



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 выполнению практических работ**

**при изучении учебной дисциплины**

**«ОП.07 ОСНОВЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ»**

по специальности

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

Практическая работа – небольшой научный отчет, обобщающий проведенную учащимся работу, которую представляют для защиты преподавателю.

В процессе практического занятия учащиеся выполняют одну или несколько практических работ (заданий) под руководством преподавателя в соответствии с изучаемым содержанием учебного материала.

Состав и содержание практических занятий направлены на реализацию Государственных требований.

Наряду с формированием умений и навыков в процессе практических занятий обобщаются, систематизируются, углубляются и конкретизируются теоретические знания, вырабатывается способность и готовность использовать теоретические знания на практике, развиваются интеллектуальные умения.

Практические занятия проводятся в форме практической подготовки в виде работ, связанных с будущей профессиональной деятельностью.

К практическим работам предъявляется ряд требований, основным из которых является полное, исчерпывающее описание всей проделанной работы, позволяющее судить о полученных результатах, степени выполнения заданий и профессиональной подготовке учащихся.

## **I. Практические работы:**

**Тема практической работы № 1 Анализ структуры базы данных на примере реальной системы, объем часов 2**

У1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

У2. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии.

У4. Эффективно взаимодействовать и работать в коллективе и команде.

У5. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации.

У6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение.

**Цель практической работы** Формирование навыков первичного обследования предметной области: определение границ системы, выявление ключевых бизнес-процессов и пользователей, нуждающихся в информации, без углубления в детальную структуру таблиц

## Задание(я)

**Задание 1. Вербальное описание системы (Интервьюирование).** Представьте, что вы — аналитик данных. Прочитайте краткое ТЗ от заказчика и сформулируйте 5-7 уточняющих вопросов, ответы на которые необходимы для начала проектирования.

- *Сценарий:* «Система контроля доступа на предприятие с распознаванием лиц».
- *Вводные:* Нужна база сотрудников, турникеты с камерами. ИИ должен узнавать лицо и открывать дверь.
- *Задача:* Напишите вопросы заказчику (например: «Нужно ли хранить фото гостей?», «Сколько лет хранится история входов?»).

**Задание 2. Выделение информационных потоков (Input -> Process -> Output).** Для сценария из Задания 1 определите:

1. **Входящая информация:** Что поступает в систему извне? (Например: *Видеопоток с камеры, данные нового сотрудника из отдела кадров*).
2. **Хранимая информация:** Что система должна «помнить» долгое время? (Общими словами: *Список людей, Журнал событий*).
3. **Исходящая информация:** Что система выдает пользователю? (Например: *Отчет об опозданиях, Сигнал тревоги*).

## Методические указания по ходу выполнения работы:

1. На этом этапе **не нужно** писать названия полей (id, name, date). Ваша задача — понять **логику работы** системы.
2. В Задании 1 вопросы должны касаться объема данных, типов пользователей и правил безопасности.
3. В Задании 2 используйте схему «Черного ящика»: мы не знаем, как устроена БД внутри, но знаем, что в неё входит и что выходит.

**Требования к оформлению отчета:** Отчет предоставляется преподавателю в электронном виде (формат .docx или .pdf) или в письменном виде (по согласованию).

## Структура отчета:

1. **Титульный лист** (ФИО студента, номер группы, название работы, дата).
2. **Цель работы** (скопировать из методических указаний).
3. **Ход работы:**
  - **Исходное задание:** Краткое описание выбранного сценария или скриншот задания.
  - **Результаты Задания 1:** Список сформулированных вопросов к заказчику (текстом).

- **Результаты Задания 2:** Структурированное описание потоков данных (списки Входящей, Хранимой и Исходящей информации).
4. **Вывод:** Краткое заключение о проделанной работе.
- *Пример вывода:* «В ходе работы были изучены методы обследования предметной области. На примере системы контроля доступа я научился выделять необходимые информационные потоки и формировать требования к будущей базе данных».

## **Тема практической работы № 2. Построение ER-диаграммы для простой предметной области, объем часов 4**

У1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

У2. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии.

У4. Эффективно взаимодействовать и работать в коллективе и команде.

У5. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации.

У6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение.

**Цель практической работы** Приобретение навыков визуального моделирования данных, освоение нотации ER-диаграмм (Entity-Relationship) и установления связей между сущностями для обеспечения целостности данных.

### **Задание(я):**

**Задание 1. Выбор инструментария.** Выберите программное средство для построения диаграмм (например, *draw.io*, *MS Visio*, *MySQL Workbench* или *DBeaver*) или подготовьтесь к черчению схемы в тетради.

**Задание 2. Построение концептуальной схемы.** Используя результаты Практической работы №1 (выделенные сущности и атрибуты):

1. Отобразите все сущности в виде прямоугольников.
2. Внутри прямоугольников запишите атрибуты, подчеркните **Первичные ключи (PK)**.
3. Соедините сущности линиями связей в соответствии с логикой предметной области.

**Задание 3. Определение кардинальности связей.** Для каждой линии связи проставьте тип отношений:

- Один к одному (1:1);
- Один ко многим (1:M);

- Многие ко многим (M:N).

*Пример для варианта «Система разметки данных» (из ПР №1):*

- Один *Заказчик* может создать много *Наборов данных* (1:M).
- Один *Разметчик* может поставить много *Тегов* (1:M).
- *Изображение* может иметь много *Тегов*, и один *Тег* (как тип) может встречаться на многих изображениях (M:N) — *требует внимания!*

**Методические указания по ходу выполнения работы (комментарии к заданиям, к выполнению заданий):**

### 1. Основные элементы ER-диаграммы:

- **Сущность (Entity):** Объект реального мира. Графически изображается прямоугольником. Имя сущности — существительное в единственном числе (например, User, Image).
- **Атрибут (Attribute):** Свойство объекта.
- **Связь (Relationship):** Ассоциация между сущностями.

### 2. Типы связей (Кардинальность):

- **1:1 (Один к одному):** Одной записи в таблице А соответствует ровно одна запись в таблице Б. *Пример: Гражданин — Паспорт.*
- **1:M (Один ко многим):** Одной записи в таблице А соответствует несколько записей в таблице Б. *Пример: Группа — Студенты.* Это самый распространенный тип связи. Реализуется путем добавления **Внешнего ключа (Foreign Key)** в таблицу со стороны «Многих».
- **M:N (Многие ко многим):** Одной записи в таблице А соответствует несколько записей в таблице Б, и наоборот. *Пример: Студенты — Дисциплины.*
  - *Важно:* В реляционных БД прямая связь M:N невозможна. На логическом уровне она должна быть раскрыта через третью (промежуточную) таблицу связей.

