Sasha Farhat
CSCI 3287

HW 06

SECTION 01

Launch

```
C:\Program Files\MongoDB\Server\4.4\bin>mongo
MongoDB shell version v4.4.2
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("0a0da5cc-0b55-4609-8d64-c77617b48efb") }
MongoDB server version: 4.4.2
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
        https://community.mongodb.com
---
The server generated these startup warnings when booting:
        2020-11-26T14:48:24.121-07:00: Access control is not enabled for the database. Read and write access to data and
 configuration is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
```

1.

```
> use new_mongo_db
switched to db new_mongo_db
>
```

2.

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

3.

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

4.

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

Sasha Farhat
CSCI 3287

5.

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

6.

```
> db.test_collection.find().pretty()
{
        "_id" : ObjectId("5fc02aff2e506c77aa0feb67"),
        "title" : "Mongo Db practice",
        "decsription" : "this is my first MongoDB document"
}
>
>
```

Without ".pretty()"

```
> db.test_collection.find()
{ "_id" : ObjectId("5fc02aff2e506c77aa0feb67"), "title" : "Mongo Db practice", "decsription" : "this is my first MongoDB
 document" }
>
```

7.

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

8.

```
> db.test_collection.remove({ status :"P"},1)
WriteResult({ "nRemoved" : 0 })
>
```

```
> db.test_collection.remove({ status :"P"})
WriteResult({ "nRemoved" : 0 })
>
```

```
> db.test_collection.remove({})
WriteResult({ "nRemoved" : 1 })
>
```

Sasha Farhat
CSCI 3287

Section 02

1. db.restaurants.count()

```
> db.restaurants.count()
25359
>
```

2.db.restaurants.distinct('cuisine').sort()

```
> 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",
```

Sasha Farhat
CSCI 3287

```
        "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"
]
```

Sasha Farhat
CSCI 3287

3. db.restaurants.find({'address.zipcode': "10025", 'cuisine': "Chinese"}, {name: 1, _id: 0}).pretty()

```
> db.restaurants.find({'address.zipcode':"10025", 'cuisine':"Chinese"}, {name:1, _id:0}).pretty()
{ "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" }
{ "name" : "Sugarcane Vietnamese Cooking" }
{ "name" : "Empire Garden" }
>
```

4.db.restaurants.find({$and:[{'borough': "Brooklyn"}, {$or:[{'cuisine': "Greek", 'cuisine': "American"}]}]}).count()

```
> db.restaurants.find({$and:[{'borough':"Brooklyn"}, {$or:[{'cuisine': "Greek", 'cuisine':"American"}]}]}).count()
1273
>
```

5.db.restaurants.find({'name': /.*Cheese.*/i}, {'name': 1, '_id': 0}).pretty()

```
> db.restaurants.find({'name':/.*Cheese.*/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
> it
{ "name" : "Sauce N Cheese 1" }
{ "name" : "5 Star Cheese Steak And Pizza" }
{ "name" : "Cheese Grille" }
{ "name" : "Pair Wine And Cheese" }
```

Sasha Farhat
CSCI 3287

6. db.restaurants.aggregate([{$match:{'cuisine': "French"}}, {$group:{_id:"$borough", count:{$sum: 1}}},{$sort:{count: -1}}])

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

7. db.restaurants.aggregate([{$match:{'cuisine':"Italian", 'borough':"Manhattan"}}, {$unwind:"$grades"}, {$group: {_id: "$name", count:{$sum:"$grades.score"}}}, {$sort:{count: -1}},{$limit: 5}])

```
> db.restaurants.aggregate([{$match:{'cuisine':"Italian", 'borough':"Manhattan"}}, {$unwind:"$grades"}, {$group:
{_id: "$name", count:{$sum:"$grades.score"}}}, {$sort:{count: -1}},{$limit: 5}])
{ "_id" : "Nanni Restaurant", "count" : 225 }
{ "_id" : "Bocca Di Bacco", "count" : 216 }
{ "_id" : "Lasagna Restaurant", "count" : 202 }
{ "_id" : "Coppola'S", "count" : 202 }
{ "_id" : "Gallo Nero", "count" : 180 }
>
```