Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

Уфимский университет науки и технологий

Факультет информатики и робототехники

Кафедра вычислительной математики и кибернетики

Отчет к лабораторной работе №4

По дисциплине «Проектирование и конструирование ПО»

По теме ««Проектирование БД. Проектирование пользовательского интерфейса»

Студенты группы ПРО-232б: Силантьев К.Ю., Смольников А.А.

Проверил: преподаватель

Юдинцев Б.С.

Уфа 2023

**Цель лабораторной работы:**

* Ознакомление с основными методами проектирования базы данных (БД).
* Ознакомление с методами проектирования пользовательского интерфейса.

**Проектирование БД**

1. Изучить дополнительный материал по теории проектирования БД (см. раздел «дополнительная информация») спроектировать БД приложения для хранения обработанных данных.
   1. БД должна содержать таблицу с информацией об обработанных файлах (минимальный набор колонок: название файла, дата обработки).
   2. Таблица обработанных файлов должна ограничивать основную таблицу с данными по внешнему ключу (FOREIGN KEY).
2. Написать SQL-скрипт для создания структуры спроектированной БД (см. пример SQL-скрипт в архиве **pikpo4\_python.zip**).
3. Изучить код примера из архива **pikpo4\_python.zip**. На основе данного примера реализовать необходимые **CRUD** (Create, Read, Update, Delete) операции для работы с БД (см. ссылку на материал по основам SQL). **Загрузить код приложения на GitHub.**
4. Проверить выполнение CRUD-операций на тестовой БД (только SQLite).

**Проектирование пользовательского интерфейса**

Во время выполнения лабораторной работы необходимо описать ожидаемое поведение разрабатываемой системы с точки зрения внешнего по отношению к ней пользователя, то есть осуществить "конструирование" внешних взаимодействий будущей ИС с пользователем без конкретизации его внутреннего устройства.

1. **Определить структуру проектируемого пользовательского интерфейса** (визуальное оформление, отвечающее за представление информации пользователю; функциональные возможности системы, включающие набор возможностей для эффективного выполнения профессиональной деятельности; техники взаимодействия пользователя с системой + дополнительные функциональные возможности системы) с учетом задачи (лекции, рекомендации Приложение 1, дополнительные функциональные возможности системы (пример) в Приложении 3).
2. **Определить стили пользовательского интерфейса** (графический (GUI, web-интерфейс (WUI), объектно-ориентированный интерфейс) с учетом задачи (лекции, рекомендации Приложение 1).
3. **Определиться с размещением элементов пользовательского интерфейса** (кнопки, иконки, выпадающие списки, поля для записи текста и пр.) (Лекции, рекомендации Приложение 1)**.**
4. **Написать Требования к интерфейсу пользователя.**
5. **Разработать взаимодействие разрабатываемой программы с пользователем**: сценарий (можно диаграммой последовательности, диаграммой взаимодействия), экранные формы, набор подсказок (перечисление), и пр.
6. **Ознакомиться с методическим материалом по базовой верстке веб-страниц**.

На основе **п.1-5** разработать (сверстать) основные html-страницы для вашего приложения, используя CSS-стили и HTML5-модель верстки

Код программы загружен на GitHub: <https://github.com/cowboboy/twitchInfoTelegramBot>

Ознакомление с основными элементами определения, представления, проектирования и моделирования программных систем с помощью языка UML.

**Ход работы:**

1. **ER-диаграмма спроектированной БД:**

Изображение выглядит как текст, снимок экрана, в помещении

Автоматически созданное описание

Рисунок 1. ER-диаграмма спроектированной БД

1. **Листинг SQL-скрипта для создания структуры БД:**

create table source\_files (

id integer PRIMARY KEY autoincrement,

filename varchar(255) NOT NULL,

processed datetime

);

CREATE TABLE wood\_removal(

id INTEGER PRIMARY KEY autoincrement,

country CHAR(128),

year1990 INT,

year1991 INT,

year1992 INT,

year1993 INT,

year1994 INT,

year1995 INT,

year1996 INT,

year1997 INT,

year1998 INT,

year1999 INT,

year2000 INT,

year2001 INT,

year2002 INT,

year2003 INT,

year2004 INT,

year2005 INT,

year2006 INT,

year2007 INT,

year2008 INT,

year2009 INT,

year2010 INT,

year2011 INT,

source\_file integer NOT NULL,

CONSTRAINT fk\_source\_files

FOREIGN KEY (source\_file)

REFERENCES source\_files(id)

ON DELETE CASCADE

);

1. **Листинг основных функций для работы с БД**

def select\_all\_from\_source\_files(connector: StoreConnector) -> List[tuple]:

query = f'SELECT \* FROM source\_files ORDER BY processed DESC'

result = connector.execute(query).fetchall()

return result

def insert\_into\_source\_files(connector: StoreConnector, filename: str):

""" Вставка в таблицу обработанных файлов """

now = datetime.now() # текущая дата и время

date\_time = now.strftime("%Y-%m-%d %H:%M:%S")

query = f'INSERT INTO source\_files (filename, processed) VALUES (\'{filename}\', \'{date\_time}\')'

result = connector.execute(query)

return result

def insert\_rows\_into\_processed\_data(connector: StoreConnector, dataframe: DataFrame):

rows = dataframe.to\_dict('records')

files\_list = select\_all\_from\_source\_files(connector)

last\_file\_id = files\_list[0][0]

if len(files\_list) > 0:

for row in rows:

connector.execute(f'INSERT INTO wood\_removal ('

f'country, '

f'year1990, '

f'year1991, '

f'year1992, '

f'year1993, '

f'year1994, '

f'year1995, '

f'year1996, '

f'year1997, '

f'year1998, '

f'year1999, '

f'year2000, '

f'year2001, '

f'year2002, '

f'year2003, '

f'year2004, '

f'year2005, '

f'year2006, '

f'year2007, '

f'year2008, '

f'year2009, '

f'year2010, '

f'year2011, '

f'source\_file)'

f'VALUES ('

f'\'{row["country"]}\', '

f'\'{row["1990"]}\', '

f'\'{row["1991"]}\', '

f'\'{row["1992"]}\', '

f'\'{row["1993"]}\', '

f'\'{row["1994"]}\', '

f'\'{row["1995"]}\', '

f'\'{row["1996"]}\', '

f'\'{row["1997"]}\', '

f'\'{row["1998"]}\', '

f'\'{row["1999"]}\', '

f'\'{row["2000"]}\', '

f'\'{row["2001"]}\', '

f'\'{row["2002"]}\', '

f'\'{row["2003"]}\', '

f'\'{row["2004"]}\', '

f'\'{row["2005"]}\', '

f'\'{row["2006"]}\', '

f'\'{row["2007"]}\', '

f'\'{row["2008"]}\', '

f'\'{row["2009"]}\', '

f'\'{row["2010"]}\', '

f'\'{row["2011"]}\', '

f'{last\_file\_id})')

print('Data was inserted successfully')

else:

print('File records not found. Data inserting was canceled.')

