

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«24» июня 2025 г.
Протокол № 2

**Рабочая программа дисциплины (модуля)
«Системы управления базами данных»**

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Backend-разработка

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Срок освоения программы: 2 года

Год набора: 2025

**Москва
2025**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения.....	4
3. Тематический план.....	6
4. Содержание дисциплины (модуля).....	6
5. Учебно-методическое обеспечение	7
6. Материально-техническое обеспечение	7
7. Методические и оценочные материалы	9

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Системы управления базами данных» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – магистратура по специальности 02.04.01 Математика и компьютерные науки, профиль Backend-разработка, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 810 от 23.08.2017 года.

Изучение дисциплины (модуля) «Системы управления базами данных» позволяет овладеть ключевыми навыками эффективного хранения, обработки и анализа данных, что является основой современных информационных технологий. Эти знания обеспечивают возможность создания надежных и безопасных информационных систем, востребованных во многих сферах деятельности.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки магистратуры по направлению 02.04.01 Математика и компьютерные науки, профиль Backend-разработка и входит в обязательную часть Блока 1.

Дисциплина (модуль) изучается на 2 курсе в 3 семестре, доступна для прохождения при условии успешного завершения дисциплин (модулей) «Java Spring (Разработка веб-приложений на Java с использованием Spring)», «Алгоритмы и структуры данных. Часть 2», «Промышленная разработка».

Цель изучения дисциплины (модуля): формирование у студентов знаний и навыков, необходимых для проектирования, разработки и управления базами данных, а также для эффективного извлечения и обработки данных.

Задачи изучения дисциплины (модуля):

- изучить основные принципы сетевого взаимодействия и концепции неблокирующего ввода-вывода в Java для создания производительных приложений;
- освоить методы работы с реляционными базами данных на разных уровнях абстракции с использованием JDBC, Spring JDBC, Jdbi и jOOQ;
- познакомиться с принципами организации обмена сообщениями и интеграции через Spring JMS и Spring AMQP;
- научиться проектировать и реализовывать REST и gRPC сервисы на базе Spring Boot с настройкой и оптимизацией логирования;
- развить навыки настройки и интеграции систем обмена сообщениями (ActiveMQ, RabbitMQ, Kafka) в составе Spring-приложений.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

- принципы SQL и NoSQL;
- CAP-теорему и PACELC;
- принципы ACID и BASE;
- архитектуру популярных СУБД.

уметь:

- применять БД типа "ключ-значение" и документные БД;
- использовать колоночные БД;
- работать с распределенными БД.

владеть:

- навыками администрирования сервера с базой данных;
- навыками проектирования архитектуры хранения данных;
- навыками выбора нужного типа БД под конкретную задачу.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
УК-1.	Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, выработать стратегию действий	УК-1.1.	Знает основные принципы системного подхода и методы критического анализа, а также теоретические основы стратегического планирования и принятия решений
		УК-1.2.	Умеет применять методы системного анализа для выявления ключевых проблемных ситуаций, формулировать гипотезы и разрабатывать альтернативные стратегии действий на основе полученных данных
		УК-1.3.	Имеет практический опыт в проведении анализа реальных проблемных ситуаций в рамках проектов, способен выработать и обосновать стратегии действий, учитывая различные аспекты и последствия
ОПК-2.	Способен создавать и исследовать новые математические модели в естественных науках, совершенствовать и разрабатывать концепции, теории и методы	ОПК-2.1.	Знает основные математические модели и методы, используемые в естественных науках, включая статистическое моделирование, дифференциальные уравнения и численные методы, а также современные подходы к исследованию и анализу данных
		ОПК-2.2.	Умеет разрабатывать и адаптировать математические модели для решения конкретных проблем в естественных науках, проводить их анализ и верификацию, а также интерпретировать полученные результаты в контексте научных исследований
		ОПК-2.3.	Имеет практический опыт создания и исследования математических моделей в рамках научных проектов или исследований, включая участие в публикациях, конференциях

