

Федеральное государственное образовательное бюджетное учреждение высшего образования

**«Финансовый университет при Правительстве
Российской Федерации»
(Финансовый университет)**

Владикавказский филиал Финуниверситета

УТВЕРЖДАЮ

Заместитель директора
по учебно-методической работе
Владикавказского филиала
Финуниверситета

З. Айларова З.К. Айларова
«31» августа 2026 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

ОП.07 Основы проектирования баз данных»

по специальности 09.02.13 Интеграция решений с применением технологий
искусственного интеллекта

Владикавказ 2026

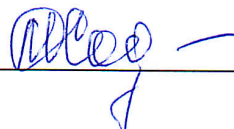
Фонд оценочных средств по дисциплине разработан на основе федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.13 Интеграция решений с применением технологий искусственного интеллекта

Составитель:

Зангиева Светлана Викторовна, преподаватель.

Фонд оценочных средств рассмотрен и рекомендован к утверждению на заседании предметной (цикловой) комиссии математики и информатики.

Протокол от «28» 08 2026 г. № 1

Председатель предметной (цикловой)
комиссии математики и информатики  М.К. Ходова

1. Паспорт фонда оценочных средств
по дисциплине **ОП.07 «Основы проектирования баз данных»**
по специальности 09.02.13 Интеграция решений с применением технологий
искусственного интеллекта

Результаты обучения (знания, умения)	Общие и профессиональные компетенции	Наименование темы	Наименование оценочного средства	
			Текущий контроль успеваемости	Промежуточная аттестация
Освоенные знания: • Основы теории баз данных и моделей данных (реляционная, иерархическая, сетевая). • Этапы и принципы проектирования БД (концептуальное, логическое, физическое). • Синтаксис и возможности языка SQL (DDL, DML, DQL). • Средства проектирования и CASE-инструменты. • Современные системы управления базами данных (СУБД) — реляционные, NoSQL, облачные решения. • Инструменты визуального проектирования (ER-диаграммы, обратное инжиниринг) и CASE-средства. • Методы извлечения данных с помощью SQL и инструментов бизнес-аналитики. • Ресурсы для поиска профессиональной информации (документация СУБД, стандарты SQL, форумы, научные статьи). • Принципы работы с большими объёмами данных и оптимизации запросов (индексы, планы выполнения). • Знание рынка труда в сфере ИТ, востребованности навыков проектирования БД и SQL. • Основы законодательства в части защиты персональных	ОК.01, ОК.02, ОК.03, ОК.04, ОК.05	Раздел 1. Основы проектирования баз данных Тема 1.1. Введение в базы данных	Вопросы для устного (письменного) опроса по теме Практические занятия – решение практических ситуационных заданий. Задание для самостоятельной работы	Экзаменационные вопросы и задания для устного (письменного) экзамена по дисциплине
		Тема 1.2. Концептуальное проектирование баз данных	Вопросы для устного (письменного) опроса по теме Практические занятия – решение практических ситуационных заданий. Задание для самостоятельной работы Тест по разделу 1	

данных и ответственности за их утечку (№ 152-ФЗ, GDPR). - Знание того, как грамотно спроектированная БД может стать конкурентным преимуществом бизнеса. - Роли и этапы командной работы над проектом БД: аналитик, архитектор, разработчик, администратор. - Методологии совместной разработки (Agile, Scrum) и их применение к задачам проектирования. - Средства совместной работы (системы контроля версий для схем БД, общие документы, доски задач). - Знание принципов документирования и кодирования, облегчающих понимание коллегами. - Профессиональная терминология в области баз данных на русском языке (сущность, атрибут, отношение, нормализация, транзакция и т.д.). - Правила составления технической документации (пояснительные записки, руководства пользователя, спецификации). - Особенности делового общения при защите проекта, презентации результатов. - Культурные особенности восприятия технических текстов в российской деловой среде.	OK.01, OK.02, OK.03, OK.04, OK.05	Раздел 2. Логическое и физическое проектирование баз данных Тема 2.1.Логическое проектирование баз данных	Вопросы для устного (письменного) опроса по теме Практические занятия – решение практических ситуационных заданий. Задание для самостоятельной работы	Экзаменационные вопросы и задания для устного (письменного) экзамена по дисциплине
		Тема 2.2. Физическое проектирование баз данных запросами	Вопросы для устного (письменного) опроса по теме Практические занятия – решение практических ситуационных заданий. Тест по разделу 2 Задание для самостоятельной работы	

Освоенные умения: • Сравнивать альтернативные схемы БД и выбирать наилучшую с точки зрения целостности, производительности и масштабируемости. • Составлять запросы SQL разной сложности (вложенные, объединения, группировки) для решения конкретных информационных потребностей. • Оценивать влияние выбранной структуры на скорость выполнения операций и корректировать проект под изменяющиеся требования. • Осуществлять поиск технической документации по конкретной СУБД и интерпретировать её для решения практических задач. • Использовать современные IDE для работы с БД для выполнения запросов и анализа результатов. • Применять средства визуализации схем и анализа зависимостей для понимания структуры существующей БД. • Интерпретировать результаты выполнения запросов и статистику работы СУБД для выявления узких мест. • Критически оценивать информацию, найденную в интернете, и адаптировать её под свою задачу. • Самостоятельно ставить цели по углублению знаний в конкретных СУБД или направлениях (NoSQL, Data Warehousing). • Оценивать временные и финансовые затраты на разработку и поддержку БД в рамках проекта. • Соблюдать правовые нормы при проектировании — разграничивать доступ, обеспечивать аудит, соблюдать требования к хранению.	ОК.01, ОК.02, ОК.03, ОК.04, ОК.05	Раздел 3. Основы работы с запросами и оптимизация баз данных Тема 3.1. Основы SQL и работа с запросами Вставка графических объектов в текстовый документ.	Вопросы для устного (письменного) опроса по теме Практические занятия – решение практических ситуационных заданий. Задание для самостоятельной работы	Экзаменационные вопросы и задания для устного (письменного) экзамена по дисциплине (рефератом) Экзамен.
		Тема 3.2 Оптимизация запросов и работы баз данных	Вопросы для устного (письменного) опроса по теме Практические занятия – решение практических ситуационных заданий. Тест по разделу 3 Задание для самостоятельной работы	

• Планировать своё профессиональное развитие, выбирая курсы, стажировки и сертификации по тематике БД. • Участвовать в командных обсуждениях предметной области и вырабатывать согласованную концептуальную модель. • Распределять задачи между участниками (например, разные модули БД) и согласовывать интерфейсы. • Выполнять ревью схем и кода SQL, давать конструктивную обратную связь. • Использовать общие репозитории для хранения скриптов миграций и документации. • Адаптировать свой стиль работы под общие стандарты команды (нейминг, оформление запросов). • Грамотно и чётко формулировать технические задания, отчёты о проектировании и сопроводительные письма. • Устно защищать принятые проектные решения, используя профессиональную лексику и понятные для заказчика формулировки. • Писать документацию по БД с учётом уровня подготовки разных групп пользователей (от ИТ-специалистов до менеджеров). • Приводить примеры, иллюстрирующие влияние проектных решений на бизнес-процессы, с учётом социокультурного контекста (законодательство, традиции делового оборота). Вести рабочие переписки и обсуждения в общепринятом деловом тоне.				
---	--	--	--	--

2. Комплект оценочных средств

1.Задания для текущего контроля успеваемости

Вопросы для устного (письменного) опроса по темам.
(ОК.01, ОК.02, ОК.03, ОК.09)

Раздел 1. Основы проектирования баз данных

Тема 1.1. Введение в базы данных

- 1.Понятие базы данных, её роль и применение.
- 2.Основы реляционной модели данных.

Тема 1.2. Концептуальное проектирование баз данных

- 1.Основы концептуального проектирования
2. ER-диаграммы, основные сущности и связи

Раздел 2. Логическое и физическое проектирование баз данных

Тема 2.1. Логическое проектирование баз данных

- 1.Переход от концептуальной модели к логической
2. Использование первичных и внешних ключей

Тема 2.2. Физическое проектирование баз данных

- 1.Оптимизация структуры таблиц
2. Создание индексов
3. Настройка хранилища данных.

Раздел 3. Основы работы с запросами и оптимизация баз данных

Тема 3.1. Основы SQL и работа с запросами

- 1.Основы языка SQL
2. Создание таблиц
3. Выполнение основных запросов (SELECT, INSERT, UPDATE, DELETE).

Тема 3.2 Оптимизация запросов и работы баз данных

- 1.Основы оптимизации запросов
2. Анализ планов выполнения запросов
3. Настройка производительности.

Критерии оценки ответов студента в процессе устного (письменного) опроса

Оценка «отлично» выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания вопросов программного материала.

Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе некоторые неточности.

Оценка «удовлетворительно» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом показавшему владение основными понятиями, выносимых на рассмотрение вопросов, необходимыми для дальнейшего обучения и способность применять полученные знания по образцу в стандартной ситуации.

Оценки «неудовлетворительно» заслуживает студент, который не знает большей части основного содержания выносимых на рассмотрение вопросов дисциплины (темы), допускает грубые ошибки в формулировках основных понятий.

2.1.3. Практические задания (ОК.01, ОК.02, ОК.03, ОК.04, ОК.05)

Раздел 1. Основы проектирования баз данных

Тема 1.1. Введение в базы данных

1. Практическое занятие: «Анализ структуры базы данных на примере реальной системы.»

Исходная ситуация

Перед вами фрагмент структуры базы данных **интернет-магазина**. База данных состоит из нескольких таблиц, которые хранят информацию о клиентах, товарах, категориях товаров и заказах.

Описание таблиц:

Таблица customers (Клиенты)

Хранит информацию о клиентах магазина.

`customer_id` — уникальный идентификатор клиента (первичный ключ).

`customer_name` — имя клиента.

`email` — адрес электронной почты.

Таблица categories (Категории)

Хранит список категорий товаров.

`category_id` — уникальный идентификатор категории (первичный ключ).

`category_name` — название категории.

Таблица products (Товары)

Хранит информацию о товарах.

product_id — уникальный идентификатор товара (первичный ключ).

product_name — название товара.

price — цена товара.

category_id — идентификатор категории, к которой относится товар (внешний ключ к таблице categories).

Таблица orders (Заказы)

Хранит информацию о заказах.

order_id — уникальный идентификатор заказа (первичный ключ).

customer_id — идентификатор клиента, оформившего заказ (внешний ключ к таблице customers).

order_date — дата оформления заказа.

Таблица order_items (Позиции заказа)

Хранит информацию о товарах в каждом заказе.

order_item_id — уникальный идентификатор позиции в заказе (первичный ключ).

order_id — идентификатор заказа (внешний ключ к таблице orders).

product_id — идентификатор товара (внешний ключ к таблице products).

quantity — количество единиц товара.

Задание

Часть 1. Определение элементов структуры (5 минут)

Ответьте на следующие вопросы, используя описание базы данных:

Сколько таблиц содержит данная база данных? Перечислите их названия.

Для таблицы products укажите:

Первичный ключ.

Внешний ключ и название таблицы, на которую он ссылается.

Для таблицы order_items укажите:

Первичный ключ.

Все внешние ключи и названия таблиц, на которые они ссылаются.

Какая таблица является связующей между orders и products?

Часть 2. Анализ связей между таблицами (5 минут)

Определите тип связи (1:1, 1:M или M:M) для каждой из следующих пар таблиц:

customers и orders

categories и products

orders и order_items

products и order_items

Для каждой связи кратко поясните свой ответ.

Часть 3. Практическое задание «Запрос к базе данных» (5 минут)

На основе структуры базы данных напишите **словесное описание** (не SQL-запрос) того, какие таблицы и как нужно объединить, чтобы получить следующий отчет:

«Список всех заказов с указанием имени клиента, даты заказа, названия каждого заказанного товара и его количества.»

Опишите последовательность действий (какие таблицы соединить и по каким полям).

Часть 4. Рефлексия (5 минут)

Ответьте на вопросы:

Зачем в таблице `order_items` хранить `product_id` и `quantity` отдельно от таблицы `orders`? Почему нельзя добавить поля с товарами прямо в таблицу `orders`?

Какую информацию о клиенте, кроме имени и email, вы бы предложили добавить в таблицу `customers` для более эффективной работы магазина? Приведите 2–3 примера полей.

Форма выполнения

Результаты оформите в виде **письменного отчёта**, содержащего ответы на все вопросы из Частей 1–4.

2. Практическое занятие: «Разработать концепцию базы данных и создать блок-схему, отражающую структуру и связи между таблицами».

Исходная ситуация

Компания «СтройПроект» занимается строительством жилых и коммерческих объектов. Руководство приняло решение разработать информационную систему для управления проектами. Вам поручено разработать **концептуальную модель базы данных** для этой системы.

Требования к системе:

Система должна хранить информацию о **проектах** (название, адрес, дата начала, дата окончания, бюджет).

В каждом проекте участвуют **сотрудники** (ФИО, должность, телефон, email).

Каждый сотрудник может быть назначен **руководителем проекта** (один сотрудник — руководитель одного или нескольких проектов).

В рамках проектов выполняются **работы** (название, описание, плановая стоимость, фактическая стоимость, статус).

Каждая работа связана с одним проектом.

Сотрудники могут быть назначены на **несколько работ** в рамках разных проектов.

Для каждой работы может быть назначен один **ответственный сотрудник**.

Задание

Часть 1. Выделение сущностей и атрибутов (5 минут)

На основе описания предметной области определите **не менее 4 сущностей**. Для каждой сущности укажите:

Название сущности.

Набор атрибутов (не менее 3).

Первичный ключ (РК).

Формат оформления:

Сущность	Атрибуты	Первичный ключ (РК)
Проект	Название, Адрес, Дата_начала, Дата_окончания, Бюджет	id_проекта
...	...	...

Часть 2. Определение связей между сущностями (5 минут)

Определите **тип связи** между каждой парой сущностей (1:1, 1:M, M:M) и укажите, в какой таблице будет располагаться внешний ключ (FK) для реализации этой связи.

Заполните таблицу:

Связь	Тип связи	Где располагается FK	Пояснение
Проект — Сотрудник (руководитель)			
Проект — Работа			
Сотрудник — Работа (ответственный)			
Сотрудник — Работа (участие)			

Дополнительно: Для связи «Сотрудник — Работа (участие)» укажите, нужна ли дополнительная таблица (ассоциативная) и почему.

