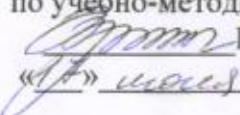


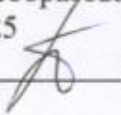
**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ НОВОСИБИРСКОЙ ОБЛАСТИ
«Новосибирский электромеханический колледж»
(ГБПОУ НСО «НЭК»)**

УТВЕРЖДАЮ
заместитель директора
по учебно-методической работе
 Е.О. Кравченко
«17» июня 2025 г.

**Методические указания
по выполнению практических работ
ОП.08 Основы проектирования баз данных**

09.02.07 Информационные системы и программирование

Разработчик: Комарова А.В.,
преподаватель

Рассмотрено
на заседании кафедры общеобразовательных дисциплин
Протокол №10 от 16.06.2025
Заведующий кафедрой  Белимова В.Г.

Методические рекомендации по выполнению практических работ учебного предмета разработаны на основе Федерального государственного образовательного стандарта (далее – ФГОС) по специальности 09.02.07 Информационные системы и программирование

Организация-разработчик: ГБПОУ НСО «НЭК»

Разработчик:

Комарова А.В., преподаватель высшей квалификационной категории

Согласовано:

Методист



Рахметова Т.Ю.

17.06.2025

Оглавление

Пояснительная записка.....	4
Порядок выполнения практической работы.....	5
Рекомендации по оформлению практической работы	6
Критерии оценки практической работы	6
ПРАКТИЧЕСКАЯ РАБОТА_№1. Нормализация реляционной БД, освоение принципов проектирования БД.....	7
ПРАКТИЧЕСКАЯ РАБОТА_№2. Нормализация реляционной БД, освоение принципов проектирования БД.....	10
ПРАКТИЧЕСКАЯ РАБОТА_№3. Задание ключей. Создание основных объектов БД	14
ПРАКТИЧЕСКАЯ РАБОТА_№4. Создание проекта БД. Создание БД. Редактирование и модификация таблиц	17
ПРАКТИЧЕСКАЯ РАБОТА_№5. Редактирование, добавление и удаление записей в таблице. Применение логических условий к записям. Открытие, редактирование и пополнение табличного файла.	21
ПРАКТИЧЕСКАЯ РАБОТА_№6. Создание ключевых полей. Задание индексов. Установление и удаление связей между таблицами.	25
ПРАКТИЧЕСКАЯ РАБОТА_№7. Проведение сортировки и фильтрации данных. Поиск данных по одному и нескольким полям. Поиск данных в таблице.	30
ПРАКТИЧЕСКАЯ РАБОТА_№8. Работа с переменными.	36
ПРАКТИЧЕСКАЯ РАБОТА_№9. Написание программного файла и работа с табличными файлами. Заполнение массива из табличного файла. Заполнение табличного файла из массива.....	40
ПРАКТИЧЕСКАЯ РАБОТА_№10. Добавление записей в табличный файл из двумерного массива. Работа с командами ввода-вывода. Использование функций для работы с массивами.	45
ПРАКТИЧЕСКАЯ РАБОТА_№11. Создание меню различных видов. Модификация и управление меню.	50
ПРАКТИЧЕСКАЯ РАБОТА_№12. Создание рабочих и системных окон. Добавление элементов управления рабочим окном.	55
ПРАКТИЧЕСКАЯ РАБОТА_№13. Создание файла проекта базы данных. Создание интерфейса входной формы. Использование исполняемого файла проекта БД, приемы создания и управления.	60
ПРАКТИЧЕСКАЯ РАБОТА_№14. Создание формы. Управление внешним видом формы.	64
ПРАКТИЧЕСКАЯ РАБОТА_№15. Задание значений и ограничений поля. Проверка введенного в поле значения. Отображение данных числового типа и типа дата	66
ПРАКТИЧЕСКАЯ РАБОТА_№16. Создание и модификация таблиц БД. Выборка данных из БД. Модификация содержимого БД. Обработка транзакций. Использование функций защиты для БД.	68
УЧЕБНО-МЕТОДИЧЕСКОЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ УЧЕБНОГО ПРЕДМЕТА	70

Пояснительная записка

Методические указания по выполнению лабораторных работ по учебной дисциплине «Основы проектирования баз данных» разработаны в соответствии с рабочей программой учебной дисциплины и предназначены для приобретения необходимых практических навыков и закрепления теоретических знаний, полученных обучающимися при изучении учебной дисциплины, обобщения и систематизации знаний перед экзаменом.

Методические указания предназначены для обучающихся специальности 09.02.07 Информационные системы и программирование.

Учебная дисциплина «Основы проектирования баз данных» является общеобразовательной учебной дисциплиной, изучается на 2 курсе и при ее изучении отводится значительное место выполнению практических работ.

Освоение содержания учебной дисциплины «Основы проектирования баз данных» во время выполнения практических работ обеспечивает достижение обучающимися следующих результатов:

Код ПК, ОК	Умения	Знания
ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 11.1-11.6	проектировать реляционную базу данных; использовать язык запросов для программного извлечения сведений из баз данных	основы теории баз данных; модели данных; особенности реляционной модели и проектирование баз данных; изобразительные средства, используемые в ER-моделировании; основы реляционной алгебры; принципы проектирования баз данных; обеспечение непротиворечивости и целостности данных; средства проектирования структур баз данных; язык запросов SQL

Матрица формирования компетенций (по работам)

№ ПР	Название практической работы	Формируемые компетенции (ОК / ПК)
1-2	Нормализация реляционной БД, освоение принципов проектирования БД	ОК 1, ОК 2, ОК 4, ОК 5, ПК 11.1
3	Задание ключей. Создание основных объектов БД	ОК 2, ОК 5, ПК 11.1, ПК 11.2
4	Создание проекта БД. Создание БД. Редактирование и модификация таблиц	ОК 2, ОК 5, ПК 11.2

№ ПР	Название практической работы	Формируемые компетенции (ОК / ПК)
5	Редактирование, добавление и удаление записей в таблице.	ОК 5, ПК 11.2, ПК 11.5
6	Создание ключевых полей. Задание индексов. Установление связей.	ОК 4, ОК 5, ПК 11.1, ПК 11.2, ПК 11.3
7	Проведение сортировки и фильтрации данных. Поиск данных.	ОК 4, ОК 5, ПК 11.5
8	Работа с переменными.	ОК 5, ПК 11.2
9-10	Написание программного файла... Работа с массивами.	ОК 2, ОК 5, ПК 11.2, ПК 11.6
11-12	Создание меню... Создание рабочих окон.	ОК 5, ПК 11.5
13	Создание файла проекта БД. Создание интерфейса входной формы.	ОК 1, ОК 2, ОК 5, ОК 9, ОК 10, ПК 11.2, ПК 11.5
14	Создание формы. Управление внешним видом формы.	ОК 5, ПК 11.5
15	Задание значений и ограничений поля. Проверка введенного значения.	ОК 5, ПК 11.2, ПК 11.4, ПК 11.5
16	Создание и модификация таблиц... Обработка транзакций. Защита БД.	ОК 4, ОК 5, ПК 11.2, ПК 11.3, ПК 11.4, ПК 11.5

В соответствии с рабочей программой учебной дисциплины «Основы проектирования баз данных» практические работы сгруппированы в конце четвертого семестра. Целесообразность данной группировки обусловлена необходимостью обобщения и систематизации знаний перед экзаменом.

Рабочая программа учебной дисциплины предусматривает проведение практических работ в объеме 32 часа.

Порядок выполнения практической работы

- записать название работы, ее цель в тетрадь;
 - выполнить основные задания в соответствии с ходом работы;
- ☐ выполнить индивидуальные задания.

Рекомендации по оформлению практической работы

Задания выполняются обучающимися по шагам. Необходимо строго придерживаться порядка действий, описанного в практической работе

Результаты выполнения практических заданий необходимо сохранять в своей папке на компьютере или USB – накопителе.

В случае пропуска занятий обучающийся осваивает материал самостоятельно в свободное от занятий время и сдает практическую работу с пояснениями о выполнении.

Критерии оценки практической работы

- наличие цели выполняемой работы, выполнение более половины основных заданий (удовлетворительно);
- наличие цели выполняемой работы, выполнение всех основных и более половины дополнительных заданий (хорошо);
- наличие цели выполняемой работы, выполнение всех основных и индивидуальных заданий (отлично).

ПРАКТИЧЕСКАЯ РАБОТА №1. Нормализация реляционной БД, освоение принципов проектирования БД

по дисциплине «Основы проектирования баз данных»

Тема: Проектирование структуры базы данных. Нормализация отношений.

1. Цель и задачи практической работы

Цель: Приобретение навыков проектирования реляционной базы данных на основе анализа требований, выделения сущностей и нормализации отношений.

Задачи:

1. Научиться анализировать словесное описание предметной области и выделять из него концептуальные требования.
 2. Научиться группировать атрибуты в сущности.
 3. Освоить метод нормализации (приведение таблиц к 1НФ, 2НФ и 3НФ) для устранения избыточности и аномалий данных.
 4. Научиться оформлять структуру спроектированных таблиц.
-

2. Теоретические сведения

- **Сущность** – это объект реального мира (например, Поставщик, Товар), информация о котором хранится в базе данных. В реляционной модели сущности соответствуют таблицам.
- **Атрибут** – это характеристика сущности (например, Название, Адрес). В таблице атрибутам соответствуют столбцы (поля).
- **Первичный ключ (РК)** – это уникальный идентификатор записи в таблице (например, id, уникальный номер). Его значение не может повторяться.
- **Нормализация** – это процесс преобразования структуры базы данных для устранения дублирования данных и обеспечения их логической целостности.

Основные нормальные формы:

1. **Первая нормальная форма (1НФ):**
 - Все значения атрибутов атомарны (неразделимы, не являются множествами или списками).
 - Все строки в таблице уникальны (нет повторяющихся записей).
 - Каждая таблица имеет первичный ключ.
 2. **Вторая нормальная форма (2НФ):**
 - Таблица находится в 1НФ.
 - Все неключевые атрибуты полностью зависят от **всего** первичного ключа (а не только от его части, если ключ составной).
 3. **Третья нормальная форма (3НФ):**
 - Таблица находится в 2НФ.
 - Все неключевые атрибуты зависят **только от первичного ключа** и не зависят друг от друга (нет транзитивных зависимостей).
-

3. Пример решения задачи (магазин «Молоко»)

Исходные требования:

Наименование товара, Количество товара, Дата накладной, Дата изготовления, Наименование поставщика, Адрес поставщика, № накладной, Цена за единицу, Стоимость товара, Ед. измерения.

Шаг 1: Выделение сущностей и атрибутов

1. **Сущность «Поставщик»:** Наименование_поставщика, Адрес_поставщика.
2. **Сущность «Товар»:** Наименование_товара, Кол-во_товара, Дата_накладной, Дата_изготовления, №_накладной, Цена_за_единицу, Стоимость_товара, Ед._измерения.

Шаг 2: Нормализация до 1НФ

Проблемы в «Товаре»: Стоимость товара = Кол-во товара * Цена за единицу. Это вычисляемое поле, его хранить не нужно.

Условия 1НФ:

1. Атрибуты атомарны.
2. Нет повторяющихся строк.
3. Есть первичный ключ.
4. Удалены вычисляемые поля.

Вывод: Добавляем уникальный id для Поставщика. В сущности «Товар» остается атрибут №_накладной, который может повторяться для разных товаров. Необходимо выделить сущность «Накладная», а товары перенести в отдельную таблицу «Поставка».

Шаг 3: Нормализация до 2НФ и 3НФ

Проектируем структуру:

1. **Таблица** Поставщики: id_поставщика (PK), Наименование, Адрес.
2. **Таблица** Товары: id_товара (PK), Наименование, Ед_измерения.
3. **Таблица** Накладные: id_накладной (PK), Номер, Дата, id_поставщика (ссылка на Поставщиков).
4. **Таблица** Поставки (связь "многие-ко-многим" между Товарами и Накладными): id_поставки (PK), id_накладной (FK), id_товара (FK), Количество, Цена_закупки, Дата_изготовления.

Результат: Все таблицы находятся в 3НФ.

4. Контрольные вопросы

1. Что такое сущность и атрибут в контексте баз данных?
2. Для чего нужен первичный ключ?
3. Перечислите три основные нормальные формы и их ключевые требования.
4. Что такое транзитивная зависимость? Приведите пример.
5. Почему не рекомендуется хранить в таблице вычисляемые поля (например, «Стоимость»)?

5. Задания к практической работе

Общее задание: Спроектируйте структуру базы данных на основе предоставленных требований, выполнив все шаги нормализации. Результат оформите в виде отчета.

Вариант 1: Библиотека

Требования: ФИО читателя, Номер читательского билета, Название книги, Автор книги, Дата выдачи, Дата возврата, Жанр книги, Издательство, Год издания, Штраф за просрочку.

Задание: Выделите не менее 4 сущностей. Приведите таблицы к 3НФ. Определите типы данных и первичные ключи.

Вариант 2: Записная книжка (Контакты)

Требования: ФИО контакта, Домашний телефон, Мобильный телефон, Рабочий телефон, email, Домашний адрес (улица, дом, квартира), Место работы, Должность, Дата рождения.

Задание: Выделите не менее 3 сущностей. Приведите таблицы к 3НФ. Предусмотрите хранение нескольких телефонов одного типа для контакта.

Вариант 3: Салон красоты

Требования: ФИО мастера, Специализация мастера, ФИО клиента, Контактный телефон клиента, Название услуги, Цена услуги, Дата и время записи, Статус записи (выполнено/запланировано).

Задание: Выделите не менее 4 сущностей. Приведите таблицы к 3НФ. Спроектируйте таблицу для хранения расписания.

Требования к отчету:

1. Титульный лист с названием работы, ФИО студента, группой.
2. Цель работы.

3. Раздел по выбранному варианту:
- **Концептуальные требования** (таблица с перечислением атрибутов).
 - **Выделение сущностей**.
 - **Процесс нормализации:** для каждой формы (1НФ, 2НФ, 3НФ) запишите выявленные проблемы и действия по их решению.
 - **Итоговая структура таблиц:** для каждой таблицы укажите ее название (рус/англ), поля (название, тип, размер, примечание – РК/ФК).
 - **Схема базы данных:** графическое изображение таблиц со связями (можно нарисовать от руки).
4. Ответы на контрольные вопросы.
5. Вывод по работе (что сделали, чему научились, какие были сложности).

ПРАКТИЧЕСКАЯ РАБОТА_№2.Нормализация реляционной БД, освоение принципов проектирования БД
по дисциплине «Основы проектирования баз данных»
Тема: Нормализация отношений. Анализ функциональных зависимостей.

1. Цели и задачи практической работы

Цель работы: Научиться выявлять функциональные зависимости в данных и применять процесс нормализации (до 3НФ) для устранения аномалий и создания оптимальной структуры таблиц.

Задачи:

1. Закрепить понятия: первичный ключ, функциональная зависимость.
 2. Научиться анализировать таблицу с данными и определять:
 - Возможные первичные ключи.
 - Функциональные зависимости между полями.
 3. Научиться последовательно приводить таблицу к первой (1НФ), второй (2НФ) и третьей (3НФ) нормальным формам.
 4. Научиться строить схему (ER-диаграмму) получившихся таблиц со связями.
-

2. Теоретические сведения

Нормализация – это процесс организации данных в базе данных для уменьшения избыточности и устранения аномалий (ошибок) при вставке, обновлении и удалении данных.

Основные нормальные формы:

1. **Первая нормальная форма (1НФ):**
 - В таблице нет повторяющихся строк.
 - Все значения в ячейках атомарны (неделимы, не являются списками).
 - У таблицы есть **первичный ключ (РК)** – уникальный идентификатор каждой записи.
2. **Вторая нормальная форма (2НФ):**
 - Таблица уже находится в 1НФ.
 - **Дополнительное условие:** Все неключевые поля (не входящие в первичный ключ) должны полностью зависеть от **всего составного первичного ключа**, а не от его части.
3. **Третья нормальная форма (3НФ):**
 - Таблица уже находится в 2НФ.
 - **Дополнительное условие:** Все неключевые поля зависят **только от первичного ключа** и не зависят друг от друга (нет **транзитивных зависимостей**).

Функциональная зависимость (ФЗ) – это связь между полями таблицы, при которой значение одного поля **однозначно определяет** значение другого поля. Записывается как **X -> Y** (читается: «X функционально определяет Y»).

Пример: В таблице Студент (Номер_зачетки, ФИО, Группа)

- Номер_зачетки -> ФИО (по номеру зачетки однозначно находим ФИО).
 - Номер_зачетки -> Группа.
 - Группа -> Номер_зачетки – **НЕВЕРНО**, так как в группе много студентов.
-

3. Пример решения задачи

Дана исходная таблица «Заказы»:

Номер_Заказа	Дата	Клиент	Телефон_клиента	Товар	Цена_товара
101	2023	Иванов	+7-999-123-45-	Монитор	15 000

Номер_Заказа	Дата	Клиент	Телефон_клиента	Товар	Цена_товара
	-10-01	А.А.	67		
101	2023-10-01	Иванов А.А.	+7-999-123-45-67	Клавиатура	3 000
102	2023-10-02	Петров А.Б.	+7-999-987-65-43	Монитор	15 000

Шаг 1: Анализ и 1НФ

- **Первичный ключ?** Не может быть Номер_Заказа, так как он повторяется для разных товаров. Первичный ключ **составной**: (Номер_Заказа, Товар).
- **Проблема 1НФ:** Поля Клиент, Телефон_клиента, Дата повторяются для одной строки заказа (не атомарны по смыслу). Таблица уже в 1НФ формально (все ячейки атомарны), но есть избыточность данных.

Шаг 2: Приведение к 2НФ

- У нас составной ключ (Номер_Заказа, Товар).
- **Анализ зависимостей:**
 - (Номер_Заказа, Товар) -> Цена_товара – зависит от всего ключа.
 - Номер_Заказа -> Дата, Клиент, Телефон_клиента – зависят только от части ключа! Это нарушает 2НФ.
- **Решение:** Выделяем две таблицы.
 1. **Таблица Заказы:** Номер_Заказа (РК), Дата, Клиент, Телефон_клиента.
 2. **Таблица Позиции_Заказов:** ID_позиции (РК), Номер_Заказа (FK), Товар, Цена_товара.

