Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

Институт №3 «Системы управления, информатика и электроэнергетика»

Кафедра 307 «Цифровые технологии и информационные системы»

ОТЧЁТ

О выполнении лабораторной работы по предмету

«Базы данных»

«Создание системы, реализующей телефонный справочник»

Выполнил:

Студент группы М3О-212Б-22

Фетисов С.А.

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2023 г.

**Содержание**

[Цель работы 2](#_Toc153740350)

[Постановка задачи и выбор средств реализации 3](#_Toc153740351)

[Структура базы данных 4](#_Toc153740352)

[Руководство пользователя 5](#_Toc153740353)

[1. Взаимодействие с дочерней таблицей 6](#_Toc153740354)

[2. Взаимодействие с родительскими таблицами 9](#_Toc153740355)

[Тестирование 10](#_Toc153740356)

[Вывод 13](#_Toc153740357)

[Приложение А 14](#_Toc153740358)

# Цель работы

Создать приложение, реализующее связь с базой данных «телефонная книга», с использованием архитектуры клиент-сервер.

# Постановка задачи и выбор средств реализации

Необходимо реализовать серверную часть на PostgreSQL, для работы с базой данных использовалась программа pgAdmin. В качестве языка программирования, для создания архитектуры клиент-сервер, был выбран язык Python, а также фреймворк PyQT5. Для расположения элементов интерфейса была использована программа Qt designer. Библиотека psycopg2 обеспечивает связь с базой данных и реализацию запросов.

# Структура базы данных

Структура базы данных телефонного справочника представлена на рисунке 1.

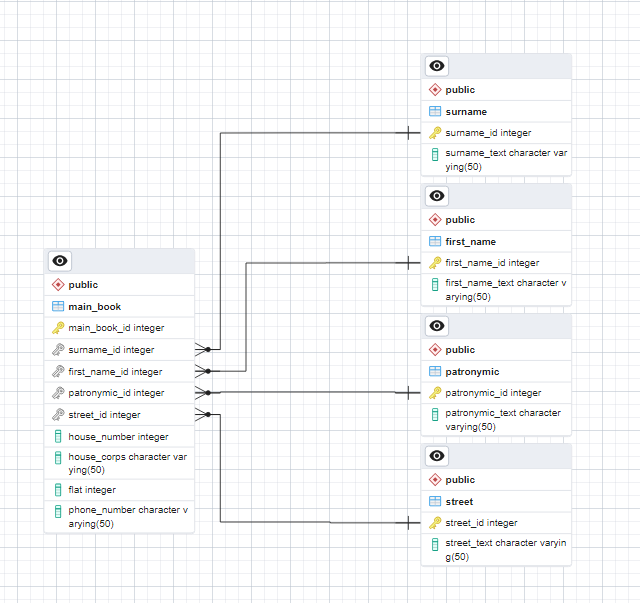


Рисунок — Структура базы данных

База данных состоит из таблиц: фамилия, имя, отчество, улица и телефонный справочник. Все таблицы связаны с телефонным справочником с помощью первичного ключа (столбец id). В самом телефонном справочнике хранится лишь телефон, номер дома, корпус и номер квартиры.

# Руководство пользователя

Созданная программа упрощает взаимодействие между пользователем и базой данных. Интерфейс программы представлен на рисунке 2.

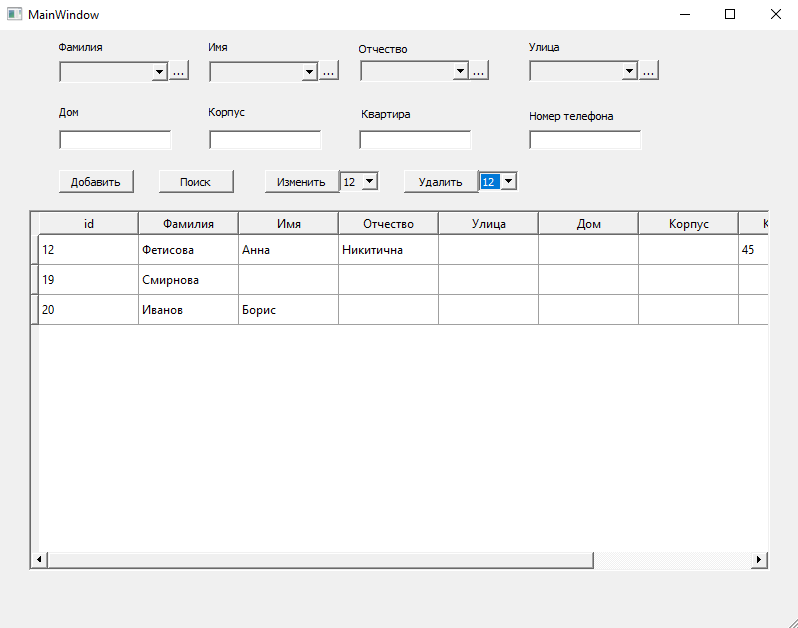


Рисунок — Интерфейс программы

# Взаимодействие с дочерней таблицей

Для добавления строки в дочернюю таблицу необходимо заполнить какие-то из полей (Фамилия, Имя, Отчество, Улица, Дом, Корпус, Квартира, Номер телефона) и нажать кнопку «Добавить»:

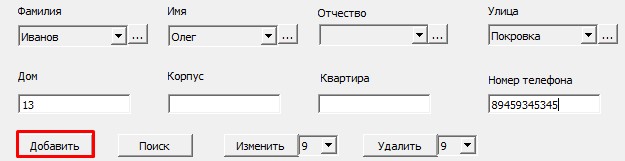


Рисунок — Добавление строки

Для того, чтобы осуществить поиск по одному или нескольким критериям в таблице нужно заполнить поля, по которым необходимо произвести поиск, затем нажать на кнопку «Поиск»:

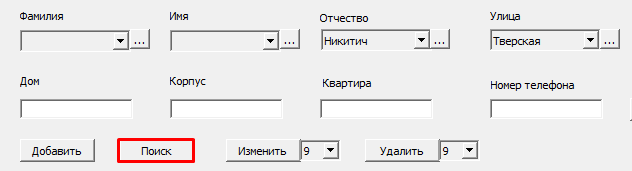


Рисунок — Поиск строки

После это будет выведено окно поиска, в котором будут находиться результаты поиска:

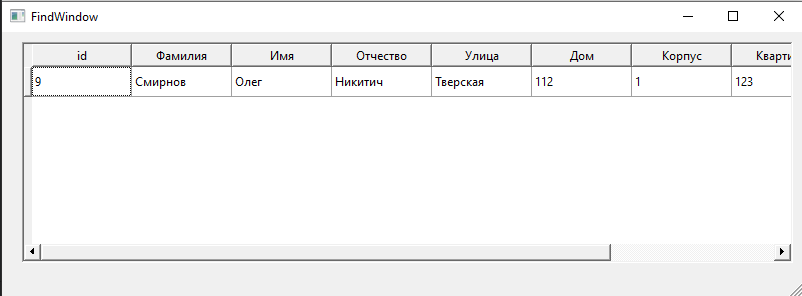


Рисунок — Результат поиска

Для изменения строки необходимо в выпадающем справа от кнопки «Изменить» списке выбрать id строки которую нужно изменить и ввести в поля (Фамилия, Имя, Отчество, Улица, Дом, Корпус, Квартира, Номер телефона) значения, которые будут в измененной строке, затем нажать кнопку «Изменить»:

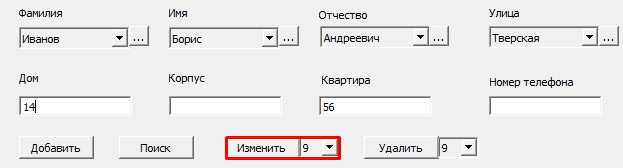


Рисунок — Изменение строки

Для удаления строки необходимо в выпадающем справа от кнопки «Удалить» списке выбрать id строки которую нужно удалить:



Рисунок — Удаление строки

Для того, чтобы открыть окно с одной из родительских таблиц необходимо нажать на кнопку «…», которая расположена рядом с названием нужной таблицы:

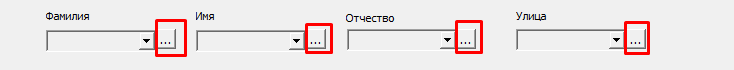


Рисунок — Открытие родительской таблицы

При нажатии на одну из этих кнопок выведется окно (рисунок 9) с соответствующей родительской таблицей:

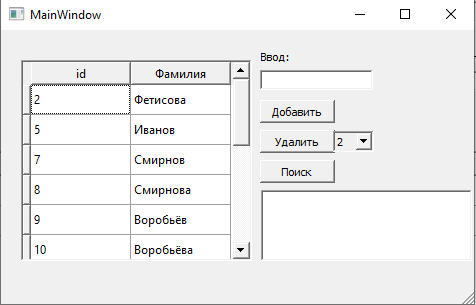


Рисунок — Окно родительской таблицы

# Взаимодействие с родительскими таблицами

Для того, чтобы добавить значение в родительскую таблицу необходимо ввести его в окно Ввода и нажать кнопку «Добавить»:

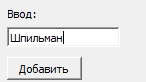


Рисунок — Добавление строки

Чтобы удалить строку родительской таблицы необходимо выбрать id строки, которую хочешь удалить, в выпадающем списке справа от кнопки «Удалить», и нажать на эту кнопку:



Рисунок — Удаление строки

Чтобы выполнить поиск по родительской таблице, необходимо ввести значение, которое нужно найти в родительской таблице и нажать кнопку «Поиск». В поле под кнопкой «Поиск» будет выведено id и значение строки, которую необходимо найти:

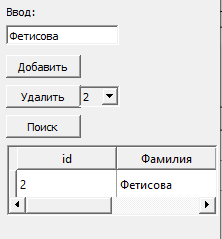


Рисунок — Поиск строки

# Тестирование

В ходе тестирования были проверены на работоспособность все элементы программы.

Добавление строки (Иванов Олег Андреевич) в дочернюю таблицу:

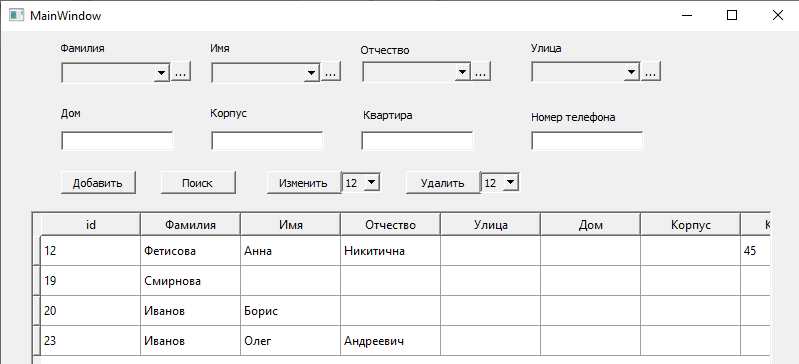


Рисунок — Добавление строки

Поиск строки (Иванов Олег Андреевич):

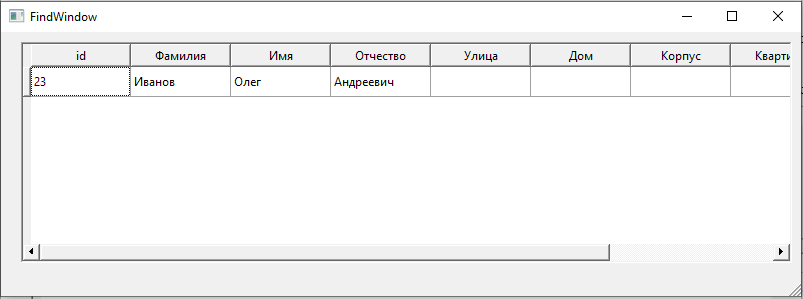


Рисунок — Поиск строки

Изменение строки (Иванов Олег Андреевич) на (Иванов Олег Никитич):

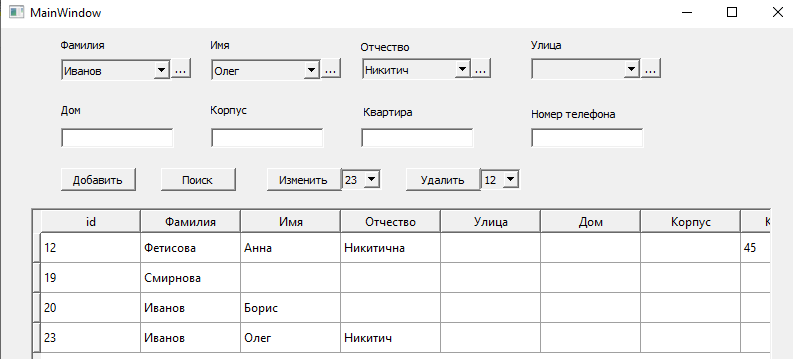


Рисунок — Изменение строки

Удаление строки (Иванов Олег Никитич):

