

Кафедра _____ прикладной информатики и высшей математики

Б1.В.01

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА)

Учебная дисциплина	<i>Система управления базами данных</i>
По направлению подготовки	<i>09.03.03</i>
	<i>«Прикладная информатика»</i>
Профиль (программа бакалавриата)	<i>«Дизайн и разработка цифровых программных продуктов»</i>
Форма обучения	<i>Очная</i>

Оценочные материалы (средства) дисциплины рассмотрены (актуализированы) и утверждены на заседании кафедры прикладной информатики и высшей математики

Протокол заседания № 10 от «25» мая 2026 г.

Заведующий кафедрой Стрекалова Наталья Борисовна

ОГЛАВЛЕНИЕ

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ И ЭТАПЫ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ.....	3
2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ОБЕСПЕЧИВАЮЩИХ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ	5
3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ.....	10
4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНОВАНИЯ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ	13
5. ДИАГНОСТИЧЕСКИЕ ЗАДАНИЯ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ УРОВЕНЬ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ.....	21
6. ЛИСТ СОГЛАСОВАНИЯ.....	40

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ И ЭТАПЫ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Оценочные материалы (средства) сформированы в соответствии с ФГОС ВО - бакалавриат по направлению подготовки 09.03.03 «Прикладная информатика», утвержденный приказом Министерства образования и науки Российской Федерации от 19.09.2017 № 922 (с изменениями и дополнениями от 26.11.2020, 08.02.2021). В соответствии с матрицей компетенций основной профессиональной образовательной программы «Дизайн и разработка цифровых программных продуктов» (уровень бакалавриата) в процессе обучения по дисциплине «Система управления базами данных» происходит формирование закрепленных за дисциплиной компетенций обучающихся. Оценка сформированности компетенций на каждом этапе обучения происходит через оценку планируемых результатов обучения по дисциплине (знаний, умений, навыков).

Результаты освоения образовательной программы (компетенции)	Индикаторы достижения компетенций	Планируемые результаты обучения по дисциплине	Этапы формирования компетенции (семестры и темы)	Оценочные средства
ПК-1 Способен выявлять информационные потребности пользователей и составлять техническое задание на разработку цифрового программного продукта	ПК-1.1. Проводит обследование организаций, выявляет информационные потребности заказчика, используя цифровые инструменты и облачные решения для сбора и анализа данных	Знать: - знает назначение, суть и способы проведения предпроектного исследования организации как стартового этапа жизненного цикла информационных систем; Уметь: - составлять техническую документацию по результатам обследования организации и выявления потребностей пользователей по созданию баз данных; Владеть: - навыками формализованного представления требований к базе данных.	Тема 1. Назначение и состав базы данных и СУБД. Классификация. Архитектура «клиент-сервер» Тема 6. Транзакции: реализация и назначение	устный опрос по теме 1 проверка практических заданий по теме 6 защита индивидуальной разработки по теме 6

ПК-2 Способен проектировать цифровой программный продукт	ПК-2.1. Разрабатывает модель цифрового программного продукта с учетом внедрения передовых ИТ-решений прикладной сферы	Знать: - принципы построения и работы с базами данных и СУБД; основные алгоритмы решения задач предметной области, их особенности и характеристики; Уметь: - выбирать оптимальные средства решения задач, минимизировать пути решения, представлять результат; Владеть: - навыками разработки архитектуры web-приложения с учетом передовых ИТ-решений прикладной сферы;	Тема 1. Назначение и состав базы данных и СУБД. Классификация. Архитектура «клиент-сервер» Тема 2. Языки запросов, их назначение. Язык SQL, стандарт и диалекты Тема 6. Транзакции: реализация и назначение	устный опрос по темам 1-2 проверка практических заданий по темам 2 и 6 защита индивидуальной разработки по теме 6
	ПК-2.2. Проектирует базы данных, в том числе используя специализированные цифровые платформы	Знать: - знать основы проектирования баз данных; Уметь: - определить необходимые функциональные возможности проектируемой СУБД; - строить концептуальную, логическую и физическую модели баз данных; Владеть: - навыками проектирования баз данных;	Тема 2. Языки запросов, их назначение. Язык SQL, стандарт и диалекты. Язык SQL, стандарт и диалекты Тема 6. Транзакции: реализация и назначение	устный опрос по теме 2 проверка практических заданий по темам 2 и 6 защита индивидуальной разработки по теме 6
ПК-3 Способен разрабатывать цифровой программный продукт	ПК-3.1. Осуществляет разработку программного кода и компонентов цифрового	Знать: - принципы хранения и обработки данных в базах данных; классификацию баз данных по структуре,	Тема 3. Раздел DQL языка SQL Тема 4. Раздел DML языка SQL Тема 5. Раздел	устный опрос по темам 3-5 проверка практических заданий по темам 3-6 защита

	программного продукта с помощью специализированных цифровых платформ для индивидуальной и совместной разработки	принципы представления информации различных типов; - синтаксис SQL Уметь: - писать и отлаживать программные коды для решения поставленных задач; Владеть: - навыками построения поисковых запросов; - навыками построения и отладки SQL-запросов; - навыками формулирования и анализа результатов запросов к базам данных.	DDL языка SQL Тема 6. Транзакции: реализация и назначение	индивидуальной разработки по теме 6
--	---	---	---	-------------------------------------

2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ОБЕСПЕЧИВАЮЩИХ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Устные опросы

Примерные вопросы для проведения устного опроса по теме «Назначение и состав базы данных и СУБД. Классификация.

Архитектура «клиент- сервер»»

1. Структуризация информации.
2. Классификация баз данных по структуре данных.
3. Принципы организации иерархической модели данных.
4. Принципы организации реляционной модели данных.
5. Общие принципы организации основных прикладных пакетов и принципы хранения информации в основных прикладных программах.
6. Организация интерфейса пользователя в различных прикладных программах.
7. Типы элементов управления WINDOWS и возможность их использования при построении интерфейса пользователя СУБД.
8. Возможности конверсии данных в различных прикладных программах

Примерные вопросы для проведения устного опроса по теме «Языки запросов, их назначение. Язык SQL, стандарт и диалекты»

1. Локальные и распределенные базы данных.
2. Архитектура «клиент-сервер».

3. Файловый сервер.
4. Сервер приложений.
5. Назначение и общие принципы организации SQL.
6. Классификация SQL-запросов.
7. Стандарт ANSI и различные диалекты SQL.
8. Использование SQL в офисных пакетах.

*Примерные вопросы для проведения устного опроса
по теме «Раздел DQL языка SQL»*

1. SQL-сервер
2. SQL-запросы группы DQL (Data Query Language).
3. Отбор записей по условию.
4. Группировка данных.
5. Сортировка.
6. Вложенные запросы.

*Примерные вопросы для проведения устного опроса
по теме «Раздел DML языка SQL»*

1. Язык SQL: назначение, структура.
2. Имена таблиц. Имена и типы полей.
3. Особенности ключевого поля.
4. SQL-запросы группы DML (Data Manipulation Language).
5. Использование фразы Where в запросах DML.
6. Добавление нескольких записей.
7. Проблемы целостности данных при добавлении, изменении и удалении.

*Примерные вопросы для проведения устного опроса
по теме «Раздел DDL языка SQL»*

1. SQL-запросы группы DDL (Data Definition Language).
2. Структура и данные.
3. Индексы.
4. Типы полей.
5. Язык SQL: назначение, структура.
6. Имена таблиц. Имена и типы полей.
7. Особенности ключевого поля.

Критерии оценивания устного опроса:

- оценка «*отлично*» выставляется обучающемуся, если он ответил развернуто на один из заданных вопросов, активно дополнял ответы других обучающихся или задавал им дополнительные вопросы;
- оценка «*хорошо*» выставляется обучающемуся, если он ответил развернуто на один из заданных вопросов или ответил кратко на ряд вопросов;
- оценка «*удовлетворительно*» выставляется обучающемуся, если он кратко ответил на один из задаваемых вопросов или просто дополнял ответы других обучающихся, задавал им дополнительные вопросы;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если он не смог ответить правильно на заданный вопрос, либо не отвечал на вопросы и не участвовал в их обсуждении.

Практические работы

Практическое задание по теме «Языки запросов, их назначение. Язык SQL, стандарт и диалекты»

Задание: создать базу данных по выданному варианту, ввести данные в таблицу. Создать связи в схеме данных в Access.

Практическое задание по теме «Раздел DQL языка SQL»

Задание: На основе созданной и наполненной базы данных выполнить SQL-запросы группы DQL (Data Query Language):

1. Выполнить запрос на отбор записей по условию.
2. Выполнить запрос на группировку данных.
3. Выполнить запрос на сортировку.
4. Выполнить вложенные запросы.

Оформить результат выполненного задания в отчет по предложенному образцу в MS Word: тема практического задания, вариант, задание, запросы, результат запроса

Практическое задание по теме «Раздел DML языка SQL»

Задание: На основе созданной и наполненной базы данных выполнить SQL-запросы группы DML (Data Manipulation Language):

1. Выполнить запрос с использованием фразы Where в запросах DML.
2. Добавить нескольких записей.
3. Оценить проблемы целостности данных при добавлении, изменении и удалении.

Оформить результат выполненного задания в отчет по предложенному образцу в MS Word: тема практического задания, вариант, задание, запросы, результат запроса

Практическое задание по теме «Раздел DDL языка SQL»

Задание: На основе созданной и наполненной базы данных выполнить SQL-запросы группы DDL (Data Definition Language):

1. Выполнить запрос с использованием структур и данных.
2. Выполнить запрос с индексами.
3. Выполнить запрос с типами полей.

Оформить результат выполненного задания в отчет по предложенному образцу в MS Word: тема практического задания, вариант, задание, запросы, результат запроса

Практическое задание по теме «Транзакции: реализация и назначение»

Задание: На основе созданной и наполненной базы данных выполнить SQL-запросы группы DQL (Data Query Language): продемонстрировать реализацию механизма транзакции.

Оформить результат выполненного задания в отчет по предложенному образцу в MS Word: тема практического задания, вариант, задание, запросы, результат запроса

Критерии оценивания практического задания:

- оценка «*отлично*» выставляется обучающемуся, который продемонстрировал высокий уровень освоения учебного материала; правильно выполнил практическое задание в полном объеме; продемонстрировал высокий уровень умения использовать теоретические знания при выполнении практических задач; при выполнении последующих заданий применяет знания предыдущие; уверенно ориентируется в возможностях используемого программного обеспечения; умеет оформлять результаты выполнения практического задания в соответствии с правилами оформления отчетных работ.
- оценка «*хорошо*» выставляется обучающемуся, который знает основные определения и понятия, необходимые для выполнения практического задания; выполнил практическое задание в полном объеме; в целом правильно, но допускает 2-3 неточности, возможно выбирает не самый оптимальный вариант выполнения задания; умеет оформлять результаты выполнения практического задания в соответствии с правилами оформления отчетных работ.
- оценка «*удовлетворительно*» выставляется обучающемуся, если задание было выполнено полностью, но присутствуют грубые ошибки, свидетельствующие о недостаточной проработке теоретического материала; умеет оформлять результаты выполнения практического задания в соответствии с правилами оформления отчетных работ.
- оценка «*неудовлетворительно*» выставляется обучающемуся, если задание было выполнено в объеме менее 50%; при выполнении задания допущены многочисленные грубые ошибки; обучающийся не владеет необходимыми теоретическими знаниями; обучающийся не владеет специализированным программным обеспечением, изучаемым в рамках дисциплины.

