

**Федеральное государственное бюджетное учреждение науки
Институт машиноведения им. А.А. Благонравова
Российской академии наук
(ИМАШ РАН)**

УТВЕРЖДАЮ
Заместитель директора по вопросам
подготовки кадров высшей
квалификации ИМАШ РАН
к.ф.-м.н. Жданов Э.Р.
« 01 » _____ 2026 года
М.П.



**Комплект оценочных материалов
по дисциплине «Современные технологии баз данных»**

Направление подготовки:	02.04.03 Математическое обеспечение и администрирование информационных систем
Профиль подготовки:	Искусственный интеллект и интеллектуальная обработка данных
Квалификация:	Магистр
Форма обучения	Очная

Содержание

1. Общие положения.....	3
2. Результаты освоения дисциплины, подлежащие проверке.....	3
3. Контроль и оценка освоения учебной дисциплины.....	4
4. Оценочные материалы для проведения текущего контроля успеваемости.....	4
5. Оценочные материалы для проведения промежуточной аттестации.....	7

1. Общие положения

Комплект оценочных материалов (далее - ОМ) предназначен для контроля и оценки образовательных достижений обучающихся, освоивших программу дисциплины «Современные технологии баз данных».

ОМ включают контрольные материалы для проведения текущего контроля успеваемости и промежуточной аттестации в форме экзамена.

ОМ разработаны на основании:

- основной профессиональной образовательной программы по направлению подготовки 02.04.03 Математическое обеспечение и администрирование информационных систем;

- рабочей программы дисциплины «Современные технологии баз данных».

2. Результаты освоения дисциплины, подлежащие проверке

В результате освоения дисциплины обучающийся должен обладать следующими компетенциями

ОПК-1	Способен находить, формулировать и решать актуальные проблемы фундаментальной и прикладной информатики и информационных технологий
ОПК-1.2	Умеет применять современные инструменты и технологии для решения актуальных проблем в сфере анализа данных и искусственного интеллекта
ОПК-1.3	Владеет навыками самостоятельного изучения и освоения новых инструментов и технологий, применяемых в области искусственного интеллекта и обработки данных
ОПК-2	Способен проектировать, разрабатывать и внедрять программные продукты и программные комплексы различного назначения
ОПК-2.3	Владеет практическими инструментами и технологиями для разработки и внедрения программных продуктов и программных комплексов различного назначения

Код индикатора компетенции	Планируемые результаты обучения по дисциплине
ОПК-1.2	<i>Знать:</i> - архитектурные особенности реляционных и нереляционных СУБД, принципы работы векторных баз данных, методы оптимизации запросов и масштабирования. <i>Уметь:</i> - проектировать структуру БД под задачи анализа данных и ИИ, писать сложные SQL-запросы, использовать NoSQL API, интегрировать Python с разными СУБД для ETL. <i>Иметь практический опыт:</i> - подготовки датасетов для обучения моделей ИИ с помощью SQL и Python.
ОПК-1.3	<i>Знать:</i> - архитектурные различия между типами СУБД (реляционные, документоориентированные, графовые, векторные). <i>Уметь:</i> - самостоятельно осваивать новые типы СУБД, анализировать документацию, адаптировать ETL-процессы. <i>Иметь практический опыт:</i> - перехода между разными СУБД при смене профиля задачи.
ОПК-2.3	<i>Знать:</i> - методы оптимизации производительности запросов, принципы выбора между SQL и NoSQL для ИИ-систем. <i>Уметь:</i>

	- проектировать архитектуру хранения данных как часть ИИ-продукта, анализировать и оптимизировать планы выполнения запросов. <i>Иметь практический опыт:</i> - выбора между SQL и NoSQL решениями при проектировании архитектуры системы ИИ.
--	---

