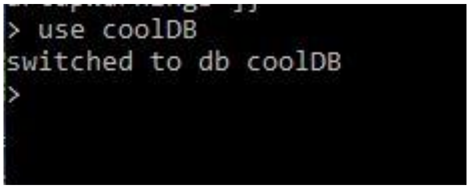
John Ortiz

# Homework 6

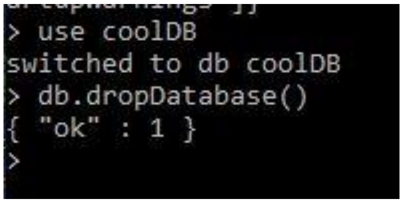
## Section 1:

1. Create a database



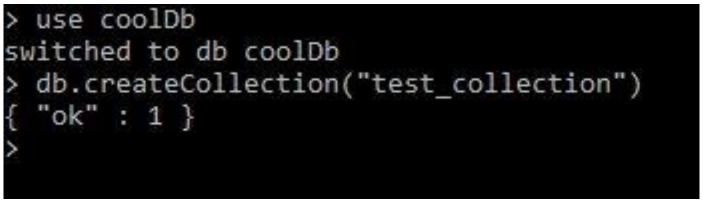
```
> use coolDB
switched to db coolDB
>
```

2. Drop a database



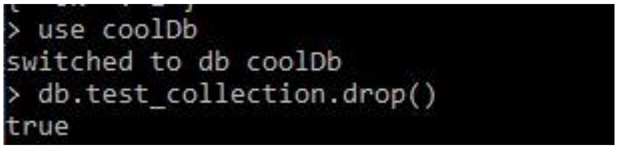
```
> use coolDB
switched to db coolDB
> db.dropDatabase()
{ "ok" : 1 }
>
```

3. Creating a collection



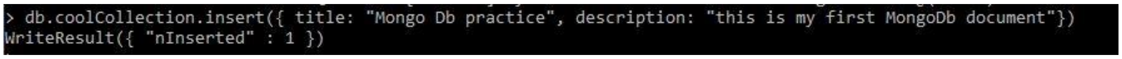
```
> use coolDb
switched to db coolDb
> db.createCollection("test_collection")
{ "ok" : 1 }
>
```

4. Dropping a collection



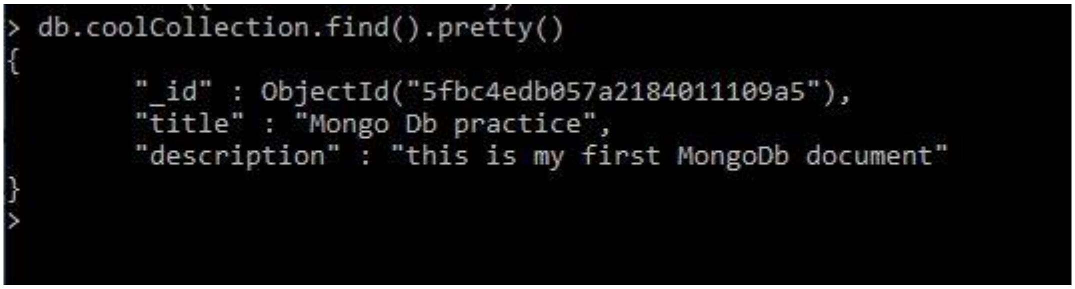
```
> use coolDb
switched to db coolDb
> db.test_collection.drop()
true
```

5. Insert a document



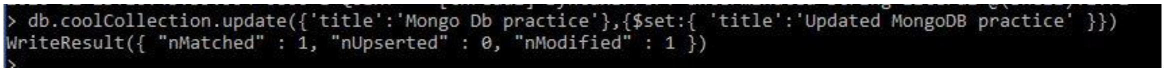
```
> db.coolCollection.insert({ title: "Mongo Db practice", description: "this is my first MongoDb document"})
WriteResult({ "nInserted" : 1 })
```

6. Query a document



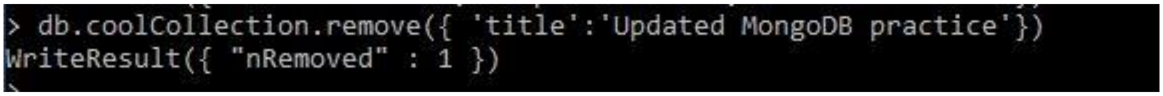
```
> db.coolCollection.find().pretty()
{
        "_id" : ObjectId("5fbc4edb057a2184011109a5"),
        "title" : "Mongo Db practice",
        "description" : "this is my first MongoDb document"
}
>
```

7. Update a document



```
> db.coolCollection.update({'title':'Mongo Db practice'},{$set:{ 'title':'Updated MongoDB practice' }})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

8. Delete a document



