

Reading. Continue with the Chapter 6 reading in ZyBooks and complete all of the corresponding reading “*activities*”. You should also complete the four “labs” in Chapter 7.

Technical Work. Consider the following database schema, which is loosely based on the CIA World Factbook.¹

Country(country_code, country_name, gdp, inflation)

- Countries consisting of their country codes (e.g., “US”), names (e.g., “United States of America”), 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

Province(province_name, country_code, area)

- Provinces (states in the US) consist of their names, their countries, and their areas (in km²), where *province_name* and *country_code* together form the primary key, and *country_code* is a foreign key to Country.country_code

City(city_name, province_name, country_code, population)

- Cities consist of their names, provinces, countries, and population, where *city_name*, *province_name*, and *country_code* together form the primary key, and *province_name* and *country_code* together are a foreign key to Province.(province_name,country_code)

Border(country_code_1, country_code_2, border_length)

- Borders between countries with their border lengths (in km), where *country_code_1* and *country_code_2* together form the primary key, and *country_code_1* and *country_code_2* are both foreign keys to Country.country_code. 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).

Assume it is possible for: (i) two different cities in different provinces to have the same name; (ii) two different provinces in different countries to have the same name; and (iii) two countries with different country codes to have the same country name.

¹<https://www.cia.gov/library/publications/the-world-factbook>

1. Implement the above tables in MySQL using create table statements. Populate your tables with enough data to fully test your answers to the following queries. All of your create table, insert, and query statements must be placed in a `hw6.sql` script file. Note you do not have to use “real” data when populating your tables. Your data also does not need to be “comprehensive”, e.g., if you do represent actual country and province data, you do not need to include all provinces within a country, and you do not need to include all cities within a province.

For the following queries, use variables within your SQL script for queries that ask for “specific values.” For instance, if we want to find the names of countries with a specific GDP value, we would create the following.

```
SET @gdp = 50;
SELECT c.country_name
FROM   country c
WHERE  c.gdp = @gdp;
```

2. Write an SQL query to find the countries containing a province with a small area, but where the country has low inflation and high gdp. Use specific values for the area, inflation, and gdp cutoffs as appropriate to fit your data. Your query should return the name, code, gdp, and inflation for each such country, and only return one row per matching country. Your query must use a comma join.

3. Rewrite your query from (2) using inner JOIN syntax.

4. Write an SQL query that finds the set of all provinces that have at least one city with a population greater than 1000. Return the country code, country name, province name, and province area. Your query must use comma joins.

5. Rewrite your query from (4) using inner JOIN syntax for all of the joins.

6. Write an SQL query that finds the total area of all of the provinces within countries that have a gdp within a given range of values. Your query should return the total area.

7. Write an SQL query that finds the minimum, maximum, and average of both gdp and inflation of each country.

8. Write an SQL query that finds the number of cities and the average population within a specific country. You choose which country.

9. Write an SQL query that finds the average population of cities that are within the same province and country as a given city. Do not include the given city's population in the average population calculation.
10. Write an SQL query that finds the average population for each country in your database. The population of a country is the total population of each city in the country's provinces.
11. Write an SQL query that finds each country's total area and average population for countries with an inflation below a specific value and with a total population above a specific value. Return each country's code, name, gdp, inflation, total area, and average population. Note that the total area of a country is the sum of the areas of its provinces.
12. Write an SQL query that finds the number of countries that a specific country borders and its corresponding average border length. You choose which country.
13. Create an SQL view named **SymmetricBorders** that uses UNION to create a "symmetric" version of the borders relation. For instance, if a row ("a", "b", 100) is in **Borders**, then both ("a", "b", 100) and ("b", "a", 100) should be in the view.
14. Using the view you created in 13, create an SQL query that finds for each country c the number of c 's bordering countries that have a larger inflation rate but a smaller gdp than c . Your query should return the name and code of the country along with the total number of matching countries.
15. Write an SQL query that finds each country $c1$'s bordering countries $c2$ such that $c1$ has at least a 10% smaller population than $c2$, covers a larger area than $c2$, and has a larger gdp than $c2$. Return the country codes for $c1$ and $c2$ along with each country's population, area, and gdp.

Similar to HW-5, create a **hw6.sql** script file that contains your table creation, table population, and queries for questions 1–15. You must also include comments in your script file, including a header file comment (with your name, the course, assignment number, and brief description) as well as comments for your tables and each query as needed. As part of your query comments include the problem number as well. If you are unsure what is being asked for above, please ask for clarification on Piazza. Submit your **hw6.sql** file to the GitHub classroom for HW-6 along with a print-out showing your database tables and the results of running your queries (a link will be provided in Piazza).