Homework 8 | CS 61A Spring 2020

λ **inst.eecs.berkeley.edu**/~cs61a/sp20/hw/hw08

Homework 8:
hw08.zip

Due by 11:59pm on Friday, April 17

Instructions

Download hw08.zip.

Submission: When you are done, submit with `python3 ok --submit`. You may submit more than once before the deadline; only the final submission will be scored. Check that you have successfully submitted your code on okpy.org. See Lab 0 for more instructions on submitting assignments.

Using Ok: If you have any questions about using Ok, please refer to this guide.

Readings: You might find the following references useful:

Section 4.3 - Declarative Programming

Grading: Homework is graded based on correctness. Each incorrect problem will decrease the total score by one point. There is a homework recovery policy as stated in the syllabus. **This homework is out of 2 points.**

Usage

First, check that a file named `sqlite_shell.py` exists alongside the assignment files. If you don't see it, or if you encounter problems with it, scroll down to the Troubleshooting section to see how to download an official precompiled SQLite binary before proceeding.

You can start an interactive SQLite session in your Terminal or Git Bash with the following command:

```
python3 sqlite_shell.py
```

While the interpreter is running, you can type `.help` to see some of the commands you can run.

To exit out of the SQLite interpreter, type `.exit` or `.quit` or press `Ctrl-C`. Remember that if you see `...>` after pressing enter, you probably forgot a `;`.

You can also run all the statements in a `.sql` file by doing the following:

1. Runs your code and then exits SQLite immediately afterwards.

```
python3 sqlite_shell.py < lab13.sql
```

2. Runs your code and then opens an interactive SQLite session, which is similar to running Python code with the interactive `-i` flag.

```
python3 sqlite_shell.py --init lab13.sql
```

To complete this homework assignment, you will need to use SQLite version 3.8.3 or greater.

To check your progress, you can run `sqlite3` directly by running:

```
python3 sqlite_shell.py --init hw08.sql
```

You should also check your work using `ok`:

```
python3 ok
```

Questions

Dog Data

In each question below, you will define a new table based on the following tables.

```
CREATE TABLE parents AS
  SELECT "abraham" AS parent, "barack" AS child UNION
  SELECT "abraham"          , "clinton"        UNION
  SELECT "delano"           , "herbert"        UNION
  SELECT "fillmore"         , "abraham"        UNION
  SELECT "fillmore"         , "delano"         UNION
  SELECT "fillmore"         , "grover"         UNION
  SELECT "eisenhower"       , "fillmore";

CREATE TABLE dogs AS
  SELECT "abraham" AS name, "long" AS fur, 26 AS height UNION
  SELECT "barack"         , "short"      , 52           UNION
  SELECT "clinton"        , "long"       , 47           UNION
  SELECT "delano"         , "long"       , 46           UNION
  SELECT "eisenhower"     , "short"      , 35           UNION
  SELECT "fillmore"       , "curly"      , 32           UNION
  SELECT "grover"         , "short"      , 28           UNION
  SELECT "herbert"        , "curly"      , 31;

CREATE TABLE sizes AS
  SELECT "toy" AS size, 24 AS min, 28 AS max UNION
  SELECT "mini"       , 28       , 35        UNION
  SELECT "medium"     , 35       , 45        UNION
  SELECT "standard"   , 45       , 60;
```

Your tables should still perform correctly even if the values in these tables change. For example, if you are asked to list all dogs with a name that starts with h, you should write:

```
SELECT name FROM dogs WHERE "h" <= name AND name < "i";
```

Instead of assuming that the dogs table has only the data above and writing

```
SELECT "herbert";
```

The former query would still be correct if the name grover were changed to hoover or a row was added with the name harry.

Q1: Size of Dogs

The Fédération Cynologique Internationale classifies a standard poodle as over 45 cm and up to 60 cm. The sizes table describes this and other such classifications, where a dog must be over the min and less than or equal to the max in height to qualify as a size. Create a size_of_dogs table with two columns, one for each dog's name and another for its size.

```
-- The size of each dog
CREATE TABLE size_of_dogs AS
  SELECT "REPLACE THIS LINE WITH YOUR SOLUTION";
```

The output should look like the following:

```
sqlite> select * from size_of_dogs;
abraham|toy
barack|standard
clinton|standard
delano|standard
eisenhower|mini
fillmore|mini
grover|toy
herbert|mini
```

Use Ok to test your code:

```
python3 ok -q size_of_dogs
```

Watch Video At: https://youtu.be/d9EPid-MJes
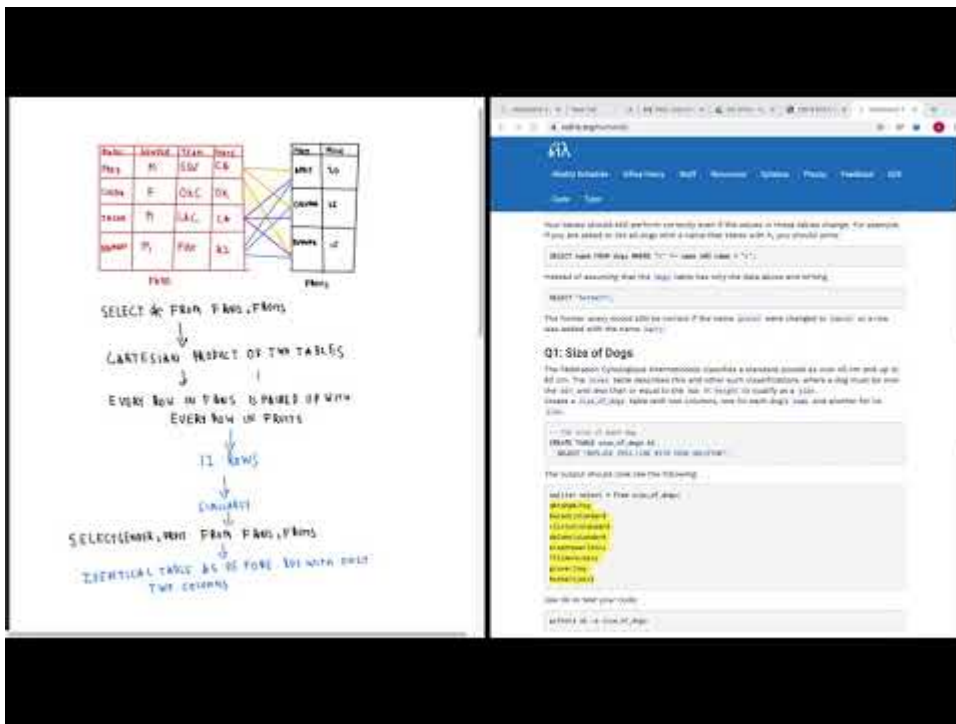
Submit

Make sure to submit this assignment by running:

```
python3 ok --submit
```

Troubleshooting/Advanced SQLite

Troubleshooting

Python already comes with a built-in SQLite database engine to process SQL. However, it doesn't come with a "shell" to let you interact with it from the terminal. Because of this, until now, you have been using a simplified SQLite shell written by us. However, you may find the shell is old, buggy, or lacking in features. In that case, you may want to download and use the official SQLite executable.

If running `python3 sqlite_shell.py` didn't work, you can download a precompiled sqlite directly by following the following instructions and then use `sqlite3` and `./sqlite3` instead of `python3 sqlite_shell.py` based on which is specified for your platform.