Часть 3. Построение блок-схемы (ER-диаграммы) (5 минут)

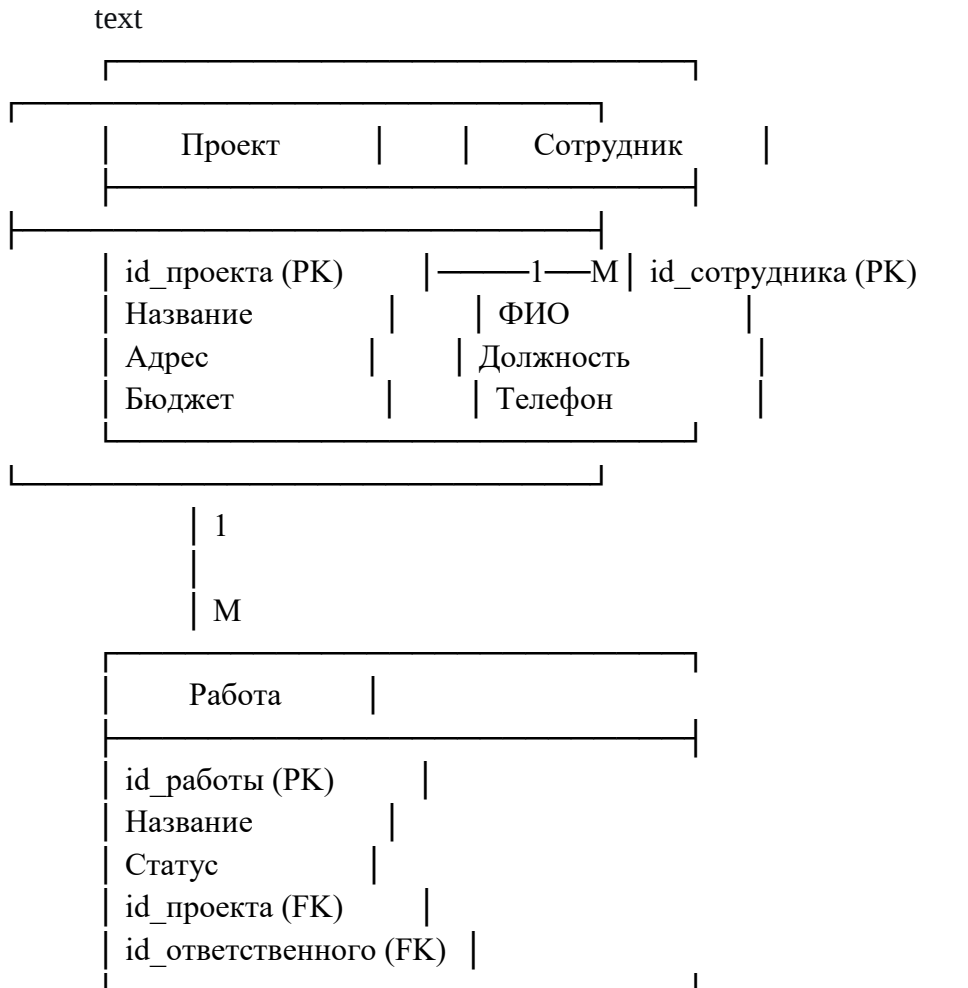
На основе определённых сущностей и связей **нарисуйте блок-схему базы данных** (можно схематично, от руки). На схеме отразите:

Прямоугольники для каждой таблицы с названием сущности.

Внутри прямоугольника перечислите **ключевые поля** (PK, FK).

Линии, соединяющие таблицы, с указанием типа связи (1:1, 1:M, M:M).

Пример оформления (текстовый вариант):



Часть 4. Анализ и рефлексия (5 минут)

Ответьте на вопросы:

Почему для связи «Сотрудник — Работа» требуется две разные связи («ответственный» и «участие»)? Как это отражено в структуре базы данных?

Какие проблемы могут возникнуть, если не разделить сотрудников на руководителей и исполнителей, а хранить все роли в одной таблице?

Какие дополнительные сущности или атрибуты вы бы добавили в базу данных для более полного учёта строительных проектов (например, поставщики, материалы, этапы строительства)?

Ответы на вопросы (Часть 4).

Форма выполнения

Результаты оформите в виде **письменного отчёта**, содержащего:

Заполненную таблицу сущностей и атрибутов (Часть 1).

Заполненную таблицу связей (Часть 2).

Блок-схему (ER-диаграмму) — рисунок или текстовое описание (Часть 3).

3. Практическое занятие: «Построение ER-диаграммы для простой предметной области»

Исходная ситуация

Предметная область: Учёт заказов в кафе

Небольшое кафе «Уют» хочет автоматизировать учёт заказов. Система должна хранить информацию о:

Блюдах – название, цена, категория (горячее, салаты, напитки и т.п.), время приготовления.

Заказах – номер заказа, дата и время, статус (новый, готовится, выполнен), столик.

Клиентах – если клиент постоянный, то хранится имя и телефон; для разовых клиентов – только номер столика.

Составе заказа – какие блюда и в каком количестве входят в каждый заказ.

Важно: один заказ может включать несколько блюд, а одно блюдо может присутствовать в разных заказах.

Задание

Часть 1. Определение сущностей, атрибутов и связей (10 минут)

На основе описания предметной области выполните следующие шаги:

Выделите все сущности (не менее 3). Для каждой сущности укажите:

Название сущности.

Атрибуты (включая первичный ключ).

Первичный ключ.

Определите связи между сущностями. Для каждой пары сущностей укажите:

Тип связи (1:1, 1:M, M:M).

Где будет располагаться внешний ключ (FK) для реализации связи.

Форма выполнения – заполните две таблицы:

Таблица 1. Сущности и атрибуты

Сущность	Атрибуты	Первичный ключ (PK)
Блюдо	Название, Цена, Категория, Время_приготовления	id_блюда
...	...	...

Таблица 2. Связи между сущностями

Связь	Тип связи	Где располагается FK	Пояснение
Заказ — Блюдо	M:M	В ассоциативной таблице «Состав_заказа»	Один заказ содержит много блюд, одно блюдо может быть в разных заказах
...	...	...	...

Часть 2. Построение ER-диаграммы (10 минут)

На основе определённых сущностей и связей нарисуйте **ER-диаграмму** (можно от руки, на листе). На схеме отразите:

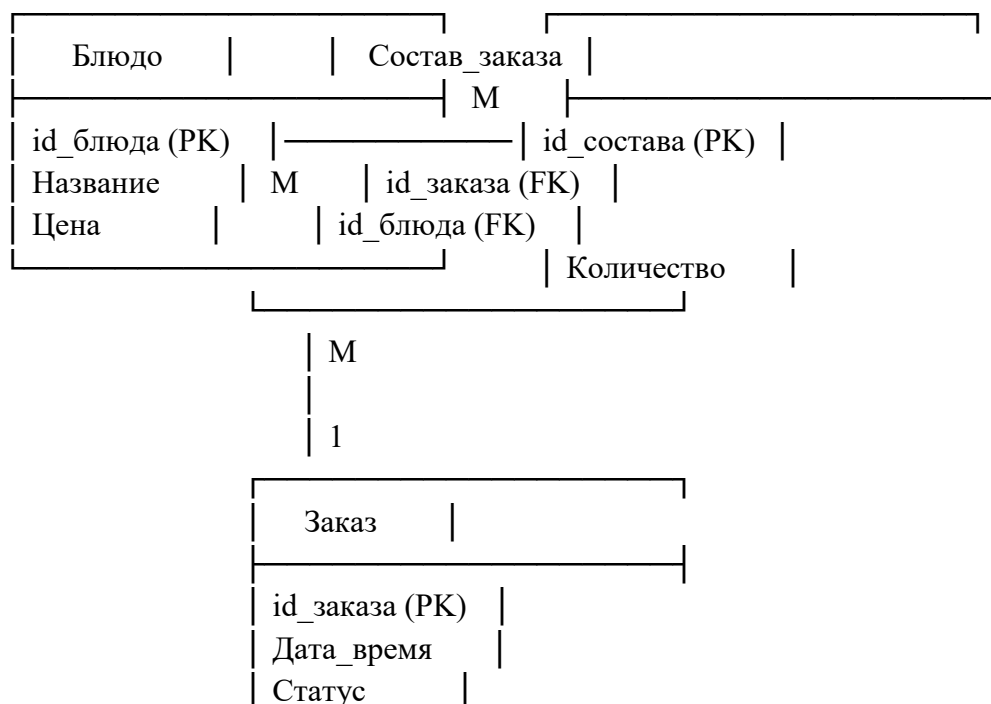
Прямоугольники для каждой сущности с названием.

Внутри прямоугольника перечислите **ключевые поля** (PK и FK).

Линии, соединяющие сущности, с указанием типа связи (1, M).

Упрощённый пример оформления:

text





Форма выполнения

Результаты оформите на **одном листе** (или в текстовом файле), содержащем:

Заполненные таблицы (Часть 1).

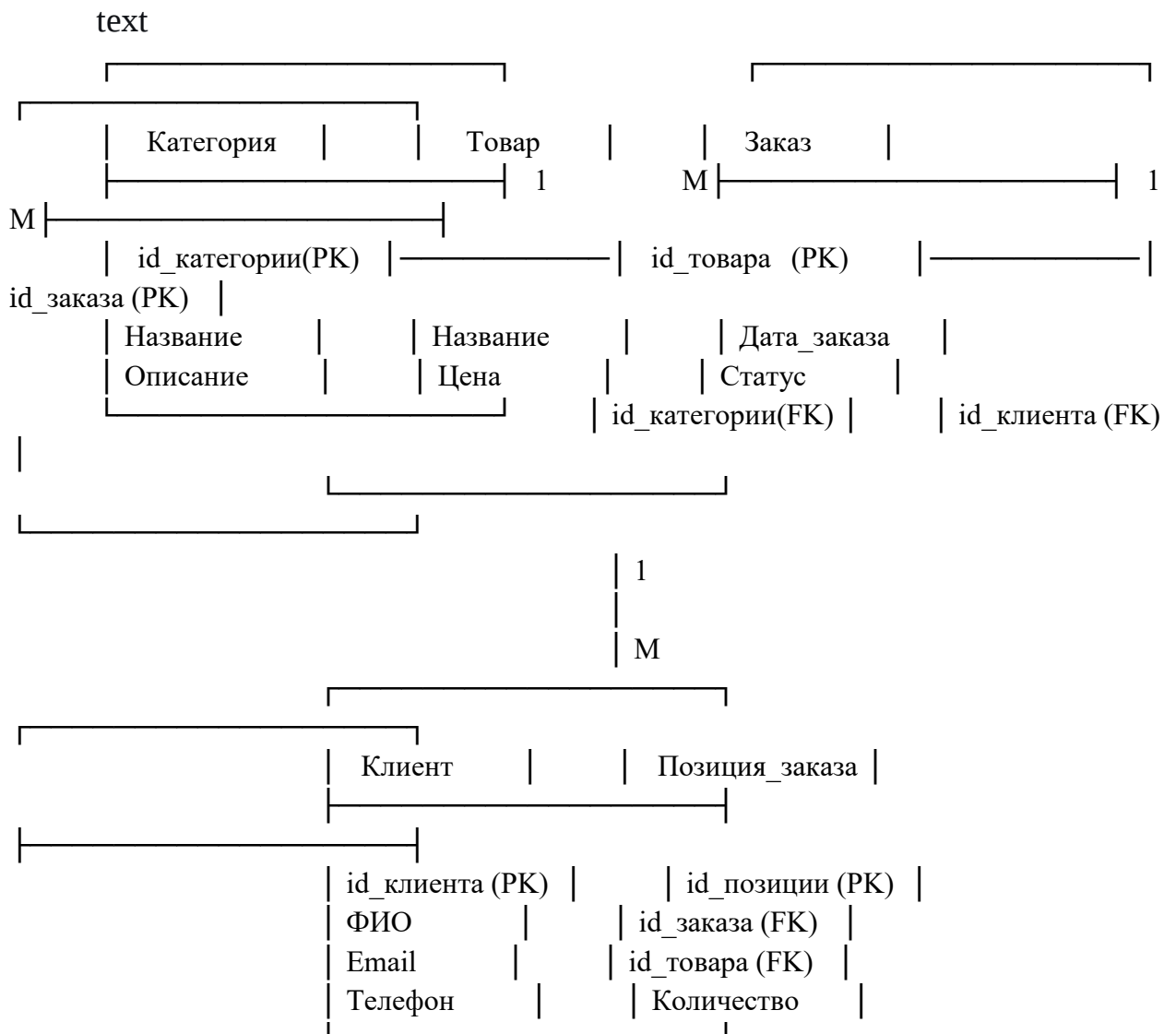
ER-диаграмму (Часть 2) – рисунок или текстовое описание.

4. Практическое занятие «Разработка реляционной схемы базы данных на основе ER-диаграммы»

Исходная ситуация

Вам представлена **ER-диаграмма** для системы учёта заказов в интернет-магазине (см. схему ниже). Ваша задача — преобразовать её в реляционную схему базы данных.

ER-диаграмма:



Описание ER-диаграммы:

Категория — содержит список категорий товаров (один товар относится к одной категории).

Товар — содержит информацию о товарах (цена, название, категория).

Клиент — содержит информацию о клиентах.

Заказ — содержит информацию о заказе (дата, статус, клиент).

Позиция_заказа — связывает заказы и товары с указанием количества (связь М:М между Заказом и Товаром).

Задание

Часть 1. Преобразование ER-диаграммы в реляционную схему (10 минут)

На основе ER-диаграммы составьте **реляционную схему базы данных**. Для каждой таблицы укажите:

Название таблицы.

Список столбцов с указанием типов данных (используйте **INTEGER**, **VARCHAR**, **DECIMAL**, **DATE**).

Первичный ключ (PK).

Внешние ключи (FK) с указанием ссылочной таблицы.

Форма выполнения — заполните таблицу:

Таблица	Столбцы (с типами)	PK	FK (ссылка)
Категория	id_категории INTEGER, Название VARCHAR (255), Описание TEXT	id_категории	—
Товар	...	...	id_категории → Категория
Клиент	...	...	—
Заказ	...	...	...
Позиция_заказа	...	...	...