```
> db.coolCollection.remove({ 'title':'Updated MongoDB practice'})
WriteResult({ "nRemoved" : 1 })
```

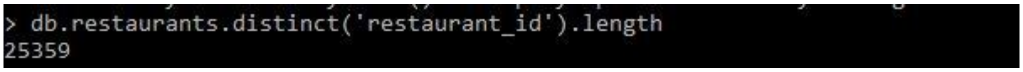
**Section 2:**

1. How many restaurants are there in this collection?

   **Query:**

   db.restaurants.distinct('restaurant_id').length

   **Screenshot:**

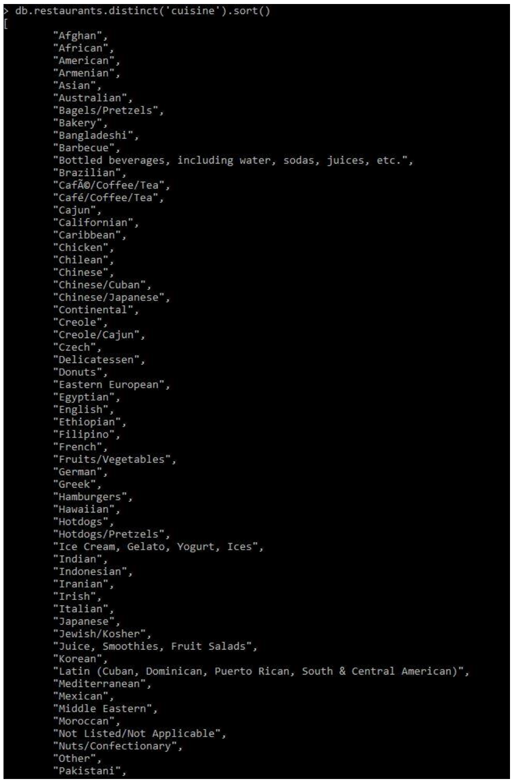
   

   ```
   > db.restaurants.distinct('restaurant_id').length
   25359
   ```

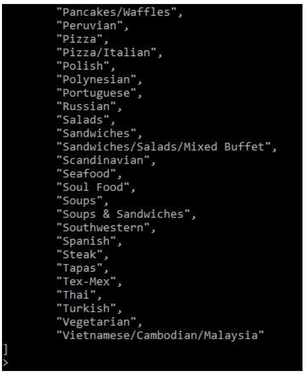
2. List in alphabetical order each different (distinct) cuisine represented in this collection

   **Query:** db.restaurants.distinct('cuisine').sort()

   **Screenshot:**

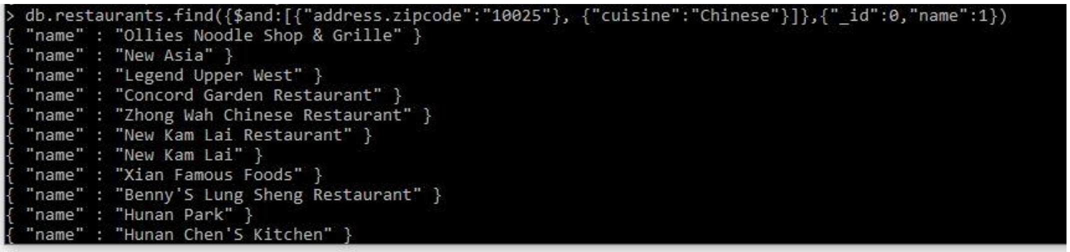
   

   ```
   > db.restaurants.distinct('cuisine').sort()
   [
           "Afghan",
           "African",
           "American",
           "Armenian",
           "Asian",
           "Australian",
           "Bagels/Pretzels",
           "Bakery",
           "Bangladeshi",
           "Barbecue",
           "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",
           "Jewish/Kosher",
           "Juice, Smoothies, Fruit Salads",
           "Korean",
           "Latin (Cuban, Dominican, Puerto Rican, South & Central American)",
           "Mediterranean",
           "Mexican",
           "Middle Eastern",
           "Moroccan",
           "Not Listed/Not Applicable",
           "Nuts/Confectionary",
           "Other",
           "Pakistani",
   ```

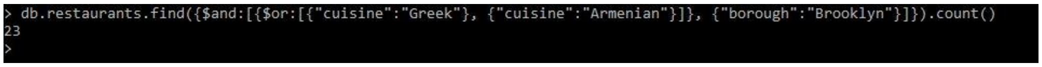
   ```
           "Pancakes/Waffles",
           "Peruvian",
           "Pizza",
           "Pizza/Italian",
           "Polish",
           "Polynesian",
           "Portuguese",
           "Russian",
           "Salads",
           "Sandwiches",
           "Sandwiches/Salads/Mixed Buffet",
           "Scandinavian",
           "Seafood",
           "Soul Food",
           "Soups",
           "Soups & Sandwiches",
           "Southwestern",
           "Spanish",
           "Steak",
           "Tapas",
           "Tex-Mex",
           "Thai",
           "Turkish",
           "Vegetarian",
           "Vietnamese/Cambodian/Malaysia"
   ]
   >
   ```

3. Return the name of all restaurants within the zipcode 10025 which serve Chinese cuisine. Return only the names of the restaurants.
   **Query:**
   db.restaurants.find({$and:[{"address.zipcode":"10025"},{"cuisine":"Chinese"}]},{"_id":0,"name":1})
   **Screenshot:**



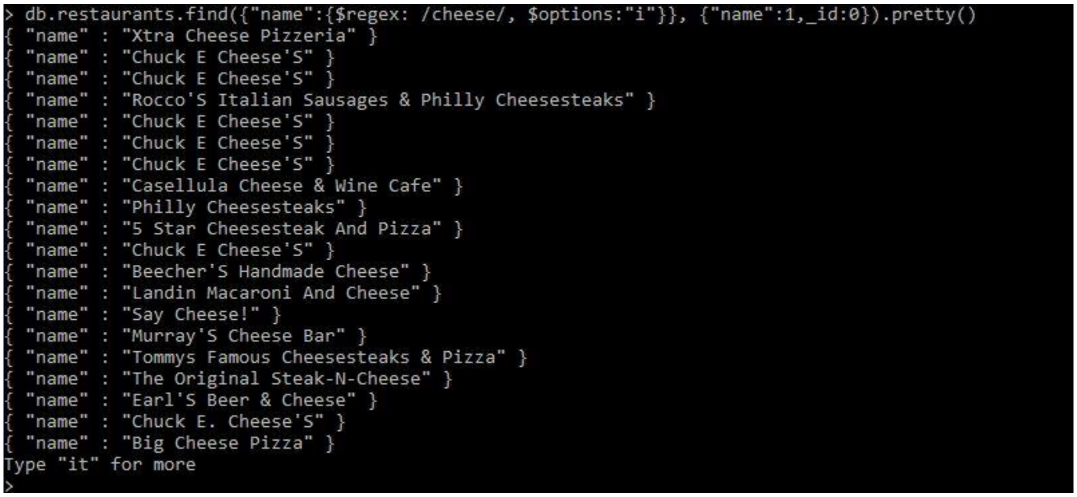
```
> db.restaurants.find({$and:[{"address.zipcode":"10025"}, {"cuisine":"Chinese"}]},{"_id":0,"name":1})
{ "name" : "Ollies Noodle Shop & Grille" }
{ "name" : "New Asia" }
{ "name" : "Legend Upper West" }
{ "name" : "Concord Garden Restaurant" }
{ "name" : "Zhong Wah Chinese Restaurant" }
{ "name" : "New Kam Lai Restaurant" }
{ "name" : "New Kam Lai" }
{ "name" : "Xian Famous Foods" }
{ "name" : "Benny'S Lung Sheng Restaurant" }
{ "name" : "Hunan Park" }
{ "name" : "Hunan Chen'S Kitchen" }
```

4. Return the count of restaurants in Brooklyn that serve either Greek or Armenian Food.
   **Query:** db.restaurants.find({$and:[{$or:[{"cuisine":"Greek"}, {"cuisine":"Armenian"}]}, {"borough":"Brooklyn"}]}).count()
   **Screenshot:**