1. **Скриншот части содержимого основной таблицы данных и связанных таблиц.**

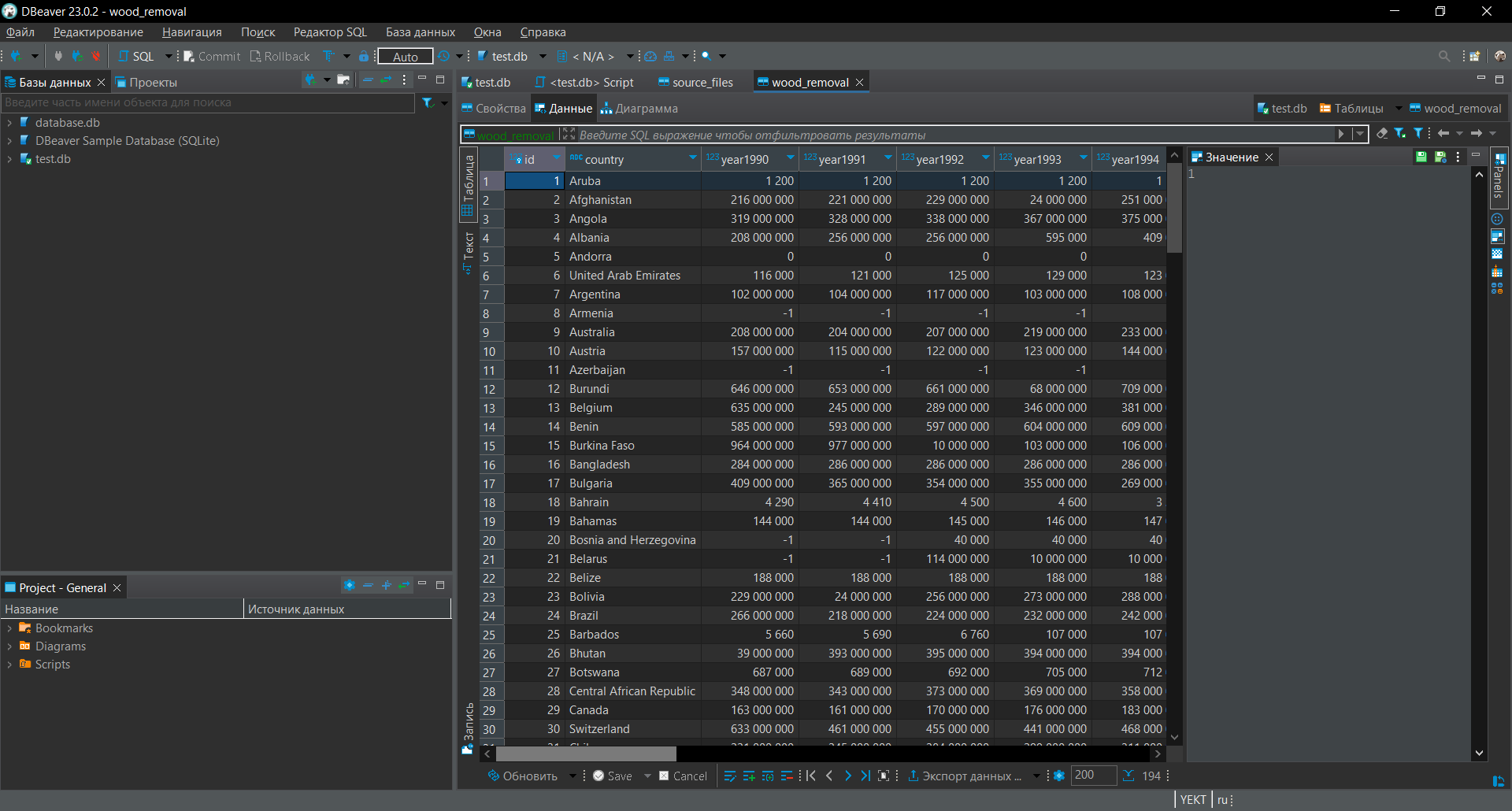


Рисунок 2. Скриншот таблицы wood\_removal

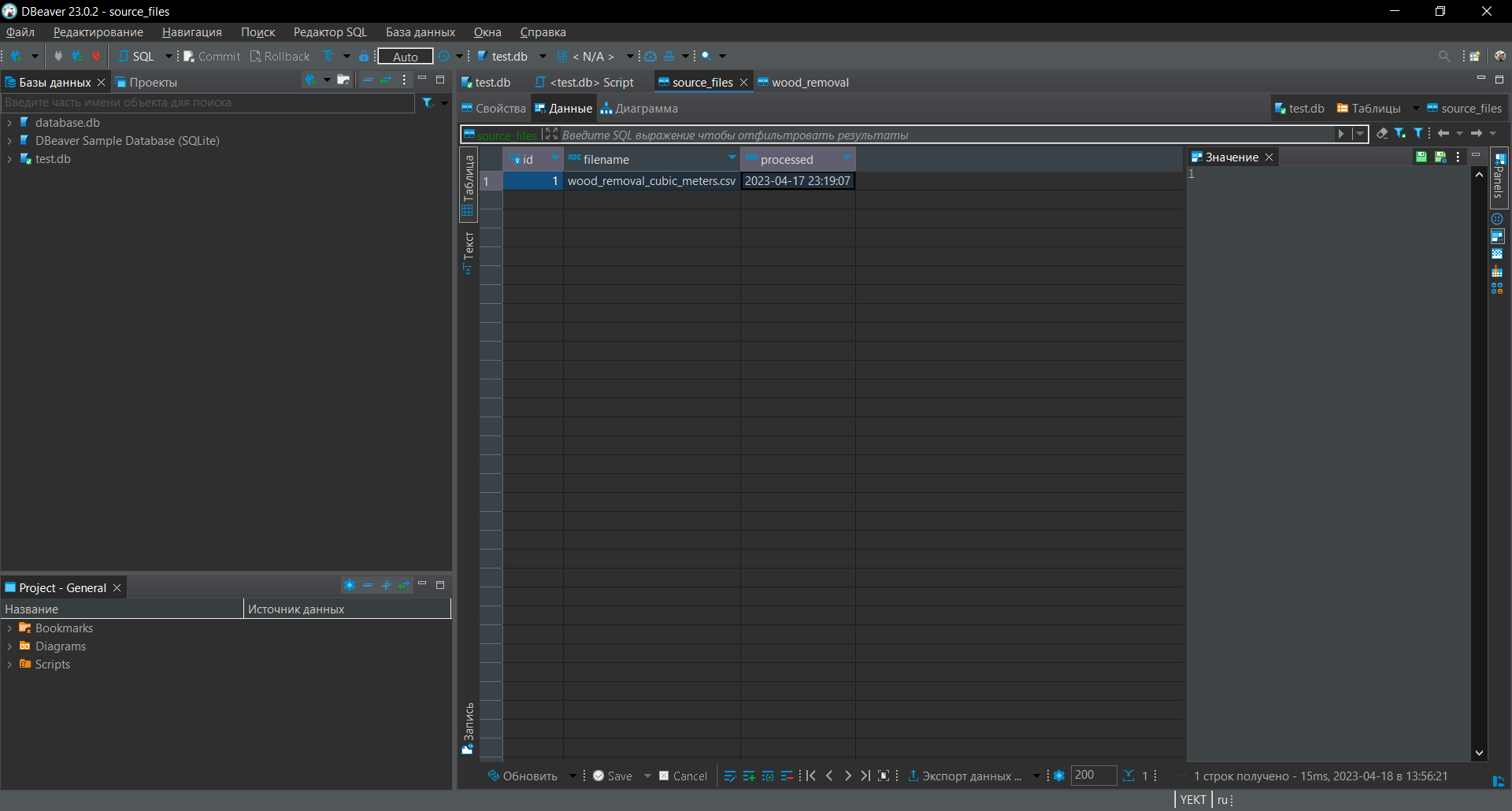


Рисунок 3. Скриншот таблицы source\_files

**Вывод**

В данной лабораторной работе мы были ознакомились с основными методами проектирования базы данных (БД), с методами проектирования пользовательского интерфейса.