



Автономная некоммерческая профессиональная образовательная организация
«МЕЖДУНАРОДНЫЙ ВОСТОЧНО-ЕВРОПЕЙСКИЙ КОЛЛЕДЖ»
Пушкинская ул., д. 268, 426008, г. Ижевск. Тел.: (3412) 77-68-24. E-mail: mveu@mveu.ru, www.mveu.ru
ИНН 1831200089. ОГРН 1201800020641

20.02.2026 г.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

**по организации и методическому сопровождению
самостоятельной работы студентов**

при изучении учебной дисциплины

«ОП.07 ОСНОВЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ»

по специальности

**09.02.13 Интеграция решений с применением технологий искусственного
интеллекта**

1. ОБЩИЕ ПОЛОЖЕНИЯ ПО ПЛАНИРОВАНИЮ И ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

1.1. Методические рекомендации по организации и методическому сопровождению самостоятельной работы обучающихся СПО разработаны согласно Федеральному закону Российской Федерации от 29 декабря 2012 г. N 273-ФЗ "Об образовании в Российской Федерации"; Федеральному государственному образовательному стандарту среднего профессионального образования (по специальности); Приказу Минпросвещения России от 24.08.2022 N 762 "Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам среднего профессионального образования", Положения об организации самостоятельной работы студентов, Методических рекомендаций по организации и методическому сопровождению самостоятельной работы студентов СПО.

1.2. Обоснование расчета времени, затрачиваемого на выполнение внеаудиторной самостоятельной работы обучающимися:

Преподаватель эмпирически определяет затраты времени на самостоятельное выполнение конкретного содержания учебного задания: на основании наблюдений за выполнением обучающимися аудиторной работы, опроса обучающихся о затратах времени на то или иное задание, хронометража собственных затрат на решение той или иной задачи из расчета уровня знаний и умений студентов. По совокупности затрачиваемых усилий и в зависимости от трудоемкости выполняемых заданий, определяется количество часов на выполнение каждого задания по самостоятельной работе. По совокупности заданий определяется объем времени на внеаудиторную самостоятельную работу по каждой теме и в целом по учебной дисциплине.

2. ВИДЫ И ФОРМЫ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ.

2.1. Учебной дисциплиной предусмотрен следующий объем самостоятельной работы обучающихся:

Вид самостоятельной работы студентов	Объем часов
Внеаудиторная самостоятельная работа	22

2.2. Формы самостоятельной работы, виды заданий по учебным темам:

Тема самостоятельной работы № 1. Сравнительный анализ моделей данных и сфер их применения, объем часов 2

Цель самостоятельной работы: Систематизация теоретических знаний о моделях данных, формирование навыка анализа применимости различных типов СУБД (реляционных и нереляционных) для конкретных задач, в том числе для систем искусственного интеллекта.

Задания для самостоятельной работы (на выбор или комплексно):

Задание 1. Составление аналитической таблицы «Эволюция моделей данных».

Проведите анализ литературы и заполните таблицу, сравнив основные модели организации данных.

Критерий сравнения	Иерархическая модель	Сетевая модель	Реляционная модель (SQL)	Постреляционная / NoSQL (для ИИ)
Принцип связи	«Дерево» (Родитель-Потомок)	«Граф» (Многие ко многим)	Таблицы и ключи	Документы, Графы, Ключ-Значение
Преимущества	Быстрый доступ по структуре	Гибкость связей	Целостность, стандартизация (SQL)	Масштабируемость, работа с Big Data
Недостатки	Сложность изменения структуры	Сложность разработки	Медленная работа на сверхбольших объемах	Слабая транзакционная целостность
Пример СУБД	IMS	IDMS	MySQL, PostgreSQL	MongoDB, Cassandra, Redis

Задание 2. Подготовка реферата / доклада (с презентацией).

Выберите одну из тем для детальной проработки. Текст должен соответствовать нормам деловой коммуникации и нормам русского языка.

- **Темы рефератов:**
 1. **История развития баз данных:** от файловых систем к современным облачным хранилищам.

2. **12 правил Эдгара Кодда:** фундамент реляционной теории и их актуальность сегодня.
3. **Векторные базы данных (Vector Databases):** роль в развитии генеративного искусственного интеллекта и LLM.
4. **Правовые и этические нормы сбора данных:** защита персональных данных (ФЗ-152) и ответственность разработчика БД.
5. **Сравнение файловой системы и СУБД:** почему Excel не является базой данных?

