Skills: SQL

Time to practice your database smarts.

Setup

First, create a virtual environment, activate it, and install the requirements.

```
$ virtualenv env
$ source env/bin/activate
$ pip install -r requirements.txt
```

Reminder: in Windows, use the command ‘virtualenv env –always-copy’ to create a virtual environment.

Create an empty database with:

```
$ createdb cars
```

Load a SQL file with data into the new (but empty) database:

```
$ psql cars < database.sql
```

Warm Up

Open a *psql* shell to this database.

```
$ psql cars
psql (9.4.5)
Type "help" for help.
cars=#
```

Once you’re in the *psql* shell, check out the tables in this database with the `\dt` command.

```
cars=# \dt
```

To see the columns and datatypes for any table, use the `\d TABLENAME` command.

```
cars=# \d brands
```

and run a couple of queries to become familiar with your data.

When you’re finished exploring, exit the **psql** shell using the command `\q`.

Directions

When you’re all set to begin your assessment, run the following command:

```
$ python quiz.py
```

Read the directions carefully for your first task. Try typing a solution into the prompt. It will keep prompting you for the answer until you answer it successfully.

In many cases, you’ll need to know the “schema” of a table in order to query it (“schema” is the name for the definition of a table, including the names of its fields and the type of data they hold).

You can see the schema for a particular table by using the `\d` command:

```
SQL [1]> \d models :cmd:

   Column   |         Type          | Modifiers
------------+-----------------------+-----------
 id         | integer               | not null
 year       | integer               | not null
 brand_name | character varying(50) |
 name       | character varying(50) | not null
(4 rows)
```

You can also see other system help with the `\help` command:

```
SQL [1]> \help

The following commands are available:

  \help    - Display this message
  \hint    - Show a hint about how to formulate the query
  \next    - Skip the current problem
  \problem - Show the current problem statement
  \q    - Quit the program
  \d <table_name> - Show the schema used to define a given table
  \t  - Show all the tables available in the database

Any other commands will be interpreted as a SQL query and executed against the
problem set database.
```

This interactive prompt keeps track of all of your correct answers for you. If you quit the program and restart it, it will remember which answers you've already given successfully and start you with your first unsolved question.

It does this by keeping all of your correct answers in a file in the exercise directory called ***answers.sql***. You don't need to do anything with this file (for example, you **don't** need to manually paste your answers in or anything like that; this is done for you).

** <#id1>When the output of the ***quiz.py*** script tells you that you're finished, check ***answers.sql*** into your GitHub

More Practice

For the following 3 questions, open a ***psql*** shell to construct the correct SQL statements. Add your solutions to the file called ***more-practice.sql***.

INSERT

Add the following rows to the models table: 2015 Chevrolet Malibu, 2015 Subaru Outback (year, brand name, and model name respectively).

CREATE TABLE

Add a table to the database called Awards with the following fields: name, year, and winner_id. Choose an appropriate datatype and nullability for each column.

More INSERT

Write an INSERT INTO statement that adds the following row to the Awards table.

Name: IIHS Safety Award Year: 2015 Winner ID: 49

Name: IIHS Safety Award Year: 2015 Winner ID: 50

Nice Work!

This assignment is due by Sunday at 9pm.