.О.14. Базы данных входят в состав оценочных материалов по образовательной программе. Совокупность оценочных материалов по всем дисциплинам (модулям) образовательной программы составляет фонд оценочных средств (далее – ФОС). ФОС используется при проведении текущего контроля успеваемости и промежуточной аттестации обучающихся с целью оценивания достижения обучающимися планируемых результатов обучения.

4.2. ФОС разработан как комплекс проверочных заданий различного типа и уровня сложности, включает критерии и шкалы оценивания, а также «ключи» правильных ответов. ФОС формируется как отдельный документ и хранится в электронном виде, доступ к ФОС предоставлен ограниченному

кругу лиц.

4.3. Для самостоятельной работы обучающихся при подготовке к текущему контролю успеваемости и промежуточной аттестации в рабочих программах дисциплин размещены типовые проверочные задания, которые можно условно разделить на задания закрытого, комбинированного и открытого типов.

Задания закрытого типа – это тестовые задания, в которых каждый вопрос сопровождается готовыми вариантами ответов, из которых необходимо выбрать один или несколько правильных.

Задания комбинированного типа – это тестовые задания, в которых каждый вопрос сопровождается готовыми вариантами ответов, из которых необходимо выбрать один или несколько правильных и обосновать свой выбор.

Задания открытого типа – это задания, в которых на каждый вопрос должен быть предложен развернутый обоснованный ответ.

В зависимости от типа задания рекомендованы определенная последовательность выполнения и система оценивания выполнения заданий.

4.4. Типы заданий, сценарии выполнения, критерии оценивания

ТИП ЗАДАНИЯ	ИНСТРУКЦИЯ	СЦЕНАРИИ ВЫПОЛНЕНИЯ	КРИТЕРИИ ОЦЕНИВАНИЯ
Задание закрытого типа с выбором одного правильного ответа из нескольких вариантов предложенных	Прочитайте текст, выберите правильный ответ	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные вариант-ты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа (например, 3 или В).	Ответ считается верным, если правильно указана цифра или буква
Задание закрытого типа на установление соответствия	Прочитайте текст и установите соответствие	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидаются пары элементов. 2. Внимательно прочитать оба списка: список 1 – вопросы, утверждения, факты, понятия и т.д.; список 2 – утверждения, свойства объектов и т.д. 3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов. 4. Записать попарно буквы и цифры (в зависимости от задания) вариантов ответа (например, А1 или Б4).	Ответ считается верным, если правильно указаны цифры или буквы

Задание закрытого типа с выбором нескольких правильных ответов из нескольких вариантов предложенных	Прочитайте текст, выберите правильные ответы	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько правильных ответов из предложенных вариантов. 2. Внимательно прочитать предложенные вариант-ты ответа. 3. Выбрать несколько правильных ответов. 4. Записать только номера (или буквы) выбранного варианта ответа (например, 1 4 или А Г).	Ответ считается верным, если правильно установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого)

Задание закрытого типа на установление последовательности	Прочитайте текст и установите последовательность	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается последовательность элементов. 2. Внимательно прочитать предложенные варианты ответа. 3. Построить верную последовательность из предложенных элементов. 4. Записать буквы/цифры (в зависимости от задания) вариантов ответа в нужной последовательности (например, БВА или 135).	Ответ считается верным, если правильно указана вся последовательность цифр
Задание комбинированного типа с выбором одного правильного ответа из предложенных и обоснованием выбора	Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать один верный ответ. 4. Записать только номер (или букву) выбранного варианта ответа. 5. Записать аргументы, обосновывающие выбор ответа (например, 4 текст обоснования).	Ответ считается верным, если правильно указана цифра или буква и приведены корректные аргументы, используемые при выборе ответа

Задание открытого типа с развернутым ответом	Прочитайте текст и запишите развернутый обоснованный ответ	1. Внимательно прочитать текст задания и понять суть вопроса. 2. Продумать логику и полноту ответа. 3. Записать ответ, используя четкие компактные формулировки. 4. В случае расчетной задачи, записать решение и ответ	Ответ считается верным: 1. Отсутствие фактических ошибок. 2. Раскрытие объема используемых понятий (полнота ответа). 3. Обоснованность ответа (наличие аргументов). 4. Логическая последовательность излагаемого материала.

4.5. Общая шкала оценивания результатов текущего контроля успеваемости и промежуточной аттестации обучающихся с применением БРС

Оценка по шкале ECTS	**Сумма баллов за все виды учебной деятельности**	**Оценка по государственной шкале**	**Определение**
A	90 – 100	«Отлично»	отличное выполнение с незначительным количеством неточностей
B	80 – 89	«Хорошо»	в целом правильно выполненная работа с незначительным количеством ошибок (до 10%)
C	75 – 79		в целом правильно выполненная работа с незначительным количеством ошибок (до 15%)
D	70 – 74	«Удовлетворительно»	неплохо, но со значительным количеством недостатков
E	60 – 69		выполнение удовлетворяет минимальные критерии
FX	35 – 59	«Не удовлетворительно»	с возможностью повторной сдачи
F	0 – 34		с обязательным повторным изучением дисциплины (выставляется комиссией)

Соотношение баллов за текущий контроль успеваемости и промежуточную аттестацию, а также повторную промежуточную аттестацию:

Максимальная сумма баллов за текущий контроль успеваемости	Максимальная сумма баллов за промежуточную аттестацию	Максимальная итоговая балльная оценка	Максимальная сумма баллов за повторную промежуточную аттестацию
100 баллов	100 баллов	100 баллов	100 баллов

5. Формы аттестации, типовые оценочные материалы для текущего контроля успеваемости обучающихся, критерии и шкалы оценивания по контрольным точкам

5.1. В ходе реализации дисциплины Б1.О.02.06 Базы данных используются следующие формы текущего контроля успеваемости обучающихся (в том числе, задания к контрольным точкам): *контрольные вопросы для проведения опроса*, *задания открытого типа на практических*

занятиях, контрольные задания

Таблица 5.1.

Распределение баллов по видам учебной деятельности (БРС)

Раздел/Темы	Формы текущего контроля		КТ
	УО	ПЗ	
Р-1. / Т-1	4	3	12
Р-1. / Т-2	4	3	
Р-1. / Т-3	4	3	
Р-2. / Т-4	4	3	12
Р-2. / Т-5	4	3	
Р-2. / Т-6	4	3	
Р-3. / Т-7	4	3	12
Р-3. / Т-8.	4	3	
Р-3. / Т-9	5	3	
Итого: 100 б	37	27	36
Р-4. / Т-10	4	3	12
Р-4. / Т-11	4	3	
Р-4. / Т-12	4	3	
Р-5. / Т-13	4	3	12
Р-5. / Т-14	4	3	
Р-5. / Т-15	4	3	
Р-6. / Т-16	4	3	12
Р-6. / Т-17	4	3	
Р-6. / Т-18	5	3	
Итого: 100 б	37	27	36

УО – устный опрос;
ТЗ – тестовое задание;
КЗ – контрольные задания;
ПЗ – практическое занятие;
Д – доклад;
КТ – контрольные точки.

Критерии оценивания опроса:

Балы	Описание критерия
4-5	Обучающийся полно излагает материал (отвечает на вопрос), дает правильное определение основных понятий; обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные; излагает материал последовательно и правильно с точки зрения норм литературного языка.
2-3	Обучающийся дает ответ, удовлетворяющий тем же требованиям, что и для оценки «отлично», но допускает 1–2 ошибки, которые сам же исправляет, и 1–2 недочета в последовательности и языковом оформлении излагаемого.
1	Обучающийся обнаруживает знание и понимание основных положений данной темы, но излагает материал неполно и допускает неточности в определении понятий или формулировке правил; не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.
0	Обучающийся обнаруживает незнание вопроса, допускает ошибки в формулировке определений и правил, искажающие их смысл,

	беспорядочно и неуверенно излагает материал.

0* - в журнал академической группы не выставляется

Критерии оценивания практических занятий:

Балы	Описание критерия	
3	Свыше 90% правильных ответов.	Обучающийся демонстрирует глубокое познание в освоенном материале.
2	Свыше 70% правильных ответов.	Обучающимся материал освоен полностью, без существенных ошибок.
1	Реализовано более 50% поставленных задач	Обучающимся материал освоен не полностью, имеются значительные пробелы в знаниях.
0	Реализовано менее 30% поставленных задач.	Обучающимся материал не освоен, знания обучающегося ниже базового уровня.

0* - в журнал академической группы не выставляется

Критерии оценивания контрольных заданий:

Балы	Описание критерия
10-12	Обучающимся задание выполнено без ошибок и в полном объеме.
7-9	Обучающимся в целом задание выполнено, имеются отдельные неточности или недостаточно полные ответы, не содержащие ошибок.
5-6	Обучающимся допущены отдельные ошибки при выполнении задания
0-4	У обучающегося отсутствуют ответы на большинство вопросов задачи, задание не выполнено или выполнено не верно.