Шаг 3: Приведение к 3НФ (анализ новой таблицы Заказы)

- В таблице Заказы: Номер_Заказа (РК) -> Клиент, Телефон_клиента.
- **Есть ли транзитивная зависимость?** Да! Клиент -> Телефон_клиента. Телефон зависит не только от ключа (Номер_Заказа), но и от другого неключевого поля (Клиент). Это нарушает 3НФ.
- **Решение:** Выделяем еще одну таблицу.
 1. **Таблица Заказы:** Номер_Заказа (РК), Дата, ID_Клиента (FK).
 2. **Таблица Клиенты:** ID_Клиента (РК), Клиент, Телефон_клиента.
 3. **Таблица Позиции_Заказов:** ID_позиции (РК), Номер_Заказа (FK), Товар, Цена_товара.

Итоговая схема:

text

[Клиенты] 1---< [Заказы] 1---< [Позиции_Заказов]
 PK:ID_Клиента PK:Номер_Заказа PK:ID_позиции
 FK:ID_Клиента FK:Номер_Заказа

4. Контрольные вопросы

1. Что такое избыточность данных и к каким проблемам (аномалиям) она приводит?
2. Объясните своими словами, что означает «поля функционально зависят от всего составного ключа» (условие 2НФ).
3. Что такое транзитивная зависимость? Приведите пример из жизни (не из БД).

4. Почему после нормализации таблиц обычно увеличивается количество таблиц в базе данных? Это хорошо или плохо?
5. Что такое внешний ключ (FK) и для чего он используется в нормализованной базе данных?

5. Задания к практической работе

Общее задание: Для вашего варианта исходной таблицы выполните следующие шаги:

1. Определите возможный **первичный ключ**.
2. Выявите все **функциональные зависимости** между полями.
3. Объясните, какая нормальная форма (1НФ, 2НФ, 3НФ) нарушена в исходной таблице и почему.
4. Проведите **нормализацию до 3НФ**. Разбейте исходную таблицу на несколько таблиц.
5. Для каждой новой таблицы укажите ее **название, первичный ключ (PK) и внешние ключи (FK)**, если они есть.
6. Нарисуйте простую схему связи между получившимися таблицами (например, в виде блок-схемы).

Варианты заданий:

Вариант

Спортсмен	Вид_спорта	Тренер
Иванов Алексей	Плавание	Петрова О.Н.
Сидорова Мария	Легкая атлетика	Петрова О.Н.
Козлов Дмитрий	Плавание	Петрова О.Н.
Новикова Анна	Гимнастика	Смирнов И.В.

Подсказка: Обратите внимание на связь между "Вид_спорта" и "Тренер".

Вариант 2

Наименование Вуза	Адрес	Команда
КрасГУ	Красноярск, пр-т Свободный,79	МатФак
КрасГУ	Красноярск, пр-т Свободный,79	ЭконФак
КГТУ	Красноярск, ул.Киренского,26	ФизФак

Подсказка: Определите, что может однозначно идентифицировать запись. Есть ли зависимость между неключевыми полями?

Требования к оформлению отчета:

1. Титульный лист.
2. Номер варианта и исходная таблица.
3. Поэтапное решение (шаги 1-6 из задания).
4. Ответы на контрольные вопросы.
5. **Вывод:** Опишите, какие недостатки (аномалии) были в исходной таблице и как нормализация их устранила.

Критерии оценки практической работы №2 «Нормализация реляционной БД»

Оценка «5» (Отлично) ставится, если студент:

- **Правильно определил первичный ключ** исходной таблицы.
- **Полно и верно выявил все функциональные зависимости (ФЗ)**, включая транзитивные.
- **Корректно и аргументированно объяснил**, какие нормальные формы нарушены в исходной таблице и почему (указал на частичную зависимость для 2НФ или транзитивную для 3НФ).
- **Безупречно провёл нормализацию до 3НФ:**
 - Количество и состав новых таблиц оптимальны и полностью устраняют выявленные зависимости.
 - Для каждой таблицы корректно указаны названия, первичные (РК) и внешние ключи (FK).
- **Схема связей** нарисована правильно, типы связей (1:M) указаны верно.
- **Ответы на контрольные вопросы** даны полные, своими словами, демонстрируют понимание теории.
- **Вывод** отражает понимание цели нормализации, описаны устранённые аномалии.
- Работа оформлена аккуратно, логично, в полном объёме.

Оценка «4» (Хорошо) ставится, если студент:

- Допустил **незначительные ошибки** в определении ФЗ (например, пропустил одну неочевидную зависимость).
- В целом правильно провёл нормализацию, но в структуре получившихся таблиц есть **неоптимальность** (например, можно было объединить два поля, или наоборот, разделить лишний раз), **не нарушающая принципы НФ**.
- В схеме связей есть **небольшая неточность** (например, не везде указаны FK).
- Ответы на вопросы даны в основном верно, но не всегда полно.
- Оформление удовлетворительное.

Оценка «3» (Удовлетворительно) ставится, если студент:

- Определил первичный ключ, но **не смог выявить ключевые ФЗ**, необходимые для нормализации.
- Провёл нормализацию, но **не до конца** (например, только до 2НФ) или с **грубыми ошибками** в структуре (состав ключей неверный, не выделена таблица для устранения транзитивной зависимости).
- Схема связей нарисована схематично или с ошибками.
- Ответы на вопросы **поверхностные**, с использованием теории без глубокого понимания.
- Работа выполнена и сдана, но демонстрирует **минимальное понимание** процесса нормализации.

Оценка «2» (Неудовлетворительно) ставится, если студент:

- **Не определил** первичный ключ.
- **Не выявил** функциональные зависимости.
- **Не провёл** нормализацию или результат полностью неверен (таблицы не соответствуют данным, нарушены основы НФ).
- **Не ответил** на большинство контрольных вопросов.
- Работа **не выполнена** или выполнена формально, без понимания сути задания.

ПРАКТИЧЕСКАЯ РАБОТА_№3. Задание ключей. Создание основных объектов БД по дисциплине «Основы проектирования баз данных»

Тема: Создание форм для ввода и просмотра данных

1. ЦЕЛИ И ЗАДАЧИ ПРАКТИЧЕСКОЙ РАБОТЫ

Цель: Научиться создавать и настраивать формы в LibreOffice Base для удобного ввода, просмотра и редактирования данных в таблицах.

Задачи:

1. Научиться создавать простые формы с помощью Мастера форм.
 2. Освоить создание форм в режиме конструктора.
 3. Научиться добавлять различные элементы управления на форму (поля, надписи, кнопки).
 4. Научиться изменять свойства элементов формы.
 5. Освоить создание форм с подчиненными формами для работы со связанными таблицами.
-

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Форма – это объект базы данных, предназначенный для удобного ввода, просмотра и редактирования данных. Формы могут отображать данные из одной или нескольких таблиц, скрывая сложность структуры базы данных от пользователя.

Основные элементы формы:

- **Поле** – отображает данные из определенного поля таблицы
- **Надпись** – текстовое пояснение для полей
- **Кнопка** – для выполнения действий (сохранение, переход между записями и др.)
- **Панель навигации** – стандартная панель для перемещения по записям

Способы создания форм в LibreOffice Base:

1. **Мастер форм** – автоматическое создание формы по выбранным полям таблицы
 2. **Конструктор форм** – ручное создание и настройка формы
 3. **Форма в режиме таблицы** – простая табличная форма
-

3. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ

Задача: Создать форму для ввода данных в таблицу "Студенты"

Шаг 1. Создание формы с помощью Мастера

1. Откройте базу данных "Учебный_отдел.odb" (созданную в предыдущей работе)
2. В левой панели выберите "Формы"
3. В разделе "Задачи" выберите "Использовать мастер для создания формы..."
4. В окне мастера:
 - Выберите таблицу "Студенты"
 - Выберите все поля для отображения в форме
 - Выберите макет формы (например, "В столбец")
 - Выберите стиль оформления
 - Введите имя формы "Форма_Студенты"
5. Нажмите "Готово"

Шаг 2. Использование формы

1. Дважды щелкните по созданной форме "Форма_Студенты"
2. Используйте кнопки навигации внизу формы:
 - ► - следующая запись
 - ◀ - предыдущая запись
 - ►| - последняя запись
 - |◀ - первая запись
 - ★ - новая запись

3. Введите несколько новых записей о студентах

Шаг 3. Создание формы в режиме конструктора

1. В разделе "Задачи" выберите "Создать форму в режиме дизайнера"
 2. Появится пустая форма и палитра элементов
 3. Из палитры перетащите на форму:
 - Надписи для заголовков
 - Поля из таблицы "Студенты"
 4. Отформатируйте элементы (измените шрифт, размер, цвет)
 5. Сохраните форму под именем "Форма_Студенты_Ручная"
-

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое форма в базе данных и для чего она нужна?
 2. Какие существуют способы создания форм в LibreOffice Base?
 3. Как добавить поле таблицы на форму?
 4. Как изменить внешний вид элемента формы?
 5. Как создать кнопку на форме?
 6. Что такое подчиненная форма и когда она используется?
 7. Как перейти от формы к таблице с данными?
 8. Какие элементы управления можно разместить на форме?
 9. Как изменить порядок табуляции на форме?
 10. Можно ли через форму удалить запись из таблицы?
-

5. ЗАДАНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ

Вариант 1: Формы для базы "Библиотека"

На основе базы данных "Библиотека.odb" (созданной ранее):

1. Создайте форму "Форма_Читатели" с помощью Мастера для таблицы "Читатели"
2. Создайте форму "Форма_Книги" в режиме конструктора для таблицы "Книги"
3. Добавьте на форму "Форма_Книги" кнопку "Новая запись"
4. Измените цвет фона формы и шрифт заголовков
5. Создайте форму для таблицы "Выдача" с отображением связанных данных из "Читатели" и "Книги"

Вариант 2: Формы для базы "Магазин"

На основе базы данных "Магазин.odb":

1. Создайте форму для ввода данных о поставщиках
2. Создайте форму для товаров с выпадающим списком выбора поставщика
3. Добавьте на форму товаров кнопки навигации (следующий, предыдущий, новый)
4. Измените размер полей на формах для удобного отображения данных
5. Создайте главную форму с кнопками перехода к другим формам

Вариант 3: Формы для базы "Поликлиника"

На основе базы данных "Поликлиника.odb":

1. Создайте форму для регистрации новых пациентов
2. Создайте форму для врачей с группировкой по специальностям
3. Создайте форму для записи на прием с выбором пациента и врача из списков
4. Добавьте на все формы заголовки и поясняющие надписи
5. Настройте порядок перехода между полями на формах

Требования к выполнению:

1. Для своего варианта создайте не менее 3 различных форм
2. Используйте как Мастер форм, так и режим конструктора
3. Настройте внешний вид форм (цвета, шрифты, расположение элементов)
4. Добавьте на формы элементы навигации
5. Протестируйте формы, введя через них не менее 5 записей в каждую таблицу

Отчет должен содержать:

1. Титульный лист
 2. Скриншоты созданных форм
 3. Описание выполненных действий для каждой формы
 4. Ответы на контрольные вопросы
 5. Выводы о проделанной работе
-

6. КРИТЕРИИ ОЦЕНКИ

Оценка "5" (Отлично):

- Созданы все требуемые формы для выбранного варианта
- Формы корректно отображают и позволяют редактировать данные
- Использованы различные способы создания форм (Мастер и конструктор)
- Настроен удобный и эстетичный интерфейс форм
- Добавлены дополнительные элементы управления
- Работа сдана в срок, отчет оформлен полностью

Оценка "4" (Хорошо):

- Созданы не все требуемые формы
- Формы функционируют, но есть незначительные ошибки в отображении
- Не все способы создания форм использованы
- Интерфейс форм требует доработки
- Отчет оформлен с небольшими замечаниями

Оценка "3" (Удовлетворительно):

- Создана только одна форма
- Форма имеет серьезные недостатки в функциональности
- Не использован режим конструктора для настройки формы
- Отчет оформлен не полностью

Оценка "2" (Неудовлетворительно):

- Формы не созданы
- Работа не выполнена или сдана не в срок

ПРАКТИЧЕСКАЯ РАБОТА_№4. Создание проекта БД. Создание БД. Редактирование и модификация таблиц

по дисциплине «Основы проектирования баз данных»

Тема: Создание проекта БД. Создание БД. Редактирование и модификация таблиц

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель: Научиться создавать проект базы данных, реализовывать его в LibreOffice Base, а также редактировать и модифицировать структуру существующих таблиц.

Задачи:

1. Разработать проект простой базы данных из 2-3 связанных таблиц.
2. Создать базу данных в LibreOffice Base на основе разработанного проекта.
3. Создать таблицы с заданными полями, типами данных и первичными ключами.
4. Научиться редактировать структуру таблиц: добавлять, удалять и переименовывать поля.
5. Освоить изменение свойств полей (тип данных, размер, формат).
6. Создать связи между таблицами.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.1. Проектирование БД

Перед созданием базы данных необходимо разработать её проект. Проект включает:

- **Определение сущностей (объектов)** – каждая сущность станет таблицей.
- **Определение атрибутов (свойств) сущностей** – станут полями таблиц.
- **Выделение первичного ключа (PK)** – уникального идентификатора для каждой записи.
- **Определение связей** между таблицами с помощью **внешних ключей (FK)**.

2.2. Типы данных в LibreOffice Base

Основные типы данных:

- **INTEGER** – целые числа
- **VARCHAR(n)** – текст длиной до n символов
- **DATE** – дата
- **DECIMAL** – числа с фиксированной точкой (для денежных сумм)
- **BOOLEAN** – логический тип (Да/Нет)

2.3. Редактирование таблиц

После создания таблиц может потребоваться их модификация:

- **Добавление полей** – для включения новых данных
- **Удаление полей** – для удаления ненужной информации
- **Изменение свойств полей** – типа данных, размера, формата
- **Создание связей** – для обеспечения целостности данных

2.4. Связи между таблицами

Связи устанавливаются между **первичным ключом (PK)** одной таблицы и **внешним ключом (FK)** другой таблицы. Основные типы связей:

- **Один-ко-многим** – одна запись в главной таблице связана с несколькими записями в подчиненной.
- **Один-к-одному** – одна запись в одной таблице связана с одной записью в другой.

3. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ

Задача: Создать базу данных "Библиотека" для учета книг и читателей.

Шаг 1. Проектирование

1. Определяем сущности:
 - **Читатели** – информация о читателях
 - **Книги** – информация о книгах
 - **Выдача** – информация о выдаче книг читателям
2. Определяем атрибуты:

- **Читатели:** ID_читателя, Фамилия, Имя, Телефон
 - **Книги:** ID_книги, Название, Автор, Год_издания
 - **Выдача:** ID_выдачи, ID_читателя, ID_книги, Дата_выдачи, Дата_возврата
3. Определяем ключи:
- РК для "Читатели" – ID_читателя
 - РК для "Книги" – ID_книги
 - РК для "Выдача" – ID_выдачи
 - FK в "Выдача": ID_читателя → "Читатели", ID_книги → "Книги"
- Шаг 2. Создание БД в LibreOffice Base
1. Запустить LibreOffice Base
 2. Создать новую базу данных "Библиотека.odb"
 3. Сохранить в указанной папке
- Шаг 3. Создание таблиц
1. **Таблица "Читатели":**
text
Поля:
 - ID_читателя: INTEGER (Автозначение: Да) - РК
 - Фамилия: VARCHAR(30)
 - Имя: VARCHAR(20)
 - Телефон: VARCHAR(15)
 2. **Таблица "Книги":**
text
Поля:
 - ID_книги: INTEGER (Автозначение: Да) - РК
 - Название: VARCHAR(100)
 - Автор: VARCHAR(50)
 - Год_издания: INTEGER
 3. **Таблица "Выдача":**
text
Поля:
 - ID_выдачи: INTEGER (Автозначение: Да) - РК
 - ID_читателя: INTEGER - FK
 - ID_книги: INTEGER - FK
 - Дата_выдачи: DATE
 - Дата_возврата: DATE
- Шаг 4. Создание связей
1. Открыть меню "Сервис" → "Связи"
 2. Добавить все три таблицы
 3. Создать связи:
 - "Читатели".ID_читателя → "Выдача".ID_читателя
 - "Книги".ID_книги → "Выдача".ID_книги
- Шаг 5. Заполнение таблиц тестовыми данными
1. Добавить 3-5 читателей
 2. Добавить 5-7 книг
 3. Создать 2-3 записи о выдаче книг
- Шаг 6. Редактирование таблиц
1. Добавить в таблицу "Книги" поле "Жанр: VARCHAR(30)"
 2. Изменить размер поля "Автор" на 70 символов
 3. Добавить в таблицу "Читатели" поле "Email: VARCHAR(50)"
-

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое первичный ключ и для чего он нужен?
2. Что такое внешний ключ и как он создается?
3. Какие типы данных вы использовали в своей базе данных?
4. Как добавить новое поле в существующую таблицу?
5. Как изменить тип данных существующего поля?
6. Для чего нужны связи между таблицами?
7. Как создать связь между таблицами в LibreOffice Base?
8. Что произойдет, если попытаться удалить запись, на которую ссылаются другие таблицы?
9. Как изменить размер текстового поля?
10. Что означает параметр "Автозначение" для поля?

5. ЗАДАНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ

Вариант 1: "Учебный отдел"

Создайте базу данных для учета студентов и предметов.

1. **Таблица "Студенты"**: ID_студента (PK), Фамилия, Имя, Группа, Год_поступления
2. **Таблица "Предметы"**: ID_предмета (PK), Название, Преподаватель, Часы
3. **Таблица "Оценки"**: ID_записи (PK), ID_студента (FK), ID_предмета (FK), Оценка, Дата

Задания:

1. Создайте проект БД с указанными таблицами.
2. Создайте базу данных "Учебный_отдел.odb".
3. Создайте все три таблицы с указанными полями.
4. Установите связи между таблицами.
5. Заполните таблицы данными (минимум по 5 записей в каждую).
6. Отредактируйте таблицы:
 - Добавьте в "Студенты" поле "Стипендия: DECIMAL(7,2)"
 - Измените размер поля "Преподаватель" на 60 символов
 - Добавьте в "Оценки" поле "Тип_экзамена: VARCHAR(20)"

Вариант 2: "Магазин"

Создайте базу данных для учета товаров и поставщиков.