**3. Нотация (Способ рисования):** Рекомендуется использовать нотацию **Crow's Foot** («Воронья лапка»):

- || — строго один.
- |<- — много (лапка).
- 0<- — ноль или много.

**4. Рекомендация:** При построении диаграммы помните, что качество связей напрямую влияет на возможность обучения ИИ. Если вы неправильно свяжете таблицу «Изображения» и таблицу «Метки» (Теги), вы не сможете корректно выгрузить обучающую выборку (X и Y) с помощью SQL-запроса JOIN.

**5. Требования к отчету:** Отчет предоставляется преподавателю в электронном виде (формат .docx или .pdf) или в письменном виде (по согласованию).

В отчете представить скриншот (или фото) разработанной ER-диаграммы. Подписать тип каждой связи. Если обнаружена связь «Многие ко многим», указать, какая новая промежуточная сущность понадобится для ее разрешения.

### **Тема практической работы № 3. Определение сущностей и атрибутов для заданной предметной области, объем часов 4**

У1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

У2. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии.

У4. Эффективно взаимодействовать и работать в коллективе и команде.

У5. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации.

У7. Содействовать сохранению окружающей среды, эффективно действовать в чрезвычайных ситуациях.

**Цель практической работы** Формирование навыков концептуального проектирования: выделение сущностей, детализация атрибутивного состава, определение типов данных и ограничений целостности на примере системы экологического мониторинга.

#### **Задание(я):**

**Контекст задачи:** Разрабатывается база данных для системы «EcoGuard AI» — системы раннего обнаружения лесных пожаров и мониторинга качества воздуха в заповедных зонах. Система собирает данные с IoT-датчиков, а модуль искусственного интеллекта прогнозирует распространение огня.

**Задание 1. Определение Сущностей (Entities).** Из описания контекста выделите 4-5 основных сущностей.

- *Подсказка:* Объектами могут быть физические устройства, зоны ответственности, события (измерения) и ответственные лица.

**Задание 2. Детализация атрибутов и словарь данных.** Для каждой выделенной сущности составьте подробную таблицу атрибутов. Укажите, какие данные являются обязательными.

*Пример оформления для сущности «Датчик»:*

| Имя атрибута   | Описание                                   | Тип данных (примерный) | Обязательно? | Примечание                            |
|----------------|--|------------------------|--------------|---------------------------------------|
| ID_Датчика     | Уникальный идентификатор датчика в системе | Целое число            | Да (PK)      | Первичный ключ (Primary Key)          |
| Гео_Широта     | Географическая координата (GPS)            | Вещественное число     | Да           | Используется для отображения на карте |
| Модель         | Модель устройства (производитель/версия)   | Текст (Строка)         | Нет          | Если не указано, считается «Стандарт» |
| Дата_Установки | Дата монтажа оборудования                  | Дата                   | Да           | Необходима для расчета износа         |

**Задание 3. Работа с доменами и ограничениями.** Для обеспечения достоверности экологических данных определите ограничения для атрибутов сущности «Показание\_Датчика» (Measurement):

1. Температура: не может быть ниже -50 и выше +100 (Степень реалистичности).
2. Уровень CO<sub>2</sub>: не может быть отрицательным.
3. Дата замера: не может быть больше текущей (в будущем).

**Задание 4. Групповая верификация (Работа в парах — У4).** Обменяйтесь разработанными схемами с соседом. Проверьте:

- Есть ли у каждой сущности первичный ключ?
- Все ли атрибуты атомарны (не делимы)?
- Позволяет ли набор атрибутов точно определить место Чрезвычайной Ситуации (координаты, зона, время)?

**Методические указания по ходу выполнения работы (комментарии к заданиям, к выполнению заданий):**

**1. Методика выделения атрибутов:** Атрибут должен быть атомарным. Нельзя создавать поле «Адрес», где записано «г. Москва, ул. Ленина, д. 5». Нужно разбивать на: «Город», «Улица», «Дом». Это критически важно для ИИ-систем, которые будут анализировать данные по регионам.

## 2. Типы ключей:

- **PK (Primary Key)** — Первичный ключ. Уникален для каждой строки (например, ID датчика). Не может быть пустым.
- **Кандидаты в ключи** — Атрибуты, которые тоже могут быть уникальными (например, Серийный номер завода), но вы выбрали другой атрибут (ID) в качестве главного.

**3. Экологический аспект:** При проектировании сущности «Событие» (Event) или «Тревога» (Alert) обязательно предусмотрите атрибуты, важные для спасателей МЧС:

- *Уровень опасности* (низкий, средний, критический).
- *Площадь возгорания* (расчетная).
- *Скорость ветра* (влияет на распространение). Правильно спроектированная БД позволяет спасти жизни и лес, ускоряя реакцию на инциденты.

**4. Требования к отчету:** Отчет предоставляется преподавателю в электронном виде (формат .docx или .pdf) или в письменном виде (по согласованию).

В отчете представить:

1. Список сущностей.
2. Заполненные таблицы атрибутов (Словарь данных).
3. Список ограничений для валидации данных (Задание 3).
4. Вывод о том, как структура данных помогает в мониторинге окружающей среды.

**Тема практической работы № 4. Построение сложной ER-диаграммы с учётом нормализации, объем часов 4**

У1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

У2. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии.

У4. Эффективно взаимодействовать и работать в коллективе и команде.

У5. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации.

У7. Содействовать сохранению окружающей среды, эффективно действовать в чрезвычайных ситуациях.

**Цель практической работы** Освоение методов нормализации отношений базы данных (приведение к 3НФ) и построение логической модели данных со сложными связями («Многие ко многим») для обеспечения целостности и отсутствия избыточности информации.

## Задание(я):

**Контекст задачи (Развитие системы EcoGuard AI):** Система экологического мониторинга расширяется. Помимо сбора данных с датчиков, теперь необходимо автоматизировать работу **Групп быстрого реагирования** (МЧС/Лесхоз), которые выезжают на тушение пожаров и ремонт оборудования.

