Homework 10: SQL **hw10.zip (hw10.zip)**

Due by 11:59pm on Thursday, December 1

Instructions

Download hw10.zip (hw10.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 (https://okpy.org/). See Lab 0 (/lab/lab00#submitting-the-assignment) for more instructions on submitting assignments.

Using Ok: If you have any questions about using Ok, please refer to this guide. (/articles/using-ok)

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: (Here we're using the `lab11.sql` file as an example.)

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

   ```
   python3 sqlite_shell.py < lab11.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 lab11.sql
```

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

```
python3 sqlite_shell.py --init hw10.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: By Parent Height

Create a table `by_parent_height` that has a column of the names of all dogs that have a `parent`, ordered by the height of the parent dog from tallest parent to shortest parent.

```
-- All dogs with parents ordered by decreasing height of their parent
CREATE TABLE by_parent_height AS
  SELECT "REPLACE THIS LINE WITH YOUR SOLUTION";
```

For example, `fillmore` has a parent `eisenhower` with height 35, and so should appear before `grover` who has a parent `fillmore` with height 32. The names of dogs with parents of the same height should appear together in any order. For example, `barack` and `clinton` should both appear at the end, but either one can come before the other.

```
sqlite> select * from by_parent_height;
herbert
fillmore
abraham
delano
grover
barack
clinton
```

Use Ok to test your code:

```
python3 ok -q by_parent_height
```

Hint Video

Q2: 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
```

Hint Video

Q3: Sentences

There are two pairs of siblings that have the same size. Create a table that contains a row with a string for each of these pairs. Each string should be a sentence describing the siblings by their size.

```
-- Filling out this helper table is optional
CREATE TABLE siblings AS
  SELECT "REPLACE THIS LINE WITH YOUR SOLUTION";

-- Sentences about siblings that are the same size
CREATE TABLE sentences AS
  SELECT "REPLACE THIS LINE WITH YOUR SOLUTION";
```

Each sibling pair should appear only once in the output, and siblings should be listed in alphabetical order (e.g. `"barack plus clinton..."` instead of `"clinton plus barack..."`), as follows:

```
sqlite> select * from sentences;
The two siblings, barack plus clinton have the same size: standard
The two siblings, abraham plus grover have the same size: toy
```

> *Hint*: First, create a helper table containing each pair of siblings. This will make comparing the sizes of siblings when constructing the main table easier.
>
> **Hint**: If you join a table with itself, use `AS` within the `FROM` clause to give each table an alias.
>
> **Hint**: In order to concatenate two strings into one, use the `||` operator.

Hint Video

Use Ok to test your code:

```
python3 ok -q sentences
```