



МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Институт №3 – «Системы управления, информатика и электроэнергетика»
Кафедра 307 – «Цифровые технологии и информационные системы»

ОТЧЁТ
О ЛАБОРАТОРНОЙ РАБОТЕ
по теме:
ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС ДЛЯ РАБОТЫ С БАЗОЙ ДАННЫХ

Принял:

Должность руководителя

подпись, дата

Фамилия И.О.

Выполнил:

Должность исполнителя

подпись, дата

Фамилия И.О.

РЕФЕРАТ

Отчёт 22 с., 25 рис., 2 ист.

БАЗЫ ДАННЫХ, POSTGRESQL, PYTHON, PYQT5, ТЕЛЕФОННЫЙ СПРАВОЧНИК

Объектом исследования является база данных телефонного справочника.

Цель работы — разработать базу данных телефонного справочника на PostgreSQL и программу с графическим интерфейсом для работы с ней.

Лабораторная работа, оформлена в соответствии с ГОСТ 7.32–2017 [1].

Исходный код проекта доступен в репозитории [2].

СОДЕРЖАНИЕ

Введение	4
1 Структура базы данных	5
1.1 Таблица записей	6
1.2 Родительские таблицы	7
1.2.1 Таблица имен	7
1.2.2 Таблица фамилий	7
1.2.3 Таблица отчеств	7
1.2.4 Таблица улиц	7
2 Программная реализация	8
2.1 Основные технологии	8
2.2 Графический интерфейс	8
2.3 Логирование	10
2.4 Конфигурация	11
2.5 Дополнительные функции	12
3 Тестирование	13
3.1 Добавление записи	13
3.2 Удаление записи	14
3.3 Редактирование записи	15
3.4 Поиск	16
3.5 Генерация данных	17
3.6 Сброс данных	18
3.7 Нарушение ссылочной целостности	19
3.8 Редактирование родительских таблиц	20
Заключение	21
Список использованных источников	22

ВВЕДЕНИЕ

Базы данных являются неотъемлемой частью современных информационных систем. Они позволяют хранить, обрабатывать и получать доступ к большим объемам данных. Телефонный справочник, представляющий собой базу данных контактов, является одним из распространенных примеров применения баз данных. Удобный пользовательский интерфейс существенно упрощает взаимодействие с базой данных и повышает эффективность работы.

В данной работе разрабатывается база данных телефонного справочника на основе PostgreSQL и создается программа с графическим интерфейсом на языке Python с использованием библиотеки PyQt5. Цель работы – реализовать функциональный интерфейс, позволяющий выполнять основные операции с базой данных: добавление, удаление, редактирование записей, поиск, а также редактирование родительских таблиц.

Работа состоит из нескольких глав. В первой главе описывается структура разработанной базы данных. Во второй главе рассматривается функционал разработанной программы. В третьей главе проводится тестирование основных функций программы. В заключении подводятся итоги работы и формулируются основные выводы.

1 Структура базы данных

База данных телефонного справочника выполнена используя СУБД PostgreSQL с помощью pgAdmin 4. Она состоит из дочерней таблицы entries и трёх родительских таблиц: names (имена), surnames (фамилии) и patronymics (отчества).

Структура базы данных представлена на рисунке 1.

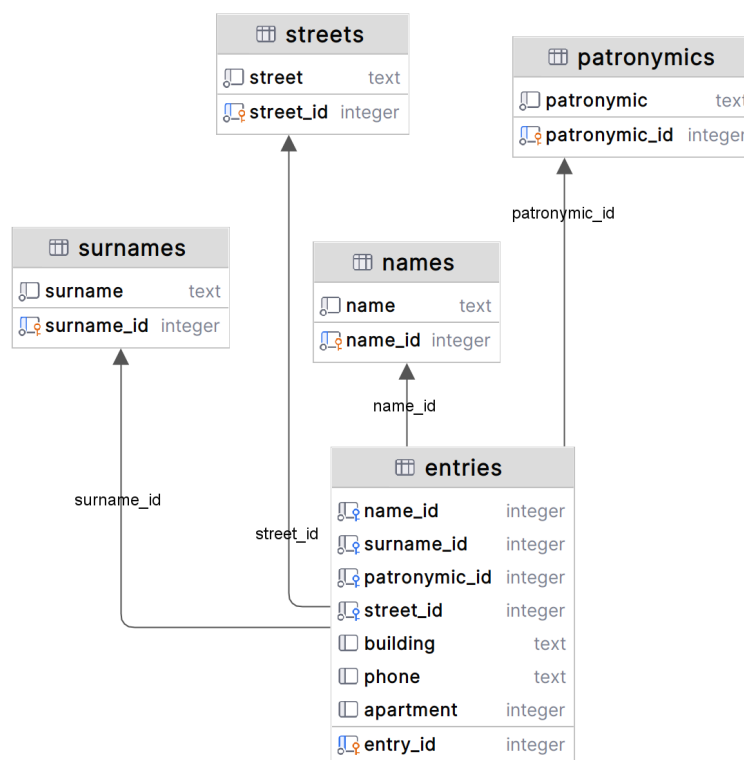


Рисунок 1 — Структура базы данных

Центральной таблицей является таблица entries (записи справочника). Для каждой записи хранятся имя, фамилия, отчество, город, улица, дом, номер квартиры и номер телефона. При этом для повышения эффективности и обеспечения целостности данных имена, фамилии, отчества и улицы хранятся в отдельных родительских таблицах: names, surnames, patronymics и streets соответственно.

В таблице entries хранятся идентификаторы (внешние ключи) соответствующих записей из родительских таблиц и остальные данные. Родительские таблицы не имеют внешних ключей, они только хранят необходимые записи.

1.1 Таблица записей

Таблица `entries` является центральной таблицей базы данных и хранит информацию о каждой записи в телефонном справочнике. Она связана с другими таблицами через внешние ключи, обеспечивая целостность данных и предотвращая дублирование информации.

