

COMP 310 — DB+Python Project 2 — Gym Generator

We've designed a database for our fictional climbing gym application, but now we need to test it out by filling it with real, or nearly real, data. While lots of real data would be great, we don't have any on hand. Instead, we'll use python to generate some fictional data.

Data Requirements

1. 2-3 gyms. You pick the names and the id for the database.
2. For each gym:
 - (a) 1-4 walls. Name them with random combinations of North, South, East, and West joined with Vert, Slab, or Overhang. For example, you could have two walls: North Vert and North Slab.
 - (b) For each wall, flip a coin to decide if it houses one or two lanes.
 - (c) 3-5 sets of 25-50 holds. A set of holds will all have the same color and manufacturer. Colors can be chosen at random from a set of your choosing. Pick at random one of the following manufacturers: Escape, Soll, Atomik, EuroHolds, or Metolious. The holds in a set should be numbered sequentially (i.e. Red 1 - Red 40), but don't need to start at 1. Choose the type of each hold at random.
 - (d) 2-3 active routes. Select walls+lanes at random from your gym. As these are active climbs, you should not select a wall+lane more than once for this. For set dates, come up with a short list of dates and choose from that list at random (with replacement, you can set more than one route on a day). Names and setter names can also be as creative as you'd like. Holds will be discussed below.
 - (e) 3-4 historical routes. Generate these like active routes but given them an end date that is a few weeks after the selected set date.
 - (f) For each route, you should select, at random, from the gym's set of holds. Then select 70-90 percent of the holds from that set. Finally, shuffle your selected subset of the holds to determine their placement. Orientations can be a random int from 1-12.

Programming Specifications

The only hard specification is that your program should be able to generate SQL files that can be used to insert your data into your database. Lots of files is fine. One franken-file is OK. Whatever works for you. Beyond that, you are highly-encouraged to make good use of all the tools Python has to offer. In particular, you might want to explore the following python modules:

- itertools
- random
- datetime

Finally, and perhaps surprisingly, *AI is fair-game on this assignment*. VS-Code has Github Co-Pilot integration and you can always ask Chat-GPT and the like for bits of python.