Name: Chakrya Ros

Homework 6

**Task 1: Basic operations in MongoDB**

**1. Create a database:**

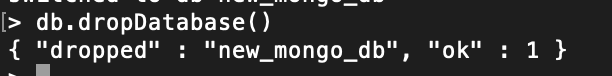
> use new\_mongo\_db



page2image61080896

2. **Drop a database:**

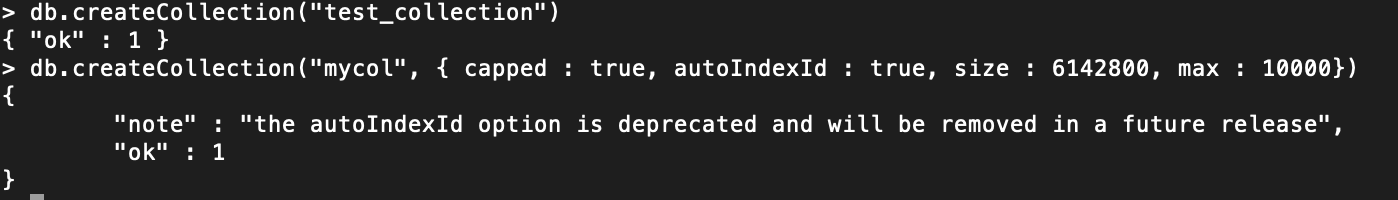
> db.dropDatabase()



**3. Creating a collection:**

> db.createCollection("test\_collection")

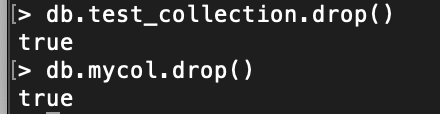
> db.createCollection("mycol", { capped : true, autoIndexId : true, size : 6142800, max : 10000})



**4. Dropping a collection:**

> db.test\_collection.drop()

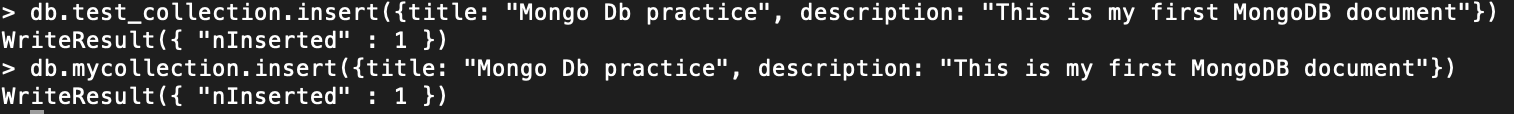
> db.mycol.drop()



**5. Insert a document:**

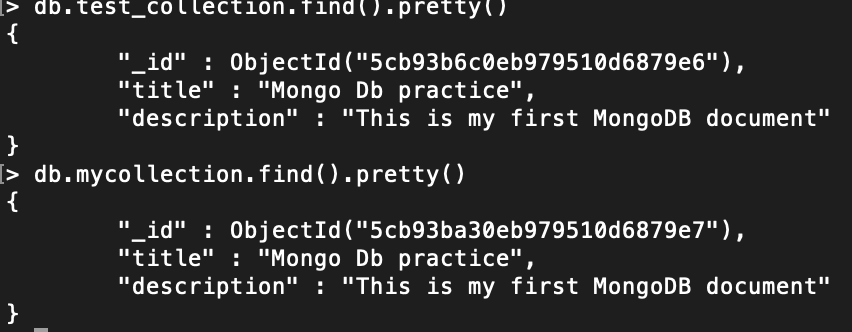
> db.test\_collection.insert({title: "Mongo Db practice", description: "This is my first MongoDB document"})

> db.mycollection.insert({title: "Mongo Db practice", description: "This is my first MongoDB document"})



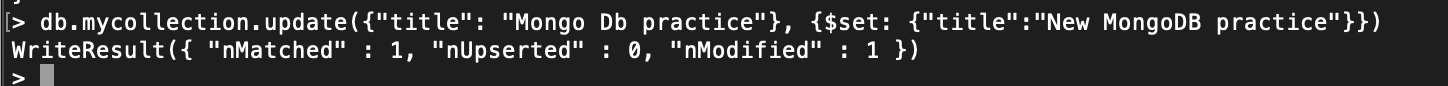
**6. Query a document:**

> db.mycollection.find().pretty()