- `entry_id` (integer, primary key) – уникальный идентификатор записи в таблице.
- `name_id` (integer, foreign key, not null) – внешний ключ, ссылающийся на таблицу `names`. Содержит идентификатор имени абонента.
- `surname_id` (integer, foreign key, not null) – внешний ключ, ссылающийся на таблицу `surnames`. Содержит идентификатор фамилии абонента.
- `patronymic_id` (integer, foreign key, not null) – внешний ключ, ссылающийся на таблицу `patronymics`. Содержит идентификатор отчества абонента.
- `street_id` (integer, foreign key, not null) – внешний ключ, ссылающийся на таблицу `streets`. Содержит идентификатор улицы.
- `building` (text) – номер дома.
- `phone` (text) – номер телефона.
- `apartment` (integer) – номер квартиры.

В таблице `entries` поля `name_id`, `surname_id`, `patronymic_id` и `street_id` являются внешними ключами. Они обеспечивают ссылочную целостность, гарантируя, что каждая запись в `entries` ссылается на существующую запись в соответствующей родительской таблице.

Для операций удаления используется правило `RESTRICT`. Оно предотвращает удаление записей из родительской таблицы, если на них существуют ссылки в таблице `entries`.

Для операций обновления применяется правило `CASCADE`, которое обеспечивает автоматическое обновление значений внешних ключей в таблице `entries`, если значение первичного ключа в родительской таблице изменяется.

1.2 Родительские таблицы

1.2.1 Таблица имен

Таблица names содержит список имен абонентов.

- name_id (integer, primary key) – уникальный идентификатор.
- name (text, not null) – строка с именем.

1.2.2 Таблица фамилий

Таблица surnames содержит список фамилий абонентов.

- surname_id (integer, primary key) – уникальный идентификатор.
- surname (text, not null) – строка с фамилией.

1.2.3 Таблица отчеств

Таблица patronymics содержит список отчеств абонентов.

- patronymic_id (integer, primary key) – уникальный идентификатор.
- patronymic (text, not null) – строка с отчеством.

1.2.4 Таблица улиц

Таблица streets содержит список улиц.

- street_id (integer, primary key) – уникальный идентификатор.
- street (text, not null) – название улицы.

2 Программная реализация

Программа для работы с телефонным справочником была разработана с использованием библиотеки PyQt5, обеспечивающей графический пользовательский интерфейс.

Основные функции приложения — создание, просмотр, редактирование и удаление данных через удобный интерфейс

2.1 Основные технологии

Программа использует следующие технологии:

- PyQt5 — создание графического интерфейса;
- psycorg2 — взаимодействие с базой данных PostgreSQL;
- loguru — логирование событий и ошибок;
- python-dotenv — управление конфигурацией через файл .env;
- pydantic — валидация входных данных;
- mimesis — генерация случайных данных.

2.2 Графический интерфейс

Интерфейс включает таблицу с данными, где пользователь может:

- просматривать текущие записи;
- добавлять новые записи через кнопку «Создать запись»;
- удалять или дублировать записи через контекстное меню.

Графический интерфейс программы представлен на рисунке 2.

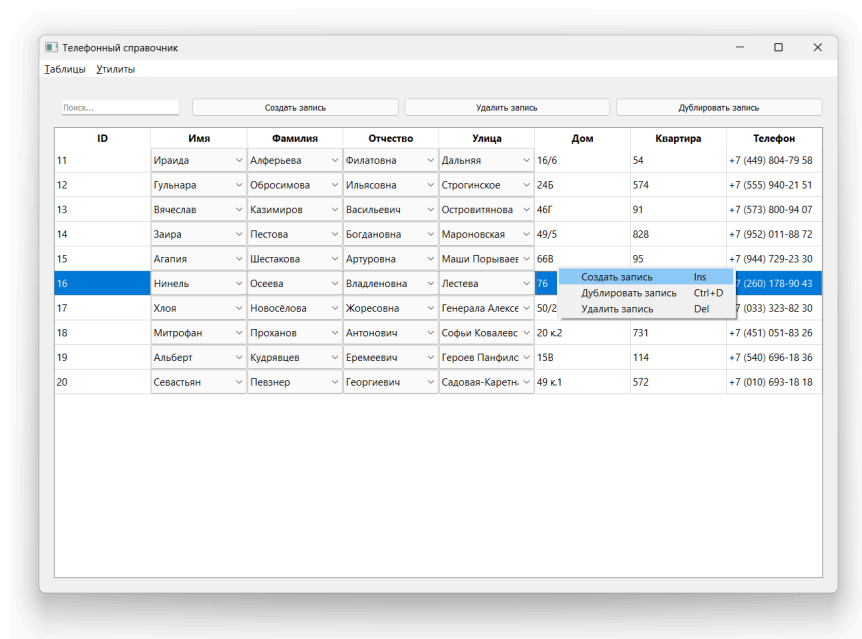


Рисунок 2 — Графический интерфейс программы

Интерфейс родительской таблицы изображён на рисунке 3.

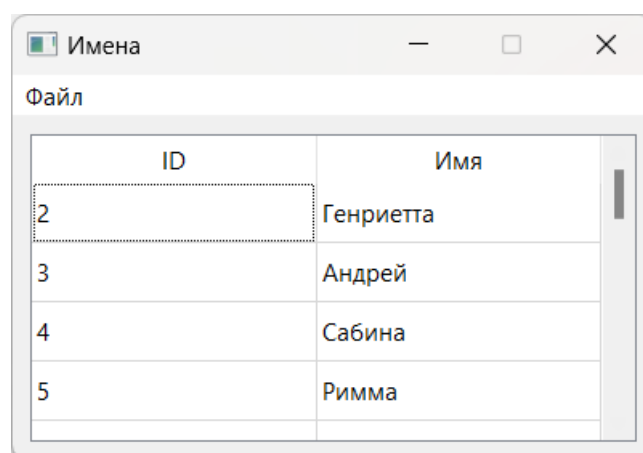


Рисунок 3 — Интерфейс для родительской таблицы

Внешние ключи в базе данных отображаются в интерфейсе как выпадающие списки (рисунок 4), позволяя пользователю выбирать доступные значения без необходимости ручного ввода.

