

A continuación se listan un conjunto de scripts escritos en Python que pueden ser útiles para determinadas operaciones con una base de datos SQLite.

Script 1:

```
"""
Convierte una lista de tuplas de filas a una lista de diccionarios con claves
los nombres de las columnas
"""
def makedicts(cursor, query, params=()):
    cursor.execute(query, params)
    colnames = [desc[0] for desc in cursor.description]
    rowdicts = [dict(zip(colnames, row)) for row in cursor.fetchall()]
    return rowdicts
```

Script 2:

```
"""
Carga de una tabla desde un archivo de texto delimitado por comas
Forma de uso: loaddb.py dbfile datafile table
"""
def login(dbfile):
    import sqlite3
    conn = sqlite3.connect(dbfile)
    curs = conn.cursor()
    return conn, curs

def loaddb(curs, table, datafile, conn=None, verbose=True):
    file = open(datafile)                                # x,x,x\nx,x,x\n
    rows = [line.rstrip().split(',') for line in file]    # [[x,x,x], [x,x,x]]
    rows = [str(tuple(rec)) for rec in rows]              # ["(x,x,x)", "(x,x,x)"]
    for recstr in rows:
        curs.execute('insert into ' + table + ' values ' + recstr)
    if conn: conn.commit()
    if verbose: print len(rows), 'filas cargadas'

if __name__ == '__main__':
    import sys
    dbfile = sys.argv[1]
    datafile = sys.argv[2]
    table = sys.argv[3]
    conn, curs = login(dbfile)
    loaddb(curs, table, datafile, conn)
```

Script 3:

"""

Muestra los contenidos de una tabla con un conjunto de tuplas o bien formateada con los nombres de las columnas

Forma de uso: dumpdb.py dbname table [-] (guion=muestra formateada)

"""

```
def showformat(recs, sept=('-' * 40)):
    print len(recs), 'registros'
    print sept
    for rec in recs:
        maxkey = max(len(key) for key in rec)
        for key in rec: # or: \t align
            print '%-*s => %s' % (maxkey, key, rec[key])
        print sept

def dumpdb(cursor, table, format=True):
    if not format:
        cursor.execute('select * from ' + table)
        while True:
            rec = cursor.fetchone()
            if not rec: break
            print rec
    else:
        from makedicts import makedicts
        recs = makedicts(cursor, 'select * from ' + table)
        showformat(recs)

if __name__ == '__main__':
    import sys
    cmdargs = sys.argv[1:]
    if '-' in cmdargs:
        format = True
        cmdargs.remove('-')
    dbname = cmdargs.pop(0)
    table = cmdargs[0]

    from loaddb import login
    conn, curs = login(dbname)
    dumpdb(curs, table, format)
```

Script 4:

```
"""
Elimina y reconstruye los archivos de la base de datos
Forma de uso: makedb.py dbname tablename
"""

import sys
if raw_input('¿Estás seguro?').lower() not in ('s', 'si'):
    sys.exit()

dbname = sys.argv[1]
table = sys.argv[2]

from loaddb import login
conn, curs = login(dbname)
try:
    curs.execute('drop table ' + table)
except:
    print 'La tabla no existe'
conn.commit()
print 'Hecho', dbname, table
```

Script 5:

```
"""
Borra todas las filas de una tabla pero no elimina la tabla o base de datos
Forma de uso: cleardb.py dbname tablename
"""

import sys
if raw_input('¿Estás seguro?').lower() not in ('s', 'si'):
    sys.exit()

dbname = sys.argv[1]
table = sys.argv[2]

from loaddb import login
conn, curs = login(dbname)
curs.execute('delete from ' + table)
conn.commit()
```

Script 6:

"""

Ejecuta una consulta y muestra el resultado formateado en forma de tabla

Forma de uso: querydb.py dbname query

"""

```
import sys
```

```
database = sys.argv[1]
```

```
querystr = sys.argv[2]
```

```
from makedicts import makedicts
```

```
from dumpdb import showformat
```

```
from loaddb import login
```

```
conn, curs = login(database)
```

```
rows = makedicts(curs, querystr)
```

```
showformat(rows)
```