**Introduction:**

In this lab, we will create an application to populate a database from the provided Geo City Lite CSV file and allow the users to search for cities near a particular lat/long or another city. The application will start by loading the data from the csv file into the database, if it has not yet been loaded. Then the application allow the user to enter the total number of cities to search for and either lat/long or a city name. For lat/long simply apply the nearby search to locate cities near the lat/long from the DB. In case of city name, use reverse search in the db to find out the lat/long for that city, before applying the same nearby search algorithm.

**Approach:**

Latitude defines how far north or south of the equator a point is positioned.  Points alongside the equator have latitudes of zero. The North Pole has a positive (north) latitude of 90, and the South Pole a negative (south) latitude of -90. Accordingly, northern-hemisphere locations have positive latitude, and southern-hemisphere locations have negative latitude. Longitude describes how far east a point is, from the prime meridian: an arbitrary line on the earth surface running from pole to pole.

Latitudes are values in the range [-90, 90]. Longitudes are values in the range (-180, 180]. These values are sometimes expressed in degrees, minutes, and seconds, rather than degrees and decimals.  If you’re planning to do calculations, convert the minutes and seconds to decimals first.

The user can provide either lat/long or a city name (reverse search the lat/long), which will be utilized to look for a certain number (based on user input) of nearby cities. Remember, since the earth is not flat, you cannot apply simple arithmetic operations to find out the nearby cities, instead use the “The Great Circle Distance Formula” to find out the nearby lat/long. A short sample SQL to do the same is as follows:

DEGREES(ACOS(COS(RADIANS(lat1)) \* COS(RADIANS(lat2)) \*

COS(RADIANS(long1) - RADIANS(long2)) +

SIN(RADIANS(lat1)) \* SIN(RADIANS(lat2))))

This formula in its current form might not work with Java so you are encouraged to do some research on the internet to find out how to do this task.

**How to Run:**

Create database.

Run the program.