**Задание 1. Нормализация данных (Декомпозиция).** Вам передан «плоский» отчет (Excel-таблица) о выездах групп, который велся вручную. В нем много дублей. *Исходные данные (ссылка на таблицу: [https://docs.google.com/spreadsheets/d/1y4RtZ\\_Dm6hOULP1LiMu6MlAkfegj1O0xOD\\_H66aiSJw/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1y4RtZ_Dm6hOULP1LiMu6MlAkfegj1O0xOD_H66aiSJw/edit?usp=sharing))*

*Ваша задача:*

1. Проанализируйте зависимости между полями.
2. Разбейте этот список на несколько связанных таблиц так, чтобы данные соответствовали **Третьей нормальной форме (3НФ)**.
  - *Подсказка:* Зависит ли ФИО командира от номера выезда или от группы? Зависит ли ГосНомер от выезда или от ID транспорта?

**Задание 2. Разрешение связи «Многие ко многим» (M:N).** В системе есть сущность «Группа реагирования» и сущность «Инцидент» (пожар/поломка).

- Одна группа может выезжать на множество инцидентов.
- Один крупный инцидент могут тушить несколько групп одновременно.

*Ваша задача:* Реализуйте эту связь на ER-диаграмме. Создайте промежуточную (ассоциативную) сущность (например, Participation / Участие\_в\_выезде), которая свяжет Группы и Инциденты. Добавьте в неё атрибуты: *Время\_прибытия* и *Роль* (Тушение/Эвакуация).

**Задание 3. Построение итоговой комплексной схемы.** В программном средстве (draw.io / Visio / Workbench) соберите полную схему, объединив результаты Работы №3 (Датчики) и текущие таблицы (Группы, Выезды). Схема должна содержать минимум 5-6 связанных сущностей.

**Задание 4. Валидация схемы.** Обсудите полученную схему в микро-группах. Проверьте сценарий Чрезвычайной Ситуации:

- *Запрос:* «Покажи телефон командира группы, которая тушила пожар в зоне №5 вчера».
- Позволяет ли ваша схема «пройти» по связям от Зоны к Командиру? Если нет — исправьте схему.

## Методические указания по ходу выполнения работы (комментарии к заданиям, к выполнению заданий):

### 1. Правила нормализации (кратко):

- **1НФ (Первая форма):** Все атрибуты атомарны (в одной ячейке — одно значение), нет повторяющихся групп полей.
- **2НФ (Вторая форма):** Таблица находится в 1НФ, и все неключевые атрибуты зависят от *всего* первичного ключа (актуально для составных ключей).
- **3НФ (Третья форма):** Таблица находится во 2НФ, и нет *транзитивных зависимостей* (зависимости неключевого атрибута от другого неключевого).
  - *Пример нарушения:* В таблице «Выезд» есть поля ID\_Группы и Командир. Командир зависит от Группы, а не от Выезда. Нужно вынести Группу и Командира в отдельную таблицу.

**2. Реализация связи M:N:** Реляционные базы данных физически не поддерживают связь «Многие ко многим» напрямую. Если вы видите, что *Объект А* связан с множеством *Объектов В* и наоборот, вы **обязаны** создать третью таблицу посередине.

- Было: Группа <---> Инцидент
- Стало: Группа -< Участие >- Инцидент (Таблица Участие содержит ID\_Группы и ID\_Инцидента как внешние ключи).

**3. Требования к отчету:** Отчет предоставляется в формате .docx или .pdf.

1. **Титульный лист.**

2. **Цель работы.**

3. **Ход работы:**

- **Блок «Нормализация»:** Схема преобразования исходного списка в набор таблиц (можно текстом или стрелками). Пример: «*Выделил таблицу Транспорт, чтобы не дублировать ГосНомера в каждом выезде*».
- **Блок «ER-диаграмма»:** Скриншот итоговой сложной схемы.

4. **Вывод:**

- Укажите, какие аномалии (дублирование, сложность обновления) были устранены благодаря нормализации.
- Подтвердите применимость схемы для реагирования на ЧС.

## Тема практической работы № 5. Преобразование ER-диаграммы в таблицы реляционной базы данных, объем часов   4

У1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

У2. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии.

У4. Эффективно взаимодействовать и работать в коллективе и команде.

У5. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации.

У9. Пользоваться профессиональной документацией на государственном и иностранном языках.

**Цель практической работы** Формирование навыков логического проектирования: трансляция концептуальной модели в реляционную схему, определение специфических типов данных СУБД, именование объектов на английском языке (транслитерация/перевод) и описание связей через внешние ключи (Foreign Keys).

### Задание(я):

Контекст задачи:

Вы — технический архитектор проекта «EcoGuard AI». Ваша ER-диаграмма (из ПР №4) утверждена заказчиком. Теперь необходимо составить Техническую спецификацию (Data Dictionary) для программистов, которые будут писать код базы данных.

### Задание 1. Именование объектов (Naming Convention) — реализация.

Базы данных не любят кириллицу. Переведите названия сущностей и атрибутов из вашего прошлого проекта на технический английский язык. Используйте стиль snake\_case (строчные буквы, разделение нижним подчеркиванием).

| Русское<br>(Концептуальное) | название | Английское<br>(Логическое) | название | таблицы |
|-----------------------------|----------|----------------------------|----------|---------|
| Группа реагирования         |          | response_groups            |          |         |
| Инцидент                    |          | incidents                  |          |         |
| Участие (промежуточная)     |          | incident_participation     |          |         |
| Датчик                      |          | sensors                    |          |         |

### Задание 2. Спецификация типов данных.

Для таблицы sensors (Датчики) и incidents (Инциденты) составьте детальную спецификацию столбцов. Выберите наиболее подходящий тип данных из списка стандартных типов SQL (INT, VARCHAR, TEXT, DATE, FLOAT, BOOLEAN).

Пример таблицы спецификации:

Таблица: sensors

| Column Name<br>(Поле) | Data Type<br>(Тип) | Constraints<br>(Ограничения) | Description (Описание)                  |
|-----------------------|--------------------|------------------------------|---|
| sensor_id             | INT                | PK, Auto Increment           | Уникальный номер<br>(Первичный ключ)    |
| model_name            | VARCHAR(100)       | NOT NULL                     | Название модели<br>устройства           |
| last_active           | DATETIME           | NULL                         | Время последнего<br>полученного сигнала |
| battery_level         | INT                | CHECK (0-100)                | Текущий заряд батареи<br>в процентах    |

### Задание 3. Описание внешних ключей (Mapping Relationships).

Опишите, как таблицы связаны друг с другом на уровне кодов.