Индивидуальная разработка по теме

Индивидуальная разработка подразумевает ряд заданий, соответствующих темам дисциплины и формируемым компетенциям. По итогам работ все выполненные задания собираются в единый документ, структурируются и оформляются в соответствии с требованиями вуза. Выполнение индивидуальной разработки обязательно для всех обучающихся, т.к. является одним из условий допуска к зачету с оценкой. Оценивание индивидуальной разработки производится целостно (при полном выполнении всех заданий), защита осуществляется на зачете с оценкой.

Примерный список предметных областей для выполнения индивидуальной разработки

1. Разработка базы данных «Гостиницы города»
2. Разработка базы данных «Земельный кадастр»
3. Разработка базы данных «Лига чемпионов по футболу»
4. Разработка базы данных «Музей»
5. Разработка базы данных «Радиостанция»
6. Разработка базы данных «Станция технического обслуживания авто»
7. Разработка базы данных «Служба такси»
8. Разработка базы данных «Чрезвычайные происшествия»
9. Разработка базы данных «Поставки строительных материалов»
10. Разработка базы данных «Магазин канцтоваров»

Задание:

1. Проанализировать предметную область, выделить сущности и атрибуты, построить концептуальную модель предметной области.
2. Создать базу данных по выданному варианту, ввести данные в таблицу. Создать связи в схеме данных в Access.
3. Выполнить запрос на отбор записей по условию.
4. Выполнить запрос на группировку данных.
5. Выполнить запрос на сортировку.
6. Выполнить вложенные запросы.
7. Выполнить запрос с использованием фразы Where в запросах DML.
8. Добавить нескольких записей.
9. Оценить проблемы целостности данных при добавлении, изменении и удалении.
10. Выполнить запрос с использованием структур и данных.
11. Выполнить запрос с индексами.
12. Выполнить запрос с типами полей.
13. На основе созданной и наполненной базы данных продемонстрировать реализацию механизма транзакции.
14. Разработать пользовательские формы (4-5, включая главную форму), отчеты (2-3, включая сложный отчет) на основе запросов.
15. Оформить отчет в MS Word по предложенному шаблону.
16. Продемонстрировать на зачете с оценкой готовую базу данных, ответить на вопросы преподавателя и аудитории.

Критерии оценивания индивидуальной разработки:

- оценка «отлично» выставляется обучающемуся, если работа выполнена в срок и в полном объеме (100%), выполненные запросы работают корректно, база данных содержит пользовательские формы и отчеты, актуальные для предметной области, отчет выполнен в полном соответствии с требованиями вуза;
- оценка «хорошо» выставляется обучающемуся, если работа выполнена почти в полном объеме – не менее 90% (допустимо отсутствие 1-2 пунктов задания), но присутствуют

некоторые недочеты (не имеющие «критичных» последствий для последующих работ), выполненные запросы работают корректно, база данных содержит пользовательские формы и отчеты, актуальные для предметной области, отчет оформлен в соответствии с требованиями с единичными нарушениями;

- оценка *«удовлетворительно»* выставляется обучающемуся, если работа была выполнена не менее, чем на 80% (могут отсутствовать 3-4 пункта задания), есть грубые ошибки, имеющие «критичные» последствия для последующих заданий, база данных содержит пользовательские формы и отчеты, которые слабо связаны с предметной областью; отчет оформлен в соответствии с требованиями с немногочисленными нарушениями;
- оценка *«неудовлетворительно»* выставляется обучающемуся, если отчет не был представлен или работа была выполнена менее, чем на 80% (отсутствует более 4 пунктов задания) и/или в работе очень много грубых ошибок (имеющие «критичных» последствий для последующих заданий) и/или отсутствует какая-либо связь между заданиями; база данных не содержит пользовательские формы и отчеты; отчет не оформлен или его оформление не соответствует требованиям.

3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

Компетенция ПК-1 Способен выявлять информационные потребности пользователей и составлять техническое задание на разработку цифрового программного продукта

Индикатор ПК-1.1. Проводит обследование организаций, выявляет информационные потребности заказчика, используя цифровые инструменты и облачные решения для сбора и анализа данных

Обучающийся знает: знает назначение, суть и способы проведения предпроектного исследования организации как стартового этапа жизненного цикла информационных систем.

Оценка достижения обучающимся запланированного результата обучения осуществляется по полноте представленного отчета по индивидуальной разработке и ответов на вопросы для устного опроса:

1. Понятие и назначение БД, СУБД
2. Классификация БД по структуре.
3. Принципы организации реляционных баз данных.
4. Назначение серверов БД. Архитектура "Клиент - Сервер".
5. Модели серверов баз данных.
6. Язык SQL: назначение, структура.
7. Имена таблиц. Имена и типы полей.

Компетенция ПК-2 Способен проектировать цифровой программный продукт

Индикатор ПК-2.1. Разрабатывает модель цифрового программного продукта с учетом внедрения передовых ИТ-решений прикладной сферы

Обучающийся знает: принципы построения и работы с базами данных и СУБД; основные алгоритмы решения задач предметной области, их особенности и характеристики.

Оценка достижения обучающимся запланированного результата обучения осуществляется по ответам на вопросы для устного опроса:

1. Запросы манипуляции данными.
2. Запросы на выборку в SQL. Структура запросов на выборку.
3. Исключение дублирующихся значений.
4. Вычисление значений на основе имеющихся полей.

Компетенция ПК-2 Способен проектировать цифровой программный продукт

Индикатор ПК-2.2. Проектирует базы данных, в том числе используя специализированные цифровые платформы

Обучающийся знает: основы проектирования баз данных

Оценка достижения обучающимся запланированного результата обучения осуществляется по ответам на вопросы для устного опроса:

1. Техническое задание
2. Спецификации на разработку БД
3. Принципы упорядочивания записей. Запросы с упорядочиванием.
4. Назначение и принципы построения интерфейса пользователя.
5. Основные элементы управления в Windows. Их назначение.

Компетенция ПК-3 Способен разрабатывать цифровой программный продукт

Индикатор ПК-3.1. Осуществляет разработку программного кода и компонентов цифрового программного продукта с помощью специализированных цифровых платформ для индивидуальной и совместной разработки

Обучающийся знает: принципы хранения и обработки данных в базах данных; классификацию баз данных по структуре, принципы представления информации различных типов; синтаксис SQL

Оценка достижения обучающимся запланированного результата обучения осуществляется по полноте представленного отчета по индивидуальной разработке и ответов на вопросы для устного опроса:

1. Запросы на выборку в SQL. Структура запросов на выборку.
2. Исключение дублирующихся значений.
3. Вычисление значений на основе имеющихся полей.
4. Принципы упорядочивания записей. Запросы с упорядочиванием.
5. Запросы на добавление данных в SQL. Структура запросов на добавление данных.
6. Добавление данных в таблицу из другой таблицы
7. Запросы изменения данных в таблице в SQL.
8. Запрос на удаление данных в SQL.
9. Назначение и принцип построения фразы WHERE в SQL.

ОЦЕНКА УМЕНИЙ И НАВЫКОВ ПО РЕЗУЛЬТАТАМ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

Компетенция ПК-1 Способен выявлять информационные потребности пользователей и составлять техническое задание на разработку цифрового программного продукта

Индикатор ПК-1.1. Проводит обследование организаций, выявляет информационные потребности заказчика, используя цифровые инструменты и облачные решения для сбора и анализа данных

Обучающийся умеет: составлять техническую документацию по результатам обследования организации и выявления потребностей пользователей по созданию баз данных.

Обучающийся владеет: навыками формализованного представления требований к базе данных.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам выполнения практических работ, а также при выполнении и защите индивидуальных разработок.

Компетенция ПК-2 Способен проектировать цифровой программный продукт

Индикатор ПК-2.1. Разрабатывает модель цифрового программного продукта с учетом внедрения передовых ИТ-решений прикладной сферы

Обучающийся умеет: выбирать оптимальные средства решения задач, минимизировать пути решения, представлять результат

Обучающийся владеет: навыками разработки архитектуры web-приложения с учетом передовых ИТ-решений прикладной сферы.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам выполнения практических работ, а также при выполнении и защите индивидуальных разработок.

Компетенция ПК-2 Способен проектировать цифровой программный продукт

Индикатор ПК-2.2. Проектирует базы данных, в том числе используя специализированные цифровые платформы

Обучающийся умеет: определить необходимые функциональные возможности проектируемой СУБД; строить концептуальную, логическую и физическую модели баз данных.

Обучающийся владеет: навыками проектирования баз данных.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам выполнения практических работ.

Компетенция ПК-3 Способен разрабатывать цифровой программный продукт

Индикатор ПК-3.1. Осуществляет разработку программного кода и компонентов цифрового программного продукта с помощью специализированных цифровых платформ для индивидуальной и совместной разработки

Обучающийся умеет: писать и отлаживать программные коды для решения поставленных задач

Обучающийся владеет: навыками построения поисковых запросов; навыками построения и отладки SQL-запросов; навыками формулирования и анализа результатов запросов к базам данных.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам выполнения практических работ, а также при выполнении и защите индивидуальных разработок.

4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНОВАНИЯ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ

ПРОЦЕДУРА ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Для контроля усвоения обучающимися данной дисциплины и достижения запланированных результатов обучения учебным планом предусмотрен зачет с оценкой, проводимый в форме устного ответа на теоретический вопрос. При выставлении итоговой оценки по дисциплине учитываются результаты текущего контроля, отражающего сформированность практических умений и навыков обучающегося. Во время обучения обучающиеся участвуют в разных мероприятиях текущего контроля: участвуют в устных теоретических опросах, выполняют практические и индивидуальные задания. Для допуска к зачету с оценкой обучающемуся необходимо, как минимум, выполнить индивидуальную разработку.

Зачет с оценкой проходит в форме устного ответа на теоретический вопрос. Допустимо совместить защиту индивидуального задания и ответ на теоретический вопрос зачета с оценкой.

