

PythonSqlite

May 23, 2022

0.0.1 Sqlite3

Yükləmək üçün

```
[1]: # Unix/Linux
```

```
sudo apt-get install sqlite3 sudo pip3 install sqlite3
```

```
[2]: #Windows
```

```
pip3 install sqlite3
```

gər sisteminizə anaconda platformasını yükləmişsinizsə.pip3 yazmağa ehtiyac yoxdur.Sadəcə olaraq pip install sqlite3 yazmağınız kifayətdir.

```
[ ]:
```

```
[3]: import sqlite3 # kitabxananı çağırırıq
```

Kitabxana,baza yaratmağımız üçün əlverişlidir.Bundan savayı mysql pymongo kimi digər kitabxanalardan da istifadə edə bilərsiniz.Qeyd edimki hər birinin kökü sql sorğu prinspinə dayanır.Və kodlarımızı yazacaq.

Bazanı hazırlamaq üçün iki üsul var.Birincisi ram üzərindən digəri həqiqi disk üzərində.

```
[4]: import sqlite3
```

```
db = sqlite3.connect('database.db')
```

Yuxarıdakı üsulla disk üzərində baza hazırlayacaqsınız.Hal-hazırda mən qovluq yolunu göstərmədiyim üçün birbaşa olduğum cari qovluqda hazırlandı

```
[5]: !dir
```

```
Volume in drive C has no label.
```

```
Volume Serial Number is A4BB-1CFC
```

```
Directory of C:\Users\garay\OneDrive\Documents\GitHub\Python3-AZ\Python-Sqlite
```

```
04/01/2020  09:17 AM    <DIR>
```

```
.
```

```
04/01/2020  09:17 AM    <DIR>
```

```
..
```

```

03/31/2020 10:49 AM <DIR> .ipynb_checkpoints
04/01/2020 09:17 AM <DIR> DATA
04/01/2020 09:17 AM 0 database.db
04/01/2020 09:16 AM 22,341 PythonSqlite.ipynb
                2 File(s)          22,341 bytes
                4 Dir(s)  317,554,704,384 bytes free

```

İkinci metod isə ram üzərində

```

[6]: import sqlite3

db = sqlite3.connect(':memory:')

```

Yuxarıdakı nümunədə isə baza adı yazmağa belə ehtiyac olmadı. Biz amma bu bəhsdə, ilkin metodu istifadə edəcəyik

```

[7]: import sqlite3

db = sqlite3.connect('DATA/database.db')

```

SQL sorğu əmrlərini sqlite tərəfindən hazırlanır. Və çalışdırıla bilməsi üçün execute əmrinə ehtiyacımız var.

```

[8]: import sqlite3

db = sqlite3.connect('DATA/database.db')
cursor = db.cursor()

```

cədvəl hazırlama SQL : CREATE TABLE person ('name','surname','email','password')

```

[9]: import sqlite3

db = sqlite3.connect('DATA/database.db')
cursor = db.cursor()
cursor.execute("CREATE TABLE person_
↳('name','surname','email','password','salary','employment')")

```

```

[9]: <sqlite3.Cursor at 0x21d329adc00>

```

gər təkrar çalışdırsanız yuxarıdakı xətanı alacaqsınız. kodlarımıza bəzi əlavələr edək

Çünki əvvəl biz cədvəli qeyd etdik çalışdırdıq, təkrar çalışdırdıqda bu tip xəta aldıq. Bunu qabaqlamaq üçün

```

[10]: import sqlite3

try:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("CREATE TABLE person_
↳('name','surname','email','password','salary','employment')")

```

```

    print('Database created')
except sqlite3.OperationalError:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("CREATE TABLE IF NOT EXISTS person_
↵('name','surname','email','password','salary','employment')")
    print('Ok')

```

Ok

[11]: *#Məlumatların əlavə edilməsi*

```

[12]: import sqlite3

try:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("CREATE TABLE person_
↵('name','surname','email','password','salary','employment')")
    print('Database created')
except sqlite3.OperationalError:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("CREATE TABLE IF NOT EXISTS person_
↵('name','surname','email','password','salary','employment')")
    print('Ok')
    cursor.execute("INSERT INTO person VALUES_
↵('Rəşad','Qarayev','garayevrashad@hotmail.com','123456',2000,'Python C/
↵C++')")
    cursor.execute("INSERT INTO person VALUES_
↵('Eldar','liyev','example@hotmail.com','asdfg',2000,'Back-end Developer')")
    cursor.execute("INSERT INTO person VALUES_
↵('Vüsalə','liyeva','valiyeva@hotmail.com','v1999',1800,'Managment')")
    cursor.execute("INSERT INTO person VALUES ('Arzu','liyeva','arzu@hotmail.
↵com','v1987',4000,'Team Leader')")
    db.commit() #məlumatları bazaya əlavə edirik

    print('Məlumatlar əlavə olundu')

