

Working with Sqlite using Python (Linux)

1. Open UBUNTU terminal
2. Enter in root mode and execute following commands to install sqlite3 database
sudo passwd root
sudo apt-get update
sudo spt-get install sqlite3
sudo apt-get update
3. Open python shell
python
4. Import sqlite3 into python
Import sqlite3

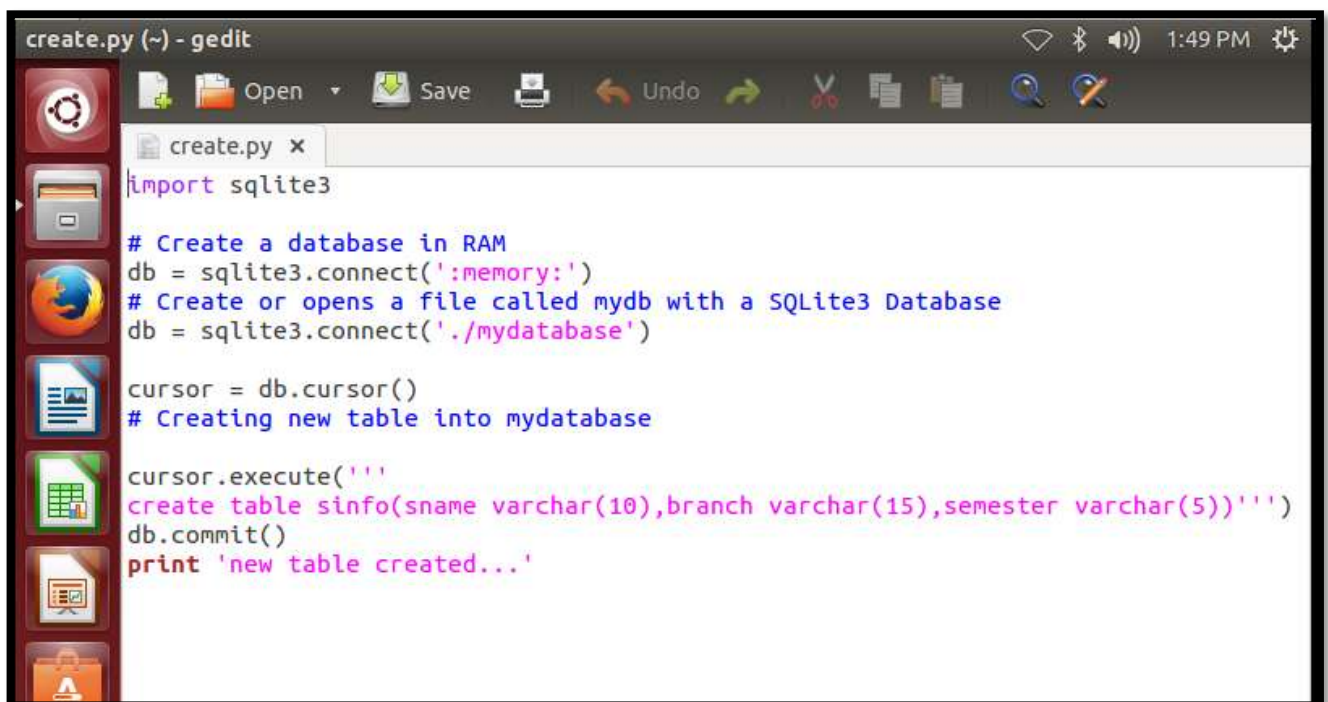
Ready to work with Sqlite3 Database using Python.

How to Create Database in RAM and Database file?

```
import sqlite3
# Create a database in RAM
db = sqlite3.connect(':memory:')
# Create or opens a file called mydatabase with a SQLite3 Database
db = sqlite3.connect('./mydatabase') // mydatabase file will be created HOME directory
```

Database operations on Sqlite3 using Python

1. Creating Database table :sinfo



The screenshot shows a Gedit text editor window titled 'create.py (~) - gedit'. The code inside the editor is as follows:

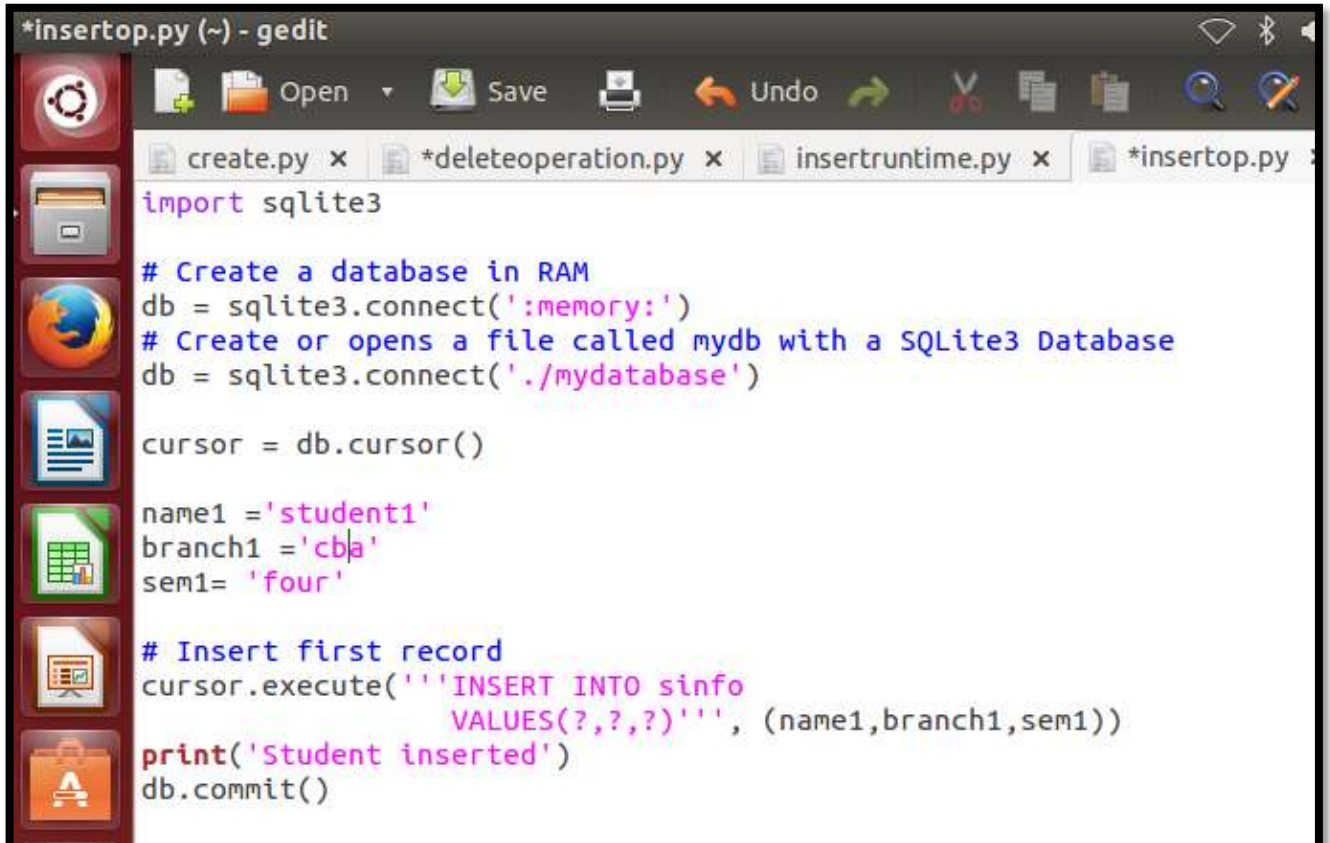
```
import sqlite3

# Create a database in RAM
db = sqlite3.connect(':memory:')
# Create or opens a file called mydb with a SQLite3 Database
db = sqlite3.connect('./mydatabase')

cursor = db.cursor()
# Creating new table into mydatabase

cursor.execute('''
create table sinfo(sname varchar(10),branch varchar(15),semester varchar(5))''')
db.commit()
print 'new table created...'
```