Часть 2. Описание связей между таблицами (10 минут)

Для каждой пары таблиц, связанных внешним ключом, опишите:

Тип связи (1:1, 1:M, M:M).

Какая таблица является родительской (главной), а какая — дочерней (подчинённой).

Какое бизнес-правило реализует эта связь.

Форма выполнения — заполните таблицу:

Связь	Тип связи	Родительская таблица	Дочерняя таблица	Бизнес-правило
Категория — Товар	1:M	Категория	Товар	Каждый товар относится только к одной категории
Товар — Позиция_заказа	...	...	...	...
Заказ — Позиция_заказа	...	...	...	...
Клиент — Заказ	...	...	...	...

Форма выполнения

Результаты оформите на **одном листе** (или в текстовом файле), содержащем:

Заполненную таблицу с реляционной схемой (Часть 1).

Заполненную таблицу с описанием связей (Часть 2).

5. Практическое занятие: «Создание базы данных с несколькими связанными таблицами».

Исходная ситуация

Вам необходимо создать базу данных для **онлайн-библиотеки**. База данных должна содержать информацию о книгах, авторах, жанрах и выдаче книг читателям.

Структура базы данных (реляционная схема):

text





Описание схемы:

Жанр — содержит список жанров книг (например, «Роман», «Детектив»). Один жанр может быть у нескольких книг.

Книга — содержит информацию о книгах (название, год издания, жанр). Каждая книга относится к одному жанру.

Автор — содержит информацию об авторах (имя, страна).

Книга_Автор — связующая таблица для связи M:M между книгами и авторами (одна книга может иметь несколько авторов, один автор может написать несколько книг).

Задание

Часть 1. Создание структуры базы данных (10 минут)

Напишите SQL-скрипты `CREATE TABLE` для всех четырёх таблиц (Жанр, Книга, Автор, Книга_Автор). Учтите следующие требования:

Для каждой таблицы определите **первичный ключ**.

Для таблиц, содержащих внешние ключи, добавьте ограничения `FOREIGN KEY`.

Для обязательных полей добавьте ограничение `NOT NULL`.

Для поля **Название** в таблице **Книга** добавьте ограничение `UNIQUE` (названия книг не должны повторяться).

Используйте подходящие типы данных.

Часть 2. Заполнение базы данных тестовыми данными (10 минут)

Напишите SQL-скрипты `INSERT INTO` для заполнения всех таблиц тестовыми данными (минимум по 3 записи в каждой таблице). Данные придумайте самостоятельно, но они должны быть логически связаны (например, книга должна ссылаться на существующий жанр и автора).

Пример для таблицы Жанр:

sql

```
INSERT INTO Жанр (id_жанра, Название) VALUES (1, 'Роман');
```

```
INSERT INTO Жанр (id_жанра, Название) VALUES (2, 'Детектив');
```

INSERT INTO Жанр (id_жанра, Название) **VALUES** (3, 'Фантастика');

Форма выполнения

Результаты оформите в виде текстового файла или на листе, содержащего:

SQL-скрипты **CREATE TABLE** для всех таблиц (Часть 1).

SQL-скрипты **INSERT INTO** для заполнения таблиц тестовыми данными (Часть 2).

Тема 1.2. Концептуальное проектирование баз данных

1. Практическое занятие: «Определение сущностей и атрибутов для заданной предметной области»

Исходная ситуация

Предметная область: Спортивный клуб

Фитнес-клуб «Энергия» ведёт учёт своих клиентов и посещений. В клубе есть несколько тренировочных залов и групповых программ.

Известно, что клуб хранит информацию о:

Клиентах — их имя, контактный телефон, дата рождения и дата первого посещения.

Абонементы — тип абонемента (например, «Месячный», «Годовой»), стоимость, дата начала и дата окончания действия. У каждого клиента может быть несколько абонементов (если он продлевает), но в каждый момент времени действует только один.

Тренировках — название тренировки (например, «Йога», «Силовая»), продолжительность в минутах, максимальное количество участников.

Посещениях — фиксируется, какой клиент, какую тренировку и когда посетил.

Задание

Часть 1. Выделение сущностей и атрибутов (10 минут)

На основе описания предметной области выделите **сущности** и для каждой из них определите **атрибуты** и **первичный ключ**.

Заполните таблицу:

Сущность	Атрибуты	Первичный ключ (PK)
Клиент	Имя, Телефон, Дата_рождения, Дата_первого_посещения	id_клиента

Сущность	Атрибуты	Первичный ключ (PK)
...	...	...

Часть 2. Определение связей между сущностями (10 минут)

Установите связи между выделенными сущностями. Для каждой пары сущностей укажите:

Тип связи (1:1, 1:M, M:M).

В какой таблице будет располагаться внешний ключ (FK).

Заполните таблицу:

Связь	Тип связи	Где располагается FK	Пояснение
Клиент — Абонемент	1:M	В таблице Абонемент (id_клиента)	Один клиент может иметь несколько абонементов
...	...	...	...

Форма выполнения

Результаты оформите на **одном листе** (или в текстовом файле), содержащем:

Заполненную таблицу сущностей и атрибутов (Часть 1).

Заполненную таблицу связей (Часть 2).

2. Практическое занятие: «Разработка базы данных с учетом предметной области и требований целевой системы».

Исходная ситуация

Предметная область: Управление мероприятиями

Компания «Событие» организует корпоративные мероприятия (конференции, семинары, тренинги). Руководство приняло решение разработать информационную систему для учёта мероприятий, участников и спикеров.

Требования к системе:

Система должна хранить информацию о **мероприятиях**: название, дата проведения, место проведения, тематика, максимальное количество участников.

В системе должны быть зарегистрированы **участники**: ФИО, email,

телефон, компания, должность.

Каждое мероприятие имеет одного или нескольких **спикеров**: ФИО, компания, тема выступления, контактный email.

Участники **регистрируются** на мероприятия. Один участник может зарегистрироваться на несколько мероприятий.

Для каждой регистрации фиксируется: дата регистрации, статус (подтверждена/ожидает/отменена), отметка об оплате.

Задание

Часть 1. Выделение сущностей и атрибутов (10 минут)

На основе описания предметной области и требований системы:

Выделите **все сущности** (не менее 3).

Для каждой сущности укажите **атрибуты** (не менее 3 на сущность) и **первичный ключ**.

Форма выполнения — заполните таблицу:

Сущность	Атрибуты	Первичный ключ (PK)
Мероприятие	Название, Дата, Место, Тематика, Макс_участников	id_мероприятия
...	...	...

Часть 2. Определение связей и внешних ключей (10 минут)

Установите связи между выделенными сущностями. Для каждой пары сущностей укажите:

Тип связи (1:1, 1:M, M:M).

В какой таблице будет располагаться внешний ключ (FK) для реализации этой связи.

Форма выполнения — заполните таблицу:

Связь	Тип связи	Где располагается FK	Пояснение
Мероприятие — Спикер	1:M	Спикер.id_мероприятия	Одно мероприятие может иметь несколько спикеров
...	...	...	...

Форма выполнения

Результаты оформите на **одном листе** (или в текстовом файле),

содержащем:

Заполненную таблицу сущностей и атрибутов (Часть 1).

Заполненную таблицу связей (Часть 2).

3. Практическое занятие: «Построение сложной ER-диаграммы с учётом нормализации».

Исходная ситуация

Предметная область: Онлайн-платформа для обучения

Вы разрабатываете базу данных для онлайн-платформы, на которой размещены образовательные курсы. Платформа должна хранить информацию о курсах, студентах, преподавателях, модулях курсов и успеваемости студентов.

Описание предметной области:

Курс имеет название, описание, уровень сложности (начальный, средний, продвинутый), стоимость и дату создания.

Преподаватель (автор курса) имеет имя, email, специализацию и рейтинг. Один преподаватель может вести несколько курсов.

Студент имеет имя, email, дату регистрации и статус (активный, заблокирован).

Курс состоит из нескольких **модулей**. Каждый модуль имеет название, порядковый номер в курсе и длительность в часах.

Студент может записаться на курс. Для каждой записи фиксируется дата записи, статус (активен, завершён, отменён) и итоговая оценка (если курс завершён).

Один студент может записаться на несколько курсов.

Дополнительное условие:

В системе также хранится информация о **сертификатах**, которые выдаются студентам после успешного завершения курса. Сертификат содержит: уникальный номер, дату выдачи, id студента, id курса, и текстовое поздравление (может быть длинным).

Задание

Часть 1. Построение ER-диаграммы (10 минут)

На основе описания предметной области выделите **все сущности** и **связи** между ними. Постройте ER-диаграмму (можно в текстовом виде, от руки или с использованием обозначений). На схеме отразите:

Названия сущностей (прямоугольники).

Первичные ключи (PK) для каждой сущности.

Внешние ключи (FK) со ссылками на родительские таблицы.

Типы связей между сущностями (1:1, 1:M, M:M).

Форма выполнения — опишите сущности и связи:

Сущности и их атрибуты:

Сущность	Атрибуты (с указанием PK/FK)
Курс	id_курса (PK), Название, Описание, Уровень, Стоимость, Дата_создания, id_преподавателя (FK)
...	...

Связи между сущностями:

Связь	Тип	Пояснение
Преподаватель — Курс	1:M	Один преподаватель ведёт несколько курсов
...	...	...

Часть 2. Нормализация таблиц (10 минут)

Проанализируйте предложенную **ненормализованную таблицу Записи_на_курсы** и приведите её к 3НФ, разделив на несколько связанных таблиц.

Исходная ненормализованная таблица Записи_на_курсы:

id_записи	Студент	Email_студента	Телефон_студента	Курс	Преподаватель	Email_преподавателя	Дата_записи	Оценка
1	Иванов И.	ivan@mail.ru	89001234567	SQL	Петров П.	petrov@mail.ru	10.01.2024	85
2	Иванов И.	ivan@mail.ru	89001234567	Python	Сидоров С.	sidorov@mail.ru	15.02.2024	92
3	Петрова А.	petrova@mail.ru	89007654321	SQL	Петров П.	petrov@mail.ru	20.01.2024	78

Задание:

Выявите все нарушения нормализации в данной таблице (избыточность, повторяющиеся данные, транзитивные зависимости).

Разделите таблицу на несколько (не менее 3) нормализованных таблиц, каждая из которых будет находиться в 3НФ.

Для каждой полученной таблицы укажите **первичный**

ключ и внешние ключи (со ссылками).

Форма выполнения

Результаты оформите на **одном листе** (или в текстовом файле), содержащем:

Описание сущностей и связей (Часть 1).

Предложения по нормализации с указанием новых таблиц и ключей (Часть 2).

4. Практическое занятие: «Создание блок-схемы базы данных в редакторе (например, draw.io, Lucidchart и т.п.)»

Исходная ситуация

Предметная область: Онлайн-библиотека

Вы разрабатываете базу данных для онлайн-библиотеки. Система должна хранить информацию о книгах, авторах, издательствах и выдаче книг читателям.

Описание предметной области:

Книга имеет уникальный инвентарный номер, название, год издания и количество доступных экземпляров. Каждая книга относится к одному издательству.

Автор имеет уникальный идентификатор, имя и страну. Одна книга может быть написана несколькими авторами, один автор может написать несколько книг.

Издательство имеет уникальный идентификатор, название и город. Одно издательство может выпускать много книг.

Читатель имеет уникальный номер читательского билета, ФИО, телефон и дату регистрации.

Выдача фиксирует факт выдачи книги читателю. Для каждой выдачи указывается: какая книга, какой читатель, дата выдачи, дата возврата (если возвращена) и статус («На руках» или «Возвращена»). Один читатель может иметь несколько выдач, одна книга может выдаваться **多次** (в разные периоды).

Задание

Часть 1. Подготовка (2 минуты)

Откройте один из редакторов:

draw.io (рекомендуется): перейдите на сайт app.diagrams.net, нажмите «Создать новую диаграмму» и выберите «Пустой» шаблон.

Lucidchart: перейдите на сайт lucidchart.com, авторизуйтесь и создайте

новый пустой документ.

В draw.io в левой панели найдите раздел «Entity Relation» (или «ER») — там находятся готовые фигуры для таблиц и связей.

Часть 2. Создание ER-диаграммы (18 минут)

На основе описания предметной области выполните в редакторе следующие шаги:

Разместите все сущности (таблицы):

Используйте фигуру «Table» (в разделе «Entity Relation») для каждой из пяти сущностей:

Книга

Автор

Издательство

Читатель

Выдача

Заполните атрибуты внутри каждой таблицы:

Для каждой сущности укажите **первичный ключ (PK)** и **атрибуты**:

Таблица	Атрибуты
Книга	id_книги (PK), Название, Год_издания, Количество_экземпляров, id_издательства (FK)
Автор	id_автора (PK), Имя, Страна
Издательство	id_издательства (PK), Название, Город
Читатель	id_читателя (PK), ФИО, Телефон, Дата_регистрации
Выдача	id_выдачи (PK), id_книги (FK), id_читателя (FK), Дата_выдачи, Дата_возврата, Статус

Установите связи между таблицами:

Используйте инструмент для рисования линий / связей (в draw.io — это стрелка в левой панели или функция «Connect»). Для каждой связи укажите **тип связи** (1:1, 1:M, M:M) с использованием нотации «воронья лапка» (crow's foot) или подпишите кратностью (например, «1» и «M» над линией).

Определите следующие связи:

Издательство — Книга (1:M)

Книга — Автор (M:M) — **потребуется создать дополнительную связующую таблицу Книга_Автор**

Книга — Выдача (1:M)

Читатель — Выдача (1:M)

Добавьте связующую таблицу Книга_Автор (для связи M:M между Книгой и Автором). Она должна содержать:

id_книги (FK)

id_автора (FK)

Составной первичный ключ: (id_книги, id_автора)

Сохраните результат:

В draw.io: нажмите «Файл» → «Сохранить как» и выберите удобный формат (например, .drawio или экспортируйте в PNG/PDF).

В Lucidchart: нажмите «Файл» → «Скачать как» и выберите формат (PNG, PDF).

Форма сдачи

Предоставьте преподавателю скриншот или экспортированное изображение вашей ER-диаграммы.