Задание 3. Кейс-стади «Проектирование "на бумаге"».

Опишите (в виде эссе на 1-2 страницы), какую модель данных (реляционную или NoSQL) и почему вы бы выбрали для следующих проектов:

- Система учета кадров государственного предприятия (строгая отчетность).
- Социальная сеть с миллионами пользователей и связей (графы, лайки).
- Хранилище датасетов (изображений и видео) для обучения нейросети.

Методические рекомендации по выполнению работы:

1. Требования к оформлению:

- Работа выполняется в текстовом редакторе (MS Word, LibreOffice).
- Шрифт Times New Roman, 14 пт, интервал 1.5.
- Обязательно наличие списка используемых источников (минимум 3 источника не старше 5 лет).
- Для Задания 1: Таблица должна быть читаемой, допускается альбомная ориентация страницы.

2. Критерии оценки:

- **Глубина проработки:** насколько полно раскрыта тема, приведены ли конкретные примеры СУБД.
- **Качество анализа:** обоснованность выводов в сравнительной таблице или кейс-стади.
- **Грамотность и стиль:** отсутствие орфографических ошибок, использование профессиональной терминологии (сущность, атрибут, кортеж, транзакция).
- **Этический аспект:** при выборе тем, связанных с безопасностью, отражено понимание ответственности за сохранность данных.

Тема самостоятельной работы № 2. Проектирование логической структуры базы данных и выбор инструментальных средств, объем часов 4

Цель самостоятельной работы: Закрепление навыков концептуального проектирования, углубленное изучение нотаций моделирования данных и анализ компромиссов между нормализацией и производительностью систем.

Задания для самостоятельной работы:

Задание 1. Кейс-стади «Умный город: Управление отходами».

Спроектируйте фрагмент базы данных для системы «Smart Waste».

- *Контекст:* В городе устанавливаются «умные» мусорные контейнеры с датчиками наполняемости. Мусоровозы должны выезжать только к переполненным бакам, чтобы экономить топливо и снижать выбросы CO₂.
- *Задача:* Нарисуйте (в тетради или редакторе) ER-схему, содержащую сущности:
 1. **Контейнер** (Геолокация, Тип отходов, Текущий % заполнения).
 2. **Мусоровоз** (Номер, Вместимость, Текущие GPS-координаты).
 3. **Маршрутный лист** (Список точек, которые нужно посетить водителю сегодня).
- *Требование:* Обеспечьте связь, позволяющую ИИ-алгоритму понять, какой мусоровоз забрал мусор из конкретного контейнера и во сколько.

Задание 2. Аналитическая таблица «Нотации моделирования».

Существует несколько стандартов рисования схем. Проведите сравнительный анализ двух самых популярных нотаций, заполнив таблицу. Это поможет вам «читать» чужие схемы в будущем.

Критерий	Нотация Питера Чена (Chen notation)	Нотация «Воронья лапка» (Crow's Foot / IE)
Как изображается сущность?	(Прямоугольник)	(Прямоугольник)
Как изображается атрибут?	(Овал, присоединенный к сущности)	(Список внутри прямоугольника)

Критерий	Нотация Питера Чена (Chen notation)	Нотация «Воронья лапка» (Crow's Foot / IE)
Как изображается связь «Многие»?	<i>(Буква N или M на линии)</i>	<i>(Разветвление линии - «лапка»)</i>
Сфера применения	<i>Концептуальное проектирование, обсуждение с заказчиком (наглядно)</i>	<i>Логическое и физическое проектирование, генерация SQL-кода</i>

Задание 3. Мини-исследование «Нормализация против Денормализации в задачах ИИ».

Подготовьте краткий письменный ответ (1 страница) на вопрос:

- *Проблема:* Мы изучили, что базы данных нужно нормализовать (убирать дубли). Однако для обучения нейросетей (Data Science) часто используют **денормализованные** витрины данных (одну гигантскую таблицу).
- *Вопрос:* Почему для OLAP-систем и обучения ИИ правила нормализации часто нарушают? В чем плюсы «плоских» таблиц для скорости чтения данных?