КРИТЕРИИ ОЦЕНИВАНИЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
Компетенция ПК-1 Способен выявлять информационные потребности пользователей и составлять техническое задание на разработку цифрового программного продукта Индикатор ПК-1.1. Проводит обследование организаций, выявляет информационные потребности заказчика, используя цифровые инструменты и облачные решения для сбора и анализа данных					
Знать:	Не знает	Фрагментарн	Общие, но не	Сформирова	Сформирова

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
- назначение, суть и способы проведения предпроектного исследования организации как стартового этапа жизненного цикла информационных систем		ые знания назначения, сути и способах проведения предпроектного исследования организации как стартового этапа жизненного цикла информационных систем	структурированные знания назначения, сути и способах проведения предпроектного исследования организации как стартового этапа жизненного цикла информационных систем	ные, но содержащие отдельные пробелы знания назначения, сути и способах проведения предпроектного исследования организации как стартового этапа жизненного цикла информационных систем	ные систематизированные прочные знания назначения, сути и способах проведения предпроектного исследования организации как стартового этапа жизненного цикла информационных систем
Уметь: - составлять техническую документацию по результатам обследования организации и выявления потребностей пользователей по созданию баз данных	Не умеет	Частично освоенные умения составлять техническую документацию по результатам обследования организации и выявления потребностей пользователей по созданию баз данных	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения составлять техническую документацию по результатам обследования организации и выявления потребностей пользователей по созданию баз данных	В целом сформированные и самостоятельно выполняемые, но не всегда интеллектуально обоснованные умения составлять техническую документацию по результатам обследования организации и выявления потребностей	Успешно сформированные, самостоятельно и осознанно выполняемые умения составлять техническую документацию по результатам обследования организации и выявления потребностей пользователей по созданию баз данных

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
				пользователем по созданию баз данных	
Владеть: - навыками формализованного представления требований к базе данных	Не владеет	Фрагментарно сформированные навыки формализованного представления требований к базе данных	В целом сформированные, но выполняемые на элементарном уровне навыки формализованного представления требований к базе данных	Сформированные, но не устойчивые в сложных ситуациях навыки формализованного представления требований к базе данных	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки формализованного представления требований к базе данных
Компетенция ПК-2 Способен проектировать цифровой программный продукт Индикатор ПК-2.1. Разрабатывает модель цифрового программного продукта с учетом внедрения передовых ИТ-решений прикладной сферы					
Знать: принципы построения и работы с базами данных и СУБД; основные алгоритмы решения задач предметной области, их особенности и характеристик и	Не знает	Фрагментарные знания принципов построения и работы с базами данных и СУБД; основных алгоритмов решения задач предметной области, их особенности и характеристик и	Общие, но не структурированные знания принципов построения и работы с базами данных и СУБД; основных алгоритмов решения задач предметной области, их особенности и характеристик и	Сформированные, но содержащие отдельные пробелы знания принципов построения и работы с базами данных и СУБД; основных алгоритмов решения задач предметной области, их особенности и характеристики	Сформированные систематизированные прочные знания принципов построения и работы с базами данных и СУБД; основных алгоритмов решения задач предметной области, их особенности и характеристики
Уметь: выбирать	Не умеет	Частично освоенные	В целом освоенные, но	В целом сформирован	Успешно сформирован

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
оптимальные средства решения задач, минимизировать пути решения, представлять результат		умения выбирать оптимальные средства решения задач, минимизировать пути решения, представлять результат	не подкрепляемые знаниями и самостоятельностью выполнения умения выбирать оптимальные средства решения задач, минимизировать пути решения, представлять результат	ные и самостоятельные выполняемые, но не всегда интеллектуально обоснованные умения выбирать оптимальные средства решения задач, минимизировать пути решения, представлять результат	ные, самостоятельные и осознанно выполняемые умения выбирать оптимальные средства решения задач, минимизировать пути решения, представлять результат
Владеть: навыками разработки архитектуры web-приложения с учетом передовых ИТ-решений прикладной сферы	Не владеет	Фрагментарно сформированные навыки разработки архитектуры web-приложения с учетом передовых ИТ-решений прикладной сферы	В целом сформированные, но выполняемые на элементарном уровне навыки разработки архитектуры web-приложения с учетом передовых ИТ-решений прикладной сферы	Сформированные, но не устойчивые в сложных ситуациях навыки разработки архитектуры web-приложения с учетом передовых ИТ-решений прикладной сферы	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки разработки архитектуры web-приложения с учетом передовых ИТ-решений прикладной сферы
Компетенция ПК-2 Способен проектировать цифровой программный продукт Индикатор ПК-2.2. Проектирует базы данных, в том числе используя специализированные цифровые платформы					
Знать: знать основы проектирования баз данных	Не знает	Фрагментарные знания основ проектирования баз данных	Общие, но не структурированные знания основ проектирования	Сформированные, но содержащие отдельные пробелы	Сформированные систематизированные прочные

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
			ия баз данных	знания основ проектирования баз данных	знания основ проектирования баз данных
Уметь: -определить необходимые функциональные возможности и проектируемой СУБД; -строить концептуальную, логическую и физическую модели баз данных;	Не умеет	Частично освоенные умения определять необходимые функциональные возможности проектируемой СУБД и строить концептуальную, логическую и физическую модели баз данных	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения определять необходимые функциональные возможности проектируемой СУБД и строить концептуальную, логическую и физическую модели баз данных	В целом сформированные и самостоятельно выполняемые, но не всегда интеллектуально обоснованные умения определять необходимые функциональные возможности проектируемой СУБД и строить концептуальную, логическую и физическую модели баз данных	Успешно сформированные, самостоятельно и осознанно выполняемые умения определять необходимые функциональные возможности проектируемой СУБД и строить концептуальную, логическую и физическую модели баз данных
Владеть: навыками проектирования баз данных	Не владеет	Фрагментарно сформированные навыки проектирования баз данных	В целом сформированные, но выполняемые на элементарном уровне навыки проектирования баз данных	Сформированные, но не устойчивые в сложных ситуациях навыки проектирования баз данных	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки проектирования баз данных

Компетенция ПК-3 Способен разрабатывать цифровой программный продукт

Индикатор ПК-3.1. Осуществляет разработку программного кода и компонентов цифрового

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
программного продукта с помощью специализированных цифровых платформ для индивидуальной и совместной разработки					
Знать: принципы хранения и обработки данных в базах данных; классификацию баз данных по структуре, принципы представления информации различных типов; синтаксис SQL	Не знает	Фрагментарные знания принципов хранения и обработки данных в базах данных; классификации баз данных по структуре, принципы представления информации различных типов; синтаксиса SQL	Общие, но не структурированные знания принципов хранения и обработки данных в базах данных; классификации баз данных по структуре, принципы представления информации различных типов; синтаксиса SQL	Сформированные, но содержащие отдельные пробелы знания принципов хранения и обработки данных в базах данных; классификации баз данных по структуре, принципы представления информации различных типов; синтаксиса SQL	Сформированные систематизированные прочные знания принципов хранения и обработки данных в базах данных; классификации баз данных по структуре, принципы представления информации различных типов; синтаксиса SQL
Уметь: - писать и отлаживать программные коды для решения поставленных задач	Не умеет	Частично освоенные умения писать и отлаживать программные коды для решения поставленных задач	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения писать и отлаживать программные коды для решения поставленных задач	В целом сформированные и самостоятельные, но выполняемые, но не всегда интеллектуально обоснованные умения писать и отлаживать программные коды для решения поставленных задач	Успешно сформированные, самостоятельно и осознанно выполняемые умения писать и отлаживать программные коды для решения поставленных задач

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
Владеть: навыками построения поисковых запросов; навыками построения и отладки SQL-запросов; навыками формулирования и анализа результатов запросов к базам данных	Не владеет	Фрагментарно сформированные навыки построения поисковых запросов; построения и отладки SQL-запросов; формулирования и анализа результатов запросов к базам данных	В целом сформированные, но выполняемые на элементарном уровне навыки построения поисковых запросов; построения и отладки SQL-запросов; формулирования и анализа результатов запросов к базам данных	Сформированные, но не устойчивые в сложных ситуациях навыки построения поисковых запросов; построения и отладки SQL-запросов; формулирования и анализа результатов запросов к базам данных	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки построения поисковых запросов; построения и отладки SQL-запросов; формулирования и анализа результатов запросов к базам данных

КРИТЕРИИ ОЦЕНКИ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Критерии оценивания для зачета с оценкой:

При выставлении итоговой оценки за дисциплину необходимо учитывать результаты текущего контроля и критерии оценивания сформированности компетенций.

Оценка **«отлично»** выставляется обучающемуся, если:

- при ответе на теоретический вопрос были продемонстрированы глубокие, систематизированные и прочные знания (в соответствии с критериями сформированности компетенций), умения логично и обоснованно излагать материал, быстро ориентироваться в дополнительных вопросах и давать полные ответы;
- индивидуальное задание выполнено безошибочно и в полном соответствии с требованиями на оценку «отлично», обучающийся в работе продемонстрировал устойчивые навыки работы и осознанно исполняемые действия для разработки базы данных, способность выполнять различные запросы и обосновывать свои действия;
- во время защиты индивидуального задания обучающийся логически стройно, доходчиво и грамотно продемонстрировал весь функционал базы данных, особенности построения запросов, полно и доступно отвечал на дополнительные вопросы;
- обучающийся участвовал во всех видах текущего контроля и получал по ним максимальные баллы (допустимо отдельные отметки «хорошо»).

Все вышеперечисленное говорит о сформированности запланированных компетенций на высоком уровне.

Оценка **«хорошо»** выставляется обучающемуся, если:

- при ответе на теоретический вопрос было продемонстрировано хорошее, но недостаточно полное изложение материала, с отдельными пробелами в знаниях (возможно, незначительными неточностями в употреблении терминов), логичное изложение материала, но недостаточно быстрая ориентация в нем при ответе на дополнительные вопросы;
- индивидуальное проектное задание выполнено на оценку «отлично» или «хорошо» (с незначительными замечаниями и/или ошибками), обучающийся продемонстрировал хорошие навыки работы и осознанно исполняемые действия по разработке баз данных;
- во время защиты индивидуального задания обучающийся достаточно подробно продемонстрировал функционал базы данных и особенности построения запросов, доступно отвечал на дополнительные вопросы;
- обучающийся участвовал во всех видах текущего контроля и получал по ним в основном отметки «хорошо».

Все вышеперечисленное говорит о сформированности запланированных компетенций на хорошем уровне.

