Part 1: SQL Table Creation.

1. (14 pts) Implement the following schema using `CREATE TABLE` statements for your
   MariaDB database. The schema is loosely based on the CIA World Factbook.[1] Your
   statements must go into a script file, with appropriate comments, called `hw5-a.sql`.
   You must pick suitable attribute data types and include primary and foreign keys as
   specified below.

   `Country(`<u>`country_code`</u>`, country_name, gdp, inflation)`

   – A country has a country codes (e.g., "US"), a full name (e.g., "United States
   of America"), a gross domestic product per capita (e.g., 46,900 dollars per
   person), and inflation rates (e.g., 3.8 percent), where `country_code` is the
   primary key. Assume that two countries with different country codes can
   have the same country name.

   `Province(`<u>`province_name`</u>`, `<u>`country_code`</u>`, area)`

   – A province (which is a state in the US) consists of a name (e.g., "Washing-
   ton"), the country code the province is located in, and the total province area
   in km². The `province_name` and `country_coode` together form the primary
   key, with `country_code` a foreign key to the `Country` table. Assume it is
   possible for two countries to have a province with the same name (e.g., Mon-
   tana exists in both the US and Bulgaria).

   `City(`<u>`city_name`</u>`, `<u>`province_name`</u>`, `<u>`country_code`</u>`, population)`

   – A city is identified by its name, province, and country, and has a total pop-
   ulation. The `province_name` and `country_code` together define a foreign
   key to the `Province` table. Assume it is possible for two provinces to have
   a city with the same name (e.g., Portland is a city in both Oregon and Maine).

   `Border(`<u>`country_code_1`</u>`, `<u>`country_code_2`</u>`, border_length)`

   – A border defines a connection between two countries, with a corresponding
   border length in km. Both *country_code_1* and *country_code_2* are sepa-
   rately foreign keys to the `Country` table. Assume there is only one row in the
   table for a given border between two countries (i.e., the table does not store
   a symmetric closure over the border relation).

2. (6 pts) Populate your tables in Question 1 using `INSERT INTO` statements with enough
   data to test your table constraints. At a minimum, you must include at least three

---

[1]https://www.cia.gov/the-world-factbook/

different countries, three different provinces per country, and three different cities per province. You must also include two borders. Include your insert statements in your `hw5-a.sql` file. Note you do not have to use "real" data when populating your tables. If you use real data, it does not have to be "comprehensive", e.g., you do not need to include all provinces within a country, and you do not need to include all cities within a province.

3. (14 pts) Implement the tables from the schema you obtained by translating your Lego ER Diagram in HW-4 (Question 4). Put all of your **CREATE TABLE** statements, with appropriate comments, in a script file named `hw5-b.sql`. Select suitable attribute data types and include all key, foreign key, and non null constraints as appropriate.

4. (6 pts) Populate your tables in Question 3 using **INSERT INTO** statements with enough data to test your implementation. You are free to use real or fake data. However, if you use real data, you do not need to be comprehensive (e.g., you do not need to include all bricks in a set, etc.).

Part 2: Project. There are no additional tasks that you need to turn in for your project for this week.

Submission. Submit your answers to the above questions as follows.

- Submit both script files `hw5-a.sql` and `hw5-b.sql`. Be sure your files are commented, including a file header with your name, the class (CPSC 321), the semester, the homework number, and a brief description. Each table should have a comment describing its purpose and any additional information that may be needed to understand the table. Comment attributes as appropriate. Each attribute does not need a comment. However, comments should be provided for attributes if they may be unclear to someone else looking at your schema.

- For each table you create and populate, provide a printout of the table contents in a separate file named `hw4-results.pdf`. Include your name, the class number, the semester, and the homework number at the top of the file. You can print the contents of the table using the simple SQL query **SELECT * FROM** *table*; by replacing *table* with the name of the table you want to print out. Note you can add these select statements to the end of your script files.

- Submit your assignment files using GitHub classroom (see instructions on piazza). An initial version of `hw5-a.sql` and `hw5-b.sql` will be provided as starter code.