

ECE:5995 Modern Databases – Fall 2023

Homework 6 – Indexing in MongoDB

Due Monday Nov 6th, 2023 11:59pm.

For this homework we will use the business collection posted in ICON.

Part 0. Import the business data.

Import the data into a collection named business in a hw6 mongo database.

Part 1. Spatial indexing in MongoDB.

MongoDB geospatial tutorial (<https://docs.mongodb.com/manual/tutorial/geospatial-tutorial/>)

1. Add a 2dsphere spatial index for the location (longitude/latitude) of the businesses using the createIndex command. What is the size of the location index in KBs and how does it compare to the size of the collection? You can use the stats command over the collection to find out the index size.
2. Execute the following queries. Use latitude 28.4519912466 and longitude -81.4780020341 as your current location. Unless otherwise specified, include the relevant business names, addresses, cities, states, stars, and categories.
 - a. Find the business located at your current location.
 - b. Find all businesses within the region/neighborhood delimited by the following polygon coordinates. List the business names and the count of businesses in the region/neighborhood (these can be separate queries):

```
{ "type": "Polygon", "coordinates": [[  
  [-81.47602558135986, 28.451751000365167],  
  [-81.47530674934387, 28.453703611765697],  
  [-81.47952318191528, 28.453807372715158],  
  [-81.4800703525543, 28.45156234034854],  
  [-81.47602558135986, 28.451751000365167]] ] }
```

- c. You want to get a bite to eat while waiting. Find the 5 closest restaurants to your current location. Restaurants are considered businesses with the word “restaurants” within the categories string.

Part 2. Text (inverted) indexes in MongoDB.

MongoDB text tutorial ([Mongo text search](#))

1. Create a text index on the name, city, state, and categories field. How big is the index in KBs and how does it compare with the size of the collection?
2. Write the following queries using the text index. For all the queries, display the name, city, state, categories, and stars fields. Unless otherwise specified, sort by the number of stars and then alphabetically by business name.
 1. Businesses with at least 4 stars with the keywords "family", "fun", AND "entertainment"
 2. Businesses in Portland with the keywords "grill" AND "cheese"
 3. Businesses in Florida with the keyword "bar" but NOT "brewery" or "restaurant". Limit to the first 10.
 4. The top 5 businesses in Florida with the keywords "family", "fun", OR "entertainment", ordered by the text search score. When using a text index, you can sort the results by an internal score computed by mongo: {score: {\$meta: 'textScore'}}.

Part 3. Connecting to MongoDB through code

Using your favorite programming language (preferably python or java), connect to your hw6 database and the business collection.

[Start Developing with MongoDB — MongoDB Drivers](#)

[Getting Started with Python and MongoDB | MongoDB](#)

[Using MongoDB with Java | MongoDB](#)

1. Provide a menu for a user to search for the three closest businesses to a given location (lat/long). Display the name, address, city, state, categories, stars, and number of reviews (review_count).
2. Create a menu for a user to search for a business using the business id and submit a new review score ranging from 1 to 5. Update the star rating of the business in the business collection using the formula: $(stars * review_count + \text{new review score}) / (review_count + 1)$ and increment the review count for each new review. Store all information about the reviewed businesses in a separate collection named "reviews".

Submission:

Submit your code, queries, and answers as a single zip file to the Dropbox for Homework 6 in ICON by **Monday November 6th 11:59pm.**