Оценка **«удовлетворительно»** выставляется обучающемуся, если:

- при ответе на теоретический вопрос было продемонстрировано поверхностное владение материалом и фрагментарные знания, отсутствие стройного и логично выстроенного ответа, затруднение в употреблении терминов и приведении примеров, плохая ориентация в материале при ответе на дополнительные вопросы;
- индивидуальное задание выполнено на оценку «хорошо» или «удовлетворительно» (с ошибками и замечаниями), с отклонениями от требований, обучающийся затруднялся в объяснении применяемых средств для разработки базы данных;
- во время защиты индивидуального задания обучающийся невнятно представлял функционал базы данных, не смог выделить особенности построения запросов, не смог ответить на все дополнительные вопросы;
- обучающийся участвовал не во всех мероприятиях текущего контроля и получал по ним в основном удовлетворительные отметки.

Все вышеперечисленное говорит о сформированности запланированных компетенций на низком уровне.

Оценка **«неудовлетворительно»** выставляется обучающемуся, если:

- при ответе на теоретический вопрос было продемонстрировано незнание значительного объема материала, фрагментарность имеющихся знаний, отсутствие стройного и логично выстроенного ответа, затруднение в употреблении терминов и приведении примеров, неумение выделить главное, сделать выводы и обобщения, неспособность отвечать на дополнительные вопросы;
- индивидуальное задание выполнено на оценку «удовлетворительно» со значительными отклонениями от требований, с грубыми ошибками и множеством замечаний, обучающийся продемонстрировал частично сформированные навыки работы, затрудняется в объяснении применяемых средств при создании базы данных, не способен внести коррективы в работу;
- во время защиты индивидуального задания обучающийся представлял функционал базы данных фрагментарно, без подробностей разработки и работы самих запросов, не смог ответить на дополнительные вопросы;
- обучающийся участвовал не во всех мероприятиях текущего контроля и получал по ним в основном удовлетворительные отметки.

Все вышеперечисленное говорит о частичной сформированности запланированных компетенций.

5. ДИАГНОСТИЧЕСКИЕ ЗАДАНИЯ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ УРОВЕНЬ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Компетенция ПК-1. Способен выявлять информационные потребности пользователей и составлять техническое задание на разработку цифрового программного продукта.

Обучающийся знает: знает назначение, суть и способы проведения предпроектного исследования организации как стартового этапа жизненного цикла информационных систем.

Обучающийся умеет: составлять техническую документацию по результатам обследования организации и выявления потребностей пользователей по созданию баз данных.

Обучающийся владеет: навыками формализованного представления требований к базе данных.

ЗАДАНИЯ ЗАКРЫТОГО ТИПА (10 заданий)

Категория: знать.

Задание 1. В каком году был принят международный стандарт, в котором опубликовано следующее определение: информация - это знание относительно фактов, событий, идей и понятий, которые в определённом контексте имеют конкретный смысл.

- 1) 2013 год (ISO/IEC 27001:2013)
- 2) 2015 год (ISO/IEC 2382:2015)
- 3) 2018 год (ISO/IEC 20000-1:2018)
- 4) 2019 год (ITIL 4)

Ответ: 2) 2015 год (ISO/IEC 2382:2015)

Задание 2. Что такое техническое задание?

- 1) план маркетинговых действий;
- 2) документ, описывающий требования к функционалу и дизайну проекта;
- 3) отчет о проделанной работе;
- 4) график разработки.

Ответ: 2) Документ, описывающий требования к функционалу и дизайну проекта

Задание 3. Какой первый шаг в анализе информационных потребностей пользователей?

- 1) проектирование базы данных;
- 2) определение целевой аудитории;
- 3) написание кода;
- 4) выбор технологического стека.

Ответ: 2) Определение целевой аудитории

Задание 4. Соблюдение этого принципа позволяет избежать дублирование информации, исключить несурзности и ошибки, уменьшить потоки вводимой и обрабатываемой информации...

- 1) принцип полной информационной совместимости между автоматизированными системами различных уровней управления;
- 2) принцип одноразовости ввода данных в органы управления;
- 3) принцип типизации проектных решений;
- 4) принцип системности и комплексного подхода к проектированию АИС.

Ответ: 2) принцип одноразовости ввода данных в органы управления.

Задание 5. Укажите недостатки каскадной модели жизненного цикла автоматизированной информационной системы.

- 1) существенное запаздывание с получением результатов;
- 2) снижение управляемости проектом в целом;

- 3) сложность определения момента перехода на следующий этап;
- 4) примитивная автоматизация существующих производственных действий работников.

Ответ: 1, 4

Задание 6. Какой элемент является обязательным при формулировке требований к базе данных в техническом задании?

- 1) список таблиц и связей между ними;
- 2) структура данных;
- 3) цветовая схема интерфейса;
- 4) требования к производительности и масштабируемости системы.

Ответ: 2) структура данных.

Задание 7. Принцип типизации проектных решений обеспечивает следующие положительные эффекты при разработке автоматизированных информационных систем ...

- 1) избегание дублирования информации;
- 2) высокое качество разработки;
- 3) сокращение сроков разработки;
- 4) возможность активного участия в управленческого персонала организации в создании АИС;
- 5) упрощение использования разделённых баз данных;
- 6) снижение затрат на разработку и поддержку информационного обеспечения.

Ответ: 2, 3, 4

Задание 8. Какую роль выполняет сервер в архитектуре «клиент-сервер»?

- 1) отправляет запросы на другие устройства;
- 2) обрабатывает запросы от клиента и предоставляет необходимые ресурсы или сервисы;
- 3) распределяет интернет-трафик между пользователями;
- 4) защищает данные пользователя от вирусов.

Ответ: 2) Обрабатывает запросы от клиента и предоставляет необходимые ресурсы или сервисы.

Задание 9. Что важно учитывать при составлении технического задания?

- 1) только желания заказчика;
- 2) только текущие тенденции в дизайне;
- 3) только стоимость разработки;
- 4) потребности пользователей и бизнес-цели заказчика.

Ответ: 4) Потребности пользователей и бизнес-цели заказчика

Задание 10. Какие из перечисленных требований являются функциональными?

- 1) время загрузки страницы;
- 2) возможность регистрации пользователей;
- 3) поддержка мобильных устройств;
- 4) возможность оставлять отзывы о продукте.

Ответы: 2, 4

ЗАДАНИЯ ОТКРЫТОГО ТИПА (15 заданий)

Категория: уметь.

Задание 11. Распределите процессы жизненного цикла АИС по группам ((1) -основные, (2) - вспомогательные, (3) - организационные).

Варианты ответов:

- 1) эксплуатация;

- 2) управление конфигурацией;
- 3) аттестация;
- 4) улучшение жизненного цикла;
- 5) управление проектами;
- 6) создание инфраструктуры проекта.

Ответ: распределение 1, 2, 2, 3, 3, 3.

Задание 12. Установите истинность или ложность утверждений о спиральной модели жизненного цикла АИС ...

- 1) каждый виток спирали соответствует созданию нового фрагмента или версии ИС;
- 2) один виток спирали представляет собой законченный проектный цикл по типу каскадной схемы;
- 3) в спиральной модели ЖЦ делаются упорные этапы интеграции и реализации;
- 4) реализуемость технических решений проверяется путём создания прототипов;
- 5) основная проблема спирального цикла – существенное запаздывание с получением результатов;
- 6) использование спиральной модели даёт снижение стыкуемость различных фрагментов информационной системы.

Ответ: да, да, нет, да, нет, да

Задание 13. Сопоставьте названия принципов разработки АИС и их формулировки.

- 1) Проектирование должно осуществляться на основе определённой цели и критериев функционирования системы, взаимные увязки организационно-технологических решений.
- 2) Не системная оптимизация конкретной подсистемы может быть эффективно, но не позволяет оптимизировать систему в целом.
- 3) Процесс разработки внедрения АИС следует рассматривать как непрерывные использование предшествующего опыта.
- 4) Применение согласованных подходов к разработке машины и баз данных входным и выходным документом программным комплексом для АС различных предприятий.
- 5) Применение инвариантных по отношению к техническому обеспечению и легко трансформируемых технологий создания АС.

Варианты:

- ~ принцип независимости структуры АС управления от используемых техники и технологий;
- ~ принцип суп-оптимизации;
- ~ принцип обратной связи;
- ~ принцип полной информационной совместимости;
- ~ принцип системности и комплексного подхода.

Ответ: 5, 2, 3, 4, 1

Задание 14. Укажите порядок следования этапов каскадной модели жизненного цикла автоматизированных информационных систем.

- 1) Реализация.
- 2) Анализ.
- 3) Проектирование.
- 4) Сопровождение.
- 5) Внедрение.

Ответ: 3 2 1 5 4

Задание 15. Согласно основному документу, определяющему стадии и этапы проектирования информационных систем (ГОСТ 34.601-90 Автоматизированные системы. Стадии создания) определить последовательность стадий: ...

- 1) разработка концепции ИС;
- 2) формирование требований к ИС;
- 3) технический проект;
- 4) эскизный проект;
- 5) техническое задание;
- 6) рабочая документация.

Ответ: 2, 1, 5, 4, 3, 6

Задание 16. Недостатки этой модели ЖЦ связаны с тем, что реальный процесс создания ПО ИС обычно не укладывается в жёсткую схему. Практически постоянно возникает потребность возвращаться к предыдущим этапам, уточнять или пересматривать принятые решения. В результате затягиваются сроки выполнения работы, пользователи могут вносить замечания лишь по завершению всех работ с системой. При этом модели автоматизируемого объекта могут устареть к моменту их утверждения.

Недостатки какой модели перечислены?

Ответ: Недостатки, описанные в вопросе, относятся к каскадной (водопадной) модели жизненного цикла разработки программного обеспечения. Эта модель предполагает строгую последовательность этапов разработки: сбор требований, проектирование, реализация, тестирование, внедрение и поддержка. Каждый этап начинается только после завершения предыдущего, и возврат к предыдущим этапам не предусмотрен или сильно ограничен.

Задание 17. Почему важно анализировать конкурентов при планировании web-приложения?

Ответ: Анализ конкурентов позволяет выявить сильные и слабые стороны существующих решений на рынке, определить незаполненные ниши и потребности пользователей, которые не удовлетворены полностью. Эта информация помогает создать уникальное предложение, которое будет выделяться среди конкурентов и лучше отвечать на запросы целевой аудитории.

Категория: **владеть**.

Задание 18. Веб-приложение «Помощник домохозяйки». Начинающий разработчик пишет небольшое резюме по вопросу дальнейших действий по созданию данного приложения. Директор IT-компании увидел в каждом предложении ошибки. Пояснить, какие ошибки допустил начинающий разработчик.

- 1) На этапе технического задания формируются определенные проектные решения.
- 2) Каждое проектное решение сопровождается документацией, разработанной заранее еще на этапе формирования требований ИС.
- 3) Разработка предварительных проектных решений по системе осуществляется на этапе технического проектирования.
- 4) Сначала разрабатывается концепция, а потом формируются требования.
- 5) БД упоминается на этапе эскизного проектирования.