```
> db.restaurants.find({$and:[{$or:[{"cuisine":"Greek"}, {"cuisine":"Armenian"}]}, {"borough":"Brooklyn"}]}).count()
23
>
```

5. Return a list of restaurants (names) which have the string “Cheese” in their name.
   **Query:** db.restaurants.find({"name":{$regex: /cheese/, $options:"i"}}, {"name":1,_id:0}).pretty()
   **Screenshot:**

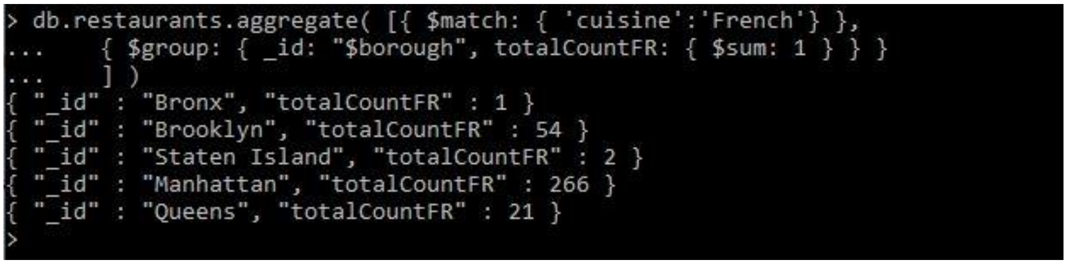


