Государственное бюджетное профессиональное учреждение



Московской области Люберецкий техникум имени героя Советского Союза, летчика космонавта Юрия Алексеевича Гагарина

**Отчет: Создание базы данных**

**и интерфейса “blagodat”**

Авторы:

Студенты специальности 09.02.07

“Информационные системы и программирование”,

2-ого курса, группы ИС-21

Попов Андрей Михайлович

Усмонов Нурсултан Нуруллоевий

Рецензент:

Тарджиманян Лия Николаевна

Оценка:\_\_\_\_\_

Дзержинский 2023 г

**Содержание:**

1. Введение.……………………………………...…………………………….…
2. Техническое задание……………………………………………………….....
3. Логическая структура………………………………………………………...
4. Физическая структура………………………………………………….……..
5. Работа с данными…..………………………………………………………
6. Инструкция пользования…….………………………………………….…
7. Разработка интерфейса…………………………………………………...…
8. Заключение…………………………………………………………………..

**Введение**

Цели

Цель отчета проектирования: закрепление теоретических знаний, а также навыков проектирования БД, полученных при изучении дисциплины «Базы данных».

Задачи:

* Разработать er диаграмму по предметной области
* По разработанной er диаграмме создать базу данных
* Заполнить данными получившуюся базу данных
* Разработать интерфейс для работы с базой данных
* Связать интерфейс с базой данных с помощью программирования

Инструменты:

* Diagrams.net(сайт для создания er диаграммы)
* Sqlite(для разработки базы данных)
* Qt5 дизайнер(для создания визуальной составляющей интерфейса)
* visual studio code/PyCharm(ide для программирования)

Выбор СУБД.

* Sqlite

Вашей задачей является разработка информационной системы для точки проката горнолыжного комплекса «Благодать» в Алтайском крае.

Горнолыжный комплекс «Благодать» расположен на курорте Белокуриха – популярном месте для отдыха и лечения. Природа щедро одарила этот удивительный уголок России. Климат курорта отличается сравнительно высокой для Сибири среднегодовой температурой. Здесь нет резкого колебания давления, большое количество ясных и солнечных дней в году. В Белокурихе комфортно отдыхать в любое время года. Курорт известен своими удивительными лечебными радоновыми водами. На территории Белокурихи расположены крупные лечебницы, оказывающие комплексное лечение при различных заболеваниях.

Горнолыжный комплекс «Благодать» внесен во Всероссийский реестр объектов спорта.

Комплекс является одним из популярных мест для активного зимнего отдыха.

Комплекс обладает развитой инфраструктурой:

• сеть взаимосвязанных трасс;

• канатно-кресельная дорога «Церковка»;

• пункты проката;

• школа горнолыжного спорта;

• отель (в нескольких шагах от подъемников);

• ресторан;

• кафе.

За отличным состоянием трасс следит штат квалифицированных специалистов, которые используют целый парк различной специализированной техники. Экологически чистая и безопасная система искусственного снегообразования позволяет позволяет значительно продлить горнолыжный сезон в Белокурихе, поэтому катание на трассах горнолыжного комплекса «Благодать» комфортно и безопасно.

На горнолыжном комплексе «Благодать» действует школа горнолыжного спорта, инструкторами которой являются специалисты высокого уровня.

В пункте проката для клиентов подберут современное горнолыжное оборудование ведущих мировых производителей.

Трассы Белокурихи – идеальное место как для тех, кто только планирует познакомиться с горными лыжами или сноубордом, так и для опытных спортсменов, стремящихся усовершенствовать свои навыки.

Требования к ИС точки проката горнолыжного комплекса «Благодать»:

• работа с единой базой данных для всех пользователей;

• ведение базы заказов;

• составление отчетов;

• асинхронное выполнение всех операций (приложение не должно «висеть»).