Ответ:

- 1) На этапе ТЗ проектные решения не формируются, это прерогатива эскизного проектирования. На этапе ТЗ разрабатывается и утверждается именно ТЗ.
- 2) Полученное проектное решение, действительно, сопровождается в итоге документацией, но разработанной не ранее эскизного проектирования.
- 3) Разработка предварительных проектных решений по системе осуществляется на этапе эскизного проектирования.
- 4) Нет. Сначала разрабатываются требования к ИС, а потом концепция.
- 5) Информация для БД анализируется на этапе изучения предметной области (разработка требований).

Задание 19. Какое отношение к веб-приложениям имеет слово - Agile?

Ответ: Agile, или гибкая методология разработки, играет ключевую роль в создании и оптимизации веб-приложений. Этот подход предполагает деление процесса разработки на короткие циклы, называемые итерациями или спринтами, что позволяет команде быстро реагировать на изменения требований и предпочтений пользователей. В контексте веб-приложений, Agile способствует повышению качества продукта за счет регулярного тестирования и обратной связи от пользователей. Также, Agile укрепляет коммуникацию внутри команды разработчиков и с заказчиком, что способствует более точному пониманию целей и задач проекта. Гибкая методология подразумевает акцент на людях и их взаимодействии, а не на жестком следовании заранее установленному плану. Это делает процесс разработки веб-приложений более адаптивным и инновационным. Agile также поддерживает принцип постоянного улучшения продукта и процессов разработки, что важно для долгосрочной эффективности и конкурентоспособности веб-приложения. В целом, применение Agile в разработке веб-приложений позволяет достигать высокого уровня удовлетворенности клиентов и пользователей за счет гибкости, открытости к изменениям и фокуса на создании ценности.

Задание 20. Какие факторы важно учитывать при выборе СУБД для web-приложения?

Ответ: При выборе системы управления базами данных (СУБД) для информационной системы (ИС) основные факторы, которые следует учитывать, включают:

1. Масштабируемость и производительность: СУБД должна быть способна адаптироваться к увеличению объема данных и росту нагрузки, обеспечивая при этом высокую скорость обработки запросов и операций с данными.

2. Безопасность данных: Важно выбирать СУБД с продвинутыми функциями безопасности для защиты от неавторизованного доступа, хакерских атак и утечек информации, включая поддержку шифрования, аутентификации и управления доступом.

3. Совместимость и гибкость: СУБД должна быть совместима с используемыми технологиями, платформами и языками программирования в рамках ИС. Гибкость в поддержке различных типов данных и возможностей позволяет более точно соответствовать функциональным требованиям проекта.

4. Удобство администрирования: Выбранная СУБД должна предлагать инструменты и интерфейсы для эффективного управления, настройки и мониторинга, что облегчает задачи администрирования и оптимизации производительности.

5. Стоимость и поддержка: Необходимо учитывать общую стоимость владения СУБД, включая лицензионные платежи, затраты на поддержку и обслуживание. Также важна доступность поддержки со стороны разработчиков СУБД и активности сообщества, что обеспечивает доступ к обновлениям, исправлениям ошибок и помощь в решении проблем.

Выбор СУБД — это комплексное решение, требующее анализа как текущих потребностей ИС, так и перспектив её развития.

Задание 21. Какие принципы нормализации данных следует учитывать при проектировании базы данных для web-приложения?

Ответ: При проектировании базы данных для web-приложения следует учитывать основные принципы нормализации данных: избегание дублирования данных (1НФ), обеспечение логической связности данных (2НФ), избавление от транзитивных зависимостей (3НФ), и при необходимости применять более высокие нормальные формы. Нормализация помогает улучшить структуру базы данных, облегчает её расширение и обновление, а также повышает эффективность запросов за счет уменьшения избыточности данных.

Задание 22. Как влияет выбор СУБД на разработку и поддержку web-приложения?

Ответ: Выбор системы управления базами данных (СУБД) оказывает глубокое влияние на различные аспекты разработки и поддержки информационной системы (ИС), включая:

1. Проектирование и архитектура ИС (способы хранения данных, влияние на архитектуру).
2. Разработка и интеграция(выбор технологий, сложность интеграции).
3. Производительность и масштабируемость (влияние на производительность, масштабируемость).
4. Безопасность (защита данных, управление доступом).
5. Поддержка и сообщество (доступность специалистов, поддержка и обновления).

Задание 23. Почему важно регулярно обновлять техническое задание в процессе разработки web-приложения?

Ответ: Регулярное обновление технического задания играет ключевую роль в процессе разработки информационной системы по нескольким причинам:

1. Актуализация требований (отражение изменений, предотвращение непонимания).
2. Гибкость проекта (адаптация к изменениям, принятие обоснованных решений).
3. Контроль качества (соответствие ожиданиям, минимизация ошибок).

В целом, регулярное обновление технического задания помогает повысить эффективность разработки, обеспечить соответствие продукта потребностям бизнеса и пользователей, а также снизить риски задержек и превышения бюджета проекта.

Вопрос 24. Как использование Use Cases может помочь в формализации требований к web-приложению?

Ответ: Использование Use Cases (прецедентов использования) играет ключевую роль в процессе формализации требований к информационным системам (ИС) по нескольким причинам:

1. Понимание пользовательских потребностей: Use Cases позволяют детально описать, как конечные пользователи будут взаимодействовать с системой для достижения своих целей. Это помогает команде разработки лучше понять и учесть потребности и ожидания пользователей.

2. Определение функциональных требований: Через сценарии использования системы можно точно определить, какие функциональные возможности должны быть реализованы, чтобы удовлетворить задачи пользователей.

3. Улучшение коммуникации: Use Cases служат общим языком для обсуждения требований между заинтересованными сторонами, включая заказчиков, разработчиков и конечных пользователей, что снижает вероятность недопонимания.

4. Предотвращение пропусков в требованиях: Детализированное описание сценариев использования помогает выявить и учесть все возможные исключительные ситуации и варианты поведения системы, минимизируя риск упущения важных аспектов при разработке.

5. Упрощение тестирования: Наличие четко определенных Use Cases упрощает разработку тестовых сценариев и помогает обеспечить тщательное тестирование всех аспектов системы.

В целом, использование Use Cases способствует более глубокому пониманию требований к ИС, обеспечивает более точное и полное описание функционала и способствует созданию качественного продукта, отвечающего потребностям пользователей.

Вопрос 25. Почему интеграция базы данных с другими системами является важной частью разработки web-приложений?

Ответ: Интеграция базы данных с другими системами является ключевой частью разработки web-приложений, поскольку она обеспечивает гладкую работу приложения в рамках более широкой экосистемы сервисов и инструментов. Это может включать интеграцию с системами управления контентом (CMS), CRM системами, платежными шлюзами, аналитическими инструментами и другими приложениями. Эффективная интеграция позволяет обмен данными между системами в реальном времени, улучшая пользовательский опыт, автоматизацию бизнес-процессов и предоставляя более полную картину для принятия решений на основе данных.

Компетенция ПК-2. Способен проектировать цифровой программный продукт.

Обучающийся знает: принципы построения и работы с базами данных и СУБД; основные алгоритмы решения задач предметной области, их особенности и характеристики; знать основы проектирования баз данных.

Обучающийся умеет: выбирать оптимальные средства решения задач, минимизировать пути решения, представлять результат; определить необходимые функциональные возможности проектируемой СУБД; строить концептуальную, логическую и физическую модели баз данных.

Обучающийся владеет: навыками разработки архитектуры web-приложения с учетом передовых ИТ-решений прикладной сферы; навыками проектирования баз данных.

ЗАДАНИЯ ЗАКРЫТОГО ТИПА (10 заданий)

Категория: знать.

Задание 1. Какой подход к проектированию баз данных сосредоточен на минимизации избыточности и предотвращении аномалий вставки, обновления и удаления?

- 1) денормализация;
- 2) нормализация;
- 3) индексирование;
- 4) шардинг;

Ответ: 2) нормализация.

Задание 2. Какая архитектура веб-приложений позволяет разделить приложение на отдельные слои, каждый из которых имеет свою специализированную функцию?

- 1) MVC (Model-View-Controller);
- 2) API-first;
- 3) Microservices;
- 4) Serverless.

Ответ: 1) MVC (Model-View-Controller)

Задание 3. Какая методология проектирования фокусируется на постепенной разработке продукта через короткие итерации?

- 1) Waterfall;
- 2) Spiral;
- 3) Agile;
- 4) RAD (Rapid Application Development).

Ответ: 3) Agile

Задание 4. Какой принцип проектирования баз данных направлен на устранение функциональных зависимостей между столбцами таблицы?

- 1) первая нормальная форма (1NF);
- 2) вторая нормальная форма (2NF);
- 3) третья нормальная форма (3NF);
- 4) четвертая нормальная форма (4NF);

Ответ: 2) Вторая нормальная форма (2NF)

Задание 5. Какой тип архитектуры веб-приложений характеризуется использованием легковесных сервисов, которые выполняют конкретные бизнес-функции и общаются через сетевые запросы?

- 1) Monolithic;
- 2) SOA (Service-Oriented Architecture);

- 3) Microservices;
- 4) PaaS (Platform as a Service).

Ответ: 3) Microservices

Задание 6. Какие языки используются для обработки данных в СУБД?

- 1) Python;
- 2) SQL;
- 3) NoSQL;
- 4) MathCad.

Ответ: 2)

Задание 7. В какой форме должны быть данные в таблице, чтобы она соответствовала первой нормальной форме (1NF)?

- 1) все значения в столбцах должны быть уникальны;
- 2) таблица должна содержать повторяющиеся группы данных;
- 3) данные в каждом столбце таблицы должны быть атомарными;
- 4) все записи должны быть доступны только по первичному ключу.

Ответ: 3) Данные в каждом столбце таблицы должны быть атомарными.

Задание 8. Как называется процесс определения структуры, компонентов, интерфейсов и других характеристик системы или компонента?

- 1) программирование;
- 2) документирование;
- 3) проектирование;
- 4) тестирование.

Ответ: 3) проектирование

Задание 9. Какой паттерн проектирования предполагает разделение приложения на три основных части: данные, пользовательский интерфейс и управляющую логику?

- 1) Singleton;
- 2) Observer;
- 3) MVC (Model-View-Controller);
- 4) Factory.

Ответ: 3) MVC (Model-View-Controller)

Задание 10. Какой метод разделения базы данных на части для улучшения ее масштабируемости и производительности?

- 1) нормализация;
- 2) репликация;
- 3) шардинг;
- 4) кластеризация.

Ответ: 3) шардинг

ЗАДАНИЯ ОТКРЫТОГО ТИПА (15 заданий)

Категория: уметь.