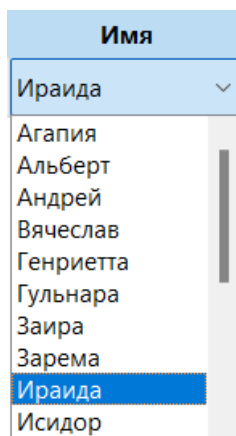


Рисунок 4 — Выпадающий список

Контекстное меню программы (рисунок 5) поддерживает работу с несколькими выбранными строками и предоставляет следующие опции с горячими клавишами:

- "Создать запись"(Ins)
- "Дублировать запись"(Ctrl+D)
- "Удалить запись"(Del)

Эти действия могут быть применены как к одной, так и к нескольким выделенным строкам таблицы.

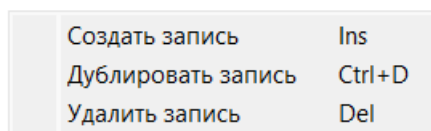


Рисунок 5 — Контекстное меню для управления записями

2.3 Логирование

Логирование осуществляется с помощью библиотеки loguru. Это позволяет отслеживать выполнение программы, фиксировать ошибки и выводить информацию о происходящих событиях.

```

2024-12-01 22:05:31.911 | INFO | database.connection:connect:20 - Подключение к базе данных...
2024-12-01 22:05:31.962 | SUCCESS | database.connection:connect:29 - Подключение к базе данных успешно выполнено
2024-12-01 22:05:32.049 | INFO | ui.table_base:load_data:228 - Загрузка данных...
2024-12-01 22:05:32.342 | INFO | database.base:get_all:38 - Получение записей из таблицы names
2024-12-01 22:05:32.342 | DEBUG | database.base:_log_query:35 - SQL запрос:
SELECT name_id, name FROM names
2024-12-01 22:05:32.345 | DEBUG | database.base:get_all:44 - Получено записей: 19
2024-12-01 22:05:32.346 | INFO | database.base:get_all:38 - Получение записей из таблицы surnames
2024-12-01 22:05:32.346 | DEBUG | database.base:_log_query:35 - SQL запрос:
SELECT surname_id, surname FROM surnames
2024-12-01 22:05:32.348 | DEBUG | database.base:get_all:44 - Получено записей: 19
2024-12-01 22:05:32.348 | INFO | database.base:get_all:38 - Получение записей из таблицы patronymics
2024-12-01 22:05:32.348 | DEBUG | database.base:_log_query:35 - SQL запрос:
SELECT patronymic_id, patronymic FROM patronymics
2024-12-01 22:05:32.349 | DEBUG | database.base:get_all:44 - Получено записей: 20
2024-12-01 22:05:32.350 | INFO | database.base:get_all:38 - Получение записей из таблицы streets
2024-12-01 22:05:32.350 | DEBUG | database.base:_log_query:35 - SQL запрос:
SELECT street_id, street FROM streets
2024-12-01 22:05:32.351 | DEBUG | database.base:get_all:44 - Получено записей: 20
2024-12-01 22:05:32.352 | SUCCESS | ui.table_base:load_headers:213 - Загрузка данных заголовков завершена
2024-12-01 22:05:32.352 | INFO | database.entry:get_all:24 - Получение всех записей из таблицы entries co
связями
2024-12-01 22:05:35.034 | INFO | database.entry:duplicate:92 - Дублирование записей с ID ['25']
2024-12-01 22:05:35.036 | SUCCESS | database.entry:duplicate:110 - Дублирование записей успешно выполнено
2024-12-01 22:05:36.572 | INFO | database.base:create:85 - Создание новых записей с данными [{'name_id': 2,
'surname_id': 1, 'patronymic_id': 1, 'street_id': 1, 'building': '', 'apartment': 0, 'phone': 79123456789}]
2024-12-01 22:05:42.018 | INFO | database.base:delete:70 - Удаление записей с ID [['21'], ['24'], ['25']]
2024-12-01 22:05:42.018 | DEBUG | database.base:_log_query:35 - SQL запрос:
DELETE FROM entries WHERE entry_id = [['21'], ['24'], ['25']]
2024-12-01 22:05:42.019 | DEBUG | database.base:delete:76 - Удалено записей: 3

```

Рисунок 6 — Пример вывода логов программы

Логирование, помимо общей информации о работе программы, также выводит SQL-запросы, отправляемые в базу данных. Это позволяет наглядно продемонстрировать взаимодействие программы с базой данных, отследить ход выполнения запросов и, при необходимости, использовать эту информацию для отладки или анализа производительности. Вывод SQL-запросов в лог существенно упрощает понимание того, как программа работает с данными.

Пример вывода логов программы представлен на рисунке 6.

2.4 Конфигурация

Для управления параметрами подключения к базе данных используется файл `.env`. Это позволяет изменять настройки без необходимости редактирования исходного кода программы. Пример содержимого файла конфигурации представлен на рисунке 7.

```

DB_HOST=localhost
DB_PORT=5432
DB_USER=postgres
DB_PASSWORD=*****
DB_NAME=phone_table

```

Рисунок 7 — Пример содержимого файла конфигурации `.env`

2.5 Дополнительные функции

Для автоматического заполнения базы данных используется библиотека mimesis. Пользователь может указать количество записей через диалоговое окно (рисунок 8).