Методические рекомендации по выполнению работы:

1. **К Заданию 1:** Помните про связь «Многие ко многим». Один маршрут включает много контейнеров, но и один контейнер за месяц попадает во много разных маршрутов. Нужна ли промежуточная таблица?
2. **К Заданию 2:** Используйте интернет-источники. Обратите внимание, что нотация Чена занимает больше места на листе, но понятнее для новичков. Нотация Crow's Foot (которую мы использовали на практике) компактнее.
3. **К Заданию 3:** Используйте ключевые слова для поиска: «*OLTP vs OLAP*», «*Star schema*» (схема Звезды), «*Denormalization for Machine Learning*».

Форма отчетности:

Файл документа (Word/PDF), содержащий:

1. Скриншот или фото схемы «Умный город».
2. Заполненную таблицу сравнения нотаций.
3. Текст ответа на вопрос про денормализацию.

Тема самостоятельной работы № 3 Стратегии обеспечения целостности данных и оптимизация логической модели, объем часов __4__

Цель самостоятельной работы: Анализ современных подходов к идентификации записей в распределенных системах, изучение методов «мягкого удаления» (Soft Delete) и работа с технической документацией по ограничениям целостности.

Задания для самостоятельной работы:

Задание 1. Исследовательская задача «UUID против INT». В практических работах мы использовали целочисленный идентификатор (ID = 1, 2, 3...). Однако в системах Big Data и микросервисах часто используют **UUID** (Universally Unique Identifier) — длинные строки вида 550e8400-e29b-41d4-a716-446655440000.

Задание: Составьте сравнительную таблицу (3-4 пункта) для этих двух типов ключей.

- Какой тип занимает меньше места на диске?
- Какой тип позволяет генерировать ключи на разных серверах без конфликтов (важно для распределенных ИИ)?
- Какой тип быстрее ищется базой данных (индексируется)?
- *Вывод:* Какой ключ вы выберете для таблицы с миллиардом записей?

Задание 2. Кейс-стади «Проблема забывчивого пользователя» (Soft Delete). Изучите концепцию **Soft Delete** («Мягкое удаление»).

- *Ситуация:* Пользователь вашего приложения нажал кнопку «Удалить аккаунт». По закону вы должны удалить персональные данные. Но этот пользователь сгенерировал тысячи полезных для обучения ИИ меток (тегов) для изображений. Если вы удалите его User_ID каскадно (CASCADE), вы потеряете ценный датасет.
- *Задача:* Опишите решение проблемы. Как изменить структуру таблицы Users (какое поле добавить), чтобы пользователь "исчез" для входа в систему, но его данные остались доступными для администратора и нейросети? Напишите название этого поля и его тип данных.

Методические рекомендации по выполнению работы:

1. **По Заданию 1:** Ищите информацию по запросам «*UUID vs Auto Increment performance*», «*Distributed systems primary keys*». Обратите внимание на то, что INT (4 байта) намного меньше UUID (16 байт или 36 символов в виде строки), что влияет на скорость обучения модели.
2. **По Заданию 2:** Концепция *Soft Delete* подразумевает введение логического флага. Обычно это поле называют is_deleted (BOOLEAN)

или `deleted_at` (DATETIME). В отчете объясните, как это помогает сохранить ссылочную целостность (Foreign Keys не ломаются, так как запись физически остается в таблице).

Форма отчетности: Файл документа (Word/PDF), включающий:

1. Сравнительную таблицу UUID/INT с выводом.
2. Описание схемы решения Soft Delete (название добавленного поля и логика работы).

Тема самостоятельной работы № 4 Администрирование и обслуживание физической базы данных, объем часов __4__

Цель самостоятельной работы: Изучение аспектов эксплуатации базы данных: выбор стратегии резервного копирования (Backup), работа с полуструктурированными данными (JSON) внутри SQL-систем и сравнение транзакционных (OLTP) и аналитических (OLAP) подходов к физическому хранению.

Задания для самостоятельной работы:

Задание 1. Разработка стратегии резервного копирования (Disaster Recovery Plan).

Для системы экологического мониторинга «EcoGuard AI» (из прошлых работ) разработайте регламент сохранения данных, чтобы не потерять историю наблюдений при сбое сервера (реализация У7).

Составьте таблицу-план:

1. **Тип данных:** (Например: «Справочник датчиков» vs «Показания за 10 лет»).
2. **Частота копирования:** (Раз в месяц? Раз в час? В реальном времени?).
3. **Тип бэкапа:** Полный (Full), Дифференциальный (Diff) или Инкрементальный (Log)?
4. Место хранения: (Тот же сервер / Облако / Ленточный накопитель).

