

Содержание

Часть 1..........................................................................................................................3

1. Описание подготовительной работы.....................................................................3

Часть 2........................................................................................................................13

2. Описание учебного приложения для работы с базой данных...........................13

2.1. Структура БД.......................................................................................................13

2.2. Описание алгоритма...........................................................................................13

2.3. Тестирование.......................................................................................................18

Часть 3........................................................................................................................20

3. Описание Приложения для выполнения индивидуального задания................20

3.1. Индивидуальное задание...................................................................................20

3.2. Описание программы.........................................................................................22

3.3. Тестирование.......................................................................................................28

# Часть 1

**1. Описание подготовительной работы**

В выполнении научно-исследовательской работы (НИР) выделено 3 последовательных этапа, рассмотрела первый:

1) Изучение основ реляционных баз данных и основных операторов языка SQL. Получение начальных сведений о базах данных (БД). На этом этапе познакомилась с основами устройства реляционных БД и терминологией предметной области, изучила специальный язык запросов к данным SQL и способы выполнения операций с БД в программах, разрабатываемых в программной среде Python.

Поскольку в данном семестре одновременно изучается программная среда Python, будут изучаться способы работы с БД SQLite из программ, написанных с использованием Python + SQL для SQLite.

# 1.1.Этап 1 НИР. Работа с данными SQLite в программах на языке Python.

Средства для работы с БД SQLite содержатся в среде Python сразу после ее инсталляции. Тем не менее, перед началом работы в среде Python с БД SQLite проверила:

- Наличие библиотеки **sqlite3.dll** – по месту установки Python в каталоге DLLS.

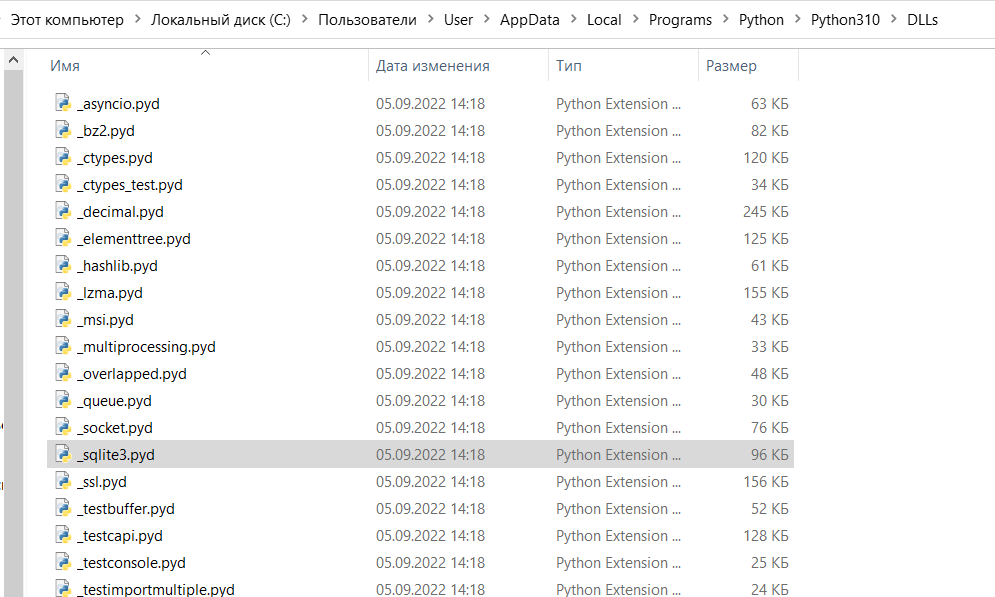


Рис.1. Проверка наличия библиотеки **sqlite3.dll.**

- Наличие подкаталога (пакета) **sqlite3** – в каталоге **Lib**.

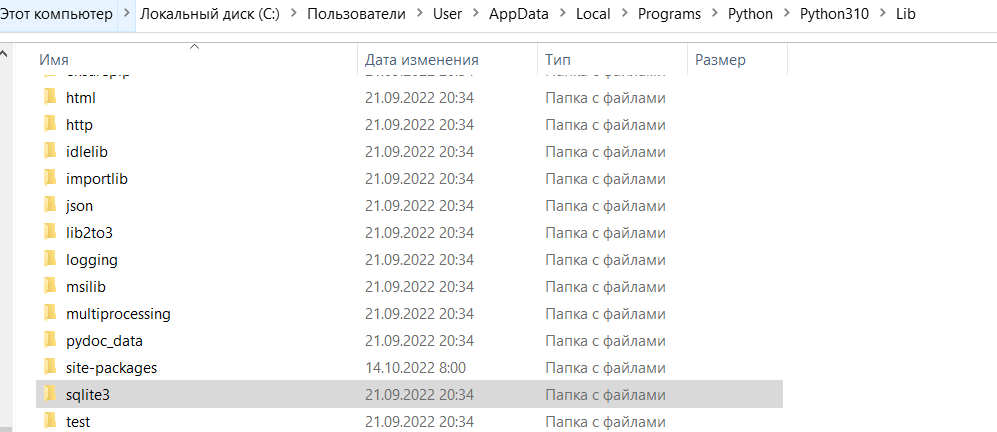


Рис.2. Проверка наличия подкаталога **sqlite3.**

# 1.2. Подготовительные операции

Создала специальный каталог для работы на этом этапе НИР и установить его в качестве рабочего каталога.

>>> **import os**

>>> **os.chdir('C:\\Users\\User\\Documents\\ПОАС\\')**

После этого импортировала модуль **sqlite3** для работы с SQLite

**>>> import sqlite3**

**1.3. Создание БД bd1.sqlite с 2 таблицами: stud и sessija**

База данных сохраняется в файле с заданным при их создании именем и с расширением **.sqlite**. Отдельные таблицы (вне БД) сохраняются в файлах с заданными при их создании именами и с расширением **.db**.

Создала новую БД

**>>> con=sqlite3.connect('bd1.sqlite')**

Создала переменную-курсор

**>>>** **cur=con.cursor()**

Создала символьную переменную со скриптом – SQL-командами создания в БД двух новых таблиц (**stud** и **sessija**) с заданными полями. В таблице **stud** нужны поля:

- код зачетки студента (первичный ключ) ;

- ФИО студента;

- e-mail студента

В таблице **sessija** каждая запись соответствует сдаче экзамена или зачета по одной из дисциплин учебного плана одним из студентов из числа представленных в таблице **stud**. Здесь нужны поля:

- код зачетки студента (вторичный ключ);

- название дисциплины;

- полученная оценка;

- ФИО преподавателя;

- дата сдачи.

**>>> sql="""\**

**CREATE TABLE stud (id TEXT PRIMARY KEY,**

**fio TEXT, email TEXT);**

**CREATE TABLE sessija (id TEXT SECONDARY KEY,**

**Disz TEXT, Ozenka INTEGER, Prepod TEXT, Dat TEXT)**

**"""**

Выполнила совокупность SQL-команд (скрипт)

**>>> cur.executescript(sql)**

Закрыла курсор и БД

**>>> cur.close()**

**>>> con.close()**

Теперь БД создана, определена ее структура, но пока она пустая – в ней нет данных!

**1.4. Запись данных в таблицы stud и sessija**

Установила соединение с ранее созданной БД

**>>> con=sqlite3.connect('bd1.sqlite')**

Создала курсор.

**>>> cur=con.cursor()**

Создала переменную с текстом SQL- команды вставки записи в таблицу **stud**

**>>> sql="""\**

**INSERT INTO stud (id,fio,email) VALUES ('0020202330','Андриянова К.Б.','AndrianovaKB@mpei.ru')**

**"""**

Выполнила отдельную SQL-команду

**>>> cur.execute(sql)**

Зафиксировала выполнение команды

**>>> con.commit()**

Аналогичные операции для вставки еще одной записи в stud

**>>> sql="""\**

**INSERT INTO stud (id,fio,email) VALUES ('0020203733','Воеводова Н.А.','VoyevodovaNA@mpei.ru')**

**"""**

**>>> cur.execute(sql)**

**>>> con.commit()**

Закрываю курсор и БД.