```

Ok

Məlumatlar əlavə olundu

[13]: *#Bazadakı məlumatları oxuyaq*

```
[14]: import sqlite3

try:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("SELECT name,surname FROM person WHERE surname='liyeva'")
    data = cursor.fetchall()
    print(data)
except sqlite3.OperationalError:
    print('Error')
```

```
[('Vüsələ', 'liyeva'), ('Arzu', 'liyeva')]
```

```
[15]: import sqlite3

try:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("SELECT name,surname FROM person WHERE surname='liyeva'")
    data = cursor.fetchall()
    for i in data:
        print('Ad :',i[0])
        print('Soyad :',i[1])
except sqlite3.OperationalError:
    print('Error')
```

Ad : Vüsələ

Soyad : liyeva

Ad : Arzu

Soyad : liyeva

Bazada liyeva soyadı olan şəxsləri sorğu ilə əldə etdik

```
[16]: #Update - məlumatları yeniləmək
```

Şirkətimizə yeni işçi gəldi və biz bu işçini işdən çıxan işçi ilə əvəzləməliyik. Yenidən bazaya əlavə etmək yerinə, çıxarılan işçinin məlumatlarını yeni işçinin məlumatları ilə əvəzləyək

```
[17]: import sqlite3

try:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("UPDATE person SET
↪name='Kəmalə',surname='Məlikova',email='km11@gmail.com',password='km2000'
↪WHERE name='Arzu'")
```

```

        cursor.execute("SELECT name,surname,email,password FROM person ")
        db.commit()
        data = cursor.fetchall()
        for i in data:
            print(i)
        db.close()
except sqlite3.OperationalError:
    print('Error')

```

```

('Rəşad', 'Qarayev', 'garayevrashad@hotmail.com', '123456')
('Eldar', 'liyev', 'example@hotmail.com', 'asdfg')
('Vüsalə', 'liyeva', 'valiyeva@hotmail.com', 'v1999')
('Kəmalə', 'Məlikova', 'km11@gmail.com', 'km2000')

```

[18]: *#Yeni sütun əlavə etmək*

```

[19]: import sqlite3

try:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("SELECT * from person")
    data = cursor.fetchall()

    for i in data:
        print(i)
    db.close()
except sqlite3.OperationalError:
    print('Error')

```

```

('Rəşad', 'Qarayev', 'garayevrashad@hotmail.com', '123456', 2000, 'Python
C/C++')
('Eldar', 'liyev', 'example@hotmail.com', 'asdfg', 2000, 'Back-end Developer')
('Vüsalə', 'liyeva', 'valiyeva@hotmail.com', 'v1999', 1800, 'Managment')
('Kəmalə', 'Məlikova', 'km11@gmail.com', 'km2000', 4000, 'Team Leader')

```

[20]: *#ORDER BY verilənləri,sorğu argumentinə uyğun sıralayır.Məsələn,idlərə adlara
↪ görə,maş az-çox məsələsinə görə*

```

[21]: import sqlite3

try:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("SELECT DISTINCT * FROM person ORDER BY name DESC") #adlara
↪ uyğun - əlifba sırası ilə

```

```

data = cursor.fetchall()
for i in data:
    print(i)
db.close()
except sqlite3.OperationalError:
    print('Error')

```

```

('Vüsələ', 'liyeva', 'valiyeva@hotmail.com', 'v1999', 1800, 'Managment')
('Rəşad', 'Qarayev', 'garayevrashad@hotmail.com', '123456', 2000, 'Python C/C++')
('Kəmalə', 'Məlikova', 'km11@gmail.com', 'km2000', 4000, 'Team Leader')
('Eldar', 'liyev', 'example@hotmail.com', 'asdfg', 2000, 'Back-end Developer')