2. Insert a record into sinfo.



```
*insertop.py (~) - gedit
import sqlite3

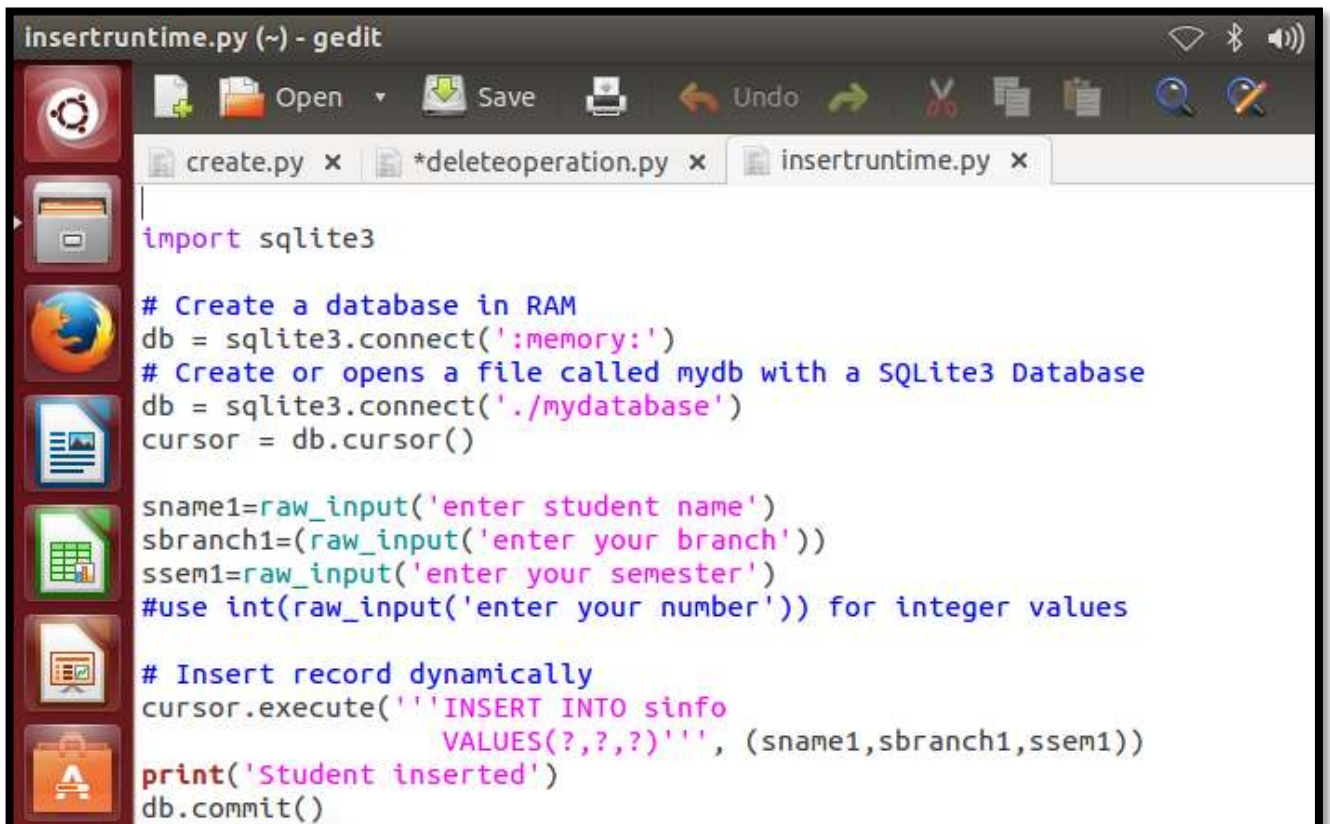
# Create a database in RAM
db = sqlite3.connect(':memory:')
# Create or opens a file called mydb with a SQLite3 Database
db = sqlite3.connect('./mydatabase')

cursor = db.cursor()

name1 = 'student1'
branch1 = 'cba'
sem1 = 'four'

# Insert first record
cursor.execute('INSERT INTO sinfo
              VALUES(?,?,?)', (name1, branch1, sem1))
print('Student inserted')
db.commit()
```

3. Insert record dynamically



```
insertruntime.py (~) - gedit
import sqlite3

# Create a database in RAM
db = sqlite3.connect(':memory:')
# Create or opens a file called mydb with a SQLite3 Database
db = sqlite3.connect('./mydatabase')
cursor = db.cursor()

sname1 = raw_input('enter student name')
sbranch1 = raw_input('enter your branch')
ssem1 = raw_input('enter your semester')
# use int(raw_input('enter your number')) for integer values

# Insert record dynamically
cursor.execute('INSERT INTO sinfo
              VALUES(?,?,?)', (sname1, sbranch1, ssem1))
print('Student inserted')
db.commit()
```

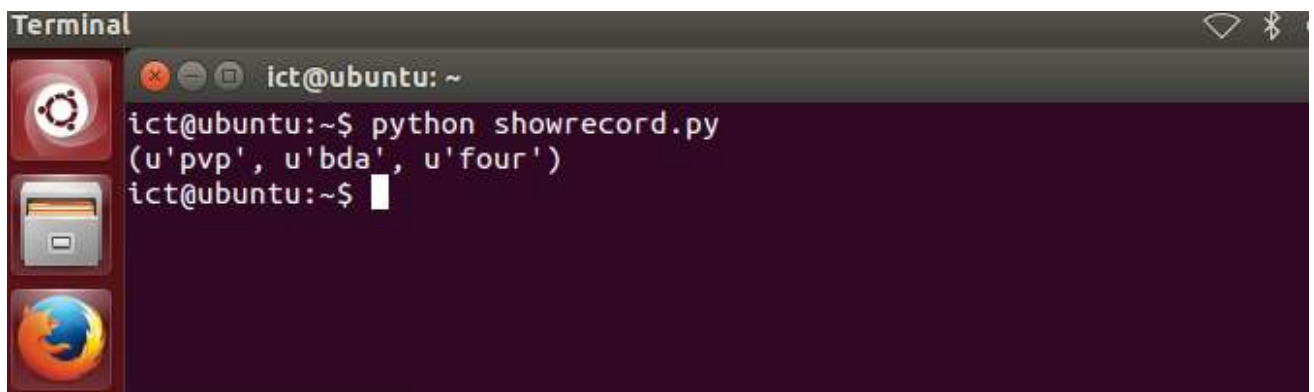
Output of above file



A terminal window titled "Terminal" with the prompt "ict@ubuntu: ~". The user has executed two Python scripts. The first script, "showrecord.py", takes three arguments: (u'pvp', u'bda', u'four'). The second script, "insertruntime.py", prompts the user for a student name (student1), a branch (MA), and a semester (second), then outputs "Student inserted".

```
ict@ubuntu:~$ python showrecord.py
(u'pvp', u'bda', u'four')
ict@ubuntu:~$ python insertruntime.py
enter student namestudent1
enter your branchMA
enter your semestersecond
Student inserted
ict@ubuntu:~$
```

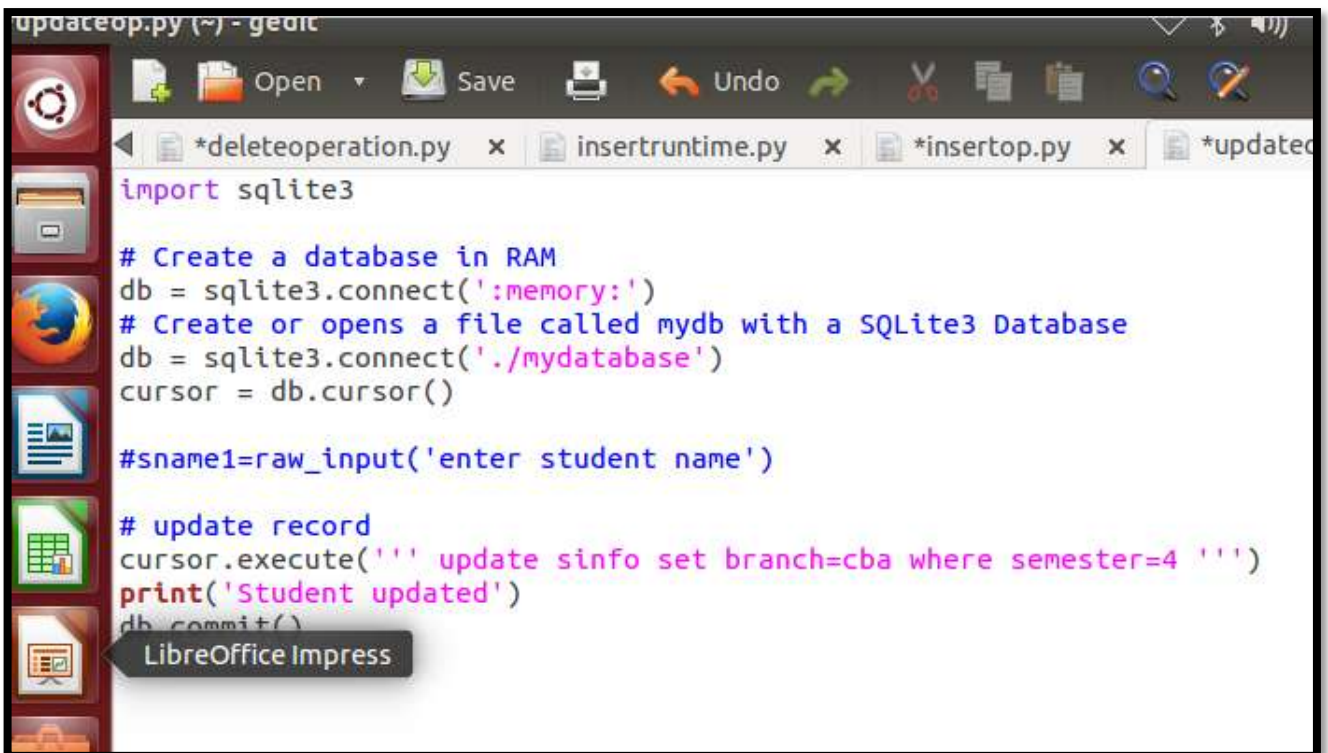
4. Display records of table



A terminal window titled "Terminal" with the prompt "ict@ubuntu: ~". The user has executed the "showrecord.py" script with arguments (u'pvp', u'bda', u'four').

```
ict@ubuntu:~$ python showrecord.py
(u'pvp', u'bda', u'four')
ict@ubuntu:~$
```

5. Update records



A screenshot of a code editor window titled "updateop.py (~) - gedit". The editor shows the following Python code:

```
import sqlite3

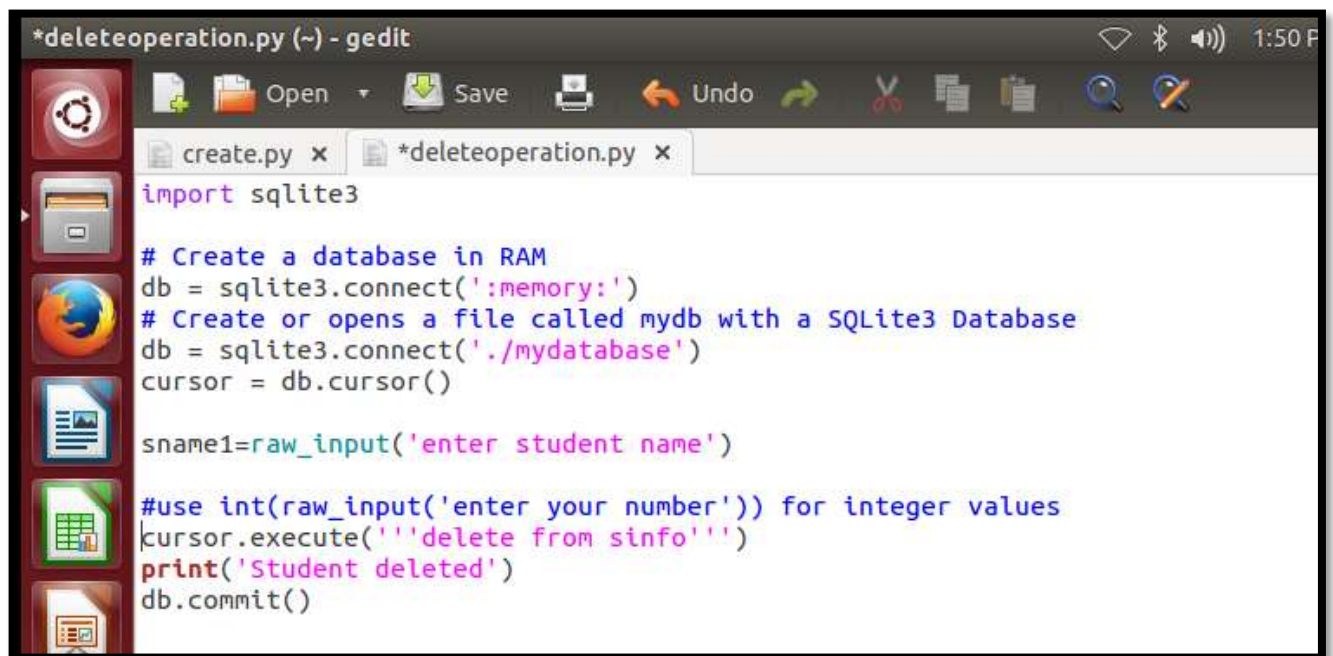
# Create a database in RAM
db = sqlite3.connect(':memory:')
# Create or opens a file called mydb with a SQLite3 Database
db = sqlite3.connect('./mydatabase')
cursor = db.cursor()

#sname1=raw_input('enter student name')

# update record
cursor.execute(''' update sinfo set branch=cba where semester=4 ''')
print('Student updated')
db.commit()
```

The editor's toolbar includes icons for Open, Save, Undo, and other standard editing functions. The file explorer on the left shows icons for the system, home, and applications. A tooltip for "LibreOffice Impress" is visible over the applications icon.

6. Delete record



The image shows a screenshot of a gedit text editor window. The title bar reads '*deleteoperation.py (~) - gedit'. The window contains a Python script that uses the sqlite3 module to delete a student record. The script includes comments in blue, code in black, and user prompts in pink. The code is as follows:

```
import sqlite3

# Create a database in RAM
db = sqlite3.connect(':memory:')
# Create or opens a file called mydb with a SQLite3 Database
db = sqlite3.connect('./mydatabase')
cursor = db.cursor()

sname1=raw_input('enter student name')

#use int(raw_input('enter your number')) for integer values
cursor.execute(''delete from sinfo'')
print('Student deleted')
db.commit()
```