1. **Таблица "Поставщики"**: ID_поставщика (PK), Название, Адрес, Телефон
2. **Таблица "Товары"**: ID_товара (PK), Название, Категория, Цена, ID_поставщика (FK)
3. **Таблица "Продажи"**: ID_продажи (PK), ID_товара (FK), Количество, Дата_продажи

Задания:

1. Разработайте проект БД.
2. Создайте базу данных "Магазин.odb".
3. Создайте таблицы с указанными полями.
4. Установите необходимые связи.
5. Заполните таблицы тестовыми данными.
6. Модифицируйте структуру:
 - Добавьте в "Товары" поле "Гарантия: INTEGER"
 - Измените тип поля "Цена" на денежный формат
 - Добавьте в "Продажи" поле "Скидка: DECIMAL(5,2)"

Вариант 3: "Поликлиника"

Создайте базу данных для регистрации пациентов и врачей.

1. **Таблица "Пациенты"**: ID_пациента (PK), Фамилия, Имя, Дата_рождения, Адрес
2. **Таблица "Врачи"**: ID_врача (PK), Фамилия, Имя, Специальность, Кабинет

3. **Таблица "Записи"**: ID_записи (PK), ID_пациента (FK), ID_врача (FK), Дата_приема, Диагноз

Задания:

1. Создайте проект БД.
2. Создайте файл БД "Поликлиника.odb".
3. Реализуйте все таблицы в LibreOffice Base.
4. Создайте связи между таблицами.
5. Введите тестовые данные (по 5-7 записей в каждую таблицу).
6. Проведите редактирование:
 - Добавьте в "Пациенты" поле "Полис: VARCHAR(20)"
 - Измените размер поля "Специальность" на 40 символов
 - Добавьте в "Записи" поле "Рекомендации: VARCHAR(200)"

Требования к отчету:

1. Титульный лист с названием работы и данными студента
2. Разработанный проект БД (список таблиц с полями и связями)
3. Скриншоты:
 - Окна созданной базы данных с таблицами
 - Режима конструктора для каждой таблицы
 - Окна связей между таблицами
 - Таблиц с заполненными данными
4. Ответы на контрольные вопросы
5. Выводы о проделанной работе

6. КРИТЕРИИ ОЦЕНКИ

Оценка "5" (Отлично):

- Полностью выполнен проект БД с правильной структурой
- Все таблицы созданы с корректными типами данных и ключами
- Установлены правильные связи между таблицами
- Выполнены все задания по редактированию таблиц
- Таблицы заполнены тестовыми данными
- Работа сдана в срок, отчет оформлен полностью

Оценка "4" (Хорошо):

- Проект БД разработан с незначительными недочетами
- Таблицы созданы, но есть небольшие ошибки в типах данных
- Связи установлены, но не все или с небольшими ошибками
- Выполнены не все задания по редактированию
- Отчет оформлен с незначительными замечаниями

Оценка "3" (Удовлетворительно):

- Проект БД разработан поверхностно
- Созданы не все таблицы или с серьезными ошибками
- Связи не установлены или установлены неправильно
- Не выполнены задания по редактированию
- Отчет оформлен не полностью

Оценка "2" (Неудовлетворительно):

- Проект БД не разработан
- Таблицы не созданы или созданы с критическими ошибками
- Работа не выполнена или сдана не в срок

ПРАКТИЧЕСКАЯ РАБОТА_№5. Редактирование, добавление и удаление записей в таблице. Применение логических условий к записям. Открытие, редактирование и пополнение табличного файла.

по дисциплине «Основы проектирования баз данных»

Тема: Редактирование, добавление и удаление записей в таблице. Применение логических условий к записям

1. ЦЕЛИ И ЗАДАЧИ ПРАКТИЧЕСКОЙ РАБОТЫ

Цель: Освоить основные операции работы с данными в таблицах LibreOffice Base: добавление, редактирование, удаление записей, а также научиться применять фильтры для отбора данных по заданным условиям.

Задачи:

1. Научиться добавлять новые записи в таблицу различными способами.
2. Освоить редактирование существующих записей.
3. Научиться удалять записи из таблицы.
4. Освоить применение фильтров для отбора данных по простым и сложным условиям.
5. Научиться импортировать данные из других файлов.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.1. Основные операции с записями

Добавление записей:

- Через табличный режим (двойной щелчок по таблице)
- Через форму (если она создана)
- С помощью кнопки "Новая запись" в панели навигации
- Импорт из других файлов

Редактирование записей:

- Прямое редактирование в ячейке таблицы
- Редактирование через форму
- Использование команды "Изменить"

Удаление записей:

- Выделение строки и нажатие Delete
- Использование контекстного меню "Удалить запись"
- Удаление через форму

2.2. Фильтрация данных

Фильтр - это инструмент для отбора записей, удовлетворяющих определенным условиям.

Типы фильтров:

- **Простой фильтр** - отбор по одному условию
- **Стандартный фильтр** - отбор по нескольким условиям
- **Расширенный фильтр** - сложные логические условия

Логические операторы:

- **И (AND)** - должны выполняться все условия
- **ИЛИ (OR)** - должно выполняться хотя бы одно условие
- **НЕ (NOT)** - отрицание условия

3. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ

Задача: Работа с таблицей "Сотрудники" компании

Шаг 1. Создание и заполнение таблицы

1. Создайте базу данных "Компания.odt"
2. Создайте таблицу "Сотрудники" со следующими полями:

text

ID_сотрудника (INTEGER, Автозначение: Да) - PK

Фамилия (VARCHAR, 30)

Имя (VARCHAR, 20)
Отдел (VARCHAR, 30)
Должность (VARCHAR, 30)
Оклад (DECIMAL, формат: Денежный)
Дата_приема (DATE)
Стаж (INTEGER)

Шаг 2. Добавление записей

1. Откройте таблицу в табличном режиме
2. Добавьте 10 записей:

text

1. Иванов, Иван, Бухгалтерия, Бухгалтер, 45000, 15.03.2020, 3
2. Петрова, Анна, Отдел кадров, Менеджер, 40000, 10.05.2019, 4
3. Сидоров, Петр, IT-отдел, Программист, 60000, 20.01.2018, 5
4. Кузнецова, Мария, Бухгалтерия, Главный бухгалтер, 55000, 05.11.2017, 6
5. Васильев, Алексей, Отдел продаж, Менеджер, 50000, 12.09.2021, 2
6. Николаева, Елена, IT-отдел, Тестировщик, 48000, 03.07.2020, 3
7. Григорьев, Дмитрий, Отдел продаж, Директор, 70000, 01.02.2016, 7
8. Тихонова, Ольга, Бухгалтерия, Экономист, 42000, 25.04.2022, 1
9. Романов, Сергей, IT-отдел, Системный администратор, 52000, 14.08.2019, 4
10. Захарова, Татьяна, Отдел кадров, Специалист, 38000, 30.10.2021, 2

Шаг 3. Редактирование записей

1. Найдите запись с фамилией "Иванов" и измените оклад на 47000
2. Измените должность "Менеджер" в отделе продаж на "Старший менеджер"
3. Исправьте дату приема для сотрудника "Романов" на 14.08.2020

Шаг 4. Применение фильтров

1. **Простой фильтр:** Показать всех сотрудников отдела "Бухгалтерия"
 - Выберите столбец "Отдел"
 - Данные → Фильтр → Стандартный фильтр
 - Условие: Отдел = "Бухгалтерия"
2. **Фильтр по диапазону:** Показать сотрудников с окладом от 45000 до 55000
 - Условие: Оклад >= 45000 И Оклад <= 55000
3. **Сложный фильтр:** Показать сотрудников IT-отдела со стажем более 3 лет
 - Условие: Отдел = "IT-отдел" И Стаж > 3

Шаг 5. Удаление записей

1. Удалите запись о сотруднике с фамилией "Тихонова"
2. Удалите всех сотрудников со стажем менее 1 года

Шаг 6. Импорт данных

1. Создайте в LibreOffice Calc файл "Новые_сотрудники.ods"
2. Добавьте 3 новых сотрудника:

text

- Смирнов, Андрей, Отдел продаж, Менеджер, 46000, 15.11.2023, 0
Федорова, Ирина, Бухгалтерия, Бухгалтер, 44000, 20.12.2023, 0
Белов, Павел, IT-отдел, Программист, 58000, 10.10.2023, 0
3. Импортируйте данные в таблицу "Сотрудники"

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие способы добавления записей в таблицу вы знаете?
2. Как исправить ошибку в уже внесенной записи?
3. Что произойдет, если удалить запись, на которую ссылаются другие таблицы?
4. Для чего нужны фильтры в базе данных?
5. Как отфильтровать записи по нескольким условиям одновременно?
6. В чем разница между операторами "И" и "ИЛИ" в фильтрах?

7. Как отменить действие фильтра и показать все записи?
 8. Можно ли импортировать данные из Excel в LibreOffice Base?
 9. Как сохранить отфильтрованные данные в отдельный файл?
 10. Что делать, если случайно удалили нужную запись?
-

5. ЗАДАНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ

Вариант 1: База данных "Учебный центр"

Создайте базу данных для учета слушателей курсов.

1. Таблица "Курсы":

text

ID_курса, Название_курса, Преподаватель, Стоимость, Длительность, Начало_курса

2. Таблица "Слушатели":

text

ID_слушателя, Фамилия, Имя, Курс, Дата_регистрации, Оплачено

Задания:

1. Добавьте 8 курсов и 15 слушателей
2. Отредактируйте: увеличьте стоимость всех курсов на 10%
3. Примените фильтры:
 - Показать курсы стоимостью более 20000
 - Показать слушателей, записавшихся в октябре 2023
 - Показать курсы, начинающиеся в ноябре-декабре 2023
4. Удалите курсы с длительностью менее 20 часов
5. Импортируйте 3 новых слушателя из файла Calc

Вариант 2: База данных "Автопарк"

Создайте базу данных для учета автомобилей компании.

1. Таблица "Автомобили":

text

ID_авто, Марка, Модель, Год_выпуска, Пробег, Статус, Стоимость_аренды

2. Таблица "Водители":

text

ID_водителя, Фамилия, Имя, Категория, Стаж, Зарплата

Задания:

1. Добавьте 10 автомобилей и 8 водителей
2. Отредактируйте: увеличьте зарплату водителям со стажем более 5 лет
3. Примените фильтры:
 - Показать автомобили марки "Toyota"
 - Показать водителей с категорией "С" и стажем более 3 лет
 - Показать автомобили 2020 года выпуска и новее
4. Удалите автомобили с пробегом более 200000 км
5. Импортируйте 2 новых автомобиля из файла Calc

Вариант 3: База данных "Библиотека"

Создайте базу данных для учета книг в библиотеке.

1. Таблица "Книги":

text

ID_книги, Название, Автор, Жанр, Год_издания, Цена, Наличие

2. Таблица "Читатели":

text

ID_читателя, Фамилия, Имя, Группа, Дата_регистрации, Книг_на_руках

Задания:

1. Добавьте 12 книг и 10 читателей
2. Отредактируйте: измените статус наличия для книг дороже 1000 рублей
3. Примените фильтры:

- Показать книги жанра "Фантастика"
- Показать читателей группы "ИСП-201"
- Показать книги изданные после 2010 года
- 4. Удалите читателей, зарегистрированных до 2020 года
- 5. Импортируйте 3 новые книги из файла Calc

Общие требования:

1. Создайте базу данных согласно вашему варианту
2. Выполните все операции с записями
3. Примените не менее 3 различных фильтров
4. Выполните импорт данных из внешнего файла
5. Сохраните результаты работы

Отчет должен содержать:

1. Титульный лист
2. Структуру созданных таблиц
3. Скриншоты:
 - Таблиц с данными
 - Примененных фильтров
 - Результатов фильтрации
 - Импортированных данных
4. Ответы на контрольные вопросы
5. Выводы о проделанной работе

6. КРИТЕРИИ ОЦЕНКИ

Оценка "5" (Отлично):

- Полностью созданы все таблицы согласно варианту
- Добавлено необходимое количество записей (100%)
- Корректно выполнены все операции редактирования
- Применены все требуемые фильтры с правильными условиями
- Успешно выполнен импорт данных из внешнего файла
- Работа сдана в срок, отчет оформлен полностью

Оценка "4" (Хорошо):

- Созданы все таблицы, но с незначительными ошибками
- Добавлено 80% требуемых записей
- Выполнены основные операции редактирования
- Применены не все фильтры или с небольшими ошибками
- Импорт выполнен с незначительными проблемами
- Отчет оформлен с небольшими замечаниями

Оценка "3" (Удовлетворительно):

- Созданы не все таблицы
- Добавлено менее 60% записей
- Операции редактирования выполнены с ошибками
- Применены только простые фильтры
- Импорт не выполнен или выполнен некорректно
- Отчет оформлен не полностью

Оценка "2" (Неудовлетворительно):

- Таблицы не созданы
- Записи не добавлены
- Фильтры не применены
- Работа не выполнена или сдана не в срок

ПРАКТИЧЕСКАЯ РАБОТА_№6. Создание ключевых полей. Задание индексов. Установление и удаление связей между таблицами.
по дисциплине «Основы проектирования баз данных»
Тема: Создание ключевых полей. Задание индексов. Установление и удаление связей между таблицами

1. ЦЕЛИ И ЗАДАЧИ ПРАКТИЧЕСКОЙ РАБОТЫ

Цель: Освоить создание и настройку ключевых полей и индексов в таблицах, научиться устанавливать связи между таблицами для обеспечения целостности данных.

Задачи:

1. Научиться создавать первичные и составные ключи в таблицах.
 2. Освоить создание индексов для ускорения поиска данных.
 3. Научиться устанавливать связи «один-ко-многим» между таблицами.
 4. Освоить настройку параметров целостности данных при связывании таблиц.
 5. Научиться удалять и изменять существующие связи.
-

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.1. Ключевые поля

Первичный ключ (Primary Key, PK) – поле или набор полей, которые однозначно идентифицируют каждую запись в таблице. Требования:

- Уникальность значений
- Отсутствие NULL-значений
- В таблице может быть только один первичный ключ

Составной ключ – первичный ключ, состоящий из двух или более полей.

Внешний ключ (Foreign Key, FK) – поле в одной таблице, которое ссылается на первичный ключ в другой таблице.

2.2. Индексы

Индекс – структура данных, ускоряющая поиск и сортировку записей в таблице. Индексы создаются по одному или нескольким полям.

Типы индексов:

- **Уникальный индекс** – гарантирует уникальность значений
- **Неуникальный индекс** – допускает повторяющиеся значения
- **Составной индекс** – создается по нескольким полям

2.3. Связи между таблицами

Связь «один-ко-многим» (1:M) – самая распространенная связь, когда одной записи в главной таблице соответствует несколько записей в подчиненной.

Целостность данных – набор правил, обеспечивающих корректность данных при их изменении:

- **Каскадное обновление** – при изменении PK в главной таблице автоматически обновляются соответствующие FK в подчиненной
 - **Каскадное удаление** – при удалении записи в главной таблице автоматически удаляются связанные записи в подчиненной
 - **Ограничение** – запрет на изменение или удаление записей, на которые есть ссылки
-

3. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ

Задача: Создать связанные таблицы для учета студентов и их оценок

Шаг 1. Создание таблицы «Студенты» с первичным ключом

1. Создайте базу данных «Учебный_процесс.odb»
2. Создайте таблицу «Студенты»:

text

Поля:

- ID_студента: INTEGER (Автозначение: Да)

- Фамилия: VARCHAR(30)

- Имя: VARCHAR(20)

- Группа: VARCHAR(10)

- Дата_рождения: DATE

3. Установите первичный ключ:

- Откройте таблицу в режиме конструктора
- Щелкните правой кнопкой на поле ID_студента
- Выберите «Первичный ключ»
- Появится значок ключа слева от поля

Шаг 2. Создание индекса

1. Для ускорения поиска по фамилии создадим индекс:

- В режиме конструктора нажмите кнопку «Индексы» на панели инструментов
- В диалоговом окне нажмите «Создать»
- Введите имя индекса: «Индекс_Фамилия»
- Выберите поле «Фамилия»
- Нажмите «Создать»

Шаг 3. Создание таблицы «Оценки» с внешним ключом

1. Создайте таблицу «Оценки»:

text

Поля:

- ID_записи: INTEGER (Автозначение: Да) - PK

- ID_студента: INTEGER - FK

- Предмет: VARCHAR(40)

- Оценка: INTEGER

- Дата_сдачи: DATE

Шаг 4. Установление связи между таблицами

1. Закройте все открытые таблицы

2. В главном окне Base выберите «Сервис» → «Связи»

3. Добавьте обе таблицы в окно связей

4. Установите связь:

- Перетащите поле ID_студента из таблицы «Студенты» на поле ID_студента в таблице «Оценки»
- В диалоговом окне «Создание связи»:
 - Установите флажок «Обновлять каскадно»
 - Установите флажок «Удалять каскадно»
 - Нажмите «Создать»

5. На схеме появится связь с обозначением «1:n»

Шаг 5. Проверка работы связи

1. Добавьте в таблицу «Студенты» 3 записи:

text

1. Иванов, Иван, ИСП-101, 15.05.2004

2. Петрова, Анна, ИСП-102, 20.03.2005

3. Сидоров, Петр, ИСП-101, 10.11.2004

2. Добавьте в таблицу «Оценки» оценки для студентов:

text

1. ID_студента=1, Математика, 5, 10.10.2023

2. ID_студента=1, Физика, 4, 12.10.2023

3. ID_студента=2, Математика, 3, 10.10.2023

3. Попробуйте добавить оценку для несуществующего студента (ID_студента=10) – система должна запретить это

Шаг 6. Удаление связи

1. Вернитесь в окно связей («Сервис» → «Связи»)

2. Щелкните правой кнопкой на линии связи
3. Выберите «Удалить»
4. Подтвердите удаление

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое первичный ключ и каковы его основные свойства?
2. В чем отличие первичного ключа от уникального индекса?
3. Что такое внешний ключ и для чего он используется?
4. Какие типы связей между таблицами вы знаете?
5. Что такое каскадное обновление и каскадное удаление?
6. Как создать составной первичный ключ?
7. Для чего нужны индексы в базе данных?
8. Что произойдет, если попытаться удалить запись, на которую ссылаются другие таблицы?
9. Можно ли изменить тип связи после ее создания?
10. Как проверить, правильно ли установлена связь между таблицами?

5. ЗАДАНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ

Вариант 1: Библиотека

Создайте базу данных для учета книг и читателей.