5. Практическое занятие: «Применить фильтрацию данных с использованием HAVING для работы с агрегированными результатами».

Исходная ситуация

Вам дана таблица Sales (Продажи) со следующей структурой:

Поле	Тип	Описание
sale_id	INTEGER	Уникальный идентификатор продажи (PK)
product	VARCHAR(100)	Название товара
category	VARCHAR(50)	Категория товара
seller	VARCHAR(100)	Продавец (ФИО)
quantity	INTEGER	Количество проданных единиц
amount	DECIMAL(10,2)	Сумма продажи
sale_date	DATE	Дата продажи

Пример данных:

sale_id	product	category	seller	quantity	amount	sale_date
1	Ноутбук	Электроника	Иванов И.	2	120000	2026-01-10
2	Телефон	Электроника	Петров П.	3	90000	2026-01-12
3	Книга	Литература	Иванов И.	5	2500	2026-01-15
4	Телефон	Электроника	Петров П.	1	30000	2026-02-05
5	Ноутбук	Электроника	Сидоров С.	1	60000	2026-02-10
6	Книга	Литература	Петров П.	3	1500	2026-02-12
7	Планшет	Электроника	Иванов И.	2	40000	2026-03-01
8	Книга	Литература	Сидоров С.	10	5000	2026-03-05

Задание

Часть 1. Написание запросов с HAVING (10 минут)

На основе таблицы Sales напишите SQL-запросы для выполнения следующих задач:

Найти категории товаров, общая сумма продаж которых превышает 100 000 рублей.

Выведите: category, total_amount. Отсортируйте по убыванию общей суммы.

Найти продавцов, у которых общее количество проданных единиц товара больше 5.

Выведите: seller, total_quantity.

Найти категории, в которых было совершено более 3 продаж (количество записей).

Выведите: category, sales_count.

Часть 2. Сравнение WHERE и HAVING (10 минут)

Ответьте на вопросы (письменно, кратко):

Объясните разницу между WHERE и HAVING. Приведите пример ситуации, где использование WHERE не подходит, а HAVING — необходимо.

Можно ли использовать WHERE вместо HAVING для фильтрации агрегированных данных? Почему?

Напишите запрос, который сначала фильтрует строки по дате (только продажи после 2026-01-15), а затем группирует по категориям и показывает только те категории, общая сумма которых превышает 50 000 рублей.

(Подсказка: нужны и `WHERE`, и `HAVING`).

Форма выполнения

Результаты оформите в виде текстового файла или на листе, содержащем:

Три SQL-запроса к таблице `Sales` (Часть 1).

Ответы на три вопроса (Часть 2).

6. Практическое занятие: «Выполнить запрос с использованием `UNION ALL` для объединения данных с учетом дубликатов».

Исходная ситуация

В компании ведётся учёт сотрудников. Данные хранятся в двух таблицах:

Таблица `Employees_Current` — действующие сотрудники:

employee_id	full_name	department	hire_date
1	Иванов И.	IT	2020-03-15
2	Петров П.	Sales	2021-07-01
3	Сидорова С.	HR	2022-01-10
4	Козлов К.	IT	2023-05-20

Таблица `Employees_Archived` — уволенные сотрудники (архив):

employee_id	full_name	department	hire_date
2	Петров П.	Sales	2021-07-01
5	Михайлов М.	Sales	2019-11-01
6	Фёдорова Ф.	IT	2020-09-15
7	Семёнов С.	HR	2018-04-10

Обратите внимание: таблицы имеют одинаковую структуру. Сотрудник Петров П. присутствует в обеих таблицах (сначала работал, потом уволился — дубликат).

Задание

Часть 1. Написание запросов с `UNION ALL` (10 минут)

На основе приведённых таблиц выполните следующие задания:

Получить полный список всех сотрудников (действующих и уволенных) с указанием статуса.

Добавьте в результат вычисляемое поле `status` со значениями 'Действующий' для таблицы `Employees_Current` и 'Уволен' для таблицы `Employees_Archived`.

Выведите: `full_name`, `department`, `status`.

Результат отсортируйте по фамилии.

Посчитать общее количество сотрудников (включая дубликаты).

Напишите запрос, который возвращает одно число — общее количество записей в обеих таблицах (с учётом дубликатов).

(Подсказка: используйте `UNION ALL` в подзапросе или в `COUNT`).

Часть 2. Сравнение UNION и UNION ALL (10 минут)

Ответьте на вопросы (письменно, кратко):

В чём разница между `UNION` и `UNION ALL`?

Почему в Задании 1 (полный список сотрудников) лучше использовать `UNION ALL`, а не `UNION`? Что бы изменилось, если бы использовали `UNION`? Дайте ответ применительно к данным из таблиц.

Напишите запрос, который возвращает список уникальных (без дубликатов) отделов (`department`), в которых работают сотрудники из обеих таблиц.

(Подсказка: здесь нужен `UNION`, а не `UNION ALL`).

Форма выполнения

Результаты оформите в виде текстового файла или на листе, содержащем:

Два запроса с `UNION ALL` (Часть 1).

Ответы на три вопроса и запрос с `UNION` (Часть 2).

Раздел 2. Логическое и физическое проектирование баз данных

Тема 2.1. Логическое проектирование баз данных

1. Практическое занятие: «Преобразование ER-диаграммы в таблицы реляционной базы данных».

Исходная ситуация

Вам представлена **ER-диаграмма** для системы управления заказами в интернет-магазине (в текстовом описании).

Сущности и атрибуты:

Клиент

Атрибуты: `id_клиента` (PK), ФИО, Телефон, Email

Заказ

Атрибуты: `id_заказа` (PK), Дата_заказа, Сумма

Товар

Атрибуты: `id_товара` (PK), Название, Цена

Состав_заказа (ассоциативная сущность)

Атрибуты: `id_заказа` (FK), `id_товара` (FK), Количество

Связи между сущностями:

Клиент — Заказ: один клиент может сделать много заказов (1:M). Связь реализуется внешним ключом в таблице `Заказ`.

Заказ — Товар: один заказ может содержать много товаров, один товар может входить во многие заказы (M:M). Связь реализуется через ассоциативную таблицу `Состав_заказа`.

Задание

Часть 1. Определение таблиц и их атрибутов (10 минут)

На основе ER-диаграммы перечислите **все таблицы** реляционной базы данных. Для каждой таблицы укажите:

Название таблицы.

Список столбцов (с типами данных) — используйте `INTEGER`, `VARCHAR`, `DECIMAL`, `DATE`).

Первичный ключ (PK).

Форма выполнения – заполните таблицу:

Таблица	Столбцы (с типами)	PK
Клиент	<code>id_клиента</code> <code>INTEGER</code> , ФИО <code>VARCHAR(255)</code> , Телефон <code>VARCHAR(20)</code> , Email <code>VARCHAR(255)</code>	<code>id_клиента</code>
...	...	...

Часть 2. Определение внешних ключей и связей (10 минут)

Для каждой таблицы, содержащей внешние ключи, укажите:

Внешний ключ (FK) — столбец и таблица, на которую он ссылается.

Тип связи (1:1, 1:M, M:M) между связываемыми таблицами.

Форма выполнения – заполните таблицу:

Таблица	FK (столбец → родительская таблица)	Тип связи
Заказ	<code>id_клиента</code> → Клиент(<code>id_клиента</code>)	1:M

Таблица	FK (столбец → родительская таблица)	Тип связи
Состав_заказа	id_заказа → Заказ(id_заказа)	M:M (через состав)
Состав_заказа	id_товара → Товар(id_товара)	M:M (через состав)

Форма выполнения

Результаты оформите в виде **текстового файла** или на **листе**, содержащем:

Заполненную таблицу с таблицами, столбцами и первичными ключами (Часть 1).

Заполненную таблицу с внешними ключами и типами связей (Часть 2).

2. Практическое занятие: «Разработка концепции базы данных и создание блок-схемы, отражающую структуру и связи между таблицами».

Исходная ситуация

Предметная область: Управление строительными проектами

Строительная компания «СтройИнвест» занимается возведением жилых и коммерческих объектов. Компания планирует разработать информационную систему для управления проектами. Вам поручено разработать **концептуальную модель базы данных**.

Требования к системе:

Хранить информацию о **проектах** (название, адрес, дата начала, дата окончания, бюджет).

В каждом проекте участвуют **сотрудники** (ФИО, должность, телефон, email).

Один сотрудник может быть назначен **руководителем проекта** (один сотрудник — руководитель одного или нескольких проектов).

В рамках проектов выполняются **работы** (название, описание, плановая стоимость, статус).

Каждая работа связана с **одним проектом**.

Сотрудники могут быть назначены на **несколько работ** в рамках разных проектов (связь M:M).

Для каждой работы может быть назначен один **ответственный сотрудник**.

Задание

Часть 1. Выделение сущностей и атрибутов (10 минут)

На основе описания предметной области выделите **все сущности** (не

менее 4). Для каждой сущности укажите:

Название сущности.

Атрибуты (не менее 3 на сущность).

Первичный ключ (РК).

Форма выполнения — заполните таблицу:

Сущность	Атрибуты	Первичный ключ (РК)
Проект	Название, Адрес, Дата_начала, Дата_окончания, Бюджет	id_проекта
...	...	...

Часть 2. Определение связей и построение блок-схемы (10 минут)

Определите связи между выделенными сущностями. Для каждой пары сущностей укажите:

Тип связи (1:1, 1:M, M:M).

В какой таблице будет располагаться внешний ключ (FK).

Форма выполнения — заполните таблицу:

Связь	Тип связи	Где располагается FK	Пояснение
Проект — Сотрудник (руководитель)	1:M	Проект.id_руководителя	Один сотрудник может руководить несколькими проектами
...	...	...	...

Нарисуйте блок-схему (ER-диаграмму) (можно от руки), отражающую:

Все таблицы (прямоугольники).

Первичные ключи (подчёркнуты).

Внешние ключи.

Линии связей между таблицами с указанием типа (1, M).

Форма выполнения

Результаты оформите на **одном листе** (или в текстовом файле), содержащем:

Заполненную таблицу сущностей и атрибутов (Часть 1).

Заполненную таблицу связей и блок-схему (Часть 2).

3. Практическое занятие: «Создание структуры базы данных в pgAdmin».

Исходная ситуация

Вы — разработчик базы данных для небольшого интернет-магазина. Вам необходимо создать структуру БД, состоящую из двух связанных таблиц: `categories` — список категорий товаров.

`products` — список товаров. Каждый товар относится к одной категории.

Структура таблиц:

Таблица `categories`

Поле	Тип	Ограничения
<code>id</code>	SERIAL	PRIMARY KEY
<code>name</code>	VARCHAR(100)	NOT NULL

Таблица `products`

Поле	Тип	Ограничения
<code>id</code>	SERIAL	PRIMARY KEY
<code>name</code>	VARCHAR(200)	NOT NULL
<code>price</code>	DECIMAL(10,2)	NOT NULL
<code>category_id</code>	INTEGER	FOREIGN KEY → categories(id)

Задание

Часть 1. Визуальное создание таблиц через ERD-инструмент (10 минут)

Выполните следующие шаги в pgAdmin:

Откройте ERD-инструмент:

В дереве объектов слева выберите вашу базу данных.

В верхнем меню выберите **Tools** → **ERD Tool**. Откроется пустая канва для создания диаграммы.

Создайте таблицу `categories`:

На панели инструментов ERD нажмите кнопку **Add table** (с иконкой «+»).

В открывшемся диалоге:

В поле **Name** введите `categories`.

На вкладке **Columns** добавьте столбцы:

id — тип **SERIAL**, отметьте галочку **Primary Key**.

name — тип **VARCHAR(100)**, отметьте **Not NULL**.

Нажмите **Save**.

Создайте таблицу products:

Повторите шаг 2 для таблицы **products** со столбцами:

id — **SERIAL**, **Primary Key**.

name — **VARCHAR(200)**, **Not NULL**.

price — **DECIMAL(10,2)**, **Not NULL**.

category_id — **INTEGER** (пока без внешнего ключа).

Установите связь между таблицами:

На канве выберите таблицу **categories** (она будет родительской).

На панели инструментов нажмите кнопку **1M** (для создания связи «один-ко-многим»).

В открывшемся диалоге укажите:

Referencing table — **products** (таблица, которая ссылается).

Local column — **category_id** (столбец в **products**).

References — **categories** (родительская таблица).

Referencing — **id** (столбец в **categories**).

Нажмите **Save**. На канве появится линия связи между таблицами.

Сгенерируйте SQL-скрипт:

На панели инструментов нажмите кнопку **Generate SQL** (с иконкой «SQL»).

Скопируйте сгенерированный SQL-код в текстовый файл (он понадобится для Части 2).

Часть 2. Создание таблиц через SQL-запросы (10 минут)

Теперь создайте те же таблицы с помощью SQL-скриптов в Query Tool.

Откройте Query Tool:

В дереве объектов выберите вашу базу данных.

Нажмите правой кнопкой мыши и выберите **Query Tool**.

Напишите и выполните SQL-скрипт для создания обеих таблиц (включая внешний ключ):

```
sql
-- Создание таблицы категорий
CREATE TABLE categories (
  id SERIAL PRIMARY KEY,
  name VARCHAR(100) NOT NULL
);
```

```
-- Создание таблицы товаров
CREATE TABLE products (
  id SERIAL PRIMARY KEY,
  name VARCHAR(200) NOT NULL,
  price DECIMAL(10,2) NOT NULL,
  category_id INTEGER REFERENCES categories(id)
);
```

Проверьте результат:

В дереве объектов обновите список таблиц (правой кнопкой → **Refresh**).

Убедитесь, что появились таблицы `categories` и `products`.

Разверните таблицу `products` и проверьте наличие внешнего ключа `category_id` с ссылкой на `categories(id)`.

Форма выполнения

Результаты оформите в виде **текстового файла** или на листе, содержащем:

Скриншот ER-диаграммы с двумя таблицами и связью между ними (Часть 1).

SQL-скрипт, сгенерированный ERD-инструментом, и SQL-скрипт, написанный вручную (Часть 2).

4. Практическое занятие: «Приведение таблицы ко всем нормальным формам».

