Homework5

Chengming Li

1. Section 1 Basic operations in MongoDB
    a. Create a database

```
mongo-client> use new_mongo_db
switched to db new_mongo_db
new_mongo_db>
```

    b. Drop a database

```
new_mongo_db> db.dropDatabase()
{ ok: 1, dropped: 'new_mongo_db' }
new_mongo_db>
```

    c. Creating a collection

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

    d. Dropping a collection

```
new_mongo_db> db.test_collection.drop()
true
new_mongo_db> db.mycol.drop()
true
new_mongo_db>
```

    e. Insert a document

```
new_mongo_db> db.test_collection.insert({
... title: "Mongo Db practice",
... description: "this is my first MongoDB document"
... })
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("643dd8f79cf4f98322e5f5c4") }
}
```

    f. Query a document

```
new_mongo_db> db.test_collection.find().pretty()
[
  {
    _id: ObjectId("643dd8f79cf4f98322e5f5c4"),
    title: 'Mongo Db practice',
    description: 'this is my first MongoDB document'
  }
]
new_mongo_db> db.test_collection.find()
[
  {
    _id: ObjectId("643dd8f79cf4f98322e5f5c4"),
    title: 'Mongo Db practice',
    description: 'this is my first MongoDB document'
  }
]
```

g. Update a document

```
new_mongo_db> db.test_collection.update({'title':'Mongo Db practice'}, {$set:{'title':'Updated MongoDB practice'}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
new_mongo_db> db.test_collection.find()
[
  {
    _id: ObjectId("643dd8f79cf4f98322e5f5c4"),
    title: 'Updated MongoDB practice',
    description: 'this is my first MongoDB document'
  }
]
```

h. Delete a document

```
new_mongo_db> db.test_collection.remove({title:"Updated MongoDB practice"}, 1)
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 1 }
```

---

2. Use the New York City Restaurants data set to answer the seven questions below.
   a. How many restaurants are there in this collection?
   db.test_collection.count()

```
mongo_db> db.test_collection.count()
25359
```

   b. List in alphabetical order each different (distinct) cuisine represented in this collection.

db.test_collection.distinct('cuisine').sort()

```
mongo_db> db.test_collection.distinct('cuisine').sort()
[
  'Afghan',
  'African',
  'American',
  'Armenian',
  'Asian',
  'Australian',
  'Bagels/Pretzels',
  'Bakery',
  'Bangladeshi',
  'Barbecue',
  'Bottled beverages, including water, sodas, juices, etc.',
  'Brazilian',
  'CafÃ©/Coffee/Tea',
  'Café/Coffee/Tea',
  'Cajun',
  'Californian',
  'Caribbean',
  'Chicken',
  'Chilean',
  'Chinese',
  'Chinese/Cuban',
  'Chinese/Japanese',
  'Continental',
  'Creole',
  'Creole/Cajun',
  'Czech',
  'Delicatessen',
  'Donuts',
  'Eastern European',
  'Egyptian',
  'English',
  'Ethiopian',
  'Filipino',
  'French',
  'Fruits/Vegetables',
  'German',
  'Greek',
  'Hamburgers',
  'Hawaiian',
  'Hotdogs',
  'Hotdogs/Pretzels',
  'Ice Cream, Gelato, Yogurt, Ices',
  'Indian',
  'Indonesian',
  'Iranian',
  'Irish',
  'Italian',
  'Japanese',
```

c. Return the name of all restaurants within the zipcode 10023 which serve Italian cuisine. Return only the names of the restaurants.

db.test_collection.find(
{$and:[{"cuisine":"Italian"},
{"address.zipcode":"10023"}]},
{"name":1 ,"_id": 0})

```
mongo_db> db.test_collection.find( { $and: [{ "cuisine
[
  { name: 'Fiorellos' },
  { name: "Gabriel'S Bar & Grill" },
  { name: 'Il Violino' },
  { name: 'Pomodoro Ristorante' },
  { name: "Nick And Toni'S Cafe" },
  { name: 'Arte Cafe' },
  { name: 'Bello Giardino' },
  { name: 'Cesca' },
  { name: 'Arte Around The Corner' },
  { name: 'Riposo 72' },
  { name: 'Salumeria Rossi Parmacotto' },
  { name: 'Luce Restaurant & Enoteca' },
  { name: 'Gina La Fornarina' },
  { name: 'Lincoln Ristorante' },
  { name: 'The Leopard At Des Artistes' },
  { name: 'Pappardella' },
  { name: 'Joanne Trattoria' }
]
```

d. Which Borough has the most Greek restaurants? How many are there?

db.test_collection.aggregate([
{$match:{cuisine:"Greek"}},
{$group: {_id:"$borough", count:{$sum:1}}},
{$sort:{count:-1}}
])