3. Контроль и оценка освоения учебной дисциплины

Элемент учебной дисциплины	Текущий контроль		Промежуточная аттестация	
	Форма контроля	Проверяемые идк	Форма контроля	Проверяемые идк
Тема 1. Введение в современные СУБД и SQL	Отчет по лабораторному практикуму	ОПК-1.2 ОПК-1.3		
Тема 1. Введение в современные СУБД и SQL	Отчет по практикуму по решению задач	ОПК-1.2 ОПК-1.3		
Тема 2. Нереляционные базы данных (NoSQL)	Отчет по лабораторному практикуму	ОПК-1.2 ОПК-1.3		
Тема 2. Нереляционные базы данных (NoSQL)	Отчет по практикуму по решению задач	ОПК-1.2 ОПК-1.3		
Тема 3. Векторные базы данных и поиск по сходству	Отчет по лабораторному практикуму	ОПК-1.2 ОПК-1.3		
Тема 3. Векторные базы данных и поиск по сходству	Отчет по практикуму по решению задач	ОПК-1.2 ОПК-1.3		
Тема 4. Оптимизация производительности и интеграция с ИИ	Отчет по лабораторному практикуму	ОПК-1.2 ОПК-2.3		
Тема 4. Оптимизация производительности и интеграция с ИИ	Отчет по практикуму по решению задач	ОПК-1.2 ОПК-2.3		
			Экзамен	ОПК-1.2 ОПК-1.3 ОПК-2.3

4. Оценочные материалы для проведения текущего контроля успеваемости

Показатели и критерии оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Лабораторный практикум
15-13 – работа выполнена в срок, самостоятельно, правильно поняты и использованы соответствующие формулы, правильно определены соответствующие спецификации, использована требуемая информация, правильно выполнены требуемые расчеты, правильно выбраны совместимые комплектующие, сделаны необходимые выводы, хорошо аргументированы, даны исчерпывающие ответы на все поставленные вопросы; 12-9 – работа выполнена в срок, самостоятельно, правильно поняты и использованы соответствующие формулы, правильно определены соответствующие спецификации, использована требуемая информация, правильно выполнены требуемые расчеты, правильно выбраны совместимые комплектующие, необходимые выводы сделаны частично, хорошо аргументированы, даны ответы на все поставленные вопросы;

8-5 – работа выполнена в срок, в основном самостоятельно, использованы соответствующие формулы; определены соответствующие спецификации, имеются ошибки в расчетах; выбраны совместимые комплектующие необходимые, выводы сделаны частично, слабо аргументированы, даны ответы не на все вопросы;
 4-1 – обучающийся подготовил работу несамостоятельно или не завершил в срок, описание спецификации содержит незначительные ошибки, выводы и ответы на вопросы отсутствуют.

Практикум по решению задач

10-8 – работа и отчет выполнены в срок, самостоятельно, правильно определены соответствующие спецификации, использована требуемая информация, правильно выполнены требуемые расчеты, сделаны необходимые выводы, хорошо аргументированы, даны исчерпывающие ответы на все поставленные вопросы;
 7-6 – работа и отчет выполнены в срок, самостоятельно, правильно определены соответствующие спецификации, использована требуемая информация, правильно выполнены требуемые расчеты, необходимые выводы сделаны частично, хорошо аргументированы, даны ответы на все поставленные вопросы;
 5-3 – работа и отчет выполнены в срок, в основном самостоятельно, выводы сделаны частично, слабо аргументированы, даны ответы не на все вопросы;
 2-0 – обучающийся подготовил работу и отчет несамостоятельно и/или не завершил в срок, описание содержит ошибки, выводы и ответы на вопросы отсутствуют.

Порядок начисления баллов

Если студент набрал достаточное количество баллов ТКУ по дисциплине (более 70 баллов) в течение семестра, он может отказаться от сдачи экзамена, и результат будет выставлен автоматически по его согласию.

Типовые контрольные задания и иные материалы, необходимые для процедуры оценивания знаний, умений, навыков и(или) опыта деятельности

Типовые задания для практикумов по решению задач

Тема 1. Введение в современные СУБД и SQL

Задача 1.

Проектирование схемы реляционной базы данных:

- изучить структуру датасета (например, датасет по продажам или фильмам с Kaggle);
- выделить основные сущности (например, Users, Products, Orders, Categories).
- нарисовать ER-диаграмму (схему «сущность-связь»), определить первичные и внешние ключи.
- написать SQL-скрипты (CREATE TABLE) для создания спроектированных таблиц в СУБД (PostgreSQL/MySQL).

