

Introduction

Implement the functions provided in the Jupyter Notebook to perform the operations as listed below. You may use the included sample.db to test your functions with the Python code provided or you may use the Jupyter Notebook environment, which has the sample database embedded within.

A. FindBusinessBasedOnCity(cityToSearch, saveLocation1, collection) – This function searches the 'collection' given to find all the business present in the city provided in 'cityToSearch' and save it to 'saveLocation1'. For each business you found, you should store the name, full address, city, and state of the business in the following format. Each line of the saved file will contain: Name\$FullAddress\$City\$State. (\$ is the separator and must be present.)

B. FindBusinessBasedOnLocation(categoriesToSearch, myLocation, maxDistance, saveLocation2, collection) – This function searches the 'collection' given to find the name of all the businesses present in the 'maxDistance' from the given 'myLocation' (please use the distance algorithm given below) and save them to 'saveLocation2'. Each line of the output file will contain the name of the business only.

Distance Algorithm

A distance algorithm will need to be used. Given two pairs of latitude and longitude as [lat2, lon2] and [lat1, lon1], you can calculate the distance between them using the formula given below:

DistanceFunction(lat2, lon2, lat1, lon1):

- var R = 3959; // miles
- var ϕ_1 = lat1.toRadians();
- var ϕ_2 = lat2.toRadians();
- var $\Delta\phi$ = (lat2-lat1).toRadians();
- var $\Delta\lambda$ = (lon2-lon1).toRadians();
- var a = Math.sin($\Delta\phi/2$) * Math.sin($\Delta\phi/2$) + Math.cos(ϕ_1) * Math.cos(ϕ_2) * Math.sin($\Delta\lambda/2$) * Math.sin($\Delta\lambda/2$);
- var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
- var d = R * c;
- d is the distance between the given pair of latitude and longitude. The distance is in **miles**.

Reference: <http://www.movable-type.co.uk/scripts/latlong.html>