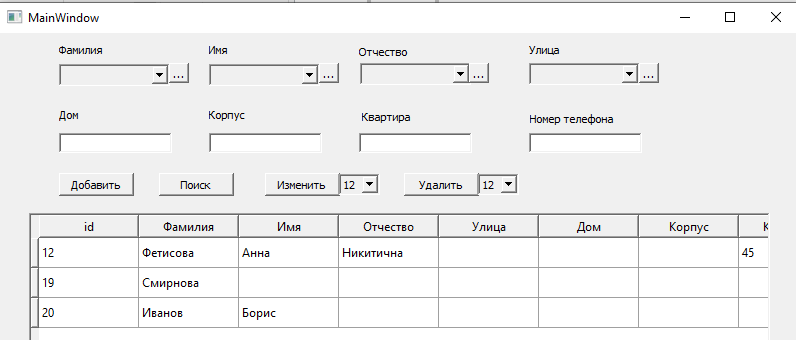


Рисунок — Удаление строки

Добавление строки (Александр) в таблицу «Имя»:

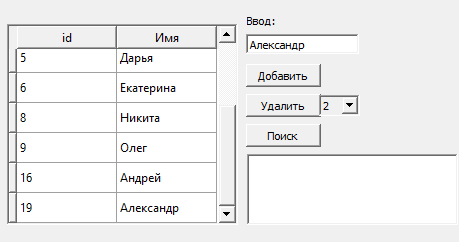


Рисунок — Добавление строки

Поиск строки (Александр) в таблице «Имя»:

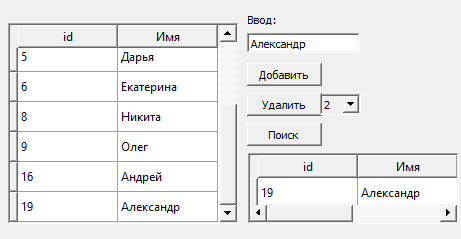


Рисунок — Поиск строки

Удаление строки (Александр) из таблицы «Имя»:

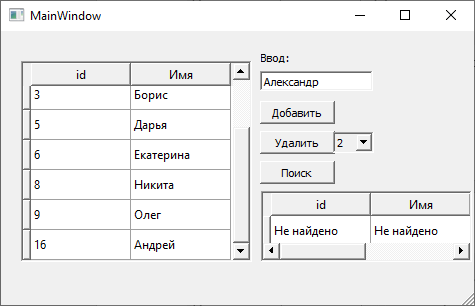


Рисунок — Удаление строки

# Вывод

В результате выполнения лабораторной работы был изучен язык SQL и его применение для создания приложения с использованием архитектуры клиент-сервер, библиотекa psycopg2 для работы с PostgreSQL с помощью языка Python и фреймворк Py QT5. Была создана база данных, реализующая телефонный справочник, а также программа клиент для связи с базой данных. Программа показала свою работоспособность в результате проведения тестирования.

# Приложение А

**Файл main.py**

from PyQt5.QtWidgets import \*

from child\_table import \*

from parent\_table import \*

from find\_table import \*

import sys

import psycopg2

class FindWindow(QtWidgets.QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.cursor = connection.cursor()

self.ui = Ui\_FindWindow()

self.ui.setupUi(self)

self.find\_data()

def find\_data(self):

surname\_sql = my\_interface.ui.comboBox.currentText()

first\_name\_sql = my\_interface.ui.comboBox\_2.currentText()

patronymic\_sql = my\_interface.ui.comboBox\_3.currentText()

street\_sql = my\_interface.ui.comboBox\_4.currentText()

house\_number\_sql = my\_interface.ui.lineEdit.text()

house\_corps\_sql = my\_interface.ui.lineEdit\_2.text()

flat\_sql = my\_interface.ui.lineEdit\_3.text()

phone\_number\_sql = my\_interface.ui.lineEdit\_4.text()

if surname\_sql != "":

self.cursor.execute('''SELECT surname\_text FROM surname WHERE surname\_text = '%s';''' % surname\_sql)

surname\_sql = str((self.cursor.fetchone())[0])

if first\_name\_sql != "":

self.cursor.execute(

'''SELECT first\_name\_text FROM first\_name WHERE first\_name\_text = '%s';''' % first\_name\_sql)

first\_name\_sql = str((self.cursor.fetchone())[0])

if patronymic\_sql != "":

self.cursor.execute(

'''SELECT patronymic\_text FROM patronymic WHERE patronymic\_text ='%s';''' % patronymic\_sql)

patronymic\_sql = str((self.cursor.fetchone())[0])

if street\_sql != "":

self.cursor.execute('''SELECT street\_text FROM street WHERE street\_text ='%s';''' % street\_sql)

street\_sql = str((self.cursor.fetchone())[0])

flag = False

if surname\_sql != '':

surname\_sql = "WHERE surname\_text = '" + surname\_sql + "'"

flag = True

if first\_name\_sql != '':

if flag:

first\_name\_sql = "AND first\_name\_text = '" +first\_name\_sql+"'"

else:

first\_name\_sql = "WHERE first\_name\_text = '" + first\_name\_sql + "'"

flag = True

if patronymic\_sql != '':

if flag:

patronymic\_sql = "AND patronymic\_text = '" + patronymic\_sql+"'"

else:

patronymic\_sql = "WHERE patronymic\_text = '" + patronymic\_sql + "'"

flag = True

if street\_sql != '':

if flag:

street\_sql = "AND street\_text= '" + street\_sql+"'"

else:

street\_sql = "WHERE street\_text= '" + street\_sql+"'"

flag = True

if house\_number\_sql != '':

if flag:

house\_number\_sql = "AND house\_number = " + house\_number\_sql

else:

house\_number\_sql = "WHERE house\_number = " + house\_number\_sql

flag = True

if house\_corps\_sql != '':

if flag:

house\_corps\_sql = "AND house\_corps = '" + house\_corps\_sql + "'"

else:

house\_corps\_sql = "WHERE house\_corps = '" + house\_corps\_sql + "'"

flag = True

if flat\_sql != '':

if flag:

flat\_sql = "AND flat = " + flat\_sql

else:

flat\_sql = "WHERE flat = " + flat\_sql

flag = True

if phone\_number\_sql != '':

if flag:

phone\_number\_sql = "AND phone\_number ='"+phone\_number\_sql + "'"

else:

phone\_number\_sql = "WHERE phone\_number ='"+phone\_number\_sql + "'"

flag = True

self.cursor.execute("""SELECT main\_book\_id,surname\_text,first\_name\_text,patronymic\_text,street\_text,house\_number,house\_corps,flat,phone\_number