Задание 11. Какие факторы следует учитывать при выборе средств для разработки веб-приложения?

Ответ: При выборе средств для разработки веб-приложения важно учитывать несколько ключевых факторов. Во-первых, необходимо определить требования к проекту, включая функциональность, производительность и масштабируемость. Во-вторых, следует оценить

уровень знаний и опыта команды разработчиков в работе с предполагаемыми технологиями. В-третьих, важно рассмотреть экосистему и поддержку выбранного стека технологий, а также его способность интегрироваться с другими инструментами и сервисами. Наконец, следует учитывать бюджет проекта и общую стоимость владения разработанным решением.

Задание 12. Как минимизировать пути решения при разработке веб-приложения?

Ответ: Минимизация путей решения в процессе разработки веб-приложения достигается путем использования проверенных паттернов проектирования, готовых библиотек и фреймворков, а также применения принципов чистого кода и рефакторинга. Эффективное планирование и разбиение проекта на мелкие, управляемые модули также способствует оптимизации процесса разработки. Ключевую роль играет предварительное тестирование и использование систем контроля версий для отслеживания изменений и быстрого отката к работоспособной версии при необходимости.

Задание 13. Какие функциональные возможности необходимо определить при проектировании СУБД для веб-приложения?

Ответ: При проектировании СУБД для веб-приложения необходимо определить следующие функциональные возможности: поддержку типов данных, необходимых для хранения информации в приложении; механизмы транзакций для обеспечения целостности данных; возможности индексации для ускорения поиска данных; настройки безопасности и доступа к данным; поддержку запросов и отчетов, необходимых для работы приложения; возможности репликации и масштабирования для обеспечения высокой доступности и производительности. Также следует учесть поддержку интеграции с другими сервисами и системами.

Задание 14. Как строится концептуальная модель базы данных?

Ответ: Концептуальная модель базы данных строится на начальном этапе проектирования и представляет собой высокоуровневую абстракцию структуры данных, используемых в приложении. Она описывает основные сущности и связи между ними без привязки к конкретным технологиям хранения данных. Создание концептуальной модели начинается с анализа требований к системе и определения ключевых объектов и процессов. Для визуализации модели часто используются ER-диаграммы (Entity-Relationship), позволяющие наглядно представить структуру базы данных.

Задание 15. В чем заключается разработка логической модели базы данных?

Ответ: Разработка логической модели базы данных включает детализацию концептуальной модели с учетом спецификаций выбранной СУБД. На этом этапе определяются точные типы данных для каждого атрибута, устанавливаются правила и ограничения (constraints), такие как первичные и внешние ключи, а также создаются индексы для оптимизации запросов. Логическая модель служит мостом между абстрактным представлением данных и их физической реализацией в базе данных.

Задание 16. Как перейти от логической к физической модели базы данных?

Ответ: Переход от логической к физической модели базы данных требует принятия решений относительно физического расположения данных на диске, выбора методов хранения (например, таблицы, индексы) и оптимизации доступа к данным (например, партиционирование). На этом этапе учитываются особенности конкретной СУБД, а также требования к производительности и масштабируемости системы. Физическая модель включает создание таблиц, настройку индексов, определение политик резервного копирования и восстановления, а также другие аспекты физического размещения данных.

Задание 17. Какие методы можно использовать для представления результатов работы веб-приложения пользователю?

Ответ: Для представления результатов работы веб-приложения пользователю можно использовать различные методы, включая динамически генерируемые веб-страницы с использованием шаблонов, интерактивные графики и диаграммы, созданные с помощью JavaScript библиотек типа Chart.js или D3.js, а также таблицы и списки с возможностью сортировки и фильтрации данных. Важно обеспечить четкую и понятную пользовательскую интерфейсную часть (UI), а также гарантировать высокую скорость загрузки страниц и отзывчивость интерфейса на различных устройствах.

Категория: **владеть**.

Задание 18. Какие современные подходы используются при проектировании баз данных для веб-приложений?

Ответ: Современные подходы к проектированию баз данных включают использование NoSQL для обработки больших объемов неструктурированных данных и обеспечения горизонтального масштабирования; применение нормализации для устранения избыточности данных и повышения их целостности; использование шардинга для распределения данных по нескольким серверам; и внедрение кэширования для ускорения доступа к часто запрашиваемой информации. Также активно применяются методы оптимизации запросов и тонкая настройка параметров СУБД для повышения производительности.

Задание 19. Какие методики могут быть использованы для обеспечения безопасности веб-приложений?

Ответ: Для обеспечения безопасности веб-приложений используются методики, такие как шифрование данных на транспортном уровне с помощью SSL/TLS; реализация надежной системы аутентификации и авторизации; защита от SQL-инъекций путем использования параметризованных запросов; предотвращение атак XSS и CSRF с помощью соответствующих заголовков безопасности и валидации пользовательского ввода. Важно также регулярно проводить аудит безопасности кода и инфраструктуры, а также обновлять используемое программное обеспечение до последних версий.

Задание 20. Какие аспекты важно учитывать при выборе стека технологий для разработки веб-приложения?

Ответ: При выборе стека технологий для веб-приложения важно учитывать следующие аспекты:

1. Требования проекта: Определите, какие функции должно выполнять приложение, и выберите технологии, наилучшим образом соответствующие этим требованиям.
2. Масштабируемость: Выбирайте технологии, которые позволяют легко масштабировать приложение в будущем.
3. Безопасность: Учитывайте уровень поддержки безопасности, который предоставляют выбранные технологии.
4. Сообщество и поддержка: Предпочтение следует отдавать технологиям с активным сообществом и хорошей документацией.
5. Совместимость: Обеспечьте совместимость выбранных технологий друг с другом.
6. Производительность: Рассмотрите, как технологии справляются с нагрузкой и временем отклика.
7. Издержки: Оцените стоимость лицензий, разработки и поддержки приложения на выбранном стеке.
8. Опыт команды: Учитывайте опыт и знания вашей команды разработчиков.
9. Возможности развития: Подумайте о перспективах развития и обновления технологий.
10. Тренды рынка: Следите за текущими тенденциями в веб-разработке.

Задание 21. Какие факторы следует учитывать при проектировании пользовательского интерфейса веб-приложений?

Ответ: При проектировании пользовательского интерфейса веб-приложений следует учитывать:

1. Пользовательский опыт (UX): Создавайте интуитивно понятный и простой интерфейс.
2. Адаптивный дизайн: Обеспечьте корректное отображение на всех типах устройств.
3. Цветовая схема и брендинг: Следите за соответствием дизайна корпоративному стилю.
4. Читаемость и типографика: Используйте читаемые шрифты и обеспечьте контрастность текста и фона.
5. Навигация: Сделайте навигацию понятной и легкодоступной.
6. Интерактивные элементы: Предусмотрите обратную связь от элементов управления.
7. Скорость загрузки: Оптимизируйте ресурсы для быстрой загрузки страниц.
8. Контент: Обеспечьте качественный и актуальный контент.
9. Доступность (Accessibility): Сделайте ваше приложение доступным для людей с ограниченными возможностями.
10. Тестирование с пользователями: Регулярно проводите тестирование интерфейса с реальными пользователями.

Задание 22. Какие подходы могут быть использованы для тестирования веб-приложений?

Ответ: Для тестирования веб-приложений можно использовать следующие подходы:

1. Функциональное тестирование: Проверка соответствия функциональности требованиям спецификации.
2. Тестирование пользовательского интерфейса: Проверка элементов интерфейса на соответствие дизайну и удобству использования.
3. Тестирование совместимости: Проверка работы приложения в различных браузерах и на разных устройствах.
4. Нагрузочное тестирование: Имитация большого числа пользователей для проверки производительности приложения под нагрузкой.
5. Тестирование безопасности: Поиск уязвимостей, которые могут быть использованы для атак на приложение.
6. Юнит-тестирование: Тестирование отдельных модулей или компонентов приложения.
7. Интеграционное тестирование: Проверка взаимодействия между различными частями приложения или внешними системами.
8. Регрессионное тестирование: Проверка, что новые изменения не привели к появлению ошибок в уже протестированных частях приложения.
9. Автоматизированное тестирование: Использование специализированного программного обеспечения для автоматизации процесса тестирования.
10. A/B тестирование: Сравнение двух версий приложения для определения более эффективной.

Задание 23. Как можно обеспечить масштабируемость веб-приложений?

Ответ: Для масштабируемости веб-приложений необходимо:

1. Горизонтальное масштабирование: Добавление дополнительных серверов для распределения нагрузки.
2. Вертикальное масштабирование: Увеличение ресурсов (например, CPU, RAM) на существующих серверах.
3. Балансировка нагрузки: Распределение запросов между серверами для оптимизации производительности.
4. Использование кэширования: Снижение нагрузки на сервер за счет временного хранения часто запрашиваемых данных.
5. Оптимизация баз данных: Использование индексов, оптимизация запросов и структуры данных для ускорения работы с БД.
6. Микросервисная архитектура: Разделение приложения на независимые компоненты, что упрощает масштабирование и обновление.

7. Автоматическое масштабирование: Использование облачных сервисов, которые автоматически адаптируют ресурсы под текущую нагрузку.

8. Оптимизация кода и ресурсов: Уменьшение объема передаваемых данных и оптимизация скриптов для ускорения загрузки страниц.

9. Репликация баз данных: Распределение данных между несколькими серверами для повышения доступности и производительности.

10. Мониторинг и анализ производительности: Отслеживание показателей работы приложения для своевременного выявления и устранения узких мест.

Задание 24. Имеют ли отношение друг к другу такие понятия как "БД" и "архитектура"? Если да, то на каком этапе разработки ИС (веб-приложения).

Ответ: Да, понятия "База данных" (БД) и "архитектура" тесно связаны друг с другом в контексте разработки информационных систем (ИС), включая веб-приложения. Отношение между этими понятиями проявляется на различных этапах разработки ИС, особенно на этапах планирования, проектирования и реализации.

1. Планирование: На этом этапе определяются основные требования к информационной системе, включая необходимость в хранении данных. Решается, какая информация будет собираться, храниться и обрабатываться, что напрямую связано с выбором или проектированием архитектуры базы данных.

2. Проектирование архитектуры системы: Архитектура ИС описывает структуру системы, включая компоненты для работы с данными (базы данных, серверы приложений, клиентские приложения). На этом этапе принимаются решения о том, как будет организована работа с данными, какие технологии будут использоваться для баз данных (например, SQL или NoSQL), как будет обеспечиваться масштабируемость и безопасность данных.

3. Проектирование базы данных: Этот этап включает создание схемы базы данных, определение таблиц, связей между ними, индексов и других элементов, необходимых для эффективного хранения и доступа к данным. Проектирование БД тесно связано с архитектурой системы, так как должно учитывать потребности приложений в данных и способы их использования.