1. **Таблица «Книги»:**
 - ID_книги (PK, INTEGER, Автозначение)
 - Название (VARCHAR, 100)
 - Автор (VARCHAR, 50)
 - Год_издания (INTEGER)
 - Создать индекс по полю «Автор»
2. **Таблица «Выдача»:**
 - ID_выдачи (PK, INTEGER, Автозначение)
 - ID_книги (FK, INTEGER)
 - ID_читателя (FK, INTEGER)
 - Дата_выдачи (DATE)
 - Дата_возврата (DATE)
3. **Таблица «Читатели»:**
 - ID_читателя (PK, INTEGER, Автозначение)
 - Фамилия (VARCHAR, 30)
 - Имя (VARCHAR, 20)
 - Группа (VARCHAR, 10)
 - Создать составной индекс по полям «Фамилия, Имя»

Задания:

1. Создайте все три таблицы с указанными полями и ключами
2. Установите связи:
 - «Книги» → «Выдача» (1:M)
 - «Читатели» → «Выдача» (1:M)
3. Настройте каскадное обновление для обеих связей
4. Добавьте тестовые данные (по 5 записей в каждую таблицу)
5. Проверьте целостность данных:
 - Попробуйте удалить книгу, которая есть в выдаче
 - Попробуйте добавить выдачу для несуществующей книги

Вариант 2: Магазины

Создайте базу данных для учета товаров и заказов.

1. **Таблица «Товары»:**
 - ID_товара (PK, INTEGER, Автозначение)

- Наименование (VARCHAR, 100)
- Категория (VARCHAR, 30)
- Цена (DECIMAL)
- Количество (INTEGER)
- Создать индекс по полю «Категория»
- 2. **Таблица «Заказы»:**
 - ID_заказа (PK, INTEGER, Автозначение)
 - Дата_заказа (DATE)
 - Клиент (VARCHAR, 50)
- 3. **Таблица «Позиции_заказов»:**
 - ID_позиции (PK, INTEGER, Автозначение)
 - ID_заказа (FK, INTEGER)
 - ID_товара (FK, INTEGER)
 - Количество (INTEGER)
 - Цена (DECIMAL)

Задания:

1. Создайте все три таблицы
2. Установите связи:
 - «Товары» → «Позиции_заказов» (1:M)
 - «Заказы» → «Позиции_заказов» (1:M)
3. Настройте каскадное удаление для связи «Заказы-Позиции»
4. Добавьте тестовые данные
5. Создайте составной первичный ключ для таблицы «Позиции_заказов» из полей ID_заказа и ID_товара

Вариант 3: Поликлиника

Создайте базу данных для регистрации пациентов и врачей.

1. **Таблица «Врачи»:**
 - ID_врача (PK, INTEGER, Автозначение)
 - Фамилия (VARCHAR, 30)
 - Имя (VARCHAR, 20)
 - Специальность (VARCHAR, 40)
 - Кабинет (INTEGER)
 - Создать индекс по полю «Специальность»
2. **Таблица «Пациенты»:**
 - ID_пациента (PK, INTEGER, Автозначение)
 - Фамилия (VARCHAR, 30)
 - Имя (VARCHAR, 20)
 - Дата_рождения (DATE)
 - Полис (VARCHAR, 20)
 - Создать уникальный индекс по полю «Полис»
3. **Таблица «Записи»:**
 - ID_записи (PK, INTEGER, Автозначение)
 - ID_врача (FK, INTEGER)
 - ID_пациента (FK, INTEGER)
 - Дата_приема (DATE)
 - Время_приема (VARCHAR, 5)
 - Диагноз (VARCHAR, 200)

Задания:

1. Создайте все три таблицы
2. Установите связи:
 - «Врачи» → «Записи» (1:M)
 - «Пациенты» → «Записи» (1:M)

3. Настройте ограничение удаления для обеих связей
4. Добавьте тестовые данные
5. Удалите одну из связей и создайте ее заново с другими параметрами

Общие требования:

1. Для своего варианта создайте все указанные таблицы
2. Создайте все необходимые первичные ключи
3. Создайте не менее 2 индексов
4. Установите все требуемые связи
5. Настройте параметры целостности данных
6. Проверьте работу связей

Отчет должен содержать:

1. Титульный лист
 2. Структуру всех таблиц с указанием ключей
 3. Скриншоты:
 - Таблиц в режиме конструктора с выделенными ключами
 - Диалогового окна создания индексов
 - Окна связей между таблицами
 - Проверки работы целостности данных
 4. Ответы на контрольные вопросы
 5. Выводы о проделанной работе
-

6. КРИТЕРИИ ОЦЕНКИ

Оценка «5» (Отлично):

- Созданы все таблицы с правильными первичными ключами
- Созданы все требуемые индексы
- Установлены все связи с правильными параметрами
- Настроена целостность данных (каскадное обновление/удаление)
- Проведена проверка работы связей
- Работа сдана в срок, отчет оформлен полностью

Оценка «4» (Хорошо):

- Созданы все таблицы, но есть небольшие ошибки в типах данных
- Созданы не все индексы
- Связи установлены, но с неоптимальными параметрами
- Проверка работы связей проведена частично
- Отчет оформлен с небольшими замечаниями

Оценка «3» (Удовлетворительно):

- Созданы не все таблицы
- Ключи созданы с ошибками
- Связи установлены некорректно или не все
- Индексы не созданы
- Отчет оформлен не полностью

Оценка «2» (Неудовлетворительно):

- Таблицы не созданы
- Ключи не созданы
- Связи не установлены
- Работа не выполнена или сдана не в срок

ПРАКТИЧЕСКАЯ РАБОТА_№7. Проведение сортировки и фильтрации данных. Поиск данных по одному и нескольким полям. Поиск данных в таблице.
по дисциплине «Основы проектирования баз данных»
Тема: Проведение сортировки и фильтрации данных. Поиск данных по одному и нескольким полям

1. ЦЕЛИ И ЗАДАЧИ ПРАКТИЧЕСКОЙ РАБОТЫ

Цель: Освоить методы сортировки, фильтрации и поиска данных в таблицах LibreOffice Base для эффективной работы с информацией.

Задачи:

1. Научиться сортировать данные по одному и нескольким полям.
 2. Освоить применение простых и сложных фильтров.
 3. Научиться использовать стандартные и расширенные фильтры.
 4. Освоить различные методы поиска данных в таблице.
 5. Научиться сохранять и применять настройки фильтров.
-

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.1. Сортировка данных

Сортировка – упорядочивание записей таблицы по возрастанию или убыванию значений в одном или нескольких столбцах.

Виды сортировки:

- **Простая сортировка** – по одному полю
- **Сложная сортировка** – по нескольким полям с определением приоритета

Порядок сортировки:

- **По возрастанию (А-Я, 0-9)**
- **По убыванию (Я-А, 9-0)**

2.2. Фильтрация данных

Фильтр – инструмент для отбора записей, удовлетворяющих заданным условиям.

Типы фильтров:

- **Автофильтр** – быстрый фильтр по значению в столбце
- **Стандартный фильтр** – фильтр с заданием условий
- **Расширенный фильтр** – сложные логические условия

Логические операторы в фильтрах:

- **=** равно
- **<>** не равно
- **<** меньше
- **>** больше
- **<=** меньше или равно
- **>=** больше или равно
- **И (AND)** – все условия должны выполняться
- **ИЛИ (OR)** – хотя бы одно условие должно выполняться

2.3. Поиск данных

Поиск – нахождение записей, содержащих определенное значение.

Способы поиска:

- **Простой поиск** – поиск по одному полю
 - **Расширенный поиск** – поиск по нескольким полям
 - **Поиск с подстановочными знаками:**
 - ***** – любое количество любых символов
 - **?** – один любой символ
-

3. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ

Задача: Работа с таблицей «Сотрудники»

Шаг 1. Подготовка данных

1. Создайте базу данных «Кадры.odb»

2. Создайте таблицу «Сотрудники»:

text

ID, Фамилия, Имя, Должность, Отдел, Оклад, Дата_приема, Стаж

3. Добавьте 12 записей:

text

1, Иванов, Иван, Бухгалтер, Бухгалтерия, 45000, 15.03.2020, 3

2, Петрова, Анна, Менеджер, Отдел кадров, 40000, 10.05.2019, 4

3, Сидоров, Петр, Программист, IT-отдел, 60000, 20.01.2018, 5

4, Кузнецова, Мария, Главный бухгалтер, Бухгалтерия, 55000, 05.11.2017, 6

5, Васильев, Алексей, Менеджер, Отдел продаж, 50000, 12.09.2021, 2

6, Николаева, Елена, Тестировщик, IT-отдел, 48000, 03.07.2020, 3

7, Григорьев, Дмитрий, Директор, Отдел продаж, 70000, 01.02.2016, 7

8, Тихонова, Ольга, Экономист, Бухгалтерия, 42000, 25.04.2022, 1

9, Романов, Сергей, Системный администратор, IT-отдел, 52000, 14.08.2019, 4

10, Захарова, Татьяна, Специалист, Отдел кадров, 38000, 30.10.2021, 2

11, Смирнов, Андрей, Менеджер, Отдел продаж, 46000, 15.11.2023, 0

12, Федорова, Ирина, Бухгалтер, Бухгалтерия, 44000, 20.12.2023, 0

Шаг 2. Сортировка данных

1. **Сортировка по одному полю:**

- Откройте таблицу
- Щелкните правой кнопкой на заголовке столбца «Фамилия»
- Выберите «Сортировать по возрастанию»
- Записи отсортируются по фамилии А-Я

2. **Сортировка по нескольким полям:**

- Данные → Сортировка
- В диалоговом окне:
 - Первый уровень: «Отдел» по возрастанию
 - Второй уровень: «Фамилия» по возрастанию
- Сотрудники отсортируются по отделам, а внутри отдела по фамилиям

Шаг 3. Фильтрация данных

1. **Автофильтр:**

- Выделите любую ячейку в столбце «Отдел»
- Данные → Фильтр → Автофильтр
- Нажмите на стрелку в заголовке столбца
- Выберите «Бухгалтерия» – останутся только сотрудники этого отдела

2. **Стандартный фильтр:**

- Данные → Фильтр → Стандартный фильтр
- Условие 1: Оклад $\geq$ 50000
- Условие 2: Стаж $\geq$ 3
- Оператор: И
- Будут показаны сотрудники с окладом от 50000 и стажем от 3 лет

3. **Расширенный фильтр:**

- Данные → Фильтр → Расширенный фильтр
- Отдел = «IT-отдел» ИЛИ Отдел = «Отдел продаж»
- Будут показаны сотрудники IT-отдела или отдела продаж

Шаг 4. Поиск данных

1. **Простой поиск:**
 - Правка → Найти
 - В поле «Найти» введите «бухгалтер»
 - Нажмите «Найти далее»
 - Будет найдена первая запись, содержащая это слово
 2. **Поиск с подстановочными знаками:**
 - В поле «Найти» введите «Пет*»
 - Найдутся все фамилии, начинающиеся на «Пет» (Петрова, Петров, и т.д.)
 3. **Расширенный поиск:**
 - В диалоговом окне поиска нажмите «Другие параметры»
 - Установите «Учитывать регистр»
 - Выберите «Искать во всех полях»
 - Введите «менеджер» – найдутся все менеджеры
- Шаг 5. Сохранение фильтра
1. Примените нужный фильтр
 2. Вид → Фильтр → Сохранить фильтр как
 3. Введите имя фильтра: «Высокооплачиваемые»
 4. Для применения сохраненного фильтра:
 - Вид → Фильтр → Выбрать фильтр
 - Выберите сохраненный фильтр

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем разница между сортировкой и фильтрацией?
2. Как отсортировать данные по нескольким полям?
3. Какие логические операторы можно использовать в фильтрах?
4. Как создать фильтр, отбирающий записи за определенный период?
5. Что такое автофильтр и как его использовать?
6. Как искать данные с использованием подстановочных знаков?
7. Можно ли сохранить настройки фильтра для повторного использования?
8. Как отменить действие фильтра и показать все записи?
9. Что такое расширенный поиск и когда его использовать?
10. Как найти все записи, содержащие определенное слово в любом поле?

5. ЗАДАНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ

Вариант 1: База данных «Студенты»

Создайте таблицу для учета успеваемости студентов.

1. **Таблица «Успеваемость»:**

text

ID, Фамилия, Имя, Группа, Предмет, Оценка, Дата_сдачи, Преподаватель

2. **Добавьте 15 записей:**

- Разные группы (ИСП-101, ИСП-102, ПКС-201)
- Разные предметы (Математика, Физика, Информатика, Базы данных)
- Оценки от 2 до 5
- Даты сдачи в течение 2023 года

Задания:

1. **Сортировка:**

- Отсортируйте по фамилии А-Я
- Отсортируйте по группе, а внутри группы по фамилии
- Отсортируйте по оценке по убыванию

2. **Фильтрация:**

- Покажите только студентов группы ИСП-101

- Покажите оценки по предмету «Базы данных»
- Покажите оценки 4 и 5
- Покажите сдачи за октябрь 2023
- 3. **Поиск:**
 - Найдите всех студентов с фамилией на «И»
 - Найдите все оценки преподавателя «Иванова»
 - Найдите сдачи за ноябрь 2023
- 4. **Сложные операции:**
 - Покажите студентов ИСП-102 с оценками 4 и 5 по математике
 - Отсортируйте по дате сдачи и покажите последние 5 записей

Вариант 2: База данных «Товары»

Создайте таблицу для учета товаров в магазине.

1. **Таблица «Товары»:**

text

Код_товара, Наименование, Категория, Цена, Количество, Поставщик, Дата_поступления

2. **Добавьте 15 записей:**

- Разные категории (Электроника, Одежда, Канцтовары, Продукты)
- Цены от 100 до 50000 рублей
- Разные поставщики
- Даты поступления в 2023 году

Задания:

1. **Сортировка:**

- Отсортируйте по цене по возрастанию
- Отсортируйте по категории, а внутри категории по наименованию
- Отсортируйте по дате поступления по убыванию

2. **Фильтрация:**

- Покажите товары категории «Электроника»
- Покажите товары дороже 10000 рублей
- Покажите товары с количеством менее 10
- Покажите товары, поступившие в 1 квартале 2023

3. **Поиск:**

- Найдите все товары с названием, содержащим «телефон»
- Найдите товары поставщика «ООО Техно»
- Найдите товары, поступившие в сентябре 2023

4. **Сложные операции:**

- Покажите канцтовары стоимостью до 500 рублей
- Отсортируйте по поставщику и покажите товары с остатком менее 5

Вариант 3: База данных «Пациенты»

Создайте таблицу для регистрации пациентов поликлиники.

1. **Таблица «Пациенты»:**

text

Номер_карты, Фамилия, Имя, Возраст, Диагноз, Врач, Дата_посещения, Температура

2. **Добавьте 15 записей:**

- Возраст от 18 до 70 лет
- Разные диагнозы (ОРВИ, Гипертония, Гастрит, Аллергия)
- Разные врачи (терапевты, специалисты)
- Даты посещений в 2023 году
- Температура от 36.0 до 39.5

Задания:

1. **Сортировка:**

- Отсортируйте по фамилии
- Отсортируйте по возрасту по убыванию

- Отсортируйте по врачу, а внутри по дате посещения
 - 2. **Фильтрация:**
 - Покажите пациентов с диагнозом «ОРВИ»
 - Покажите пациентов старше 50 лет
 - Покажите посещения врача «Иванова А.И.»
 - Покажите пациентов с температурой выше 37.5
 - 3. **Поиск:**
 - Найдите всех пациентов с фамилией на «П»
 - Найдите посещения в октябре 2023
 - Найдите пациентов с диагнозом, содержащим «ит»
 - 4. **Сложные операции:**
 - Покажите пациентов младше 30 лет с температурой выше 38
 - Отсортируйте по дате и покажите последние 3 посещения
- Общие требования:**
1. Создайте базу данных согласно вашему варианту
 2. Добавьте не менее 15 записей
 3. Выполните все указанные операции сортировки
 4. Примените не менее 4 различных фильтров
 5. Выполните все виды поиска
 6. Сохраните один фильтр для повторного использования
- Отчет должен содержать:**
1. Титульный лист
 2. Структуру созданной таблицы
 3. Скриншоты:
 - Таблицы с исходными данными
 - Результаты сортировки
 - Диалоговых окон фильтров
 - Результаты применения фильтров
 - Диалогового окна поиска
 4. Ответы на контрольные вопросы
 5. Выводы о проделанной работе

6. КРИТЕРИИ ОЦЕНКИ

Оценка «5» (Отлично):

- Создана таблица с полной структурой
- Добавлено 15+ записей с разнообразными данными
- Корректно выполнены все виды сортировки
- Применены все требуемые фильтры с правильными условиями
- Выполнены все виды поиска
- Сохранен фильтр для повторного использования
- Работа сдана в срок, отчет оформлен полностью

Оценка «4» (Хорошо):

- Создана таблица с незначительными ошибками в структуре
- Добавлено 10-14 записей
- Выполнены основные виды сортировки
- Применены не все фильтры или с небольшими ошибками
- Поиск выполнен частично
- Отчет оформлен с небольшими замечаниями

Оценка «3» (Удовлетворительно):

- Таблица создана с ошибками в структуре
- Добавлено менее 10 записей
- Выполнена только простая сортировка

- Применены 1-2 простых фильтра
- Поиск не выполнен или выполнен некорректно
- Отчет оформлен не полностью

Оценка «2» (Неудовлетворительно):

- Таблица не создана
- Записи не добавлены
- Операции не выполнены
- Работа не сдана или сдана не в срок

ПРАКТИЧЕСКАЯ РАБОТА №8. Работа с переменными.

по дисциплине «Основы проектирования баз данных»

Тема: Работа с переменными. Создание и выполнение запросов в LibreOffice Base. Построитель запросов.

1. ЦЕЛИ И ЗАДАЧИ ПРАКТИЧЕСКОЙ РАБОТЫ

Цель: Научиться создавать, настраивать и выполнять различные запросы для выборки и анализа данных с помощью графического конструктора запросов в LibreOffice Base.

Задачи:

1. Научиться создавать простые запросы на выборку данных из одной таблицы.
2. Освоить добавление условий отбора (фильтрации) в запросы.
3. Научиться сортировать результаты запроса.
4. Освоить создание запросов на основе нескольких связанных таблиц.
5. Научиться выполнять расчеты в запросах (суммы, средние значения).

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Запрос — это объект базы данных, который позволяет получить нужные данные из одной или нескольких таблиц в удобном виде, не изменяя сами таблицы.

Конструктор запросов (Построитель запросов) — это графический инструмент в LibreOffice Base, который позволяет создавать запросы без написания кода на SQL, путем перетаскивания полей и настройки условий.