0* - в журнал академической группы не выставляется

5.2. Типовые оценочные материалы для текущего контроля успеваемости обучающихся (вне контрольных точек):

РАЗДЕЛ 1. ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

Тема 1. Математические основы построения реляционных БД. Определение и терминология. Математические отношения. Ключи. Целостность.

Контрольные вопросы для проведения опроса:

1. Дайте определение сущности предметной области и приведите пример.
2. Чем математическое отношение отличается от обычной таблицы в терминах теории множеств?
3. Какими двумя обязательными свойствами обладает первичный ключ?
4. Объясните правило ссылочной целостности на примере таблиц «Заказы» и «Клиенты».

Практические занятия:

Задания открытого типа с развернутым ответом.

Для предметной области «Библиотека» (книги, читатели, выдаче книг) выполнить:

Выделить все сущности (минимум 3).

Для каждой сущности определить атрибуты.

Назначить первичный ключ для каждой таблицы.

Определить внешние ключи и указать, на какие таблицы они ссылаются.

Нарисовать ER-диаграмму (прямоугольники – сущности, ромбы – связи, обозначить 1:1, 1:M, M:N).

Для таблицы «Выдача» указать: какой атрибут является первичным ключом, какой – внешним, и почему внешний ключ не может ссылаться на несуществующую запись.

Тема 2. Разработка модели данных на основе нормализации. Цель нормализации. Избыточность данных и аномалии обработки. Нормальные формы 1НФ и 2НФ

Контрольные вопросы для проведения опроса:

1. Назовите три типа аномалий обработки данных и кратко охарактеризуйте каждый.
2. Какое требование предъявляет первая нормальная форма (1НФ) к значениям атрибутов?
3. Чем частичная функциональная зависимость отличается от полной?
4. Почему таблица с составным первичным ключом может быть в 1НФ, но не во 2НФ?

Практические занятия:

Задания открытого типа с развернутым ответом.

Дана ненормализованная таблица «Заказы»:

Номер_заказа	Дата	Клиент	Адрес	Товары (список через запятую)	Цены (список через запятую)

Привести таблицу к 1НФ (устранить повторяющиеся группы, обеспечить атомарность).

Полученную таблицу (уже в 1НФ) привести ко 2НФ, если она имеет составной первичный ключ.

В явном виде указать, какие аномалии (вставки, обновления, удаления) были устранены на каждом шаге.

Записать результирующий набор таблиц (схему БД) со связями.

Тема 3. 3 НФ, составной ключ

Контрольные вопросы для проведения опроса:

1. Сформулируйте определение третьей нормальной формы (3НФ).
2. Что такое транзитивная зависимость? Приведите пример.
3. В каких случаях применяется составной первичный ключ?
4. Можно ли считать таблицу «Сотрудники (ID_сотр, Имя, ID_отдела, Название_отдела, Адрес_отдела)» находящейся в 3НФ? Почему?

Практические занятия:

Задания открытого типа с развернутым ответом.

Дана таблица «Сотрудники» (уже во 2НФ):

| ID_сотр (ПК) | ФИО | ID_отдела | Название_отдела | Телефон_отдела |

Обнаружить транзитивную зависимость (неключевой атрибут зависит от другого неключевого).

Привести таблицу к 3НФ, выполнив декомпозицию (разделение на две таблицы).

Для таблицы-связки «Студенты_Дисциплины» составить составной первичный ключ. Объяснить, почему простой первичный ключ (суррогатный ID) здесь не всегда лучше.

Спроектировать БД в 3НФ для предметной области «Университет» (студенты, группы, факультеты, дисциплины, оценки) – минимум 4 таблицы с указанием первичных и внешних ключей.

РАЗДЕЛ 2. ОСНОВЫ ЯЗЫКА SQL

Тема 4. Синтаксис SQL-операторов CREATE TABLE, INSERT, DELETE, UPDATE

Контрольные вопросы для проведения опроса:

1. Какие ограничения целостности можно задать при создании таблицы через CREATE TABLE?
2. В чем разница между DELETE FROM table_name; и TRUNCATE TABLE table_name;?
3. Что произойдёт, если в операторе UPDATE опустить предложение WHERE?
4. Как добавить в таблицу новую строку, указав значения только для части столбцов?

Практические занятия:

Задания открытого типа с развернутым ответом.

Создать три таблицы: `Клиенты` (id_cli INT PRIMARY KEY, имя VARCHAR(50), email VARCHAR(100) UNIQUE), `Товары` (id_tov INT PRIMARY KEY, название VARCHAR(80), цена DECIMAL(10,2) CHECK (цена > 0)), `Заказы` (id_zak INT PRIMARY KEY, id_cli INT, id_tov INT, FOREIGN KEY ...).

Вставить в каждую таблицу по 3-5 тестовых строк через INSERT.

Выполнить UPDATE: увеличить цену всех товаров на 10%.

Выполнить DELETE: удалить заказы с id_cli = 2. Затем удалить клиента с id_cli = 2 (при наличии внешнего ключа – обратить внимание на ошибку или каскад).

Продемонстрировать разницу между DELETE без WHERE и TRUNCATE.

Тема 5. Синтаксис оператора SELECT. Выборка данных по критериям

Контрольные вопросы для проведения опроса:

1. В каком порядке логически выполняются основные предложения запроса `SELECT`?
2. Чем отличается оператор `LIKE '%a'` от `LIKE 'a%'`?
3. Каким образом проверить, что значение столбца равно `NULL`? Почему нельзя использовать `= NULL`?
4. Что вернёт запрос `SELECT * FROM Товары WHERE Цена BETWEEN 100 AND 200`?

Практические занятия:

Задания открытого типа с развернутым ответом.

На учебной БД (библиотека, магазин или университет) выполнить запросы:

Вывести все записи из таблицы «Книги».

Вывести названия книг, у которых цена больше 500.

Вывести книги, выпущенные в 2020–2023 годах (BETWEEN).

Вывести читателей с фамилией, начинающейся на «Ива» (LIKE).

Вывести заказы, у которых поле «дата_доставки» равно NULL.

Вывести уникальные города, в которых живут клиенты (DISTINCT).

Вывести первых 5 самых дорогих товаров (ORDER BY + LIMIT).

Тема 6. Оператор join. Левое и правое соединение. Full join

Контрольные вопросы для проведения опроса:

1. Что вернёт `INNER JOIN` двух таблиц, если в левой таблице есть строки, не имеющие соответствия в правой?
2. В каком сценарии практического применения `LEFT JOIN` будет незаменим?
3. Чем результат `LEFT JOIN` отличается от результата `RIGHT JOIN` при перестановке таблиц?
4. Что такое декартово произведение (CROSS JOIN) и когда оно возникает?

Практические занятия:

Задания открытого типа с развернутым ответом.

Имеются таблицы: Клиенты(id, имя) и Заказы(id, id_клиента, сумма).

Написать INNER JOIN, который выведет все заказы с именами клиентов.

Написать LEFT JOIN, который выведет всех клиентов и их заказы, включая тех, у кого заказов нет.

Написать RIGHT JOIN – убедиться, что результат совпадает с LEFT JOIN при перестановке таблиц.

Добавить третью таблицу Товары(id, название, id_заказа) и выполнить двойной JOIN (Клиенты → Заказы → Товары).

Смоделировать ситуацию: найти клиентов, у которых нет ни одного заказа, через LEFT JOIN с проверкой NULL.

РАЗДЕЛ 3. СЛОЖНЫЕ ТИПЫ ЗАПРОСОВ

Тема 7. Групповые операции в запросах, сортировка, вычисления в запросах

Контрольные вопросы для проведения опроса:

1. Какие пять агрегатных функций вы знаете и что каждая из них возвращает?
2. Почему WHERE нельзя использовать для фильтрации групп, и чем его заменяют?

3. Что произойдёт, если в SELECT указать столбец, не входящий ни в агрегатную функцию, ни в GROUP BY?

4. В каком порядке должны следовать предложения GROUP BY, HAVING, ORDER BY в запросе?

5. Чем отличается COUNT(*) от COUNT(имя_столбца)?

Практические занятия:

Задания открытого типа с развернутым ответом.

На таблице Продажи(id, товар, категория, количество, цена, дата):

Подсчитать общее количество проданных товаров (SUM).

Вывести среднюю цену по каждой категории (AVG + GROUP BY).

Найти максимальную и минимальную цену товара.

Вывести категории, у которых суммарное количество продаж больше 100 (HAVING).

Подсчитать количество заказов по каждому месяцу (с использованием MONTH(дата)).

Вычислить для каждой строки общую сумму = количество * цена (вычисляемый столбец).

Отсортировать результат по убыванию общей суммы.

Тема 8. Использование подзапросов. Встроенные функции (day(), month(), year() и др.)

Контрольные вопросы для проведения опроса:

1. Что такое скалярный подзапрос и где его можно использовать?