			или коллаборациях, где были разработаны и апробированы новые концепции и методы
ПК-2.	Способен математически корректно ставить естественнонаучные и прикладные задачи	ПК-2.1.	Знает основные методы и подходы к математическому моделированию, а также теоретические основы естественных и прикладных наук, необходимые для корректной формулировки задач
		ПК-2.2.	Умеет анализировать практические ситуации и формулировать на их основе математические модели, включая выбор адекватных методов решения и формулировку условий задачи
		ПК-2.3.	Имеет практический опыт в разработке и решении математических задач в рамках проектов или научных исследований, где были успешно поставлены и решены естественнонаучные и прикладные задачи
ПК-6.	Способен разрабатывать программное обеспечение для решения прикладных задач в сфере профессиональной деятельности	ПК-6.1.	Знает основные языки программирования, методы разработки программного обеспечения, а также принципы проектирования и архитектуры программных систем, применяемых в конкретной предметной области
		ПК-6.2.	Умеет анализировать прикладные задачи, разрабатывать алгоритмы и реализовывать их в виде программного обеспечения, используя современные инструменты и технологии, а также проводить тестирование и отладку созданных решений
		ПК-6.3.	Имеет практический опыт разработки программного обеспечения в рамках реальных проектов, включая участие в командах, где были успешно реализованы решения для конкретных прикладных задач в сфере профессиональной деятельности

3. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		<i>Очная форма</i>				
		Аудиторная работа		Контроль	Самостоятельная работа	
Лекции	Семинары (практические занятия)					
1	Реляционная СУБД PostgreSQL	12	12		46	Домашнее задание
2	Распределенные системы	9	9		40	Домашнее задание Коллоквиум
3	Базы данных NoSQL и их разновидности	9	9		40	Домашнее задание
	<i>Зачет с оценкой</i>			4		
	Итого:	30	30	4	126	
	Объем дисциплины (модуля) (в ак. ч.)	190				
	Объем дисциплины (модуля) (в зач. ед.)	5				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Реляционная СУБД PostgreSQL	Введение в СУБД Транзакции в РСУБД Согласованность данных Индексы PostgreSQL: теория и практика Масштабирование PostgreSQL. Резервное копирование и репликация
2	Распределенные системы	Распределенные СУБД: Apache Cassandra Распределенные СУБД: MongoDB Очереди сообщений: Kafka CAP-теорема и принципы PACELC
3	Базы данных NoSQL и их разновидности	KV хранилища. Redis Хранилище S3 Аналитические БД. ClickHouse NewSQL и современные тренды

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Нестеров, С. А. Базы данных : учебник и практикум для вузов / С. А. Нестеров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 258 с. — (Высшее образование). — ISBN 978-5-534-18107-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560753>.

2. Маркин, А. В. Базы данных. PostgreSQL : учебник для вузов / А. В. Маркин. — Москва : Издательство Юрайт, 2025. — 828 с. — (Высшее образование). — ISBN 978-5-534-21779-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/582050>.

Дополнительная литература:

1. Рогов, Е.В. PostgreSQL 16 изнутри : практическое руководство / Е. В. Рогов. — Москва : ДМК Пресс, 2024. - 666 с. – ISBN 978-5-93700-305-8. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2205083>.

2. Карпова, И. П. Базы данных : учебное пособие / И. П. Карпова. - Санкт-Петербург : Питер, 2021. - 240 с. - (Серия «Учебное пособие»). - ISBN 978-5-4461-9681-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1857026>.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- механическими калькуляторами;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и

обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование»	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		

Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Системы управления базами данных» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, домашние задания, коллоквиум, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Семинар — это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Коллоквиум – устные ответы на вопросы, список которых известен студенту заранее.

В процессе подготовки к коллоквиуму необходимо проанализировать учебные
Электронный документ

материалы, ознакомившись с лекциями, учебниками и дополнительными источниками, акцентируя внимание на ключевых темах. Рекомендуется создать структурированные конспекты, выделяя основные идеи, термины и формулы.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Системы управления базами данных»

Оценивание уровня учебных достижений обучающихся по дисциплине (модулю) осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме *зачета с оценкой*, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом:

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Зачтено	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину (модуль). Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
9	Отлично	Зачтено	
8	Отлично	Зачтено	
7	Хорошо	Зачтено	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически
6	Хорошо	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
5	Удовлетворительно	Зачтено	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	Зачтено	
3	Не сдан	Не зачтено	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	Не зачтено	
1	Не сдан	Не зачтено	

Дисциплина (модуль) «Системы управления базами данных» оценивается следующим образом:

Активность	Вес	Описание
Домашние задания	60%	За каждое из заданий можно набрать 10 баллов
Коллоквиум	40%	Устные ответы на вопросы, список которых известен студенту заранее

Формула расчёта итоговой оценки по дисциплине (модулю) «Системы управления базами данных»: $\langle 0,6 \times \text{среднее за домашние задания} + 0,4 \times \text{коллоквиум} \rangle$.