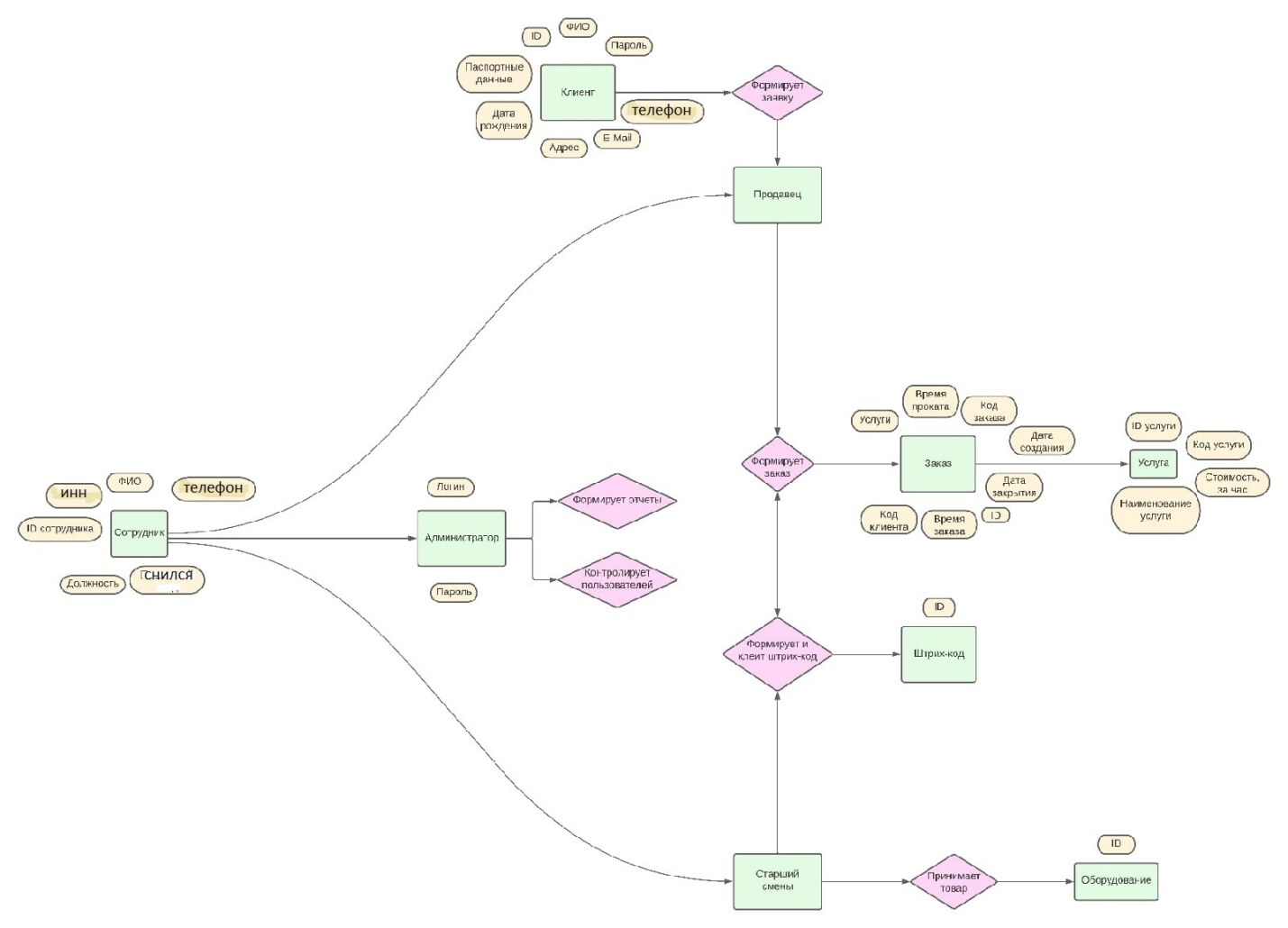
Используемые технологии при разработке:

• разработка клиент-серверных приложений;

• создание штрих-кодов.

**Логическая структура**

Создание ER-diagramma для дальнейшей работы с БД.



**Физическая структура**

Проектирование физической структуры базы данных.

На базе er-диаграммы создаём базу данных в Sqlite. У нас есть три сущности: услуга, клиенты, сводная.