Для каждой связи «Один ко многим» укажите:

1. Какая таблица является **Родительской** (Parent)?
2. Какая таблица является **Дочерней** (Child)?
3. Какое поле в дочерней таблице станет **Foreign Key (FK)**?

Пример:

Связь Группы -> Транспорт:

- Parent: vehicles (Транспорт)
- Child: response\_groups (Группы)
- FK Column: В таблицу response\_groups добавляем поле vehicle\_id.

### Задание 4. Работа с документацией.

Изучите фрагмент документации к СУБД (на английском):

*"Use DECIMAL(P,S) for fixed-point arithmetic (e.g., money), and FLOAT for approximate scientific calculations."*

*Вопрос:* Какой тип данных вы выберете для хранения точных GPS-координат (широта и долгота) в системе мониторинга пожаров? Обоснуйте выбор, опираясь на текст выше.

## Методические указания по ходу выполнения работы (комментарии к заданиям, к выполнению заданий):

### 1. Переход от Концептуальной к Логической модели:

- **Концептуальная:** Понятна заказчику. ("Студент учится в Группе").
- **Логическая:** Понятна программисту. ("Таблица students имеет колонку group\_id, которая ссылается на groups.id").
- В логической модели исчезают связи "Многие ко многим" — они превращаются в физические таблицы-связки.

### 2. Типы данных (Шпаргалка):

- INT — целые числа (ID, количество).
- VARCHAR(N) — короткий текст до N символов (ФИО, название, госномер).
- TEXT — длинный текст (описание инцидента).
- FLOAT / DOUBLE — дробные числа (координаты, температура).
- BOOLEAN — истина/ложь (статус "Активен/Неактивен").

### 3. Правила именования:

В профессиональной разработке принято использовать английский язык.

- *Плохо:* Date\_Rozhdeniya, Table1.
- *Хорошо:* birth\_date, users.

Соблюдение этого правила демонстрирует вашу культуру работы с кодом.

### 4. Требования к отчету:

Отчет предоставляется в формате .docx или .pdf.

#### 1. Титульный лист.

#### 2. Цель работы.

#### 3. Ход работы:

- Таблица перевода названий (Задание 1).
- Спецификация таблиц (Задание 2 — минимум 2 таблицы).
- Список внешних ключей (Задание 3).
- Ответ на вопрос по документации (Задание 4).

#### 4. Вывод: (Например: «Научился переводить визуальную схему в спецификацию таблиц, выбрал типы данных для хранения информации о лесных пожарах...»).

## Тема практической работы № 6. Определение первичных и внешних ключей в таблицах, объем часов 4

У1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

У2. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии.

У4. Эффективно взаимодействовать и работать в коллективе и команде.

У5. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации.

У9. Пользоваться профессиональной документацией на государственном и иностранном языках.

**Цель практической работы** Углубленное изучение механизмов ссылочной целостности: выбор стратегии создания первичных ключей (естественные vs суррогатные), проектирование составных ключей и определение правил поведения базы данных при удалении связанных записей (CASCADE, RESTRICT).

### Задание(я):

**Контекст задачи:** Вы продолжаете работу над проектом «EcoGuard AI».

Структура таблиц готова (ПР №5). Теперь необходимо защитить данные от ошибок. Например, нельзя допустить ситуацию, когда Группу реагирования удалили, а в журнале «Выездов» остались записи, ссылающиеся на несуществующую группу (проблема «висячих ссылок»).

**Задание 1. Выбор типа Первичного ключа (Natural vs Surrogate).** Для следующих сущностей выберите, какой атрибут станет Первичным ключом (PK). Обоснуйте выбор.

1. **Таблица users (Сотрудники):** Использовать «Табельный номер» (который может измениться при переводе) или создать искусственный user\_id?
2. **Таблица vehicles (Транспорт):** Использовать «VIN-номер» (длинная строка из 17 символов, но уникальная) или vehicle\_id (короткое число)?
3. **Таблица country\_codes (Справочник стран для оборудования):** Использовать международный код (RU, EN, CN) или придумать ID?

**Задание 2. Настройка ссылочной целостности (Referential Integrity actions).**

Вам необходимо определить правило ON DELETE для внешних ключей.

*Ситуация:* Администратор пытается удалить запись о «Транспорте №5» из таблицы vehicles. Но этот транспорт указан в таблице incidents (на нем выезжали на пожар).

*Выберите одно из действий для схемы БД и поясните почему:*

- **Вариант А (CASCADE):** Автоматически удалить все записи об инцидентах, связанных с этой машиной.
- **Вариант Б (RESTRICT):** Запретить удаление машины, пока есть связанные инциденты (выдать ошибку).
- **Вариант В (SET NULL):** Удалить машину, а в поле `vehicle_id` в таблице инцидентов поставить пустое значение (NULL).

**Задание 3. Проектирование Составного первичного ключа (Composite Key).** Рассмотрите таблицу-связку `incident_participation` (Участие группы в инциденте), созданную в ПР №4. В этой таблице два внешних ключа: `group_id` и `incident_id`.

- *Задача:* Объясните, почему в этой таблице Первичным ключом должна стать пара этих полей (`group_id + incident_id`), а не каждое по отдельности.
- *Оформите:* Запишите структуру таблицы с указанием РК (`group_id, incident_id`).

#### **Задание 4. Работа с документацией.**

Прочитайте отрывок из документации PostgreSQL:

*"A foreign key must reference columns that either represent a primary key or a unique constraint. Adding a foreign key index is strongly recommended to improve JOIN performance."*

*Задание:* Переведите рекомендацию. На основе текста ответьте: нужно ли создавать индексы для полей, которые являются внешними ключами (FK) в вашей системе «EcoGuard»?

