Homework Assignment #1

(Schemas and SQL Refresher)

Congratulations are in order! You've just landed a Data Science job at Netflicks.com, where you will be working in their Movie Intelligence department. As its name might suggest, Netflicks is a new startup that's hoping to succeed in the same space as the better known company Netflix. To get a jump start on their plans and ultimately their success, they've heard that Data Science is the answer to all problems, so they're planning to leverage everything they can find in the way of data. One resource they're interested in using is the Internet Move Database Data that IMDB makes available, and you've been tasked with getting them started down that path. Your first assignment is to get ahold of the data, familiarize yourself with it, load it 'as is' into a relational database, explore it via SQL queries, and then report back to your boss about (1) what you did and how, (2) some of the 'gotchas' that you discovered about this data set (which, by the way, is quite clean in comparison with other data sets that you will encounter), and (3) a redesign of the SQL schema for the IMDB data that would be more friendly and more performant going forward.

Get Ready...

You should start adding PostgreSQL to the set of applications that you have installed on your favorite laptop or other computer if you don't already have it. Your boss has heard good things about the EDB distribution of PostgreSQL, so should use Google to find, download, and install the latest version of PostgreSQL from there (or elsewhere, if you prefer). Fire it up and use its **psql** command-line interface to create a toy table, insert a few toy rows, and then run a few toy queries to make sure you're good to go.

Get (Data) Set...

Your next step is to grab yourself a copy of the relevant IMDB data. To facilitate that process, your boss has given you pointers to a description of the IMDB data files - which themselves are in a tab-separated delimited file format (not unlike what you'd get if you were to export an Excel spreadsheet in the .txt format) - as well as to a Google Drive location containing the files and a second Google Drive location containing a small sample version of each of the files so that you can look at them to help choose appropriate field types as you go down the PostgreSQL path and need to define the SQL schemas for your tables:

Schema: http://www.imdb.com/interfaces/

Full Data Set:
https://drive.google.com/drive/folders/1b7dkQtaAdDzy1GWayshah2oXpfsrmW5K?usp=sharing

Sample Data Set (first 11 lines from each file):
https://drive.google.com/drive/folders/1I6J7WVL4dMeD_AlWHJ14DvgE7yVHXvKz

Go...!

It's time to get to work. Here's what you are to do (using psql whatever alternative PostgreSQL interface you've chosen to use):

1. Create an appropriate PostgreSQL table (using **CREATE TABLE**) for each of the seven data files. Be sure to pick appropriate data types (**int**, **float**, or **varchar**) for each of the columns of your tables, and also specify an appropriate **PRIMARY KEY** for each table if there is one.

2. Use the PostgreSQL **\COPY** command to load your tables from the data files. Note that you should probably work with the sample data files before you try loading the real files. (Also, you may want to use Google to find PostgreSQL's online documentation and bookmark a link to those docs for your chosen PostgreSQL release.) Hint: Your **\COPY** commands may ultimately need to specify, among other things, the **DELIMITER** character (tab), the **NULL** string (which you should identify from the data), and also a **QUOTE** character (you'll see why when dig into a problem that you'll encounter when loading the data files). You may also need to iterate a few times on your answer to step one in order to get all the data loaded successfully.

3. Explore the data using a **SELECT** statement in SQL (and a **CREATE INDEX** statement when so indicated) to formulate and answer the following questions (queries) on it. Use the *[indicated]* data to answer the questions.
    a. How many people are in the IMDB database in total? *[nameBasics]*
    b. List the birth years represented in the IMDB database along with the number of people born in each of those years. *[nameBasics]*
    c. For people born in this millenium, list the birth years along with the number of people born in each of those years. (Try this query and note how long it takes to run. Create an index to make the **WHERE** clause of your query run faster. Run the query again and note how long it takes.) *[nameBasics]*
    d. Each person's record in the IMDB database has a field that indicates their "primary profession". How many distinct values does that field have? Does this number match what you would have (perhaps naively) expected? Use the **LIMIT** clause in SQL to look at a few rows to see what's going on with the data. *[nameBasics]*
    e. Read the "Hints on Multivalued Data" appendix at the end of this assignment. Use the PostgreSQL *string_to_array()* function to list the id, the name, and the primary profession list for each person born in 2013 or later. *[nameBasics]*
    f. Incorporate the **UNNEST** operator of PostgreSQL to create a normalized list with the id, name, and primary profession of each person born in 2013 or later. (Keep your list entries for each person together by using an **ORDER BY** clause.) *[nameBasics]*

g. Revisit query **d**. Now that you have a better handle on the IMDB data and how to manipulate it, how many distinct profession values does the data set *really* have? *[nameBasics]*
h. Drill down into your answer to **g** by listing all of the the distinct profession values. *[nameBasics]*
i. It's time to do a bit of data deep-diving. With all that's been happening in the world, your boss is obsessed with Fox News and, as a rather strange side effect, persons named 'Fox'. List the information recorded about each person whose first name is 'Fox' (e.g., 'Fox Lancaster') who are deceased. (Try this query and note how long it takes to run. Create an index to make your query run faster and run the query again and note how long it takes.) *[nameBasics]*
j. The data includes info about which shows the people in the database were or are best known for, so dive deeper and list (in a normalized form) the person name and show types and title(s) for the dead Foxes. *[nameBasics, titleBasics]*
k. Your boss is a big Star Trek fan, so list the IMDB information about the first Star Trek movie (the movie whose name was simply 'Star Trek'). (You will probably want to create an index to accelerate this query and the next few queries; go ahead and do that.) [*titleBasics]*
l. List the person name, birth year, and primary profession for those persons who are known for being in the first Star Trek movie and who are *just* actors (i.e., for whom that's their only profession). *[nameBasics, titleBasics]*
m. Analyze the demographics of the first Star Trek movie by producing a list of the birth years of the people known for that movie who have acting as *one* of their primary professions; along with each birth year, list the number of actors born in that year. Look carefully at the biggest entry in the list - what does that tell you about the data? *[nameBasics, titleBasics]*

Revisiting the IMDB Schema

Your initial relational schema - the 'as is' schema - violates the relational model's First Normal Form (1NF) rule. You have reason to believe that the data would be easier to manipulate, and more efficiently, if it were normalized. Design a revised schema and specify the set of CREATE TABLE statements needed to create it. (*Note:* You don't need to populate the revised tables.)

What To Turn In

When you have finished your assignment you should use EEE to turn in a PDF file that lists all of the PostgreSQL statements that you ran - from start to finish - to create the tables, load data into them, run all the queries, and create any indexes that you added along the way to make the queries run faster. Your file should also include your notes on query timing, where such notes were asked for, as well as your answers to the questions interspersed with the queries. Last but not least, your file should include your recommendation for the revised relational schema for the IMDB data - in 1NF - along with a brief explanation for your recommendation.

APPENDIX: Hints on Multivalued Data

Running and pondering the following set of PostgreSQL statements together with their results may be helpful to you as you tackle some of this first assignment:

```
CREATE TABLE Person (
    id int,
    name varchar(20),
    hobbies varchar(80)
);

INSERT INTO Person VALUES
   (1, 'Joe Cool', 'karate,skiing,skydiving'),
   (2, 'Susan Smith', 'scuba,piano'),
   (3, 'Hans Solo', 'flying');

SELECT * FROM Person p;

SELECT p.id, p.name, string_to_array(p.hobbies,',')
FROM Person p;

SELECT p.id, p.name, hobby
FROM Person p, UNNEST(string_to_array(p.hobbies,',')) AS hobby
ORDER BY p.id;
```