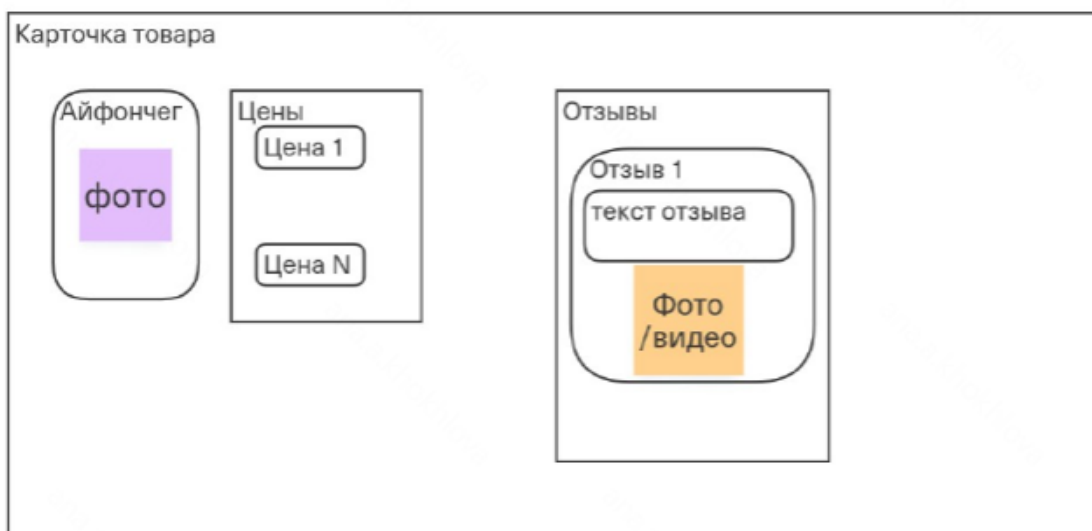
Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание

Задача.

Есть интернет магазин и у него есть товары, у каждого товара есть представление в виде карточки, в карточках есть имена товаров, цены, есть отзывы. Отзывы могут содержать текст, картинки и видео



Подзадача1.

Реализовать схему данных в cassandra, подходящую для реализации записи информации о товарах, чтении карточки товара, записи отзывов, удаления отзывов, чтения отзывов

Подзадача2.

Реализовать приложение на java, которое из файла config читает креды cassandra кластера, если нет таблиц их создаёт, реализует http интерфейс, позволяющий получить информацию о товаре, записать информацию о товаре, прочитать отзывы, записать отзывы.

Формат файла config

```
contactpoint: ip|fqdn[,ip|fqdn]
user: username
password: password
keyspace: keyspace_name
```

Http ручки которые должны быть реализованы:

1. /get/card
2. /get/review
3. /list/reviews
4. /list/cards
5. /put/card
6. /put/review

Электронный документ

Домашнее задание

Задание:

Вы разработчик приложения для онлайн-магазина электронной техники. Вам необходимо реализовать систему хранения данных пользователей и товаров магазина с использованием MongoDB.

Задание состоит из двух частей:

Часть I. Создание структуры базы данных и наполнение её тестовыми данными

Требования:

1. Создание базы данных. Назовите базу данных `shop_db`.
2. Создание коллекции. Внутри базы данных создайте две коллекции:
 - Коллекция `users`: хранит информацию о зарегистрированных пользователях сайта.
 - Коллекция `products`: хранит каталог товаров магазина.

Структура документа `users`: Минимально необходимая структура:

`_id`: уникальный идентификатор пользователя (использовать встроенный объект `ObjectId`).

`username`: имя пользователя.

`email`: email адрес пользователя.

`registration_date`: дата регистрации (использовать тип `Date`).

`address`: адрес доставки (может включать город, улицу, дом и квартиру).

`orders_count`: количество сделанных заказов.

Пример записи:

```
{
  "_id": ObjectId(),
  "username": "Alexander",
  "email": "alex@mail.ru",
  "registration_date": ISODate("2023-10-15T12:34:56Z"),
  "address": {
    "city": "Москва",
    "street": "Краснопрудная ул.",
    "house_number": "16",
    "apartment": "22"
  },
  "orders_count": 3
}
```

Структура документа `products`: Минимальная структура:

`_id`: уникальный идентификатор товара.

