Database Systems - Assignment II

Download the `world_data.sql` file*, and run it to create and populate the tables. You can run the file as follows:

        psql -U postgres -f world_data.sql

Part I: The Schema.

Inspect the schema to see what attributes we have:

```
SELECT * FROM city LIMIT 2;
SELECT * FROM country LIMIT 2;
SELECT * FROM countrylanguage LIMIT 2;
```

1. Draw a schema diagram of this data, showing how the data is arranged,
        and how the tables relate to each other.

        Just use basic drawing software here – nothing fancy.
        Include any links between tables – such as FOREIGN KEY relationships.

        Hints:

        To do this, you must identify all PRIMARY KEYs and FOREIGN KEYs.
        Open the world_data.sql file in a text editor, and search it for keywords.
        Most text editors allow searching via **CTRL + f**

        You can also find this information by directly querying the database schema.
        There is a built-in table called *information_schema*, which contains all of the
        information about our tables, and we can ask it questions:

```
SELECT table_name, column_name, constraint_name
FROM information_schema.constraint_column_usage;
```

2. Is this database in Normal Form?

        If so, which one is it in? 1NF, 2NF, 3NF?
        If not, what is preventing it from being normalized? Can it be normalized?

        Hints:
        Look directly at the data, and also at the CREATE TABLE and ALTER TABLE
            statements in the `world_data.sql` file.
        The tables also have CONSTRAINTs – what do they tell you?

---

*Notice that all this data is *already* in SQL format, as opposed to Assignment 1, where we loaded the data from a comma-separated values file (".csv"). Getting this data into SQL format was accomplished by a built-in utility called pg_dump – which makes it easy to export entire databases into a file of SQL commands which we can re-load whenever we want. We will be using this utility later in this course.

Part II: Queries.

Warm-ups: Some SQL practice:

1. What are the top ten countries by economic activity (Gross National Product - 'gnp').

2. What are the top ten countries by GNP per capita?
    (watch out for division by zero here !)

3. What are the ten most densely populated countries, and ten least
    densely populated countries?

4. What different forms of government are represented in this data?
    ('DISTINCT' keyword should help here.)

    Which forms of government are most frequent?
    (distinct, count, group by order by)

5. Which countries have the highest life expectancy? (watch for NULLs).

Getting more serious – joins, joins with conditions, joins that require subqueries:

6. What are the top ten countries by total population, and what is the official language
    spoken there? (basic inner join)

7. What are the top ten most populated cities – along with which country they are in,
    and what continent they are on? (basic inner join)

8. What is the official language of the top ten cities you found in Question #7?
    (three-way inner join).

9. Which of the cities from Question #7 are capitals of their country?
    (requires a join and a subquery).

10. For the cities found in Question#9, what percentage of the country's population
    lives in the capital city? (watch your int's vs floats !).

Turn in the following:
 1. *Your schema diagram – as a .png, .jpg, or .pdf*
    *** DO NOT SEND ANY CELL PHONE PICTURES OF DIAGRAMS ! ***

 2. *A .sql file with your queries to answer Questions #1 - #10.*
    For queries, only submit the query (I don't need results - I can run them).
    Text answers to Question 2 can just be included at the top of
    your .sql file in comments.

---

*Notice that all this data is *already* in SQL format, as opposed to Assignment 1, where we loaded the data from a comma-separated values file (".csv"). Getting this data into SQL format was accomplished by a built-in utility called pg_dump – which makes it easy to export entire databases into a file of SQL commands which we can re-load whenever we want. We will be using this utility later in this course.