FROM

main\_book

LEFT JOIN first\_name ON main\_book.first\_name\_id = first\_name.first\_name\_id

LEFT JOIN surname ON main\_book.surname\_id = surname.surname\_id

LEFT JOIN patronymic ON main\_book.patronymic\_id = patronymic.patronymic\_id

LEFT JOIN street ON main\_book.street\_id = street.street\_id

%s %s %s %s %s %s %s %s

ORDER BY main\_book\_id asc;""" % (surname\_sql,

first\_name\_sql,

patronymic\_sql,

street\_sql,

house\_number\_sql,

house\_corps\_sql,

flat\_sql,

phone\_number\_sql))

self.table = self.cursor.fetchall()

if self.table != []:

self.ui.tableWidget.setColumnCount(len(self.table[0]))

self.ui.tableWidget.setRowCount(len(self.table))

row = 0

for tup in self.table:

col = 0

for item in tup:

if item is None:

cellinfo = QTableWidgetItem(item)

else:

cellinfo = QTableWidgetItem(str(item))

self.ui.tableWidget.setItem(row, col, cellinfo)

cellinfo.setFlags(

QtCore.Qt.ItemIsSelectable | QtCore.Qt.ItemIsEnabled) # делаем ячейки нередактируемыми

col += 1

row += 1

self.ui.tableWidget.setHorizontalHeaderLabels(

('id', 'Фамилия', 'Имя', 'Отчество', 'Улица', 'Дом', 'Корпус', 'Квартира', 'Номер телефона'))

self.ui.tableWidget.setVerticalHeaderLabels('' for i in range(len(self.table)))

else:

self.ui.tableWidget.setColumnCount(1)

self.ui.tableWidget.setRowCount(1)

self.ui.tableWidget.setItem(0, 0, QTableWidgetItem("Не найдено"))

self.ui.tableWidget.setHorizontalHeaderLabels([''])

self.ui.tableWidget.setVerticalHeaderLabels([''])

class NewWindow(QtWidgets.QMainWindow):

def \_\_init\_\_(self, number\_window, col\_name, table\_name):

super().\_\_init\_\_()

self.cursor = connection.cursor()

self.number\_window = number\_window

self.col\_name = col\_name

self.table\_name = table\_name

self.ui = Ui\_AnotherWindow()

self.ui.setupUi(self)

self.ui.tableWidget.setColumnCount(len(my\_interface.record\_arr[self.number\_window][0]))

self.update\_data()

self.ui.pushButton\_.clicked.connect(self.find\_data)

self.ui.pushButton\_1.clicked.connect(self.delete\_row)

self.ui.pushButton\_2.clicked.connect(self.add)

def update\_data(self):

self.cursor.execute('''SELECT main\_book\_id,surname\_text,first\_name\_text,patronymic\_text,

street\_text,house\_number,house\_corps,flat,phone\_number

FROM

first\_name RIGHT JOIN

main\_book ON main\_book.first\_name\_id = first\_name.first\_name\_id

LEFT JOIN surname ON main\_book.surname\_id = surname.surname\_id

LEFT JOIN patronymic ON main\_book.patronymic\_id = patronymic.patronymic\_id

LEFT JOIN street ON main\_book.street\_id = street.street\_id

ORDER BY main\_book\_id asc; ''')

self.record = self.cursor.fetchall()

self.record\_arr = []

self.cursor.execute('''SELECT surname\_id,surname\_text FROM surname''')

surname\_arr = self.cursor.fetchall()

self.record\_arr.append(surname\_arr)

self.cursor.execute('''SELECT first\_name\_id,first\_name\_text FROM first\_name''')

first\_name\_arr = self.cursor.fetchall()

self.record\_arr.append(first\_name\_arr)

self.cursor.execute('''SELECT patronymic\_id,patronymic\_text FROM patronymic''')

patronymic\_arr = self.cursor.fetchall()

self.record\_arr.append(patronymic\_arr)

self.cursor.execute('''SELECT street\_id,street\_text FROM street''')

street\_arr = self.cursor.fetchall()

self.record\_arr.append(street\_arr)

self.ui.tableWidget.setRowCount(len(self.record\_arr[self.number\_window]))

self.ui.tableWidget.setHorizontalHeaderLabels(('id', self.col\_name))

self.ui.tableWidget.setVerticalHeaderLabels(('' for i in range(len(self.record\_arr[self.number\_window]))))

row = 0

for tup in self.record\_arr[self.number\_window]:

col = 0

for item in tup:

if item is None:

cellinfo = QTableWidgetItem(item)

else:

cellinfo = QTableWidgetItem(str(item))

self.ui.tableWidget.setItem(row, col, cellinfo)

cellinfo.setFlags(

QtCore.Qt.ItemIsSelectable | QtCore.Qt.ItemIsEnabled) # делаем ячейки нередактируемыми

col += 1

row += 1

self.ui.comboBox.clear()

if self.table\_name == "surname":

for i in range(len(surname\_arr)):

self.ui.comboBox.addItem(str(surname\_arr[i][0]))

elif self.table\_name == "first\_name":

for i in range(len(first\_name\_arr)):

self.ui.comboBox.addItem(str(first\_name\_arr[i][0]))

elif self.table\_name == "patronymic":

for i in range(len(patronymic\_arr)):

self.ui.comboBox.addItem(str(patronymic\_arr[i][0]))

elif self.table\_name == "street":

for i in range(len(street\_arr)):

self.ui.comboBox.addItem(str(street\_arr[i][0]))

def add(self):

print(self.record\_arr[self.number\_window])

col\_text\_sql = self.ui.lineEdit.text()

#my\_interface.cursor.execute()

self.cursor.execute('''INSERT INTO %s(%s\_text) VALUES ('%s');''' %(self.table\_name,self.table\_name, col\_text\_sql))

my\_interface.clear\_field()

self.update\_data()

def delete\_row(self):

try:

self.cursor.execute('''DELETE FROM %s

WHERE %s\_id = %s;''' % (self.table\_name,self.table\_name,self.ui.comboBox.currentText()))

except(Exception):

print(Exception)

finally:

my\_interface.clear\_field()

self.update\_data()

def find\_data(self):

col\_text\_sql = self.ui.lineEdit.text()

self.cursor.execute('''SELECT %s\_id,%s\_text FROM %s WHERE %s\_text = '%s';''' % (self.table\_name,

self.table\_name,

self.table\_name,

self.table\_name,

col\_text\_sql))

self.find\_table = self.cursor.fetchall()

if self.find\_table!=[]:

self.ui.tableWidget\_2.setColumnCount(2)

self.ui.tableWidget\_2.setRowCount(len(self.find\_table))

row = 0

for tup in self.find\_table:

col = 0

for item in tup:

if item is None:

cellinfo = QTableWidgetItem(item)

else:

cellinfo = QTableWidgetItem(str(item))

self.ui.tableWidget\_2.setItem(row, col, cellinfo)

cellinfo.setFlags(

QtCore.Qt.ItemIsSelectable | QtCore.Qt.ItemIsEnabled) # делаем ячейки нередактируемыми

col += 1

row += 1

else:

self.ui.tableWidget\_2.setColumnCount(2)

self.ui.tableWidget\_2.setRowCount(1)

self.ui.tableWidget\_2.setItem(0, 0, QTableWidgetItem("Не найдено"))

self.ui.tableWidget\_2.setItem(0, 1, QTableWidgetItem("Не найдено"))

self.ui.tableWidget\_2.setHorizontalHeaderLabels(

('id', self.col\_name))

self.ui.tableWidget\_2.setVerticalHeaderLabels('' for i in range(len(self.find\_table)))

class MainWindow(QtWidgets.QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.ui = Ui\_MainWindow()

self.ui.setupUi(self)

self.cursor = connection.cursor()

self.clear\_field()

self.update\_data()

self.ui.tableWidget.setHorizontalHeaderLabels(

('id', 'Фамилия', 'Имя', 'Отчество', 'Улица', 'Дом', 'Корпус', 'Квартира', 'Номер телефона')

)

self.ui.pushButton\_2.clicked.connect(self.add)

self.ui.pushButton\_3.clicked.connect(self.show\_find\_window)

self.ui.pushButton\_4.clicked.connect(self.update\_row)

self.ui.pushButton\_5.clicked.connect(self.delete\_row)

self.ui.pushButton\_6.clicked.connect(self.show\_surname\_window)

self.ui.pushButton\_7.clicked.connect(self.show\_first\_name\_window)

self.ui.pushButton\_8.clicked.connect(self.show\_patronymic\_window)

self.ui.pushButton\_9.clicked.connect(self.show\_street\_window)

def clear\_field(self):

self.ui.comboBox.clear()

self.ui.comboBox\_2.clear()

self.ui.comboBox\_3.clear()

self.ui.comboBox\_4.clear()

self.ui.comboBox\_5.clear()

self.ui.comboBox\_6.clear()

self.ui.lineEdit.clear()

self.ui.lineEdit\_2.clear()

self.ui.lineEdit\_3.clear()

self.ui.lineEdit\_4.clear()

self.ui.comboBox.addItem("")

self.ui.comboBox\_2.addItem("")

self.ui.comboBox\_3.addItem("")

self.ui.comboBox\_4.addItem("")

self.cursor.execute('''SELECT main\_book\_id,surname\_text,first\_name\_text,patronymic\_text,

street\_text,house\_number,house\_corps,flat,phone\_number

FROM

first\_name RIGHT JOIN

main\_book ON main\_book.first\_name\_id = first\_name.first\_name\_id

LEFT JOIN surname ON main\_book.surname\_id = surname.surname\_id

LEFT JOIN patronymic ON main\_book.patronymic\_id = patronymic.patronymic\_id

LEFT JOIN street ON main\_book.street\_id = street.street\_id

ORDER BY main\_book\_id asc; ''')

self.record = self.cursor.fetchall()

self.record\_arr = []

self.cursor.execute('''SELECT surname\_id,surname\_text FROM surname''')

surname\_arr = self.cursor.fetchall()

self.record\_arr.append(surname\_arr)

self.cursor.execute('''SELECT first\_name\_id,first\_name\_text FROM first\_name''')

first\_name\_arr = self.cursor.fetchall()

self.record\_arr.append(first\_name\_arr)

self.cursor.execute('''SELECT patronymic\_id,patronymic\_text FROM patronymic''')

patronymic\_arr = self.cursor.fetchall()

self.record\_arr.append(patronymic\_arr)

self.cursor.execute('''SELECT street\_id,street\_text FROM street''')

street\_arr = self.cursor.fetchall()

self.record\_arr.append(street\_arr)

for i in range(len(surname\_arr)):

self.ui.comboBox.addItem(str(surname\_arr[i][1]))

for i in range(len(first\_name\_arr)):

self.ui.comboBox\_2.addItem(str(first\_name\_arr[i][1]))

for i in range(len(patronymic\_arr)):

self.ui.comboBox\_3.addItem(str(patronymic\_arr[i][1]))

for i in range(len(street\_arr)):

self.ui.comboBox\_4.addItem(str(street\_arr[i][1]))

for tup in self.record:

self.ui.comboBox\_5.addItem(str(tup[0]))

self.ui.comboBox\_6.addItem(str(tup[0]))

def update\_data(self):

self.cursor.execute('''SELECT main\_book\_id,surname\_text,first\_name\_text,patronymic\_text,

street\_text,house\_number,house\_corps,flat,phone\_number

FROM

first\_name RIGHT JOIN

main\_book ON main\_book.first\_name\_id = first\_name.first\_name\_id

LEFT JOIN surname ON main\_book.surname\_id = surname.surname\_id

LEFT JOIN patronymic ON main\_book.patronymic\_id = patronymic.patronymic\_id

LEFT JOIN street ON main\_book.street\_id = street.street\_id

ORDER BY main\_book\_id asc; ''')

self.record = self.cursor.fetchall()

if self.record!=[]:

self.ui.tableWidget.setColumnCount(len(self.record[0]))

self.ui.tableWidget.setRowCount(len(self.record))

self.ui.tableWidget.setHorizontalHeaderLabels(

('id', 'Фамилия', 'Имя', 'Отчество', 'Улица', 'Дом', 'Корпус', 'Квартира', 'Номер телефона')

)

self.ui.tableWidget.setVerticalHeaderLabels('' for i in range(len(self.record)))

"""

self.record\_arr = []

self.cursor.execute('''SELECT surname\_id,surname\_text FROM surname''')

surname\_arr = self.cursor.fetchall()

self.record\_arr.append(surname\_arr)

self.cursor.execute('''SELECT first\_name\_id,first\_name\_text FROM first\_name''')

first\_name\_arr = self.cursor.fetchall()

self.record\_arr.append(first\_name\_arr)

self.cursor.execute('''SELECT patronymic\_id,patronymic\_text FROM patronymic''')

patronymic\_arr = self.cursor.fetchall()

self.record\_arr.append(patronymic\_arr)

self.cursor.execute('''SELECT street\_id,street\_text FROM street''')

street\_arr = self.cursor.fetchall()

self.record\_arr.append(street\_arr)

self.ui.comboBox.clear()

self.ui.comboBox\_2.clear()

self.ui.comboBox\_3.clear()

self.ui.comboBox\_4.clear()

self.ui.comboBox\_5.clear()

self.ui.comboBox\_6.clear()

self.ui.comboBox.addItem("")

self.ui.comboBox\_2.addItem("")

self.ui.comboBox\_3.addItem("")

self.ui.comboBox\_4.addItem("")

for i in range(len(surname\_arr)):

self.ui.comboBox.addItem(str(surname\_arr[i][1]))

for i in range(len(first\_name\_arr)):

self.ui.comboBox\_2.addItem(str(first\_name\_arr[i][1]))

for i in range(len(patronymic\_arr)):

self.ui.comboBox\_3.addItem(str(patronymic\_arr[i][1]))

for i in range(len(street\_arr)):

self.ui.comboBox\_4.addItem(str(street\_arr[i][1]))

for tup in self.record:

self.ui.comboBox\_5.addItem(str(tup[0]))

self.ui.comboBox\_6.addItem(str(tup[0]))

"""

row = 0

for tup in self.record:

col = 0

for item in tup:

if item is None:

cellinfo = QTableWidgetItem(item)

else:

cellinfo = QTableWidgetItem(str(item))

self.ui.tableWidget.setItem(row, col, cellinfo)

cellinfo.setFlags(

QtCore.Qt.ItemIsSelectable | QtCore.Qt.ItemIsEnabled) # делаем ячейки нередактируемыми

col += 1

row += 1

if self.record == []:

for col in range(9):

self.ui.tableWidget.setItem(0, col, QTableWidgetItem(""))

def show\_surname\_window(self):

self.surname\_w = NewWindow(0,"Фамилия","surname")

self.surname\_w.show()

def show\_first\_name\_window(self):

self.first\_name\_w = NewWindow(1,"Имя","first\_name")

self.first\_name\_w.show()

def show\_patronymic\_window(self):

self.patronymic\_w = NewWindow(2,"Отчество","patronymic")

self.patronymic\_w.show()

def show\_street\_window(self):

self.street\_w = NewWindow(3,"Улица","street")

self.street\_w.show()

def show\_find\_window(self):

self.find\_w = FindWindow()

self.find\_w.show()

def add(self):

surname\_sql = self.ui.comboBox.currentText()

first\_name\_sql = self.ui.comboBox\_2.currentText()

patronymic\_sql = self.ui.comboBox\_3.currentText()

street\_sql = self.ui.comboBox\_4.currentText()

house\_number\_sql = self.ui.lineEdit.text()

house\_corps\_sql = self.ui.lineEdit\_2.text()

flat\_sql = self.ui.lineEdit\_3.text()

phone\_number\_sql = self.ui.lineEdit\_4.text()

if surname\_sql != "":

self.cursor.execute('''SELECT surname\_id FROM surname WHERE surname\_text = '%s';''' % surname\_sql)

surname\_sql = str((self.cursor.fetchone())[0])

if first\_name\_sql != "":

self.cursor.execute('''SELECT first\_name\_id FROM first\_name WHERE first\_name\_text = '%s';''' % first\_name\_sql)

first\_name\_sql = str((self.cursor.fetchone())[0])

if patronymic\_sql != "":

self.cursor.execute('''SELECT patronymic\_id FROM patronymic WHERE patronymic\_text ='%s';''' % patronymic\_sql)

patronymic\_sql = str((self.cursor.fetchone())[0])

if street\_sql != "":

self.cursor.execute('''SELECT street\_id FROM street WHERE street\_text ='%s';''' % street\_sql)

street\_sql = str((self.cursor.fetchone())[0])

if surname\_sql =='':

surname\_sql = "NULL"

if first\_name\_sql != '':

first\_name\_sql =", "+first\_name\_sql

else:

first\_name\_sql = ",NULL"

if patronymic\_sql != '' :

patronymic\_sql =", "+patronymic\_sql

else:

patronymic\_sql = ", NULL"

if street\_sql != '':

street\_sql = ", "+street\_sql

else:

street\_sql = ", NULL"

if house\_number\_sql != '':

house\_number\_sql = ", "+house\_number\_sql

else:

house\_number\_sql = ", NULL"

if house\_corps\_sql != '':

house\_corps\_sql = ", "+"'"+house\_corps\_sql+"'"

else:

house\_corps\_sql = ", NULL"

if flat\_sql != '':

flat\_sql = ", "+flat\_sql

else:

flat\_sql = ", NULL"

if phone\_number\_sql!='':

phone\_number\_sql =", '"+phone\_number\_sql + "'"

else:

phone\_number\_sql = ", NULL"

self.cursor.execute("""INSERT INTO main\_book VALUES (DEFAULT,%s%s%s%s%s%s%s%s);""" % (surname\_sql,

first\_name\_sql,

patronymic\_sql,

street\_sql,

house\_number\_sql,

house\_corps\_sql,

flat\_sql,

phone\_number\_sql))

self.update\_data()

self.clear\_field()

def delete\_row(self):

self.cursor.execute('''DELETE FROM main\_book

WHERE main\_book\_id = %s;''' %self.ui.comboBox\_6.currentText())

self.update\_data()

self.clear\_field()

def update\_row(self):

surname\_sql = self.ui.comboBox.currentText()

first\_name\_sql = self.ui.comboBox\_2.currentText()

patronymic\_sql = self.ui.comboBox\_3.currentText()

street\_sql = self.ui.comboBox\_4.currentText()

house\_number\_sql = self.ui.lineEdit.text()

house\_corps\_sql = self.ui.lineEdit\_2.text()

flat\_sql = self.ui.lineEdit\_3.text()

phone\_number\_sql = self.ui.lineEdit\_4.text()

if surname\_sql != "":

self.cursor.execute('''SELECT surname\_id FROM surname WHERE surname\_text = '%s';''' % surname\_sql)

surname\_sql = str((self.cursor.fetchone())[0])

if first\_name\_sql != "":

self.cursor.execute(

'''SELECT first\_name\_id FROM first\_name WHERE first\_name\_text = '%s';''' % first\_name\_sql)

first\_name\_sql = str((self.cursor.fetchone())[0])

if patronymic\_sql != "":

self.cursor.execute(

'''SELECT patronymic\_id FROM patronymic WHERE patronymic\_text ='%s';''' % patronymic\_sql)

patronymic\_sql = str((self.cursor.fetchone())[0])

if street\_sql != "":

self.cursor.execute('''SELECT street\_id FROM street WHERE street\_text ='%s';''' % street\_sql)