Тема 2. Нереляционные базы данных (NoSQL)

Задача 1.

Моделирование данных и работа с MongoDB.

- взять ту же предметную область, что и в Теме 1;
- спроектировать структуру коллекций MongoDB для хранения данных о пользователях и их заказах, используя подход «вложения» (embedding) для связи «один-ко-многим» (например, хранить массив заказов прямо в документе пользователя);
- написать Python-скрипт для загрузки данных из CSV/JSON в MongoDB;
- написать сложные запросы с использованием агрегационного пайплайна (\$match, \$group, \$project);
- найти топ-5 самых продаваемых товаров за последний месяц и рассчитать общую выручку по каждому.

Тема 3. Векторные базы данных и поиск по сходству

Задача 1.

Реализация пайплайна векторного поиска:

- установить библиотеку sentence-transformers.
- загрузить датасет текстов (например, описания товаров или статей);
- написать скрипт для генерации векторных представлений (эмбеддингов) для каждого текста с помощью предобученной модели (например, all-MiniLM-L6-v2);
- загрузить полученные векторы и соответствующие им тексты/ID в векторную базу данных (например, ChromaDB, Milvus или Weaviate);
- семантический поиск: Написать функцию поиска: принять на вход текстовый запрос, сгенерировать его эмбеддинг и найти в БД наиболее похожие документы по косинусному расстоянию.

Тема 4. Оптимизация производительности и интеграция с ИИ

Задача 1.

Проектирование архитектуры хранения данных для ИИ-системы:

Описать гипотетический ИИ-продукт (например, рекомендательная система интернет-магазина).

Спроектировать архитектуру хранения данных: Метаданные товаров и пользователей $\rightarrow$ Реляционная СУБД (PostgreSQL).

История действий пользователей $\rightarrow$ NoSQL/Колоночная БД для логов.

Векторы эмбеддингов товаров $\rightarrow$ Векторная БД.

Нарисовать схему ETL-процесса: как данные из разных источников попадают в модель и как результаты возвращаются пользователю.

Типовые задания для лабораторных практикумов

Тема 1. Введение в современные СУБД и SQL

Задача 1.

Написание базовых и аналитических SQL-запросов.

- загрузить тестовые данные в созданные таблицы.
- вывести список всех товаров определенной категории с фильтрацией по цене (SELECT ... FROM ... WHERE);
- посчитать среднюю стоимость заказа для каждого пользователя (GROUP BY, агрегатные функции AVG, COUNT);
- вывести список заказов с именами пользователей и названиями товаров, используя INNER JOIN;
- найти пользователей, которые не совершали покупок, используя LEFT JOIN и проверку на NULL.

Тема 2. Нереляционные базы данных (NoSQL)

Задача 1.

CRUD-операции и моделирование связей:

- создать базу данных и коллекцию в MongoDB;
- вставить несколько документов вручную или через скрипт;
- найти документы по сложным критериям (например, пользователи из определенного города, старше N лет);
- обновить поле в документе (например, изменить статус заказа на «Доставлен»);
- реализовать связь «один-ко-многим» с помощью ссылок (DBRef или просто ObjectId) вместо вложения. Сравнить два подхода.

Тема 3. Векторные базы данных и поиск по сходству

Задача 1.

Работа с векторной БД через API:

- подключиться к векторной БД;
- создать коллекцию/индекс для хранения векторов;
- добавить несколько векторов в коллекцию;
- выполнить запрос на поиск ближайших соседей (k-NN) для тестового вектора.

Тема 4. Оптимизация производительности и интеграция с ИИ

Профилирование и оптимизация «тяжелого» запроса + ETL-пайплайн.

Задача 1. Профилирование

Получить от преподавателя или найти самостоятельно «медленный» SQL-запрос (например, выборка с несколькими JOIN без индексов на большой таблице).

Использовать команду EXPLAIN ANALYZE для анализа плана выполнения.

Применить методы оптимизации: Создание индексов по полям в WHERE и JOIN.