Обоснуйте: Почему для справочников не нужно делать бэкап каждый час?

Задание 2. Исследование: «JSON внутри SQL».

Современные СУБД (PostgreSQL, MySQL) умеют хранить не только жесткие таблицы, но и гибкие документы JSON. Это важно для ИИ, когда датчики присылают разные наборы параметров.

Задача: Найдите информацию и напишите краткий конспект (1 стр.):

- Какой тип данных используется для хранения JSON (например, JSONB в Postgres)?
- Можно ли создать **индекс** по отдельному ключу внутри JSON-документа?
- В чем минус хранения всех данных в JSON по сравнению с обычными столбцами (размер, скорость)?

Задание 3. Сравнительный анализ OLTP vs OLAP.

Для обучения ИИ и работы обычного сайта используются разные физические принципы организации БД. Заполните сравнительную таблицу.

Критерий	OLTP (Online Transaction Processing)	OLAP (Online Analytical Processing)
Пример использования	Покупка в интернет-магазине, запись лога	Отчет о продажах за год, обучение нейросети
Характер запросов	Много мелких, быстрых, простых	Мало, но очень тяжелые и сложные
Ориентация хранения	Строчная (Row-store)	Колоночная (Column-store)
Приоритет	Скорость записи и целостность	Скорость чтения больших массивов

Методические рекомендации по выполнению работы:

1. К Заданию 1 (Бэкапы):

- *Полный бэкап:* Копирует всё. Долго, занимает много места.
- *Инкрементальный:* Копирует только то, что изменилось с последнего раза. Быстро.
- *Совет:* Для системы мониторинга обычно делают Полный бэкап раз в неделю, а Инкрементальный — каждые 6-12 часов.

2. К Заданию 2 (JSON):

- Ищите по запросам: «*PostgreSQL JSONB performance*», «*Indexing JSON data in SQL*».
- Это задание готовит вас к реальности, где "Чистых" реляционных баз почти не осталось — везде гибридные подходы.

3. К заданию 3 (OLTP/OLAP):

- Это фундаментальное различие. Для специальности 09.02.13 важно понимать, что обучающую выборку для ИИ лучше формировать на OLAP-архитектуре (ClickHouse, Data Warehouse), а управлять пользователями — на OLTP (PostgreSQL).

Форма отчетности:

Единый файл документа (Word/PDF), содержащий:

1. Таблицу «Регламент резервного копирования» с обоснованием.
2. Конспект-ответы на вопросы по JSON.
3. Заполненную сравнительную таблицу OLTP vs OLAP.

Тема самостоятельной работы № 5 Формирование и очистка наборов данных (Dataset) с использованием команд SQL, объем часов __4__

Цель самостоятельной работы: Закрепление навыков манипулирования данными (DML), освоение методов предварительной обработки данных («чистки») перед машинным обучением: удаление дублей, исправление аномалий и заполнение пропусков средствами языка SQL.

Задания для самостоятельной работы:

Контекст задачи (Data Engineering): Вы готовите данные для обучения нейросети, которая распознает животных на фото. Данные загружены в таблицу `Raw_Images` («Сырые изображения»), но они «грязные»: есть пустые поля, ошибочные метки и битые ссылки. Ваша задача — привести таблицу в порядок.

Задание 1. Подготовка схемы и данных (DDL + INSERT). Напишите скрипт, который:

1. Создаст таблицу `Raw_Images` с полями:
 - `img_id` (INT, PK).
 - `file_path` (VARCHAR, путь к файлу).

- `category` (VARCHAR, класс: 'cat', 'dog', 'car'...).
 - `resolution_x` (INT, ширина).
 - `resolution_y` (INT, высота).
 - `is_broken` (BOOLEAN, флаг «битый файл», по умолчанию FALSE).
2. Вставит 5-6 тестовых строк, среди которых **обязательно** должны быть:
- Строка с пустым (NULL) значением в поле `category`.
 - Строка с отрицательным разрешением (`resolution_x = -100` — явная ошибка).
 - Строка с нормальными данными.

Задание 2. Логическая чистка данных (UPDATE). Напишите запросы для исправления ошибок:

1. **Исправление аномалий:** Если разрешение (`resolution_x` или `resolution_y`) меньше нуля, установите флаг `is_broken = TRUE`.
2. **Заполнение пропусков (Imputation):** Если категория не указана (NULL), установите значение 'unknown'.