street\_sql = str((self.cursor.fetchone())[0])

if surname\_sql == '':

surname\_sql = "NULL"

if first\_name\_sql == '':

first\_name\_sql = "NULL"

if patronymic\_sql == '':

patronymic\_sql = "NULL"

if street\_sql == '':

street\_sql = "NULL"

if house\_number\_sql == '':

house\_number\_sql = "NULL"

if house\_corps\_sql == '':

house\_corps\_sql = "NULL"

else:

house\_corps\_sql = "'"+house\_corps\_sql+"'"

if flat\_sql == '':

flat\_sql = "NULL"

if phone\_number\_sql == '':

phone\_number\_sql = "NULL"

else:

phone\_number\_sql = "'" + phone\_number\_sql + "'"

self.cursor.execute('''UPDATE main\_book

SET surname\_id = %s,

first\_name\_id = %s,

patronymic\_id = %s,

street\_id = %s,

house\_number = %s,

house\_corps = %s,

flat = %s,

phone\_number = %s

WHERE main\_book\_id = %s;

''' % (surname\_sql,

first\_name\_sql,

patronymic\_sql,

street\_sql,

house\_number\_sql,

house\_corps\_sql,

flat\_sql,

phone\_number\_sql,self.ui.comboBox\_5.currentText()))

self.update\_data()

# Подключение к существующей базе данных

connection = psycopg2.connect(user="postgres",

password="1111",

host="localhost",

port="5432",

database="phone\_book\_db")

connection.autocommit = True

app = QtWidgets.QApplication([])

my\_interface = MainWindow()

my\_interface.show()

sys.exit(app.exec())

**Файл parent\_table.py**

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui\_AnotherWindow(object):

def setupUi(self, MainWindow):

MainWindow.setObjectName("MainWindow")

MainWindow.resize(473, 274)

self.centralwidget = QtWidgets.QWidget(MainWindow)

self.centralwidget.setObjectName("centralwidget")

self.tableWidget = QtWidgets.QTableWidget(self.centralwidget)

self.tableWidget.setGeometry(QtCore.QRect(20, 30, 231, 201))

self.tableWidget.setObjectName("tableWidget")

self.tableWidget.setColumnCount(0)

self.tableWidget.setRowCount(0)

self.pushButton\_ = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_.setGeometry(QtCore.QRect(259, 130, 75, 23))

self.pushButton\_.setObjectName("pushButton\_")

self.pushButton\_2 = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_2.setGeometry(QtCore.QRect(259, 70, 75, 23))

self.pushButton\_2.setObjectName("pushButton\_2")

self.pushButton\_1 = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_1.setGeometry(QtCore.QRect(259, 100, 75, 23))

self.pushButton\_1.setObjectName("pushButton\_1")

self.lineEdit = QtWidgets.QLineEdit(self.centralwidget)

self.lineEdit.setGeometry(QtCore.QRect(259, 40, 113, 20))

self.lineEdit.setObjectName("lineEdit")

self.comboBox = QtWidgets.QComboBox(self.centralwidget)

self.comboBox.setGeometry(QtCore.QRect(332, 101, 41, 21))

self.comboBox.setObjectName("comboBox")

self.tableWidget\_2 = QtWidgets.QTableWidget(self.centralwidget)

self.tableWidget\_2.setGeometry(QtCore.QRect(260, 160, 211, 71))

self.tableWidget\_2.setObjectName("tableWidget\_2")

self.tableWidget\_2.setColumnCount(0)

self.tableWidget\_2.setRowCount(0)

self.label = QtWidgets.QLabel(self.centralwidget)

self.label.setGeometry(QtCore.QRect(260, 20, 47, 13))

self.label.setObjectName("label")

MainWindow.setCentralWidget(self.centralwidget)

self.statusbar = QtWidgets.QStatusBar(MainWindow)

self.statusbar.setObjectName("statusbar")

MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)

QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):

\_translate = QtCore.QCoreApplication.translate

MainWindow.setWindowTitle(\_translate("MainWindow", "MainWindow"))

self.pushButton\_.setText(\_translate("MainWindow", "Поиск"))

self.pushButton\_2.setText(\_translate("MainWindow", "Добавить"))

self.pushButton\_1.setText(\_translate("MainWindow", "Удалить"))

self.label.setText(\_translate("MainWindow", "Ввод:"))

**Файл child\_table.py**

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui\_MainWindow(object):

def setupUi(self, MainWindow):

MainWindow.setObjectName("MainWindow")

MainWindow.resize(800, 600)

self.centralwidget = QtWidgets.QWidget(MainWindow)

self.centralwidget.setObjectName("centralwidget")

'''

self.pushButton = QtWidgets.QPushButton(self.centralwidget)

self.pushButton.setGeometry(QtCore.QRect(670, 99, 121, 23))

self.pushButton.setObjectName("pushButton")

'''

self.tableWidget = QtWidgets.QTableWidget(self.centralwidget)

self.tableWidget.setGeometry(QtCore.QRect(30, 180, 741, 361))

self.tableWidget.setObjectName("tableWidget")

self.tableWidget.setColumnCount(0)

self.tableWidget.setRowCount(0)

self.comboBox = QtWidgets.QComboBox(self.centralwidget)

self.comboBox.setGeometry(QtCore.QRect(60, 31, 111, 22))

self.comboBox.setObjectName("comboBox")

self.comboBox\_2 = QtWidgets.QComboBox(self.centralwidget)

self.comboBox\_2.setGeometry(QtCore.QRect(210, 31, 111, 22))

self.comboBox\_2.setObjectName("comboBox\_2")

self.comboBox\_3 = QtWidgets.QComboBox(self.centralwidget)

self.comboBox\_3.setGeometry(QtCore.QRect(361, 30, 111, 22))

self.comboBox\_3.setObjectName("comboBox\_3")

self.comboBox\_4 = QtWidgets.QComboBox(self.centralwidget)

self.comboBox\_4.setGeometry(QtCore.QRect(530, 30, 111, 22))

self.comboBox\_4.setObjectName("comboBox\_4")

self.lineEdit = QtWidgets.QLineEdit(self.centralwidget)

self.lineEdit.setGeometry(QtCore.QRect(60, 100, 113, 20))

self.lineEdit.setObjectName("lineEdit")

self.lineEdit\_2 = QtWidgets.QLineEdit(self.centralwidget)

self.lineEdit\_2.setGeometry(QtCore.QRect(210, 100, 113, 20))

self.lineEdit\_2.setObjectName("lineEdit\_2")

self.lineEdit\_3 = QtWidgets.QLineEdit(self.centralwidget)

