Problem Set 1: SQL

Assigned: 9/12/2016

Due: 9/19/2016 11:59 PM

Submit to the 6.830 Stellar Site (`https://stellar.mit.edu/S/course/6/fa16/6.830/homework/`)

You may work in pairs on this problem set. Clearly indicate the name of your partner. Only one of you needs to submit on Stellar.

1 Introduction

The purpose of this assignment is to give you hands-on experience with the SQL programming language. SQL is a declarative language in which you specify the data you want in terms of its properties. This assignment focuses on the `SELECT` subset of SQL, which is all about *querying* data rather than modifying it.

We will be using a PostgreSQL server, which provides a standards-compliant SQL implementation. In reality, there are slight variations between the SQL dialects of different vendors (PostgreSQL, MySQL, SQLite, Oracle, Microsoft, etc.) —especially with respect to built-in functions. The SQL tutorial at `http://sqlzoo.net/`, provides a good introduction to the basic features of SQL. After following this tutorial you should be able to answer most of the problems in this problem set.

We are using version 9.4.8 of Postgres, so the documentation at `https://www.postgresql.org/docs/9.4/static/` may be useful as well. You may also wish to refer to Chapter 5 of "Database Management Systems."

To access the server, you can log in to `athena.dialup.mit.edu` and start a session with:

```
psql -h geops.csail.mit.edu -p 5433 yelp
```

You must be signed up to stellar for us to know you are in the class. If you prefer, you can log in from other machines provided they have a postgres client binary (aka `psql`) and you are authenticated to Kerberos with your MIT user name.

2 Dataset

The data for this assignment is a subset of the yelp database. This subset includes four basic yelp tables: *business, users, reviews* and *tips*. You can get an idea of the data by visiting `https://www.yelp.com/dataset_challenge`. The data may not match 100% of what you see because our server has a subset of the whole dataset.

3 Using the Database

Once connected, there are two kinds of commands useful to a database user. The first kind are the `psql` client meta-commands. The most important one is `\?`, which gives you help on meta-commands. There are two others you will find very useful:

First, you can list the tables in the database with `\dt`:

```
yelp=# \dt
         List of relations
 Schema |   Name   | Type  | Owner
--------+----------+-------+-------
 public | business | table | admin6830
 public | reviews  | table | admin6830
```

```
 public | tips     | table | admin6830
 public | users    | table | admin6830
(4 rows)
```

Second, you can view the schema (recall, that the "schema" of a database is like a class definition in an object oriented language) of a given table with `\d` *table_name*:

```
yelp=# \d business;
                            Table "public.business"
    Column    |          Type          |                Modifiers
--------------+------------------------+----------------------------------------
 business_id  | character(22)          | not null
 name         | character varying(96)  | not null
 full_address | character varying(128) | not null
 city         | character varying(32)  | not null
 state        | character(2)           | not null
 latitude     | double precision       | not null default (0)::double precision
 longitude    | double precision       | not null default (0)::double precision
 stars        | double precision       | not null default (0)::double precision
Indexes:
    "business_pkey" PRIMARY KEY, btree (business_id)
```

The type of command you can issue is a SQL expressions. All SQL expressions in PostgreSQL must be terminated with a semi-colon. For example, to get a list of all records in the `business` table, you would type:

```
SELECT * FROM business LIMIT 10;
```

This query requests a maximum 10 rows from the table. Using `LIMIT` in this manner one can explore the data small bits at a time. If you really wanted to produce all the records, though, the query is:

```
SELECT * FROM business;
```

You can use `Ctrl+C` to end a query that is taking too long (it is very possible to write such bad queries even unintentionally). Note that using the `LIMIT` keyword by itself offers no guarantee on which 10 rows from the result are returned, so do not assume an ordering.

Finally, you can change the way `psql` displays the result sets to suit you better. In particular, wrapped lines in `less` can make the output of wide tables hard to read. If this bothers you, you can try exiting the client (you can use `Ctrl+D`) and starting it again with the `LESS` env. variable set like this:

```
LESS='-S' psql -h geops.csail.mit.edu -p 5433 yelp
```

4 Questions

For each question, please include both the **SQL query** and the **result** in your answer. The answers do not have to be one-liners: you can save the results of a previous query, if that is convenient to you, using `create temp table`. Also, if the query is taking too long then try changing it. All questions have solutions that run within seconds.

Q1. Find reviews whose text contains the word 'tasty' with either uppercase or lowercase 't'.

Q2. Find the 10 longest business names.

Q3. Find the number of reviews of the business from the previous query.

Q4. Find the name of the user with the largest number of fans.

Q5. Find the name of the user with the largest number of tips.

Q6. Find the name and the average stars of the business with the largest number of reviews.

Q7. Find the name and the average stars of the businesses with more than 500 reviews.

Q8. Find the names of businesses without any reviews.

Q9. Amongst users with at least 20 reviews, find the 2 "most similar" users, where the similarity of two users is defined as the fraction of shared businesses they've visited. Specifically if A and B are the sets of businesses the two users have visited, define similarity as $\frac{|A \cap B|}{|A \cup B|}$, where $|x|$ is the number of elements in x. Assume a user has visited a business if he or she has input a review or a tip on a business.

Q10. Amongst users with more than 50 reviews, for the user_id whose reviews have the lowest average star rating, find the names of the businesses that user gave the highest star ratings to.

Q11. Find the names of users who reviewed more than 5 businesses on each of 3 consecutive days.