SCHOOL OF SOFTWARE ENGINEERING OF USTC

FIRST SEMESTER, 2012–2013
Campus: Suzhou

AGILE SOFTWARE DEVELOPMENT

Object-Relational Mapping

(Time allowed: TWO hours)

The objective of this experiment class is to practice the Object-Relational mapping lecture. To do so, we will implement the database for a collection of music files. We will use Python and SQLAlchemy. This work will be used for the next experiment class : making a web-based music management tool.

Note that on Linux, installing the standard package for SQLAlchemy of your distribution, is the best approach. Documentation for SQLAlchemy is available online and is very complete.

1. Create the file *database.py*, like in the Object-Relational mapping lecture. The music database should be stored in a file named *music.db*.

2. Create the file *model.py*, that will define all the objects for our music database. Defines the *Band* object (a group of people who creates music), mapping for the *band* table. *Band* should have the following fields:

- Name
- First year active
- Last year active
- Place of origin

What type for the fields is up to you.

3. Create a file *db-init.py*. This script will fill the database with bands in a text file *bands.txt*. Take the following bands

- Boards of Canada, 1986 - present, Edinburgh, Scotland
- Bibio, 2005 - present, West Midlands, England
- Hexstatic, 1997 - present, England

You do not have to know them, they are just taken as examples.

4. Create the *Album* object, associated to the *album* table. One album can be belong to one and only one *Band*. *Album* should have the following fields:

- Name
- Year of publication
- Label

Create a file *albums.txt*, containing the definitions of all the albums for the bands in *bands.txt*. Update *db-init.py* so that those albums are added to the database. You can use Wikipedia to find the albums of the bands of the previous examples.

CONTINUED

5. Write a request, with SQLAlchemy, that returns all the bands which produced an album with the *Warp* label.

6. Create the *Track* object, associated to the *track* table. One track can be belong to one and only one *Album*. *Track* should have the following fields:

- Name
- Duration

Modify *albums.txt* and *db-init.py* so that for each album we got their associated tracks.

7. Write a request, with SQLAlchemy, that returns all the band that wrote a specific track (For instance, which band wrote the track *Roygbiv* ?).

8. Create a *Genre* object. A *genre* is type of music. Usually, a band does music that might belong to several genre.

- Boards of Canada : electronic, ambient, psychedelic, experimental
- Bibio : pop, experimental, folk
- Hexstatic : electronic, acid-jazz

How are you going to modelize the relationship between *Band* and *Genre* ? Update *db-init.py* and *bands.txt* to include the information about the genre.