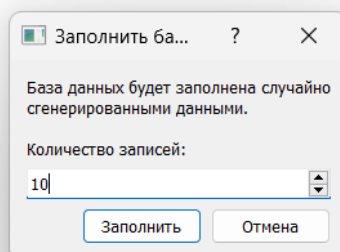


Рисунок 8 — Диалоговое окно для заполнения базы данных

Функция сброса базы данных удаляет все записи из таблиц и обнуляет счётчики PostgreSQL, используя команды SQL. Пользователь может отменить операцию сброса базы данных через диалоговое окно (рисунок 9).

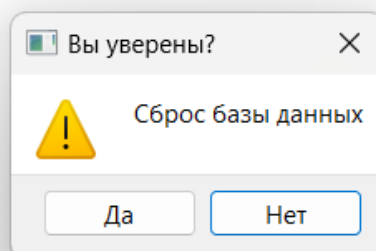


Рисунок 9 — Диалоговое окно сброса базы данных

3 Тестирование

3.1 Добавление записи

Тестирование показало корректную работу добавления новых записей в справочник. На рисунке 10 представлен интерфейс перед добавлением записи, а на рисунке 11 — после успешного добавления.

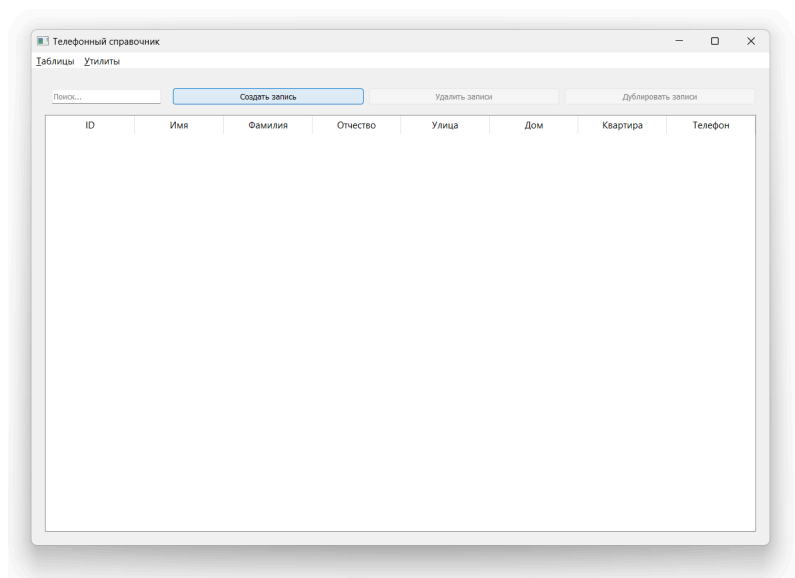


Рисунок 10 — Интерфейс перед добавлением записи

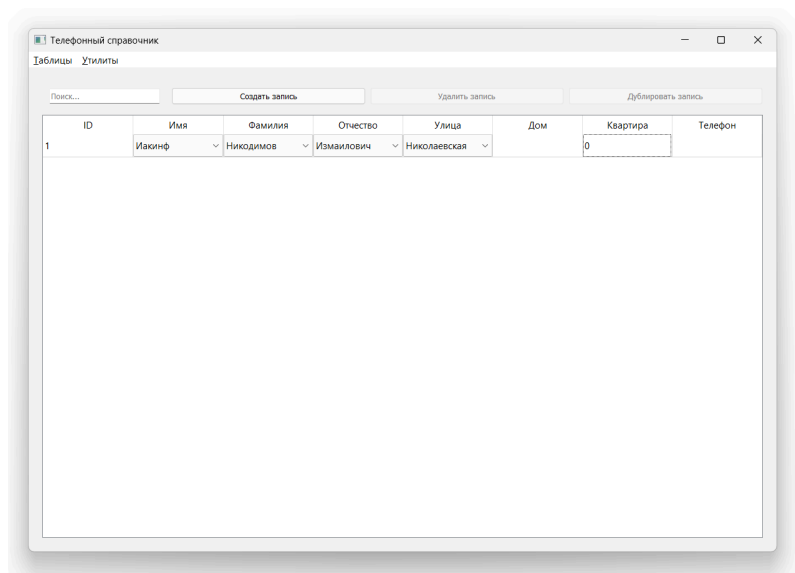


Рисунок 11 — Интерфейс после добавления записи

3.2 Удаление записи

Удаление записей работает корректно. На рисунке 12 представлен интерфейс перед удалением записи, а на рисунке 13 — после удаления.