Задание 3. Фильтрация и удаление (DELETE / SELECT).

1. **Удаление мусора:** Удалите из таблицы все записи, помеченные как битые (`is_broken = TRUE`).
2. **Формирование выборки:** Напишите запрос `SELECT`, который выгрузит только те картинки, которые подходят для обучения (например: категория не 'unknown' И разрешение больше 64x64 пикселей).

Задание 4. Исследовательский вопрос «TRUNCATE vs DELETE» (У9). В документации к SQL есть две команды для удаления всех данных: `DELETE FROM table` и `TRUNCATE TABLE table`. *Задание:* Найдите информацию и заполните таблицу отличий (3 пункта).

- Какую команду можно отменить (Rollback)?
- Какая команда сбрасывает счетчик Auto Increment (id) обратно до 1?
- Какая команда работает быстрее на миллионе строк?

Методические рекомендации по выполнению работы:

1. **Синтаксис NULL:** Помните, что в SQL нельзя писать `WHERE category = NULL`. Правильный синтаксис: `WHERE category IS NULL` (или `IS NOT NULL`).
2. **Значимость для ИИ:** Качество данных важнее, чем качество алгоритма. Если вы подадите на вход нейросети картинки с разрешением "-100", она не обучится. Умение чистить данные через `UPDATE ... WHERE` — базовый навык Data Engineer.

Форма отчетности: Файл (Word/PDF), содержащий:

1. SQL-код создания и наполнения таблицы.
2. SQL-код запросов «чистки» (`UPDATE`, `DELETE`).
3. Скриншот таблицы после всех манипуляций (должна остаться чистая, корректная база).
4. Заполненная таблица сравнения `DELETE` и `TRUNCATE`.

Тема самостоятельной работы № 6 Продвинутые методы оптимизации: типы индексов и стратегии кэширования, объем часов __4__

Цель самостоятельной работы: Углубленное изучение структур данных для индексации (B-Tree, Hash, GIN/GiST), понимание принципов работы кэширования (Redis/Memcached) и анализ проблемы «N+1 запросов» как одной из самых частых причин низкой производительности систем.

Задания для самостоятельной работы:

Задание 1. Сравнительный анализ типов индексов (Research).

На практических занятиях мы использовали стандартный B-Tree индекс. Но для ИИ и поиска по документам существуют другие типы.

Задание: Заполните таблицу применимости индексов. Найдите информацию, для каких задач лучше подходят следующие структуры:

Тип индекса	Принцип работы (кратко)	Для каких данных идеален?	Пример из жизни
B-Tree	Сбалансированное дерево	Числа, даты, сортировка	Поиск по ID или дате

Тип индекса	Принцип работы (кратко)	Для каких данных идеален?	Пример из жизни
Hash	Хеш-таблица (Ключ-значение)	?	?
GIN / GiST	Обратный индекс	Тексты, массивы, JSON	Поиск слова внутри статьи
Spatial (R-Tree)	Пространственное дерево	Геометрия	Поиск объектов на карте

Задание 2. Архитектурный паттерн «Кэширование» (Caching).

База данных — самое узкое место системы. Чтобы снизить нагрузку, используют кэш (временную память).

Сценарий: На главной странице системы «EcoGuard» висит виджет: «Всего пожаров за 10 лет: 15 432». Этот запрос сканирует миллионы строк.

Задача: Опишите алгоритм работы с кэшем (Redis) для этого виджета:

1. Пользователь заходит на сайт. Куда сначала обращается система: в БД или в Кэш?
2. Что происходит, если в кэше есть данные (Cache Hit)?
3. Что происходит, если в кэше нет данных (Cache Miss)?
4. Как долго хранить это число в кэше (TTL — Time To Live), чтобы данные не устарели слишком сильно?

Задание 3. Проблема «N+1» (The N+1 Query Problem).

Это классическая ошибка начинающих разработчиков, использующих ORM (связку кода и БД).

Пример кода (псевдокод):

```
Python
groups = get_all_groups()           # 1 запрос: SELECT * FROM Groups
for group in groups:
    print(group.get_commander())    # N запросов: SELECT * FROM
Commanders WHERE group_id = ?
```

Задание:

1. Объясните, почему этот код «убьет» базу данных, если групп будет 1000? (Сколько всего запросов будет выполнено?).
2. Как переписать этот алгоритм, чтобы получить все данные за 2 запроса или за 1 сложный запрос?