`name`: название товара.

`price`: цена товара.

category: категория товара (например, телефоны, ноутбуки, планшеты).
stock_quantity: количество единиц товара на складе.
rating: рейтинг товара (число от 0 до 5).

Пример записи:

```
{
  "_id": "ObjectId()",
  "name": "iPhone XR",
  "price": 50000,
  "category": "телефоны",
  "stock_quantity": 10,
  "rating": 4.5
}
```

3. Наполнение базового набора данных.

Добавьте минимум пять разных пользователей и десять разных товаров в соответствующие коллекции.

Часть II. Работа с базой данных и выполнение операций

Задача №1: Поиск данных

Напишите запросы для вывода списка пользователей, проживающих в городе Москва. Получите список товаров категории "телефоны", стоимость которых меньше 30 тысяч рублей. Отсортируйте товары по убыванию рейтинга и выведите первые три лучших продукта.

Задача №2: Агрегация данных

Рассчитайте среднее значение цены товаров каждой категории. Определите общее количество товаров каждого типа ("телефон", "ноутбук", "планшет"). Выведите список пользователей, совершивших более трех покупок, отсортированных по количеству заказов.

Задача №3: Индексация и оптимизация производительности

Создайте индексы для полей category и price в коллекции продуктов. Выполните запрос на выборку товаров определенной категории и посмотрите на производительность запроса с помощью оператора .explain().

Формат сдачи ДЗ:

В качестве решения части 1 предоставьте код на Python/Java для создания структуры и наполнения СУБД

В качестве решения части 2 нужно предоставить список запросов к СУБД

Домашнее задание

Задание:

Вы разрабатываете сервис погоды. За последнее время увеличилось количество новых пользователей. Как следствие увеличилась нагрузка на основное хранилище данных и увеличилось время ответа пользователю. Для решения данных проблем вам нужно реализовать стратегию Cache-aside кеширования с использованием Redis.

Задание 1.

Разверните Redis (хранилище для кеша) и PostgreSQL (основное хранилище данных)

Для PostgreSQL выполните инициализацию таблицы weather путем запуска скрипта init.sql

В PostgreSQL создастся таблица weather со следующими атрибутами:

- city – название города
- temperature_c – температура в городе
- humidity_pct – влажность воздуха
- updated_at – время последнего обновления записи

В таблицу weather будут добавлены данные из скрипта

Задание 2.

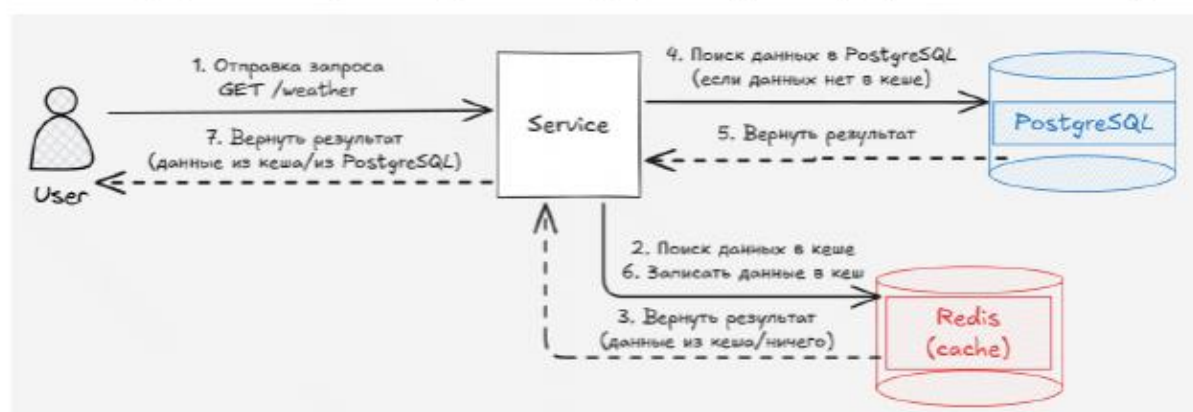
Разработайте метод **GET /weather** в соответствии с логикой со схемы ниже

В качестве query параметров **GET /weather** должен принимать параметр city

Логика метода **GET /weather**:

1. Клиент отправляет запрос **GET /weather?city=X**, где X – название города
2. Запрос отправляется в Redis для поиска данных в кеше по ключу **city**
 - 2.1 Если по ключу city в кеше найдены данные, то результат возвращается клиенту
 - 2.2 Если по ключу city в кеше данные не найдены, то запрос отправляется в PostgreSQL
3. PostgreSQL возвращает результат запроса, результат запроса сохраняется в кеш
4. Пользователю возвращается результат обработки запроса (данные из кеша/из PostgreSQL)

В качестве результата **GET /weather** должен возвращать все данные для указанного ключа **city**



Формат сдачи ДЗ:

В качестве решения задания 1 и 2 предоставьте код на Python/Java

Блок 1. MongoDB

1. Архитектура хранения данных. Опишите модель хранения данных в MongoDB. Какие основные структурные единицы? В чём преимущества и ограничения данного подхода?
2. Подходы к моделированию данных. Сравните Embedded и Reference модели проектирования документов. В каких сценариях предпочтительнее каждый из подходов? Приведите примеры использования и охарактеризуйте компромиссы в производительности, целостности данных и гибкости схемы.
3. Индексация данных. Какие основные типы индексов поддерживает MongoDB? Опишите назначение и использование различных типов индексов. Как индексы влияют на производительность запросов и операции записи?
4. Агрегация и анализ данных. Объясните принцип работы агрегационных конвейеров (aggregation pipelines) в MongoDB. Какие основные этапы (stages) используются для фильтрации, группировки, сортировки и объединения данных? Чем агрегация отличается от простых запросов find()?
5. Системные поля и идентификация. Какова роль поля `_id` в документах MongoDB? Из чего состоит стандартный ObjectId и как его структура влияет на сортировку, уникальность и возможности шардирования? Можно ли использовать кастомные значения для `_id`?

Блок 2. Redis

1. Архитектура хранения данных. Как Redis организует хранение данных? Какие структуры данных он поддерживает и как они влияют на выбор сценариев использования? Опишите преимущества и недостатки in-memory подхода с точки зрения производительности, масштабируемости и стоимости.
2. Архитектурный выбор: однопоточная модель. Почему Redis использует однопоточную модель обработки команд? Какие проблемы решает этот подход и как он влияет на атомарность операций? Какие механизмы компенсируют потенциальные ограничения производительности в высоконагруженных системах?
3. Обеспечение персистентности данных. Какие два основных механизма сохранения данных на диск поддерживает Redis? Сравните подходы RDB (snapshots) и AOF (append-only file) по критериям: надежность, производительность, влияние на восстановление после сбоя и потребление дискового пространства. Как выбрать подходящую стратегию?
4. Оптимизация сетевого взаимодействия: Pipelining. Объясните концепцию пайплайнинга в Redis. Приведите пример использования и оцените потенциальный выигрыш в производительности.
5. Транзакционность и атомарные операции. Какие команды и конструкции используются для работы с транзакциями в Redis? В чем заключаются ключевые отличия транзакций Redis от классических реляционных? Какие ограничения следует учитывать при их использовании?

Блок 3. S3-хранилища

1. **Эволюция систем хранения.** Сравните блочные, файловые и объектные хранилища по ключевым критериям: единица данных, структура организации данных, протоколы доступа, масштабируемость и оптимальные сценарии использования.
2. **Архитектура S3.** Опишите трехкомпонентную архитектуру S3: сетевой уровень/API Gateway, сервис хранения данных и сервис метаданных. Какие задачи решает каждый компонент и какие технологии используются (например, Erasure Coding, Consistent Hashing)?
3. **Типы бакетов.** В чем заключаются ключевые различия между универсальными бакетами (General-purpose), бакетами с директориями (Directory buckets) и табличными бакетами (Table buckets)? Для каких рабочих нагрузок предназначен каждый тип?
4. **Классы хранения S3.** Опишите иерархию классов хранения от "горячих" к "холодным" данным. Чем отличаются гарантии доступности, стоимость и скорость доступа между S3 Standard, S3 Standard-IA, S3 Glacier и S3 Express One Zone?
5. **Структура объекта S3.** Что такое объект в S3 и из каких компонентов он состоит? В чем разница между системными и пользовательскими метаданными, а также между метаданными и тегами объекта?