2. Напишите логическое выражение, которое извлечёт год из поля `Дата_рождения`.

3. Как с помощью подзапроса найти сотрудников, у которых зарплата выше средней?

4. Какая функция объединяет несколько строк в одну с разделителем (в MySQL)?

Практические занятия:

Задания открытого типа с развернутым ответом.

Найти сотрудников, у которых зарплата выше средней (подзапрос в WHERE).

Найти товары, которые ни разу не заказывались (подзапрос с NOT IN).

С помощью EXISTS найти клиентов, у которых есть хотя бы один заказ.

В SELECT вывести название товара и разницу между его ценой и средней ценой всех товаров (скалярный подзапрос).

Извлечь год, месяц и день из поля «дата_рождения» и вывести отдельными столбцами.

Вывести ФИО одной строкой через CONCAT().

Округлить цены до двух знаков после запятой (ROUND).

Тема 9. Создание и использование представлений

Контрольные вопросы для проведения опроса:

1. Чем представление (VIEW) отличается от обычной таблицы с точки зрения хранения данных?
2. Назовите две основные причины для создания представления в базе данных.
3. Почему представление, содержащее `GROUP BY`, обычно нельзя обновлять (INSERT/UPDATE)?
4. Какой командой удаляется представление?
5. Если изменить базовую таблицу, изменится ли автоматически представление, построенное на ней?

Практические занятия:

Задания открытого типа с развернутым ответом.

Создать представление `Активные_клиенты`, которое показывает клиентов, сделавших хотя бы один заказ за последние 30 дней.

Создать представление `Отчет_по_продажам`, объединяющее таблицы Заказы, Клиенты, Товары и включающее вычисляемое поле «итоговая_сумма».

Выполнить запрос к созданному представлению, как к обычной таблице.

Попытаться выполнить UPDATE через представление (если представление необновляемое – проанализировать ошибку).

Удалить одно из представлений (DROP VIEW).

РАЗДЕЛ 4. ПРОЦЕДУРЫ И ФУНКЦИИ В СУБД

Тема 10. Язык создания хранимых процедур и функций. Передача параметров в хранимые процедуры и функции

Контрольные вопросы для проведения опроса:

1. Назовите два ключевых отличия хранимой функции от хранимой

процедуры.

2. Какие три типа параметров могут быть у процедуры (`IN`, `OUT`, `INOUT`)? Что делает каждый?
3. Где можно использовать хранимую функцию в SQL-запросе, а где – процедуру?
4. Почему хранимые процедуры могут уменьшить сетевой трафик между приложением и сервером БД?
5. Какой синтаксис используется для вызова хранимой процедуры, а какой – для функции?

Практические занятия:

Задания открытого типа с развернутым ответом.

Создать хранимую функцию `Возраст(дата_рождения DATE) RETURNS INT`, которая возвращает полное количество лет.

Использовать эту функцию внутри SELECT.

Создать хранимую процедуру `Добавить_Клиента(IN имя VARCHAR, IN email VARCHAR)`, которая вставляет нового клиента.

Создать процедуру с OUT-параметром `Получить_Кол_Заказов(IN id_клиента INT, OUT кол_во INT)`, которая возвращает количество заказов клиента.

Вызвать процедуру и вывести значение OUT-параметра (в среде СУБД или через PHP).

Тема 11. Создание триггеров

Контрольные вопросы для проведения опроса:

1. Какие три события могут активировать триггер?
2. В чем разница между `BEFORE` и `AFTER` триггером?
3. Для чего нужны псевдотаблицы `OLD` и `NEW` в теле триггера?
4. Какой уровень срабатывания (`FOR EACH ROW` или `FOR EACH STATEMENT`) используется чаще и почему?
5. Может ли триггер вызвать другой триггер? Что такое каскадное срабатывание?

Практические занятия:

Задания открытого типа с развернутым ответом.

На таблице `Заказы`:

Создать триггер `BEFORE INSERT`, который автоматически устанавливает дату создания заказа = текущая дата (NOW()), если она не передана.

Создать триггер `AFTER DELETE`, который логирует удаленные заказы в таблицу `Лог_удалений` (id_заказа, дата_удаления, пользователь).

Создать триггер `BEFORE UPDATE`, который запрещает уменьшать цену товара более чем на 20% (выбросить ошибку).

Проверить срабатывание триггеров: вставить заказ без даты, удалить заказ, попытаться обновить цену.

Просмотреть список триггеров в базе данных и удалить один из них

Тема 12. Работа с транзакциями и типами блокировок

Контрольные вопросы для проведения опроса:

1. Расшифруйте аббревиатуру ACID и кратко объясните каждое свойство.
2. Что произойдёт с транзакцией, если после `BEGIN` произойдет сбой питания, но до `COMMIT`?
3. Какой уровень изоляции защищает от «грязного чтения», но допускает «неповторяемое чтение»?
4. Что такое deadlock (взаимная блокировка) и как СУБД обычно её разрешает?
5. Какую команду нужно выполнить, чтобы отменить все изменения с начала транзакции?

Практические занятия:

Задания открытого типа с развернутым ответом.

Открыть транзакцию (BEGIN), выполнить INSERT в таблицу, затем выполнить ROLLBACK – убедиться, что данные не сохранились.

Открыть транзакцию, выполнить UPDATE, затем COMMIT – убедиться, что изменения применились.

Смоделировать ситуацию «грязного чтения»: в первой сессии выполнить UPDATE без COMMIT, во второй сессии (с уровнем изоляции READ UNCOMMITTED) прочитать изменённые, но ещё не зафиксированные данные.

Смоделировать deadlock: две сессии, каждая обновляет две таблицы в разном порядке, добиться взаимоблокировки и посмотреть, как СУБД разрешает ситуацию.

Использовать SAVEPOINT внутри транзакции и выполнить ROLLBACK TO SAVEPOINT.

РАЗДЕЛ 5. ОСНОВЫ ВЕБ-ТЕХНОЛОГИЙ (Apache + PHP +

MySQL)

Тема 13. Определение, основные понятия: веб-сервер, серверный язык программирования (Apache, PHP, MySQL)

Контрольные вопросы для проведения опроса:

1. Какую роль выполняет Apache в связке Apache + PHP + MySQL?
2. Где исполняется PHP-код (на сервере или в браузере пользователя) и видит ли пользователь исходный код PHP?
3. Каким образом PHP-скрипт передаёт SQL-запрос в MySQL и получает результат?
4. Что произойдёт, если в браузер обратиться к файлу `script.php`, а веб-сервер не настроен на обработку PHP (отдаст как текстовый файл)?
5. Полный цикл обработки запроса: браузер → Apache → PHP → MySQL → PHP → Apache → браузер. Объясните, что происходит на каждом шаге.

Практические занятия:

Задания открытого типа с развернутым ответом.

Запустить локальный веб-сервер (OpenServer / XAMPP) с активными сервисами Apache и MySQL.

В корневой директории DocumentRoot создать файл `info.php` с кодом `<?php phpinfo(); ?>`.

Открыть в браузере `http://localhost/info.php` и найти в выводе разделы: Apache, mysqli, mysqlnd.

Создать файл `test_db.php` с кодом подключения к MySQL (mysqli_connect) и вывести версию сервера (`SELECT VERSION()`).

Нарисовать схему полного цикла обработки запроса: браузер → Apache → PHP → MySQL → PHP → Apache → браузер, подписав протоколы и интерфейсы.

Тема 14. Инсталляция Apache+PHP+MySql на различные операционные системы

Контрольные вопросы для проведения опроса:

1. Назовите две готовые сборки (веб-стека) для Windows, включающие Apache, PHP и MySQL.
2. Как проверить, что Apache и MySQL успешно запущены после установки XAMPP?
3. Какой файл в конфигурации PHP (`php.ini`) нужно изменить для увеличения максимального размера загружаемого файла?

4. Что такое DocumentRoot и где он обычно располагается в OpenServer или XAMPP?
5. Для чего используется phpMyAdmin в составе веб-стека?

Практические занятия:

Задания открытого типа с развернутым ответом.

Установить OpenServer или XAMPP на учебный компьютер (или в виртуальную машину).

Запустить панель управления, проверить зелёные индикаторы Apache и MySQL.

Настроить виртуальный хост с именем `mysite.local`, корневая директория – `C:/openserver/domains/mysite.local` (или аналог).

В файле `php.ini` включить отображение всех ошибок (`display_errors = On`, `error_reporting = E_ALL`).

Зайти в phpMyAdmin и создать новую базу данных `test_db` и пользователя `test_user` с паролем.

Прописать в hosts-файл запись `127.0.0.1 mysite.local` и открыть проект в браузере.

Тема 15. Введение в язык PHP

Контрольные вопросы для проведения опроса:

1. Какой синтаксис используется для встраивания PHP-кода в HTML-страницу?
2. Каким символом начинаются имена всех переменных в PHP?
3. Назовите суперглобальный массив, который содержит данные, переданные через метод POST.
4. Какой суперглобальный массив содержит параметры из адресной строки (после знака `?`)?
5. Что выведет код: `echo "5" + "10" . " рублей";`? Почему?

