

УДК 004.65

**СОЗДАНИЕ СОБСТВЕННОЙ БАЗЫ ДАННЫХ  
НА PYTHON ДЛЯ НАХОЖДЕНИЯ ИСХОДНОЙ ИНФОРМАЦИИ  
ПО МАРКАМ ПРОВОДОВ  
CREATING YOUR OWN DATABASE ON PYTHON TO FIND ORIGINAL  
INFORMATION ON WIRE BRANDS**

К.А. Марчук

Научный руководитель – Е.М. Гецман, старший преподаватель  
Белорусский национальный технический университет, г. Минск

K. Marchuk

Supervisor – E. Hetsman, Senior Lecturer  
Belarusian national technical university, Minsk

***Аннотация:** создание своей базы данных на Python, для ускорения процесса поиска нужной информации из справочников и книг.*

***Abstract:** creating your own database in Python, to speed up the process of finding the necessary information from reference books and books.*

***Ключевые слова:** СУБД, SQL, базы данных.*

***Keywords:** DBMS, SQL, databases.*

### **Введение**

В действительности, при поиске информации для решения задач многие сталкиваются с необходимостью использовать справочную литературу, что отнимает много времени, а было бы неплохо иметь всю необходимую информацию в виде определенной таблицы, к которой можно было бы своевременно обратиться и получить всю информацию либо же интересующую. Решение данной проблемы состоит в создании отдельной собственной базы данных. В современном мире насчитывается большое количество средств, предназначенных для хранения информации. Одним из наиболее распространенных способов являются базы данных, для работы с которыми используются различные системы управления. Данный способ хранения предполагает, что все данные четко структурированы и занесены в специальные таблицы. Они, в свою очередь, состоят из столбцов-атрибутов определенного типа данных [1]. В данной работе демонстрируются возможности Python при создании собственной базы данных при помощи библиотеки sqlite3.

### **Основная часть**

Электрические кабели и провода – основа энергетической системы. Решения типовых задач предусматривает нахождение данных (технических характеристик) из справочной литературы по конкретной марке провода либо кабеля, что в свою очередь затрачивает много времени. В качестве примера создадим упрощенную таблицу электрических характеристик провода марки АС разных сечений со значением удельного активного и реактивного сопротивления. Данный код служит для заполнения базы данных вручную.

```

import sqlite3
with sqlite3.connect('server.db') as data: #создаем базу
    sql = data.cursor()#обращаемся к экземпляру класса Cursor
    # создаем таблицу провод
    sql.execute("""CREATE TABLE IF NOT EXISTS provod(
    mark TEXT,|
    r0 TEXT,
    x0 TEXT
    )""")
    data.commit()# сохраняем изменение
    # Получаем исходные данные
    mark = input('Марка провода:')
    r0 = input("Сопротивление активное:")
    x0 = input("Сопротивление реактивное:")
    #SQL запрос (получаем марку провода)
    sql.execute(f"SELECT mark FROM provod WHERE mark = '{mark}'")
    if sql.fetchone() is None:
        sql.execute(f"INSERT INTO provod VALUES (?,?,?)", (mark,r0,x0))#добавляем данные, если выполняется условие
        data.commit()
        print("Зарегистрировано")
    else:
        print("Такая запись уже есть")

```

Рисунок 1 – Код для создания своей базы данных проводов для ЛЭП

В результате выполнения программы в директории проекта появилась база данных server.db, содержащая таблицу provod, которая и содержит информацию о электрических характеристиках проводов марки АС разных сечений со значением удельного активного и реактивного сопротивления. (рис.2).

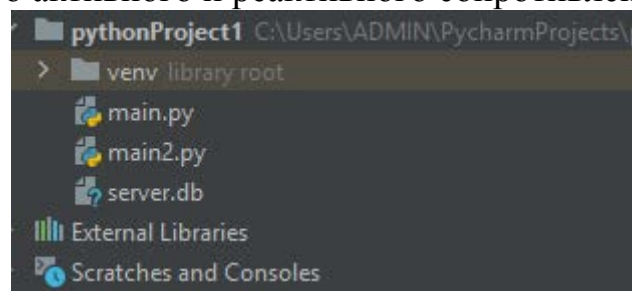


Рисунок 2 – База данных в директории

Для того чтобы считать базу данных следует воспользоваться следующим кодом (рис.3):

```

import sqlite3
with sqlite3.connect('server.db') as data:
    sql = data.cursor()#обращаемся к экземпляру класса Cursor
    for i in sql.execute("SELECT * FROM provod"):# считаем всю базу данных
        print(i)

```

Рисунок 3 – Код для считывания всей базы данных

В результате выполнения кода на (рис.3), получим информацию о электрических характеристиках проводов марки АС, которая и была предварительно занесена (рис.4):

```

('АС - 70/11', '0.46', '0.276')
('АС - 95/16', '0.33', '0.234')

```

Рисунок 4 – Данные из консоли

Полученную информацию можно сохранить в блокнот, Microsoft (MS)

Excel или MS Word и в дальнейшем вывести на печать. Допустим необходимо найти исходную информацию (электрические характеристики) по проводу марки АС 90/11, а база данных может хранить в себе массу информации

и без сортировки или фильтров потребуется больше времени на поиск необходимой информации, нежели со справочником. С помощью кода (рис.5) следует взять с созданной базы данных только конкретную информацию.

```
import sqlite3
with sqlite3.connect('server.db') as data:
    sql = data.cursor()#обращаемся к экземпляру класса Cursor

    mark = input('Марка провода:')

    sql.execute(f"SELECT mark FROM provod WHERE mark = '{mark}'")
    if sql.fetchone() is None:
        print("НЕ зарегистрировано")
    else:
        for i in sql.execute(f"SELECT mark,r0,x0 FROM provod WHERE mark = '{mark}'"):
            print(i)
```

Рисунок 5 – Код для считывания нужной информации

## Заключение

В данной работе были продемонстрированы возможности Python для работы с базами данных при помощи библиотеки sqlite3, при этом их можно использовать не только для хранения данных из справочников и технической документации, но и для хранения информации личного характера, предварительно зашифрованной. Использование информации с базы данных куда удобнее, чем использование информации с блокнота, так как нет нужды вносить параметры в блокнот используя справочник, так как информация уже занесена и ее необходимо лишь достать.

## Литература

1. SQLite // [электронный ресурс] – Режим доступа: <https://lecturesdb.readthedocs.io/databases/sqlite.html>. - Дата доступа: 22.10.2021.
2. SQL – команды [Электронный ресурс] – Режим доступа: <https://zametkinapolyah.ru/zametki-o-mysql/chast-2-2-pervoe-znakomstvo-s-sqlite3-sozdanie-baz-dannyx-sqlite3-rasshirenie-fajlov-baz-dannyx-v-sqlite3.html>. – Дата доступа: 22.10.2021.