Python SQLite Insert Data Example

To insert data into SQLite table using Python, you must follow the below steps-

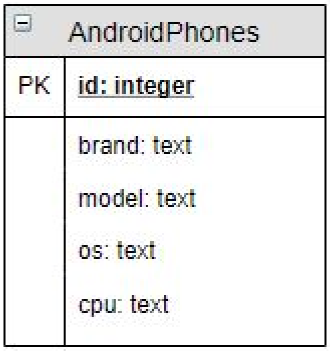
1. Connect to the SQLite database by calling `connect()` function.
2. Then, call `cursor()` function to obtain `Cursor` object.
3. Pass insert statement as an argument to `execute()` function.
4. Call `commit()` function to save the changes to the database. After this, close the connection by calling `close()` function.

Parameter Binding in Python SQLite?

Question marks(?) are used as a placeholder for passing arguments to insert statement. Python supports two types of binding-

1. **By Position**- We specify positional arguments using tuple or list.
2. **By Name**- We specify named arguments using dictionary.

Here, we will use AndroidPhones table, whose fields are shown in the below ER diagram-



AndroidPhones	
PK	id: integer
	brand: text model: text os: text cpu: text

```
import sqlite3

conn = sqlite3.connect('mobiledevices.db')
print('Connected to database successfully.')

cur = conn.cursor()
phone1 = ("Samsung", "Galaxy A7 2018", "Android v8.0 Oreo", "Octa core 
phone2 = ["LG", "G7 Fit", "Android v8.1 Oreo", "Quad core 2.15 GHz"]
phone3 = {"brand":"Motorola", "model":"G6 Plus", "os":"Android v8.0 Ore

cur.execute("insert into androidphones(brand, model, os, cpu) values(?,
cur.execute("insert into androidphones(brand, model, os, cpu) values(?,
cur.execute("insert into androidphones(brand, model, os, cpu) values(:b
print('Records inserted successfully.')

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

When you run the above program, you will get 3 records into androidphones table.

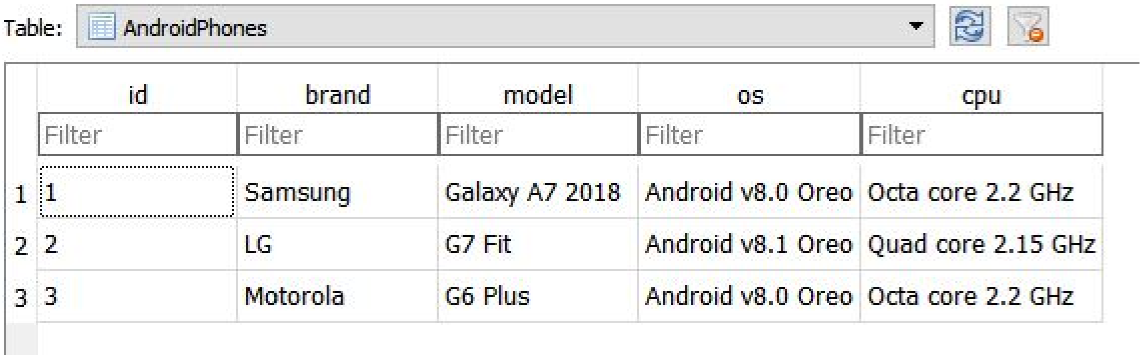


Table: AndroidPhones

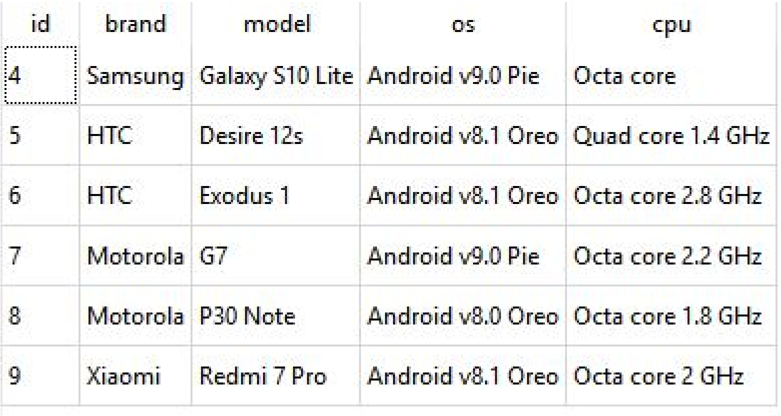
	id	brand	model	os	cpu
1	1	Samsung	Galaxy A7 2018	Android v8.0 Oreo	Octa core 2.2 GHz
2	2	LG	G7 Fit	Android v8.1 Oreo	Quad core 2.15 GHz
3	3	Motorola	G6 Plus	Android v8.0 Oreo	Octa core 2.2 GHz

How to insert more than one record in SQLite table using Python?

To insert more than one record, you have to call executemany() function instead of execute() function. The second argument of this function will take nested tuple, nested list, tuple within a list.

```
import sqlite3

conn = sqlite3.connect('mobiledevices.db')
print('Connected to database successfully.')

cur = conn.cursor()
phone1 = (
    ("Samsung", "Galaxy S10 Lite", "Android v9.0 Pie", "Octa core"),
    ("HTC", "Desire 12s", "Android v8.1 Oreo", "Quad core 1.4 GHz")
)
phone2 = [
    ["HTC", "Exodus 1", "Android v8.1 Oreo", "Octa core 2.8 GHz"],
    ["Motorola", "G7", "Android v9.0 Pie", "Octa core 2.2 GHz"]
]
phone3 = [
    ("Motorola", "P30 Note", "Android v8.0 Oreo", "Octa core 1.8 GHz"),
    ("Xiaomi", "Redmi 7 Pro", "Android v8.1 Oreo", "Octa core 2 GHz")
]

cur.executemany("insert into androidphones(brand, model, os, cpu) value
cur.executemany("insert into androidphones(brand, model, os, cpu) value
cur.executemany("insert into androidphones(brand, model, os, cpu) value
print('Records inserted successfully.')

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

When you run the above program, you will get 6 records into androidphones table.



id	brand	model	os	cpu
4	Samsung	Galaxy S10 Lite	Android v9.0 Pie	Octa core
5	HTC	Desire 12s	Android v8.1 Oreo	Quad core 1.4 GHz
6	HTC	Exodus 1	Android v8.1 Oreo	Octa core 2.8 GHz
7	Motorola	G7	Android v9.0 Pie	Octa core 2.2 GHz
8	Motorola	P30 Note	Android v8.0 Oreo	Octa core 1.8 GHz
9	Xiaomi	Redmi 7 Pro	Android v8.1 Oreo	Octa core 2 GHz