Практические занятия:

Задания открытого типа с развернутым ответом.

Написать скрипт, который выводит "Привет, мир!" и текущую дату в формате "ДД.ММ.ГГГГ".

Создать страницу с формой (метод GET), передающей имя пользователя. Вывести "Привет, {имя}".

Создать страницу с формой (метод POST) для ввода двух чисел и вывести их сумму.

Объявить массив из 5 названий городов и вывести его в виде

маркированного списка через цикл foreach.

Написать функцию `isAdult($age)`, которая возвращает true, если возраст >= 18, и false в противном случае. Использовать её в условии.

РАЗДЕЛ 6. РАБОТА С БАЗАМИ ДАННЫХ ИЗ ЯЗЫКА PHP

Тема 16. Подключение к базе. Получение данных из базы; Передача параметров через адресную строку. Метод GET

Контрольные вопросы для проведения опроса:

1. Какое расширение PHP (назовите хотя бы одно) предназначено для работы с MySQL?
2. В чем заключается опасность SQL-инъекции при использовании метода GET?
3. Какой функцией PHP нужно экранировать данные перед подстановкой в SQL-запрос (в процедурном mysqli)?
4. Как получить одну строку результата в виде ассоциативного массива после выполнения `mysqli_query()`?
5. Где в URL-адресе находятся параметры, переданные методом GET, и как они отделяются от адреса страницы?

Практические занятия:

Задания открытого типа с развернутым ответом.

Создать базу данных `university` и таблицу `students` (id, name, group_id).

Написать PHP-скрипт `list.php`, который подключается к БД и выводит всех студентов в виде HTML-таблицы.

Добавить в таблицу ссылку «Подробнее» для каждого студента: `student.php?id=XXX`.

Написать скрипт `student.php`, который принимает GET-параметр `id`, выполняет `SELECT * FROM students WHERE id = $id` и выводит информацию о студенте.

Продемонстрировать защиту от SQL-инъекции: ввести в адресную строку `student.php?id=1 OR 1=1` и показать разницу между небезопасным кодом и кодом с экранированием (mysqli_real_escape_string) или подготовленными выражениями.

Тема 17. Добавление данных в базу. Форма регистрации пользователей. Метод POST

Контрольные вопросы для проведения опроса:

1. В каком массиве PHP оказываются данные, отправленные из

HTML-формы с `method="POST"`?

2. Почему пароль пользователя нужно хранить не в открытом виде, а с использованием `password_hash()`?

3. Зачем выполнять `header('Location: ...')` после успешной вставки данных в БД?

4. Какие проверки (валидацию) необходимо выполнить перед вставкой данных регистрации пользователя?

5. В чем отличие передачи данных методом POST от GET с точки зрения безопасности и видимости?

Практические занятия:

Задания открытого типа с развернутым ответом.

Создать таблицу `users` (id, login, email, password_hash, created_at).

Создать HTML-форму регистрации (логин, email, пароль, подтверждение пароля) с `method="POST"`.

Написать обработчик `register.php`, который:

Проверяет, что все поля заполнены.

Проверяет уникальность логина и email (SELECT перед INSERT).

Проверяет, что пароль и подтверждение совпадают, а длина пароля >= 6 символов.

Хэширует пароль через `password_hash()`.

Вставляет запись в таблицу.

При успехе выполняет редирект на страницу приветствия, при ошибке – выводит сообщение.

Создать страницу логина (form с полями login/пароль) и обработчик, проверяющий пароль через `password_verify()`.

После успешного логина сохранять `$_SESSION['user_id']` и выводить «Добро пожаловать, {логин}»

Тема 18. Создание формы для редактирования данных

Контрольные вопросы для проведения опроса:

1. Зачем в форму редактирования данных добавляют скрытое поле `<input type="hidden" name="id">`?

2. Опишите два этапа (два HTTP-запроса) при редактировании записи: что происходит на первом и на втором?

3. Как предварительно заполнить поля формы существующими значениями из базы данных?

4. Какой SQL-оператор используется для сохранения изменённых данных после редактирования?

5. Почему при редактировании нужно проверять, имеет ли текущий пользователь право изменять именно эту запись?

Практические занятия:

Задания открытого типа с развернутым ответом.

Создать страницу `profile.php`, на которой авторизованный пользователь видит свои данные (логин, email).

Добавить на страницу ссылку «Редактировать профиль» → `edit_profile.php?id=XXX`.

Написать `edit_profile.php`, который:

Получает ID из GET-параметра.

Выполняет SELECT и предварительно заполняет поля формы (значения в `value`).

Содержит скрытое поле `<input type="hidden" name="id" value="<?= $id ?>">`.

Написать обработчик `update_profile.php`, который:

Принимает POST-данные (id, login, email, новый_пароль (опционально)).

Проверяет, что логин/email не заняты другими пользователями (кроме текущего).

Формирует UPDATE: если поле пароля не пустое – обновляет и пароль (хэшируя), иначе только login/email.

Выполняет редирект обратно на `profile.php` с сообщением об успехе.

Продемонстрировать попытку редактирования чужого профиля через подмену ID в скрытом поле – реализовать проверку прав (`$_SESSION['user_id'] == $id_из_формы`).

5.3. Один или несколько тематических блоков дисциплины завершаются контрольной точки по разделу (далее – КТ). Текущий контроль успеваемости по дисциплине предусматривает не менее 2 (двух) и не более 10 (десяти) КТ в течение периода освоения дисциплины.

Максимальное количество баллов за любой тип работ в рамках КТ составляет 100 (сто) баллов.

Распределение весовых коэффициентов по КТ в рамках текущего контроля успеваемости по дисциплине и формулы расчета:

Наименование контрольной работы	Максимальное количество баллов за работу в рамках КР, которое может набрать студент	Коэффициент веса контрольной работы	Результат контрольной работы, участвующий в формировании итоговой балльной оценки по дисциплине
КТ 1	100	0,12	12
КТ 2	100	0,12	12
КТ 3	100	0,12	12
Итого:	x	0,36	36
КТ 4	100	0,12	12
КТ 5	100	0,12	12
КТ 6	100	0,12	12
Итого:	x	0,36	36

Формула расчета результата контрольной работы:

Результат контрольной работы = Количество баллов за точку в рамках КТ X Коэффициент веса контрольной точки.

5.4. Формы текущего контроля успеваемости обучающихся в рамках КТ и типовые оценочные материалы:

КТ 1

Задание 1. Проектирование ER-диаграммы

Для предметной области «Спортивный клуб»:

Тренеры (ФИО, стаж, специализация)

Группы (название, день недели, время начала)

Участники (ФИО, дата рождения, телефон)

Связи: тренер ведет одну или несколько групп, участник может

записаться в несколько групп, группа может состоять из нескольких участников.

Требуется:

Выделить все сущности и атрибуты.

Указать первичные ключи для каждой сущности.

Нарисовать ER-диаграмму (обозначить типы связей: 1:1, 1:M, M:N).

Для связи M:N предложить таблицу-связку и указать ее составной первичный ключ.

Задание 2. Нормализация таблицы

Дана таблица «Поставки» в ненормализованной форме:

Код_поставки	Поставщик	Город	Товар1	Кол-во1	Цена1	Товар2	Кол-во2
101	ООО «Мир»	Тула	Принтер	2	5000	Сканер	1
102	ЗАО «Техно»	Тула	Ноутбук	3	25000	NULL	NULL

Требуется:

Привести таблицу к 1НФ (устранить повторяющиеся группы, обеспечить атомарность).

Для полученной таблицы (в 1НФ) определить составной первичный ключ.

Привести таблицу ко 2НФ (устранить частичные зависимости). Показать результирующие таблицы.

Привести полученные таблицы к 3НФ (устранить транзитивные зависимости, если они есть).

Записать финальную схему БД (список таблиц с полями и связями).

Задание 3. Кейс-задача на ключи

В таблице «Оценки» есть поля: `ID_студента`, `ID_дисциплины`, `ID_преподавателя`, `Оценка`, `Дата_сдачи`. Известно, что один преподаватель ведет много дисциплин, но одна дисциплина ведется только одним преподавателем. Студент может получить несколько оценок по одной дисциплине в разные даты.

Какой первичный ключ (простой или составной) вы предложите для этой таблицы? Почему?

Какое поле (или поля) будет внешним ключом и на какую таблицу (таблицы) оно будет ссылаться?

Можно ли удалить преподавателя из справочника, если у него есть записи в таблице «Оценки»? Как это регулируется средствами SQL?