Исходная ситуация

В компании ведётся учёт заказов. Используется одна таблица `Orders` со следующей структурой и данными:

order_id	customer_name	customer_phone	product_name	product_category	product_price	quantity	order_date
101	Иванов И.	89001234567	Ноутбук	Электроника	60000	1	2026-01-10
101	Иванов И.	89001234567	Мышь	Электроника	1500	2	2026-01-10
102	Петрова А.	89007654321	Книга	Литература	500	3	2026-01-12
103	Иванов	8900123	Книга	Литерат	500	1	2026-

order_id	customer_name	customer_phone	product_name	product_category	product_price	quantity	order_date
	И.	4567		ур			01-15

Проблемы текущей таблицы:

Повторяются данные о клиентах (имя, телефон) для каждого заказа.

Повторяются данные о товарах (название, категория, цена) для каждого заказа.

Существует частичная зависимость: `product_category` зависит только от `product_name`, а не от всего первичного ключа.

Существует транзитивная зависимость: `product_price` зависит от `product_name`, а не от первичного ключа.

Задание

Часть 1. Проверка нормальных форм (10 минут)

Определите, находится ли таблица в 1НФ. Если нет, предложите, как её привести к 1НФ.

(Подсказка: проверьте атомарность значений и уникальность строк).

Определите первичный ключ таблицы (он может быть составным). Объясните свой выбор.

Определите, находится ли таблица в 2НФ. Если нет, укажите, какие частичные зависимости существуют, и объясните, почему они нарушают 2НФ.

Определите, находится ли таблица в 3НФ. Если нет, укажите, какие транзитивные зависимости существуют.

Часть 2. Приведение к 3НФ (10 минут)

Разделите исходную таблицу `Orders` на **несколько нормализованных таблиц** (минимум 3), чтобы каждая из них находилась в 3НФ. Для каждой новой таблицы:

Укажите название.

Перечислите столбцы.

Обозначьте **первичный ключ** и **внешние ключи** (если есть).

Поясните, как ваше разделение устраняет:

Повторение данных о клиентах.

Повторение данных о товарах.

Частичные и транзитивные зависимости.

Форма выполнения

Результаты оформите в виде **текстового файла** или на **листе**,

содержащем:

Ответы на вопросы Части 1.

Список новых таблиц с ключами и пояснением (Часть 2).

5. Практическое занятие: «Использование триггеров для реализации проверки данных перед их внесением в таблицу».

Исходная ситуация

В компании «СтройПроект» ведётся учёт сотрудников в таблице `Employees`. Руководство хочет, чтобы база данных автоматически проверяла корректность вводимых данных на уровне СУБД, не полагаясь только на приложение.

Таблица `Employees` имеет следующую структуру:

Поле	Тип	Ограничения
<code>id</code>	<code>SERIAL</code>	<code>PRIMARY KEY</code>
<code>full_name</code>	<code>VARCHAR(255)</code>	<code>NOT NULL</code>
<code>email</code>	<code>VARCHAR(255)</code>	<code>NOT NULL</code>
<code>salary</code>	<code>DECIMAL(10,2)</code>	<code>NOT NULL</code>
<code>hire_date</code>	<code>DATE</code>	<code>NOT NULL</code>
<code>department</code>	<code>VARCHAR(100)</code>	

Дополнительные бизнес-правила, которые должны проверяться при вставке или обновлении данных:

Проверка salary: Зарплата должна быть больше 0. Минимальная зарплата в компании — 10 000 рублей.

Проверка email: Email должен содержать символ @ (базовая проверка).

Проверка hire_date: Дата приёма на работу не должна быть в будущем (не позже текущей даты).

Задание

Часть 1. Создание триггерной функции и триггера (10 минут)

Напишите SQL-скрипт, который:

Создаёт триггерную функцию `validate_employee()`, которая проверяет все три бизнес-правила:

Если salary < 10000 — выбрасывать исключение с сообщением «Зарплата не может быть меньше 10000».

Если email не содержит символ @ — выбрасывать исключение «Некорректный email».

Если hire_date > CURRENT_DATE — выбрасывать исключение «Дата приёма не может быть в будущем».

Создаёт триггер employee_validation, который выполняет эту функцию перед вставкой или обновлением каждой строки в таблице Employees.

Подсказка: используйте NEW для доступа к вставляемым/обновляемым значениям.

Часть 2. Проверка работы триггера (10 минут)

Напишите два INSERT-запроса:

Первый (должен пройти успешно): вставьте сотрудника с корректными данными (зарплата > 10000, email содержит @, дата приёма — сегодня или в прошлом).

Второй (должен вызвать ошибку триггера): вставьте сотрудника с зарплатой 5000 (или с некорректным email, или с датой в будущем). Запишите текст ошибки, который вернула СУБД.

Напишите UPDATE-запрос, который пытается изменить зарплату существующего сотрудника на отрицательное значение (например, -1000). Запишите текст ошибки, которая возникнет.

Кратко ответьте на вопросы:

Почему проверка зарплаты через CHECK-ограничение (CHECK (salary > 0)) не подходит для требования «зарплата не меньше 10000»? (В чём отличие?)

Какие ещё проверки данных можно реализовать с помощью триггеров, которые нельзя сделать через CHECK?

Форма выполнения

Результаты оформите в виде текстового файла или на листе, содержащем:

SQL-код триггерной функции и триггера (Часть 1).

SQL-запросы для проверки и текст ошибок (Часть 2).

Ответы на вопросы (Часть 2).

Критерии оценки выполнения практических заданий

Оценка «отлично» выставляется студенту, если практическое задание выполнено в полном объеме с соблюдением необходимой последовательности действий; итогом решения задания явилось получение правильных результатов и выводов; в ходе решения задания верно и аккуратно выполнены все вычисления и записи.

Оценка «хорошо» выставляется студенту, если практическое задание выполнено правильно с учетом мелких погрешностей или недочетов.

Оценка «удовлетворительно» выставляется студенту, если практическое задание выполнено правильно не менее чем наполовину, допущены некоторые погрешности или одна грубая ошибка.

Оценки «неудовлетворительно» заслуживает студент, если при решении практического задания допущены две (и более) грубые ошибки, или задание не решено полностью.

Вопросы компьютерного тестирования (ОК.01, ОК.02, ОК.03, ОК.04, ОК.05)

Раздел 1. Основы проектирования баз данных

Вопрос 1. База данных - это:

1. совокупность данных, организованных по определенным правилам;
2. совокупность программ для хранения и обработки больших массивов информации;
3. интерфейс, поддерживающий наполнение и манипулирование данными;
4. определенная совокупность информации

.

Вопрос 2. Наиболее распространенными в практике являются:

1. распределенные базы данных;
2. иерархические базы данных;
3. сетевые базы данных;
4. реляционные базы данных

.

Вопрос 3. Наиболее точным аналогом реляционной базы данных может служить:

1. неупорядоченное множество данных;
2. вектор;
3. генеалогическое дерево;
4. двумерная таблица.

Вопрос 4. Таблицы в базах данных предназначены:

1. для хранения данных базы;

2. для отбора и обработки данных базы;
3. для ввода данных базы и их просмотра;
4. для автоматического выполнения группы команд;
5. для выполнения сложных программных действий

.

Вопрос 5. Что из перечисленного не является объектом Access:

1. модули;
2. таблицы;
3. макросы;
4. ключи;
5. формы;
6. отчеты;
7. запросы?

Вопрос 6. Для чего предназначены запросы:

1. для хранения данных базы;
2. для отбора и обработки данных базы;
3. для ввода данных базы и их просмотра;
4. для автоматического выполнения группы команд;
5. для выполнения сложных программных действий;
6. для вывода обработанных данных базы на принтер?

Вопрос 7. Для чего предназначены формы:

1. для хранения данных базы;
2. для отбора и обработки данных базы;
3. для ввода данных базы и их просмотра;
4. для автоматического выполнения группы команд;
5. для выполнения сложных программных действий?

Вопрос 8. Для чего предназначены модули:

1. для хранения данных базы;
2. для отбора и обработки данных базы;
3. для ввода данных базы и их просмотра;
4. для автоматического выполнения группы команд;
5. для выполнения сложных программных действий?

Вопрос 9. Для чего предназначены макросы:

1. для хранения данных базы;
2. для отбора и обработки данных базы;
3. для ввода данных базы и их просмотра;
4. для автоматического выполнения группы команд;
5. для выполнения сложных программных действий?

Вопрос 10. В каком режиме работает с базой данных пользователь:

251. в проектировочном;
2. в любительском;
3. в заданном;
4. в эксплуатационном?

Вопрос 11. В каком диалоговом окне создают связи между полями таблиц базы данных:

1. таблица связей;
2. схема связей;
3. схема данных;
4. таблица данных?

Вопрос 12. Почему при закрытии таблицы программа Access не предлагает выполнить сохранение внесенных данных:

1. недоработка программы;
2. потому что данные сохраняются сразу после ввода в таблицу;
3. потому что данные сохраняются только после закрытия всей базы данных?

Вопрос 13. Без каких объектов не может существовать база данных:

1. без модулей;
2. без отчетов;
3. без таблиц;
4. без форм;
5. без макросов;
6. без запросов?

Вопрос 14. В каких элементах таблицы хранятся данные базы:

1. в полях;
2. в строках;
3. в столбцах;
4. в записях;
5. в ячейках?

Вопрос 15. Содержит ли какую-либо информацию таблица, в которой нет ни одной записи?

1. пустая таблица не содержит никакой информации;
2. пустая таблица содержит информацию о структуре базы данных;
3. пустая таблица содержит информацию о будущих записях;
4. таблица без записей существовать не может.

Вопрос 16. Содержит ли какую-либо информацию таблица, в которой нет полей?

1. содержит информацию о структуре базы данных;
2. не содержит никакой информации;
3. таблица без полей существовать не может;
4. содержит информацию о будущих записях.

Вопрос 17. В чем состоит особенность поля "счетчик"?

1. служит для ввода числовых данных;
2. служит для ввода действительных чисел;
3. данные хранятся не в поле, а в другом месте, а в поле хранится только указатель на то, где расположен текст;
4. имеет ограниченный размер;
5. имеет свойство автоматического наращивания.

Вопрос 18. В чем состоит особенность поля "мемо"?

1. служит для ввода числовых данных;
2. служит для ввода действительных чисел;
3. данные хранятся не в поле, а в другом месте, а в поле хранится только указатель на то, где расположен текст;
4. имеет ограниченный размер;
5. имеет свойство автоматического наращивания.

Вопрос 19. Какое поле можно считать уникальным?

1. поле, значения в котором не могут повторяться;
2. поле, которое носит уникальное имя;
3. поле, значение которого имеют свойство наращивания.

Вопрос 20. Ключами поиска в системах управления базами данных (СУБД) называются:

1. диапазон записей файла БД, в котором осуществляется поиск;
2. логические выражения, определяющие условия поиска;
3. поля, по значению которых осуществляется поиск;
4. номера записей, удовлетворяющих условиям поиска;
5. номер первой по порядку записи, удовлетворяющей условиям поиска?

21. В каком порядке должны выполняться уровни проектирования БД?

1. физический, логический, концептуальный
2. концептуальный, физический, логический
3. концептуальный, логический, физический
4. внешний, физический, концептуальный

Раздел 2. Логическое и физическое проектирование баз данных

Вопрос 1. ER-диаграмма – это:

1. результат логического уровня проектирования
2. обязательный этап проектирования БД

3. средство установления связей между таблицами
4. графическая модель предметной области

Вопрос 2. Выбрать правильное высказывание из приведенных ниже:

1. тип сущности ГОРОД включает экземпляр сущности МОСКВА
1. сущности ГОРОД и МОСКВА являются типами сущности
2. сущности ГОРОД и МОСКВА являются экземплярами сущности
3. тип сущности МОСКВА включает экземпляр сущности ГОРОД

Вопрос 3. Определите тип связи между объектами «Преподаватель» и «Дисциплина»,

если один преподаватель может вести занятия по нескольким дисциплинам.

1. «многие – к – одному»
2. «один – к – одному»
3. «многие – ко – многим»
4. «один – ко – многим»

Вопрос 4. Какой элемент не используется в модели «сущность – связь»?

1. узел
1. сущность
2. связь
3. атрибут

Вопрос 5. Какие виды связей из перечисленных непосредственно поддерживаются в реляционной модели данных?

1. «один-к-одному», «один-ко-многим», «многие-к-одному»
1. «один-к-одному», «один-ко-многим», «многие-к-одному», «многие-ко-многим»
2. «один-к-одному», «один-ко-многим»
- 3.

Вопрос 6. Определите тип связи между объектами «Преподаватель» и «Дисциплина»,

если один преподаватель может вести занятия по нескольким дисциплинам, и

занятия по одной дисциплине могут вести несколько преподавателей.

1. «многие – к – одному»
2. «многие – ко – многим»
3. «один – к – одному»
4. «один – ко – многим»

Вопрос 7. Какие виды связей из перечисленных имеют место в реляционной модели данных?

1. «один-к-одному», «один-ко-многим», «многие-к-одному», «многие-ко-многим»

1. «один-к-одному», «один-ко-многим»

2. «один-к-одному», «один-ко-многим», «многие-к-одному»

Вопрос 8. Определите тип отношения между таблицами «Город» и «Район», если

каждому городу соответствует несколько районов.

1. «многие – к – одному»

2. «один – ко – многим»

3. «многие – ко – многим»

4. «один – к – одному»

Вопрос 9. Определите тип отношения между таблицами «Преподаватели» и «Студенты», если один преподаватель обучает разных студентов.

1. «один – ко – многим»

1. «один – к – одному»

2. «многие – к – одному»

3. «многие – ко – многим»

Вопрос 10. Определите тип отношения между таблицами «Преподаватели» и «Студенты», если один преподаватель обучает разных студентов.

1. «один – ко – многим»

1. «один – к – одному»

2. «многие – к – одному»

3. «многие – ко – многим»

Вопрос 11. Определите тип отношения между таблицами «Поставщики» и «Товары»,

если каждый поставщик поставяет несколько товаров.

1. «один – ко – многим»

291. «многие – ко – многим»

2. «один – к – одному»

3. «многие – к – одному»

Вопрос 12. Определите тип отношения между таблицами «Преподаватели» и «Студенты», если одного студента обучают разные преподаватели.