Переписать запрос с использованием оконных функций или CTE для устранения подзапросов.

Проверить скорость выполнения до и после оптимизации.

Задача 2. Пайплайн на Python

Написать скрипт, который:

- извлекает данные из реляционной БД с помощью SQLAlchemy/psycopg2.
- проводит предобработку данных в Pandas (очистка, заполнение пропусков, кодирование категориальных признаков).
- разделяет данные на обучающую и тестовую выборки.
- обучает простую модель машинного обучения (например, логистическую регрессию или случайный лес) с помощью scikit-learn и сохраняет модель на диск.

5. Оценочные материалы для проведения промежуточной аттестации

Промежуточная аттестация по дисциплине проводится в форме экзамена.

Экзамен представляет собой выполнение обучающимся заданий билета, включающего в себя:

Задание №1 – теоретический вопрос на знание базовых понятий предметной области дисциплины, а также позволяющий оценить степень владения обучающегося принципами предметной области дисциплины, понимание их особенностей и взаимосвязи между ними.

Задание №2 – задание на анализ ситуации из предметной области дисциплины и выявление способности обучающегося выбирать и применять соответствующие принципы и методы решения практических проблем, близких к профессиональной деятельности.

Задания №3 – задания на проверку умений и навыков, полученных в результате освоения дисциплины.

Выполнение обучающимся заданий билета оценивается по следующей балльной шкале:

Задание 1: 0-30 баллов

Задание 2: 0-30 баллов

Задание 3: 0-40 баллов

-90 и более (отлично) – ответ правильный, логически выстроен, приведены необходимые выкладки, использована профессиональная лексика. Задания решены правильно. Обучающийся правильно интерпретирует полученный результат.

-70 и более (хорошо) – ответ в целом правильный, логически выстроен, приведены необходимые выкладки, использована профессиональная лексика. Ход решения задания правильный, ответ неверный. Обучающийся в целом правильно интерпретирует полученный результат.

-50 и более (удовлетворительно)– ответ в основном правильный, логически выстроен, приведены не все необходимые выкладки, использована профессиональная лексика. Задания решены частично.

-менее 50 (неудовлетворительно)– ответы на теоретическую часть неправильные или неполные. Задания не решены.

Типовые задания для проведения промежуточной аттестации обучающихся

Задания 1 типа

1. Эволюция систем управления базами данных: от иерархических и сетевых моделей к реляционным и NoSQL. Ключевые этапы развития.
2. Роль данных в жизненном цикле систем искусственного интеллекта (Data-Centric AI). Почему качество и доступность данных важнее сложности модели?
3. Основные принципы реляционной модели данных: отношения, атрибуты, кортежи. Отличия от других моделей.
4. Классификация команд языка SQL: DDL, DML, DCL. Приведите примеры операторов для каждой группы (CREATE, SELECT, GRANT).
5. Типы данных в SQL: числовые, строковые, дата/время, логические. Почему правильный выбор типа данных важен для производительности?
6. Нормализация базы данных. Назовите нормальные формы (1НФ, 2НФ, 3НФ) и цели их применения. Что такое денормализация и когда она оправдана?
7. Проектирование схемы БД по реальному датасету (например, с Kaggle). Процесс: от анализа сущностей до построения ER-диаграммы и написания DDL-скриптов.
8. Операция соединения таблиц (JOIN) в SQL. Различия между INNER JOIN, LEFT JOIN, RIGHT JOIN и FULL OUTER JOIN на примерах.
9. Агрегация данных в SQL: функции COUNT, SUM, AVG, MIN, MAX. Использование оператора GROUP BY и фильтрации HAVING.
10. Концепция NoSQL: причины возникновения и отличия от традиционных реляционных СУБД. Принципы модели BASE в сравнении с ACID.
11. Классификация NoSQL СУБД: Key-Value, Document, Column-family, Graph. Приведите примеры популярных решений для каждого типа (Redis, MongoDB, Cassandra, Neo4j).
12. Сценарии использования NoSQL в задачах Big Data и ИИ: работа с неструктурированными данными, высокая скорость записи, масштабируемость.
13. Особенности документной модели данных на примере MongoDB. Понятие BSON-документа, коллекции и динамической схемы.
14. Моделирование связей «один-ко-многим» в MongoDB: два подхода (вложение документов и использование ссылок). Преимущества и недостатки каждого метода.
15. Агрегационный фреймворк в MongoDB как аналог SQL-запросов с GROUP BY и JOIN. Основные этапы конвейера агрегации (\$match, \$group, \$project).
16. Проблема «проклятия размерности» при поиске векторов эмбедингов. Почему традиционные индексы (B-Tree) неэффективны для многомерных пространств?
17. Метрики расстояния в векторном поиске: косинусное сходство, евклидово расстояние, манхэттенское расстояние. В каких случаях какая метрика предпочтительна?
18. Архитектура векторных баз данных (Pinecone, Weaviate). Основные компоненты: хранение векторов, индекс ANN, фильтрация по метаданным.
19. Индексы Approximate Nearest Neighbor (ANN). Алгоритмы HNSW и IVF: принцип работы и баланс между скоростью поиска и точностью.
20. Пайплайн реализации семантического поиска: генерация эмбедингов текста/изображений с помощью предобученных моделей -> загрузка в векторную БД -> выполнение поискового запроса по смыслу.

