Part 1: Reading Assignment. Read the following sections in the textbook and then answer the questions below.

- Ch. 3: 3.3.3, 3.4, 3.5, 3.6, 3.7.1

1. Briefly describe the difference between a natural join and a foreign key, and whether these two things are related at all.

2. Given an example of one of your queries from HW3 as well as a rewritten example that is an equivalent query but that uses join syntax instead.

3. Briefly describe the reasons why someone would want to use a rename operation in a query.

4. Briefly describe what the purpose of the “order by” clause is and give examples of how you could meaningfully use it in queries against your music schema. When comparing two versions of the same query, where one uses order by and the other doesn’t, which do you think is more expensive (in terms of query time) to execute and why?

Part 2: Database Implementation and Queries II. Consider the following database schema, which is loosely based on the CIA World Factbook.¹

Country(code, country_name, gdp, inflation)
- Countries consisting of their 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 code is the primary key

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

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

Border(country1, country2, border_length)
- Borders between countries with their border lengths (in km), where country1 and country2 together form the primary key, and country1 and country2 are both foreign keys to 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).

1. Implement the above tables in MySQL using create table statements. Populate your tables with enough data to test your answers to the following queries.

2. Write an SQL query that finds the names of all countries with low inflation and high gdp. You can give specific values for these to fit your data.

3. Write an SQL query that finds the set of all province names and areas that have at least one city with a population greater than 1000.

4. Write an SQL query that finds the total area of the provinces within a specific country. You choose which country. Note that you don’t need to define all provinces of the country you chose.

5. Write an SQL query that finds the minimum, maximum, and average of both gdp and inflation of all countries in your database.

6. Write an SQL query that finds the number of cities within a specific country. You choose which country. Note that like for provinces, you don’t have to define all the cities within a country.

7. Write an SQL query that finds the names of countries with both a lower gdp and higher inflation rate than a country it borders.

8. Write an SQL query that that finds the number of countries that a specific country borders and its corresponding average border length. You choose which country.

**Turn in** your CREATE TABLE statements, the contents of your table as shown by MySQL (e.g., using SELECT * FROM ...), your SQL queries, and the result of executing these queries on your tables. All of the above should be captured in a single hw4.sql script. Be sure to submit your script to the course submission site and turn in both your reading and homework together with a cover sheet.