Основные элементы конструктора запросов:

- **Панель таблиц (диаграмма данных):** Здесь отображаются таблицы, из которых выбираются данные.
- **Сетка запроса (бланк запроса по образцу):** Сюда добавляются поля для отображения в результатах.
- **Параметры для каждого поля:**
 - **Поле:** Выбранное поле таблицы.
 - **Таблица:** Из какой таблицы взято поле.
 - **Сортировка:** Порядок сортировки (по возрастанию, по убыванию, нет).
 - **Показать:** Отображать ли это поле в результатах (галочка).
 - **Условие отбора:** Критерий для фильтрации записей (например, ="Бухгалтерия" или >50000).

3. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ

Задача: Создать базу данных «Кадры» с таблицей «Сотрудники» и выполнить несколько запросов.

Шаг 1. Подготовка данных

1. Создайте базу данных Кадры.odb.
2. Создайте таблицу «Сотрудники» в режиме конструктора:

ID (INTEGER, Автозначение: Да, Первичный ключ)

Фамилия (VARCHAR, 30)

Имя (VARCHAR, 20)

Отдел (VARCHAR, 40)

Должность (VARCHAR, 30)

Оклад (DECIMAL, Формат: Денежный)

Дата_приема (DATE)

3. Заполните таблицу данными (6-8 записей). Пример:

- Иванов Иван, Бухгалтерия, Бухгалтер, 45000, 15.03.2020
- Петрова Анна, Отдел кадров, Менеджер, 40000, 10.05.2019
- Сидоров Петр, IT-отдел, Программист, 60000, 20.01.2018
- Кузнецова Мария, Бухгалтерия, Главный бухгалтер, 55000, 05.11.2017
- Васильев Алексей, Отдел продаж, Менеджер, 50000, 12.09.2021
- Николаева Елена, IT-отдел, Тестировщик, 48000, 03.07.2020

Шаг 2. Создание простого запроса на выборку

Цель: Вывести список всех сотрудников с их отделами и должностями.

1. В окне базы данных перейдите в раздел «Запросы».
2. В области «Задачи» нажмите «Создать запрос в режиме дизайна».
3. В диалоговом окне «Добавить таблицу» выберите таблицу «Сотрудники» и нажмите «Добавить», затем закройте окно.
4. В верхней части конструктора появится таблица. В нижней части — сетка для настройки.
5. Дважды щелкните по следующим полям в таблице сверху, чтобы добавить их в сетку: **Фамилия**, **Имя**, **Отдел**, **Должность**.
6. В столбце «Сортировка» для поля **Фамилия** выберите «По возрастанию».
7. Нажмите кнопку «Выполнить запрос» (зеленая стрелка "►" на панели инструментов).
8. Сохраните запрос под именем **Список_сотрудников**.

Шаг 3. Запрос с условием отбора

Цель: Найти всех сотрудников отдела «Бухгалтерия».

1. Создайте новый запрос (или скопируйте предыдущий).
2. Добавьте таблицу «Сотрудники».
3. Добавьте в сетку поля: **Фамилия**, **Имя**, **Отдел**, **Оклад**.
4. В строке «Условие отбора» для поля **Отдел** введите: «Бухгалтерия».
5. Выполните запрос.
6. Сохраните его как **Сотрудники_Бухгалтерии**.

Шаг 4. Запрос с вычисляемым полем и сортировкой

Цель: Вывести сотрудников с окладом выше 45000, отсортированных по убыванию оклада.

1. Создайте новый запрос.
2. Добавьте таблицу «Сотрудники».
3. Добавьте поля: **Фамилия**, **Имя**, **Должность**, **Оклад**.
4. В строке «Условие отбора» для поля **Оклад** введите: >45000.
5. В столбце «Сортировка» для поля **Оклад** выберите «По убыванию».
6. Выполните и сохраните запрос как **Высокооплачиваемые_сотрудники**.

Шаг 5. Запрос на основе двух таблиц (если есть связи)

Если у вас есть связанные таблицы (например, «Сотрудники» и «Проекты»), вы можете создать запрос, объединяющий данные из них. Конструктор автоматически отобразит связь между таблицами.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для чего в базе данных используются запросы?
2. Как запустить конструктор запросов в LibreOffice Base?
3. Как добавить поле из таблицы в запрос?
4. Где и как задать условие для отбора записей (например, "отдел = IT")?
5. Как отсортировать результаты запроса по фамилии от А до Я?
6. Что означает галочка в столбце «Показать»?
7. Можно ли в одном запросе использовать поля из разных таблиц? Что для этого нужно?
8. Чем отличается работа с данными через запрос от работы напрямую с таблицей?
9. Как создать копию существующего запроса для его доработки?
10. Для чего нужна кнопка «Выполнить запрос»?

5. ЗАДАНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ

Общее для всех вариантов: Создайте базу данных по своему варианту, заполните таблицы не менее чем 8 записями и выполните запросы с помощью **Конструктора запросов** (Построителя запросов).

Вариант 1: Библиотека

1. Таблица

«Книги»: ID_книги, Название, Автор, Жанр, Год_издания, Цена, Наличие (логическое).

2. Создайте запросы:

- «Все книги»: Вывести название, автора и жанр всех книг, отсортированных по названию.
- «Детективы»: Вывести все книги жанра «Детектив».
- «Новые книги»: Вывести книги, изданные после 2015 года, отсортировать по году издания (сначала новые).
- «В наличии»: Вывести названия и цены только тех книг, которые есть в наличии (Наличие = Да).

Вариант 2: Студенческая группа

1. Таблица «Студенты»: ID, Фамилия, Имя, Дата_рождения, Группа, Средний_балл.

2. Создайте запросы:

- «Список группы»: Вывести фамилию, имя и группу всех студентов, отсортировать по фамилии.
- «Отличники»: Вывести фамилию и имя студентов со средним баллом больше или равным 4.5.
- «Именинники»: Вывести студентов, у которых день рождения в определенном месяце (например, используйте условие $\text{Месяц(Дата_рождения)}=3$ для марта). Подсказка: в условии можно использовать функцию MONTH().
- «Средний балл по группам»: Создать запрос, который выводит название группы и средний балл по этой группе (используйте итоговую функцию СРЗНАЧ в столбце Средний_балл и настройте Группировку по полю Группа).

Вариант 3: Склад товаров

1. Таблица

«Товары»: Артикул, Наименование, Категория, Цена, Количество_на_складе, Поставщик.

2. Создайте запросы:

- «Весь склад»: Вывести наименование, категорию и количество всех товаров.
- «Товары категории»: Вывести товары конкретной категории (например, «Электроника»), отсортированные по цене (от дешевых).
- «Необходимо заказать»: Вывести наименование и количество товаров, которых на складе меньше 5 штук.
- «Общая стоимость остатка»: Для каждой категории товаров подсчитать общую стоимость остатка на складе (это $\text{Цена} * \text{Количество}$). Подсказка: в свободный столбец сетки запроса введите выражение: $[\text{Товары}].[Цена] * [\text{Товары}].[Количество_на_складе]$. В свойствах поля задайте ему имя, например, Стоимость_остатка. Включите группировку по Категория и итоговую функцию СУММ для нового поля.

Требования к отчету:

1. Титульный лист.
 2. Название базы данных и структура таблицы вашего варианта.
 3. Для каждого созданного запроса (4 шт.):
 - Название запроса и его цель.
 - Скриншот конструктора запроса с заполненной сеткой условий.
 - Скриншот результата выполнения запроса (таблицы с данными).
 4. Ответы на контрольные вопросы.
 5. Вывод: что нового узнали, какие были трудности.
-

6. КРИТЕРИИ ОЦЕНКИ

Оценка «5» (Отлично):

- База данных создана корректно, таблица заполнена 8+ записями.
- Созданы **все 4 запроса** с помощью конструктора.
- Запросы работают верно, условия и сортировка заданы правильно.
- В запросах использованы разные возможности: сортировка, условия отбора, итоговые функции.
- Отчет содержит все требуемые скриншоты и оформлен аккуратно.

Оценка «4» (Хорошо):

- База данных создана, но есть незначительные ошибки в типах данных.
- Созданы 3-4 запроса, но в 1-2 есть небольшие ошибки в условиях.
- Отчет оформлен, но скриншоты или пояснения неполные.

Оценка «3» (Удовлетворительно):

- База данных создана с ошибками.
- Создано только 1-2 простых запроса.
- Отчет оформлен не полностью.

Оценка «2» (Неудовлетворительно):

ПРАКТИЧЕСКАЯ РАБОТА_№9. Написание программного файла и работа с табличными файлами. Заполнение массива из табличного файла. Заполнение табличного файла из массива.

по дисциплине «Основы проектирования баз данных»

Тема: Импорт и экспорт данных. Работа с табличными файлами

1. ЦЕЛИ И ЗАДАЧИ ПРАКТИЧЕСКОЙ РАБОТЫ

Цель: Научиться импортировать данные из внешних файлов в базу данных и экспортировать данные из базы данных во внешние файлы.

Задачи:

1. Научиться импортировать данные из табличных файлов (CSV, Calc) в таблицы LibreOffice Base.
 2. Освоить экспорт данных из таблиц LibreOffice Base в табличные файлы.
 3. Научиться корректировать структуру данных при импорте.
 4. Освоить работу с различными форматами файлов.
 5. Научиться решать проблемы совместимости форматов данных.
-

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.1. Форматы данных

Форматы для импорта/экспорта:

- **CSV (Comma Separated Values)** – текстовый формат, где данные разделены запятыми или другими разделителями
- **ODS** – формат электронных таблиц LibreOffice Calc
- **XLS/XLSX** – форматы Microsoft Excel
- **TXT** – простой текстовый файл

2.2. Процесс импорта

Импорт – загрузка данных из внешнего файла в таблицу базы данных.

Основные шаги:

1. Подготовка файла-источника
2. Создание или выбор целевой таблицы
3. Сопоставление полей файла и таблицы
4. Настройка параметров импорта
5. Проверка результата

2.3. Процесс экспорта

Экспорт – сохранение данных из таблицы базы данных во внешний файл.

Основные шаги:

1. Выбор данных для экспорта
 2. Выбор формата экспорта
 3. Настройка параметров экспорта
 4. Сохранение файла
-

3. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ

Задача: Импорт данных из CSV-файла в таблицу «Сотрудники» и экспорт данных в файл Calc

Шаг 1. Подготовка CSV-файла

1. Откройте Блокнот или LibreOffice Calc
2. Создайте файл с данными:

text

Фамилия,Имя,Должность,Отдел,Оклад,Дата_приема

Иванов,Иван,Бухгалтер,Бухгалтерия,45000,15.03.2020

Петрова,Анна,Менеджер,Отдел кадров,40000,10.05.2019

Сидоров,Петр,Программист,IT-отдел,60000,20.01.2018

Кузнецова, Мария, Главный бухгалтер, Бухгалтерия, 55000, 05.11.2017

3. Сохраните файл как «Сотрудники_импорт.csv»

Шаг 2. Создание таблицы в базе данных

1. Создайте новую базу данных «Импорт_экспорт.odb»

2. Создайте таблицу «Сотрудники»:

text

ID (INTEGER, Автозначение: Да, PK)

Фамилия (VARCHAR, 30)

Имя (VARCHAR, 20)

Должность (VARCHAR, 30)

Отдел (VARCHAR, 40)

Оклад (DECIMAL)

Дата_приема (DATE)

Шаг 3. Импорт данных из CSV-файла

1. В главном окне Base выберите «Таблицы»

2. Щелкните правой кнопкой на таблице «Сотрудники»

3. Выберите «Копировать...»

4. В диалоговом окне выберите:

○ Откуда: «Из файла»

○ Тип файла: «Текстовый CSV»

○ Нажмите «Обзор» и выберите ваш CSV-файл

5. Нажмите «Далее» и выполните настройки:

○ Разделитель полей: запятая

○ Разделитель текста: кавычки

○ Кодировка: UTF-8

○ Первая строка содержит имена столбцов: да

6. Сопоставьте поля файла с полями таблицы

7. Нажмите «Создать»

8. Проверьте, что данные загружены корректно

Шаг 4. Экспорт данных в файл Calc

1. Откройте таблицу «Сотрудники» в режиме таблицы

2. Выделите все записи (Ctrl+A)

3. Правка → Копировать

4. Откройте LibreOffice Calc

5. Создайте новый документ

6. Правка → Вставить

7. Сохраните файл как «Экспорт_сотрудников.ods»

Шаг 5. Импорт из Calc-файла

1. В LibreOffice Calc добавьте несколько новых записей в файл

2. Сохраните файл

3. В Base создайте новую таблицу «Сотрудники_новая»

4. Используйте импорт из файла Calc

5. Выберите файл «Экспорт_сотрудников.ods»

6. Настройте параметры импорта

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие форматы файлов можно импортировать в LibreOffice Base?

2. Как подготовить CSV-файл для импорта?

3. Что такое разделитель полей и для чего он нужен?

4. Как сопоставить поля файла и таблицы при импорте?

5. Как экспортировать данные из таблицы в файл Calc?

6. Какие проблемы могут возникнуть при импорте данных?

7. Как исправить ошибки кодировки при импорте?
 8. Можно ли импортировать данные в существующую таблицу с записями?
 9. Как экспортировать не все данные, а только часть?
 10. Для чего нужно указывать тип данных при импорте?
-

5. ЗАДАНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ

Вариант 1: База данных «Учебный центр»

1. Подготовьте CSV-файл «Курсы.csv»:

text

Название_курса,Преподаватель,Длительность,Стоимость,Начало_курса
Базы данных,Иванов И.И.,40,15000,10.10.2023
Программирование,Петров П.П.,60,20000,15.09.2023
Сетевые технологии,Сидоров С.С.,50,18000,20.11.2023
Веб-дизайн,Кузнецова М.М.,45,16000,05.10.2023

2. Подготовьте файл Calc «Слушатели.ods»:

text

Фамилия,Имя,Курс,Телефон,Дата_регистрации
Алексеев,Алексей,Базы данных,89161234567,01.09.2023
Борисова,Мария,Программирование,89162345678,10.09.2023
Васильев,Дмитрий,Сетевые технологии,89163456789,15.10.2023
Григорьева,Ольга,Веб-дизайн,89164567890,20.09.2023

Задания:

1. Создайте базу данных «Учебный_центр_импорт.odb»
2. Создайте таблицу «Курсы» с полями, соответствующими CSV-файлу
3. Импортируйте данные из «Курсы.csv» в таблицу
4. Создайте таблицу «Слушатели» с полями, соответствующими Calc-файлу
5. Импортируйте данные из «Слушатели.ods» в таблицу
6. Экспортируйте таблицу «Курсы» в новый файл Calc
7. Добавьте в экспортированный файл 2 новых курса
8. Импортируйте обновленные данные обратно в новую таблицу «Курсы_обновленные»
9. Создайте отчет об ошибках импорта (если они возникли)
10. Проверьте целостность импортированных данных

Вариант 2: База данных «Библиотека»

1. Подготовьте CSV-файл «Книги_импорт.csv»:

text

Название,Автор,Жанр,Год_издания,Цена,ISBN
Мастер и Маргарита,Булгаков М.А.,Роман,1966,500,978-5-17-090644-7
Преступление и наказание,Достоевский Ф.М.,Роман,1866,450,978-5-17-090987-5
1984,Оруэлл Дж.,Фантастика,1949,600,978-5-17-090234-0
Алые паруса,Грин А.С.,Повесть,1923,350,978-5-17-091234-9

2. Подготовьте файл Calc «Читатели_импорт.ods»:

text

Фамилия,Имя,Номер_билета,Группа,Телефон
Иванов,Иван,001,ИСП-101,89161112233
Петрова,Анна,002,ИСП-102,89162223344
Сидоров,Петр,003,ИСП-101,89163334455
Кузнецова,Мария,004,ИСП-103,89164445566

Задания:

1. Создайте базу данных «Библиотека_импорт.odb»
2. Создайте таблицу «Книги» для импорта CSV-данных
3. Импортируйте данные из CSV-файла

4. Создайте таблицу «Читатели» для импорта Calc-данных
5. Импортируйте данные из Calc-файла
6. Экспортируйте обе таблицы в один файл Calc на разных листах
7. Исправьте в экспортированном файле ошибки данных (если есть)
8. Добавьте 3 новые записи в каждый лист
9. Импортируйте исправленные данные в новые таблицы
10. Сравните исходные и импортированные данные

Вариант 3: База данных «Магазин»

1. **Подготовьте CSV-файл «Товары_импорт.csv»:**

text

Наименование,Категория,Производитель,Цена,Количество,Срок_годности

Молоко,Молочные продукты,Компания А,80,50,10.12.2023

Хлеб,Хлебобулочные,Компания Б,45,100,05.12.2023

Яблоки,Фрукты,Компания В,120,30,15.12.2023

Сахар,Бакалея,Компания Г,65,40,01.06.2024

2. **Подготовьте файл Calc «Поставщики_импорт.ods»:**

text

Название_компании,Адрес,Телефон,Контактное_лицо,Рейтинг

Компания А,ул. Ленина, 10,89161111111,Иванов И.И.,5

Компания Б,ул. Советская, 25,89162222222,Петров П.П.,4

Компания В,ул. Мира, 15,89163333333,Сидоров С.С.,5

Компания Г,ул. Центральная, 8,89164444444,Кузнецова М.М.,3

Задания:

1. Создайте базу данных «Магазин_импорт.odb»
2. Создайте таблицы для импорта данных из обоих файлов
3. Импортируйте данные из CSV-файла
4. Импортируйте данные из Calc-файла
5. Экспортируйте данные в CSV-файл с разделителем точка с запятой
6. Экспортируйте данные в текстовый файл с фиксированной шириной полей
7. Создайте новый CSV-файл с измененными данными
8. Импортируйте измененные данные
9. Проведите сравнение различных форматов экспорта
10. Составьте инструкцию по импорту/экспорту данных

Общие требования:

1. Для своего варианта подготовьте оба файла с данными
2. Создайте базу данных с соответствующими таблицами
3. Выполните импорт из обоих типов файлов
4. Выполните экспорт в разные форматы
5. Проведите редактирование данных в файлах
6. Повторите импорт обновленных данных

Отчет должен содержать:

1. Титульный лист
2. Содержимое подготовленных файлов
3. Структуру созданных таблиц
4. Скриншоты:
 - Процесса импорта данных
 - Таблиц с импортированными данными
 - Процесса экспорта данных
 - Экспортированных файлов
 - Сравнения данных до и после редактирования
5. Описание возникших проблем и способов их решения
6. Ответы на контрольные вопросы

7. Выводы о проделанной работе

6. КРИТЕРИИ ОЦЕНКИ

Оценка «5» (Отлично):

- Подготовлены корректные файлы с данными
- Созданы все необходимые таблицы в базе данных
- Успешно выполнен импорт из обоих типов файлов
- Успешно выполнен экспорт в разные форматы
- Проведено редактирование и повторный импорт данных
- Документированы все этапы работы
- Работа сдана в срок, отчет оформлен полностью

Оценка «4» (Хорошо):

- Файлы подготовлены с незначительными ошибками
- Импорт выполнен, но с небольшими проблемами
- Экспорт выполнен не во все требуемые форматы
- Редактирование данных проведено частично
- Отчет оформлен с небольшими замечаниями

Оценка «3» (Удовлетворительно):

- Файлы подготовлены с ошибками
- Импорт выполнен только из одного файла
- Экспорт выполнен с ошибками
- Отчет оформлен не полностью

Оценка «2» (Неудовлетворительно):

- Файлы не подготовлены
- Импорт/экспорт не выполнен
- Работа не выполнена или сдана не в срок

ПРАКТИЧЕСКАЯ РАБОТА_№10. Добавление записей в табличный файл из двумерного массива. Работа с командами ввода-вывода. Использование функций для работы с массивами.