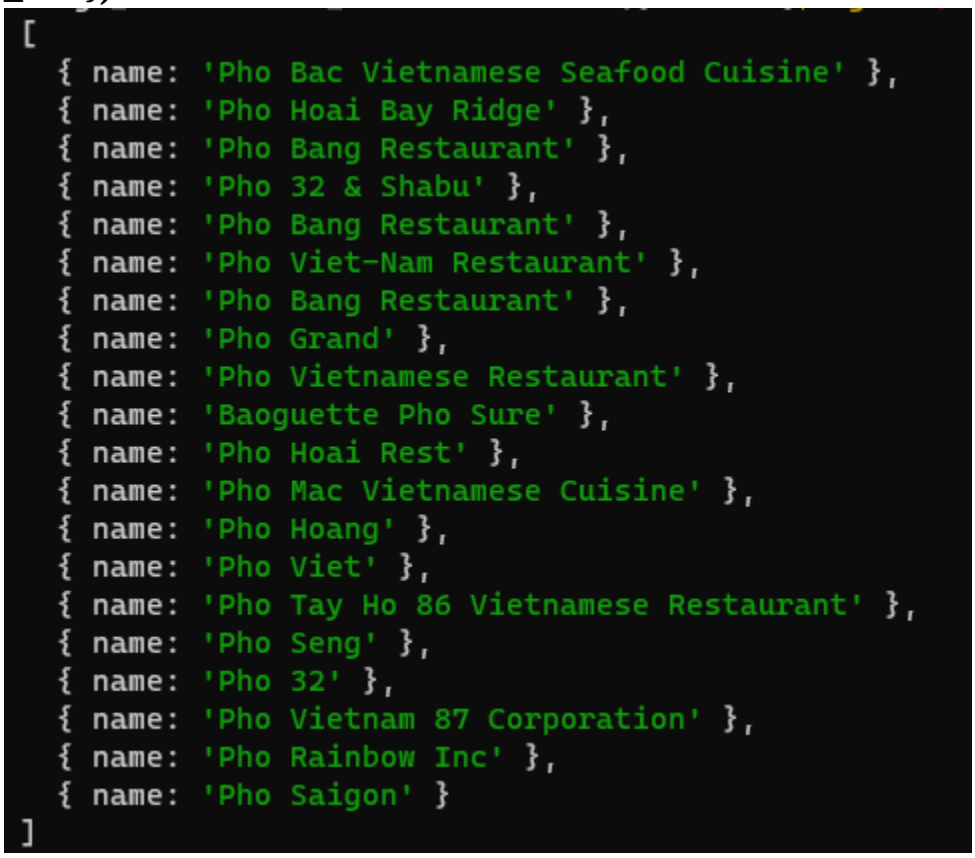
Answer:

{ _id: 'Queens', count: 58 }

```
mongo_db> db.test_collection.aggregate([{$match:{cuisine:"Greek"}},{$group: {_id:"$borough", count:{$sum:1}}},{$sort:{count:-1}}])
[
  { _id: 'Queens', count: 58 },
  { _id: 'Manhattan', count: 35 },
  { _id: 'Brooklyn', count: 14 },
  { _id: 'Bronx', count: 4 }
]
```

e. Return a list of restaurants (names) which have the string “Pho ” in their name. (“Pho” is a wonderful and delicious Vietnamese noodle soup.)

db.test_collection.find({"name":{$regex: /Pho /}},{"name": 1, _id:0})



```
[
  { name: 'Pho Bac Vietnamese Seafood Cuisine' },
  { name: 'Pho Hoai Bay Ridge' },
  { name: 'Pho Bang Restaurant' },
  { name: 'Pho 32 & Shabu' },
  { name: 'Pho Bang Restaurant' },
  { name: 'Pho Viet-Nam Restaurant' },
  { name: 'Pho Bang Restaurant' },
  { name: 'Pho Grand' },
  { name: 'Pho Vietnamese Restaurant' },
  { name: 'Baoguette Pho Sure' },
  { name: 'Pho Hoai Rest' },
  { name: 'Pho Mac Vietnamese Cuisine' },
  { name: 'Pho Hoang' },
  { name: 'Pho Viet' },
  { name: 'Pho Tay Ho 86 Vietnamese Restaurant' },
  { name: 'Pho Seng' },
  { name: 'Pho 32' },
  { name: 'Pho Vietnam 87 Corporation' },
  { name: 'Pho Rainbow Inc' },
  { name: 'Pho Saigon' }
]
```

f. 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.)

db.test_collection.aggregate([{
$match:{cuisine:"Italian"}},
{$group: {_id:"$borough", count:{$sum:1}}},
{$sort:{count:-1}}
])

```
mongo_db> db.test_collection.aggregate([{$match:{cuisine:"Italian"}},{$group: {_id:"$borough", count:{$sum:1}}},{$sort:{count:-1}}])
[
  { _id: 'Manhattan', count: 621 },
  { _id: 'Brooklyn', count: 192 },
  { _id: 'Queens', count: 131 },
  { _id: 'Staten Island', count: 73 },
  { _id: 'Bronx', count: 52 }
]
```

g. 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.)

db.test_collection.aggregate([
{$match:{cuisine:"Greek",borough:"Brooklyn"}},
{$unwind:"$grades"},
{$group:{_id:"$name",total_score:{$sum:"$grades.score"}}},
{$sort:{total_score:-1}},
{$limit: 5}])

```
mongo_db> db.test_collection.aggregate([{$match:
[
  { _id: 'Spartan Souvlaki', total_score: 81 },
  { _id: 'Yiasou', total_score: 44 },
  { _id: 'Plaka Restaurant', total_score: 41 },
  { _id: 'Faros Restaurant', total_score: 40 },
  { _id: 'Meze', total_score: 39 }
]
```