Задание 4. Экологический аспект оптимизации (Green IT).

Напишите небольшое эссе (0.5 страницы): «Как оптимизация SQL-запросов влияет на углеродный след дата-центров».

Опорные вопросы:

- Как связаны нагрузка на процессор (CPU) и потребление электричества?
- Почему «медленный запрос» = «грязный запрос»?

Методические рекомендации по выполнению работы:

1. К Заданию 1 (Индексы):
Ищите информацию в документации PostgreSQL (раздел Index Types). Это самая продвинутая СУБД в плане типов индексов. GIN индексы критически важны для специальности 09.02.13, так как часто используются для поиска векторов в ИИ.
2. К Заданию 2 (Кэш):
Запомните правило: «Самый быстрый запрос тот, который не дошел до базы данных». Кэширование позволяет обслуживать тысячи пользователей, не нагружая диск сервера.
3. К Заданию 3 (N+1):
Эта проблема актуальна для любого языка программирования (Python, Java, C#). Решение обычно кроется в "Жадной загрузке" (Eager Loading) или использовании JOIN.

Форма отчетности:

Файл (Word/PDF), содержащий:

1. Заполненную таблицу индексов.
2. Блок-схему или текстовое описание алгоритма кэширования.
3. Разбор решения проблемы N+1 (расчет количества запросов "До" и "После").
4. Эссе на тему Green IT.

3. ОРГАНИЗАЦИЯ КОНТРОЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Результаты самостоятельной работы

Оценки за выполнение заданий могут выставляться по пятибалльной системе или в форме зачета и учитываться как показатели текущей успеваемости обучающихся.

Качественная оценка индивидуальных образовательных достижений		Критерии оценки результата
балл (оценка)	вербальный аналог	
5	отлично	Представленные работы высокого качества, уровень выполнения отвечает всем требованиям, теоретическое содержание освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, выполнены все предусмотренные программой обучения задания. Студент демонстрирует глубокое понимание взаимосвязей между объектами базы данных, умение аргументированно обосновывать выбор технических решений (типов данных, индексов), творческий подход к решению задач, а также высокую культуру оформления отчета и соблюдение сроков сдачи.
4	хорошо	Уровень выполнения работы отвечает всем требованиям, теоретическое содержание освоено полностью без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения задания выполнены, некоторые из выполненных заданий, возможно, содержат ошибки. Студент демонстрирует уверенное владение материалом, но допускает незначительные неточности в формулировках или оформлении SQL-кода, способен решить задачу, но без глубокого анализа альтернативных вариантов.
3	удовлетворительно	Уровень выполнения работы отвечает большинству основных требований, теоретическое содержание освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения заданий выполнено, некоторые виды заданий выполнены с ошибками. Студент понимает основную суть задания, но испытывает

		затруднения при детализации, задание выполнено по образцу, самостоятельные выводы отсутствуют или поверхностны.
2	не удовлетворительно	Теоретическое содержание освоено частично, необходимые практические навыки работы не сформированы, большинство предусмотренных программой обучения заданий не выполнено. В работе допущены грубые ошибки, делающие базу данных неработоспособной (нарушение нормализации, синтаксические ошибки), или студент не может ответить на уточняющие вопросы преподавателя по тексту представленной работы.

4. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ОБУЧЕНИЯ.

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

4.1. Основные электронные издания

1. Ткаченко, С. Н. Основы проектирования баз данных : учебник / С. Н. Ткаченко. — Москва : КноРус, 2024. — 176 с. — ISBN 978-5-406-12054-5. — URL: <https://book.ru/book/950600>

4.2. Дополнительные источники

1. Илюшечкин, В. М. Основы использования и проектирования баз данных : учебник для среднего профессионального образования / В. М. Илюшечкин. — Москва : Издательство Юрайт, 2026. — 213 с. — (Профессиональное образование). — ISBN 978-5-534-01283-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/585059>
2. Стружкин, Н. П. Базы данных: проектирование : учебник для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2026. — 477 с. — (Профессиональное образование). — ISBN 978-5-534-11635-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/587735>
3. Советов, Б. Я. Базы данных : учебник для среднего профессионального образования / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 4-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 403 с. — (Профессиональное образование). — ISBN 978-5-534-18784-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/585513>