по дисциплине «Основы проектирования баз данных»

Тема: Продвинутое методы импорта/экспорта данных. Работа с формами ввода данных

1. ЦЕЛИ И ЗАДАЧИ ПРАКТИЧЕСКОЙ РАБОТЫ

Цель: Научиться использовать формы для массового ввода данных и освоить сложные сценарии импорта/экспорта в LibreOffice Base.

Задачи:

1. Научиться создавать формы для удобного ввода данных в таблицы.
2. Освоить массовый ввод данных через формы.
3. Научиться импортировать данные с изменением структуры.
4. Освоить экспорт данных с фильтрацией.
5. Научиться решать типичные проблемы при работе с данными.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.1. Формы для ввода данных

Форма – это специальный объект базы данных, который позволяет удобно вводить, просматривать и редактировать данные. Формы могут быть связаны с одной или несколькими таблицами.

Преимущества использования форм:

- Удобный интерфейс для пользователя
- Проверка вводимых данных
- Возможность группировки полей
- Автоматическое заполнение некоторых полей

2.2. Массовый ввод данных

Массовый ввод – добавление нескольких записей одновременно. В LibreOffice Base можно использовать:

1. Формы с кнопкой «Новая запись»
2. Импорт из подготовленных файлов
3. Копирование данных из других приложений

2.3. Преобразование данных при импорте

Часто данные в файле имеют структуру, отличную от структуры таблицы. В этом случае необходимо:

- Изменять порядок полей
- Объединять или разделять данные
- Преобразовывать форматы данных

3. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ

Задача: Создать систему массового ввода данных о сотрудниках с использованием форм и сложного импорта

Шаг 1. Создание базы данных и таблицы

1. Создайте базу данных «Кадры_массовый.odb»
2. Создайте таблицу «Сотрудники_расширенная»:

text

ID (INTEGER, Автозначение: Да, PK)

Табельный_номер (VARCHAR, 10)

Фамилия (VARCHAR, 30)

Имя (VARCHAR, 20)

Отчество (VARCHAR, 30)

Должность (VARCHAR, 40)

Отдел (VARCHAR, 50)

Оклад (DECIMAL)

Дата_приема (DATE)

Стаж (INTEGER)

Проект (VARCHAR, 50)

Шаг 2. Создание формы для массового ввода

1. В разделе «Формы» нажмите «Использовать мастер для создания формы»
2. Выберите таблицу «Сотрудники_расширенная»
3. Добавьте все поля в форму
4. Выберите макет «В столбец»
5. Выберите стиль оформления
6. Дайте форме имя «Форма_ввода_сотрудников»
7. Нажмите «Готово»

Шаг 3. Настройка формы

1. Откройте форму в режиме конструктора
2. Добавьте на форму:
 - Заголовок «Массовый ввод сотрудников»
 - Кнопку «Новая запись» (из палитры элементов)
 - Кнопку «Сохранить»
3. Настройте размеры полей для удобного ввода
4. Сохраните форму

Шаг 4. Массовый ввод данных через форму

Используя созданную форму, введите 10 записей:

text

1. ТН-001, Иванов, Иван, Иванович, Программист, IT-отдел, 60000, 15.03.2020, 3, Проект А
2. ТН-002, Петрова, Анна, Петровна, Бухгалтер, Бухгалтерия, 45000, 10.05.2019, 4, Проект Б
3. ТН-003, Сидоров, Петр, Сергеевич, Менеджер, Отдел продаж, 50000, 20.01.2018, 5, Проект В

... (добавьте 7 своих записей)

Шаг 5. Подготовка файла для сложного импорта

Создайте файл CSV «Дополнительные_сотрудники.csv» с измененной структурой:

text

Номер, ФИО, Должность и отдел, Зарплата, Дата найма

ТН-011, "Смирнов Алексей Николаевич", "Инженер (Технический отдел)", 55000, 2022-11-15

ТН-012, "Козлова Елена Викторовна", "Дизайнер (Отдел маркетинга)", 48000, 2021-09-20

ТН-013, "Федоров Дмитрий Андреевич", "Аналитик (IT-отдел)", 62000, 2020-07-10

Шаг 6. Сложный импорт с преобразованием данных

1. Создайте новую таблицу «Импорт_временная» для загрузки CSV
2. Импортируйте данные из CSV-файла
3. Создайте запрос для преобразования данных:
 - Разделите поле «ФИО» на Фамилия, Имя, Отчество
 - Разделите поле «Должность и отдел» на Должность и Отдел
 - Преобразуйте формат даты
4. Экспортируйте преобразованные данные в новый файл
5. Импортируйте преобразованные данные в основную таблицу

Шаг 7. Экспорт с фильтрацией

1. Создайте фильтр для отбора сотрудников IT-отдела
2. Экспортируйте отфильтрованные данные в файл Calc
3. Создайте фильтр для сотрудников с окладом более 50000
4. Экспортируйте эти данные в CSV-файл

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для чего используются формы в базе данных?
2. Как создать форму для ввода данных?
3. Какие преимущества дает использование форм по сравнению с прямым вводом в таблицу?
4. Как организовать массовый ввод данных через форму?
5. Что делать, если структура импортируемого файла не совпадает со структурой таблицы?
6. Как разделить объединенные данные (например, ФИО) при импорте?
7. Как экспортировать не все данные, а только часть?
8. Какие проблемы могут возникнуть при массовом импорте данных?
9. Как проверить корректность импортированных данных?
10. Можно ли автоматизировать процесс импорта/экспорта?

5. ЗАДАНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ

Вариант 1: База данных «Университет»

1. **Создайте таблицу «Студенты_полная»:**

text

ID, Номер_зачетки, ФИО, Дата_рождения, Группа, Специальность, Средний_балл, Форма_обучения, Статус

2. **Создайте форму «Форма_ввода_студентов»:**

- Добавьте все поля таблицы
- Сгруппируйте поля логически
- Добавьте кнопки навигации
- Настройте удобный интерфейс

3. **Подготовьте CSV-файл с объединенными данными:**

text

Зачетка, ФИО и дата_рождения, Группа и специальность, Балл, Информация

2023001, "Иванов Иван 15.05.2002", "ИСП-101 (Информационные системы)", 4.5, "Бюджет/Активный"

2023002, "Петрова Анна 20.03.2003", "ПКС-102 (Программирование)", 4.2, "Контракт/Активный"

2023003, "Сидоров Петр 10.11.2002", "ИСП-101 (Информационные системы)", 3.8, "Бюджет/Академич"

Задания:

1. Создайте базу данных «Университет_массовый.odb»
2. Создайте таблицу и форму для ввода данных
3. Введите через форму 8 записей о студентах
4. Импортируйте данные из CSV-файла во временную таблицу
5. Разработайте метод разделения объединенных полей
6. Импортируйте преобразованные данные в основную таблицу
7. Экспортируйте данные только по специальности «Информационные системы»
8. Экспортируйте данные студентов с баллом выше 4.0
9. Создайте отчет о количестве импортированных записей
10. Проведите проверку целостности данных

Вариант 2: База данных «Больница»

1. **Создайте таблицу «Пациенты_полная»:**

text

ID, Номер_карты, ФИО_пациента, Дата_рождения, Диагноз, Врач, Отделение, Дата_поступления, Статус, Лечащий_врач

2. **Создайте форму «Форма_регистрации_пациентов»:**

- Сделайте удобный интерфейс для медсестры
- Добавьте проверку вводимых данных
- Сгруппируйте поля по смыслу

3. **Подготовьте файл для импорта:**

text

Карта, Данные_пациента, Медицинская_информация, Госпитализация

001, "Смирнов А.Л. (1985-04-15)", "ОРВИ / Иванова И.И.", "Терапевтическое / 2023-11-10 / На лечении"

002, "Козлова Е.В. (1978-09-22)", "Гипертония / Петров П.П.", "Кардиология / 2023-11-12 / Выписан"

003, "Федоров Д.А. (1992-07-30)", "Гастрит / Сидорова С.С.", "Гастроэнтерология / 2023-11-15 / На лечении"

Задания:

1. Создайте базу данных «Больница_массовый.odb»
2. Создайте таблицу и форму для регистрации пациентов
3. Введите через форму 8 записей
4. Импортируйте данные из файла с разделением сложных полей
5. Создайте фильтр для пациентов терапевтического отделения
6. Экспортируйте данные текущих пациентов
7. Экспортируйте данные за последнюю неделю
8. Создайте форму для быстрого поиска пациента
9. Проведите сравнение ручного и импортированного ввода
10. Составьте инструкцию по массовому вводу данных

Вариант 3: База данных «Магазин_товаров»

1. **Создайте таблицу «Товары_полная»:**

text

ID, Артикул, Наименование_и_описание, Категория_и_подкатегория, Цены, Количества, Поставщик_и_контакты, Дата_поступления

2. **Создайте форму «Форма_учета_товаров»:**

- Сделайте интерфейс для кладовщика
- Добавьте поля для расчетов
- Настройте автоматические вычисления

3. **Подготовьте данные для импорта:**

text

Код, Товар_и_категория, Ценовая_информация, Складская_информация, Контакты

001, "Ноутбук Acer (Электроника/Компьютеры)", "45000/50000/47000", "10/5/15", "ООО Техно (89161112233)"

002, "Молоко (Продукты/Молочные)", "80/90/85", "50/30/80", "Ферма (89162223344)"

003, "Книга (Канцелярия/Книги)", "500/600/550", "20/10/30", "Издательство (89163334455)"

Задания:

1. Создайте базу данных «Магазин_массовый.odb»
2. Создайте таблицу и форму для учета товаров
3. Введите через форму 10 товаров
4. Импортируйте данные из файла с преобразованием структуры
5. Создайте фильтр для товаров категории «Электроника»
6. Экспортируйте данные по товарам с остатком менее 10
7. Экспортируйте данные поставщиков
8. Создайте форму для быстрого добавления товаров
9. Проведите инвентаризацию импортированных данных
10. Разработайте систему проверки дублирования товаров

Общие требования:

1. Для своего варианта создайте полную базу данных
2. Разработайте удобную форму для ввода данных
3. Выполните массовый ввод через форму (8-10 записей)
4. Подготовьте и импортируйте данные из файла сложной структуры
5. Выполните преобразование данных при импорте
6. Проведите экспорт с различными фильтрами
7. Протестируйте все созданные объекты
8. Документируйте процесс работы

Отчет должен содержать:

1. Титульный лист
 2. Структуру базы данных
 3. Скриншоты:
 - Созданной формы
 - Процесса ввода данных через форму
 - Файлов для импорта
 - Процесса импорта с преобразованием данных
 - Результаты экспорта с фильтрацией
 4. Описание решенных проблем
 5. Сравнение эффективности разных методов ввода
 6. Ответы на контрольные вопросы
 7. Выводы и рекомендации
-

6. КРИТЕРИИ ОЦЕНКИ**Оценка «5» (Отлично):**

- Создана полноценная база данных с таблицей и формой
- Разработана удобная форма для массового ввода
- Выполнен массовый ввод через форму (8-10 записей)
- Успешно импортированы данные из файла сложной структуры
- Выполнено преобразование данных при импорте
- Проведен экспорт с различными фильтрами
- Документированы все этапы работы
- Работа сдана в срок, отчет оформлен полностью

Оценка «4» (Хорошо):

- Создана база данных, но форма требует доработки
- Выполнен ввод данных, но менее 8 записей
- Импорт выполнен с небольшими проблемами
- Не все виды экспорта выполнены
- Отчет оформлен с незначительными замечаниями

Оценка «3» (Удовлетворительно):

- База данных создана с ошибками
- Форма не функциональна
- Импорт/экспорт выполнен частично
- Отчет оформлен не полностью

Оценка «2» (Неудовлетворительно):

- База данных не создана
- Форма не создана
- Импорт/экспорт не выполнен
- Работа не выполнена или сдана не в срок

ПРАКТИЧЕСКАЯ РАБОТА №11. Создание меню различных видов. Модификация и управление меню.

по дисциплине «Основы проектирования баз данных»

Тема: Создание пользовательских форм с элементами управления. Управление интерфейсом базы данных

1. ЦЕЛИ И ЗАДАЧИ ПРАКТИЧЕСКОЙ РАБОТЫ

Цель: Научиться создавать сложные формы с различными элементами управления и организовывать удобный пользовательский интерфейс для работы с базой данных.

Задачи:

1. Научиться добавлять различные элементы управления на формы (кнопки, выпадающие списки, флажки).
2. Освоить создание главной формы с меню для навигации по базе данных.
3. Научиться настраивать внешний вид форм (цвета, шрифты, расположение элементов).
4. Освоить связывание кнопок с действиями (открытие форм, таблиц, отчетов).
5. Научиться создавать удобный интерфейс для конечного пользователя.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.1. Элементы управления в формах

Элементы управления – объекты на форме, которые позволяют пользователю взаимодействовать с базой данных.

Основные элементы управления:

- **Надпись** – статический текст (заголовки, пояснения)
- **Текстовое поле** – для отображения и ввода данных
- **Кнопка** – для выполнения действий
- **Выпадающий список** – для выбора из заранее заданных значений
- **Флажок** – для выбора Да/Нет
- **Поле со списком** – комбинация текстового поля и списка
- **Группа** – для объединения связанных элементов

2.2. Главная форма и навигация

Главная форма – центральный интерфейс базы данных, с которого начинается работа.

Элементы навигации:

- Кнопки для открытия различных форм
- Меню с разделами
- Панель быстрого доступа
- Гиперссылки между формами

2.3. Настройка внешнего вида

Дизайн формы включает:

- Выбор цветовой схемы
- Настройку шрифтов и размеров
- Расположение элементов
- Добавление графических элементов
- Настройку выравнивания

3. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ

Задача: Создать главное меню и функциональные формы для базы данных «Библиотека»

Шаг 1. Создание базы данных и таблиц

1. Создайте базу данных «Библиотека_меню.odb»
2. Создайте таблицы:
 - **«Книги»** (ID_книги, Название, Автор, Жанр, Год, Наличие)
 - **«Читатели»** (ID_читателя, Фамилия, Имя, Группа, Телефон)

- **«Выдача»** (ID_выдачи, ID_книги, ID_читателя, Дата_выдачи, Дата_возврата)

Шаг 2. Создание главной формы-меню

1. В разделе «Формы» выберите «Создать форму в режиме дизайна»
2. В конструкторе увеличьте размер формы (примерно 10x15 см)
3. Добавьте элементы:
 - **Надпись:** «БИБЛИОТЕКА» (крупный шрифт, по центру)
 - **Надпись:** «Главное меню» (под заголовком)
 - **Кнопка 1:** «Книги» (откроет форму с книгами)
 - **Кнопка 2:** «Читатели» (откроет форму с читателями)
 - **Кнопка 3:** «Выдача книг» (откроет форму выдачи)
 - **Кнопка 4:** «Отчеты» (откроет отчеты)
 - **Кнопка 5:** «Выход» (закроет базу данных)
4. Настройте кнопки:
 - Размер: 3x1 см
 - Цвет фона: синий
 - Цвет текста: белый
 - Шрифт: Arial, 12pt
5. Расположите кнопки вертикальным списком

Шаг 3. Создание функциональных форм

1. **Форма «Книги»:**
 - Используйте Мастер форм для таблицы «Книги»
 - Добавьте кнопки: «Новая запись», «Сохранить», «Удалить», «Назад в меню»
 - Настройте цвета: светло-голубой фон, темно-синие надписи
2. **Форма «Читатели»:**
 - Создайте в режиме конструктора
 - Добавьте поля: Фамилия, Имя, Группа, Телефон
 - Добавьте выпадающий список для группы (значения: ИСП-101, ИСП-102, ПКС-201)
3. **Форма «Выдача книг»:**
 - Создайте форму с полями из таблицы «Выдача»
 - Добавьте поля со списком для выбора книги и читателя

Шаг 4. Связывание кнопок с действиями

1. На главной форме выделите кнопку «Книги»
2. Щелкните правой кнопкой → «Свойства элемента управления»
3. На вкладке «События» найдите «При нажатии мыши»
4. Выберите «Открыть форму» → «Книги»
5. Повторите для других кнопок
6. Для кнопки «Выход» выберите действие «Выход»

Шаг 5. Настройка внешнего вида

1. Выделите всю форму
2. Свойства формы → Фон → выберите цвет
3. Добавьте изображение-логотип (если есть)
4. Настройте выравнивание всех элементов
5. Проверьте удобство использования

Шаг 6. Тестирование меню

1. Сохраните главную форму как «Главное_меню»
2. Установите ее как стартовую форму:
 - Сервис → Параметры → База данных
 - В разделе «При открытии» выберите «Открыть форму»
 - Выберите «Главное_меню»
3. Закройте и откройте базу данных

4. Проверьте работу всех кнопок

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для чего нужна главная форма в базе данных?
2. Какие элементы управления можно добавить на форму?
3. Как создать кнопку, которая открывает другую форму?
4. Что такое выпадающий список и как его создать?
5. Как изменить цвет фона формы?
6. Как настроить размеры элементов на форме?
7. Как сделать форму стартовой (открывающейся при запуске)?
8. Какие действия можно назначить на кнопку?
9. Как создать группу связанных элементов на форме?
10. Как проверить работоспособность созданного меню?

5. ЗАДАНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ

Вариант 1: Система «Учебный отдел»

Создайте базу данных для учета студентов и предметов.

Таблицы:

1. «Студенты»: ID, Фамилия, Имя, Группа, Средний_балл, Статус
2. «Предметы»: ID, Название, Преподаватель, Часы, Семестр
3. «Оценки»: ID, ID_студента, ID_предмета, Оценка, Дата

Задания:

1. **Создайте главное меню:**
 - Заголовок: «УЧЕБНЫЙ ОТДЕЛ»
 - Кнопки: «Студенты», «Предметы», «Ввод оценок», «Отчеты», «Выход»
 - Дизайн: сине-белая цветовая схема
2. **Создайте форму «Студенты»:**
 - Все поля из таблицы
 - Выпадающий список для группы (ИСП-101, ИСП-102, ПКС-201)
 - Флажок «Активный студент» (статус)
 - Кнопки: «Добавить», «Сохранить», «Удалить», «Назад»
3. **Создайте форму «Ввод оценок»:**
 - Поля: Студент (выпадающий список), Предмет (выпадающий список), Оценка, Дата
 - Группа кнопок для оценок (2, 3, 4, 5)
 - Автоматическое заполнение даты
4. **Настройте навигацию:**
 - Свяжите все кнопки главного меню с соответствующими формами
 - Добавьте кнопку «Назад» на каждой форме
 - Сделайте главное меню стартовой формой
5. **Улучшите интерфейс:**
 - Добавьте подсказки к элементам
 - Настройте удобные размеры полей
 - Проверьте работу всех элементов

Вариант 2: Система «Магазин»

Создайте базу данных для учета товаров и продаж.

Таблицы:

1. «Товары»: ID, Наименование, Категория, Цена, Количество, Поставщик
2. «Продажи»: ID, ID_товара, Количество, Дата_продажи, Сумма
3. «Поставщики»: ID, Название, Телефон, Рейтинг

Задания:

1. **Создайте главное меню:**
 - Заголовок: «МАГАЗИН ЭЛЕКТРОНИКИ»
 - Кнопки: «Товары», «Продажи», «Поставщики», «Отчеты», «Выход»
 - Дизайн: зеленоватая цветовая схема
2. **Создайте форму «Товары»:**
 - Все поля из таблицы
 - Выпадающий список для категории (Телефоны, Ноутбуки, Аксессуары)
 - Поле с форматированием для цены
 - Индикатор количества (цвет меняется при малом остатке)
3. **Создайте форму «Продажи»:**
 - Поля: Товар (выпадающий список), Количество, Дата, Сумма (авторасчет)
 - Кнопки: «Рассчитать сумму», «Оформить продажу», «Отмена»
4. **Создайте форму «Поставщики»:**
 - Поля: Название, Телефон, Рейтинг (шкала 1-5)
 - Кнопки для быстрого выбора рейтинга
5. **Настройте взаимодействие:**
 - Сделайте главное меню центральным узлом
 - Добавьте переходы между формами
 - Проверьте работоспособность всех связей

Вариант 3: Система «Поликлиника»

Создайте базу данных для регистрации пациентов.

Таблицы:

1. **«Пациенты»:** ID, Фамилия, Имя, Дата_рождения, Полис, Диагноз
2. **«Врачи»:** ID, Фамилия, Имя, Специальность, Кабинет, График
3. **«Записи»:** ID, ID_пациента, ID_врача, Дата, Время, Статус

Задания:

1. **Создайте главное меню:**
 - Заголовок: «ПОЛИКЛИНИКА №1»
 - Кнопки: «Пациенты», «Врачи», «Запись на прием», «Расписание», «Выход»
 - Дизайн: медицинская тема (бело-голубые тона)
2. **Создайте форму «Пациенты»:**
 - Все поля из таблицы
 - Группа для персональных данных
 - Группа для медицинских данных
 - Кнопка «Новая карта»
3. **Создайте форму «Запись на прием»:**
 - Выбор пациента (выпадающий список)
 - Выбор врача (выпадающий список с фильтром по специальности)
 - Выбор даты и времени
 - Кнопка «Записать»
4. **Создайте форму «Расписание врачей»:**
 - Табличный вид
 - Фильтр по дням недели
 - Цветовое кодирование занятых/свободных слотов
5. **Создайте удобный интерфейс:**
 - Единый стиль для всех форм
 - Подсказки для сложных элементов
 - Проверка вводимых данных

Общие требования для всех вариантов:

1. Создайте базу данных с тремя таблицами
2. Разработайте главное меню с 5-6 кнопками

3. Создайте не менее 3 функциональных форм
4. Используйте различные элементы управления
5. Настройте навигацию между формами
6. Сделайте интерфейс удобным и эстетичным
7. Протестируйте все функции

Отчет должен содержать:

1. Титульный лист
 2. Описание структуры базы данных
 3. Скриншоты:
 - Главного меню
 - Каждой созданной формы
 - Настройки свойств элементов
 - Процесса связывания кнопок с действиями
 4. Описание использованных элементов управления
 5. Инструкцию по использованию созданного интерфейса
 6. Ответы на контрольные вопросы
 7. Выводы о проделанной работе
-

6. КРИТЕРИИ ОЦЕНКИ

Оценка «5» (Отлично):

- Полностью создана база данных со всеми таблицами
- Разработано удобное главное меню с навигацией
- Созданы все требуемые формы с различными элементами управления
- Настроены связи между формами
- Интерфейс эстетичен и удобен в использовании
- Все элементы работают корректно
- Работа сдана в срок, отчет оформлен полностью

Оценка «4» (Хорошо):

- Создана база данных, но некоторые таблицы требуют доработки
- Главное меню создано, но навигация неполная
- Созданы не все формы или с ограниченным функционалом
- Интерфейс требует улучшения
- Отчет оформлен с небольшими замечаниями

Оценка «3» (Удовлетворительно):

- База данных создана с ошибками
- Главное меню отсутствует или не функционально
- Создана только одна форма
- Элементы управления не настроены
- Отчет оформлен не полностью

Оценка «2» (Неудовлетворительно):

- База данных не создана
- Формы не созданы
- Меню отсутствует
- Работа не выполнена или сдана не в срок

ПРАКТИЧЕСКАЯ РАБОТА_№12. Создание рабочих и системных окон. Добавление элементов управления рабочим окном.

по дисциплине «Основы проектирования баз данных»

Тема: Основы SQL. Создание и заполнение таблиц. Простые запросы SELECT

РЕКОМЕНДАЦИЯ ПО ВЫБОРУ СИСТЕМЫ ДЛЯ РАБОТЫ

Для выполнения этой работы рекомендуется использовать:

Вариант 1: Онлайн-редакторы (самый простой способ)

1. **SQLite Online** - <https://sqliteonline.com/>
 - Не требует установки
 - Работает прямо в браузере
 - Поддерживает стандартный SQL
 - Бесплатно
2. **DB Fiddle** - <https://www.db-fiddle.com/>
 - Можно выбрать разные СУБД
 - Удобный интерфейс
 - Возможность сохранять свои работы

Вариант 2: Локальные программы

1. **SQLiteStudio** (бесплатно) - <https://sqlitestudio.pl/>
 - Легкая установка
 - Русский интерфейс
 - Отличный для обучения
2. **DB Browser for SQLite** (бесплатно) - <https://sqlitebrowser.org/>
 - Простой интерфейс
 - Подходит для начинающих
 - Много обучающих материалов

Вариант 3: В рамках курса (если используете на занятиях)

- **LibreOffice Base** (в режиме SQL)
- **Microsoft SQL Server Express** (бесплатно)
- **MySQL Workbench**

Начинающим рекомендуется: SQLite Online или SQLiteStudio

1. ЦЕЛИ И ЗАДАЧИ ПРАКТИЧЕСКОЙ РАБОТЫ

Цель: Освоить базовые команды SQL для создания таблиц, добавления данных и выполнения простых запросов на выборку.

Задачи:

1. Научиться создавать таблицы с помощью команды CREATE TABLE.
 2. Освоить добавление данных в таблицы командой INSERT.
 3. Научиться выполнять простые запросы SELECT для выборки данных.
 4. Освоить использование условия WHERE для фильтрации записей.
 5. Научиться сортировать результаты с помощью ORDER BY.
-

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.1. Что такое SQL

SQL (Structured Query Language) - язык структурированных запросов для работы с базами данных.

2.2. Основные команды SQL

1. **CREATE TABLE** - создание таблицы
2. **INSERT INTO** - добавление данных
3. **SELECT** - выборка данных
4. **WHERE** - условие отбора

5. ORDER BY - сортировка результатов

2.3. Синтаксис SQL

- Команды пишутся в верхнем или нижнем регистре
- Каждая команда заканчивается точкой с запятой (;)
- Текстовые значения заключаются в одинарные кавычки ('')

3. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ

Задача: Создать базу данных для учета студентов

Шаг 1. Создание таблицы

sql

-- Создаем таблицу "Студенты"

```
CREATE TABLE Студенты (  
    id INTEGER PRIMARY KEY,  
    фамилия TEXT NOT NULL,  
    имя TEXT NOT NULL,  
    группа TEXT,  
    средний_балл REAL,  
    дата_поступления DATE  
);
```

Объяснение:

- CREATE TABLE - команда создания таблицы
- Студенты - название таблицы
- id INTEGER PRIMARY KEY - уникальный идентификатор
- TEXT - текстовый тип данных
- REAL - дробное число
- DATE - дата
- NOT NULL - поле не может быть пустым

Шаг 2. Добавление данных

sql

-- Добавляем записи в таблицу

```
INSERT INTO Студенты (фамилия, имя, группа, средний_балл, дата_поступления)  
VALUES  
('Иванов', 'Иван', 'ИСП-101', 4.5, '2022-09-01'),  
('Петрова', 'Анна', 'ИСП-101', 4.2, '2022-09-01'),  
('Сидоров', 'Петр', 'ИСП-102', 3.8, '2022-09-01'),  
('Кузнецова', 'Мария', 'ИСП-101', 4.7, '2022-09-01'),  
('Васильев', 'Алексей', 'ИСП-102', 4.0, '2022-09-01');
```

Шаг 3. Простая выборка данных

sql

-- Выбрать всех студентов

```
SELECT * FROM Студенты;
```

Шаг 4. Выборка с условием

sql

-- Студенты группы ИСП-101

```
SELECT * FROM Студенты WHERE группа = 'ИСП-101';
```

Шаг 5. Выборка конкретных полей

sql

-- Только фамилия, имя и балл

```
SELECT фамилия, имя, средний_балл FROM Студенты;
```

Шаг 6. Сортировка результатов

sql

-- Сортировка по фамилии (А-Я)

```
SELECT * FROM Студенты ORDER BY фамилия ASC;
```

-- Сортировка по баллу (по убыванию)

```
SELECT * FROM Студенты ORDER BY средний_балл DESC;
```

Шаг 7. Выборка с несколькими условиями

sql

-- Студенты ИСП-101 с баллом выше 4.0

```
SELECT * FROM Студенты
```

```
WHERE группа = 'ИСП-101' AND средний_балл > 4.0;
```

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какая команда SQL используется для создания таблицы?
2. Как добавить новую запись в таблицу?
3. Что означает символ * в команде SELECT?
4. Для чего используется ключевое слово WHERE?
5. Как отсортировать результаты по возрастанию?
6. Что такое PRIMARY KEY?
7. Какие типы данных вы знаете в SQL?
8. Как выбрать только определенные столбцы из таблицы?
9. Что означает ключевое слово NOT NULL?
10. Как выбрать записи, удовлетворяющие нескольким условиям?

5. ЗАДАНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ

Вариант 1: Библиотека

Создайте базу данных для учета книг в библиотеке.

Задание:

1. Создайте таблицу Книги со следующими полями:
 - id (уникальный идентификатор)
 - название (текст, обязательно)
 - автор (текст)
 - год_издания (целое число)
 - жанр (текст)
 - цена (дробное число)
2. Добавьте в таблицу 7 книг (придумайте сами).
3. Выполните запросы:
 - Показать все книги
 - Показать книги определенного автора
 - Показать книги, изданные после 2010 года
 - Показать книги жанра "Фантастика"
 - Показать книги, отсортированные по цене (от дешевых к дорогим)
 - Показать книги дороже 500 рублей
 - Показать название и автора книг, изданных в 2020 году

Вариант 2: Магазин

Создайте базу данных для учета товаров в магазине.

Задание:

1. Создайте таблицу Товары со следующими полями:
 - код_товара (уникальный идентификатор)
 - наименование (текст, обязательно)
 - категория (текст)
 - цена (дробное число)
 - количество (целое число)
 - поставщик (текст)

2. Добавьте в таблицу 7 товаров (продукты, электроника, одежда).
3. Выполните запросы:
 - Показать все товары
 - Показать товары категории "Электроника"
 - Показать товары дешевле 1000 рублей
 - Показать товары, которых меньше 10 штук на складе
 - Показать товары определенного поставщика
 - Показать товары, отсортированные по количеству (от большего к меньшему)
 - Показать наименование и цену товаров категории "Продукты"

Вариант 3: Студенты

Создайте базу данных для учета успеваемости студентов.

Задание:

1. Создайте таблицу **Успеваемость** со следующими полями:
 - id (уникальный идентификатор)
 - фамилия (текст, обязательно)
 - имя (текст, обязательно)
 - предмет (текст)
 - оценка (целое число)
 - дата_сдачи (дата)
 - преподаватель (текст)
2. Добавьте в таблицу 7 записей о сдаче экзаменов.
3. Выполните запросы:
 - Показать все записи
 - Показать оценки по предмету "Математика"
 - Показать оценки 4 и 5
 - Показать сдачи за октябрь 2023 года
 - Показать студентов определенного преподавателя
 - Показать записи, отсортированные по дате (новые сначала)
 - Показать фамилии и оценки студентов, сдавших "Физику"

Общие требования:

1. Используйте одну из рекомендованных систем
2. Напишите SQL-код для создания таблицы
3. Напишите SQL-код для добавления данных
4. Напишите все требуемые запросы SELECT
5. Проверьте выполнение каждого запроса
6. Сохраните результаты работы

Отчет должен содержать:

1. Титульный лист
2. Выбранную систему для работы
3. Полный SQL-код:
 - Создание таблицы
 - Добавление данных
 - Все запросы SELECT
4. Результаты выполнения каждого запроса (скриншоты или текстовый вывод)
5. Ответы на контрольные вопросы
6. Выводы о проделанной работе

6. КРИТЕРИИ ОЦЕНКИ

Оценка «5» (Отлично):

- Правильно создана таблица с корректными типами данных
- Добавлено 7+ записей с разнообразными данными
- Написаны все требуемые запросы SELECT

- Запросы работают без ошибок
- Использованы различные условия и сортировка
- Результаты запросов соответствуют условиям
- Работа сдана в срок, отчет оформлен полностью

Оценка «4» (Хорошо):

- Таблица создана с незначительными ошибками
- Добавлено 5-6 записей
- Написаны не все запросы или с небольшими ошибками
- Результаты в основном соответствуют условиям
- Отчет оформлен с небольшими замечаниями

Оценка «3» (Удовлетворительно):

- Таблица создана с серьезными ошибками
- Добавлено менее 5 записей
- Написаны только простые запросы
- Много ошибок в синтаксисе
- Отчет оформлен не полностью

Оценка «2» (Неудовлетворительно):

- SQL-код не написан
- Таблица не создана
- Запросы не выполнены
- Работа не сдана или сдана не в срок

Рекомендуемое время выполнения: 2 академических часа

ПРИМЕР РАБОТЫ В SQLITE ONLINE:

1. Откройте сайт <https://sqliteonline.com/>
2. В левой части выберите "SQLite"
3. В правом окне напишите SQL-код
4. Нажмите кнопку "Run" (выполнить)
5. Результаты появятся в нижней части экрана
6. Для сохранения: скопируйте код в текстовый файл

Пример начала работы:

sql

-- Первый запрос: создание таблицы

```
CREATE TABLE Студенты (
  id INTEGER PRIMARY KEY,
  фамилия TEXT NOT NULL,
  имя TEXT NOT NULL
);
```

-- Второй запрос: добавление данных

```
INSERT INTO Студенты (фамилия, имя) VALUES ('Иванов', 'Иван');
```

-- Третий запрос: выборка данных

```
SELECT * FROM Студенты;
```

Удачи в изучении SQL!

ПРАКТИЧЕСКАЯ РАБОТА_№13. Создание файла проекта базы данных. Создание интерфейса входной формы. Использование исполняемого файла проекта БД, приемы создания и управления.

по дисциплине «Основы проектирования баз данных»

Тема: Создание законченного проекта базы данных

1. ЦЕЛИ И ЗАДАЧИ ПРАКТИЧЕСКОЙ РАБОТЫ

Цель: Создать завершённый проект базы данных с интерфейсом пользователя.

Задачи:

1. Создать базу данных с таблицами в LibreOffice Base.
2. Разработать входную форму для приветствия пользователя.
3. Создать главное меню для навигации.
4. Настроить автоматический запуск проекта.
5. Создать ярлык для быстрого доступа.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.1. Проект базы данных

Проект БД - это законченная программа, которая включает:

- Базу данных с таблицами
- Формы для работы с данными
- Меню для навигации
- Отчеты для печати

2.2. Входная форма

Первое окно, которое видит пользователь. Содержит:

- Название системы
- Кнопки для выбора действия
- Информацию о программе

2.3. Автозапуск формы

В LibreOffice Base можно настроить:

- Какую форму открывать при запуске
- Скрытие лишних окон
- Установку пароля на базу

2.4. Ярлык для запуска

Можно создать ярлык, который:

- Открывает базу данных
- Автоматически запускает нужную форму
- Упрощает доступ к программе

3. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ

Задача: Создать проект "Школьная библиотека"

Шаг 1. Создание базы данных

1. Откройте LibreOffice Base
2. Создайте новую базу "Школьная_библиотека.odb"
3. Создайте таблицу "Книги":
 - ID (INTEGER, Автозначение: Да, PK)
 - Название (VARCHAR, 100)
 - Автор (VARCHAR, 50)
 - Класс (VARCHAR, 10)
 - Наличие (Да/Нет)

Шаг 2. Создание входной формы

1. В разделе "Формы" → "Создать форму в режиме дизайна"
2. Сделайте форму размером 12x8 см
3. Добавьте элементы:

text

Заголовок: "ШКОЛЬНАЯ БИБЛИОТЕКА"

Надпись: "Система учета учебников"

Кнопка 1: "Начать работу"

Кнопка 2: "Справка"

Кнопка 3: "Выход"

4. Сохраните как "Вход"

Шаг 3. Создание главного меню

1. Создайте новую форму

2. Добавьте:

text

Заголовок: "ГЛАВНОЕ МЕНЮ"

Кнопки:

- "Учебники"

- "Ученики"

- "Выдача книг"

- "Отчеты"

- "Назад"

3. Сохраните как "Меню"

Шаг 4. Создание рабочей формы

1. Создайте форму для таблицы "Книги"

2. Используйте Мастер форм

3. Добавьте кнопки для работы

Шаг 5. Настройка автозапуска

1. Сервис → Параметры → База данных

2. В разделе "При открытии":

○ Выберите "Открыть форму"

○ Выберите "Вход"

3. Нажмите "ОК"

Шаг 6. Создание ярлыка

1. Создайте ярлык LibreOffice Base на рабочем столе

2. Переименуйте в "Школьная библиотека"

3. В свойствах добавьте путь к файлу базы

Шаг 7. Тестирование проекта

1. Закройте базу данных

2. Откройте через ярлык

3. Проверьте работу всех кнопок

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое проект базы данных?

