Python Course

Month 2 / Week 1 / Day 1

Setup

We will download a new sample database called **sakila** from this GitHub account. We'll be working with this database for the next few days.

1. Download "sakila.db" (Click on it at the link above; then click the 'Download' button on the next page).
2. Open Sqlite Browser, click 'Open Database', and browse to select the sakila database.
3. Here is a diagram of the tables in the server. Take a look at it and learn about the tables, their columns, and the relationships between the different tables.

Exercise 1
We will use the newly installed sakila database.

1. Get a list of all film languages
2. Get a list of all films joined with their languages – select only the film title, description, and language name. Try your query with different joins:
   1. Get all films, even if they don't have languages
   2. Get all languages, even if there are no films in those languages. Which languages are these?
3. You are going to babysit your cousin, and you want to find a few movies that he can watch with you.
   1. Find out how many films there are for each rating
   2. Get a list of all the movies that have a rating of G or PG-13
      1. Filter this list further: look for only movies that are under 2 hours long, and whose rental price (rental_rate) is under 3.00. Sort the list alphabetically.
   3. Find a customer in the customer table, and change his/her details to your details, using SQL UPDATE.
   4. Now find the customer's address, and use UPDATE to change it to an address of your own (or make one up).

Exercise 2

1. Create a new table called customer_review, to contain data about film reviews that customers will make. It should have the following columns:
   1. review_id – a primary key, non null, auto-increment
   2. film_id – references the film table. The film that is being reviewed.
   3. language_id – references the language table. What language the review is in.
   4. title – the title of the review
   5. score – the rating of the review (1-10)
   6. review_text – the text of the review. No limit on the length.
   7. last_update – when the review was last updated.
2. Add 3 movie reviews. Make sure you link them to valid objects in the other tables.
3. Use SQL to delete 2 of the reviews by ID.

Exercise 3
1. Find out how many rentals are still outstanding.
2. Mark the 30 most expensive movies which are outstanding (ie. have not been returned to the store yet) as returned.