Задания 2 типа

1. Вы — аналитик в e-commerce. Нужно спроектировать БД для хранения истории покупок и просмотров товаров (миллиарды записей). Объем данных растет на 100 ГБ в день. Какую архитектуру хранения данных вы выберете и почему?
2. В системе онлайн-кинотеатра нужно реализовать поиск фильмов по описанию: «похожие на "Начало" с Леонардо ДиКаприо». Пользователи вводят текст на естественном языке. Какой тип базы данных и подход к поиску вы примените?
3. У вас есть реляционная БД с таблицей `orders` (100 млн строк). Запрос `SELECT \* FROM orders WHERE user\_id = ? AND status = 'NEW'` выполняется 5 секунд. Опишите пошаговый алгоритм оптимизации этого запроса.
4. Стартап запускает мобильное приложение. На старте данных мало, но ожидается взрывной рост. Требуется хранить профили пользователей (JSON-структура) и их связи. Какую СУБД вы выберете для MVP и как будете масштабировать решение в будущем?
5. В базе данных интернет-магазина нужно хранить каталог товаров (иерархия категорий, характеристики) и отзывы. Как вы спроектируете схему БД, чтобы эффективно выполнять поиск по характеристикам и быстро получать отзывы к товару?
6. Вам нужно построить пайплайн для обучения модели скоринга кредитного риска. Данные лежат в PostgreSQL, а обучающая выборка должна быть подготовлена в Pandas. Опишите последовательность действий и SQL-запросы для извлечения данных.
7. В системе социальной сети нужно реализовать функционал «рекомендации друзей» и «пути знакомства». Какая СУБД лучше всего подходит для этой задачи и почему? Опишите модель данных.
8. Векторный поиск по описаниям товаров начал работать медленно при росте базы до 10 млн векторов. Какие методы оптимизации векторных БД вы примените, не жертвуя сильно качеством поиска?
9. Банковская система требует строгой консистентности данных при переводе средств между счетами. Почему для этой задачи реляционная СУБД с поддержкой ACID-транзакций предпочтительнее NoSQL-решений?
10. Вы работаете с логами веб-сервера (терабайты неструктурированных данных). Нужно быстро находить события по IP-адресу и строить отчеты по кодам ответа HTTP. Какой тип NoSQL-хранилища оптимален для этой задачи?
11. В базе данных есть таблица `users` и `sessions`. Нужно найти пользователей, которые были активны более 30 дней назад, но имеют неоплаченные подписки. Напишите оптимальный SQL-запрос с использованием JOIN.
12. Для системы распознавания лиц нужно хранить эмбединги (векторы) миллионов пользователей и выполнять поиск совпадений за миллисекунды. Объясните, почему обычная реляционная БД не справится, и как работает векторный индекс.
13. Вам нужно перенести данные о связях «сотрудник — руководитель» из Excel-таблицы в графовую СУБД (Neo4j). Опишите модель данных (узлы и связи) и напишите Cypher-запрос для импорта.
14. В MongoDB хранится коллекция товаров с вложенными отзывами. Нужно найти товары с рейтингом выше 4.5, у которых есть отзыв от пользователя "Ivan_77". Напишите агрегационный запрос.
15. Вы проектируете архитектуру ИИ-ассистента поддержки. Ему нужен мгновенный доступ к базе знаний (FAQ) и истории диалогов с клиентом. Предложите комбинацию СУБД для хранения этих данных.
16. Запрос на подсчет уникальных пользователей за месяц выполняется недопустимо долго из-за полного сканирования таблицы логов объемом 5 ТБ. Как изменить структуру хранения или запрос, чтобы ускорить расчет?
17. В системе рекомендаций статей нужно учитывать не только текстовое сходство, но и граф связей «автор — соавтор». Как объединить возможности векторного поиска и графовой СУБД в одном решении?
18. При росте нагрузки на основной сервер БД время ответа увеличилось вдвое. У