4. Реализация и развертывание: На этом этапе база данных создается на основе спроектированной схемы, а приложение разрабатывается с учетом выбранной архитектуры. Важно обеспечить интеграцию между различными компонентами системы, включая базу данных.

5. Тестирование и оптимизация: После реализации следует этап тестирования, на котором проверяется корректность работы ИС, включая производительность и надежность работы с базой данных. Возможно, потребуется оптимизация запросов к БД или изменение архитектуры системы для улучшения ее характеристик.

В целом, база данных является ключевым компонентом информационной системы, и ее проектирование и реализация тесно связаны с архитектурой всей системы. Правильное планирование и проектирование архитектуры ИС и БД обеспечивает высокую производительность, масштабируемость и удобство дальнейшей поддержки системы.

Задание 25. Какие основные принципы проектирования баз данных следует учитывать при разработке ИС (веб-приложения), чтобы обеспечить ее эффективность, масштабируемость и безопасность, исходя из существующих регламентирующих документов и стандартов?

Ответ: Принципы:

1. Нормализация данных: Нормализация помогает избежать избыточности данных и обеспечивает их целостность. Это достигается путем разделения информации на логически связанные таблицы и минимизации повторения данных. Нормализация до третьей нормальной формы (3NF) часто считается достаточной для большинства приложений, но в некоторых случаях может потребоваться дальнейшая нормализация.

2. Обеспечение целостности данных: Целостность данных подразумевает сохранение точности, согласованности и надежности данных в течение всего жизненного цикла. Это

достигается через использование ограничений целостности, таких как первичные ключи, внешние ключи, ограничения уникальности и проверки.

3. **Безопасность данных:** Безопасность является критически важным аспектом при проектировании баз данных. Необходимо реализовать механизмы аутентификации и авторизации, шифрование данных, а также управление доступом на уровне ролей для защиты данных от неавторизованного доступа.

4. **Масштабируемость:** Проектирование баз данных должно учитывать потенциальный рост системы. Это означает, что архитектура базы данных должна быть способна адаптироваться к увеличению объема данных и количества пользователей без значительного снижения производительности.

5. **Производительность:** Оптимизация запросов, индексация ключевых столбцов и правильное проектирование схемы базы данных могут значительно улучшить производительность системы. Важно учитывать эти аспекты на этапе проектирования, чтобы избежать проблем с производительностью в будущем.

6. **Резервное копирование и восстановление:** Необходимо предусмотреть стратегии резервного копирования и восстановления для обеспечения защиты данных от потери или повреждения. Разработка плана резервного копирования и восстановления является неотъемлемой частью проектирования базы данных.

7. **Соблюдение стандартов:** Следование стандартам и лучшим практикам в области разработки баз данных помогает обеспечить совместимость, унификацию подходов и упрощает поддержку системы. Включение в процесс проектирования знаний о существующих регламентирующих документах и стандартах повышает качество конечного продукта.

Компетенция ПК-3. Способен разрабатывать цифровой программный продукт.

Обучающийся знает: принципы хранения и обработки данных в базах данных; классификацию баз данных по структуре, принципы представления информации различных типов; синтаксис SQL.

Обучающийся умеет: писать и отлаживать программные коды для решения поставленных задач.

Обучающийся владеет: навыками построения поисковых запросов; навыками построения и отладки SQL-запросов; навыками формулирования и анализа результатов запросов к базам данных.

ЗАДАНИЯ ЗАКРЫТОГО ТИПА (10 заданий)

Категория: **знать.**

Задание 1. Какой подход к проектированию баз данных использует предварительно определенные запросы и структуры данных для ускорения процессов чтения данных?

- 1) нормализация;
- 2) денормализация;
- 3) шардинг;
- 4) репликация.

Ответ: 2) Денормализация

Задание 2. Что такое NoSQL база данных?

- 1) база данных, не поддерживающая SQL;
- 2) нереляционная база данных;
- 3) база данных без какой-либо схемы;
- 4) база данных, оптимизированная для чтения.

Ответ: 2) NoSQL означает «не только SQL» и относится к нереляционным базам данных, которые могут хранить данные в формате, отличном от табличного.

Задание 3. Какой SQL запрос вернет количество уникальных пользователей?

- 1) SELECT COUNT(users) FROM table;
- 2) SELECT COUNT(DISTINCT users) FROM table;
- 3) SELECT DISTINCT COUNT(users) FROM table;
- 4) SELECT UNIQUE COUNT(users) FROM table;

Ответ: 2) COUNT(DISTINCT column_name) используется для подсчета уникальных значений в столбце.

Задание 4. Какая команда SQL используется для удаления записей из таблицы без удаления самой таблицы?

- 1) DROP;
- 2) DELETE;
- 3) TRUNCATE;
- 4) REMOVE.

Ответ: 2) Команда DELETE используется для удаления записей из таблицы.

Задание 5. Какой оператор SQL используется для выборки данных из базы данных?

- 1) SELECT;
- 2) INSERT;
- 3) UPDATE;
- 4) DELETE.

Ответ: 1) SELECT

Задание 6. Какое ключевое слово используется для возвращения только различных значений?

- 1) UNIQUE;
- 2) DIFFERENT;
- 3) DISTINCT;
- 4) SINGLE.

Ответ: 3) DISTINCT

Задание 7. Какой оператор используется для фильтрации записей?

- 1) FILTER;
- 2) WHERE;
- 3) SELECT;
- 4) ORDER BY.

Ответ: 2) WHERE

Задание 8. Какой оператор SQL используется для объединения строк двух или более таблиц, основываясь на связанных столбцах между ними?

- 1) JOIN CONNECT;
- 2) COMBINE;
- 3) CONNECT;
- 4) JOIN.

Ответ: 4) JOIN

Задание 9. Какая функция SQL используется для возвращения наибольшего значения из выбранного столбца?

- 1) MAX();
- 2) TOP();
- 3) SUM();
- 4) MAX().

Ответ: 1) MAX()

Задание 10. Какое выражение используется для сортировки результатов запроса по возрастанию или убыванию?

- 1) SORT BY;
- 2) ORDER BY;
- 3) ALIGN BY;
- 4) GROUP BY.

Ответ: 2) ORDER BY

ЗАДАНИЯ ОТКРЫТОГО ТИПА (15 заданий)

Категория: уметь.

Задание 11. Какие инструменты и техники используются для отладки веб-приложений?

Ответ: Для отладки веб-приложений широко используются инструменты разработчика в браузерах, такие как Chrome DevTools или Firefox Developer Tools. Они позволяют просматривать и редактировать HTML и CSS в реальном времени, отлаживать JavaScript, анализировать производительность и сетевую активность. Также полезными являются логгирование с помощью console.log() для отслеживания значений переменных и хода выполнения программы, точки останова (breakpoints) для приостановки выполнения кода в определенных местах и шаг за шагом проходить через код, а также использование unit-тестирования для проверки отдельных функций и компонентов приложения на корректность работы.

Задание 12. Какие основные принципы следует учитывать при написании программного кода для веб-приложений?

Ответ: При написании программного кода для веб-приложений важно следовать принципам чистого кода, таким как читаемость, простота и повторное использование. Следует избегать дублирования кода, стремиться к модульности и низкой связанности компонентов, использовать понятные имена переменных и функций. Важно также учитывать безопасность данных, предотвращая уязвимости, такие как SQL-инъекции или XSS-атаки. Применение комментариев для объяснения сложной логики и следование стандартам стиля кодирования повышают его качество и облегчают поддержку.

Задание 13. Как обеспечить адаптивность веб-приложения к различным устройствам и экранам?

Ответ: Для обеспечения адаптивности веб-приложения к различным устройствам и размерам экранов используется адаптивная верстка с помощью CSS медиа-запросов, которые позволяют применять различные стили в зависимости от характеристик устройства пользователя. Важно также использовать гибкие единицы измерения (например, проценты, vw, vh) для размеров элементов, чтобы они могли масштабироваться в зависимости от размера экрана. Использование фреймворков для адаптивного дизайна, таких как Bootstrap или Foundation, может значительно упростить процесс разработки, предоставляя готовые компоненты и сетки, адаптирующиеся под разные устройства. Кроме того, важно тестировать веб-приложение на различных устройствах и разрешениях экрана, используя эмуляторы и реальные устройства, чтобы убедиться в корректности отображения и функционирования на всех целевых платформах. Внедрение принципов "мобильной первоначальности" (Mobile First) также способствует созданию более гибких и адаптивных веб-приложений, поскольку подход предполагает разработку сначала для мобильных устройств с последующей адаптацией под более крупные экраны.

Задание 14. Какие методы оптимизации производительности веб-приложений вы знаете?

Ответ: Для оптимизации производительности веб-приложений можно использовать минификацию и объединение файлов CSS и JavaScript для уменьшения количества HTTP-запросов, асинхронную загрузку скриптов для ускорения отображения содержимого страницы, оптимизацию изображений через сжатие и выбор подходящего формата. Важно также использовать кэширование на стороне клиента и сервера для быстрого доступа к повторно используемым данным, а также применять ленивую загрузку (lazy loading) для контента, который не нужен сразу при загрузке страницы. Оптимизация запросов к базе данных и использование CDN также способствуют повышению производительности.

Задание 15. Как можно обеспечить безопасность данных в веб-приложении?

Ответ: Для обеспечения безопасности данных в веб-приложении необходимо использовать HTTPS для шифрования данных между клиентом и сервером, реализовывать надежную систему аутентификации и авторизации, защищать от SQL-инъекций с помощью параметризованных запросов и ORM, предотвращать XSS-атаки путем очистки пользовательского ввода. Важно также регулярно обновлять все используемые библиотеки и фреймворки до последних версий для исправления известных уязвимостей, а также проводить аудит безопасности кода и инфраструктуры.

Задание 16. Какие подходы к тестированию веб-приложений вы считаете наиболее эффективными?

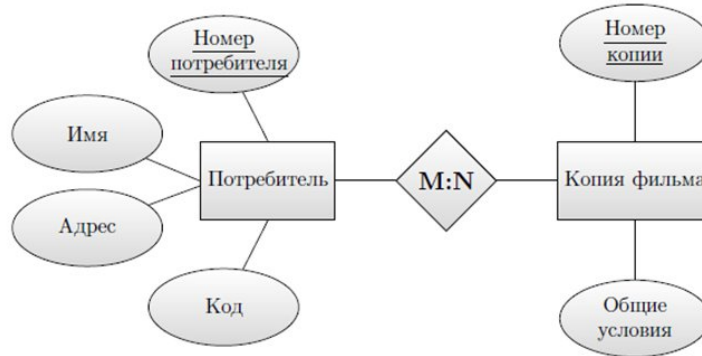
Ответ: Наиболее эффективными считаются подходы к тестированию, сочетающие различные виды тестов: модульные тесты для проверки отдельных функций и компонентов приложения на корректность работы, интеграционные тесты для проверки взаимодействия между различными частями системы, функциональные тесты для проверки соответствия приложения требованиям спецификации. Использование автоматизированных инструментов тестирования и непрерывная интеграция (CI/CD) позволяют быстро выявлять и исправлять ошибки. Также важны тестирование производительности для оценки скорости работы приложения под нагрузкой и тестирование безопасности для выявления уязвимостей.

Задание 17. Какие основные проблемы могут возникнуть при разработке веб-приложений и как их решить?

Ответ: При разработке веб-приложений часто возникают проблемы, такие как неоптимизированная производительность, сложности с обеспечением безопасности, кроссбраузерная совместимость и проблемы масштабируемости. Для решения проблемы производительности важно оптимизировать загрузку ресурсов, минимизировать размеры файлов CSS и JavaScript, а также использовать кэширование. Безопасность можно усилить путем реализации проверок на стороне сервера, шифрования данных и использования HTTPS. Кроссбраузерная совместимость достигается путем использования стандартных веб-технологий и тестирования приложения в различных браузерах. Для обеспечения масштабируемости важно предусмотреть горизонтальное масштабирование и использовать облачные решения, позволяющие легко добавлять ресурсы по мере необходимости. Эти подходы помогут справиться с основными проблемами при разработке веб-приложений.

Категория: **владеть.**

Задание 18. Проанализировать данную схему и ответить на комплекс вопросов: 1) название схемы; 2) этап разработки ИС (веб-приложения); 3) необходимость преобразования схемы; 4) возможность представления схему в не графическом формате.



Ответ:

- 1) Название схемы: ER-диаграмма (Entity-Relationship diagram).
- 2) Этап разработки ИС (веб-приложения): этап проектирования БД, подэтап – инфологическое проектирование БД. Также не будет ошибкой, если написать «концептуальное проектирование» БД. Она помогает определить и визуализировать структуру данных и связи между сущностями в системе.
- 3) Необходимость преобразования схемы: да, обычно необходимо преобразование такой схемы для реализации на практике. Связь "много-ко-многом" между сущностями "Копия фильма" и "Потребитель" требует создания дополнительной связующей таблицы в реальной базе данных для корректного отображения этих отношений. Это преобразование является частью перехода от концептуальной модели к логической и далее к физической модели базы данных.
- 4) Возможность представления схемы в не графическом формате: да, схему можно представить в не графическом формате, например, с помощью описания отношений и атрибутов сущностей в текстовом виде или через спецификации и таблицы связей. Также можно использовать языки разметки или моделирования, такие как XML или UML (Unified Modeling Language), для текстового представления структур данных и их взаимосвязей.

Задание 19. Какие типы поисковых запросов могут быть использованы при разработке веб-приложений?

Ответ: При разработке веб-приложений используются различные типы поисковых запросов для обеспечения эффективного доступа к данным. Ключевыми являются точные запросы, которые ищут совпадения по конкретному значению, и полнотекстовые запросы, позволяющие осуществлять поиск по содержанию текстовых данных. Также важны запросы с использованием регулярных выражений для сложных условий поиска и фильтрационные запросы для отбора данных по определенным критериям. Использование этих типов запросов позволяет создавать мощные и гибкие механизмы поиска в веб-приложениях.

Задание 20. Какие методы отладки SQL-запросов вы считаете наиболее эффективными?

Ответ: Для эффективной отладки SQL-запросов рекомендуется использовать несколько подходов. Во-первых, визуализация плана выполнения запроса помогает понять, как СУБД обрабатывает запрос и где возможны узкие места. Во-вторых, разбиение сложных запросов на более мелкие части позволяет выявить ошибки в логике. Также полезным является использование временных таблиц для проверки промежуточных результатов. Применение этих методов способствует более глубокому пониманию работы запросов и ускоряет процесс отладки.

Задание 21. Дана SQL-команда:

```

SELECT Disziplina, Ozenka
FROM STUDENT
WHERE STUDENT.Kod_s = (SELECT Kod_s FROM STUDENT WHERE
Fam='Абрикосов');
  
```

Описать результаты работы данной команды.

Ответ: Этот SQL-запрос выбирает фамилию (Fam) из таблицы STUDENT, где код студента (Kod_s) равен коду студента с фамилией 'Абрикосов'. Вложенный запрос (SELECT Kod_s FROM STUDENT WHERE Fam='Абрикосов') ищет код студента (Kod_s), у которого фамилия 'Абрикосов', а затем основной запрос использует этот код для выбора записей с таким же кодом, но выводиться будут значения полей «Дисциплина» и «Оценка».

Однако, если в таблице есть несколько студентов с фамилией «Абрикосов», то выводиться будут оценки по всем студентам.

Задание 22. Сопоставьте методы оптимизации SQL-запросов с их примерами.

- А) Применение индексов.
- Б) Минимизация использования JOIN.
- В) Ограничение объема возвращаемых данных.
- Г) Избегание подзапросов, где это возможно.

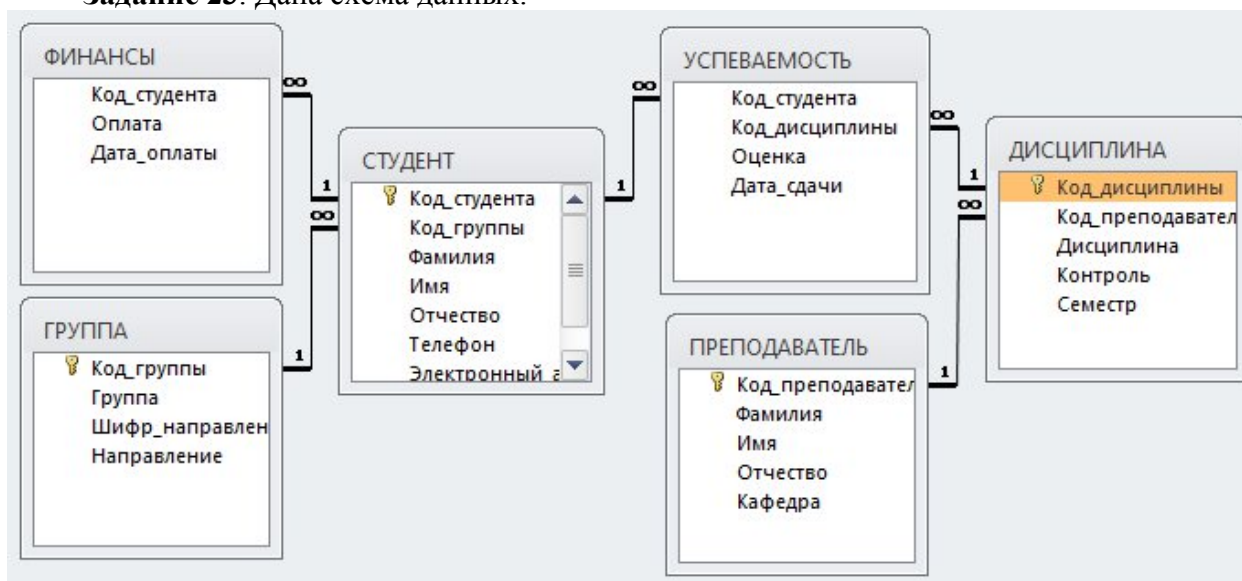
Примеры:

- 1) Использование предложения LIMIT для сокращения числа записей, возвращаемых запросом.
- 2) Создание индекса на столбце, который часто используется в предложении WHERE.
- 3) Замена подзапроса на временную таблицу или CTE для улучшения читаемости и производительности.
- 4) Переписывание запроса таким образом, чтобы объединять таблицы только при абсолютной необходимости.

Ответ:

А - 2, Б - 4, В - 1, Г - 3.

Задание 23. Дана схема данных.



Написать команду-запрос: вывести всех студентов, у которых финансовые задолженности (предполагается, что оператор работает с отфильтрованной БД за второе полугодие 2023 год, стоимость обучения в год – 120 000 р.);

Ответ:

Для вопроса 1)
SELECT СТУДЕНТ.Фамилия, Sum([Оплата]) AS Оплачено
FROM СТУДЕНТ INNER JOIN ФИНАНСЫ ON СТУДЕНТ.Код_студента =
ФИНАНСЫ.Код_студента
GROUP BY СТУДЕНТ.Фамилия
HAVING Sum([Оплата])<60000;

Задание 24. Используя схему задания 24, написать команду-запрос: вывести список студентов, данные по которым отсутствуют в таблице УСПЕВАЕМОСТЬ.

Ответ:

```
SELECT СТУДЕНТ.Код_студента, СТУДЕНТ.Фамилия  
FROM СТУДЕНТ  
WHERE Студент.Код_студента NOT IN (SELECT УСПЕВАЕМОСТЬ.Код_студента  
FROM УСПЕВАЕМОСТЬ);
```

Задание 25. Как можно оптимизировать SQL-запросы для улучшения производительности веб-приложения?

Ответ: Оптимизация SQL-запросов играет ключевую роль в повышении производительности информационных систем. Первым шагом является анализ плана выполнения запросов, что позволяет выявить наиболее затратные по времени операции. Создание индексов на полях, которые часто используются в запросах, может значительно ускорить процесс выборки данных, минимизируя время доступа к ним. Важно стремиться к сокращению количества операций JOIN, так как они могут значительно увеличивать время выполнения запроса, особенно при работе с большими объемами данных. При возможности следует избегать подзапросов, заменяя их более эффективными конструкциями, например, временными таблицами или CTE (Common Table Expressions).

Оптимизация условий в предложении WHERE играет не меньшую роль, поскольку правильно сформулированные условия могут значительно уменьшить объем обрабатываемых данных. Важно также учитывать использование функций в условиях WHERE, так как их применение к полям может привести к невозможности использования индексов. Применение агрегатных функций и группировка данных должны осуществляться с учетом необходимости обработки, чтобы не создавать дополнительную нагрузку на систему.

В целом, для оптимизации SQL-запросов необходимо тщательно анализировать их структуру, выбирать наиболее эффективные методы доступа к данным и минимизировать объем обрабатываемой информации. Постоянный мониторинг и анализ производительности запросов помогут поддерживать систему в оптимальном состоянии и обеспечат быстрый доступ к данным.

6. ЛИСТ СОГЛАСОВАНИЯ

Составил:

О.И. Подулыбина, ст. преп.



(подпись)

Заведующий кафедрой

Н.Б. Стрекалова, д.п.н., доцент



(подпись)

Заведующий выпускающей кафедрой

Н.Б. Стрекалова, д.п.н., доцент



(подпись)