**Методические указания по ходу выполнения работы (комментарии к заданиям, к выполнению заданий):**

##### **1. Виды ключей:**

- **Суррогатный ключ (Surrogate Key):** Искусственное поле (обычно ID, счетчик 1, 2, 3...), не имеющее смысла в реальном мире. *Рекомендуется для большинства таблиц в ИИ-проектах для скорости.*
- **Естественный ключ (Natural Key):** Данные, которые уже есть у объекта (Номер паспорта, Email, VIN, ISBN). *Удобно для поиска, но если данные изменятся (смена паспорта), придется менять ссылки во всей базе.*

##### **2. Составной ключ:**

Используется, когда одна запись уникально идентифицируется не одним полем, а комбинацией. Например, один и тот же `group_id` встречается много раз, один `incident_id` — много раз. Но сочетание "Группа Альфа на Пожаре №1" — уникально.

**3. Правила целостности (Integrity):** Для систем сбора исторических данных (как наша «EcoGuard») **категорически не рекомендуется** использовать CASCADE для важных данных. Если вы удалите машину, вы не должны стереть историю пожаров, которые она тушила. Лучше использовать RESTRICT (защита от случайного удаления) или SET NULL (сохранение факта выезда без указания машины).

**4. Требования к отчету:** Отчет предоставляется в формате .docx или .pdf.

1. **Титульный лист.**

2. **Цель работы.**

3. **Ход работы:**

- Решение Задания 1 с обоснованием (3 пункта).
- Выбор стратегии удаления для Задания 2 (почему этот вариант лучше для аналитики?).
- Схема таблицы с составным ключом (Задание 3).
- Перевод и ответ на вопрос по документации (Задание 4).

4. **Вывод:** (Например: «Изучил различия ключей. Для системы выбрал стратегию RESTRICT, чтобы сохранить исторические данные для обучения ИИ...»).

**Тема практической работы № 7 Создание базы данных в СУБД на основе логической модели, объем часов \_\_4\_\_**

У1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

У2. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии.

У4. Эффективно взаимодействовать и работать в коллективе и команде.

У5. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации.

У7. Содействовать сохранению окружающей среды, эффективно действовать в чрезвычайных ситуациях.

**Цель практической работы** Реализация физической схемы данных в среде конкретной СУБД: трансляция логической модели в физические объекты, настройка специфических типов данных и ограничений целостности для обеспечения хранения экологических данных.

**Задание(я):**

**Инструментарий:** Работа выполняется в любой СУБД (PostgreSQL, MySQL, SQLite) с использованием графического интерфейса (DBeaver, pgAdmin, Workbench) или консоли.

**Задание 1. Физическая реализация справочников.** Используя спроектированную ранее схему «EcoGuard AI»:

1. Создайте пустую базу данных Eco\_Monitoring.
2. Реализуйте таблицы-справочники (Zones, Sensor\_Models).
3. **Важно:** При создании выберите физические типы данных, соответствующие вашей СУБД (например, VARCHAR vs TEXT, INTEGER vs SMALLINT). Обоснуйте выбор типов с точки зрения экономии памяти (Физическое проектирование).

**Задание 2. Реализация связей и ограничений (Constraints).** Реализуйте таблицу Sensors и таблицу фактов Incidents (Инциденты). Настройте физические ограничения:

- PRIMARY KEY: Назначьте первичные ключи.
- FOREIGN KEY: Установите связи. Для связи «Инцидент -> Зона» настройте правило ON DELETE RESTRICT (запрет удаления зоны, если в ней были пожары).
- DEFAULT: Для поля Created\_At (Дата создания записи) установите значение по умолчанию CURRENT\_TIMESTAMP.
- CHECK: Добавьте проверку для координат (широта от -90 до 90).

**Задание 3. Генерация схемы (Reverse Engineering).** После создания таблиц используйте инструмент СУБД для построения **Диаграммы физической модели**. Сравните её с вашей логической схемой «на бумаге». Все ли связи отобразились корректно?

**Задание 4. Заполнение тестовыми данными (Mock Data).** Внесите в базу данных 3-5 строк информации (можно вручную через интерфейс или скриптом). Данные должны моделировать реальную чрезвычайную ситуацию (например, «Резкое повышение CO2 в секторе 4»).

**Методические указания по ходу выполнения работы (комментарии к заданиям, к выполнению заданий):**

### **1. Отличие Логической модели от Физической:**

- В логической модели мы писали просто «Строка» или «Текст».
- В физической модели (в СУБД) вы должны выбрать: CHAR(10) (фиксированная длина, быстрее) или VARCHAR(50) (переменная длина, экономит место). Это и есть задача физического проектирования.

**2. Ограничения (Constraints):** Это «защита от дурака» на уровне физического файла БД. Даже если программист ошибется в коде программы, база данных не даст записать «кривые» координаты или удалить используемый справочник.

### 3. Требования к отчету:

- Скриншоты созданных таблиц в интерфейсе СУБД (структура столбцов).
- Скриншот полученной ER-диаграммы (Reverse Engineering).
- Скриншот таблицы с данными (подтверждение, что данные успешно внесены).

### Тема практической работы № 8 Настройка индексов для ускорения запросов, объем часов \_\_4\_\_

У1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

У2. Использовать современные средства поиска, анализа и интерпретации информации.

У4. Эффективно взаимодействовать и работать в коллективе и команде.

У5. Осуществлять устную и письменную коммуникацию.

У7. Содействовать сохранению окружающей среды.

**Цель практической работы** Формирование навыков оптимизации производительности базы данных: анализ планов выполнения запросов (Execution Plan), создание индексов различных типов и оценка их влияния на скорость обработки больших массивов данных.

#### Задание(я):

**Контекст задачи:** База данных наполнена большим объемом информации (имитация Big Data). Необходимо ускорить поиск критически важных событий для системы оповещения о пожарах.

**Задание 1. Анализ производительности без индексов (Full Scan).** Выполните поиск записи по критерию, который не является первичным ключом. *Пример запроса:* «Найти все датчики модели "Sensor-X200"». Посмотрите **План выполнения (Execution Plan)**. Найдите параметр стоимости (Cost) и тип операции (Seq Scan / Full Table Scan). Запишите эти значения.

#### Задание 2. Создание индекса (B-Tree).

Создайте индекс для поля, по которому производился поиск (Model\_Name). Повторите поиск и снова посмотрите План выполнения.

- Изменился ли тип операции? (Должно стать Index Scan).
- Как изменилась стоимость запроса?

