

## Overview

This Project is worth 100 points (out of 1000) toward your final grade. It is due on Sunday, November 27 at 11:59 p.m. Your assignment submission should be a document saved and submitted as a PDF file via the link found in the assignment section of “Week 13” in Canvas which is the same place where you found this file.

This assignment will give you hands-on practice in working with the MongoDB “NoSQL” database software.

## Objectives

1. Download and install MongoDB
2. Create a MongoDB database to store a collection of documents
3. Load a large amount of document-based data into the collection
4. Query the document collection to research a topic and answer questions

## Introduction:

The following links are helpful for an introduction MongoDB concepts and learning the basics of the MongoDB query language.

1. [https://www.tutorialspoint.com/mongodb/mongodb\\_overview.htm](https://www.tutorialspoint.com/mongodb/mongodb_overview.htm) : This link provides a basic overview of MongoDB and the basic queries for inserting documents, updating documents, query documents etc.
2. <https://www.guru99.com/mongodb-tutorials.html#1> : This is a beginner tutorial and also has links for installing and running mongoDB with images which may be helpful for students.
3. <https://docs.mongodb.com/manual/introduction/> : The official MongoDB documentation page.

## Installation:

Please follow the installation instructions handed out during the recitation session on Friday, April 1.

First, you can install Docker. Then you can install MongoDB to run under Docker.

Or, if you prefer, you can simply download and install MongoDB on your PC (without using Docker.)

After you have installed MongoDB, you will import a “Big Data” dataset into your MongoDB instance and run some queries.

## Query Exercises:

This document contains two sections. In the first, **Section 1**, you will have the opportunity to play with different basic operations in MongoDB. You can then use MongoDB with a real-world data set, and answer the questions below based on the data set in **Section 2**.

### Section 1: Basic operations in MongoDB

#### 1. Create a database: `use DATABASE_NAME`

Ex: `>use new_mongo_db`

The above command will create a database if it does not exist and/or use the database if it already exists.

Replace `DATABASE_NAME` with the name of the database you want to create.

#### 2. Drop a database: `db.dropDatabase()`

Ex: `>use new_mongo_db`  
`switched to db new_mongo_db`  
`>db.dropDatabase()`

- i. First you need to switch (with a “USE”) to the database to be dropped. Then the above command will drop that database.

#### 3. Creating a collection: `db.createCollection(name, options)`

Ex: >use new\_mongo\_db  
switched to db new\_mongo\_db  
>db.createCollection("test\_collection")  
{ "ok" : 1 }

i. The above command creates the collection. But providing some initial additional options could be useful in certain circumstances.

ii.

1. >db.createCollection("mycol", { capped : true, autoIndexId : true, size : 6142800, max : 10000 } )  
{ "ok" : 1 }

2. For more information on the options, please check the following link.

<https://docs.mongodb.com/manual/reference/method/db.createCollection/>

iii. In mongoDB, it is not necessary to create a collection. When a new document is inserted, mongoDB creates a collection automatically.

#### 4. Dropping a collection: `db.COLLECTION_NAME.drop()`

Ex: >use new\_mongo\_db  
switched to db new\_mongo\_db  
>db.test\_collection.drop()  
True

i.

ii. First you need to switch (with a "USE") to the database, and then use the above command to delete the collection.

iii.

#### 5. Insert a document: `db.COLLECTION_NAME.insert(document)`

Ex: > db.test\_collection.insert({  
title: "Mongo Db practice",  
description: "this is my first MongoDB document"  
})

Replace the `COLLECTION_NAME` with the name of the collection into which you wish to insert documents.

**6. Query a document:** `db.COLLECTION_NAME.find()`

i.

Ex: `>db.test_collection.find().pretty()`

ii. The above query will display the documents present in the collection.

iii. Try it with and without the `".pretty()"` option.

**7. Update a document:** `db.test_collection.update(SELECTION_CRITERIA, UPDATED_DATA)`

Ex: `>db.test_collection.update({'title':'MongoDB practice'},{$set:{ 'title':'Updated MongoDB practice'}})`

i.

ii. The above example is used to update the documents that contain 'title' as 'MongoDB practice' to 'Updated MongoDB practice'

**8. Delete a Document:** `db.COLLECTION_NAME.remove(DELETION_CRITERIA)`

i. Remove only one record:

1. `db.test_collection.remove({ status : "P" },1)`

2. Here the first document which has this key value pair will be deleted.

ii. Remove all records matching a condition:

1. `db.test_collection.remove({ status : "P" })`

2. Here all the documents which have this key value pair will be deleted.

## Section 2: Use a real-world data set to answer the seven questions below

1. Download the “primer-data.json” JSON dataset using the link found in the Week 14 Module.
2. From your **Terminal** type the following command to import the dataset into your Mongo collection.

```
mongoimport --db test --collection restaurants --drop --file  
~/Desktop/primer-data.json
```

The above `mongoimport` command converts the json file and stores it as a set of documents with the collection name of “restaurants”.

**NOTE:** This command is run in the Docker / Terminal Shell / Command Prompt, not within the Mongo interface.

**ALSO NOTE:** This command includes the name of your mongodb database (highlighted in **GREEN** above.) And it includes the name of your collection also highlighted in **GREEN**. If you changed the names of the database and/or collection as you did the import, you will have to change it in the import command string as well.

**ALSO NOTE:** This command points to the PATH where you put the file when you downloaded it (highlighted in **BLUE** above.) If you put it on your desktop, then the command should work as typed above. If you put the file elsewhere when you downloaded it, then you must modify the import command to point to the path where you put the file when you downloaded it.

Answer the following questions by writing queries and displaying the results.

- (1) How many restaurants are there in this collection?
- (2) List in alphabetical order each different (distinct) cuisine represented in this collection.
- (3) Return the name of all restaurants within the zipcode 10023 which serve Italian cuisine. Return only the names of the restaurants.
- (4) Which Borough has the most Greek restaurants? How many are there?
- (5) Return a list of restaurants (names) which have the string “Pho ” in their name. (“Pho” is a wonderful and delicious Vietnamese noodle soup.)
- (6) Return a list of boroughs ranked by the number of Italian restaurants in the borough. That is, for each borough, find how many restaurants serve Italian cuisine and print the borough and the number of such restaurants sorted descending by this number. (HINT: use the aggregate

method, and use a \$group and a \$sum.)

(7) Find the top 5 Greek restaurants in Brooklyn that have the highest total score. Return for each restaurant the restaurant's name and the total score. (HINT: use the aggregate method with \$unwind to parse out the scores array, followed by a \$group and a \$sum.)

## Submission Requirement:

Capture screen shots to show evidence of having completed **Section 1** described above. Number each screen shot with the number of the assigned operation/task (1 – 8). Assemble (Copy & Paste) all screen shots into a document.

For **Section 2**, screenshot or copy/paste your MongoDB command and output for each question. If the output exceeds 24 lines, just capture the first 24 lines.

Save your document as a PDF and upload to Canvas.

### NOTES:

You may work with a partner on completing this assignment. However, this is an **individual** assignment; each one must submit your own final deliverable for this assignment.

If you did work with a partner, be sure to specify your partner's name on the document you submit.