```
> db.restaurants.find({"name":{$regex: /cheese/, $options:"i"}}, {"name":1,_id:0}).pretty()
{ "name" : "Xtra Cheese Pizzeria" }
{ "name" : "Chuck E Cheese'S" }
{ "name" : "Chuck E Cheese'S" }
{ "name" : "Rocco'S Italian Sausages & Philly Cheesesteaks" }
{ "name" : "Chuck E Cheese'S" }
{ "name" : "Chuck E Cheese'S" }
{ "name" : "Chuck E Cheese'S" }
{ "name" : "Casellula Cheese & Wine Cafe" }
{ "name" : "Philly Cheesesteaks" }
{ "name" : "5 Star Cheesesteak And Pizza" }
{ "name" : "Chuck E Cheese'S" }
{ "name" : "Beecher'S Handmade Cheese" }
{ "name" : "Landin Macaroni And Cheese" }
{ "name" : "Say Cheese!" }
{ "name" : "Murray'S Cheese Bar" }
{ "name" : "Tommys Famous Cheesesteaks & Pizza" }
{ "name" : "The Original Steak-N-Cheese" }
{ "name" : "Earl'S Beer & Cheese" }
{ "name" : "Chuck E. Cheese'S" }
{ "name" : "Big Cheese Pizza" }
Type "it" for more
>
```

6. Return a list of boroughs ranked by the number of French restaurants in the borough. That is, for each borough, find how many restaurants serve French 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.)
   **Query:** db.restaurants.aggregate( [{ $match: { 'cuisine':'French'} },

{ $group: { _id: "$borough", totalCountFR: { $sum: 1 } } }
] )

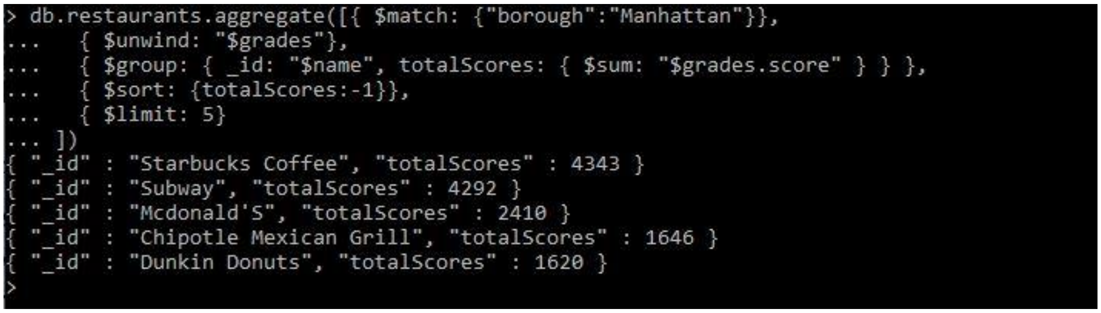
**Screenshot:**



```
> db.restaurants.aggregate( [{ $match: { 'cuisine':'French'} },
...    { $group: { _id: "$borough", totalCountFR: { $sum: 1 } } }
...    ] )
{ "_id" : "Bronx", "totalCountFR" : 1 }
{ "_id" : "Brooklyn", "totalCountFR" : 54 }
{ "_id" : "Staten Island", "totalCountFR" : 2 }
{ "_id" : "Manhattan", "totalCountFR" : 266 }
{ "_id" : "Queens", "totalCountFR" : 21 }
>
```

7. Find the top 5 Italian 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.)
**Query:** db.restaurants.aggregate([{ $match: {"borough":"Manhattan"}},
{ $unwind: "$grades"},
{ $group: { _id: "$name", totalScores: { $sum: "$grades.score" } } },
{ $sort: {totalScores:-1}},
{ $limit: 5}
])

**Screenshot:**



```
> db.restaurants.aggregate([{ $match: {"borough":"Manhattan"}},
...   { $unwind: "$grades"},
...   { $group: { _id: "$name", totalScores: { $sum: "$grades.score" } } },
...   { $sort: {totalScores:-1}},
...   { $limit: 5}
... ])
{ "_id" : "Starbucks Coffee", "totalScores" : 4343 }
{ "_id" : "Subway", "totalScores" : 4292 }
{ "_id" : "Mcdonald'S", "totalScores" : 2410 }
{ "_id" : "Chipotle Mexican Grill", "totalScores" : 1646 }
{ "_id" : "Dunkin Donuts", "totalScores" : 1620 }
>
```