Таблица “admin” имеет следующие атрибуты:

ID

Login

Pass

ID\_worker

Таблица “barcode” имеет следующие атрибуты:

ID

Senior\_officer\_id

barcode

Таблица “client” имеет следующие атрибуты:

Id

Fio

Код клиента

Серия

Номер

День Рождения

Post index

City

Street

House

Flat

Login

Pass

Таблица “Instrument” имеет следующие атрибуты:

ID

Name

Таблица “marketer” имеет следующие атрибуты:

Client\_id

Workers\_id

Таблица “senior\_officer” имеет следующие атрибуты:

Workers\_id

Instrument\_id

Таблица “services” имеет следующие атрибуты:

ID

Name

Code

Count

Таблица “workers” имеет следующие атрибуты:

Id

Post

Name

INN

SNILS

Phone

Таблица “zakazi” имеет следующие атрибуты:

Id

Код заказа

Дата создания

Время заказа

Код клиента

Дата закрытия

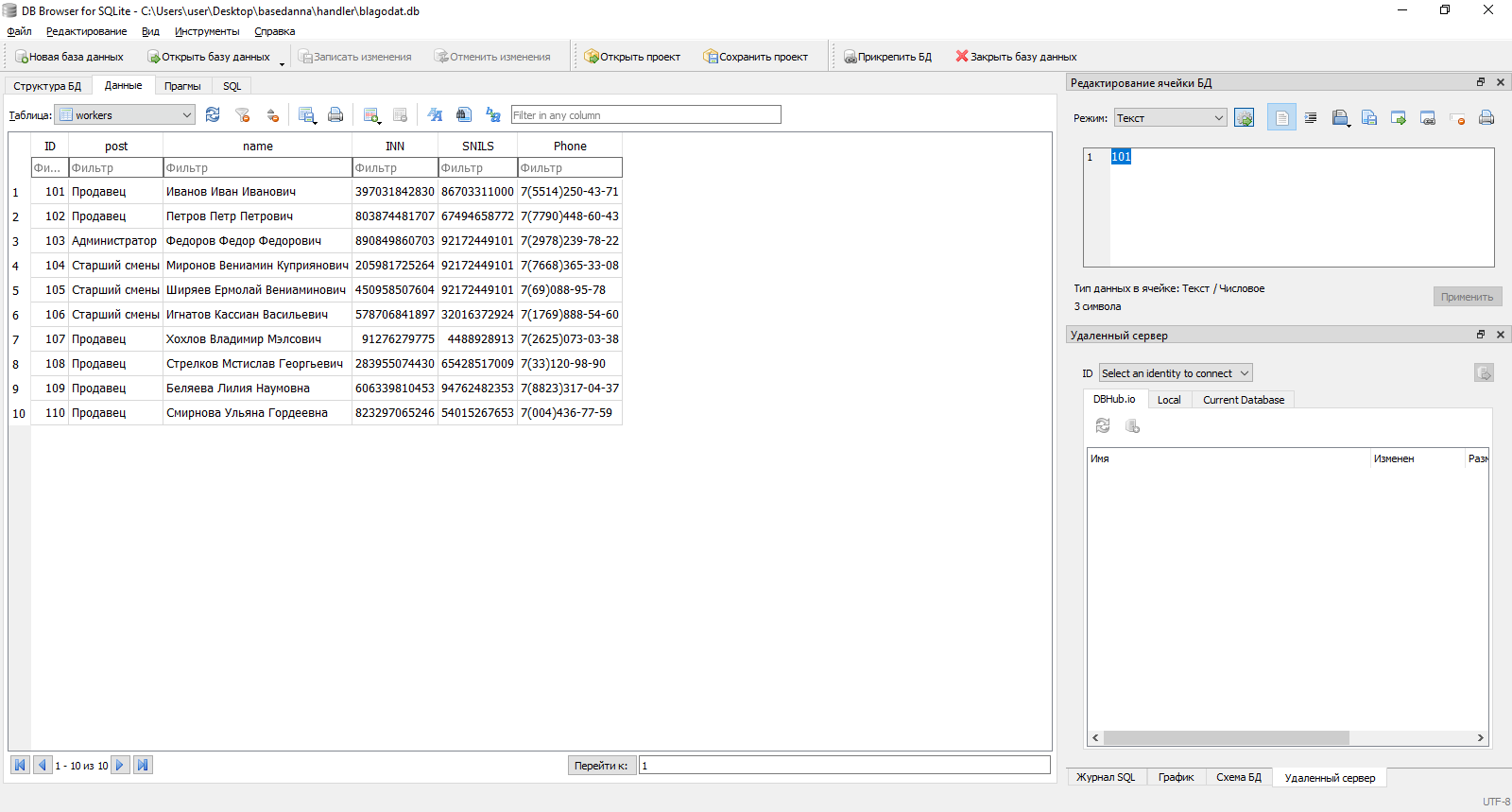
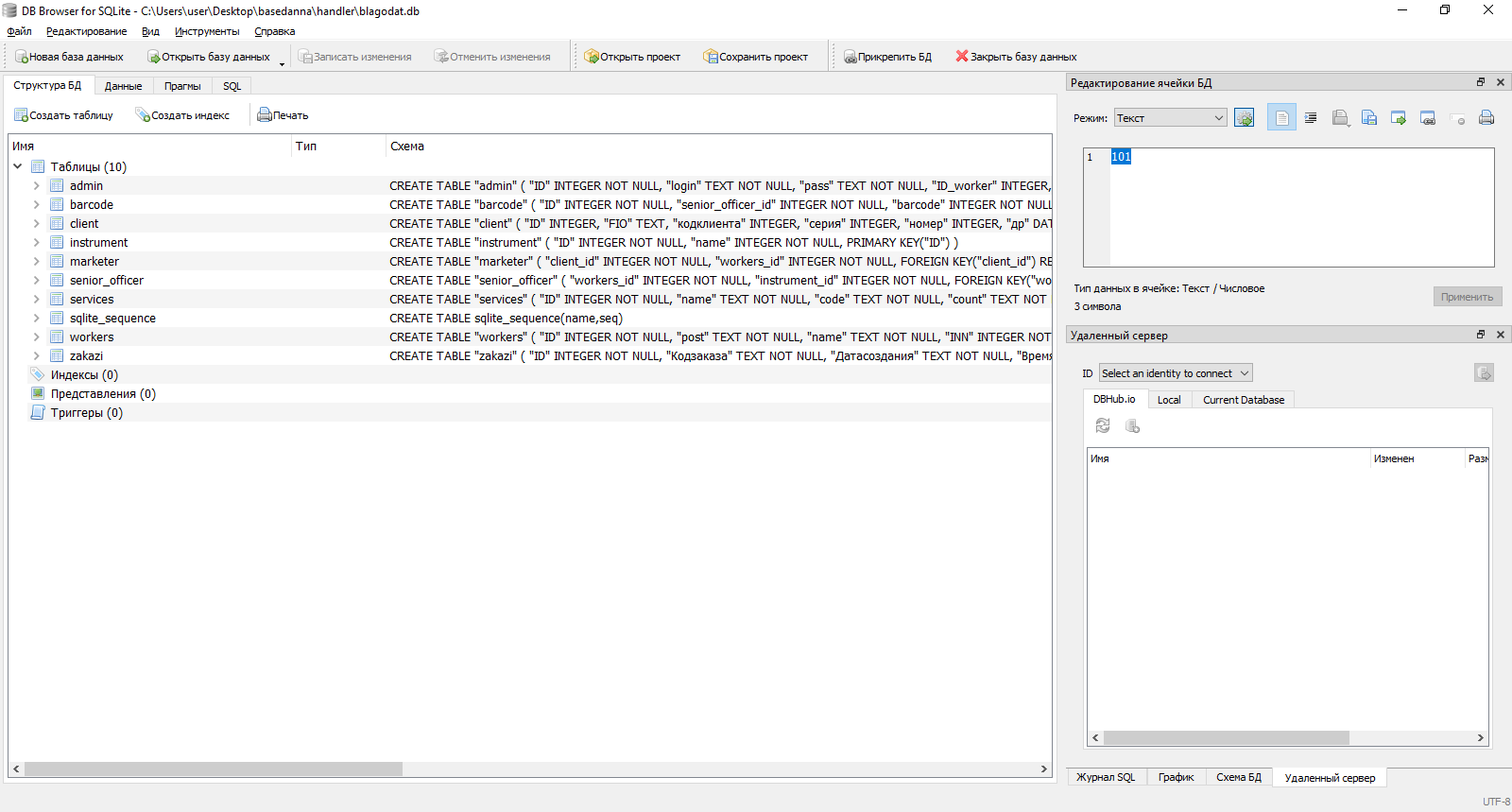
Время в часах

Код услуги

Старший смены

**Работа с данными**

Ввод данных в БД

Первичный ввод записей в базу данных будет производится при помощи импортирования данных из Excel. Далее ввод данных будет производится непосредственно с формы.