```

```

[22]: import sqlite3

try:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("SELECT DISTINCT * FROM person ORDER BY salary DESC") #maaşa_
    ↪görə çoxdan-aza doğru
    data = cursor.fetchall()
    for i in data:
        print(i)
    db.close()
except sqlite3.OperationalError:
    print('Error')

```

```

('Kəmalə', 'Məlikova', 'km11@gmail.com', 'km2000', 4000, 'Team Leader')
('Rəşad', 'Qarayev', 'garayevrashad@hotmail.com', '123456', 2000, 'Python C/C++')
('Eldar', 'liyev', 'example@hotmail.com', 'asdfg', 2000, 'Back-end Developer')
('Vüsələ', 'liyeva', 'valiyeva@hotmail.com', 'v1999', 1800, 'Managment')

```

[23]: *#maaşı 2000 dən aşağı olanları ekrana çap edək*

```

[24]: import sqlite3

try:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("SELECT DISTINCT * FROM person WHERE salary < 2000") #maaşa_
    ↪görə çoxdan-aza doğru
    data = cursor.fetchall()
    for i in data:
        print(i)

    db.close()

```

```
except sqlite3.OperationalError:
    print('Error')
```

('Vüsalə', 'liyeva', 'valiyeva@hotmail.com', 'v1999', 1800, 'Managment')

[25]: *#Aşağı maaşlı işçinin maaşını artıraraq*

[26]: `import sqlite3`

```
try:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("UPDATE person SET salary=2400 WHERE name='Vüsalə'")
    cursor.execute("SELECT * from person")
    data = cursor.fetchall()
    for i in data:
        print(i)
    db.commit()
    db.close()
except sqlite3.OperationalError:
    print('Error')
```

('Rəşad', 'Qarayev', 'garayevrashad@hotmail.com', '123456', 2000, 'Python C/C++')
('Eldar', 'liyev', 'example@hotmail.com', 'asdfg', 2000, 'Back-end Developer')
('Vüsalə', 'liyeva', 'valiyeva@hotmail.com', 'v1999', 2400, 'Managment')
('Kəmalə', 'Məlikova', 'km11@gmail.com', 'km2000', 4000, 'Team Leader')

[27]: *#Şəxsi bazadan kənarlaşdıraraq*

[28]: *#DELETE FROM*

[29]: `import sqlite3`

```
try:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("DELETE FROM person WHERE employment == 'Back-end_↵
↳Developer'")
    cursor.execute("SELECT * from person")
    data = cursor.fetchall()
    for i in data:
        print(i)
    db.commit()
    db.close()
except sqlite3.OperationalError:
    print('Error')
```

```
(('Rəşad', 'Qarayev', 'garayevrashad@hotmail.com', '123456', 2000, 'Python C/C++')
('Vüsalə', 'liyeva', 'valiyeva@hotmail.com', 'v1999', 2400, 'Managment')
('Kəmalə', 'Məlikova', 'km11@gmail.com', 'km2000', 4000, 'Team Leader'))
```

İstənilən məlumata görə şəxsi bazadan kənarlaşdırmaq mümkündür. DELETE FROM person WHERE salary == 4000 qeyd etsəydik, bu səfər maaşa görə şəxsi kənarlaşdıracaqdıq

```
[30]: #Sum hesabı cəm
```

Şirkət daxilində işçilərə verilən maaşın ümumi cəmini əldə edə bilərik.

```
[35]: import sqlite3

try:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("SELECT SUM(salary) FROM person")
    data = cursor.fetchall()
    print(data)
    db.commit()
    db.close()
except sqlite3.OperationalError:
    print('Error')
```

```
[(8400,)]
```

```
[36]: # mean və ya average termini
#İşçilərin ümumi maaşlarının orta qiyətini çıxara bilərik. İşçi maaşları
↳ toplanır daha sonra işçilərin sayına bölünür
```

```
[ ]:
```

```
[39]: import sqlite3

try:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("SELECT AVG(salary) FROM person")
    data = cursor.fetchall()
    print(data)
    db.commit()
    db.close()
except sqlite3.OperationalError:
    print('Error')
```

```
[(2800.0,)]
```

```
[40]: # min və max
```



```
[49]: import sqlite3

try:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("SELECT max(salary) FROM person")
    maximum = cursor.fetchall()
    print(maximum)
    db.close()
except sqlite3.OperationalError:
    print('Error')
```