#### Задание 3. Настройка уникального индекса (Unique Index).

В таблице Sensors есть поле Serial\_Number (Серийный заводской номер). Оно не является первичным ключом, но должно быть уникальным. Создайте **Уникальный индекс** для этого поля. Попробуйте внести в таблицу две записи с одинаковым серийным номером. Опишите результат (реакцию СУБД).

#### **Задание 4. Анализ "Стоимости" индекса (Trade-off).**

Обсудите и напишите вывод: Почему создание индексов замедляет операции вставки данных (INSERT) и обновления (UPDATE)? Для каких таблиц экологического мониторинга индексы жизненно необходимы, а для каких (например, архив логов) они будут лишними?

**Методические указания по ходу выполнения работы (комментарии к заданиям, к выполнению заданий):**

**1. Что такое Индекс:** Это специальная структура данных (обычно дерево B-Tree), которая хранится отдельно от таблицы и помогает СУБД находить строки не перебором всего файла, а по указателям.

#### **2. Как смотреть План выполнения:**

- В DBeaver/pgAdmin есть кнопка «Explain Execution Plan» (часто иконка с лупой или деревом).
- Ключевые слова для поиска:
  - Seq Scan — плохо (перебор всех строк).
  - Index Scan — хорошо (поиск по индексу).

#### **3. Требования к отчету:**

- Скриншот плана запроса «ДО» оптимизации.
- Скриншот созданного индекса (или SQL-код его создания).
- Скриншот плана запроса «ПОСЛЕ» оптимизации.
- Вывод о том, во сколько раз (или на сколько процентов) снизилась стоимость запроса.

**Тема практической работы № 9 Написание SQL-запросов для выборки данных из базы, объем часов \_\_4\_\_**

У1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

У2. Использовать современные средства поиска, анализа и интерпретации информации.

У5. Осуществлять устную и письменную коммуникацию на государственном языке РФ.

У8. Использовать средства физической культуры для поддержания здоровья.

У9. Пользоваться профессиональной документацией на государственном и иностранном языках.

**Цель практической работы** Формирование навыков извлечения информации из реляционных баз данных с использованием языка структурированных запросов SQL: освоение операторов проекции, фильтрации, сортировки и агрегации данных.

**Задание(я):**

**Контекст:** Работаем с таблицей `Sensor_Readings` (Показания датчиков), которая содержит поля: `id`, `sensor_id`, `value` (значение), `location` (зона), `created_at` (дата и время).

**Задание 1. Простая выборка (SELECT ... FROM).** Напишите запросы для решения следующих задач:

1. Вывести **все** данные из таблицы показаний (познакомиться с содержимым).
2. Вывести только столбцы `location` и `value` (проекция), чтобы не нагружать сеть лишними данными.

**Задание 2. Фильтрация данных (WHERE).** ИИ для обучения нужны не все данные, а только специфические. Напишите запросы:

1. Найти все показания, где уровень CO2 (`value`) превышает 800 единиц (потенциальный пожар).
2. Найти все записи, сделанные конкретным датчиком (`sensor_id = 55`).
3. **У9 (Documentation):** Используя документацию по операторам сравнения, выберите записи, где значение находится в диапазоне от 200 до 400 (`BETWEEN` или `>= AND <=`).

**Задание 3. Сортировка и ограничение (ORDER BY, LIMIT).**

1. Выведите список инцидентов, отсортированный по уровню опасности (от самого высокого к низкому).
2. Выведите «Топ-10» самых свежих измерений (сортировка по дате по убыванию + ограничение количества).

**Задание 4. Агрегация и группировка (GROUP BY).** Аналитику (или нейросети) часто нужны сводные данные.

1. Посчитайте общее количество записей в таблице (`COUNT`).

2. Найдите **среднее** значение температуры (AVG) для каждой зоны леса (GROUP BY location).

**Методические указания по ходу выполнения работы (комментарии к заданиям, к выполнению заданий):**

**1. Структура запроса SELECT (шпаргалка):** Порядок слов в SQL строгий (нарушать нельзя):

#### SQL

```
SELECT столбец1, столбец2 -- Что показать?
FROM таблица -- Откуда взять?
WHERE условие -- Какой фильтр?
ORDER BY поле DESC -- Как сортировать?
LIMIT 10; -- Сколько строк дать?
```

**2. Логические операторы:**

- AND — И (должны совпасть оба условия).
- OR — ИЛИ (должно совпасть хотя бы одно).
- NOT — НЕ (отрицание).

**3. Требования к отчету:** В отчете (Docx/PDF) представить таблицу:

- Левая колонка: Формулировка задания (например, "Найти среднее...").
- Правая колонка: Текст вашего SQL-запроса.
- Скриншот результата выполнения (табличка с данными), подтверждающий, что код работает.

**Тема практической работы № 10 Создание и модификация данных в таблицах, объем часов \_\_4\_\_**

У1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

У2. Использовать современные средства поиска, анализа и интерпретации информации.

У5. Осуществлять устную и письменную коммуникацию на государственном языке РФ.

У8. Использовать средства физической культуры для поддержания здоровья.

У9. Пользоваться профессиональной документацией на государственном и иностранном языках.

**Цель практической работы** *Освоение языка манипулирования данными (DML): навыки добавления новых записей, пакетной загрузки данных, изменения*

существующих значений и удаления устаревшей информации в соответствии с требованиями бизнес-логики.

### **Задание(я):**

**Контекст:** Система «EcoGuard» развивается. Необходимо добавить новый модуль для учета **Дронов-разведчиков** и управлять их статусами.

**Задание 1. Создание таблицы (DDL Recap).** Напишите запрос `CREATE TABLE` для создания таблицы `Drones`:

- `id` (Целое число, Первичный ключ, Автоинкремент).
- `model` (Текст, Название модели).
- `battery_level` (Целое число, Заряд %).
- `status` (Текст: 'Base', 'Flying', 'Repair').

### **Задание 2. Вставка данных (INSERT).**

1. Добавьте в базу одного нового дрона модели "SkyWatcher-1" с зарядом 100% и статусом 'Base'.
2. **Массовая вставка:** Одним запросом добавьте сразу три дрона разных моделей. (Это экономит ресурсы сервера по сравнению с тремя отдельными запросами).

**Задание 3. Обновление данных (UPDATE).** Дроны начали работу. Данные изменились.

1. Дрон с `id=1` взлетел. Измените его статус на 'Flying' и уменьшите заряд батареи до 90.
2. **Массовое обновление:** Прошла буря. Установите статус 'Repair' (В ремонт) для **всех** дронов, у которых заряд батареи меньше 20%.