КТ 2

Задание 1. Написание DDL- и DML-запросов

Дана схема БД:

```sql
CREATE TABLE Клиенты (
id INT PRIMARY KEY,
имя VARCHAR(50),
город VARCHAR(50)
);

CREATE TABLE Заказы (
id INT PRIMARY KEY,
дата DATE,
id_клиента INT,
сумма DECIMAL(10,2),
FOREIGN KEY (id_клиента) REFERENCES Клиенты(id)
);
```

Напишите SQL-запросы:

Добавить нового клиента с `id=10`, именем «Иванов», городом «Москва».

Увеличить сумму всех заказов на 5% для клиента с `id=10`.

Удалить все заказы с датой ранее '2024-01-01'.

Создать таблицу «Товары» с полями: код (целое, первичный ключ), название (строка до 100 символов, NOT NULL), цена (число с двумя знаками, CHECK цена >= 0).

Задание 2. Запросы на выборку с критериями

На той же схеме БД (Клиенты, Заказы) напишите запросы:

Вывести всех клиентов из города «Тула».

Вывести заказы с суммой от 1000 до 5000 (включительно).

Вывести клиентов, у которых имя начинается на «С» и заканчивается на «в».

Вывести первые 5 самых дорогих заказов (поле `сумма`), отсортированных по убыванию.

Задание 3. Запросы с JOIN

Написать следующие запросы:

INNER JOIN: Вывести список всех заказов с именем клиента и городом.

LEFT JOIN: Вывести всех клиентов и их заказы, включая клиентов, у которых нет заказов. Для клиентов без заказов в поле `сумма` должно быть NULL.

Соединение трех таблиц: Добавьте третью таблицу `Сотрудники(id, имя, id_отдела, зарплата)` и таблицу `Отделы(id, название)`. Напишите запрос, который выводит имя сотрудника, название его отдела и зарплату.

Анализ результатов: Объясните, сколько строк вернет `INNER JOIN` между таблицами Клиенты (5 записей) и Заказы (8 записей), если:

У всех 5 клиентов есть хотя бы один заказ.

У 2 клиентов нет заказов.

КТ 3

Задание 1. Группировка и агрегация

Дана таблица «Продажи»:

id	товар	категория	количество	цена

Напишите запросы:

Подсчитать общее количество проданных товаров по каждой категории.

Найти среднюю цену товара (уникальные товары, а не продажи) – подумайте, как исключить дублирование товаров.

Вывести категории, у которых суммарная выручка (количество * цена) больше 10000.

Для каждого месяца (из поля `дата`) вывести количество продаж и общую выручку. Использовать `MONTH(дата)`.

Задание 2. Подзапросы и встроенные функции

Написать запросы:

Найти сотрудников, у которых зарплата выше средней по всем сотрудникам (подзапрос в `WHERE`).

Вывести товары, которые ни разу не были проданы (подзапрос с `NOT IN`).

С помощью `EXISTS` найти клиентов, которые сделали хотя бы один заказ (использовать таблицы `Клиенты` и `Заказы`).

Используя встроенную функцию, извлечь из поля `дата_рождения` год и вывести сотрудников, родившихся после 2000 года.

Задание 3. Представления

Создать представление `Техника`, которое показывает все товары из категории «Электроника» с ценой > 5000.

Создать представление `Активные_клиенты`, объединяющее таблицы `Клиенты` и `Заказы` и показывающее только тех клиентов, у которых общая сумма заказов > 5000.

Объясните, можно ли обновить данные через представление `Техника` (вставить новый товар)? Почему?

Напишите команду удаления представления `Активные_клиенты`

КТ 4

Задание 1. Хранимые процедуры и функции

Написать хранимую функцию `Отобразить_скидку(сумма DECIMAL) RETURNS DECIMAL`, которая возвращает:

0% скидка при сумме < 1000

5% скидка при сумме от 1000 до 5000

10% скидка при сумме > 5000

Написать хранимую процедуру `Увеличить_цену(процент INT)`, которая увеличивает цену всех товаров на заданный процент.

Написать процедуру с `OUT`-параметром `Посчитать_клиентов(OUT кол_во INT)`, которая возвращает количество клиентов в таблице.

Как вызвать процедуру и как вызвать функцию? Напишите синтаксис вызова.

Задание 2. Триггеры

На таблице «Заказы» (поля: id, дата, сумма, статус):

Написать триггер `BEFORE INSERT`, который автоматически устанавливает текущую дату в поле `дата`, если оно не указано.

Написать триггер `AFTER UPDATE`, который при изменении статуса заказа на «Отменен» записывает в таблицу `Лог_отмен` (id_заказа, дата_отмены, старая_сумма) – используя `OLD`.

Написать триггер `BEFORE UPDATE`, который запрещает уменьшать сумму заказа более чем на 50%. При нарушении генерировать ошибку с текстом «Слишком большое уменьшение суммы».

Вопрос: Может ли триггер вызвать другой триггер? Если да, то как этого избежать?

Задание 3. Транзакции и блокировки

Написать последовательность команд для выполнения операции перевода 500 рублей со счета №1 на счет №2 в одной транзакции (с проверкой, что на счете №1 достаточно средств). Показать `BEGIN`, `UPDATE`, `COMMIT` или `ROLLBACK`.

Объяснить, что произойдет, если между двумя `UPDATE` в транзакции произойдет сбой питания до `COMMIT`.

Что такое «грязное чтение»? Какой уровень изоляции его предотвращает?

Описать ситуацию взаимоблокировки (deadlock) на двух таблицах. Какая транзакция будет отменена СУБД?

КТ 5

Задание 1. Инсталляция и настройка

Ответьте на вопросы (без фактической установки, но с описанием действий):

Как проверить, что Apache и MySQL успешно запущены после установки OpenServer/XAMPP?

Какой файл нужно отредактировать, чтобы включить отображение всех ошибок PHP?

Как через phpMyAdmin создать новую базу данных и пользователя?

Задание 2. Написание базовых PHP-скриптов

Напишите код PHP для следующих задач (каждая задача – отдельный скрипт):

Скрипт выводит «Здравствуйте, Гость!». Если передан параметр `name` методом GET, то выводит «Здравствуйте, {значение name}!».

Скрипт обрабатывает форму с двумя числами (переданы методом POST) и выводит их сумму и произведение.

Объявите массив $colors = ['красный', 'зеленый', 'синий']. С помощью цикла foreach выведите его в виде нумерованного списка (`<ol>`).

Напишите функцию `isEven($n)`, которая возвращает true, если число четное. Используйте ее для проверки числа, переданного через GET.

Задание 3. Исправление ошибок

Ниже приведен PHP-код с ошибками. Найдите **3 ошибки** (синтаксические или логические) и исправьте их:

```
php
<?php
name = $_GET['name'];
echo "Привет, . $name;
if ($name = 'Admin') {
print "Добро пожаловать в админку!";
}
$arr = array(1, 2, 3)
foreach($arr as $value)
echo $value;
?>
```

Перечислите найденные ошибки и напишите исправленный код.

КТ 6

Задание 1. Подключение к БД и получение данных

Напишите PHP-код, который:

Подключается к MySQL (сервер localhost, пользователь root, пароль – пустой, база данных `university`).

Выполняет запрос `SELECT * FROM students`.

Выводит всех студентов в виде HTML-таблицы (столбцы: id, имя, группа).

Для каждого студента добавляет ссылку «Редактировать»: `edit.php?id=XXX`.

Реализует защиту от SQL-инъекций (используйте подготовленные выражения или `mysqli_real_escape_string`).

Задание 2. Регистрация пользователей (метод POST)

Напишите обработчик `register.php`, который:

Принимает методом POST поля: `login`, `email`, `password`, `confirm_password`.

Проверяет:

Все поля заполнены.

`password` и `confirm_password` совпадают.

Длина пароля не менее 6 символов.

Логин и email уникальны (выполните SELECT перед INSERT).

Если проверки пройдены – хэширует пароль через `password_hash()` и вставляет запись в таблицу `users`.

При успехе выполняет редирект на `login.php` с сообщением «Регистрация успешна».

При ошибке выводит понятное сообщение пользователю.

Задание 3. Редактирование данных

Дана страница `edit_user.php?id=5`, которая выводит форму редактирования пользователя (логин, email, пароль – опционально):

Напишите код, который загружает текущие данные пользователя из БД по `id`, полученному через `$_GET`, и подставляет их в атрибуты `value` полей формы.

Форма должна содержать скрытое поле с `id` пользователя.

Обработчик `update_user.php` должен:

Принять `id`, `login`, `email`, `new_password` (может быть пустым).

Проверить, что новый логин и email не заняты другими пользователями (исключая текущего `id`).

Если `new_password` не пустой – обновить и пароль (хэшируя), иначе – обновить только логин и email.

После обновления выполнить редирект на страницу списка пользователей.

Добавьте проверку прав: разрешить редактирование только если `$_SESSION['user_id'] == $id` (упрощенно – без полной авторизации, просто смоделируйте проверку).