1. «один – к – одному»

2. «многие – к – одному»

3. «один – ко – многим»

4. «многие – ко – многим»

Вопрос 13. Если проектирование начинается с анализа задач и функций, обеспечивающих реализацию информационных потребностей пользователей, то имеем

дело с:

1. объектным подходом проектирования предметной области
2. функциональным подходом проектирования предметной области
3. предметным подходом проектирования предметной области

Вопрос 14. Все возможные атрибуты сущности, уникальным образом ее идентифицирующие, называются:

1. альтернативными ключами
2. потенциальными ключами
3. индексами

Вопрос 15. Какая целостность данных обеспечивает корректное и полноценное перемещение среди сущностей, связанных между собой?

1. целостность по сущностям
2. ссылочная целостность
3. целостность первичных ключей
4. целостность доменов

Вопрос 16. Какая целостность данных обеспечивается использованием первичного ключа?

1. первичная целостность
2. ссылочная целостность
3. целостность по сущностям
4. целостность доменов

Вопрос 17. Какая целостность данных реализуется внешним ключом?

1. реляционная целостность
2. целостность по сущностям
3. ссылочная целостность
4. целостность доменов

Вопрос 18. Какой тип данных присваивается искусственному первичному ключу?

1. счетчик
1. числовой
2. поле объекта OLE
3. текстовый

Вопрос 19. Набор правил, используемых для поддержания отношений между записями в связанных таблицах, называется:

1. условиями целостности данных

1. условиями добавления данных
2. условиями удаления данных
3. условиями сохранения данных

Раздел 3. Основы работы с запросами и оптимизация баз данных

Вопрос1. Моделью, какого уровня проектирования баз данных является инфологическая модель?

1. концептуального
2. физического
3. компьютерного
4. логического

Вопрос2. Модель проектирования БД, которая представляет собой отображение логических связей между элементами данных безотносительно к их содержанию и среде хранения, называется:

1. даталогической моделью
1. внешней моделью
2. инфологической моделью
3. физической моделью

Вопрос3. Модель проектирования БД, которая представляет собой описание предметной области, выполненное без ориентации на используемые в дальнейшем программные и технические средства, называется:

1. физической моделью
2. инфологической моделью
3. внешней моделью
4. даталогической моделью

Вопрос4. Процесс создания приложения баз данных начинается:

1. с разработки структуры данных
2. с разработки информационно-логической модели предметной области
3. с разработки структуры реляционных таблиц

Вопрос5. Какая стадия является наиболее значительной в жизненном цикле приложения?

1. реализации
2. тестирования
3. проектирования
4. эксплуатации

Вопрос6. Область применения БД определяется на этапе:

1. проектирования БД
2. сбора и анализа требований пользователей
3. планирования разработки БД
4. определения требований к системе

Вопрос7. Общая стоимость проекта определяется на этапе:

1. планирования разработки БД
1. проектирования БД
2. сбора и анализа требований пользователей
3. определения требований к системе

Вопрос8. Пользовательский интерфейс разрабатывается на этапе:

1. тестирования БД
2. реализации БД
3. разработки приложений
4. проектирования БД

Вопрос9. Спецификации требований пользователей составляются на этапе:

1. планирования разработки БД
2. определения требований к системе
3. сбора и анализа требований пользователей
4. проектирования БД

Критерии оценки знаний студентов при проведении тестирования

Оценка «отлично» выставляется при условии правильного ответа студента не менее чем на 86 % тестовых заданий;

Оценка «хорошо» выставляется при условии правильного ответа студента на 70-85 % тестовых заданий;

Оценка «удовлетворительно» выставляется при условии правильного ответа студента на 50-69 % тестовых заданий;

Оценки «неудовлетворительно» заслуживает студент при условии правильного ответа студента менее чем на 50 % тестовых заданий.

1.4. Задания для самостоятельной работы (ОК.01, ОК.02, ОК.03, ОК.04, ОК.05)

Раздел 1. Основы проектирования баз данных

Тема 1.1. Введение в базы данных

«Изучение основных приемов нормализации баз данных»

Исходная ситуация

В компании ведётся учёт заказов в одной таблице **Orders**:

order_id	order_date	customer_name	customer_phone	product_name	product_category	product_price	quantity
101	2026-01-10	Иванов Иван	89001234567	Ноутбук	Электроника	75000	1
101	2026-01-10	Иванов Иван	89001234567	Мышь	Электроника	2500	2
102	2026-01-12	Петрова Анна	89007654321	Книга	Литература	500	3
103	2026-01-15	Иванов Иван	89001234567	Книга	Литература	500	1
103	2026-01-15	Иванов Иван	89001234567	Смартфон	Электроника	45000	1

Проблемы текущей таблицы:

Данные о клиентах повторяются для каждого заказа (**customer_name**, **customer_phone**).

Данные о товарах повторяются (**product_name**, **product_category**, **product_price**).

Существует частичная зависимость: **product_category** зависит только от **product_name**, а не от всего первичного ключа.

Существует транзитивная зависимость: **product_price** зависит от **product_name**.

Задание

Часть 1. Определение нарушений нормальных форм

На основе данных таблицы **ответьте на вопросы**:

Находится ли таблица в 1НФ? Если нет, укажите, какое правило нарушено.

Каков первичный ключ таблицы? (Он может быть составным). Поясните, почему вы выбрали именно такой ключ.

Находится ли таблица в 2НФ? Если нет, укажите, какие **частичные зависимости** существуют в таблице (какие атрибуты зависят не от всего первичного ключа, а только от его части).

Находится ли таблица в 3НФ? Если нет, укажите, какие транзитивные зависимости существуют (какой неключевой атрибут зависит от другого неключевого атрибута).

Часть 2. Приведение к 3НФ

На основе выявленных нарушений разделите исходную таблицу на несколько таблиц, каждая из которых будет находиться в 3НФ. Заполните таблицу:

Новая таблица	Столбцы	Первичный ключ (РК)	Внешние ключи (FK)
...	...	...	...

Поясните, какие аномалии устраняются в результате нормализации.

Форма выполнения

Результаты оформите в виде текстового файла или на листе, содержащем:

Ответы на четыре вопроса (Часть 1).

Заполненную таблицу с новыми таблицами и ключами (Часть 2).

Тема 1.2. Концептуальное проектирование баз данных

«Приобретение опыта проектирования реляционных баз данных по выбранной предметной области»

Исходная ситуация

Вы — системный аналитик в компании, которая планирует разработать информационную систему для **выбранной вами предметной области**. Вам необходимо спроектировать структуру базы данных для этой системы.

Выберите одну из предложенных предметных областей (или предложите свою, согласовав с преподавателем):

№	Предметная область	Краткое описание
1	Библиотека	Учёт книг, читателей, выдачи книг, авторов, издательств
2	Интернет-магазин	Учёт товаров, заказов, клиентов, поставщиков, категорий
3	Спортивный клуб	Учёт клиентов, абонементов, тренировок, посещений, тренеров
4	Система управления проектами	Учёт проектов, сотрудников, задач, назначений, сроков
5	Гостиница	Учёт номеров, бронирований, клиентов, сотрудников, услуг
6	Учебное заведение	Учёт студентов, групп, преподавателей, предметов,

№	Предметная область	Краткое описание
		оценок

Задание

Выполните проектирование базы данных для выбранной предметной области, следуя этапам ниже:

1. Выделение сущностей и атрибутов (5 минут)

На основе выбранной предметной области выделите не менее 4 сущностей. Для каждой сущности укажите:

Название сущности.

Атрибуты (не менее 3 на сущность).

Первичный ключ (РК).

Форма выполнения — заполните таблицу:

Сущность	Атрибуты	Первичный ключ (РК)
1. ...	...	...
2. ...	...	...
3. ...	...	...
4. ...	...	...

2. Определение связей между сущностями (5 минут)

Установите связи между выделенными сущностями. Для каждой пары сущностей укажите:

Тип связи (1:1, 1:M, M:M).

В какой таблице будет располагаться внешний ключ (FK).

Форма выполнения — заполните таблицу:

Связь	Тип связи	Где располагается FK	Пояснение
...	...	...	...

3. Построение ER-диаграммы (5 минут)

На основе определённых сущностей и связей нарисуйте ER-диаграмму (можно схематично, от руки). На схеме отразите:

Прямоугольники для каждой таблицы с названием сущности.

Внутри прямоугольника перечислите ключевые поля (РК, FK).

Линии, соединяющие таблицы, с указанием типа связи (1, M).

4. Преобразование в реляционную схему (5 минут)

На основе ER-диаграммы составьте реляционную схему базы данных. Для каждой таблицы укажите:

Название таблицы.
 Список столбцов (с типами данных
 – INTEGER, VARCHAR, DECIMAL, DATE).
 Первичный ключ (PK).
 Внешние ключи (FK) с указанием ссылочной таблицы.
Форма выполнения — заполните таблицу:

Таблица	Столбцы (с типами)	PK	FK (ссылка)
...	...	...	...

Форма выполнения
 Результаты оформите на одном листе (или в текстовом файле),
 содержащем:
 Заполненную таблицу сущностей и атрибутов (шаг 1).
 Заполненную таблицу связей (шаг 2).
 ER-диаграмму (шаг 3) — рисунок или текстовое описание.
 Заполненную таблицу реляционной схемы (шаг 4).

Раздел 2. Логическое и физическое проектирование баз данных

Тема 2.1. Логическое проектирование баз данных

**«Приведение структуры БД к третьей нормальной форме.
 Реализация связи многие- ко многим через промежуточные таблицы»**

Исходная ситуация

В компании ведётся учёт заказов в таблице Orders:

order_id	order_date	customer_name	customer_phone	product_name	product_category	product_price	quantity
101	2026-01-10	Иванов Иван	89001234567	Ноутбук	Электроника	75000	1
101	2026-01-10	Иванов Иван	89001234567	Мышь	Электроника	2500	2
102	2026-01-12	Петрова Анна	89007654321	Книга	Литература	500	3
103	2026-01-15	Иванов Иван	89001234567	Книга	Литература	500	1
103	2026-01-15	Иванов Иван	89001234567	Смартфон	Электроника	45000	1

Проблемы текущей таблицы:

Данные о клиентах повторяются для каждого заказа (customer_name, customer_phone).

Данные о товарах повторяются

(product_name, product_category, product_price).

Существует частичная зависимость: product_category зависит только от product_name, а не от всего первичного ключа.

Существует транзитивная зависимость: product_price зависит от product_name.

Задание

Часть 1. Анализ нарушений нормальных форм (10 минут)

На основе данных таблицы ответьте на вопросы:

Находится ли таблица в 1НФ? Если нет, укажите, какое правило нарушено.

Каков первичный ключ таблицы? (Он может быть составным). Поясните, почему вы выбрали именно такой ключ.

Находится ли таблица в 2НФ? Если нет, укажите, какие **частичные зависимости** существуют в таблице (какие атрибуты зависят не от всего первичного ключа, а только от его части).

Находится ли таблица в 3НФ? Если нет, укажите, какие **транзитивные зависимости** существуют (какой неключевой атрибут зависит от другого неключевого атрибута).

Какая связь существует между заказами и товарами? (1:1, 1:M или M:M). Как это влияет на структуру таблицы?

Часть 2. Приведение к 3НФ (10 минут)

На основе выявленных нарушений разделите исходную таблицу на несколько таблиц, каждая из которых будет находиться в 3НФ. **Обязательно реализуйте связь M:M** между заказами и товарами через промежуточную таблицу.

Форма выполнения — заполните таблицу:

Новая таблица	Столбцы	Первичный ключ (PK)	Внешние ключи (FK)
...	...	...	...

Форма выполнения

Результаты оформите в виде текстового файла или на листе, содержащем:

Ответы на пять вопросов (Часть 1).

Заполненную таблицу с новыми таблицами и ключами (Часть 2).

Тема 2.2. Физическое проектирование баз данных

«Создание схемы базы данных: проектирование физической модели, с указанием типа данных для каждого поля»

Исходная ситуация

Вам дана логическая модель базы данных для системы управления

проектами. Необходимо спроектировать **физическую модель** — выбрать конкретные типы данных, ограничения и значения по умолчанию.

Логическая модель:

Таблица Projects (Проекты)

Поле	Описание
project_id	Уникальный идентификатор проекта
name	Название проекта (не более 100 символов)
description	Описание проекта (может быть длинным)
start_date	Дата начала проекта
end_date	Дата окончания проекта (может быть NULL)
budget	Бюджет проекта (в рублях, с копейками)
is_active	Активен ли проект (TRUE/FALSE)

Таблица Employees (Сотрудники)

Поле	Описание
employee_id	Уникальный идентификатор сотрудника
first_name	Имя (не более 50 символов)
last_name	Фамилия (не более 50 символов)
email	Электронная почта (уникальная, не более 100 символов)
phone	Номер телефона (не более 20 символов)
hire_date	Дата приёма на работу
salary	Зарплата (в рублях, с копейками)

Таблица Assignments (Назначения сотрудников на проекты)

Поле	Описание
assignment_id	Уникальный идентификатор назначения
project_id	Идентификатор проекта (внешний ключ)
employee_id	Идентификатор сотрудника (внешний ключ)
role	Роль в проекте (не более 50 символов)

Поле	Описание
hours_per_week	Количество часов в неделю (целое число)
assigned_date	Дата назначения

Задание

Часть 1. Создание физической модели (10 минут)

На основе логической модели спроектируйте **физическую модель** базы данных. Для каждой таблицы укажите:

Название таблицы.

Список столбцов с **конкретными типами данных**, обоснованием выбора типа и ограничениями (**NOT NULL**, **UNIQUE**, **DEFAULT**, **CHECK** при необходимости).

Первичный ключ (PK).

Внешние ключи (FK) с указанием ссылочной таблицы.

Форма выполнения — заполните таблицу:

Таблица	Столбец	Тип данных	Ограничения / Обоснование
Projects	project_id	SERIAL	PRIMARY KEY, автоинкремент
Projects	name	VARCHAR(100)	NOT NULL — название обязательно
...	...	...	...

Часть 2. Обоснование выбора типов данных (10 минут)

Для каждого столбца таблицы **Employees** напишите **краткое обоснование** (1–2 предложения), почему вы выбрали именно этот тип данных.