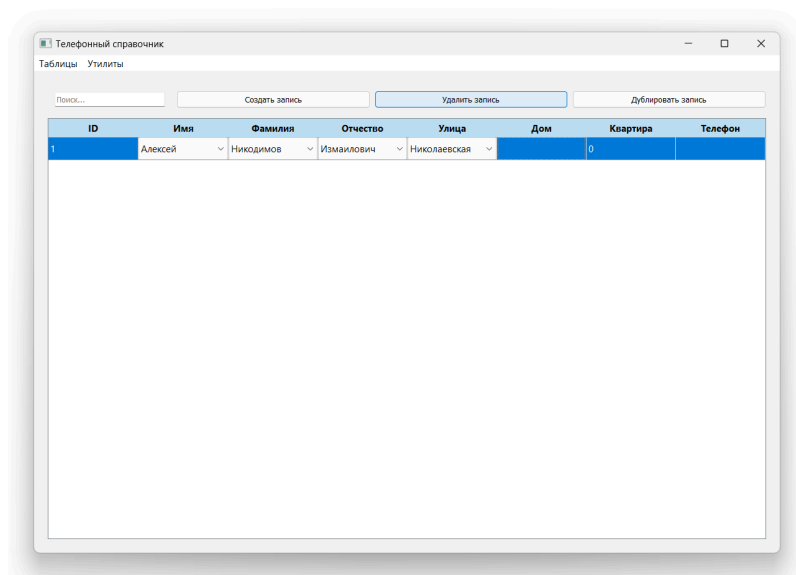


Рисунок 12 — Интерфейс перед удалением записи

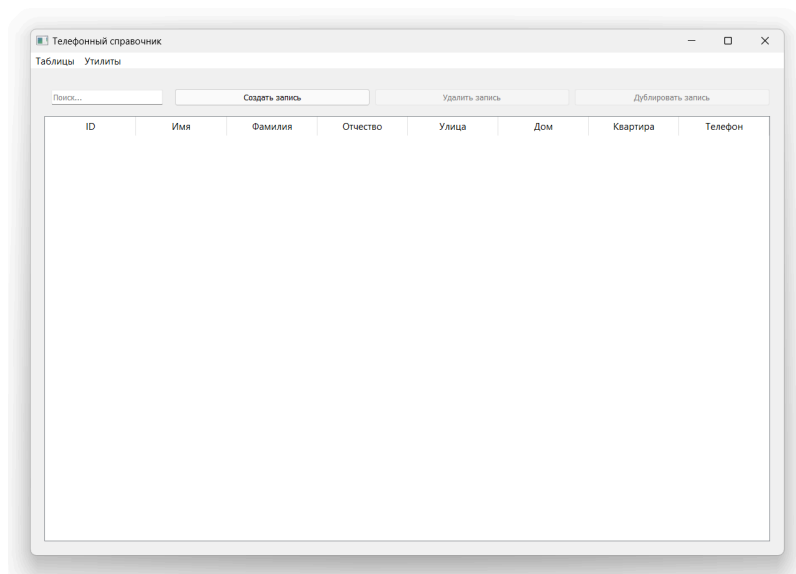


Рисунок 13 — Интерфейс после удаления записи

3.3 Редактирование записи

Изменение данных в существующих записях работает корректно. На рисунке 14 представлен интерфейс перед редактированием записи, а на рисунке 15 — после редактирования.

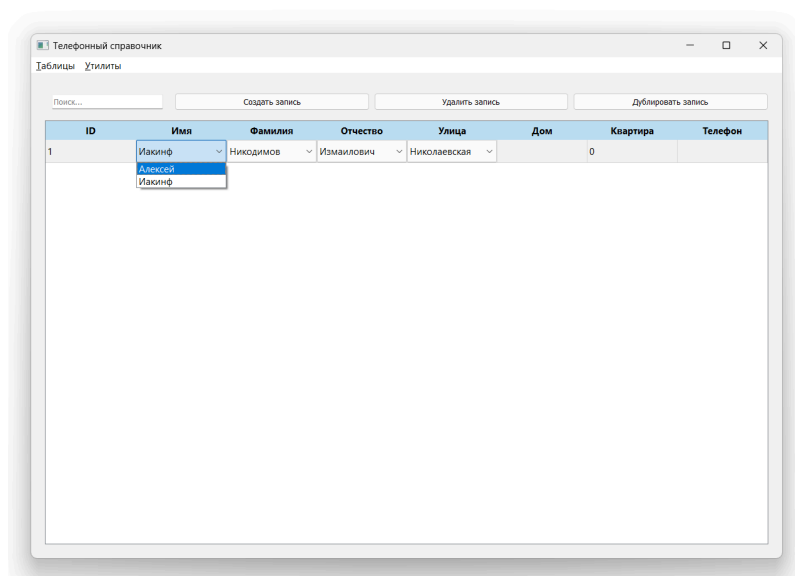


Рисунок 14 — Интерфейс перед редактированием записи

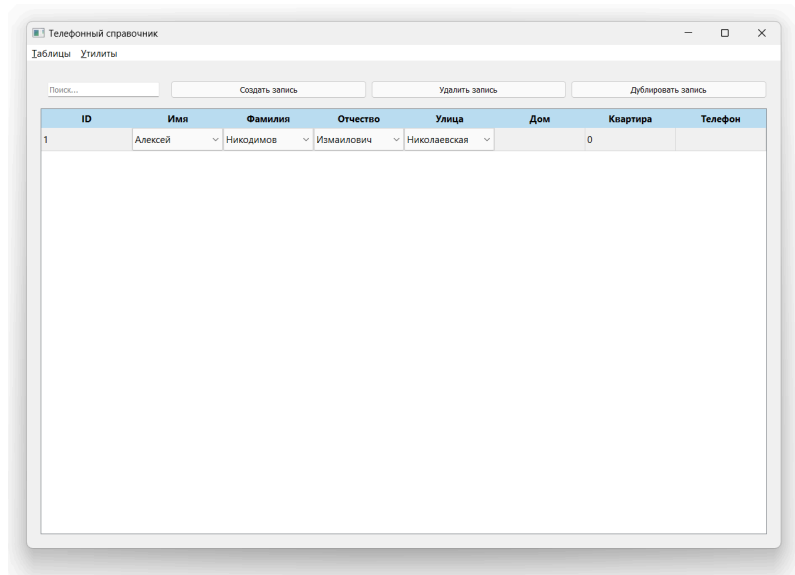


Рисунок 15 — Интерфейс после редактирования записи

3.4 Поиск

Тестирование поиска по различным критериям показало корректность работы функции поиска. Поиск осуществляется по всем колонкам таблицы, что позволяет быстро находить нужные записи по любому из доступных атрибутов. На рисунке 16 представлен интерфейс перед поиском, а на рисунке 17 — после ввода текста в поле для поиска.