### **Задание 4. Удаление данных (DELETE).**

1. Дрон модели "Old-v1" списан. Удалите его из базы по `id`.
2. **Очистка (Опасно!):** Напишите (но не выполняйте без разрешения преподавателя) запрос, который удалил бы **всех** дронов со статусом 'Repair'.

**Задание 5. Анализ ошибок.** Попробуйте выполнить запрос `INSERT INTO Drones (id, model) VALUES (1, 'Test');`. Система выдаст ошибку на английском языке (например: *Duplicate entry '1' for key 'PRIMARY'*). Переведите ошибку и объясните в отчете, почему СУБД запретила это действие.

**Методические указания по ходу выполнения работы (комментарии к заданиям, к выполнению заданий):**

## 1. Синтаксис модификации:

- `INSERT INTO table (col1, col2) VALUES (val1, val2);`
- `UPDATE table SET col1 = new_val WHERE id = 1; —`  
**Внимание!** Всегда пишите `WHERE`, иначе обновите всю таблицу.
- `DELETE FROM table WHERE id = 1; —` **Внимание!** Без `WHERE` удалит всё.

**2. Безопасный режим:** В некоторых SQL-редакторах (например, в MySQL Workbench) по умолчанию включен режим `Safe Updates`, который запрещает `UPDATE/DELETE` без использования первичного ключа. Если запрос не работает, проверьте настройки.

**3. Требования к отчету:** Отчет (Docx/PDF) должен содержать:

1. Код запросов для каждого задания.
2. Скриншоты:
  - Таблицы ПОСЛЕ вставки.
  - Таблицы ПОСЛЕ обновления (чтобы было видно, что статусы изменились).
3. Текст ошибки из Задания 5 с переводом и объяснением.

## Тема практической работы № 11 Оптимизация сложных SQL-запросов, объем часов \_4\_

У1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

У2. Использовать современные средства поиска, анализа и интерпретации информации.

У3. Планировать и реализовывать профессиональное и личностное развитие.

У5. Осуществлять устную и письменную коммуникацию.

У7. Содействовать сохранению окружающей среды.

**Цель практической работы** Формирование навыков рефакторинга SQL-кода для повышения производительности: замена неоптимальных конструкций (подзапросы, wildcards) на эффективные, анализ стоимости выборки и применение фильтрации на ранних этапах выполнения запроса.

### Задание(я):

**Контекст задачи:** В системе «EcoGuard AI» модуль аналитики работает слишком медленно. При поступлении сигнала о пожаре диспетчер ждет отчет 30 секунд. Ваша задача — переписать запросы так, чтобы они выполнялись мгновенно.

**Задание 1. «Убийца производительности»:** Подзапросы vs JOIN. *Ситуация:* Разработчик написал запрос для поиска датчиков в «активных» зонах, используя IN (SELECT ...). Это заставляет СУБД выполнять вложенный запрос для каждой строки. *Исходный медленный запрос:*

### SQL

```
SELECT * FROM Sensors
WHERE location_id IN (SELECT id FROM Zones WHERE status = 'Danger');
```

*Задание:* Перепишите этот запрос, используя INNER JOIN. Объясните в отчете, почему соединение таблиц обычно работает быстрее, чем зависимый подзапрос.

**Задание 2. Проблема поиска текста (LIKE).** *Ситуация:* Нужно найти все инциденты, в описании которых упоминается слово "дым". *Запросы на анализ:*

1. SELECT \* FROM Incidents WHERE description LIKE '%дым%';
2. SELECT \* FROM Incidents WHERE description LIKE 'дым%'; *Задание:* Сравните эти два запроса с помощью EXPLAIN. Какой из них **не может** использовать стандартный индекс (B-Tree) и почему? Какой вариант предпочтительнее для ИИ-поисковика?

**Задание 3. Фильтрация перед Группировкой (HAVING vs WHERE).** *Ситуация:* Нужно посчитать среднюю температуру только по зонам "Сибири". *Плохой вариант:* Сначала сгруппировать весь мир, а потом отсеять лишнее через HAVING. *Задание:* Напишите оптимизированный запрос, который сначала фильтрует данные (WHERE region = 'Siberia'), а только потом группирует их. *Обоснование:* Как это экономит память сервера?

**Задание 4. Саморазвитие.** Технологии меняются. То, что было медленно 5 лет назад, сегодня может работать быстро. *Задание:* Найдите в интернете статью (на Хабр, Medium или в документации PostgreSQL/MySQL) по теме "CTE vs Subquery performance" (Common Table Expressions против подзапросов). Напишите краткое резюме (2-3 предложения): всегда ли нужно избавляться от подзапросов в современных версиях СУБД?

**Методические указания по ходу выполнения работы (комментарии к заданиям, к выполнению заданий):**

**1. Принцип «SARGable»:** Запросы должны быть *Search ARGument ABLE* (дружелюбны к индексам).

- LIKE '%text' — не SARGable (индекс бесполезен, нужно сканировать всё).
- LIKE 'text%' — SARGable (можно найти по алфавиту).

**2. Порядок выполнения SQL:** Вы должны помнить, что SQL выполняется не в том порядке, как пишется.

1. FROM (взяли данные) -> 2. WHERE (отфильтровали лишнее) -> 3. GROUP BY (свернули) -> 4. HAVING (отфильтровали группы) -> 5. SELECT (показали).
- Вывод:* Чем раньше вы отсечете лишние строки (на этапе WHERE), тем быстрее отработает GROUP BY.

### 3. Требования к отчету: В отчете (Docx/PDF) представить:

- Пары запросов: «Исходный (Медленный)» и «Оптимизированный (Быстрый)».
- Скриншоты планов выполнения (EXPLAIN) для задания 2.
- Ответ на вопрос из задания 4 со ссылкой на источник.

## Тема практической работы № 12 Настройка параметров производительности базы данных, объем часов \_\_4\_\_

У1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

У2. Использовать современные средства поиска, анализа и интерпретации информации.

У3. Планировать и реализовывать профессиональное и личностное развитие.

У5. Осуществлять устную и письменную коммуникацию.

У7. Содействовать сохранению окружающей среды.

**Цель практической работы** Изучение серверной конфигурации СУБД: расчет оптимального выделения оперативной памяти (Buffer Pool), настройка ограничений подключений и управление журналами медленных запросов для диагностики проблем.

### Задание(я):

**Контекст задачи:** Код оптимизирован (в ПР №11), но сервер всё равно «тормозит». Оказывается, база данных использует настройки «по умолчанию», рассчитанные на слабые компьютеры. Вам, как администратору (DBA), нужно перенастроить конфигурационный файл (my.cnf или postgresql.conf) под мощный сервер для ИИ.

**Задание 1. Расчет буфера памяти (Buffer Pool / Shared Buffers).** База данных работает быстро, когда держит «горячие» данные в оперативной памяти (RAM), а не на диске. *Вводные:* Сервер «EcoGuard» имеет **16 ГБ RAM**. На сервере стоит *только* база данных. *Задание:* Рассчитайте, какое значение нужно установить для параметра `innodb_buffer_pool_size` (MySQL) или `shared_buffers` + кэш ОС (PostgreSQL).

- *Подсказка:* Обычно базе отдают 70-80% памяти.

- *Результат:* Запишите строку конфигурации, например:  
`innodb_buffer_pool_size = 12G.`

**Задание 2. Управление подключениями (Max Connections).** Микросервисы ИИ открывают сотни соединений. Если лимит исчерпан, система мониторинга пожаров упадет с ошибкой "Too many connections". *Задание:*

1. Узнайте текущее значение параметра `max_connections` (команда `SHOW VARIABLES LIKE 'max_connections';`).
2. Предложите новое значение, если известно, что работает 500 дронов и 50 пользователей-аналитиков.

**Задание 3. Логирование медленных запросов (Slow Query Log).** Чтобы знать, что оптимизировать, нужно поймать «тормозящие» запросы. *Задание:* Составьте список настроек для включения лога:

1. Включить лог (`slow_query_log = 1`).
2. Установить порог времени: считать «медленным» всё, что дольше **1 секунды** (`long_query_time = 1`). *Вопрос:* Почему нельзя ставить порог 0 секунд (логировать всё) на боевом сервере?

**Задание 4. Кейс «Надежность против Скорости» (ACID vs Performance).** Для ИИ иногда важнее скорость загрузки данных, чем 100% сохранность каждой строчки логов. *Исследование:* Изучите параметр `innodb_flush_log_at_trx_commit` (или `fsync` в Postgres).

- Какой режим (0, 1 или 2) самый безопасный (данные не пропадут при выключении света)?
- Какой режим самый быстрый?
- Какой режим вы выберете для таблицы `Forest_Fires` (критичные данные), а какой — для `Debug_Logs` (мусорные логи)?

**Методические указания по ходу выполнения работы (комментарии к заданиям, к выполнению заданий):**

**1. Работа с конфигурацией:** В реальной жизни эти настройки находятся в текстовых файлах (`my.ini`, `postgresql.conf`). Изменение этих параметров требует перезагрузки службы СУБД. В рамках работы мы моделируем этот процесс (пишем расчеты), так как у вас может не быть прав администратора на учебных ПК.

**2. Профессиональное развитие:** Настройки "по умолчанию" в СУБД часто устаревшие. Хороший специалист всегда проверяет конфигурацию под конкретное "железо". Не полагайтесь на авось — считайте мегабайты.

**3. Экологический аспект:** Правильно настроенный сервер тратит меньше электроэнергии (CPU не работает вхолостую) и быстрее реагирует на сигналы бедствия.

#### 4. Требования к отчету:

- Расчет памяти для Задания 1.
- Список команд/строк конфигурации для Заданий 2 и 3.
- Обоснованное решение по настройке надежности (Задание 4).

## II. Общие рекомендации

По всем вопросам, связанным с изучением дисциплины (включая самостоятельную работу), консультироваться с преподавателем.

## III. Контроль и оценка результатов

Оценка за выполнение практической работы выставляется по пятибалльной системе и учитывается как показатель текущей успеваемости студента.

| Качественная оценка индивидуальных образовательных достижений |                   | Критерии оценки результата  |
|---|-------------------|---|
| балл (оценка)   | вербальный аналог |   |
| 5   | отлично           | Работа выполнена в полном объеме с соблюдением необходимой последовательности действий, обучающийся самостоятельно и правильно воспроизвел предложенный алгоритм работы, программный код (SQL-запросы) или спроектированные схемы баз данных функционируют корректно и соответствуют требованиям задания. Обучающийся показывает глубокое понимание технологического процесса, умеет обосновать выбранный метод решения (например, выбор типа данных или индекса), соблюдает правила техники безопасности и нормы времени, а также демонстрирует культуру оформления кода (отступы, комментарии) и грамотно интерпретирует полученные результаты. |
| 4   | хорошо            | Работа выполнена полностью, но при выполнении были допущены незначительные ошибки технического или логического характера, которые были исправлены обучающимся самостоятельно или после наводящего вопроса   |

|   |                      |   |
|---|----------------------|---|
|   |                      | преподавателя. Программный код работоспособен, но может содержать неоптимальные решения (например, избыточные подзапросы) или незначительные нарушения стандартов оформления. Обучающийся в целом владеет приемами работы, соблюдает основные требования к качеству результата, но испытывает небольшие затруднения при анализе сложных ситуаций.   |
| 3 | удовлетворительно    | Работа выполнена не полностью (выполнены только обязательные базовые задания, пропущены задания повышенной сложности), или в ходе выполнения были допущены ошибки, влияющие на результат, которые обучающийся смог исправить только при непосредственной помощи преподавателя. Спроектированная база данных функционирует, но имеет недостатки в структуре (нарушение нормализации, отсутствие необходимых ограничений целостности) или SQL-запросы выполняются некорректно в граничных условиях. Обучающийся понимает общий принцип работы, но имеет пробелы в практических навыках. |
| 2 | не удовлетворительно | Работа не выполнена или выполнена с грубыми нарушениями, делающими результат непригодным к использованию (база данных не создается, запросы не выполняются, схема противоречит логике предметной области). Обучающийся не может самостоятельно воспроизвести алгоритм действий, не понимает сущности выполняемых операций, допускает нарушения правил техники безопасности или не может объяснить назначение написанного им программного кода, что свидетельствует об отсутствии сформированных практических умений.  |