Блок 4. Apache Kafka

1. **Эволюция систем обмена сообщениями.** Сравните традиционные очереди сообщений (Message Queues), системы публикации-подписки (Pub-Sub) и потоковые платформы (Streaming Platforms) по ключевым критериям: модель данных, масштабируемость, гарантии порядка доставки и возможность повторной обработки данных.
2. **Проблемы прямой интеграции в распределенных системах.** Какие три основные группы проблем решает внедрение брокера (потоковой платформы) вместо прямого взаимодействия между продюсерами и консьюмерами? Кратко охарактеризуйте каждую.
3. **Архитектурные компоненты Kafka.** Опишите назначение и взаимодействие основных компонентов архитектуры Kafka: Producer API, Consumer API, Kafka Connect, Kafka Streams и ksqlDB.
4. **Топики и партиции.** Что такое топик и партиция в Kafka? Какую роль играет ключ (key) сообщения при определении партиции для записи? Каковы последствия выбора стратегии с ключом и без ключа для порядка доставки и балансировки нагрузки?
5. **Репликация и надежность.** Опишите процесс репликации данных в Kafka. Что такое ISR, лидер, фолловер и High Watermark? Как настройка `acks` на продюсере влияет на гарантии доставки и задержку?

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Какой из следующих методов резервного копирования является наиболее безопасным? а) Полное резервное копирование б) Инкрементальное резервное копирование в) Дифференциальное резервное копирование г) Локальное резервное копирование	a	УК-1
2.	Какой из следующих уровней согласованности в Cassandra обеспечивает наименьшую задержку? а) QUORUM б) ONE в) ALL г) LOCAL_QUORUM	b	ПК-6
3.	Что из следующего является основным преимуществом документных баз данных, таких как MongoDB? а) Строгая схема б) Высокая производительность при сложных запросах в) Гибкость в структуре данных г) Поддержка транзакций	c	ОПК-2
4.	Какой из следующих принципов описывает CAP-теорема? а) Консистентность, доступность, разделяемость б) Консистентность, доступность, партиционирование в) Консистентность, доступность, производительность г) Консистентность, отказоустойчивость, скорость	b	УК-1
5.	Что из следующего является характеристикой ACID? а) Доступность б) Согласованность в) Отказоустойчивость г) Масштабируемость	b	ОПК-2
6.	Какой тип базы данных подходит для хранения больших объемов неструктурированных данных?	Нереляционная	УК-1
7.	Как называется процесс, при котором данные из одной базы данных переносятся в другую?	Миграция	ОПК-2
8.	Какой инструмент можно использовать для мониторинга производительности СУБД?	pgAdmin (или аналог)	ПК-6
9.	Какой тип индекса в реляционных СУБД позволяет ускорить поиск данных?	В-дерево	ПК-2
10.	Как называется модель, которая описывает взаимодействие клиентских и серверных приложений?	Клиент-серверная модель	ОПК-2
11.	Какой термин используется для обозначения непрерывного доступа к данным в распределенных системах?	Доступность	УК-1
12.	Какой тип СУБД использует ключ-значение для хранения данных?	Key-Value	ПК-2

13.	Как называется метод, который позволяет выполнять несколько операций в рамках одной транзакции?	Атомарность	ОПК-2
14.	Какой принцип описывает возможность системы продолжать работу даже при сбоях?	Отказоустойчивость	ПК-2
15.	Как называется метод, используемый в MongoDB для горизонтального масштабирования?	Шардинг	УК-1
16.	В контексте проблемной ситуации с высокими требованиями к согласованности данных и транзакциям, какой тип СУБД следует выбрать для стратегии действий: реляционную или NoSQL?	реляционная СУБД	УК-1
17.	Укажите математическую модель, лежащую в основе CAP-теоремы для анализа компромиссов в распределенных системах.	Тройная дилемма/ тройная дилемма/ triple trade-off	ОПК-2
18.	Укажите количество свойств ACID, обеспечивающих надежность транзакций в реляционных СУБД вроде PostgreSQL.	4/ Четыре/ четыре	ПК-2
19.	Назовите инструмент PostgreSQL для создания логической репликации данных между серверами в распределенной среде.	pg_logical/ логическая репликация	ПК-6
20.	Назовите язык запросов, используемый в MongoDB для разработки программного обеспечения с гибкой схемой документов.	JSON-подобный/ document query language	ПК-6