****

**7. Update a document:**

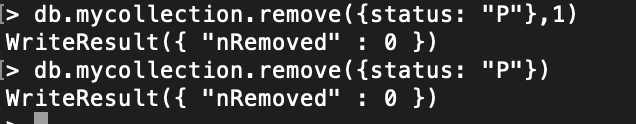
> db.mycollection.update({"title": "Mongo Db practice"}, {$set: {"title":"New MongoDB practice"}})



8. **Delete a Document:**

> db.mycollection.remove({status: "P"},1)

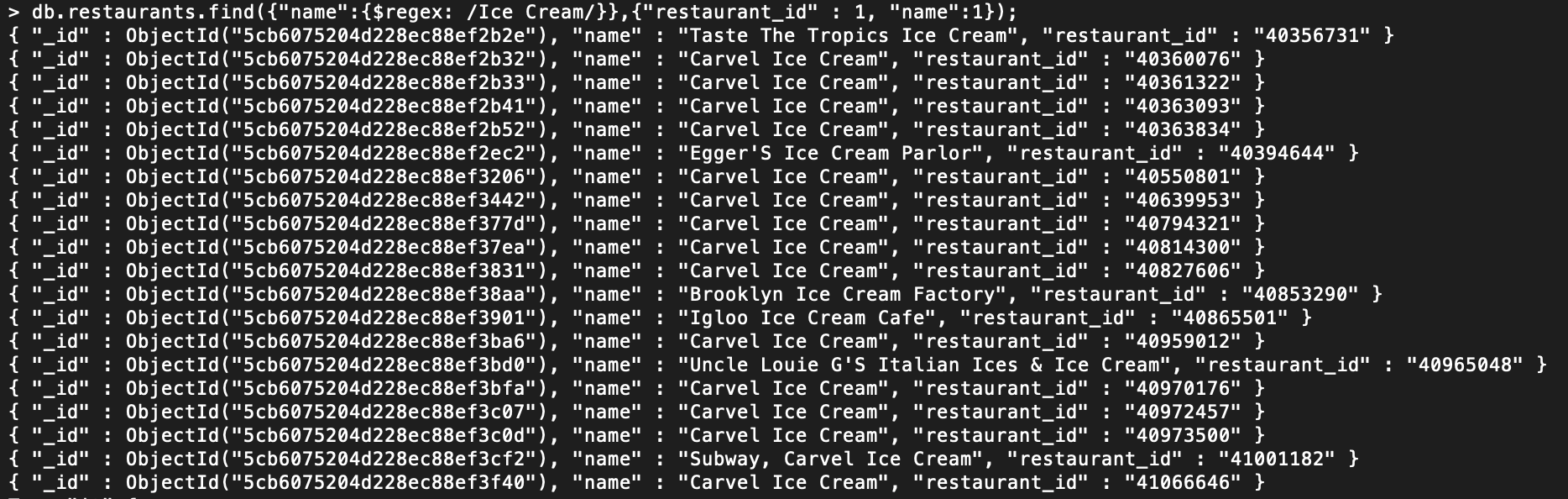
> db.mycollection.remove({status: "P"})