6. Формы промежуточной аттестации, критерии и шкала оценивания, типовые оценочные материалы по дисциплине

6.1. Промежуточная аттестация проводится в форме *зачета* в третьем семестре и *экзамена* в четвертом семестре в письменной форме. Обучающийся получает экзаменационный билет с вариантами заданий. Обучающийся получает чистые маркированные листы бумаги для записей, затем приступает к выполнению. Необходимо дать ответ в письменном виде, подробно изложив ход мыслей.

6.2. Типовые оценочные материалы промежуточной аттестации.

РАЗДЕЛ 1. ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

Тема 1. Математические основы построения реляционных БД. Определение и терминология. Математические отношения. Ключи. Целостность

Экзаменационные вопросы:

1. Дайте определение реляционной модели данных. Каковы её основные математические предпосылки?
2. Что такое домен, кортеж, отношение? Как эти понятия соотносятся с табличной структурой БД?
3. Что такое первичный ключ? Какими свойствами он обладает?
4. Что такое внешний ключ? Как он обеспечивает ссылочную целостность?
5. Сформулируйте два основных правила целостности в реляционных БД (целостность сущностей и ссылочная целостность).

Экзаменационное практическое задание:

Дано описание предметной области: «Книжный магазин». Книги имеют название, автора, жанр, цену. Покупатели имеют имя, телефон, email. Заказы содержат дату заказа и список купленных книг с количеством.
Задание: Выделите сущности, атрибуты, первичные ключи. Нарисуйте ER-диаграмму. Укажите внешние ключи.

Тема 2. Разработка модели данных на основе нормализации. Цель нормализации. Избыточность данных и аномалии обработки. Нормальные формы 1НФ и 2НФ

Экзаменационные вопросы:

6. Сформулируйте цель нормализации. Какие проблемы решает нормализация?

7. Что такое избыточность данных? Приведите пример.

8. Назовите и кратко охарактеризуйте три типа аномалий обработки данных.

9. Дайте определение первой нормальной формы (1НФ). Какое требование предъявляется к атрибутам?

10. Дайте определение второй нормальной формы (2НФ). Какое требование связано с составным первичным ключом?

Экзаменационное практическое задание:

Дана ненормализованная таблица «Студенты_Курсы»: (ID_Студента, ФИО, Курс1, Оценка1, Курс2, Оценка2).
Задание: Приведите таблицу к 1НФ, затем ко 2НФ. Покажите все промежуточные и конечные таблицы.

Тема 3. 3 НФ, составной ключ

Экзаменационные вопросы:

11. Дайте определение третьей нормальной формы (3НФ).

12. Что такое транзитивная функциональная зависимость? Приведите пример.

13. В каких случаях применяется составной первичный ключ? Приведите пример из практики.

14. Чем отличается суррогатный (искусственный) первичный ключ от естественного? Назовите плюсы и минусы.

15. Можно ли считать таблицу, находящуюся в 3НФ, полностью свободной от избыточности? Почему?

Экзаменационное практическое задание:

Дана таблица «Сотрудники» (во 2НФ): (ID_Сотр, ФИО, ID_Отдела, Название_Отдела, Телефон_Отдела).
Задание: Определите транзитивную зависимость. Приведите таблицу к 3НФ. Запишите финальную схему.

РАЗДЕЛ 2. ОСНОВЫ ЯЗЫКА SQL

Тема 4. Синтаксис SQL-операторов CREATE TABLE, INSERT, DELETE, UPDATE

Экзаменационные вопросы:

16. Какие типы данных SQL вы знаете? Для чего используются VARCHAR, INT, DATE, DECIMAL?

17. Какие ограничения целостности (constraints) можно задать при создании таблицы через CREATE TABLE?

18. В чем разница между DELETE FROM table и TRUNCATE TABLE table?

19. Какой синтаксис оператора UPDATE? Что произойдет, если забыть WHERE?

20. Как вставить данные в таблицу, указав значения только для части столбцов? Приведите пример.

Экзаменационное практическое задание:

Напишите SQL-запросы:

Создать таблицу `Товары` (id INT PRIMARY KEY, название VARCHAR(100) NOT NULL, цена DECIMAL(10,2) CHECK (цена > 0)).

Вставить три товара: «Молоко», 80, «Хлеб», 40, «Сыр», 350.

Увеличить цену всех товаров на 10%.

Удалить товары дешевле 50.

Тема 5. Синтаксис оператора SELECT. Выборка данных по критериям

Экзаменационные вопросы:

21. Напишите полный синтаксис оператора SELECT (основные предложения в правильном порядке).

22. Как проверить, что значение столбца равно NULL? Почему нельзя использовать = NULL?

23. Для чего нужен оператор LIKE? Какие символы-шаблоны используются?

24. Что делает оператор IN? В чем его отличие от цепочки OR?

25. Как отсортировать результат запроса по убыванию и ограничить количество выводимых строк?

Экзаменационное практическое задание:

Дана таблица `Сотрудники` (id, имя, должность, оклад, дата_приема).

Задание: Напишите запросы:

Вывести всех сотрудников с окладом больше 50000.

Вывести сотрудников, принятых на работу в 2023 году.

Вывести сотрудников, чья фамилия начинается на «Ива».

Вывести трех самых высокооплачиваемых сотрудников.

Тема 6. Оператор join. Левое и правое соединение. Full join

Экзаменационные вопросы:

26. Что такое INNER JOIN? Что произойдет, если условие соединения не выполнено?
27. Чем LEFT JOIN отличается от INNER JOIN?
28. В каком случае используется RIGHT JOIN? Как его можно заменить на LEFT JOIN?
29. Что такое FULL OUTER JOIN? В каких СУБД он поддерживается?
30. Что получится, если указать JOIN без условия ON (перекрестное соединение)?

Экзаменационное практическое задание:

Даны таблицы: `Клиенты(id, имя)` и `Заказы(id, id_клиента, сумма)`.

Задание: Напишите запросы:

Вывести все заказы с именами клиентов.

Вывести всех клиентов и их заказы (включая клиентов без заказов).

Найти клиентов, у которых нет ни одного заказа (используя LEFT JOIN и проверку NULL).

РАЗДЕЛ 3. СЛОЖНЫЕ ТИПЫ ЗАПРОСОВ

Тема 7. Групповые операции в запросах, сортировка, вычисления в запросах

Экзаменационные вопросы:

31. Назовите пять основных агрегатных функций SQL и что каждая из них возвращает.
32. В чем разница между WHERE и HAVING? Приведите пример, когда WHERE не подходит.
33. Какое правило нужно соблюдать при использовании GROUP BY относительно столбцов в SELECT?
34. Как выполнить вычисления внутри SELECT (например, цена * количество)? Как назвать такой столбец?
35. В каком порядке должны следовать предложения SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY?

Экзаменационное практическое задание:

Дана таблица `Продажи` (товар, категория, количество, цена).

Задание: Напишите запросы:

Вывести общее количество проданных товаров по каждой категории.

Найти максимальную цену товара в каждой категории.

Вывести категории, у которых суммарная выручка (количество * цена)

больше 10000.

Вывести среднюю цену товаров (без учета количества, а по уникальным товарам).

Тема 8. Использование подзапросов. Встроенные функции (day(), month(), year() и др.)

Экзаменационные вопросы:

36. Что такое подзапрос? Какие типы подзапросов существуют (по возвращаемому значению)?
37. В чем разница между подзапросом с IN и подзапросом с EXISTS?
38. Для чего используются функции YEAR(), MONTH(), DAY()? Приведите пример.
39. Назовите три строковые функции SQL и объясните, что они делают.
40. Какой функцией можно округлить число до двух знаков после запятой?

Экзаменационное практическое задание:

Напишите запросы:

Найти сотрудников с зарплатой выше средней по всем сотрудникам (подзапрос).

Найти отделы, в которых работает хотя бы один сотрудник с зарплатой > 70000 (подзапрос с EXISTS).

Вывести имя сотрудника и разницу между его зарплатой и средней зарплатой по его отделу (скалярный подзапрос).

Из поля `дата_рождения` (DATE) извлечь год и вывести сотрудников, родившихся после 2000 года.

Тема 9. Создание и использование представлений

Экзаменационные вопросы:

41. Что такое представление (VIEW) и как оно физически хранится в БД?
42. Назовите три причины использовать представления в проектировании БД.
43. Какие ограничения существуют для обновляемых представлений (через INSERT/UPDATE)?
44. Какой командой создается представление? Как оно удаляется?
45. Если изменить базовую таблицу, изменится ли представление автоматически?

Экзаменационное практическое задание:

Напишите на псевдокоде или реальном SQL (в зависимости от СУБД):

Хранимую функцию `Скидка(сумма DECIMAL) RETURNS DECIMAL`: если сумма > 5000, вернуть сумму * 0.9, иначе вернуть сумму.

Хранимую процедуру `Удалить_Товар(IN id_товара INT)`, которая удаляет товар по id.

Хранимую процедуру с OUT-параметром `Кол_Товаров_В_Категории(IN категория VARCHAR, OUT кол_во INT)`.