Пример оформления:

Столбец	Тип данных	Обоснование
employee_id	SERIAL	Автоинкрементный идентификатор — оптимальное решение для первичного ключа, гарантирует уникальность
first_name	VARCHAR(50)	Имена обычно не превышают 50 символов, VARCHAR экономит место по сравнению с CHAR
...	...	...

Форма выполнения

Результаты оформите в виде **текстового файла** или на **листе**, содержащем:

Заполненную таблицу с типами данных и ограничениями для всех трёх таблиц (Часть 1).

Таблицу с обоснованием выбора типов данных для таблицы `Employees` (Часть 2).

Раздел 3. Основы работы с запросами и оптимизация баз данных

Тема 3.1. Основы SQL и работа с запросами

«Создание SQL-скрипта для создания таблиц и внесения ключей»

Исходная ситуация

Вам необходимо создать базу данных для университета. База данных должна состоять из следующих таблиц:

Таблица `Faculties` (Факультеты)

Поле	Тип	Описание
<code>faculty_id</code>	<code>INTEGER</code>	Уникальный идентификатор факультета (PK)
<code>faculty_name</code>	<code>VARCHAR(100)</code>	Название факультета
<code>dean</code>	<code>VARCHAR(100)</code>	Имя декана

Таблица `Departments` (Кафедры)

Поле	Тип	Описание
<code>department_id</code>	<code>INTEGER</code>	Уникальный идентификатор кафедры (PK)
<code>department_name</code>	<code>VARCHAR(100)</code>	Название кафедры
<code>faculty_id</code>	<code>INTEGER</code>	Идентификатор факультета (FK → <code>Faculties</code>)
<code>head</code>	<code>VARCHAR(100)</code>	Имя заведующего кафедрой

Таблица `Teachers` (Преподаватели)

Поле	Тип	Описание
<code>teacher_id</code>	<code>INTEGER</code>	Уникальный идентификатор преподавателя (PK)
<code>first_name</code>	<code>VARCHAR(50)</code>	Имя
<code>last_name</code>	<code>VARCHAR(50)</code>	Фамилия
<code>department_id</code>	<code>INTEGER</code>	Идентификатор кафедры (FK → <code>Departments</code>)
<code>email</code>	<code>VARCHAR(100)</code>	Электронная почта (уникальный)
<code>hire_date</code>	<code>DATE</code>	Дата приёма на работу

Таблица Courses (Курсы/Дисциплины)

Поле	Тип	Описание
course_id	INTEGER	Уникальный идентификатор курса (PK)
course_name	VARCHAR(100)	Название курса
credits	INTEGER	Количество кредитов
teacher_id	INTEGER	Идентификатор преподавателя (FK → Teachers)

Задание

Часть 1. Создание SQL-скрипта (15 минут)

Напишите **полный SQL-скрипт**, который:

Создаёт все четыре таблицы в правильном порядке (сначала родительские, потом дочерние).

Для каждой таблицы определяет **первичный ключ (PRIMARY KEY)**.

Для таблиц, содержащих внешние ключи, добавляет **внешние ключи (FOREIGN KEY REFERENCES)**.

Добавляет ограничения:

NOT NULL для обязательных полей.

UNIQUE для поля email в таблице Teachers.

Добавляет значения **по умолчанию** (например, CURRENT_DATE для hire_date).

Добавляет **проверочные ограничения (CHECK)** для credits (должно быть > 0) и faculty_id (должен существовать).

Скрипт должен быть синтаксически корректен для PostgreSQL.

Часть 2. Заполнение тестовыми данными (5 минут)

Добавьте в скрипт **SQL-команды INSERT** для заполнения каждой таблицы хотя бы **двумя записями**. Данные придумайте самостоятельно, но они должны быть логически связаны (например, кафедра должна ссылаться на существующий факультет).

Форма выполнения

Результаты оформите в виде **текстового файла**, содержащего **один полный SQL-скрипт** (с комментариями для наглядности).

Тема 3.2 Оптимизация запросов и работы баз данных

«Создание запросов на выборку для получения аналитических отчетов. Создание ограничения для контроля корректности»

Исходная ситуация

В компании ведётся учёт продаж в таблице Sales:

sale_id	product_name	category	seller	quantity	amount	sale_date
1	Ноутбук	Электроника	Иванов И.	2	120000	2026-01-10
2	Телефон	Электроника	Петров П.	3	90000	2026-01-12
3	Книга	Литература	Иванов И.	5	2500	2026-01-15
4	Телефон	Электроника	Петров П.	1	30000	2026-02-05
5	Ноутбук	Электроника	Сидоров С.	1	60000	2026-02-10
6	Книга	Литература	Петров П.	3	1500	2026-02-12
7	Планшет	Электроника	Иванов И.	2	40000	2026-03-01
8	Книга	Литература	Сидоров С.	10	5000	2026-03-05

Задание

Часть 1. Создание аналитических запросов (10 минут)

На основе таблицы Sales напишите SQL-запросы для получения следующих аналитических отчетов:

Общая сумма продаж по каждой категории товаров.

Выведите: category, total_amount. Отсортируйте по убыванию суммы.

Количество проданных единиц товара по каждому продавцу.

Выведите: seller, total_quantity. Отсортируйте по убыванию количества.

Средняя сумма продажи по каждому продавцу.

Выведите: seller, avg_amount (округлённую до 2 знаков). Отсортируйте по убыванию средней суммы.

Самый популярный товар (по количеству проданных единиц) в каждой категории.

(Подсказка: используйте подзапрос или оконные функции)

Выведите: category, product_name, total_quantity.

Часть 2. Создание ограничения для контроля корректности (10 минут)

В таблице Sales отсутствуют ограничения для проверки корректности данных. Добавьте **ограничения CHECK** для следующих полей:

quantity — количество должно быть больше 0.

amount — сумма продажи должна быть больше 0.

sale_date — дата продажи не может быть в будущем (не позже текущей даты).

Напишите **SQL-скрипт**, который:

Создаёт таблицу Sales заново с добавленными ограничениями CHECK.

Заполняет её тестовыми данными (из исходной ситуации).

Форма выполнения

Результаты оформите в виде **текстового файла** или на листе, содержащем:

Четыре аналитических запроса (Часть 1).

SQL-скрипт CREATE TABLE с ограничениями и INSERT (Часть 2).

Критерии оценки выполнения заданий для самостоятельной работы

Оценка «отлично» выставляется студенту, если задание выполнено в полном объеме с соблюдением необходимой последовательности действий; итогом решения задания явилось получение правильных результатов и выводов; в ходе решения задания верно и аккуратно выполнены все вычисления и записи.

Оценка «хорошо» выставляется студенту, если задание выполнено правильно с учетом мелких погрешностей или недочетов.

Оценка «удовлетворительно» выставляется студенту, если задание выполнено правильно не менее чем наполовину, допущены некоторые погрешности или одна грубая ошибка.

Оценки «неудовлетворительно» заслуживает студент, если при решении задания допущены две (и более) грубые ошибки, или задание не решено полностью.

2. Вопросы и задания для промежуточной аттестации
Экзаменационные вопросы и задания для устного (письменного)
экзамена по дисциплине

Экзаменационные вопросы для устного (письменного) экзамена по
дисциплине
(ОК.01, ОК.02, ОК.03, ОК.04, ОК.05)

1. Этапы развития баз данных.
2. Методика проектирования баз данных.
3. Сетевая и иерархическая модели данных.
4. Реляционная модель данных. Основные понятия.
5. Понятие отношения. Свойства отношений
6. Операции над отношениями
7. Ключи и возможные индексы реляционных отношений.
8. Понятие базы данных. Основные характеристики БД
9. Проектирование баз данных с использованием принципов нормализации
10. Понятия функциональной зависимости, полной функциональной зависимости, транзитивности, многозначной зависимости
11. Первая и вторая нормальные формы
12. Третья и четвертая нормальные формы
13. Понятия целостности данных отношения и целостности данных по ссылкам.
14. Семантическое моделирование. ER-диаграммы
15. Правила порождения отношений из модели «сущность-связь». Бинарные связи.
16. Правила порождения отношений из модели «сущность-связь». N-арные связи и иерархические связи.
17. Трехуровневая организация БД. Основные требования к организации БД.
18. Проектирование БД. Инфологическая и даталогическая модели данных.
19. Требования, предъявляемые к СУБД и к серверу баз данных.
20. Основные объекты СУБД
21. Трехзвенная архитектура «клиент-сервер».
22. Архитектура «клиент-сервер». Модель файлового сервера.
23. Архитектура «клиент-сервер». Модель удаленного доступа к данным
24. Архитектура «клиент-сервер». Модель сервера баз данных.
25. Архитектура «клиент-сервер». AS-модель
26. Архитектура «клиент-сервер». Активный сервер
27. CASE-средства для проектирования баз данных.
28. Возможности пакета ERWin. Forward Engineering.
29. Возможности пакета ERWin. Reverse Engineering
30. Возможности пакета ERWin. Функция Complete Compare

- 31. Основные подразделы SQL
- 32. Команды создания схемы данных в SQL
- 33. Команды манипулирования данными в SQL
- 34. Понятие транзакции. Свойства транзакций
- 35. Виды блокировок объектов БД.
- 36. Проблемы блокировок транзакций.
- 37. Методы тиражирования баз данных. 38.
- 38. Постреляционные СУБД

Экзаменационные задания для устного (письменного) экзамена по дисциплине

(ОК.01, ОК.02, ОК.03, ОК.04, ОК.05)

Задание 1.

Компания «СтройПроект» занимается строительством жилых и коммерческих объектов. Руководство приняло решение разработать информационную систему для управления проектами. Вам поручено разработать **концептуальную модель базы данных** для этой системы.

Требования к системе:

Система должна хранить информацию о **проектах** (название, адрес, дата начала, дата окончания, бюджет).

В каждом проекте участвуют **сотрудники** (ФИО, должность, телефон, email).

Каждый сотрудник может быть назначен **руководителем проекта** (один сотрудник — руководитель одного или нескольких проектов).

В рамках проектов выполняются **работы** (название, описание, плановая стоимость, фактическая стоимость, статус).

Каждая работа связана с одним проектом.

Сотрудники могут быть назначены на **несколько работ** в рамках разных проектов.

Для каждой работы может быть назначен один **ответственный сотрудник**.

Задание

Часть 1. Выделение сущностей и атрибутов (5 минут)

На основе описания предметной области определите **не менее 4 сущностей**. Для каждой сущности укажите:

Название сущности.

Набор атрибутов (не менее 3).

Первичный ключ (РК).

Формат оформления:

Сущность	Атрибуты	Первичный ключ (РК)
Проект	Название, Адрес, Дата_начала, Дата_окончания, Бюджет	id_проекта
...	...	...

Задание 2.

Небольшое кафе «Уют» хочет автоматизировать учёт заказов. Система должна хранить информацию о:

Блюдах – название, цена, категория (горячее, салаты, напитки и т.п.), время приготовления.

Заказах – номер заказа, дата и время, статус (новый, готовится, выполнен), столик.

Клиентах – если клиент постоянный, то хранится имя и телефон; для разовых клиентов – только номер столика.

Составе заказа – какие блюда и в каком количестве входят в каждый заказ.

Важно: один заказ может включать несколько блюд, а одно блюдо может присутствовать в разных заказах.

Задание

Часть 1. Определение сущностей, атрибутов и связей (10 минут)

На основе описания предметной области выполните следующие шаги:

Выделите все сущности (не менее 3). Для каждой сущности укажите: Название сущности.

Атрибуты (включая первичный ключ).

Первичный ключ.

Определите связи между сущностями. Для каждой пары сущностей укажите:

Тип связи (1:1, 1:M, M:M).

Где будет располагаться внешний ключ (FK) для реализации связи.

Форма выполнения – заполните две таблицы:

Таблица 1. Сущности и атрибуты

Сущность	Атрибуты	Первичный ключ (РК)
Блюдо	Название, Цена, Категория, Время_приготовления	id_блюда

Сущность	Атрибуты	Первичный ключ (PK)
...	...	...

Таблица 2. Связи между сущностями

Связь	Тип связи	Где располагается FK	Пояснение
Заказ — Блюдо	M:M	В ассоциативной таблице «Состав_заказа»	Один заказ содержит много блюд, одно блюдо может быть в разных заказах
...	...	...	...

Задание 3.

Фитнес-клуб «Энергия» ведёт учёт своих клиентов и посещений. В клубе есть несколько тренировочных залов и групповых программ.

Известно, что клуб хранит информацию о:

Клиентах — их имя, контактный телефон, дата рождения и дата первого посещения.

Абонементы — тип абонемента (например, «Месячный», «Годовой»), стоимость, дата начала и дата окончания действия. У каждого клиента может быть несколько абонементов (если он продлевает), но в каждый момент времени действует только один.

Тренировках — название тренировки (например, «Йога», «Силовая»), продолжительность в минутах, максимальное количество участников.

Посещениях — фиксируется, какой клиент, какую тренировку и когда посетил.

Задание

Часть 1. Выделение сущностей и атрибутов (10 минут)

На основе описания предметной области выделите **сущности** и для каждой из них определите **атрибуты** и **первичный ключ**.

Заполните таблицу:

Сущность	Атрибуты	Первичный ключ (PK)
Клиент	Имя, Телефон, Дата_рождения, Дата_первого_посещения	id_клиента

Сущность	Атрибуты	Первичный ключ (РК)
...	...	...

Задание 4.

Компания «Событие» организует корпоративные мероприятия (конференции, семинары, тренинги). Руководство приняло решение разработать информационную систему для учёта мероприятий, участников и спикеров.

Требования к системе:

Система должна хранить информацию о **мероприятиях**: название, дата проведения, место проведения, тематика, максимальное количество участников.

В системе должны быть зарегистрированы **участники**: ФИО, email, телефон, компания, должность.

Каждое мероприятие имеет одного или нескольких **спикеров**: ФИО, компания, тема выступления, контактный email.

Участники **регистрируются** на мероприятия. Один участник может зарегистрироваться на несколько мероприятий.

Для каждой регистрации фиксируется: дата регистрации, статус (подтверждена/ожидает/отменена), отметка об оплате.

Задание

Часть 1. Выделение сущностей и атрибутов (10 минут)

На основе описания предметной области и требований системы:

Выделите **все сущности** (не менее 3).

Для каждой сущности укажите **атрибуты** (не менее 3 на сущность) и **первичный ключ**.

Форма выполнения — заполните таблицу:

Сущность	Атрибуты	Первичный ключ (РК)
Мероприятие	Название, Дата, Место, Тематика, Макс_участников	id_мероприятия
...	...	...

Задание 5.

Вы разрабатываете базу данных для онлайн-платформы, на которой

размещены образовательные курсы. Платформа должна хранить информацию о курсах, студентах, преподавателях, модулях курсов и успеваемости студентов.

Описание предметной области:

Курс имеет название, описание, уровень сложности (начальный, средний, продвинутый), стоимость и дату создания.

Преподаватель (автор курса) имеет имя, email, специализацию и рейтинг. Один преподаватель может вести несколько курсов.

Студент имеет имя, email, дату регистрации и статус (активный, заблокирован).

Курс состоит из нескольких **модулей**. Каждый модуль имеет название, порядковый номер в курсе и длительность в часах.

Студент может записаться на курс. Для каждой записи фиксируется дата записи, статус (активен, завершён, отменён) и итоговая оценка (если курс завершён).

Один студент может записаться на несколько курсов.

Дополнительное условие:

В системе также хранится информация о **сертификатах**, которые выдаются студентам после успешного завершения курса. Сертификат содержит: уникальный номер, дату выдачи, id студента, id курса, и текстовое поздравление (может быть длинным).

Задание

Часть 1. Построение ER-диаграммы (10 минут)

На основе описания предметной области выделите **все сущности** и **связи** между ними. Постройте ER-диаграмму (можно в текстовом виде, от руки или с использованием обозначений). На схеме отразите:

Названия сущностей (прямоугольники).

Первичные ключи (PK) для каждой сущности.

Внешние ключи (FK) со ссылками на родительские таблицы.

Типы связей между сущностями (1:1, 1:M, M:M).

Форма выполнения — опишите сущности и связи:

Сущности и их атрибуты:

Сущность	Атрибуты (с указанием PK/FK)
Курс	id_курса (PK), Название, Описание, Уровень, Стоимость, Дата_создания, id_преподавателя (FK)
...	...

Связи между сущностями:

Связь	Тип	Пояснение
Преподаватель — Курс	1:M	Один преподаватель ведёт несколько курсов
...	...	...

Задание 6.

Вы разрабатываете базу данных для онлайн-библиотеки. Система должна хранить информацию о книгах, авторах, издательствах и выдаче книг читателям.

Описание предметной области:

Книга имеет уникальный инвентарный номер, название, год издания и количество доступных экземпляров. Каждая книга относится к одному издательству.

Автор имеет уникальный идентификатор, имя и страну. Одна книга может быть написана несколькими авторами, один автор может написать несколько книг.

Издательство имеет уникальный идентификатор, название и город. Одно издательство может выпускать много книг.

Читатель имеет уникальный номер читательского билета, ФИО, телефон и дату регистрации.

Выдача фиксирует факт выдачи книги читателю. Для каждой выдачи указывается: какая книга, какой читатель, дата выдачи, дата возврата (если возвращена) и статус («На руках» или «Возвращена»). Один читатель может иметь несколько выдач, одна книга может выдаваться多次 (в разные периоды).

Задание

Часть 1. Подготовка (2 минуты)

Откройте один из редакторов:

draw.io (рекомендуется): перейдите на сайт app.diagrams.net, нажмите «Создать новую диаграмму» и выберите «Пустой» шаблон.

Lucidchart: перейдите на сайт lucidchart.com, авторизуйтесь и создайте новый пустой документ.

В draw.io в левой панели найдите раздел «Entity Relation» (или «ER») — там находятся готовые фигуры для таблиц и связей.

Задание 7.

Вам дана таблица **Sales** (Продажи) со следующей структурой:

Поле	Тип	Описание
sale_id	INTEGER	Уникальный идентификатор продажи (PK)
product	VARCHAR(100)	Название товара
category	VARCHAR(50)	Категория товара
seller	VARCHAR(100)	Продавец (ФИО)
quantity	INTEGER	Количество проданных единиц
amount	DECIMAL(10,2)	Сумма продажи
sale_date	DATE	Дата продажи

Пример данных:

sale_id	product	category	seller	quantity	amount	sale_date
1	Ноутбук	Электроника	Иванов И.	2	120000	2026-01-10
2	Телефон	Электроника	Петров П.	3	90000	2026-01-12
3	Книга	Литература	Иванов И.	5	2500	2026-01-15
4	Телефон	Электроника	Петров П.	1	30000	2026-02-05
5	Ноутбук	Электроника	Сидоров С.	1	60000	2026-02-10
6	Книга	Литература	Петров П.	3	1500	2026-02-12
7	Планшет	Электроника	Иванов И.	2	40000	2026-03-01
8	Книга	Литература	Сидоров С.	10	5000	2026-03-05

Задание

Часть 1. Написание запросов с HAVING (10 минут)

На основе таблицы Sales напишите SQL-запросы для выполнения следующих задач:

Найти категории товаров, общая сумма продаж которых превышает 100 000 рублей.

Выведите: category, total_amount. Отсортируйте по убыванию общей суммы.

Найти продавцов, у которых общее количество проданных единиц товара больше 5.

Выведите: `seller`, `total_quantity`.

Найти категории, в которых было совершено более 3 продаж (количество записей).

Выведите: `category`, `sales_count`.

Задание 8.

В компании ведётся учёт сотрудников. Данные хранятся в двух таблицах:

Таблица `Employees_Current` — действующие сотрудники:

employee_id	full_name	department	hire_date
1	Иванов И.	IT	2020-03-15
2	Петров П.	Sales	2021-07-01
3	Сидорова С.	HR	2022-01-10
4	Козлов К.	IT	2023-05-20

Таблица `Employees_Archived` — уволенные сотрудники (архив):

employee_id	full_name	department	hire_date
2	Петров П.	Sales	2021-07-01
5	Михайлов М.	Sales	2019-11-01
6	Фёдорова Ф.	IT	2020-09-15
7	Семёнов С.	HR	2018-04-10

Обратите внимание: таблицы имеют одинаковую структуру. Сотрудник Петров П. присутствует в обеих таблицах (сначала работал, потом уволился — дубликат).

Задание

Часть 1. Написание запросов с UNION ALL (10 минут)

На основе приведённых таблиц выполните следующие задания:

Получить полный список всех сотрудников (действующих и уволенных) с указанием статуса.

Добавьте в результат вычисляемое поле `status` со значениями 'Действующий' для таблицы `Employees_Current` и 'Уволен' для

таблицы `Employees_Archived`.

Выведите: `full_name`, `department`, `status`.

Результат отсортируйте по фамилии.

Посчитать общее количество сотрудников (включая дубликаты).

Напишите запрос, который возвращает одно число — общее количество записей в обеих таблицах (с учётом дубликатов).

(Подсказка: используйте `UNION ALL` в подзапросе или в `COUNT`).

Задание 9.

Вам представлена **ER-диаграмма** для системы управления заказами в интернет-магазине (в текстовом описании).

Сущности и атрибуты:

Клиент

Атрибуты: `id_клиента` (PK), ФИО, Телефон, Email

Заказ

Атрибуты: `id_заказа` (PK), Дата_заказа, Сумма

Товар

Атрибуты: `id_товара` (PK), Название, Цена

Состав_заказа (ассоциативная сущность)

Атрибуты: `id_заказа` (FK), `id_товара` (FK), Количество

Связи между сущностями:

Клиент — Заказ: один клиент может сделать много заказов (1:M). Связь реализуется внешним ключом в таблице `Заказ`.

Заказ — Товар: один заказ может содержать много товаров, один товар может входить во многие заказы (M:M). Связь реализуется через ассоциативную таблицу `Состав_заказа`.

Задание

Часть 1. Определение таблиц и их атрибутов (10 минут)

На основе ER-диаграммы перечислите **все таблицы** реляционной базы данных. Для каждой таблицы укажите:

Название таблицы.

Список столбцов (с типами данных — используйте `INTEGER`, `VARCHAR`, `DECIMAL`, `DATE`).

Первичный ключ (PK).

Форма выполнения – заполните таблицу:

Таблица	Столбцы (с типами)	PK
Клиент	<code>id_клиента</code> <code>INTEGER</code> , ФИО <code>VARCHAR(255)</code> , Телефон <code>VARCHAR</code>	<code>id_клие</code>

Таблица	Столбцы (с типами)	РК
т	(20), Email VARCHAR(255)	нта
...	...	...

Задание 10.

Строительная компания «СтройИнвест» занимается возведением жилых и коммерческих объектов. Компания планирует разработать информационную систему для управления проектами. Вам поручено разработать **концептуальную модель базы данных**.

Требования к системе:

Хранить информацию о **проектах** (название, адрес, дата начала, дата окончания, бюджет).

В каждом проекте участвуют **сотрудники** (ФИО, должность, телефон, email).

Один сотрудник может быть назначен **руководителем проекта** (один сотрудник — руководитель одного или нескольких проектов).

В рамках проектов выполняются **работы** (название, описание, плановая стоимость, статус).

Каждая работа связана с **одним проектом**.

Сотрудники могут быть назначены на **несколько работ** в рамках разных проектов (связь М:М).

Для каждой работы может быть назначен один **ответственный сотрудник**.

Задание

Часть 1. Выделение сущностей и атрибутов (10 минут)

На основе описания предметной области выделите **все сущности** (не менее 4). Для каждой сущности укажите:

Название сущности.

Атрибуты (не менее 3 на сущность).

Первичный ключ (РК).

Форма выполнения — заполните таблицу:

Сущность	Атрибуты	Первичный ключ (РК)
Проект	Название, Адрес, Дата_начала, Дата_окончания, Бюджет	id_проекта

Сущность	Атрибуты	Первичный ключ (РК)
...	...	...

Задание 11.

В компании ведётся учёт заказов. Используется одна таблица **Orders** со следующей структурой и данными:

order_id	customer_name	customer_phone	product_name	product_category	product_price	quantity	order_date
101	Иванов И.	89001234567	Ноутбук	Электроника	60000	1	2026-01-10
101	Иванов И.	89001234567	Мышь	Электроника	1500	2	2026-01-10
102	Петрова А.	89007654321	Книга	Литература	500	3	2026-01-12
103	Иванов И.	89001234567	Книга	Литература	500	1	2026-01-15

Проблемы текущей таблицы:

Повторяются данные о клиентах (имя, телефон) для каждого заказа.

Повторяются данные о товарах (название, категория, цена) для каждого заказа.

Существует частичная зависимость: **product_category** зависит только от **product_name**, а не от всего первичного ключа.

Существует транзитивная зависимость: **product_price** зависит от **product_name**, а не от первичного ключа.

Задание

Часть 1. Проверка нормальных форм (10 минут)

Определите, находится ли таблица в 1НФ. Если нет, предложите, как её привести к 1НФ.

(Подсказка: проверьте атомарность значений и уникальность строк).

Определите первичный ключ таблицы (он может быть составным). Объясните свой выбор.

Определите, находится ли таблица в 2НФ. Если нет, укажите, какие частичные зависимости существуют, и объясните, почему они нарушают 2НФ.

Определите, находится ли таблица в 3НФ. Если нет, укажите, какие

транзитивные зависимости существуют.

Задание 12.

В компании «СтройПроект» ведётся учёт сотрудников в таблице `Employees`. Руководство хочет, чтобы база данных автоматически проверяла корректность вводимых данных на уровне СУБД, не полагаясь только на приложение.

Таблица `Employees` имеет следующую структуру:

Поле	Тип	Ограничения
<code>id</code>	SERIAL	PRIMARY KEY
<code>full_name</code>	VARCHAR(255)	NOT NULL
<code>email</code>	VARCHAR(255)	NOT NULL
<code>salary</code>	DECIMAL(10,2)	NOT NULL
<code>hire_date</code>	DATE	NOT NULL
<code>department</code>	VARCHAR(100)	

Дополнительные бизнес-правила, которые должны проверяться при вставке или обновлении данных:

Проверка salary: Зарплата должна быть больше 0. Минимальная зарплата в компании — 10 000 рублей.

Проверка email: Email должен содержать символ @ (базовая проверка).

Проверка hire_date: Дата приёма на работу не должна быть в будущем (не позже текущей даты).

Задание

Часть 1. Создание триггерной функции и триггера (10 минут)

Напишите SQL-скрипт, который:

Создаёт триггерную функцию `validate_employee()`, которая проверяет все три бизнес-правила:

Если `salary` < 10000 — выбрасывать исключение с сообщением «Зарплата не может быть меньше 10000».

Если `email` не содержит символ @ — выбрасывать исключение «Некорректный email».

Если `hire_date` > CURRENT_DATE — выбрасывать исключение «Дата приёма не может быть в будущем».

Создаёт триггер `employee_validation`, который выполняет эту функцию перед вставкой или обновлением каждой строки в таблице `Employees`.

Подсказка: используйте `NEW` для доступа к вставляемым/обновляемым значениям.

Критерии оценки устного (письменного) экзамена по дисциплине

Оценка «отлично» выставляется, если ответ студента полный, логичный, структурированный и исчерпывающий. Студент демонстрирует глубокое понимание теоретического материала, свободно и корректно оперирует профессиональной терминологией. Практическое задание выполнено в полном объеме.

Оценка «хорошо» выставляется, если ответ студента достаточно полный и логичный, демонстрирует уверенное владение теоретическим материалом. Допускаются незначительные шероховатости в формулировках или мелкие неточности в терминологии. Практическое задание выполнено, предложенное решение в целом верное и охватывает основные аспекты задачи, но содержит незначительные недочеты в аргументации или выборе оптимальных инструментов.

Оценка «удовлетворительно» выставляется, если ответ студента фрагментарный, демонстрирует поверхностное понимание теоретического материала и неуверенное владение базовой терминологией. Логика изложения нарушена. Практическое задание выполнено с существенными недочетами: предложенное решение охватывает лишь часть задачи, студент испытывает трудности с обоснованием выбора методов и не видит взаимосвязи между этапами работы.

Оценка «неудовлетворительно» выставляется, если ответ студента не соответствует поставленным вопросам, демонстрирует отсутствие понимания теоретического материала и неумение применять его на практике. Студент не владеет базовой профессиональной терминологией. Практическое задание не выполнено или содержит грубые ошибки.