**Task 2: Use real-world data set to answer the following question**

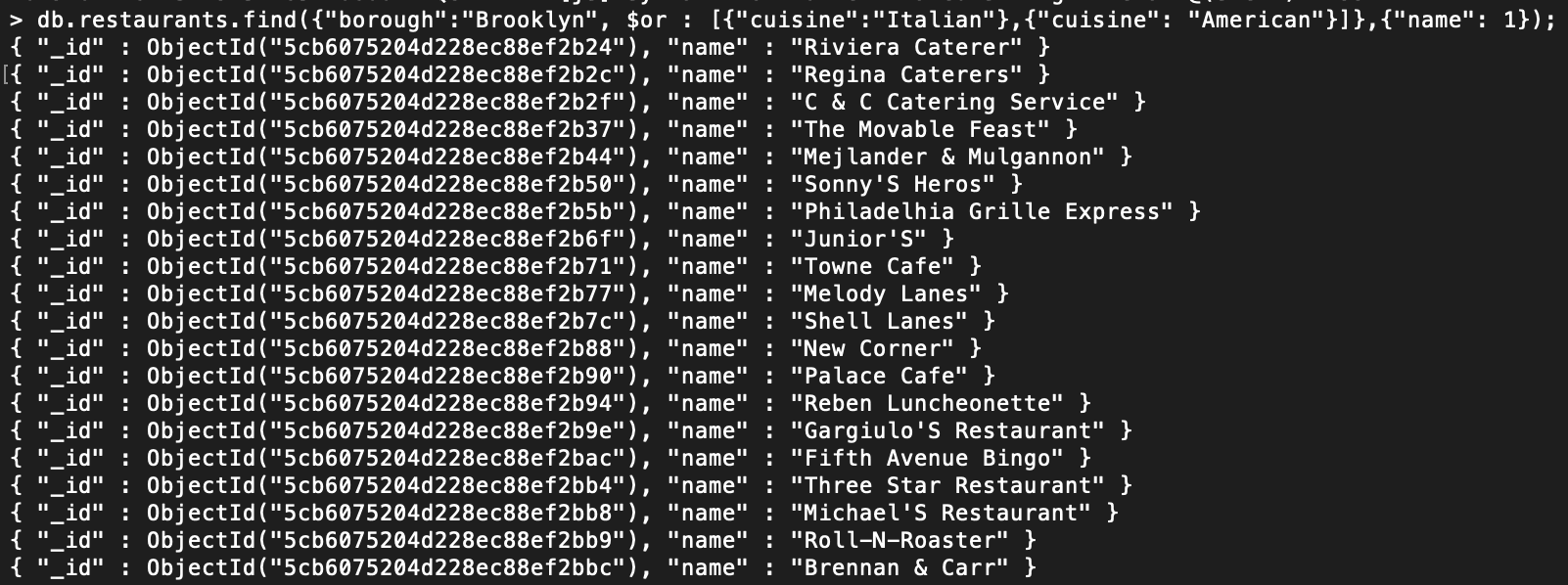
(1) List the restaurants that have the string “Ice Cream” in their name. Return only the restaurant id and name. (HINT: use Regex.)

> db.restaurants.find({"name":{$regex: /Ice Cream/}},{"restaurant\_id" : 1, "name":1});



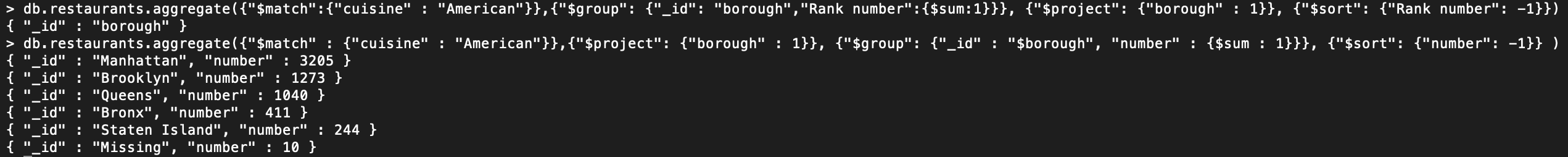
(2) Find the names of all restaurants that serve either Italian or American cuisine and are located in the Brooklyn borough.

> db.restaurants.find({"borough":"Brooklyn", $or : [{"cuisine" : "Italian"},{"cuisine" : "American"}]},{"name" : 1});



(3) Return a list of boroughs ranked by the number of American restaurants in the borough. That is, for each borough, find how many restaurants serve American 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.)

> db.restaurants.aggregate({"$match" : {"cuisine" : "American"}},{"$project": {"borough" : 1}}, {"$group": {"\_id" : "$borough", "number" : {$sum : 1}}}, {"$sort": {"number": -1}});



(4) Find the top 5 American restaurants in Manhattan 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.)

db.restaurants.aggregate([{$unwind: "$grades"}, {$match: {cuisine: "American", borough: "Manhattan"}}, {$group: {\_id: "$\_id", name: {$first: "$name"}, totalscore: {$sum: "$grades.score"}}}, {$sort: {totalscore: -1}}, {$limit: 5}])



(5) Consider the area of the location field identified by the vertices [ -74 , 40.5 ] , [ -74 , 40.7 ] , [ -73.5 , 40.5 ] and [ -73.5 , 40.7 ]. Find the number of restaurants in this area that have received a grade score (at least one) more than 75. No need to sum scores. (Hint: count the restaurants whose location coordinates mathematically fall within the bounds set by the coordinates in the question.)

> db.restaurants.find({$and:[{grades: {$elemMatch:{score:{$gt : 75}}}},{"address.coord" :{$geoWithin :{$box:[[-74,40.7],[73.5,40.5]]}}}]}).count()