[(4000,)]

```
[50]: import sqlite3

try:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("SELECT min(salary) FROM person")
    minimum = cursor.fetchall()
    print(minimum)
    db.close()
except sqlite3.OperationalError:
    print('Error')
```

[(2000,)]

```
[51]: #like
```

Bəzi vəziyyətlərdə ad soyad kimi elementləri bilmədiyimiz təqdirdə baş hərfi ilə də elementləri əldə etmək kimi vəziyyətlə rastlaşa bilərik

```
[56]: import sqlite3

try:
    db = sqlite3.connect('DATA/database.db')
    cursor = db.cursor()
    cursor.execute("SELECT * FROM person WHERE name LIKE 'R%'")
    data = cursor.fetchall()
    print(data)
    db.close()
except sqlite3.OperationalError:
    print('Error')
```

[('Rəşad', 'Qarayev', 'garayevrashad@hotmail.com', '123456', 2000, 'Python C/C++')]

```
[57]: #R ilə başlayan elementi sorğuya çəkdik
```

```
[ ]:
```

Obyekt Yönlümlü proqramlaşdırmadan istifadə edərək proqram hazırlayaq

```
[7]: import sqlite3

class Company():
    def __init__(self):
        self.fullname = None
        self.email = None
        self.mobile = None
        self.salary = None
        self.employment = None
        self.db = sqlite3.connect('database.db')
        self.cursor = self.db.cursor()
        self.cursor.execute("CREATE TABLE IF NOT EXISTS person_
↪('fullname','email','mobile','salary','employment')")
        self.db.commit()
        self.db.close()#Təhlükəsizlik səbəbiylə bazanı bağlamağı unutmuruz
    def add(self):
        while True:
            try:
                self.fullname = input('Ad və Soyad(exam. Rəşad Qarayev):')
                self.email = input('Poçt:')
                self.mobile = input('laqə nömrəsi:')
                self.salary = int(input('Maaş:'))
                self.employment = input('Məşğulluq (for exam: python):')
                self.db = sqlite3.connect('database.db')
                self.cursor = self.db.cursor()
                if self.employment == 'Şirkət rəhbəri':
                    self.cursor.execute("INSERT INTO person VALUES_
↪('{}','{}','{}','{}','Şirkət rəhbəri')".format(self.fullname,self.email,self.
↪mobile,self.salary))
                else:
                    self.cursor.execute("INSERT INTO person VALUES_
↪('{}','{}','{}','{}','{}')".format(self.fullname,self.email,self.mobile,self.
↪salary,self.employment))
                self.db.commit()
                self.db.close()#Təhlükəsizlik səbəbiylə bazanı bağlamağı_
↪unutmuruq
                print('Məlumatlar bazaya əlavə olundu')
                break
            except sqlite3.OperationalError:
                print('Error')
```

```

        except ValueError:
            print('Şəxs haqqında yalnız məlumat daxil etmisiniz')

class Database(Company):
    def __init__(self):
        Company.__init__(self)

    def __str__(self):
        self.db = sqlite3.connect('database.db')
        self.cursor = self.db.cursor()
        self.cursor.execute("SELECT fullname,email,mobile,salary,employment_
↪FROM person")
        self.cursor.execute("SELECT DISTINCT * FROM person ORDER BY salary_
↪DESC")
        self.data = self.cursor.fetchall()
        for i in self.data:

            print(i)
        self.db.close()

    def _update(self):
        self.fullname = input('Şəxsin ad və soyadını qeyd edin:')
        self.salary = int(input('Maaşını qeyd edin:'))
        self.db = sqlite3.connect('database.db')
        self.cursor = self.db.cursor()
        self.cursor.execute("UPDATE person SET salary='{}' WHERE fullname='{}'".
↪format(self.salary,self.fullname))
        self.cursor.execute("SELECT DISTINCT * FROM person ORDER BY salary_
↪DESC")
        self.db.commit()
        self.db.close() #Təhlükəsizlik səbəbiylə bazanı bağlamağı unutmuruz
        print('{} şəxsin maaşı yeniləndi'.format(self.fullname))

    def _delete(self):
        try:
            self.params = input('Kənarlaşdıracağınız şəxsin poçt ünvanını qeyd_
↪edin:')
            self.db = sqlite3.connect('database.db')
            self.cursor = self.db.cursor()
            self.cursor.execute("DELETE FROM person WHERE email == '{}'".
↪format(self.params))
            print('{} şəxs,bazadan kənarlaşdırıldı.'.format(self.params))
            self.cursor.execute("SELECT DISTINCT * FROM person ORDER BY salary_
↪DESC")
            self.db.commit()
            self.db.close() #Təhlükəsizlik səbəbiylə bazanı bağlamağı unutmuruz
        except sqlite3.OperationalError:

```

```

        print('Ops')
class Manager(Company):
    def __init__(self):
        Company.__init__(self)

    def __str__(self):
        self.db = sqlite3.connect('database.db')
        self.cursor = self.db.cursor()
        self.cursor.execute("SELECT fullname,email,mobile,salary,employment_
↪FROM person WHERE employment='Şirkət rəhbəri'")

        self.data = self.cursor.fetchall()
        for i in self.data:
            print(i)
        self.db.close()

```

```
[8]: database = Database()
```

```
[9]: manager = Manager()
```

```
[10]: print(dir(database))
```

```

['_class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__',
 '__format__', '__ge__', '__getattr__', '__gt__', '__hash__', '__init__',
 '__init_subclass__', '__le__', '__lt__', '__module__', '__ne__', '__new__',
 '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__',
 '__str__', '__subclasshook__', '__weakref__', '_delete', '_update', 'add',
 'cursor', 'db', 'email', 'employment', 'fullname', 'mobile', 'salary']

```

```
[11]: database.__str__()
```

```
[12]: database.add()
```

```

Ad və Soyad(exam. Rəşad Qarayev):Rəşad Qarayev
Poçt:garayevrashad@hotmail.com
laqə nömrəsi:+994500
Maaş:4000
Məşğulluq (for exam: python):Python Developer
Məlumatlar bazaya əlavə olundu

```

```
[13]: database.__str__()
```

```

('Rəşad Qarayev', 'garayevrashad@hotmail.com', '+994500', '4000', 'Python
Developer')

```

```
[14]: database.add()
```

```
Ad və Soyad(exam. Rəşad Qarayev):Nigar Mahmudova
Poçt:ng@gmail.com
laqə nömrəsi:+994600
Maaş:3800
Məşğulluq (for exam: python):Back-end Developer
Məlumatlar bazaya əlavə olundu
```

```
[15]: database.__str__()
```

```
('Rəşad Qarayev', 'garayevrashad@hotmail.com', '+994500', '4000', 'Python
Developer')
('Nigar Mahmudova', 'ng@gmail.com', '+994600', '3800', 'Back-end Developer')
```

```
[16]: database.add()
database.__str__()
```

```
Ad və Soyad(exam. Rəşad Qarayev):Niyazi liyev
Poçt:na@hotmail.com
laqə nömrəsi:+9945512
Maaş:3300
Məşğulluq (for exam: python):Qrafik Dizayner
Məlumatlar bazaya əlavə olundu
('Rəşad Qarayev', 'garayevrashad@hotmail.com', '+994500', '4000', 'Python
Developer')
('Nigar Mahmudova', 'ng@gmail.com', '+994600', '3800', 'Back-end Developer')
('Niyazi liyev', 'na@hotmail.com', '+9945512', '3300', 'Qrafik Dizayner')
```

```
[17]: manager
```

```
[17]: <__main__.Manager at 0x2dc56e04518>
```

```
[18]: manager.__str__()
```

```
[19]: manager.add()
```

```
Ad və Soyad(exam. Rəşad Qarayev):Vahid ləkberov
Poçt:vahid@gmail.com
laqə nömrəsi:+9947023
Maaş:7800
Məşğulluq (for exam: python):Şirkət rəhbəri
Məlumatlar bazaya əlavə olundu
```

```
[20]: manager.__str__()
```

```
('Vahid ləkberov', 'vahid@gmail.com', '+9947023', '7800', 'Şirkət rəhbəri')
```

```
[21]: database.__str__()
```

```
('Vahid ləkberov', 'vahid@gmail.com', '+9947023', '7800', 'Şirkət rəhbəri')
('Rəşad Qarayev', 'garayevrashad@hotmail.com', '+994500', '4000', 'Python
Developer')
('Nigar Mahmudova', 'ng@gmail.com', '+994600', '3800', 'Back-end Developer')
('Niyazi liyev', 'na@hotmail.com', '+9945512', '3300', 'Qrafik Dizayner')
```

[]: