Lab 4

Table of Contents

Overview

So far in this course you have been introduced to various topics, including:

- Sqlite3 + Programming editor for <.sql> scripts (Notepad++/TextMate).
- Creation of tables on the command line and in scripts
- How to drop tables
- How to insert values into tables
- How to constrain in various ways data insertions, using keys (foreign), references etc.

Furthermore, in the lecture we looked at an example scenario, and some implementation detail (hotel booking scenario etc). Make sure you have watched the lecture before doing this lab… some of the concepts explained there are relevant to understanding what follows and the scenario explained there is also relevant, as is the SQL code.

So, in this lab, the challenge is to create a set of scripts that are called from a top-level script file, which is called, and the database is created when control is sent back to the sqlite3 shell. Logic inside each script creates appropriate tables when called, inserts test data etc. Once the data is loaded, you can call some commands on the database. Note, while you fill follow a simple spec, below, the data and the commands you use are up to you… they are nothing to do, really, with the design of the database.

Implement your own script-based system

Implement your own SQL for the architecture, below. As Feynman once said…don't be "put off by the word 'architecture'; it's just a big word for how we arrange things". The architecture below depicts how certain entities are arranged in a 'Dental Scenario':

Scenario. A database is required to store data on a dental practice that has numerous branches at different locations. The architecture below is a small part of a larger architecture, and this is the part you need to implement as a script-based system. Notice, there are two kinds of people… but you ***do not*** *need to implement any inheritance…just keep it simple. The kinds of people are* ***Professional*** *and* ***Patient****. When a patient registers with a given practice they can register with more than one Professional, and a Professional will have visits from more than one Patient… in other words this is a many-to-many relationaship, which should be realised with a intermediate link/bridge table (****PatientRegistration****). Additionally, each professional is located at one branch* ***Location*** *of the business.*

The ERD diagram is accompanied by a set of schema, one for each entity (Professional, PatientRegistration, Patient, Location). You should think about the primary keys and the foreign keys implied by the design…and remember these are about implementing common-sense constraints on the data when it gets input to the database.

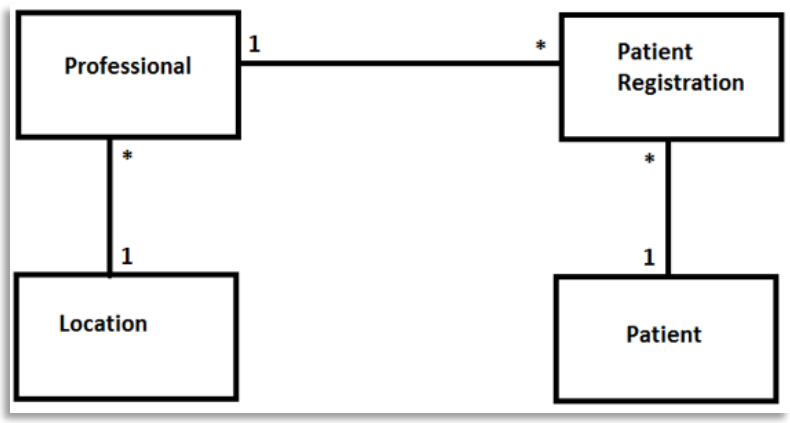

Figure 1: ERD for dental scenario.

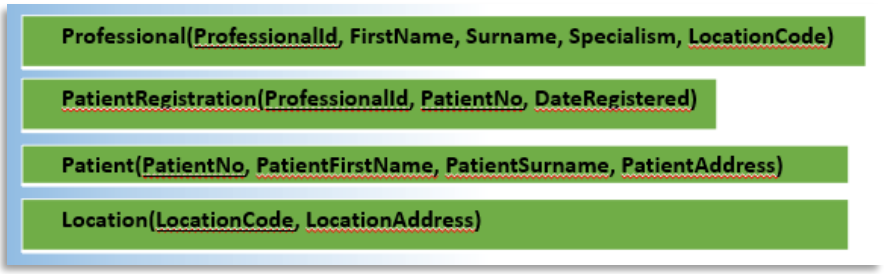

Figure 2: four schema for the four entities.

Suggested work-flow etc

You have already acquired the knowledge to do this task from the topics presented in this course thus far, but just some tips/reminders.

- Arrange your environment (sqlite3, SQL editor) so you are working efficiently. There are videos available on this, as previously mentioned.
- Think about what scripts you will create.
- Create them all and make sure they are getting called from a top-level file.
- Use the dot commands in your scripts and call echo function to indicate which script you are in.
- Develop your SQL implementation incrementally and remember to think about the dependencies and the script-call order, given the architecture of the problem and the architecture of your solution.
- Finally, as the Physicists Richard Feynman once wrote… “Don’t be put off by the word ‘architecture’; it’s just a big word for how we arrange things” (Feynman, 1996, page 4), but the solution you create is yours so try and arrange it with care, but without worrying about making mistakes or asking for help.

Reference

Feynman, R. P (1996) Then Feynman Lectures on Computation. *Edited by Hey, A. and Allen, R*. Addison-Wesley. New York.