Организация корректировки БД.

Корректироваться данных производится с формы, или непосредственно в Sqlite.

**Инструкция пользования**

Работа с формами бд (Администратор)

Описание информационных потребностей пользователей и выбор способов их реализации.

* Регистрация на форме
* Авторизация на форме
* Добавление данных в базу данных
* Изменение данных в таблице

Авторизация:

Если Вы уже зарегистрированы для работы с формами(интерфейсом) как администратор, то вам нужно вписать свои данные в нужные ячейки без ошибок.

Дальше вас переносит на форму с выбором баз данных, на ней вы выбираете, нажимая на кнопку с названием базы данных которая вам нужна.

Но если вас нету в базе данных как администратора или у вас нету данных, то вы можете зарегистрироваться в ней, ведя данные и добавив их в базу данных нажав кнопку Sign UP, далее проделываете выше сказанное.

Добавление новых записей

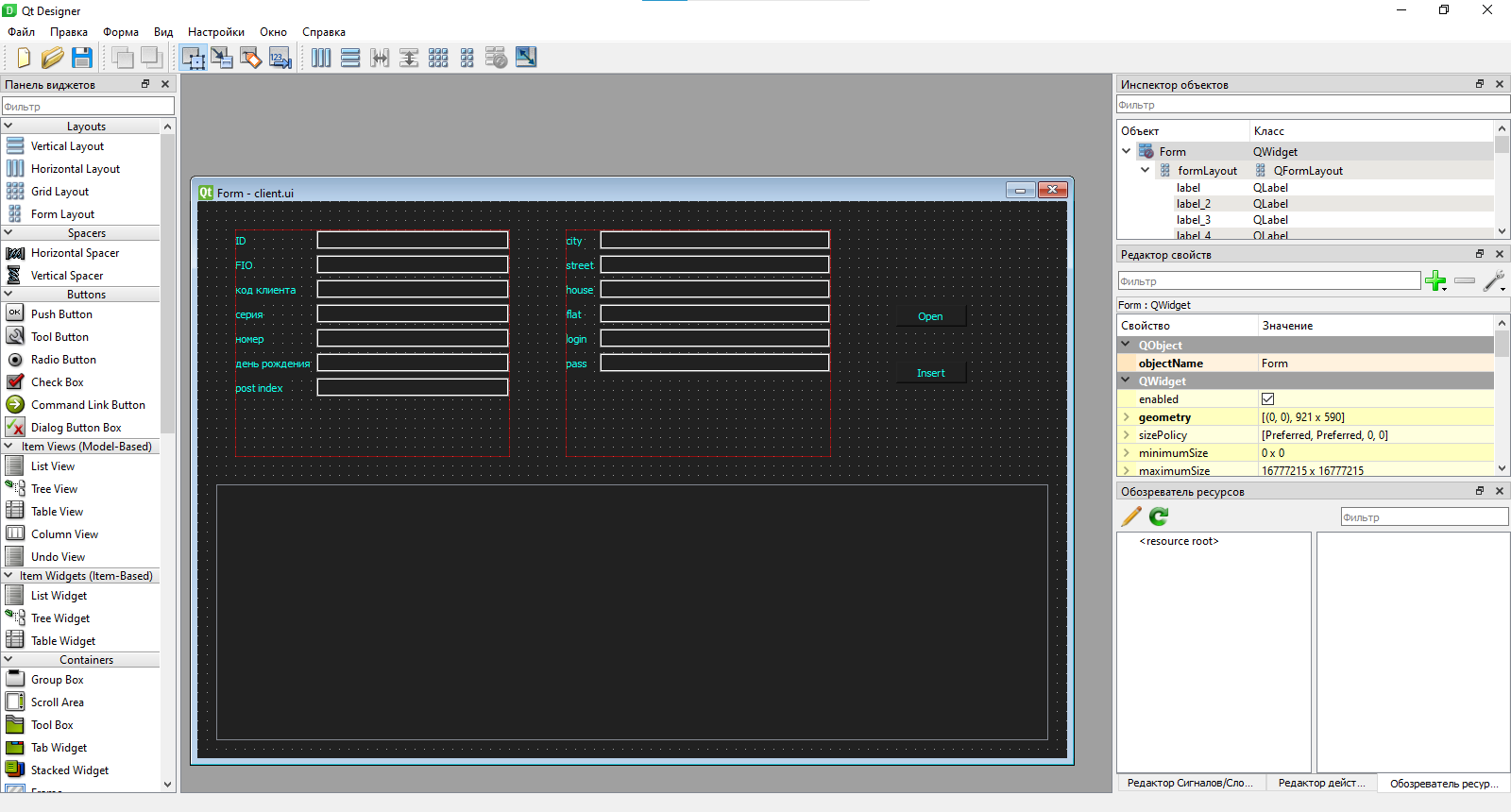
Администратор может изменять даны в таблице, после того как Вы выбрали определенную базу данных, то у вас откроется форма этой базы данных. Вы должны нажать кнопку Open и у вас откроется база данных, в который Вы увидите её структуру, чтобы добавить данные, вам надо внести определенные значения в поля, и нажать кнопку Insert, после чего база данных обновится и добавить вашу запись.

Переключение между формами

Чтобы переключаться между формами, ванн надо вернуться на форму с выбором форм и там выбирать нужную вам форму.

**Разработка интерфейса.**

Для начала в QT 5 дизайнере необходимо создать внешний вид форм, перенести на окно необходимые кнопки, поля для ввода данных, текстовые значения. После создания внешнего вида формы нам необходимо добавить функции на кнопки и поля ввода, чтобы наш интерес исправно функционировал.

 Назначение команд для функционирования форм

import sys

import sqlite3

from PyQt5.QtWidgets import QApplication, QWidget, QTableWidgetItem

from service import Ui\_Form

#STAFF\_POSTS = ['бухгалтер', 'менеджер', 'программист']

class MyWidget(QWidget, Ui\_Form):

    def \_\_init\_\_(self):

        super(MyWidget, self).\_\_init\_\_()

        self.setupUi(self)

        #self.cbPost.addItems(STAFF\_POSTS)

        self.pushButton.clicked.connect(self.open)

        self.pushButton\_2.clicked.connect(self.insert)

    def open(self):

        #try:

        self.conn = sqlite3.connect('handler\\blagodat.db')

        cur = self.conn.cursor()

        data = cur.execute("select \* from 'services'")

        col\_name = [i[0] for i in data.description]

        data\_rows = data.fetchall()

        #except Exception as e:

            # print("Ошибка с подключением к базе данных")

            # return e

        self.twStaffs.setColumnCount(len(col\_name))

        self.twStaffs.setHorizontalHeaderLabels(col\_name)

        self.twStaffs.setRowCount(0)

        for i, row in enumerate(data\_rows):

            self.twStaffs.setRowCount(self.twStaffs.rowCount()+1)

            for j, elem in enumerate(row):

                self.twStaffs.setItem(i, j, QTableWidgetItem(str(elem)))

        self.twStaffs.resizeColumnsToContents()

    def update\_twStaffs(self, query="select \* from services"):

        try:

            cur = self.conn.cursor()

            data = cur.execute(query).fetchall()

        except Exception as e:

            print(f"Проблемы с подключением к БД. {e}")

            return e

        self.twStaffs.setRowCount(0)

        for i, row in enumerate(data):

            self.twStaffs.setRowCount(self.twStaffs.rowCount() +1)

            for j, elem in enumerate(row):

                self.twStaffs.setItem(i, j, QTableWidgetItem(str(elem)))

            self.twStaffs.resizeColumnsToContents()

    def insert(self):

        row = [self.leID.text(), self.lename.text(), self.lecode.text(), self.lecount.text()]

        try:

            cur = self.conn.cursor()

            cur.execute(f"""insert into workers (ID, name, code, count)

            values('{row[0]}', '{row[1]}', '{row[2]}','{row[3]}')""")

            self.conn.commit()

            cur.close()

        except Exception as e:

            print("Не смогли добавить запись.")

            return e

        self.update\_twStaffs()

if \_\_name\_\_ == '\_\_main\_\_':

    app = QApplication(sys.argv)

    ex = MyWidget()

    ex.show()

    sys.exit(app.exec\_())

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui\_Form(object):

    def setupUi(self, Form):

        Form.setObjectName("Form")

        Form.resize(609, 415)

        sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Preferred, QtWidgets.QSizePolicy.Preferred)

        sizePolicy.setHorizontalStretch(0)

        sizePolicy.setVerticalStretch(0)

        sizePolicy.setHeightForWidth(Form.sizePolicy().hasHeightForWidth())

        Form.setSizePolicy(sizePolicy)

        Form.setStyleSheet("background-color: rgb(33, 33, 33)")

        self.twStaffs = QtWidgets.QTableWidget(Form)

        self.twStaffs.setGeometry(QtCore.QRect(15, 190, 571, 192))

        self.twStaffs.setStyleSheet("color: rgb(20, 255, 236)")

        self.twStaffs.setObjectName("twStaffs")

        self.twStaffs.setColumnCount(0)

        self.twStaffs.setRowCount(0)

        self.splitter = QtWidgets.QSplitter(Form)

        self.splitter.setGeometry(QtCore.QRect(430, 50, 75, 46))

        self.splitter.setOrientation(QtCore.Qt.Vertical)

        self.splitter.setObjectName("splitter")

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

        self.pushButton.setStyleSheet("color: rgb(20, 255, 236)")

        self.pushButton.setObjectName("pushButton")

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

        self.pushButton\_2.setStyleSheet("color: rgb(20, 255, 236)")

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

        self.widget = QtWidgets.QWidget(Form)

        self.widget.setGeometry(QtCore.QRect(20, 30, 311, 141))

        self.widget.setObjectName("widget")

        self.formLayout = QtWidgets.QFormLayout(self.widget)

        self.formLayout.setContentsMargins(0, 0, 0, 0)

        self.formLayout.setObjectName("formLayout")

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

        self.label.setStyleSheet("color: rgb(20, 255, 236)")

        self.label.setObjectName("label")

        self.formLayout.setWidget(0, QtWidgets.QFormLayout.LabelRole, self.label)

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

        self.label\_2.setStyleSheet("color: rgb(20, 255, 236)")

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

        self.formLayout.setWidget(1, QtWidgets.QFormLayout.LabelRole, self.label\_2)

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

        self.label\_3.setStyleSheet("color: rgb(20, 255, 236)")

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

        self.formLayout.setWidget(2, QtWidgets.QFormLayout.LabelRole, self.label\_3)

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

        self.label\_4.setStyleSheet("color: rgb(20, 255, 236)")

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

        self.formLayout.setWidget(3, QtWidgets.QFormLayout.LabelRole, self.label\_4)

        self.leID = QtWidgets.QLineEdit(self.widget)

        self.leID.setObjectName("leID")

        self.formLayout.setWidget(0, QtWidgets.QFormLayout.FieldRole, self.leID)

        self.lename = QtWidgets.QLineEdit(self.widget)

        self.lename.setObjectName("lename")

        self.formLayout.setWidget(1, QtWidgets.QFormLayout.FieldRole, self.lename)

        self.lecode = QtWidgets.QLineEdit(self.widget)

        self.lecode.setObjectName("lecode")

        self.formLayout.setWidget(2, QtWidgets.QFormLayout.FieldRole, self.lecode)

        self.lecount = QtWidgets.QLineEdit(self.widget)

        self.lecount.setObjectName("lecount")

        self.formLayout.setWidget(3, QtWidgets.QFormLayout.FieldRole, self.lecount)

        self.retranslateUi(Form)

        QtCore.QMetaObject.connectSlotsByName(Form)

    def retranslateUi(self, Form):

        \_translate = QtCore.QCoreApplication.translate

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

        self.pushButton.setText(\_translate("Form", "Open"))

        self.pushButton\_2.setText(\_translate("Form", "Insert"))

        self.label.setText(\_translate("Form", "ID"))

        self.label\_2.setText(\_translate("Form", "name"))

        self.label\_3.setText(\_translate("Form", "code"))

        self.label\_4.setText(\_translate("Form", "count"))

<?xml version="1.0" encoding="UTF-8"?>

<ui version="4.0">

 <class>Form</class>

 <widget class="QWidget" name="Form">

  <property name="geometry">

   <rect>

    <x>0</x>

    <y>0</y>

    <width>609</width>

    <height>415</height>

   </rect>

  </property>

  <property name="sizePolicy">

   <sizepolicy hsizetype="Preferred" vsizetype="Preferred">

    <horstretch>0</horstretch>

    <verstretch>0</verstretch>

   </sizepolicy>

  </property>

  <property name="windowTitle">

   <string>Form</string>

  </property>

  <property name="styleSheet">

   <string notr="true">background-color: rgb(33, 33, 33)</string>

  </property>

  <widget class="QTableWidget" name="twStaffs">

   <property name="geometry">

    <rect>

     <x>15</x>

     <y>190</y>

     <width>571</width>

     <height>192</height>

    </rect>

   </property>

   <property name="styleSheet">

    <string notr="true">color: rgb(20, 255, 236)</string>

   </property>

  </widget>

  <widget class="QSplitter" name="splitter">

   <property name="geometry">

    <rect>

     <x>430</x>

     <y>50</y>

     <width>75</width>

     <height>46</height>

    </rect>

   </property>

   <property name="orientation">

    <enum>Qt::Vertical</enum>

   </property>

   <widget class="QPushButton" name="pushButton">

    <property name="styleSheet">

     <string notr="true">color: rgb(20, 255, 236)</string>

    </property>

    <property name="text">

     <string>Open</string>

    </property>

   </widget>

   <widget class="QPushButton" name="pushButton\_2">

    <property name="styleSheet">

     <string notr="true">color: rgb(20, 255, 236)</string>

    </property>

    <property name="text">

     <string>Insert</string>

    </property>

   </widget>

  </widget>

  <widget class="QWidget" name="">

   <property name="geometry">

    <rect>

     <x>20</x>

     <y>30</y>

     <width>311</width>

     <height>141</height>

    </rect>

   </property>

   <layout class="QFormLayout" name="formLayout">

    <item row="0" column="0">

     <widget class="QLabel" name="label">

      <property name="styleSheet">

       <string notr="true">color: rgb(20, 255, 236)</string>

      </property>

      <property name="text">

       <string>ID</string>

      </property>

     </widget>

    </item>

    <item row="1" column="0">

     <widget class="QLabel" name="label\_2">

      <property name="styleSheet">

       <string notr="true">color: rgb(20, 255, 236)</string>

      </property>

      <property name="text">

       <string>name</string>

      </property>

     </widget>

    </item>

    <item row="2" column="0">

     <widget class="QLabel" name="label\_3">

      <property name="styleSheet">

       <string notr="true">color: rgb(20, 255, 236)</string>

      </property>

      <property name="text">

       <string>code</string>

      </property>

     </widget>

    </item>

    <item row="3" column="0">

     <widget class="QLabel" name="label\_4">

      <property name="styleSheet">

       <string notr="true">color: rgb(20, 255, 236)</string>

      </property>

      <property name="text">

       <string>count</string>

      </property>

     </widget>

    </item>

    <item row="0" column="1">

     <widget class="QLineEdit" name="leID"/>

    </item>

    <item row="1" column="1">

     <widget class="QLineEdit" name="lename"/>

    </item>

    <item row="2" column="1">

     <widget class="QLineEdit" name="lecode"/>

    </item>

    <item row="3" column="1">

     <widget class="QLineEdit" name="lecount"/>

    </item>

   </layout>

  </widget>

 </widget>

 <resources/>

 <connections/>

</ui>

Все остальные формы работают аналогично данной форме Service.

**Заключение**

В ходе проделанной работы были закреплены теоретические знаний, а также навыки проектирования БД, полученные при изучении дисциплины «Базы данных».

Выполнены все поставленные задачи: была разработана Er-diagramma, по Er-diagramme, была создана база данных, и была заполнена определенными данными, так же был разработан интерфейс непосредственно под самубазу данных и были созданы файлу с разрешением .py где мы связывали и создавали формы для работы с базами данных.