Musical Track Database

upload your SQLite3 database:

(.sqlite)

See the example code for the use of the connect() statement.

This application will read an iTunes export file in XML

and produce a properly normalized database with this structure:

CREATE TABLE Artist (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

name TEXT UNIQUE

);

CREATE TABLE Genre (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

name TEXT UNIQUE

);

CREATE TABLE Album (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

artist\_id INTEGER,

title TEXT UNIQUE

);

CREATE TABLE Track (

id INTEGER NOT NULL PRIMARY KEY

AUTOINCREMENT UNIQUE,

title TEXT UNIQUE,

album\_id INTEGER,

genre\_id INTEGER,

len INTEGER, rating INTEGER, count INTEGER

);

If you run the program multiple times in testing or with different files,

make sure to empty out the data before each run.

You can use this code as a starting point for your application: http://www.pythonlearn.com/code/tracks.zip.

The ZIP file contains the Library.xml file to be used for this assignment.

To grade this assignment, the program will run a query like this

on your uploaded database and look for the data it expects to see:

SELECT Track.title, Artist.name, Album.title, Genre.name

FROM Track JOIN Genre JOIN Album JOIN Artist

ON Track.genre\_id = Genre.ID and Track.album\_id = Album.id

AND Album.artist\_id = Artist.id

ORDER BY Artist.name LIMIT 3

The expected result of this query on your database is:

Track Artist Album Genre

Chase the Ace AC/DC Who Made Who Rock

D.T. AC/DC Who Made Who Rock

For Those About To Rock (We Salute You) AC/DC Who Made Who Rock