**>>> cur.close()**

**>>> con.close()**

# 1.5. Множественная вставка данных в таблицы БД

Создала список с 2-мя кортежами – для записи в таблицу **sessija**. Здесь в кортеже 1-е число – код зачетки, 2-ой элемент – название предмета, 3-ий – ФИО преподавателя, 4-ый – оценка за экзамен, 5-ый – дата выставления оценки.

**arr=[('0020200989', 'Теоретическая механика', 'Панкратьева Г.В.', 5, '02.07.2021'),( '0020200990', ' Математический анализ', 'Бирюков А.М.',5, '25.06.2021')]**

Как и в таблице stud использовала в таблице sessija одинаковых студентов:

**0020202330 –** Андриянова К.Б., **0020203733 –** Воеводова Н.А.

Выполняю «множественную» команду вставки строк из arr

**>>> con=sqlite3.connect('bd1.sqlite')**

**>>> cur=con.cursor()**

**>>> sql="""\**

**INSERT INTO sessija (id, Disz, Ozenka, Prepod, Dat)**

**VALUES (?,?,?,?,?)**

**"""**

Выполнила «множественную» команду вставки строк из **arr**

**>>> cur.executemany(sql,arr)**

**>>> con.commit()**

В таблицу **sessija** вставила 2 строки с данными из кортежей в **arr**.

Закрыла курсор и БД.

**>>> cur.close()**

**>>> con.close()**

# 1.6. Чтение и отображение содержимого одной из таблиц в БД

Создаем и выполняем программу test3.py, позволяющую увидеть содержимое указанной таблицы в заданной БД.

Записала программу в файл test3.py:

**import os,sqlite3**

**def select\_cmd():**

**#просмотр таблицы с именем в tblname из БД с именем в dbname**

**sql = 'SELECT \* FROM {}'.format(tblname)**

**with con:**

**data = cur.execute(sql).fetchall()**

**return (data)**

**dbname=' '**

**###Выбор БД с проверкой ее наличия**

**while (os.path.isfile(dbname)!=True):**

**dbname = input('Укажите имя файла SQLite: ')**

**if (os.path.isfile(dbname)==True): break**

**print('Нет такого файла!')**

**tblname = input('Укажите имя таблицы: ') #ввод имени таблицы**

**con = sqlite3.connect(dbname)**

**cur = con.cursor()**

**dan=select\_cmd()**

**nzap=len(dan)**

**print('Таблица: ',tblname,' из БД ',dbname)**

**for i in range(nzap):**

**print(dan[i])**

**cur.close()**

**con.close()**

Запустила на выполнение программу test3 и ввела имена ранее созданной БД и одной из таблиц

================ RESTART: C:/Users/User/Documents/ПОАС/test3.py

Укажите имя файла SQLite: bd1.sqlite

Укажите имя таблицы: stud

Таблица: stud из БД bd1.sqlite

('0020202330', 'Андриянова К.Б.', 'AndrianovaKB@mpei.ru')

('0020203733', 'Воеводова Н.А.', 'VoyevodovaNA@mpei.ru')

Отобразились данные таблицы stud.

Аналогичным образом вывела данные из второй таблицы:

================ RESTART: C:/Users/User/Documents/ПОАС/test3.py

Укажите имя файла SQLite: bd1.sqlite

Укажите имя таблицы: sessija

Таблица: sessija из БД bd1.sqlite

('0020200989', 'Теоретическая механика', 'Панкратьева Г.В.', '5', '02.07.2021')

('0020200990', ' Математический анализ', 'Бирюков А.М.', '5', '25.06.2021')

Отобразились данные таблицы sessija.

# 1.7. Чтение данных из таблицы БД

Пусть имеется БД **bd1.sqlite**, содержащая таблицу **stud**. Получим содержимое всех записей этой таблицы.

Открыла БД и создала курсор:

**>>> con=sqlite3.connect("bd1.sqlite")**

**>>> cur=con.cursor()**

Выполнила SQL- команду для чтения всего содержимого указанной таблицы из БД в курсор:

**>>> cur.execute("SELECT \* FROM stud")**

Переписала содержимое курсора в список с указанным именем (элементы списка – кортежи со значениями полей в записях):

**>>> ar=cur.fetchall()**

Закрыла курсор и соединение с БД:

**>>> cur.close()**

**>>> con.close()**

**>>> print(ar)**

[('0020202330', 'Андриянова К.Б.', 'AndrianovaKB@mpei.ru'), ('0020203733', 'Воеводова Н.А.', 'VoyevodovaNA@mpei.ru')]

# 1.8. Получение списка таблиц в составе БД.

Программы могут работать с разными базами данных (подобно тому, как они могут работать с разными файлами). Поэтому важно уметь в программе определить структуру БД, с которой она начинает работу.

Для того, чтобы определить, какие таблицы входят в состав БД выполнила следующую последовательность операторов:

Открыла БД – устанавила связь с **con**

**>>> con=sqlite3.connect('vuz.sqlite')**

Создала курсор

**>>> cur=con.cursor()**

Cформировала SQL- запрос на выборку данных о структуре БД

**>>> sql="""\**

**SELECT name FROM sqlite\_master WHERE type IN ('table','view') AND name NOT LIKE 'sqlite\_%' UNION ALL SELECT name FROM sqlite\_temp\_master WHERE type IN ('table','view') ORDER BY 1;**

**"""**

Исполнила запрос

**>>> cur.execute(sql)**

На всякий случай зафиксировала его

**>>> con.commit()**

Создала список кортежей, в которых содержатся имена входящих в БД таблиц (при необходимости его можно сохранить в отдельном объекте и извлечь из кортежей имена таблиц)

**>>> cur.fetchall()**

[('vuz\_rf',), ('vuzkart',), ('vuzstat',)]

В БД **vuz.sqlite** находятся 3 таблицы с данными именами.

Закрыла курсор и БД

**>>> cur.close()**

**>>> con.close()**

# 1.9. Считывание информации о структуре таблицы, имеющейся в БД

Рассмотрела ту же БД **VUZ.sqlite**. Получила список имен всех полей таблицы **vuzkart**:

Установила соединение с БД

**>>> con=sqlite3.connect('VUZ.sqlite')**

Создала вспомогательную функцию

**>>> def my\_factory(c,r):**

**###Функция для извлечения имен полей**

**d={}**

**for i,name in enumerate(c.description):**

**d[name[0]]=r[i]**

**d[i]=r[i]**

**return(d)**

Записала эту функцию в атрибут **row\_factory** объекта **con** (БД)

**>>> con.row\_factory=my\_factory**

Создала курсор

**>>> cur=con.cursor()**

Исполнила SQL запрос

**>>> cur.execute('SELECT \* FROM vuzkart')**

Записала в объект **ar** (словарь) данные из одной записи в БД

**>>> ar=cur.fetchone()**

Как только БД больше не нужна – закрыла курсор и БД!

**>>> cur.close()**

**>>> con.close()**

Записала в объект names список имен полей таблицы

**>>> fld\_names=list(ar.keys())[::2]**

Отобразила этот список

**>>> print(fld\_names)**

**['codvuz', 'z1', 'z1full', 'z2', 'z2ustav', 'foundyear', 'z8', 'z9', 'z12', 'e\_mail', 'www', 'z15', 'region', 'city', 'status', 'oblname', 'gr\_ved', 'prof']**

В таблице **vuzkart** находятся 18 полей с данными именами.

# 1.10. Ввод данных по запросу с клавиатуры и занесение их в таблицу

При выполнении этих операций использовала ранее созданную БД **bd1.sqlite** и данные еще одного студента из группы.

