

```
26 """
27 Thierry Bertin-Mahieux (2011) Columbia University
28 tb2332@columbia.edu
29
30 This code demo the use of the artist_similarity.db
31 This is almost the same code as demo_artist_similarity.py
32 in the github repository.
33
34 This is part of the Million Song Dataset project from
35 LabROSA (Columbia University) and The Echo Nest.
36
37 Copyright 2011, Thierry Bertin-Mahieux
38
39 This program is free software: you can redistribute it and/or modify
40 it under the terms of the GNU General Public License as published by
41 the Free Software Foundation, either version 3 of the License, or
42 (at your option) any later version.
43
44 This program is distributed in the hope that it will be useful,
45 but WITHOUT ANY WARRANTY; without even the implied warranty of
46 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
47 GNU General Public License for more details.
48
49 You should have received a copy of the GNU General Public License
50 along with this program. If not, see <http://www.gnu.org/licenses/>.
51 """
52
53 import os
54 import sys
55 import glob
56 import time
57 import datetime
58 import numpy as np
59 try:
60     import sqlite3
61 except ImportError:
62     print 'you need sqlite3 installed to use this program'
63     sys.exit(0)
64
65
66 def encode_string(s):
67     """
68     Simple utility function to make sure a string is proper
69     to be used in a SQLite query
70     (different than postgresql, no N to specify unicode)
71     EXAMPLE:
72     That's my boy! -> 'That''s my boy!'
73     """
74     return "'" + s.replace("'", "'') + "'"
75
76 # PATH TO artist_similarity.db
77 # CHANGE THIS TO YOUR LOCAL CONFIGURATION
78 # IT SHOULD BE IN THE ADDITIONAL FILES
79 # (you can use 'subset_artist_similarity.db')
80 dbfile = '/home/thierry/Columbia/MSongsDB/Tasks_Demos/SQLite/
81         artist_similarity.db'
82
83 # connect to the SQLite database
84 conn = sqlite3.connect(dbfile)
```

```
84
85 # from that connection, get a cursor to do queries
86 # NOTE: we could query directly from the connection object
87 c = conn.cursor()
88
89 print '***** GENERAL SQLITE DEMO *****'
      ***** GENERAL SQLITE DEMO *****
66
67 # list all tables in that dataset
68 # note that sqlite does the actual job when we call fetchall() or fetchone()
69 q = "SELECT name FROM sqlite_master WHERE type='table' ORDER BY name"
70 res = c.execute(q)
71 print "* tables contained in that SQLite file/database (there should be 3):"
      * tables contained in that SQLite file/database (there should be 3):
72 print res.fetchall()
      [(u'artists',), (u'similarity',)]
73
74 # list all indices
75 q = "SELECT name FROM sqlite_master WHERE type='index' ORDER BY name"
76 res = c.execute(q)
77 print '* indices in the database to make reads faster:'
      * indices in the database to make reads faster:
78 print res.fetchall()
      [(u'idx_sim_target',), (u'idx_target_sim',), (u'sqlite_autoindex_artists_1',)]
80
81 print '***** ARTISTS TABLE DEMO *****'
      ***** ARTISTS TABLE DEMO *****
81
82 # list all artist ID
83 q = "SELECT artist_id FROM artists"
84 res = c.execute(q)
85 print "* number of artist Echo Nest ID in 'artists' table:"
      * number of artist Echo Nest ID in 'artists' table:
86 print len(res.fetchall())
      44745
88
89 print '***** ARTIST SIMILARITY DEMO *****'
      ***** ARTIST SIMILARITY DEMO *****
89
90 # get a random similarity relationship
91 q = "SELECT target,similar FROM similarity LIMIT 1"
92 res = c.execute(q)
93 a,s = res.fetchone()
94 print '* one random similarity relationship (A->B means B similar to A):'
      * one random similarity relationship (A->B means B similar to A):
95 print a, '->', s
      AR002UA1187B9A637D -> ARQDOR81187FB3B06C
```

```
96
97 # count number of similar artist to a in previous call
98 q = "SELECT Count(similar) FROM similarity WHERE target="+encode_string(a)
99 res = c.execute(q)
100 print '* artist',a,'has that many similar artists in the dataset:'
```

* artist AR002UA1187B9A637D has that many similar artists in the dataset:

```
101 print res.fetchone()[0]
```

47

```
102
103 # count number of artist s (c queries up) is similar to
104 q = "SELECT Count(target) FROM similarity WHERE similar="+encode_string(s)
105 res = c.execute(q)
106 print '* artist',s,'is similar to that many artists in the dataset:'
```

* artist ARQDOR81187FB3B06C is similar to that many artists in the dataset
:

```
107 print res.fetchone()[0]
```

154

```
108
109 # DONE
110 # close cursor and connection
111 # (if for some reason you added stuff to the db or alter
112 # a table, you need to also do a conn.commit())
113 c.close()
114 conn.close()
```