

SQLite

1. CREATE TABLE /DATABASE	page1	8. UPDATE TABLE	page 5
2. INSERT ONE CUSTOMER	page 2	9. DELETE RECORD	page 5
3. INSERT MANY CUSTOMERS	page 2	10. ORDER RESULTS	page 6
4. SELECT CUSTOMERS	page 3	11. AND / OR	page 6
5. SELECT CUSTOMERS (list)	page 3	12. LIMITING TABE	page 7
6. PRIMARY KEY ID	page 4	13. DELETE TABLE	page 7
7. HOW TO SEARCH	page 4	14-15. NEW FILE/ FUNCTION	page 8

STEP 1 - CREATE TABLE /DATABASE

```
import sqlite3
```

```
conn = sqlite3.connect('customer.db') => create database file
```

first we have to create a cursor

```
c = conn.cursor()
```

create a table

```

c.execute(""" CREATE TABLE customers (
    first_name  text,
    last_name   text,
    email       text
)
""")

```

needs to be capitalize

!!!! 3 quotations marks

```

    datatype
    first_name  text,
    last_name   text,
    email       text
)
""")

```

!!!! 3 quotations marks

#SQLite3 has only five datatypes
NULL (does exist or does not exist)
INTEGER
REAL
TEXT
BLOB

Commit our command

```
conn.commit()
```

Close our connection

```
conn.close()
```

STEP 2 - INSERT ONE CUSTOMER

```
import sqlite3

conn = sqlite3.connect('customer.db')

c = conn.cursor()

# Insert data into our table

c.execute("INSERT INTO customers VALUES ('Adam','Krubá','adamsts@seznam.cz')")
#we inserted a record

conn.commit()

conn.close()
```

STEP 3 – INSERT MANY CUSTOMERS

```
import sqlite3

conn = sqlite3.connect('customer.db')

c = conn.cursor()

#we create a list
many_customers = [
    ('Pepa','Novak', 'novak@email.com'),
    ('Josef','Novak', 'novakjosef@email.com'),
    ('Jiri','Zeman', ' zeman@email.com')
]

!!!!!!!!!!!!
c.executemany("INSERT INTO customers VALUES (?,?,:)“, many_customers )

conn.commit()

conn.close()
```

STEP 4 – SELECT CUSTOMERS

```

import sqlite3

conn = sqlite3.connect('customer.db')

c = conn.cursor()

c.execute("SELECT * FROM customers")
# c.fetchone()      # last item from the table
# c.fetchmany(3)    # how many we want
# c.fetchall()      # all

print(c.fetchall())

conn.commit()

conn.close()

```

STEP 5 – SELECT CUSTOMERS (list)

```

import sqlite3

conn = sqlite3.connect('customer.db')

c = conn.cursor()

c.execute("SELECT * FROM customers")

items = c.fetchall()

for item in tiems:
    print(item)

or

print("NAME" + "\t\tEMAIL")
for item in tiems:
    print(item[0] + ' ' + item[1] + ' ' + item[2])

print(c.fetchall())
conn.commit()
conn.close()

```

STEP 6 – PRIMARY KEY ID

Sqlite3 create another column sort of in the background with a specific primary key a unique ID , it's called row ID

```
import sqlite3

conn = sqlite3.connect('customer.db')

c = conn.cursor()

c.execute("SELECT rowid,      * FROM customers")
                must be a comma !!!!!

items = c.fetchall()

for item in tiems:
    print(item)

print(c.fetchall())

conn.commit()

conn.close()
```

STEP 7 – HOW TO SEARCH

```
import sqlite3

conn = sqlite3.connect('customer.db')

c = conn.cursor()

# if there is a age or number we can use == , >=.....)
c.execute("SELECT * FROM customers WHERE last_name= ' Kruba' ")

or
c.execute("SELECT * FROM customers WHERE email LIKE '%kr%'; ")

items = c.fetchall()

for item in tiems:
    print(item)

conn.commit()
onn.close()
```

STEP 8 – UPDATE TABLE

```
import sqlite3

conn = sqlite3.connect('customer.db')

c = conn.cursor()

c.execute(""" UPDATE customers SET first_name = 'Petr' WHERE last_name = 'Krubá'

""")
```

OR – BETTER SOLUTION

```
c.execute(""" UPDATE customers SET first_name = 'Eda' WHERE rowid =1

""")
```

```
conn.commit()
```

```
c.execute("SELECT rowid, * FROM customers")
```

```
items = c.fetchall()
```

```
for item in items:
    print(item)
```

```
conn.commit()
conn.close()
```

STEP 9 – DELETE RECORD

```
import sqlite3

conn = sqlite3.connect('customer.db')

c = conn.cursor()

c.execute("DELETE from customers WHERE rowid= 6 ")

conn.commit()
```

```
items = c.fetchall()
```

```
for item in tiems:
    print(item)
```

```
conn.commit()
conn.close()
```

STEP 10 – ORDER RESULT

```
import sqlite3

conn = sqlite3.connect('customer.db')

c = conn.cursor()

c.execute("SELECT rowid, FROM customers ORDER BY rowid")
c.execute("SELECT rowid, FROM customers ORDER BY rowid DESC")
c.execute("SELECT rowid, FROM customers ORDER BY last_name")
c.execute("SELECT rowid, FROM customers ORDER BY last_name DESC")

items = c.fetchall()

for item in tiems:
    print(item)

conn.commit()

conn.close()
```

STEP 11 - AND / OR

```
import sqlite3

conn = sqlite3.connect('customer.db')

c = conn.cursor()

c.execute("SELECT rowid, FROM customers WHERE last_name LIKE 'No%' AND rowid = 5 ")
c.execute("SELECT rowid, FROM customers WHERE last_name LIKE 'No%' OR rowid = 5 ")

items = c.fetchall()

for item in tiems:
    print(item)

conn.commit()

conn.close()
```

STEP 12 – LIMITING RESULTS

```
import sqlite3

conn = sqlite3.connect('customer.db')

c = conn.cursor()

c.execute("SELECT rowid, FROM customers LIMIT 2 ")
OR
c.execute("SELECT rowid, FROM customers ORDER BY rowid DESC LIMIT 2 ")

items = c.fetchall()

for item in tiems:
    print(item)

conn.commit()

conn.close()
```

STEP 13 – DELETE TABLE

```
import sqlite3

conn = sqlite3.connect('customer.db')

c = conn.cursor()

c.execute("DROP TABLE customers ")
conn.commit()

c.execute("SELECT rowid, * FROM customers")

items = c.fetchall()

for item in tiems:
    print(item)

conn.commit()

conn.close()
```

STEP 14 - CREATE A FUNCTION**database.py**

```
import sqlite3

def show_all():
    conn = sqlite3.connect('customer.db')
    c = conn.cursor()

    c.execute("SELECT * FROM customers")
    items = c.fetchall()

    for item in items:
        print(item)

    conn.commit()
    conn.close()
```

new_file.py

```
import database
database.show_all()
```

STEP 15 - NEW FILE / FUNCTION**database.py**

```
.....
def add_one(first,last,email):
    conn = sqlite3.connect('customer.db')
    c = conn.cursor()
    c.execute("INSERT INTO customers VALUES (?,?,"?), (first,last,email))
    conn.commit()
    conn.close()
```

new_file.py

```
database.add_one('ales','Krubu','aleskruba@seznam.cz')
```

atc