**>>> con=sqlite3.connect('bd1.sqlite')**

**>>> cur=con.cursor()**

**>>> ar=[]; vv=[]**

**>>> vv.append(input('Код зачетки студента='))**

В ответ на запрос ввела код зачетки одного из студентов (из числа ранее не вводившихся в таблицу **stud**).

Код зачетки студента=20202335

**>>> vv.append(input('ФИО студента = '))**

В ответе на запрос указала ФИО студента.

ФИО студента = Жильцов А.И.

**>>> vv.append(input('e-mail студента ='))**

Ввела e-mail студента.

e-mail студента =ZhiltsovAI@mpei.ru

**>>> ar.append(tuple(vv))**

Отобразила объект **ar**.

**>>>** ar

[('20202335', 'Жильцов А.И.', 'ZhiltsovAI@mpei.ru')]

Создала и исполнила SQL-запрос для занесения новой строки в таблицу **stud**.

**>>> sql='INSERT INTO stud (id,fio,email) VALUES (?,?,?)'**

**>>> cur.executemany(sql,ar)**

**>>> con.commit()**

Теперь строка занесена в таблицу и для контроля отобразила все ее содержимое.

**>>> cur.execute('SELECT \* FROM stud')**

**>>> ar1=cur.fetchall()**

**>>> print(ar1)**

**[('0020202330', 'Андриянова К.Б.', 'AndrianovaKB@mpei.ru'), ('0020203733', 'Воеводова Н.А.', 'VoyevodovaNA@mpei.ru'), ('20202335', 'Жильцов А.И.', 'ZhiltsovAI@mpei.ru')]**

**>>> cur.close()**

**>>> con.close()**

**Часть 2**

**2. Описание учебного приложения для работы с базой данных**

**2.1 Структура БД**

Цель этой части научно-исследовательской работы – получение первичного опыта создания программы для работы с базой данных с использованием среды Python.

В результате выполнения этой части НИР должна появиться БД, включающая одну таблицу со следующей структурой:

| **Содержание поля** | **Тип информации в поле** |
| --- | --- |
| Код дисциплины по учебному плану | текст |
| Название дисциплины | текст |
| Номер семестра с аттестацией по дисциплине | целое число |
| Тип аттестации (экзамен/зачет) | текст |
| Дата аттестации | дата |
| ФИО преподавателя, проводившего аттестацию | текст |
| Должность преподавателя | текст |
| Полученная оценка | целое число |
| Дата занесения/обновления записи | дата |

**2.2. Описание алгоритма**

На примере этой БД следует освоить программирование работы с данными в среде Python. В этой части работы необходимо выполнить следующие операции:

1. Создать базу данных, содержащую таблицу с заданной структурой. Имена БД, таблицы и полей выбрать самостоятельно.

>>> import os

>>> os.chdir('C:\\Users\\User\\Documents\\ПОАС\\')

>>> import sqlite3

>>> con=sqlite3.connect('bd2.sqlite')

>>> cur=con.cursor()

>>> sql="""\

CREATE TABLE ID(DisciplineCode TEXT, NameOfDiscipline TEXT, SemesterNumber INTEGER, AttestationType TEXT,AttestationDate TEXT, TeacherName TEXT, TeachingPosition TEXT, Score INTEGER, RecordingDate TEXT)

"""

>>> cur.executescript(sql)

<sqlite3.Cursor object at 0x000001D76EC87840>

>>> cur.close()

>>> con.close()

1. Разработать программу на языке Python, обеспечивающую открытие созданной БД и выполнение следующих функций по выбору пользователя:

- Отображение текущего содержимого таблицы БД на экране в виде таблицы.

- Сохранение текущего содержимого таблицы БД в текстовый файл с задаваемым именем.

- Выбор пользователем имени одного из полей БД и задание условия по значениям этого поля (логическое выражение). Отображение в виде таблицы подмножества строк, удовлетворяющих заданному условию.

- Выбор операции с подмножеством строк: удаление из БД, замена значений во всех строках в указанном поле на заданное значение.

- Добавление новой строки с заданными значениями полей в таблицу БД.

- Завершение работы с программой.

def select\_cmd():

sql = 'SELECT \* FROM {}'.format(name)

with con:

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

return (data)

#создала список кортежей данных моей зачетной книжки

arr=[('B1.0.10','Физика',1,'экзамен','18.01.2021','Корецкая И.В.','старший преподаватель',4,'12.07.2022'),

('B1.0.13.03','Математический анализ ч.2',2,'зачет','17.06.2021','Булычева О.Н.','доцент',5,'12.07.2022'),

('B1.0.13.02','Математический анализ',2,'экзамен','08.07.2022','Мамонтов А.И.','доцент',5,'12.07.2022'),

('B1.0.04','Программирование и основы алгоритмизации',1,'экзамен','27.01.2021','Меньшикова К.Г.','доцент',4,'12.07.2022'),

('B1.0.16','СМИИ',4,'зачет','09.06.2022','Рябов С.Н.','доцент',5,'12.07.2022'),

('B1.0.13.04','Вычислительные методы',4,'зачет','14.06.2022','Бирюков А.М.','доцент',4,'12.07.2022'),

('B1.0.22','Методы оптимизации',4,'экзамен','17.06.2022','Скибицкий Н.В.','доцент',4,'12.07.2022'),

('B1.0.18','Электротехника',4,'экзамен','22.06.2022','Федорова Е.М.','доцент',4,'12.07.2022'),

('B1.0.19','Электроника',4,'экзамен','28.06.2022','Жохова М.П.','доцент',5,'12.07.2022'),

('B1.0.20','ИТ',4,'экзамен','04.07.2022','Фомин Г.А.','доцент',4,'12.07.2022'),

('B1.0.16','СМИИ',3,'экзамен','09.06.2022','Рябов С.Н.','доцент',5,'12.07.2022')]

con=sqlite3.connect('bd3.sqlite')

cur=con.cursor()

sql="""\

INSERT INTO ID (DisciplineCode, NameOfDiscipline, SemesterNumber,

AttestationType,AttestationDate, TeacherName,

TeachingPosition, Score, RecordingDate)

VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?);

"""

cur.executemany(sql,arr) #загрузила данные списка arr в базу данных

con.commit()

cur.close()

con.close()

bd=' '

while (os.path.isfile(bd)!=True):

bd = input('Укажите имя файла SQLite: ')

if(os.path.isfile(bd)==True): break

print('Нет такого файла!')

flag=True

while(flag):

con = sqlite3.connect(bd)

cur = con.cursor()

name=cur.execute("SELECT name FROM sqlite\_master WHERE TYPE = 'table' ").fetchall()[0][0]

cur.execute('SELECT \* FROM {}'.format(name))

dan=select\_cmd()

kolstr=len(dan)

cur.close()

con.close()

operation=int(input("""

Действия с таблицей:

1 - отображение текущего содержимого

2 - сохранение текущего содержимого в текстовый файл

3 - отображение строки по заданному условию

4 - удаление, замена значений на заданное

5 - добавление строки

6 - завершение работы

"""))

if operation==1:

print('Таблица: ',name,' из БД ',bd)

for i in range(kolstr):

print(dan[i])

if operation==2:

filename=input("Введите имя файла: ")

file=open(filename,'a')

file.write("Таблица:"+name+"Из БД: "+bd+'\n')

for i in range(kolstr):

file.write(str(dan[i])+"\n")

file.close()

print("Информация сохранена в файл")

if operation==3:

NamePole=input("Введите имя поля: ")

sql = 'SELECT {} FROM {}'.format(NamePole,name)

condition=input("Введите условие выполнения операции:\n")

if condition != '':

sql += ' WHERE {}'.format(condition)

con = sqlite3.connect(bd)

cur = con.cursor()

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

cur.close()

con.close()

print(data)

if operation==4:

oper=input("""

Выберите вид операции:

del - удаление

upd - замена\n""")

condition=input("Введите условие выполнения операции:\n")

if(oper=="del"):

sql = 'DELETE FROM {}'.format(name)

if condition != '':

sql+=' WHERE {}'.format(condition)

con=sqlite3.connect(bd)

cur=con.cursor()

cur.execute(sql)

con.commit()

cur.close()

con.close()

if(oper=="upd"):

pole=input('Введите имя поля:')

znach=input('Введите значение для замены:')

s=' '

s+=pole+"="+znach

sql='UPDATE {} SET {} WHERE {}'.format(name,s,condition)

con=sqlite3.connect(bd)

cur=con.cursor()

cur.execute(sql)

con.commit()

cur.close()

con.close()

if operation==5:

sql="""\

INSERT INTO ID (DisciplineCode, NameOfDiscipline, SemesterNumber,

AttestationType,AttestationDate, TeacherName,

TeachingPosition, Score, RecordingDate)

VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?);

"""

DC=input('Код дисциплины по учебному плану')

NOD=input('Название дисциплины')

SN=int(input('Номер семестра с аттестацией по дисциплине'))

AT=input('Тип аттестации')

AD=input('Дата аттестации')

TN=input('ФИО преподавателя, проводившего аттестацию')

TP=input('Должность преподавателя')

S=int(input('Полученная оценка'))

RD=input('Дата занесения/обновления записи')

data\_tuple=(DC,NOD,SN,AT,AD,TN,TP,S,RD)

con=sqlite3.connect(bd)

cur=con.cursor()

cur.execute(sql, data\_tuple)

con.commit()

cur.close()

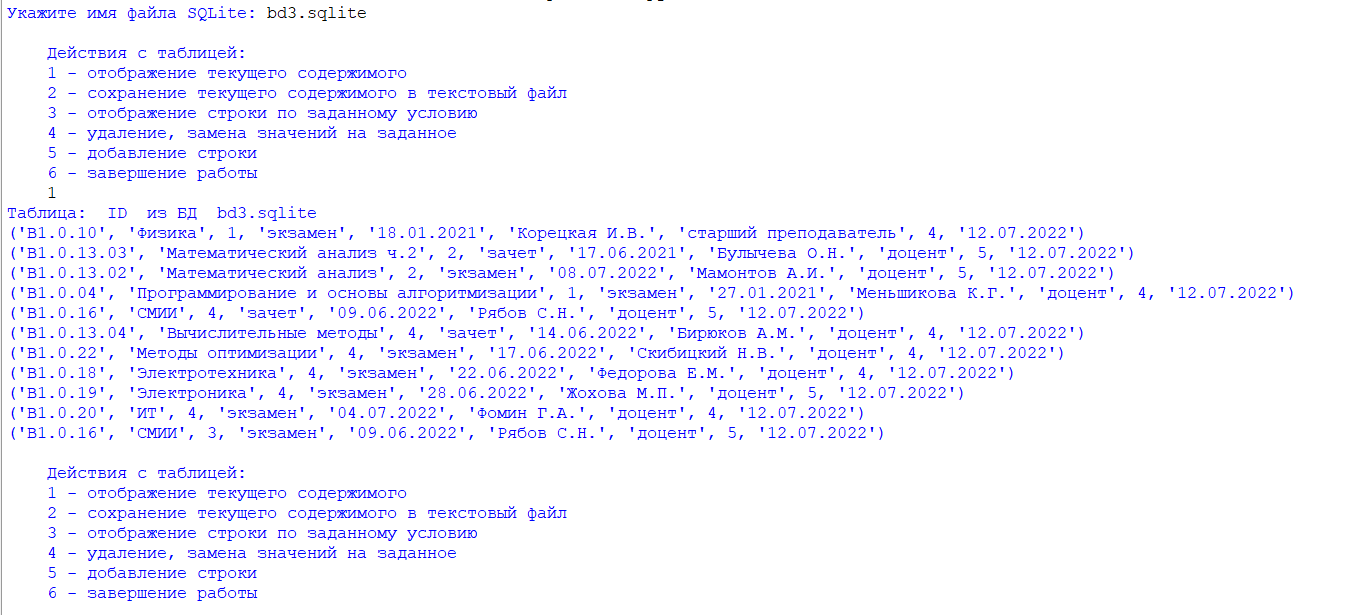
con.close()

if operation==6:

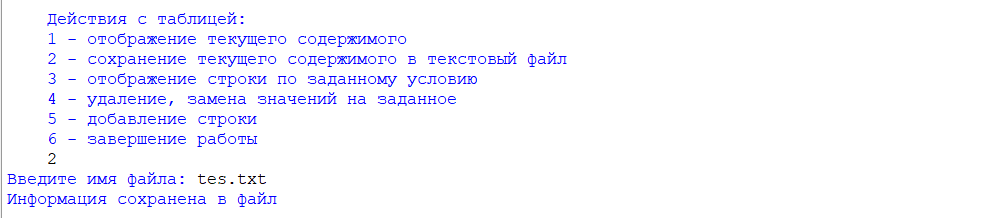
flag=False

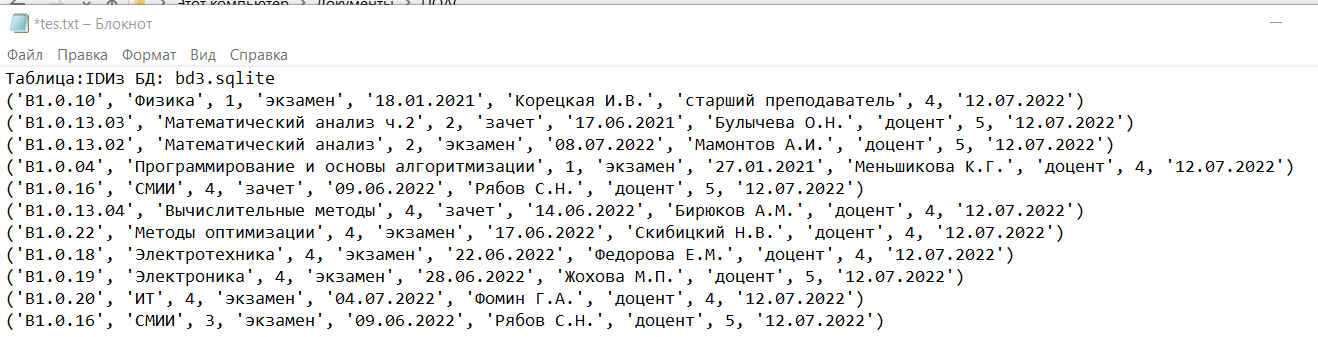
**Тестирование**

1) Отображение текущего содержимого таблицы БД на экране в виде таблицы.

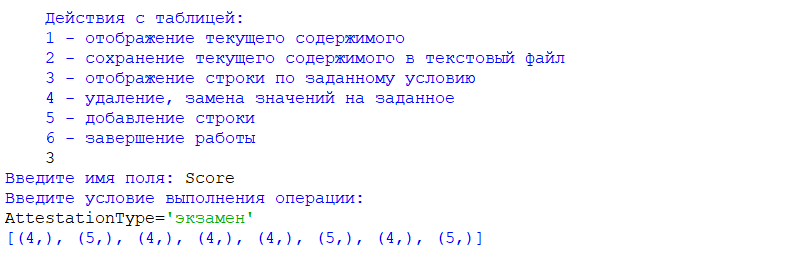


2) Сохранение текущего содержимого таблицы БД в текстовый файл с задаваемым именем.





3) Выбор пользователем имени одного из полей БД и задание условия по значениям этого поля (логическое выражение). Отображение в виде таблицы подмножества строк, удовлетворяющих заданному условию.



4) Выбор операции с подмножеством строк:

- замена значений во всех строках в указанном поле на заданное значение



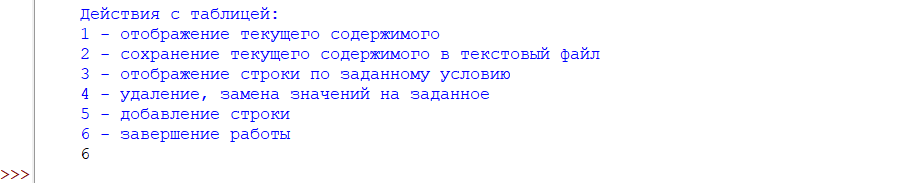
- удаление из БД



5) Добавление новой строки с заданными значениями полей в таблицу БД.



6) Завершение работы с программой.



**Часть 3**

# 3. Описание Приложения для выполнения индивидуального задания

При выполнении этой части НИР студент должен получить у руководителя НИР номер варианта своего индивидуального задания, создать отдельный каталог для решения задачи и записать в него БД **VUZ.sqlite** (исходные данные для всех индивидуальных заданий)**.** После этого необходимо внимательно изучить содержание индивидуального задания и, при необходимости, получить необходимые разъяснения у руководителя.

**Внимание!** Помимо функций, приведенных в индивидуальном задании, программа должна обеспечивать отображение, по выбору пользователя, каждой из двух таблиц, содержащихся в БД, а также предложение завершить программу.

Используя среду Python, разработать программу, реализующую требуемую функциональность программы, произвести ее тестирование, выполнить требуемые расчет. Подготовить отчет по НИР в соответствии с требованиями из разд.7. Представить отчет на рассмотрение руководителю НИР.

**Исходные данные для выполнения индивидуальных заданий – БД VUZ.sqlite, содержащая 2 таблицы:**

1. Таблица "Картотека вузов" - **vuzkart** с записями, содержащими сведения о вузах России.
2. Таблица **vuzstat**, содержащая статистические данные по вузам.

**Структура таблиц с данными**

**Таблица vuzkart**

| Ключ | Имя столбца | Тип данных | Описание |
| --- | --- | --- | --- |
|  | codvuz | char(6) | Код вуза (первичный) |
|  | z1 | char(200) | Полное наименование вуза |
|  | z1full | char(250) | Полное юридическое наименование вуза |
|  | z2 | char(12) | Сокращённое наименование |
|  | z2ustav | char(100) | Сокращённое наименование по уставу |
|  | foundyear | integer(4) | Год основания вуза |
|  | z8 | char(110) | Юридический адрес вуза |
|  | z9 | char(40) | Справочный телефон вуза |
|  | z12 | char(20) | Номер факса вуза |
|  | e\_mail | char(30) | Адрес электронной почты |
|  | www | char(30) | Адрес официального сайта вуза |
|  | z15 | char(40) | Ректор вуза |
|  | region | char(17) | Федеральный округ |
|  | city | char(20) | Город |
|  | status | char(15) | Статус вуза: университет, академия, институт |
|  | oblname | char(40) | Субъект федерации по расположению вуза |
|  | gr\_ved | char(3) | Категория вуза - принадлежность к ведущим вузам (ФУ – федеральный университет, НИУ – национальный исследовательский университет) |
|  | prof | char(2) | Профиль вуза: ИТ – инженерно-технический, КЛ – классический университет, ГП – гуманитарно-педагогический, МП - многопрофильный |

**Таблица vuzstat**

| Ключ | Имя столбца | Тип данных | Описание |
| --- | --- | --- | --- |
|  | codvuz | char(6) | Код вуза (вторичный) |
|  | PPS | integer(4) | Численность профессорско-преподавательского состава (ППС) |
|  | PR | integer(4) | Профессорско-преподавательский состав (ППС) – профессоры |
|  | DC | integer(4) | ППС - доценты |
|  | DN | integer(4) | ППС - доктора наук |
|  | KN | integer(4) | ППС - кандидаты наук |
|  | ZOB | integer(4) | Количество зданий общежитий |
|  | STUD | integer(4) | Количество студентов, всего |
|  | BAC | integer(4) | Обучается по программам бакалавриата |
|  | SPEC | integer(4) | Обучается по программам подготовки специалистов |
|  | MAG | integer(4) | Обучается по программам магистратуры |
|  | ST\_OCH | integer(4) | Студентов по очной форме обучения |
|  | ST\_OCH\_Z | integer(4) | Студентов по очно-заочной форме обучения |
|  | ST\_Z | integer(4) | Студентов по заочной форме обучения |
|  | ASP | integer(4) | Аспирантов всего |
|  | ASP\_OCH | integer(4) | Аспирантов по очной форме обучения |

**3.1. Индивидульное задание**

Вариант 24.

1. Обеспечить выбор из списка федерального округа, интересующего пользо-

вателя. Составить и отобразить на экране перечень полных наименований ву-

зов, расположенных в выбранном округе и для которых имеется информация

о сайте вуза.

2. Обеспечить возможность пользователю выбрать из списка город России, в

котором находятся вузы, представленные в базе данных или выбрать значе-

ние «Все». Для выбранного города рассчитать и представить в виде таблицы

распределение количества студентов, обучающихся в вузах города, по уров-

ням подготовки (бакалавр, специалист, магистр). Таблица должна иметь

столбцы: порядковый номер, уровень подготовки, количество студентов,

обучающихся. в выбранном городе по данному уровню подготовки, процент

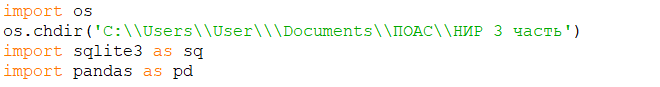
от общего количества студентов в городе. Последняя строка таблицы — ито-

говая, со значениями: в столбце «уровень подготовки» - значение «Все», в

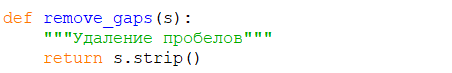
столбце «количество студентов» - общее количество студентов в городе.

**3.2. Описание программы**

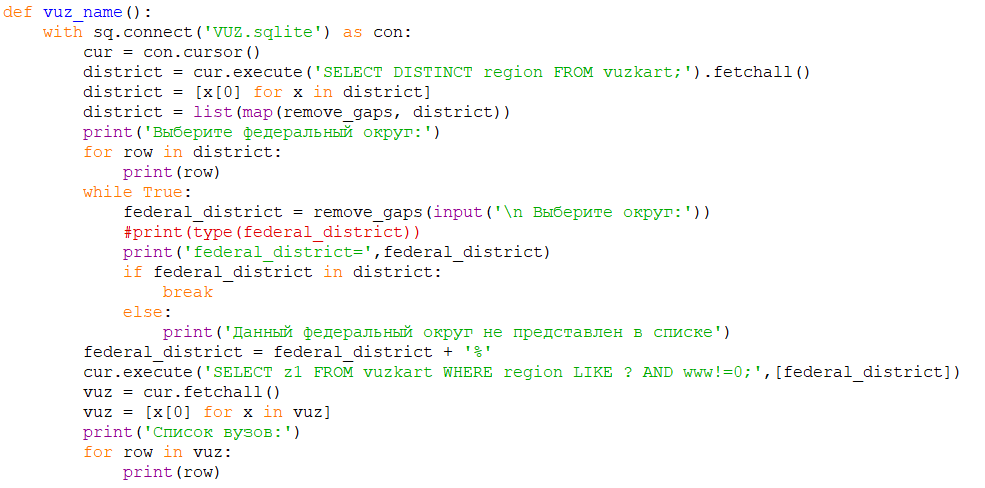
Импортировала необходимые библиотеки и установила рабочий каталог:



Написала функцию для удаления начальных и конечных пробелов:

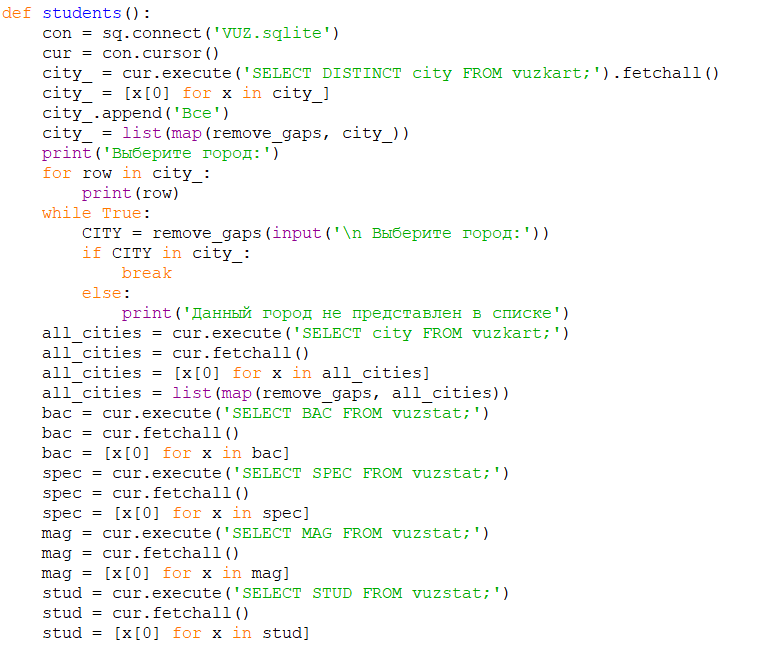


Функция для задачи 1. Использую конструкцию «with» для открытия и закрытия БД, так как она надежнее. При вызове функции, программа подключается к БД «VUZ.sqlite». Пользователь выбирает регион из представленного списка. Регионы извлекается из БД с условием «DISTINCT», то есть без повторов. Извлекаются они в переменную district, к каждому элементу которого применяется функция remove\_gaps для удаления пробелов. Происходит проверка: если пользователь ввел неверный регион, программа просит ввести регион снова. Затем из БД vuzkart извлекаются названия вузов, соответствующие выбранному региону, и у которых имеется информация о сайте, то есть www!=0.

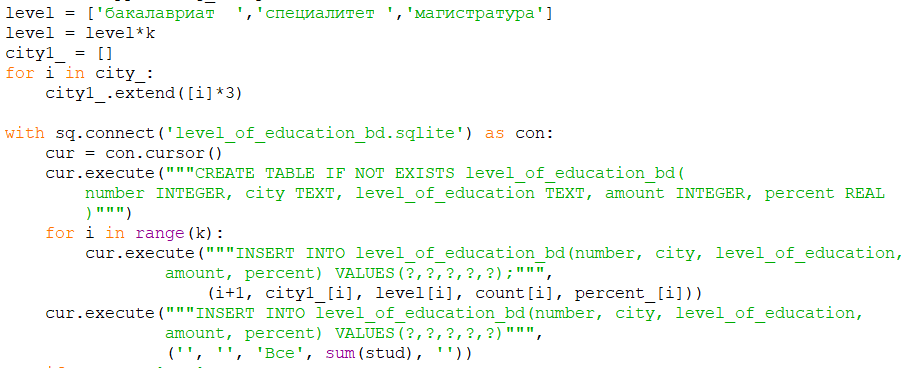


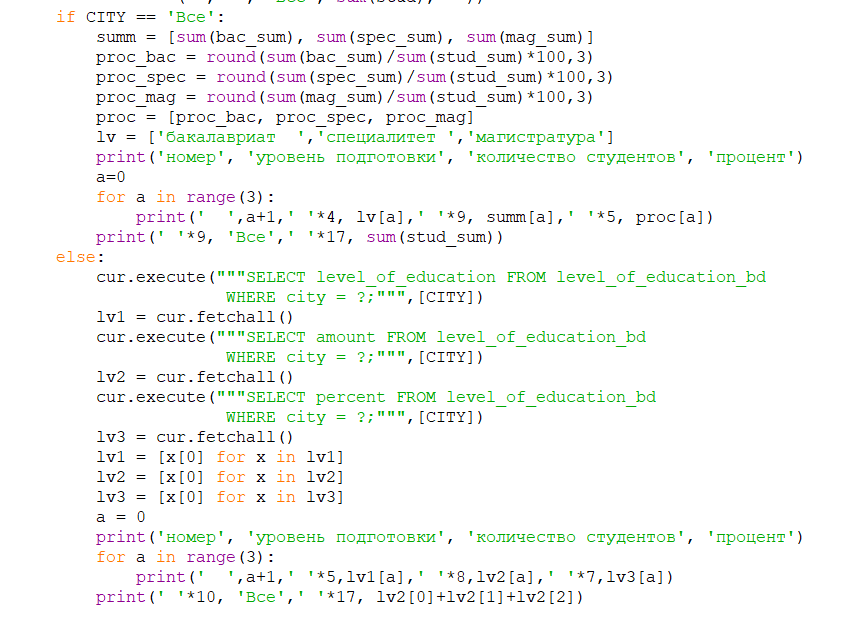
Функция для задачи 2. Программа просит пользователя ввести название города из представленного уникального списка городов. Затем проверяется наличие введенного города в списке, в случае его отсутствия программа выводит сообщение «Данный город не представлен в списке» и просит ввести город заново. Функция получает данные из 2 разных таблиц (список городов, количество бакалавров, специалистов и магистров, общее количество студентов).

Считаем эти данные для каждого города. Затем программа высчитывает процентное соотношение количества учащихся на определенном уровне подготовки к общему числу учащихся в каждом городе. Создаем свою БД level\_of\_education\_bd.sqlite, куда записываем данные всех городов по количеству обучающихся на разных уровнях подготовки. В случае выбора определенного города из нашей БД извлекаются нужные нам переменные, соответствующие выбранному городу и выводятся на консоль. При выборе пользователем «Все» считаются суммы количества обучающихся на каждом уровне подготовки по всем городам, по этим суммам высчитываются процентные соотношение количества обучающихся на каждом уровне подготовки к общему количеству студентов по всем городам. Затем результаты выводятся в виде таблички.

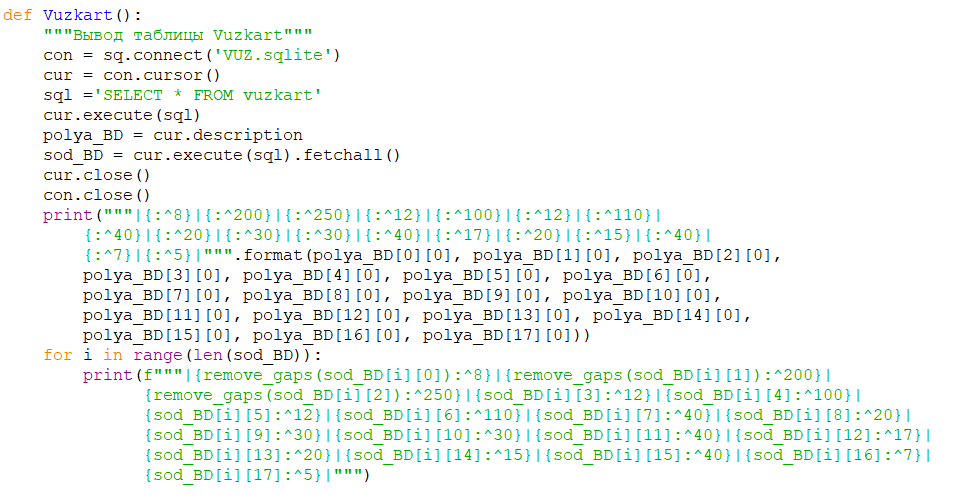




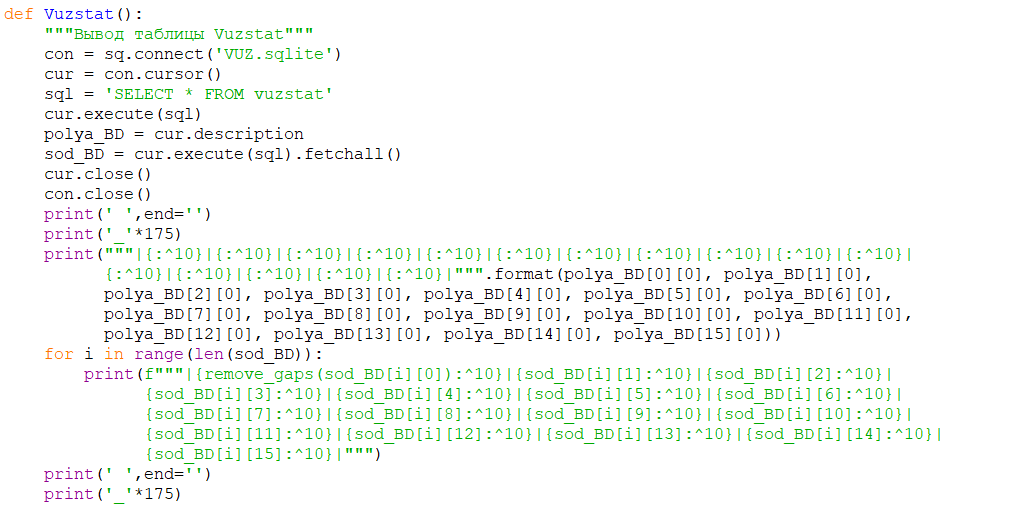




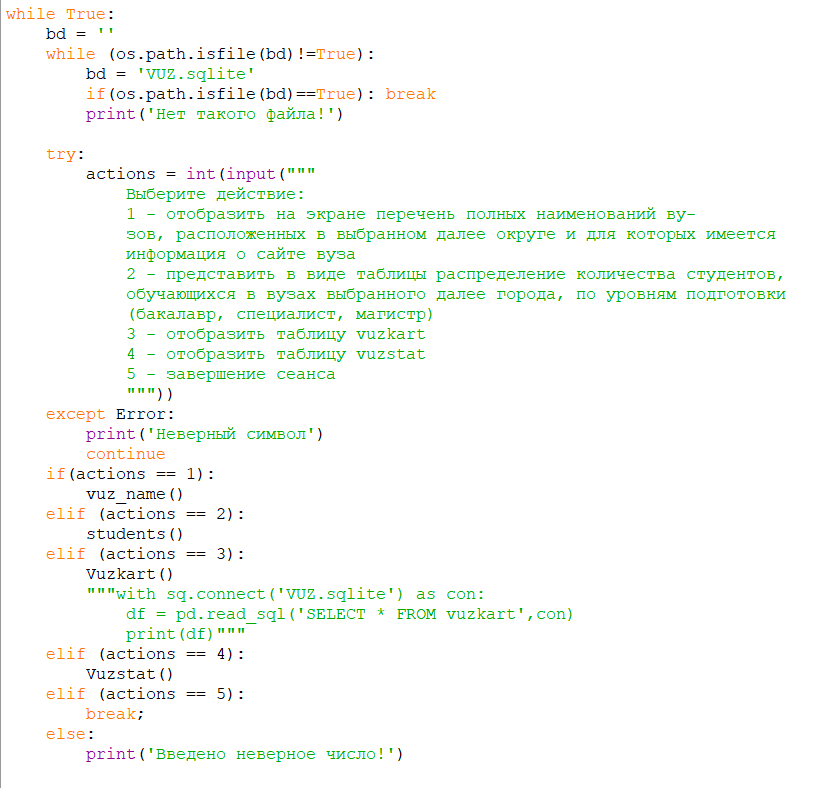
Эта функция выводит таблицу vuzkart.



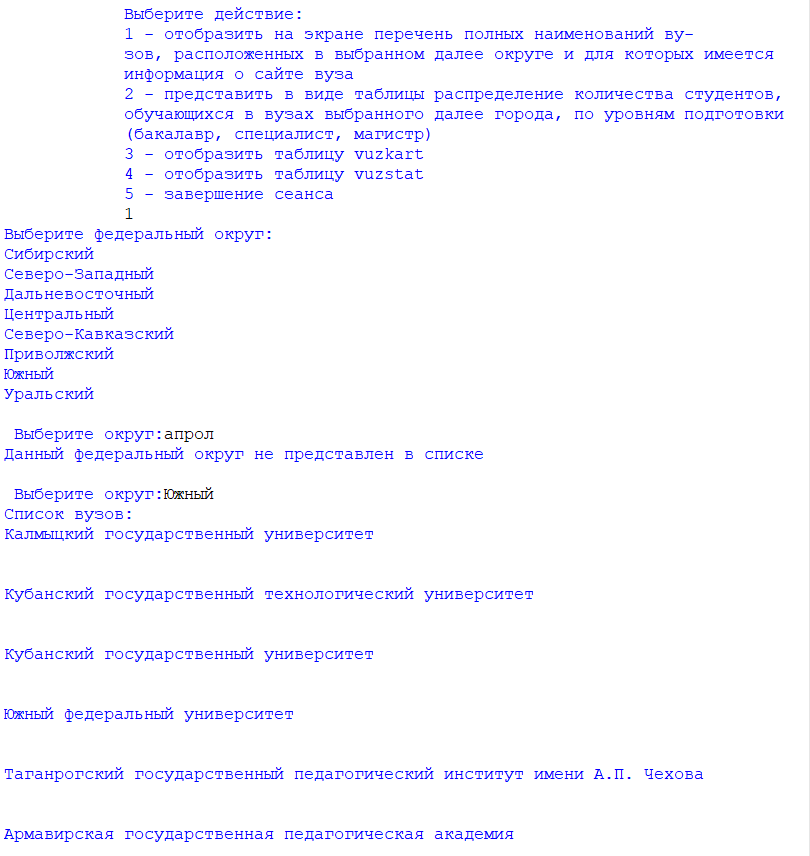
Эта функция выводит таблицу vuzstat.



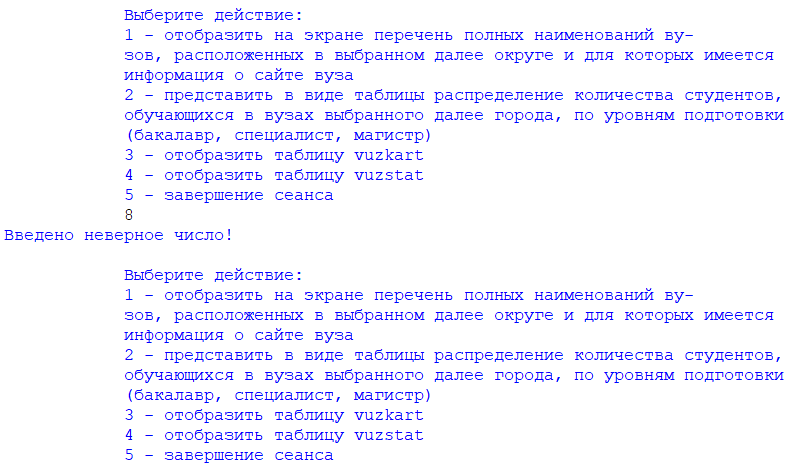
Главная программа. В самом начале проверяется наличие БД VUZ.sqlite. Затем пользователю предлагается меню действий. При выборе «1» вызывается функция vuz\_name(), «2» - students(), «3» - Vuzkart, «4» - Vuzstat, «5» - завершение сеанса. Для третьего пункта я сделала два варианта вывода таблицы. Здесь также есть проверка на ввод некорректного символа.

  
**3.3. Тестирование**

Введем на запрос «Выберите действие:» число 1, вывелся список Федеральных округов, на запрос «Выберите округ:» пишем любой набор символов, тогда программа выводит сообщение «Данный федеральный округ не представлен в списке». Затем вводим ФО из представленного списка и программа выводит список вузов, находящиеся в данном ФО.



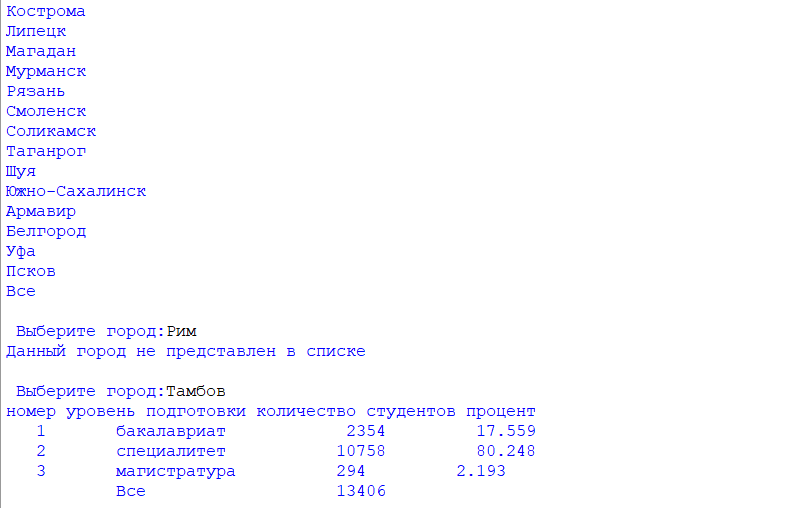
Попробуем ввести на запрос «Выберите действие:» любое другое число, не входящее в промежуток [1;5]. Как мы видим программа пишет «Введено неверное число!» и выводит список действий снова.



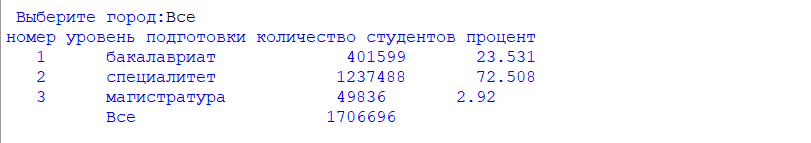
Теперь на запрос «Выберите действие:» выбираем действие 2, выводится список городов, а также строчка «Все». На запрос «Выберите город» снова вводим набор символов или любой город, не представленный в списке, программа выводит сообщение «Данный город не представлен в списке». Теперь вводим название города из списка. Программа выводит табличку из 4 столбцов и 4 строк, в которой представлены данные о количестве студентов на трех программах обучения и их соотношение к общему количеству студентов в процентах в данном городе.







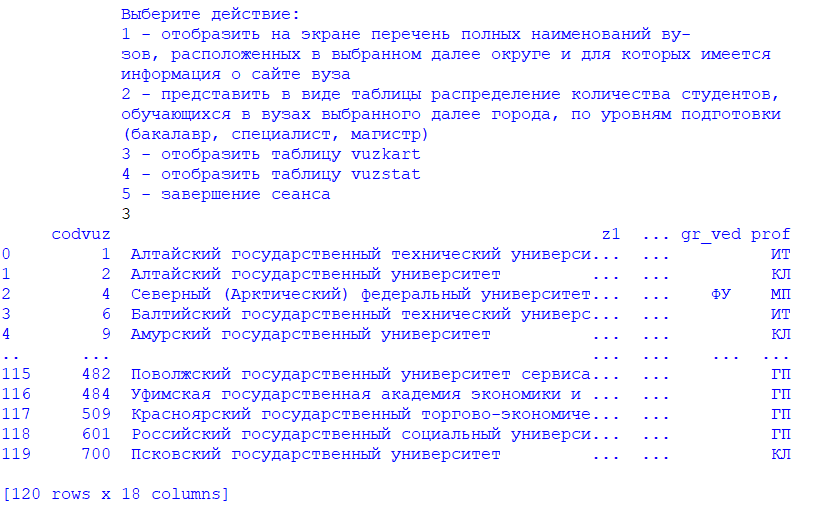
Теперь из представленного списка городов выбираем «Все». Выводится табличка с теми же данными, но уже с информацией по всем городам из списка.



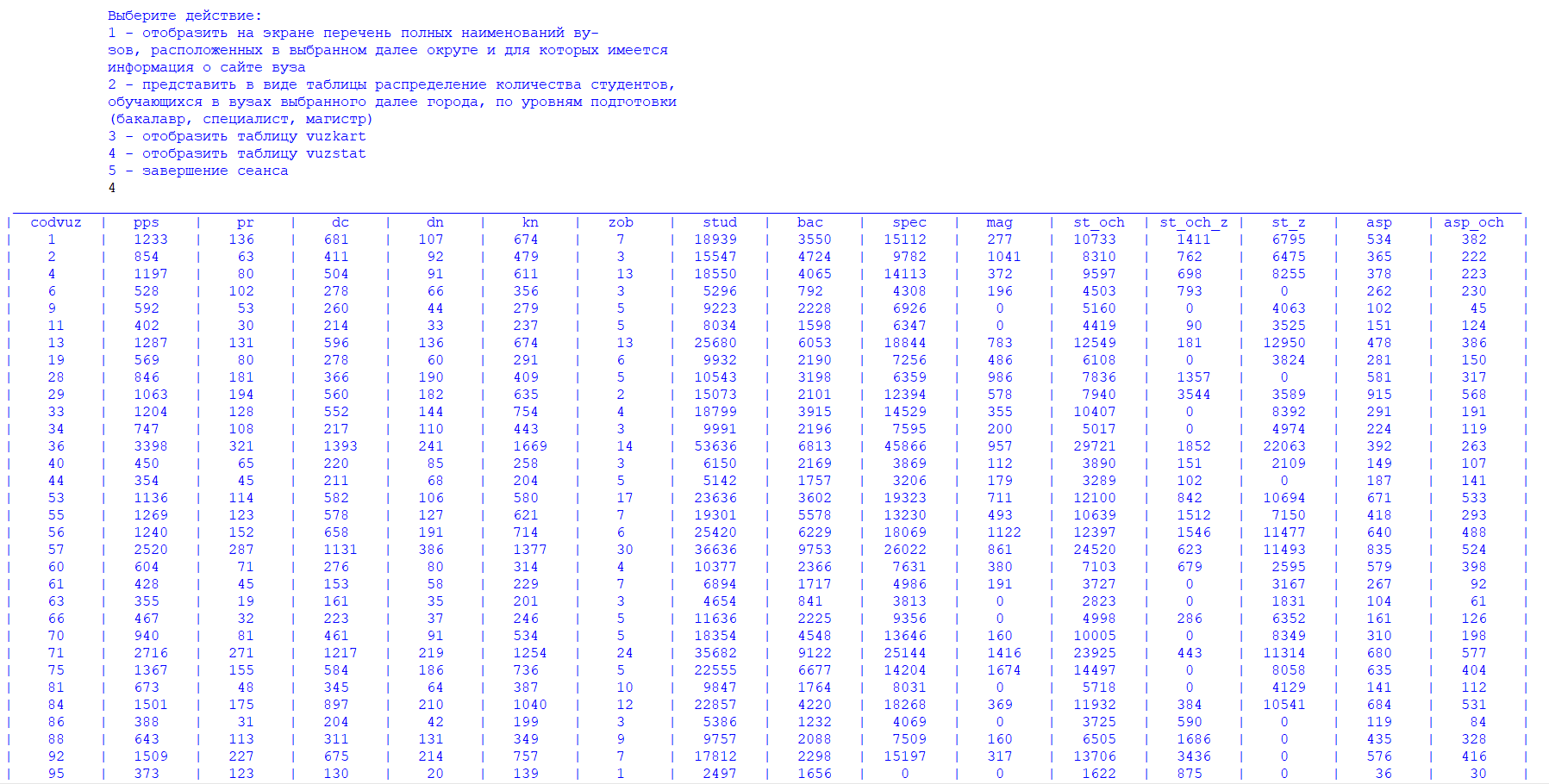
На запрос «Выберите действие:» выбираем цифру 3. Выводится табличка vuzkart с записями, содержащими сведения о вузах России.



Также можно вывести табличку vuzkart из БД, я прописала этот пункт в программе и закомментировала.



На запрос «Выберите действие:» выбираем цифру 4. Выводится таблица vuzstat, содержащая статистические данные по вузам.



При вводе цифры 5 программа завершает свою работу.