2. Для чего нужна входная форма?

3. Как создать главное меню?

4. Как настроить автозапуск формы?

5. Как создать ярлык для базы данных?

6. Какие элементы можно добавить на форму?

7. Как связать кнопки с действиями?

8. Как проверить работу проекта?

9. Можно ли поставить пароль на базу?

10. Как добавить справку в проект?

5. ЗАДАНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ

Вариант 1: Проект "Спортивная секция"

Создайте систему учета посещений.

Задание:

1. **База данных:**
 - Таблица "Спортсмены" (ФИО, возраст, группа, телефон)
 - Таблица "Тренировки" (дата, время, тема, тренер)
2. **Интерфейс:**
 - Входная форма с названием секции
 - Главное меню с 4 кнопками
 - Форма для регистрации спортсменов
 - Форма для учета тренировок
3. **Настройка:**
 - Автозапуск входной формы
 - Ярлык "Спортивная секция"

Вариант 2: Проект "Школьное питание"

Создайте систему учета завтраков.

Задание:

1. **База данных:**
 - Таблица "Ученики" (ФИО, класс, льготы)
 - Таблица "Питание" (дата, завтрак, обед)
2. **Интерфейс:**
 - Входная форма со школьной эмблемой
 - Главное меню на 5 разделов
 - Форма для учета питания
 - Форма для отчетов
3. **Настройка:**
 - Запуск с входной формы
 - Ярлык "Школьное питание"

Вариант 3: Проект "Классный журнал"

Создайте систему учета оценок.

Задание:

1. **База данных:**
 - Таблица "Ученики" (ФИО, дата рождения, родители)
 - Таблица "Оценки" (предмет, оценка, дата, учитель)
2. **Интерфейс:**
 - Входная форма с номером школы
 - Главное меню для учителя
 - Форма для ввода оценок
 - Форма для просмотра успеваемости
3. **Настройка:**
 - Автоматический запуск
 - Ярлык "Классный журнал"

Требования для всех вариантов:

1. Создать базу данных с 2 таблицами
2. Создать входную форму с 3 кнопками
3. Создать главное меню с 4-5 кнопками
4. Создать 1 рабочую форму
5. Настроить автозапуск
6. Создать ярлык на рабочем столе

Что должно быть в отчете:

1. Титульный лист
2. Структура базы данных
3. Скриншоты:
 - Входной формы

- Главного меню
 - Рабочей формы
 - Настроек автозапуска
 - Ярлыка на рабочем столе
 - 4. Описание работы проекта
 - 5. Ответы на вопросы
 - 6. Выводы
-

6. КРИТЕРИИ ОЦЕНКИ

Оценка «5» (Отлично):

- Создана база данных с 2 таблицами
- Есть входная форма и главное меню
- Создана рабочая форма для данных
- Настроен автозапуск
- Есть рабочий ярлык
- Все формы работают правильно
- Отчет полный и аккуратный

Оценка «4» (Хорошо):

- База создана, но 1 таблица
- Формы есть, но дизайн простой
- Автозапуск настроен
- Ярлык создан
- Есть небольшие ошибки
- Отчет с замечаниями

Оценка «3» (Удовлетворительно):

- Только база данных
- Одна форма создана
- Нет автозапуска
- Нет ярлыка
- Много ошибок
- Отчет неполный

Оценка «2» (Неудовлетворительно):

- Ничего не создано
- Работа не сдана

Советы:

1. Начните с создания базы
2. Потом делайте формы
3. Потом настройте автозапуск
4. В конце создайте ярлык
5. Тестируйте после каждого шага
6. Сохраняйте работу часто

Что должно получиться:

text

Ваш проект/

- Файл базы данных.odt
- Входная форма
- Главное меню
- Форма для работы
- Ярлык на рабочем столе

Удачи в создании первого проекта!

ПРАКТИЧЕСКАЯ РАБОТА_№14. Создание формы. Управление внешним видом формы.

Цель работы: Освоить технику создания дочерних форм для работы с данными. Научиться управлять свойствами форм и элементов управления для создания удобного и функционального интерфейса.

Теоретические сведения:

1. **Дочерние формы (Data Modules, Child Forms):** Формы, предназначенные для выполнения конкретных задач: редактирование таблиц, ввод заказов, просмотр отчетов. Могут быть модалными (блокирующими главное окно) или немодальными.

2. **Элементы управления для работы с данными:**

- TDBGrid: Табличное представление данных.
- TDBEdit, TDBMemo, TDBComboBox: Поля ввода, привязанные к конкретным полям таблицы.
- TDBNavigator: Панель навигации и основных операций (добавить, удалить, сохранить).

3. **Свойства элементов управления:**

- DataSource: Связь с источником данных.
- DataField: Привязка к конкретному полю набора данных.
- ReadOnly, Enabled, Visible: Управление доступностью и видимостью.
- Color, Font, Align: Настройка внешнего вида и расположения.

4. **Стили и темы:** Использование стандартных и пользовательских стилей для унификации интерфейса.

Пример решения задачи (Создание формы для таблицы catalogs):

1. Создать новую форму (File -> New -> Form).
2. Положить на форму компоненты: TFDQuery (с SQL SELECT * FROM catalogs), TDataSource, TDBGrid, TDBNavigator.
3. Связать компоненты: DBGrid1.DataSource -> DataSource1, DataSource1.DataSet -> FDQuery1, DBNavigator1.DataSource -> DataSource1.
4. В обработчике события OnShow формы активировать запрос: FDQuery1.Open();.
5. Настроить ширину столбцов и заголовки в DBGrid.

Контрольные вопросы:

1. Какие элементы управления чаще всего используются для создания интерфейса ввода/редактирования данных?
2. В чем разница между модалной и немодальной формой? Где целесообразно использовать каждую?
3. Как связать поле TDBEdit с полем cat_name таблицы catalogs?
4. Для чего нужен компонент TDataSource?

Задания к практической работе:

1. **Форма "Каталоги":** Создайте форму для просмотра и редактирования таблицы catalogs. Используйте TDBGrid и TDBNavigator. Добавьте возможность фильтрации списка по имени каталога через TEdit и кнопку "Найти".
2. **Форма "Книги":** Создайте более сложную форму для таблицы books. Используйте:
 - TDBGrid для общего списка.
 - Отдельные TDBEdit/TDBMemo для детального просмотра и редактирования выбранной записи.
 - TDBLookupComboBox для выбора каталога (поле b_cat_ID), чтобы пользователь выбирал название из списка, а не вводил код.
3. **Управление видом:** Настройте шрифты, цвета и выравнивание для всех созданных форм. Сделайте форму "Книги" немодальной, а форму "О программе" — модалной.
4. **Интеграция с главной формой:** Реализуйте открытие созданных форм из меню и с кнопок главной формы.

Критерии оценки:

- Созданы и корректно работают формы для таблиц catalogs и books.
- Реализована навигация и редактирование данных.
- Использован TDBLookupComboBox для удобного выбора каталога.
- Внешний вид форм настроен, интерфейс интуитивно понятен.

ПРАКТИЧЕСКАЯ РАБОТА №15. Задание значений и ограничений поля. Проверка введенного в поле значения. Отображение данных числового типа и типа дата.

Цель работы: Научиться контролировать целостность данных на уровне интерфейса приложения. Освоить методы валидации вводимых данных и форматированного отображения специальных типов.

Теоретические сведения:

- Валидация данных:** Проверка корректности данных на стороне клиента перед отправкой в БД.
 - На уровне поля:** События `OnChange`, `OnExit`. Проверка на пустое значение, формат (регулярные выражения), диапазон.
 - На уровне формы/записи:** Событие `BeforePost` набора данных. Комплексная проверка связанных полей.
- Ограничения:**
 - `TDBEdit.MaxLength`: Ограничение длины вводимой строки.
 - `TDBEdit.ReadOnly`: Запрет на редактирование.
 - `TMaskEdit`: Задание маски ввода (телефон, дата).
- Форматирование отображения:**
 - Для чисел (`b_price`): `TField.DisplayFormat` (например, `###0.00 руб.`).
 - Для дат (`o_time`): `TField.DisplayFormat` (например, `dd.mm.yyyy hh:nn`).
- Значения по умолчанию:** Задание начальных значений для новых записей (например, текущая дата, статус "passive").

Пример решения задачи (Проверка цены книги):

- Для `TDBEdit`, связанного с полем `b_price`, создать обработчик `OnExit`.
- В коде проверить значение:

```
pascal
var
  PriceVal: Double;
begin
  if TryStrToFloat(DBEditPrice.Text, PriceVal) then
  begin
    if (PriceVal < 0) or (PriceVal > 100000) then
    begin
      ShowMessage('Цена должна быть от 0 до 100 000!');
      DBEditPrice.SetFocus; // Вернуть фокус на поле
    end;
  end
  else
    ShowMessage('Введите корректное число!');
end;
```

Контрольные вопросы:

- Почему важна валидация данных на стороне клиента, если ограничения есть в БД?
- Какое событие компонента `TDataSet` наиболее подходит для комплексной проверки записи перед сохранением?
- Как отобразить цену в виде "1 250.50 руб."?
- Как задать автоматическую подстановку текущей даты в поле `o_time` при создании нового заказа?

Задания к практической работе:

- Валидация в форме "Книги":**
 - Установите `MaxLength` для поля "Название".
 - Реализуйте проверку года издания (`b_year`): значение должно быть между 1900 и текущим годом.

- Реализуйте проверку цены (b_price) и количества (b_count) на неотрицательное значение (используйте OnExit или BeforePost).

2. Форматирование:

- Настройте отображение цены книги с двумя десятичными знаками и разделителем тысяч.
- Настройте отображение даты-времени в заказе (o_time) в формате "ДД.ММ.ГГГГ ЧЧ:ММ".

3. Форма "Клиенты": Создайте форму для таблицы users.

- Используйте TDBComboBox или TDBRadioGroup для поля статуса (u_status), чтобы пользователь выбирал только из допустимых значений ('active', 'passive', 'lock', 'gold').
- Для поля u_email реализуйте простую проверку формата "наличие символа @".

4. Значения по умолчанию: Настройте для новой книги установку количества (b_count) = 0, а для нового заказа — установку текущей даты-времени в поле o_time.

Критерии оценки:

- Реализована проверка корректности вводимых числовых данных и дат.
- Данные отформатированы для удобного отображения.
- Для поля статуса пользователя используется компонент выбора из списка.
- Заданы корректные значения по умолчанию для новых записей.

ПРАКТИЧЕСКАЯ РАБОТА_№16. Создание и модификация таблиц БД. Выборка данных из БД. Модификация содержимого БД. Обработка транзакций. Использование функций защиты для БД.

Цель работы: Интегрировать знания по построению запросов и управлению данными в графическом приложении. Реализовать сложную логику работы с данными (транзакции, поиск, отчеты) через интерфейс.

Теоретические сведения:

1. **Динамические запросы:** Использование компонента `TFDQuery` для выполнения произвольных SQL-команд (`SELECT`, `INSERT`, `UPDATE`, `DELETE`) по требованию пользователя (например, по нажатию кнопки).
2. **Поиск и фильтрация:** Формирование параметризованных запросов (`WHERE b_name LIKE :SearchName`) для реализации поиска по форме.
3. **Транзакции в приложении:** Группировка нескольких операций обновления БД (например, создание заказа и уменьшение остатка книги) в одну логическую единицу с помощью методов `StartTransaction`, `Commit`, `Rollback` компонента подключения.
4. **Отчеты и выборки:** Использование запросов с агрегатными функциями (`SUM`, `COUNT`, `GROUP BY`) для вывода сводной информации в `TDBGrid` или `TStringGrid`.

Пример решения задачи (Оформление заказа с транзакцией):

1. На форме оформления заказа создать две кнопки: "Подтвердить" и "Отменить".

2. В коде кнопки "Подтвердить":

pascal

```
FDConnection1.StartTransaction; // Начало транзакции
```

```
try
```

```
    // 1. Вставить запись в таблицу orders
```

```
    FDQueryOrderInsert.ParamByName('p_user_id').AsInteger := ...;
```

```
    FDQueryOrderInsert.ParamByName('p_book_id').AsInteger := ...;
```

```
    FDQueryOrderInsert.ExecSQL;
```

```
    // 2. Уменьшить количество книг на складе
```

```
    FDQueryUpdateStock.ParamByName('p_book_id').AsInteger := ...;
```

```
    FDQueryUpdateStock.ParamByName('p_count').AsInteger := ...;
```

```
    FDQueryUpdateStock.ExecSQL;
```

```
    FDConnection1.Commit; // Если все успешно, фиксируем
```

```
    ShowMessage('Заказ оформлен!');
```

```
except
```

```
on E: Exception do
```

```
begin
```

```
    FDConnection1.Rollback; // Если ошибка, откатываем
```

```
    ShowMessage('Ошибка: ' + E.Message);
```

```
end;
```

```
end;
```

Контрольные вопросы:

1. Как с помощью `TFDQuery` выполнить запрос, который не возвращает набор данных (например, `UPDATE`)?
2. Для чего нужны параметризованные запросы? В чем их преимущество перед конкатенацией строк SQL?
3. Опишите логику транзакции при оформлении возврата книги (удаление заказа, увеличение остатка).
4. Как в приложении можно реализовать вывод отчета "Топ-5 самых продаваемых книг"?

Задания к практической работе:

1. **Форма "Заказы":** Создайте форму для работы с таблицей `orders`. Реализуйте:
 - Мастер оформления нового заказа с `TDBLookupComboBox` для выбора клиента и книги.
 - Кнопку "Оформить заказ", в обработчике которой выполняется **транзакция**, включающая `INSERT` в `orders` и `UPDATE` (уменьшение) поля `b_count` в таблице `books`.
2. **Поиск и фильтры:**
 - На форме "Книги" реализуйте сложный фильтр: поиск по названию (частично) и выбор каталога из выпадающего списка.
 - Используйте параметризованный запрос для формирования условия `WHERE`.
3. **Отчеты:** Создайте отдельную форму или вкладку для отчетов:
 - "Продажи по каталогам": `SELECT cat_name, COUNT(*) as cnt FROM books JOIN catalogs ... JOIN orders ... GROUP BY cat_name`.
 - "Активные клиенты": список клиентов со статусом 'active' и 'gold', сделавших более 1 заказа.
4. **Повышение безопасности:** Перенесите логин и пароль для подключения к БД из кода программы в зашифрованный конфигурационный файл. Реализуйте простую форму входа в приложение (не в БД) для оператора.

Критерии оценки:

- Реализовано оформление заказа с использованием транзакции.
- Создана функциональная система поиска/фильтрации книг.
- Реализованы и работают минимум два отбора (отчета) с группировкой данных.
-

УЧЕБНО-МЕТОДИЧЕСКОЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ УЧЕБНОГО ПРЕДМЕТА

Требования к минимальному материально-техническому обеспечению

Программа учебной дисциплины реализуется на базе лаборатории программного обеспечения компьютерных сетей, программирования и баз данных.

Оборудование кабинета:

- посадочные места по количеству обучающихся;
- рабочее место преподавателя;
- комплект учебно-наглядных пособий, в т.ч. на электронных носителях.

Технические средства обучения:

- компьютеры с лицензионным программным обеспечением на каждом посадочном месте обучающихся и на рабочем месте преподавателя.

Информационное обеспечение обучения

Рекомендуемая литература

Для студентов

1. Голицына, О. Л. Основы проектирования баз данных : учебное пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. — 2-е изд., перераб. и доп. — Москва : ФОРУМ : ИНФРА-М, 2021. — 416 с. — (Среднее профессиональное образование). - ISBN 978-5-91134-655-3. Электронный ресурс. Режим доступа: сетевой <https://znanium.com/catalog/product/1190668> (дата обращения: 08.04.2021).

Для преподавателей

2. Илющечкин, В. М. Основы использования и проектирования баз данных : учебник для среднего профессионального образования / В. М. Илющечкин. — испр. и доп. — Москва : Издательство Юрайт, 2020. — 213 с. — (Профессиональное образование). ISBN 978-5-534-01283-5. — Электронный ресурс. Режим доступа: сетевой

Интернет-ресурсы

Электронный ресурс. Режим доступа: сетевой

3. URL: <https://urait.ru/bcode/452874> (дата обращения: 08.04.2021).
4. Стружкин, Н. П. Базы данных: проектирование : учебник для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2020. — 477 с. — (Профессиональное образование). — ISBN 978-5-534-11635-9.
5. URL: <https://urait.ru/bcode/457135> (дата обращения: 08.04.2021). Нестеров, С.А. Базы данных : учебник и практикум для среднего профессионального образования / С.А. Нестеров. — Москва : Издательство Юрайт, 2020. — 230 с. — (Профессиональное образование). — ISBN 978-5-534-11629-8.
6. Электронный ресурс. Режим доступа: сетевой
7. URL: <https://urait.ru/bcode/457142> (дата обращения: 08.04.2021).
8. Стружкин, Н. П. Базы данных: проектирование. Практикум : учебное пособие для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2020. — 291 с. — (Профессиональное образование). — ISBN 978-5-534-08140-4. — Электронный ресурс. Режим доступа: сетевой URL: <https://urait.ru/bcode/455865> (дата обращения: 08.04.2021)