РАЗДЕЛ 4. ПРОЦЕДУРЫ И ФУНКЦИИ В СУБД

Тема 10. Язык создания хранимых процедур и функций. Передача параметров в хранимые процедуры и функции

Экзаменационные вопросы:

46. Что такое хранимая процедура? Чем она отличается от обычного SQL-скрипта?
47. Назовите два отличия хранимой функции от хранимой процедуры.
48. Какие типы параметров бывают у хранимой процедуры (IN, OUT, INOUT)? Объясните каждый.
49. Как вызвать хранимую процедуру? Как вызвать хранимую функцию?
50. В чем преимущество использования хранимых процедур с точки зрения сетевого трафика?

Экзаменационное практическое задание:

Напишите триггеры:

Триггер `BEFORE INSERT` на таблицу `Заказы`, который автоматически устанавливает дату заказа = CURRENT_DATE, если поле `дата` не заполнено.

Триггер `AFTER DELETE` на таблицу `Сотрудники`, который записывает удаленного сотрудника в таблицу `Архив_Сотрудников`.

Триггер `BEFORE UPDATE` на таблицу `Цены`, который запрещает снижать цену более чем на 30% (выдать ошибку).

Тема 11. Создание триггеров

Экзаменационные вопросы:

51. Что такое триггер? Какие события могут активировать триггер?

52. В чем разница между BEFORE и AFTER триггерами?

53. Что такое FOR EACH ROW? Для чего он нужен?

54. Какие псевдотаблицы OLD и NEW доступны в триггерах INSERT, UPDATE, DELETE?

55. Может ли триггер вызвать другой триггер? Как это называется и как избежать бесконечной рекурсии?

Экзаменационное практическое задание:

Напишите последовательность команд:

Открыть транзакцию, перевести 1000 рублей со счета №1 на счет №2, затем подтвердить транзакцию.

Открыть транзакцию, выполнить два UPDATE, затем откатить (ROLLBACK).

Используя SAVEPOINT, записать: BEGIN → INSERT → SAVEPOINT sp1 → INSERT → ROLLBACK TO SAVEPOINT sp1 → COMMIT. Объясните, сколько записей останется в таблице.

Объясните, что такое «фантомное чтение» и какой уровень изоляции его предотвращает.

Тема 12. Работа с транзакциями и типами блокировок

Экзаменационные вопросы:

56. Расшифруйте аббревиатуру ACID. Кратко объясните каждое свойство.

57. Какие команды используются для управления транзакциями в SQL?

58. Что такое «грязное чтение»? Какой уровень изоляции его предотвращает?

59. Что такое deadlock (взаимная блокировка)? Как СУБД обычно её разрешает?

60. Какой уровень изоляции гарантирует максимальную целостность, но снижает производительность?

Экзаменационное практическое задание:

Опишите последовательность шагов (словами или схемой):

Пользователь открывает браузер и вводит http://mysite.ru/catalog.php?id=5.

Что происходит на Apache? Как запрос попадает к PHP?

Как PHP подключается к MySQL и выполняет запрос `SELECT * FROM goods WHERE id=5`?

Как формируется HTML-ответ и возвращается пользователю? Нарисуйте схему со стрелками между компонентами.

РАЗДЕЛ 5. ОСНОВЫ ВЕБ-ТЕХНОЛОГИЙ (APACHE + PHP + MYSQL)

Тема 13. Определение, основные понятия: веб-сервер, серверный язык программирования (Apache, PHP, MySQL)

Экзаменационные вопросы:

61. Какую функцию выполняет Apache в веб-приложении?
62. Где выполняется PHP-код и видит ли пользователь исходный код PHP?
63. Как PHP-скрипт отправляет запрос к MySQL и получает результат?
64. Опишите полный цикл обработки запроса: браузер → ... → браузер.
65. Чем отличается серверный язык программирования от клиентского (JavaScript)?

Экзаменационное практическое задание:

Тема 14. Инсталляция Apache+PHP+MySql на различные операционные системы

Экзаменационные вопросы:

66. Назовите две готовые сборки (веб-стека) для локальной разработки.
67. Как проверить, что Apache и MySQL запущены и работают корректно?
68. Где находится конфигурационный файл PHP (php.ini) и какие важные настройки там можно изменить?
69. Что такое виртуальный хост и для чего он нужен?
70. Как через phpMyAdmin создать новую базу данных и пользователя?

Экзаменационное практическое задание:

Опишите действия (без фактического выполнения, а в виде инструкции):

Как установить и запустить OpenServer или XAMPP на Windows?

Как создать виртуальный хост `myapp.local` и проверить его работу?

Как включить отображение всех ошибок PHP через `php.ini`?

Как создать базу данных `exam_db` и пользователя `exam_user` с паролем через phpMyAdmin?

Тема 15. Введение в язык PHP

Экзаменационные вопросы:

71. Как встраивается PHP-код в HTML-страницу?
72. С какого символа начинаются все переменные в PHP?
73. Назовите минимум три суперглобальных массива в PHP и объясните их назначение.
74. Какой оператор используется для конкатенации строк в PHP?
75. Как объявить функцию в PHP и вернуть из неё значение?

Экзаменационное практическое задание:

Напишите PHP-код:

Скрипт, который выводит «Сегодня {дата}» (используйте `date('d.m.Y')`).

Скрипт, который принимает через GET параметр `name` и выводит «Привет, {имя}». Если параметр не передан — «Привет, Гость».

Функцию `maxOfTwo($a, $b)`, которая возвращает большее из двух чисел. Используйте её для чисел 15 и 27.

Выведите массив `$fruits = ['яблоко', 'банан', 'апельсин']` в виде маркированного списка HTML через цикл.

РАЗДЕЛ 6. РАБОТА С БАЗАМИ ДАННЫХ ИЗ ЯЗЫКА PHP

Тема 16. Подключение к базе. Получение данных из базы; Передача параметров через адресную строку. Метод GET

Экзаменационные вопросы:

76. Как подключиться к MySQL из PHP с использованием mysqli?
77. В каком суперглобальном массиве PHP хранятся параметры, переданные через метод GET?
78. Что такое SQL-инъекция и как от нее защититься в PHP (назовите два способа)?
79. Как выполнить SQL-запрос SELECT из PHP и получить результат в виде ассоциативного массива?
80. Почему нельзя напрямую подставлять $_GET['id'] в SQL-запрос без обработки?

Экзаменационное практическое задание:

Напишите PHP-код:

Подключение к MySQL (сервер localhost, пользователь root, пароль пустой, база данных `test`).

Выполнение запроса `SELECT * FROM users`.

Вывод всех пользователей в HTML-таблицу (столбцы: id, name, email).

Защита от SQL-инъекций на примере: `$id = $_GET['id']; $sql = "SELECT * FROM users WHERE id = $id";` — перепишите этот фрагмент с использованием подготовленного выражения (prepared statement).

Тема 17. Добавление данных в базу. Форма регистрации пользователей. Метод POST

Экзаменационные вопросы:

81. В каком суперглобальном массиве PHP находятся данные, переданные методом POST?
82. Как хэшировать пароль в PHP и как проверить его при авторизации?
83. Зачем выполнять header('Location: ...') после успешной вставки данных?
84. Какие этапы валидации нужно пройти перед регистрацией нового пользователя?
85. В чем разница между GET и POST с точки зрения безопасности и видимости данных?

Экзаменационное практическое задание:

Напишите PHP-обработчик регистрации:

Принимает из POST-формы `login`, `email`, `password`.

Проверяет, что все поля заполнены, email корректен (используйте `filter_var`), пароль длиннее 6 символов.

Хэширует пароль через `password_hash()`.

Выполняет `INSERT` в таблицу `users`, предполагая, что соединение уже установлено.

При успехе — редирект на `login.php`, при ошибке — вывод сообщения.

Тема 18. Создание формы для редактирования данных

Экзаменационные вопросы:

86. Зачем в форму редактирования добавляют скрытое поле с id?
87. Как предварительно заполнить поля формы существующими значениями из БД?
88. Какой SQL-оператор используется для сохранения изменённых данных?
89. Почему при редактировании нужно проверять права пользователя (что он редактирует именно свои данные)?

90. Как обработать ситуацию, когда при редактировании другой пользователь уже удалил эту запись

Экзаменационное практическое задание:

Напишите PHP-код для редактирования профиля пользователя:

Получение `id` из `$_GET` и загрузка текущих данных из БД (SELECT).

Форма с полями `name`, `email`, скрытое поле с `id`.

Обработчик обновления (принимает POST): проверяет, что email не занят другим пользователем (кроме текущего id), формирует UPDATE.

Проверка прав: разрешить редактирование только если `$_SESSION['user_id'] == $id`.

После обновления — редирект на страницу профиля.

6.3. Критерии и шкала оценивания на основе БРС.

Соответствие государственной шкалы оценивания академической успеваемости и шкалы ECTS при экзамене

Оценка по шкале ECTS	**Сумма баллов за все виды учебной деятельности**	**Оценка по государственной шкале**	**Определение**
A	90 – 100	«Отлично»	отличное выполнение с незначительным количеством неточностей
B	80 – 89	«Хорошо»	в целом правильно выполненная работа с незначительным количеством ошибок (до 10%)
C	75 – 79		в целом правильно выполненная работа с незначительным количеством ошибок (до 15%)
D	70 – 74	«Удовлетворительно»	неплохо, но со значительным количеством недостатков
E	60 – 69		выполнение удовлетворяет минимальные критерии
FX	35 – 59	«Не удовлетворительно»	с возможностью повторной сдачи
F	0 – 34		с обязательным повторным изучением дисциплины (выставляется комиссией)

6.4. Описание дополнительных материалов и оборудования, необходимых для выполнения проверочных заданий

Компьютер с операционной системой RedOS, на котором установлены Apache, PHP, Mysql, phpMyAdmin, VSCode (или другой редактор).

7. Методические материалы по освоению дисциплины

Получение углубленных знаний по изучаемой дисциплине достигается за счет дополнительных часов к аудиторной работе самостоятельной работы студентов. Выделяемые часы целесообразно использовать для знакомства с дополнительной научной литературой по проблематике дисциплины, анализа научных концепций и современных подходов к осмыслению рассматриваемых проблем. К самостоятельному виду работы студентов относится работа в библиотеках, в электронных поисковых системах и т.п. по сбору материалов, необходимых для проведения практических занятий или выполнения конкретных заданий преподавателя по изучаемым темам. Студенты могут установить диалог с преподавателем, получать консультации по выполнению заданий. В качестве оценочных средств на протяжении семестра используются тестовые и иные задания.

Обучение по дисциплине «Базы данных» предполагает изучение курса на аудиторных занятиях (лекции, практические занятия) и самостоятельную работу студентов. Практические занятия дисциплины предполагают их проведение в различных формах с целью выявления полученных знаний, умений, навыков и компетенций с проведением контрольных мероприятий. С целью обеспечения успешного обучения студент должен готовиться к лекции, поскольку она является важнейшей формой организации учебного процесса, поскольку:

– знакомит с новым учебным материалом;
– разъясняет учебные элементы, трудные для понимания;
– систематизирует учебный материал;
– ориентирует в учебном процессе.

Работа обучающегося на лекции:

Слушание и запись лекций – сложный вид вузовской аудиторной работы. Внимательное слушание и конспектирование лекций предполагает интенсивную умственную деятельность обучающегося. Краткие записи лекций, их конспектирование помогает усвоить учебный материал. Конспект является полезным тогда, когда записано самое существенное, основное и сделано это самим обучающимся.

Подготовка к практическим занятиям:

Подготовку к каждому практическому занятию каждый обучающийся должен начать с ознакомления с планом, который отражает содержание предложенной темы. Тщательное продумывание и изучение вопросов плана основывается на проработке текущего материала лекции, а затем изучения обязательной и дополнительной литературы, рекомендованную к данной теме. практического занятия и по возможности подготовить по нему презентацию. Если программой дисциплины предусмотрено выполнение практического задания, то его необходимо выполнить с учетом предложенной инструкции. Все новые понятия по изучаемой теме необходимо внести в глоссарий, который целесообразно вести с самого начала изучения курса. Результат такой работы должен проявиться в способности обучающегося свободно ответить на теоретические вопросы практического занятия, его выступлении и участии в коллективном обсуждении вопросов изучаемой темы, правильном выполнении практических заданий и контрольных работ.

Структура практического занятия:

В зависимости от содержания и количества отведенного времени на изучение каждой темы может практическое занятие состоять из четырех-пяти частей:

1. Устный опрос.

2. Обсуждение теоретических вопросов, определенных программой дисциплины.

3. Выполнение практических заданий с последующим разбором полученных результатов или обсуждение практического задания, выполненного дома.

4. Подведение итогов занятия.

Работа с литературными источниками:

В процессе подготовки к практическим занятиям, обучающимся необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у обучающихся свое отношение к конкретной проблеме. Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем, что позволяет обучающимся проявить свою индивидуальность в рамках выступления на занятиях, выявить широкий спектр мнений по изучаемой проблеме.

8. Учебная литература и ресурсы информационно-телекоммуникационной сети Интернет

8.1. Основная литература

1. Токмаков, Г. П. Базы данных: Модели и структуры данных, язык SQL, программирование баз данных : учебное пособие / Г. П. Токмаков. — Ульяновск : УлГТУ, 2021. — 362 с. — ISBN 978-5-9795-2184-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/259706 (дата обращения: 01.05.2026). — Режим доступа: для авториз. пользователей.

2. Волк, В. К. Базы данных. Проектирование, программирование, управление и администрирование : учебник для вузов / В. К. Волк. — 5-е изд., стер. — Санкт-Петербург : Лань, 2025. — 244 с. — ISBN 978-5-507-53648-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/493991 (дата обращения: 01.05.2026). — Режим доступа: для авториз. пользователей.

3. Минязев, Р. Ш. Скриптовые языки web-программирования (JavaScript, PHP, html/CSS) : учебно-методическое пособие / Р. Ш. Минязев. — Казань : КНИТУ-КАИ, 2022. — 60 с. — ISBN 978-5-7579-2632-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/399557 (дата обращения: 01.05.2026). — Режим доступа: для авториз. пользователей.

4. Ружицкая, Е. А. Основы web-технологий: работа с данными формы в PHP: практическое пособие : учебное пособие / Е. А. Ружицкая, Е. П. Кечко. — Гомель : ГГУ имени Ф. Скорины, 2022. — 47 с. — ISBN 978-985-577-883-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/320990 (дата обращения: 01.05.2026). — Режим доступа: для авториз. пользователей.

8.2. Дополнительная литература

1. Баланов, А. Н. Бэкенд-разработка веб-приложений: архитектура, проектирование и управление проектами : учебное пособие для вузов / А. Н. Баланов. — 2-е изд., стер. — Санкт-Петербург : Лань, 2025. — 312 с. — ISBN 978-5-507-52472-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/451820 (дата обращения: 01.05.2026). — Режим доступа: для авториз. пользователей.

2. Филиппов, Ф. В. HTTP + PHP в примерах и задачах : учебное пособие / Ф. В. Филиппов, А. Н. Губин. — Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2015. — 67 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/180044 (дата обращения: 01.05.2026). — Режим доступа: для авториз. пользователей.

3. Побединский, Е. В. Проектирование веб-сайтов с

использованием технологий PHP, HTML, CSS и WordPress : учебное пособие / Е. В. Побединский, В. В. Побединский. — Екатеринбург : УГЛТУ, 2018. — 115 с. — ISBN 978-5-94984-651-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/142518 (дата обращения: 01.05.2026). — Режим доступа: для авториз. пользователей.

4. Янцев, В. В. JavaScript и PHP. Content management system / В. В. Янцев. — 2-е изд., испр. — Санкт-Петербург : Лань, 2023. — 192 с. — ISBN 978-5-507-48326-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/346460 (дата обращения: 01.05.2026). — Режим доступа: для авториз. пользователей.

8.3.Нормативные правовые документы и иная правовая информация

1. Конституция Российской Федерации. – Текст : электронный // Сайт Президента Российской Федерации. – URL: http://www.kremlin.ru/acts/constitution

8.4 Интернет-ресурсы

1. Информационно-правовой портал ГАРАНТ.РУ. – URL: https://www.garant.ru/
2. Научная электронная библиотека eLIBRARY.RU. – URL: https://elibrary.ru/
3. Научная электронная библиотека «КиберЛенинка». – URL: https://cyberleninka.ru
4. Электронно-библиотечная система «Лань». – URL: http://e.lanbook.com
5. База знаний по ОС RedOS – URL: https://redos.red-soft.ru/base/
6. Документация по Mysql – URL: https://metanit.com/sql/mysql/
7. Документация по PHP – URL: https://www.php.net/manual/ru/index.php

9. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

Материально-техническое обеспечение дисциплины включает в себя:

- лекционные аудитории, оборудованные видеопроекционным оборудованием для презентаций, средствами звуковоспроизведения, экраном;
- помещения для проведения семинарских и практических занятий,

оборудованные учебной мебелью.

Дисциплина поддержана соответствующими программными продуктами с открытой лицензией: RedOS, MariaDB, Apache, PHP, phpMyAdmin.

Вуз обеспечивает каждого обучающегося рабочим местом в компьютерном классе в соответствии с объемом изучаемых дисциплин, обеспечивает выход в сеть Интернет.

Помещения для самостоятельной работы обучающихся включают следующую оснащенность: столы аудиторные, стулья, доски аудиторные, компьютеры с подключением к локальной сети института (для компьютерных аудиторий) и Интернет. Для изучения учебной дисциплины используются автоматизированная библиотечная информационная система и электронные библиотечные системы.