self.lineEdit\_3.setGeometry(QtCore.QRect(360, 100, 113, 20))

self.lineEdit\_3.setObjectName("lineEdit\_3")

self.lineEdit\_4 = QtWidgets.QLineEdit(self.centralwidget)

self.lineEdit\_4.setGeometry(QtCore.QRect(530, 100, 113, 20))

self.lineEdit\_4.setText("")

self.lineEdit\_4.setObjectName("lineEdit\_4")

self.pushButton\_2 = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_2.setGeometry(QtCore.QRect(60, 140, 75, 23))

self.pushButton\_2.setObjectName("pushButton\_2")

self.pushButton\_4 = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_4.setGeometry(QtCore.QRect(266, 140, 75, 23))

self.pushButton\_4.setObjectName("pushButton\_4")

self.pushButton\_5 = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_5.setGeometry(QtCore.QRect(405, 140, 75, 23))

self.pushButton\_5.setObjectName("pushButton\_5")

self.pushButton\_6 = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_6.setEnabled(True)

self.pushButton\_6.setGeometry(QtCore.QRect(170, 30, 20, 20))

self.pushButton\_6.setObjectName("pushButton\_6")

self.pushButton\_7 = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_7.setEnabled(True)

self.pushButton\_7.setGeometry(QtCore.QRect(320, 30, 20, 20))

self.pushButton\_7.setObjectName("pushButton\_7")

self.pushButton\_8 = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_8.setEnabled(True)

self.pushButton\_8.setGeometry(QtCore.QRect(470, 30, 20, 20))

self.pushButton\_8.setObjectName("pushButton\_8")

self.pushButton\_9 = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_9.setEnabled(True)

self.pushButton\_9.setGeometry(QtCore.QRect(640, 30, 20, 20))

self.pushButton\_9.setObjectName("pushButton\_9")

self.comboBox\_5 = QtWidgets.QComboBox(self.centralwidget)

self.comboBox\_5.setGeometry(QtCore.QRect(340, 141, 41, 21))

self.comboBox\_5.setObjectName("comboBox\_5")

self.comboBox\_6 = QtWidgets.QComboBox(self.centralwidget)

self.comboBox\_6.setGeometry(QtCore.QRect(479, 141, 41, 21))

self.comboBox\_6.setObjectName("comboBox\_6")

self.label = QtWidgets.QLabel(self.centralwidget)

self.label.setGeometry(QtCore.QRect(60, 10, 47, 13))

self.label.setObjectName("label")

self.label\_2 = QtWidgets.QLabel(self.centralwidget)

self.label\_2.setGeometry(QtCore.QRect(210, 10, 47, 13))

self.label\_2.setObjectName("label\_2")

self.label\_3 = QtWidgets.QLabel(self.centralwidget)

self.label\_3.setGeometry(QtCore.QRect(360, 10, 61, 16))

self.label\_3.setObjectName("label\_3")

self.label\_4 = QtWidgets.QLabel(self.centralwidget)

self.label\_4.setGeometry(QtCore.QRect(530, 10, 47, 13))

self.label\_4.setObjectName("label\_4")

self.label\_5 = QtWidgets.QLabel(self.centralwidget)

self.label\_5.setGeometry(QtCore.QRect(60, 75, 47, 13))

self.label\_5.setObjectName("label\_5")

self.label\_6 = QtWidgets.QLabel(self.centralwidget)

self.label\_6.setGeometry(QtCore.QRect(210, 75, 47, 13))

self.label\_6.setObjectName("label\_6")

self.label\_7 = QtWidgets.QLabel(self.centralwidget)

self.label\_7.setGeometry(QtCore.QRect(363, 75, 51, 16))

self.label\_7.setObjectName("label\_7")

self.label\_8 = QtWidgets.QLabel(self.centralwidget)

self.label\_8.setGeometry(QtCore.QRect(531, 75, 91, 20))

self.label\_8.setObjectName("label\_8")

self.pushButton\_3 = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_3.setGeometry(QtCore.QRect(160, 140, 75, 23))

self.pushButton\_3.setObjectName("pushButton\_3")

MainWindow.setCentralWidget(self.centralwidget)

self.statusbar = QtWidgets.QStatusBar(MainWindow)

self.statusbar.setObjectName("statusbar")

MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)

# type: ignore

QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):

\_translate = QtCore.QCoreApplication.translate

MainWindow.setWindowTitle(\_translate("MainWindow", "MainWindow"))

#self.pushButton.setText(\_translate("MainWindow", "Очистить поля ввода"))

self.pushButton\_2.setText(\_translate("MainWindow", "Добавить"))

self.pushButton\_4.setText(\_translate("MainWindow", "Изменить"))

self.pushButton\_5.setText(\_translate("MainWindow", "Удалить"))

self.pushButton\_6.setText(\_translate("MainWindow", "..."))

self.pushButton\_7.setText(\_translate("MainWindow", "..."))

self.pushButton\_8.setText(\_translate("MainWindow", "..."))

self.pushButton\_9.setText(\_translate("MainWindow", "..."))

self.label.setText(\_translate("MainWindow", "Фамилия"))

self.label\_2.setText(\_translate("MainWindow", "Имя"))

self.label\_3.setText(\_translate("MainWindow", "Отчество"))

self.label\_4.setText(\_translate("MainWindow", "Улица"))

self.label\_5.setText(\_translate("MainWindow", "Дом"))

self.label\_6.setText(\_translate("MainWindow", "Корпус"))

self.label\_7.setText(\_translate("MainWindow", "Квартира"))

self.label\_8.setText(\_translate("MainWindow", "Номер телефона"))

self.pushButton\_3.setText(\_translate("MainWindow", "Поиск"))

**Файл find\_table.py**

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui\_FindWindow(object):

def setupUi(self, MainWindow):

MainWindow.setObjectName("Окно поиска")

MainWindow.resize(800, 264)

self.centralwidget = QtWidgets.QWidget(MainWindow)

self.centralwidget.setObjectName("centralwidget")

self.tableWidget = QtWidgets.QTableWidget(self.centralwidget)

self.tableWidget.setGeometry(QtCore.QRect(20, 10, 771, 221))

self.tableWidget.setObjectName("tableWidget")

self.tableWidget.setColumnCount(0)

self.tableWidget.setRowCount(0)

MainWindow.setCentralWidget(self.centralwidget)

self.statusbar = QtWidgets.QStatusBar(MainWindow)

self.statusbar.setObjectName("statusbar")

MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)

QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):

\_translate = QtCore.QCoreApplication.translate

MainWindow.setWindowTitle(\_translate("MainWindow", "FindWindow"))