вас есть бюджет только на одну меру: купить SSD или настроить репликацию для чтения? Что вы выберете и в каком случае?

19. Вам нужно реализовать полнотекстовый поиск по статьям в блоге с учетом морфологии русского языка внутри SQL-запроса. Какие инструменты/подходы в PostgreSQL вы используете?

20. В MongoDB данные о заказах хранятся как массив внутри документа пользователя. При росте количества заказов (более 1000 на пользователя) операции обновления документа начали блокировать друг друга. Как изменить модель данных, чтобы решить эту проблему?

Задания 3 типа

Задание 1. Спроектировать и реализовать базу данных для интернет-магазина. Включает анализ предметной области, создание ER-диаграммы, нормализацию до 3НФ, написание DDL-скриптов для создания таблиц и их наполнение тестовыми данными.

Задание 2. Написать аналитический SQL-запрос для получения топ-5 категорий по выручке (с использованием JOIN, GROUP BY и оконных функций). Проанализировать его план выполнения через EXPLAIN ANALYZE и оптимизировать, создав индексы и/или переписав запрос.

Задание 3. Перенести данные о пользователях и их заказах из реляционной модели в MongoDB, используя подход «вложения» документов. Написать скрипт на Python (PyMongo) для миграции данных и агрегационный запрос для поиска активных пользователей.

Задание 4. Создать систему семантического поиска по товарам. Требуется сгенерировать текстовые эмбединги через Sentence-Transformers, загрузить их в векторную БД (например, ChromaDB) и реализовать функцию поиска похожих товаров по смысловому запросу.

Задание 5. Разработать скрипт для извлечения данных из реляционной БД в Pandas DataFrame, провести их предобработку (очистка, кодирование признаков), разделить на обучающую и тестовую выборки и сохранить итоговый датасет в файл формата Parquet.

Задание 6. Спроектировать архитектуру хранения данных для рекомендательной системы стримингового сервиса. Необходимо выбрать типы СУБД (реляционная, NoSQL, векторная) для разных компонентов (профили, каталог, история просмотров) и обосновать свой выбор.

Задание 7. Спроектировать модель данных в Neo4j для социальной сети (узлы User, связи FRIEND). Написать скрипт на Python для создания графа и Cypher-запрос для поиска «друзей друзей» заданного пользователя.

Задание 8. Предложить решение проблемы производительности при росте одной большой таблицы. Описать стратегию шардирования (например, по user_id), спроектировать распределение данных по серверам и привести пример конфигурации.

Задание 9. Создать гибридную систему, где основная информация о товарах хранится в PostgreSQL, а динамические характеристики и отзывы — в MongoDB. Написать скрипт на Python, который делает запросы к обеим базам и объединяет результаты в один отчет.

Задание 10.

Дан текст «тяжелого» SQL-запроса. Студент должен выполнить команду EXPLAIN ANALYZE, проанализировать результат, найти причину медленной работы (например, полное сканирование таблицы) и предложить конкретные шаги по оптимизации.