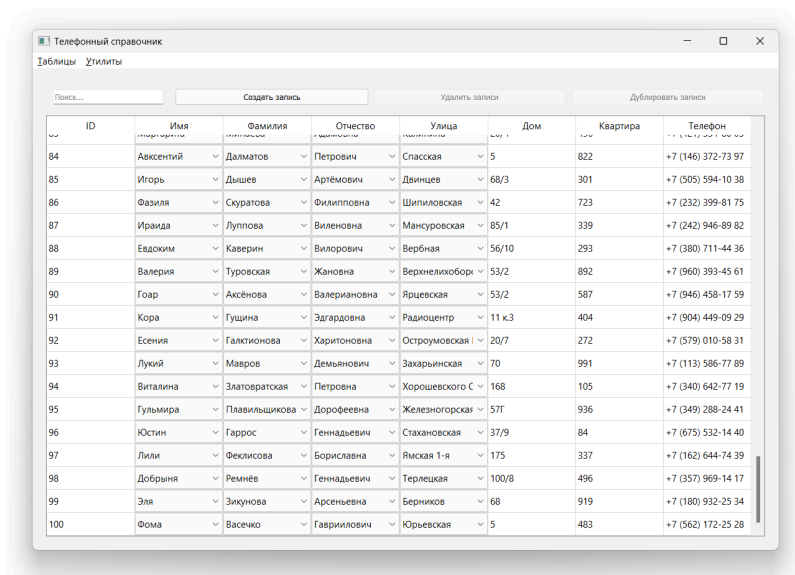


Рисунок 16 — Интерфейс перед поиском

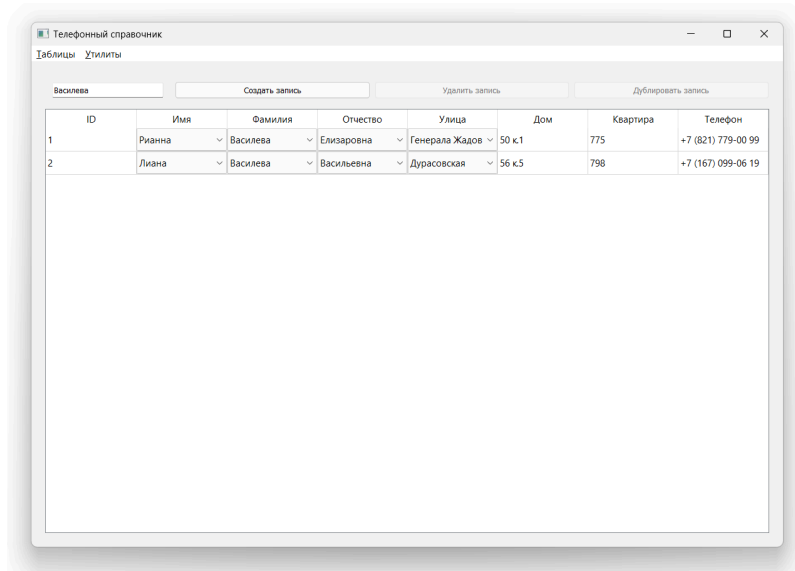


Рисунок 17 — Интерфейс после поиска

3.5 Генерация данных

Генерация данных работает корректно. На рисунке 18 представлен интерфейс перед генерацией данных, а на рисунке 19 — после генерации.

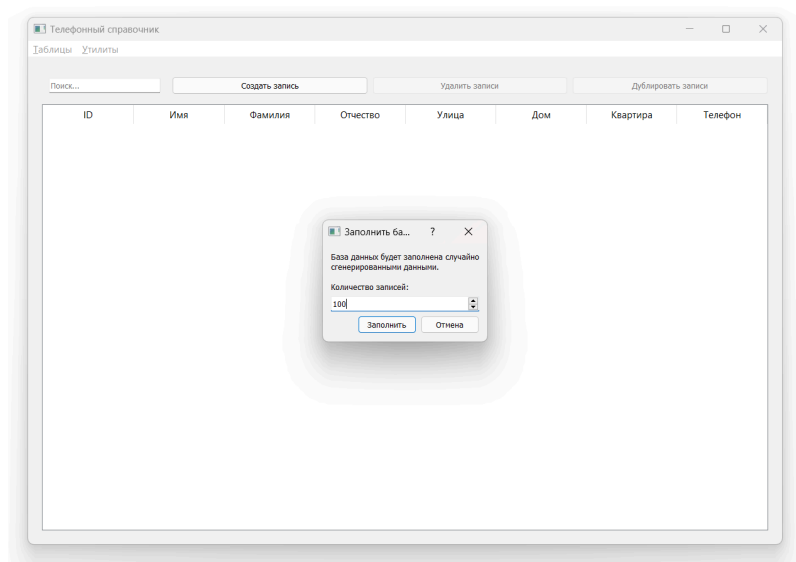


Рисунок 18 — Интерфейс перед генерацией данных

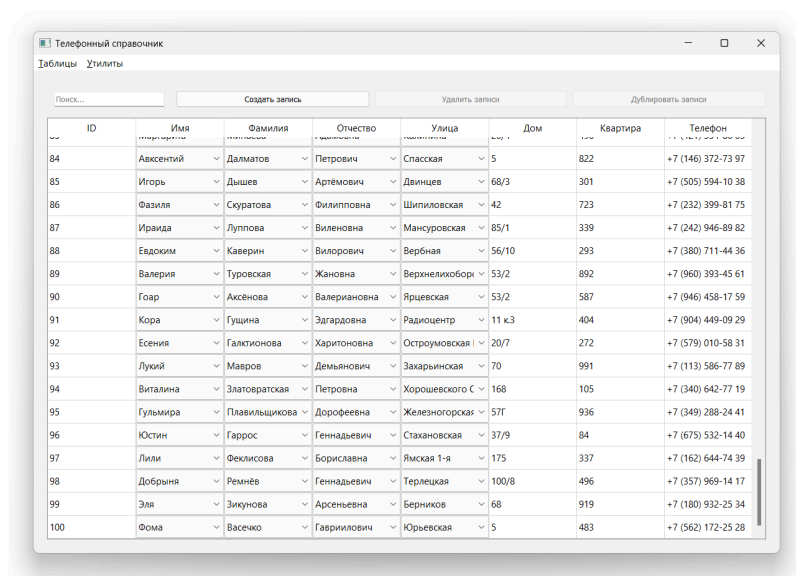


Рисунок 19 — Интерфейс после генерации данных

3.6 Сброс данных

Сброс данных работает корректно. На рисунке 20 представлен интерфейс перед сбросом данных, а на рисунке 21 — после сброса.

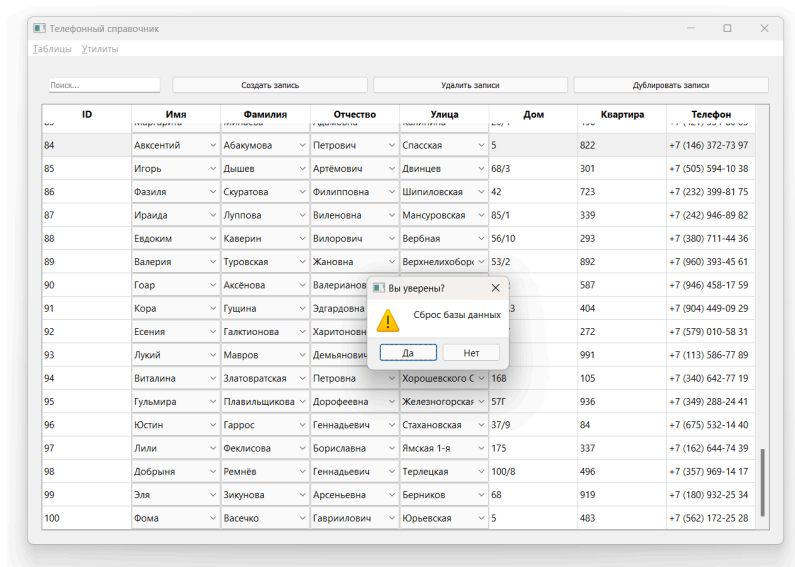


Рисунок 20 — Интерфейс перед сбросом данных

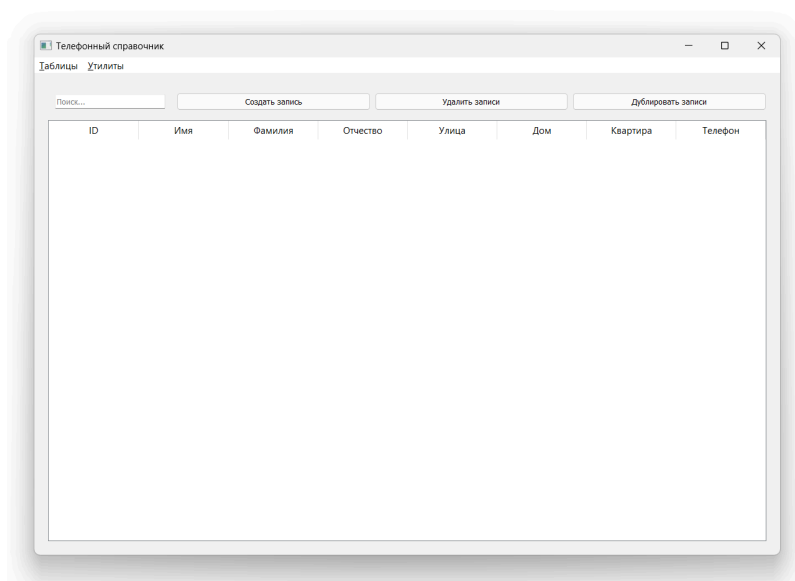


Рисунок 21 — Интерфейс после сброса данных

3.7 Нарушение ссылочной целостности

Ограничения ссылочной целостности предотвращают удаление записей из родительских таблиц, на которые есть ссылки. На рисунке 22 представлена попытка удаления записи из родительской таблицы, на которую ссылаются записи в главной таблице. На рисунке 23 показано сообщение об ошибке, указывающее на нарушение ссылочной целостности.

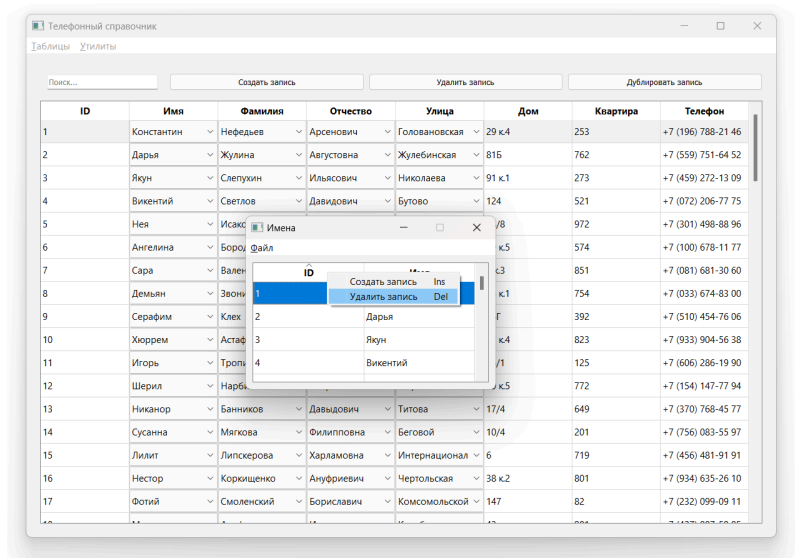


Рисунок 22 — Попытка удаления записи из родительской таблицы

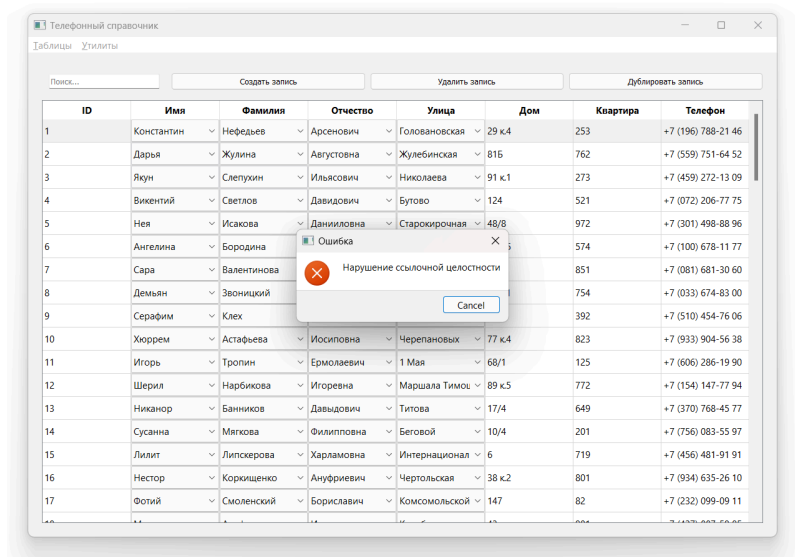


Рисунок 23 — Сообщение об ошибке нарушения ссылочной целостности

3.8 Редактирование родительских таблиц

Добавление, редактирование и удаление записей в родительских таблицах работают корректно. На рисунке 24 представлен интерфейс перед изменением родительской таблицы, а на рисунке 25 — после изменения.

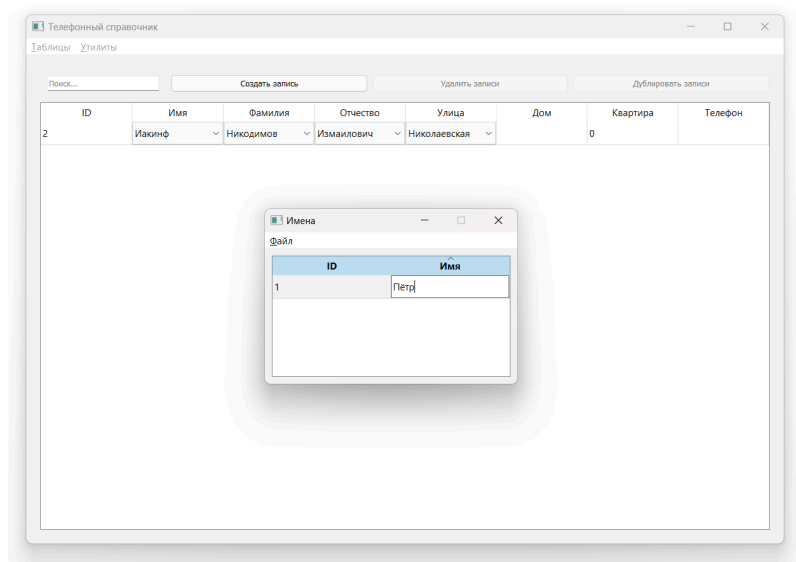


Рисунок 24 — Интерфейс перед изменением родительской таблицы

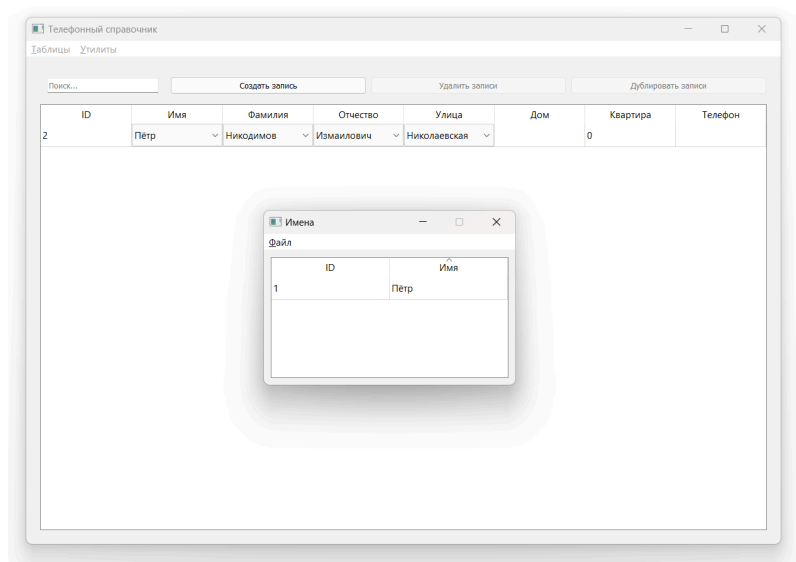


Рисунок 25 — Интерфейс после изменения родительской таблицы

ЗАКЛЮЧЕНИЕ

В рамках данной лабораторной работы была разработана и реализована реляционная база данных для телефонного справочника, используя систему управления базами данных PostgreSQL. Архитектура базы данных спроектирована с учетом требований целостности и эффективности хранения данных, обеспечивая оптимальную организацию информации.

Для взаимодействия с базой данных было разработано пользовательское приложение с графическим интерфейсом на базе библиотеки PyQt5. Приложение предоставляет функционал для выполнения основных операций: добавление, просмотр, редактирование и удаление записей. Использование выпадающих списков для выбора значений внешних ключей упрощает работу пользователя и предотвращает ошибки, связанные с вводом данных.

Дополнительные функции автоматизации, такие как заполнение и сброс данных, повышают удобство использования приложения. Внедрение логирования с помощью библиотеки loguru обеспечивает прозрачность работы приложения и позволяет отслеживать SQL-запросы, что упрощает отладку и анализ производительности.

Тщательное тестирование подтвердило корректность работы всех реализованных функций. Механизмы обеспечения ссылочной целостности гарантируют надежность хранения данных и защищают от непреднамеренного удаления связанной информации.

В результате, поставленные задачи были выполнены. Разработанная база данных и приложение для работы с ней полностью соответствуют заданным требованиям и демонстрируют эффективность выбранных подходов к проектированию и реализации.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Стандартиформ. ГОСТ 7.32–2017. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления. М: Стандартиформ, 2017. с. 32.
2. Лебедев П.А. Репозиторий проекта [электронный ресурс]. 2024. URL: <https://github.com/f0rgenet/phone-table>.