

Jordan Mitchell

11/5/2024

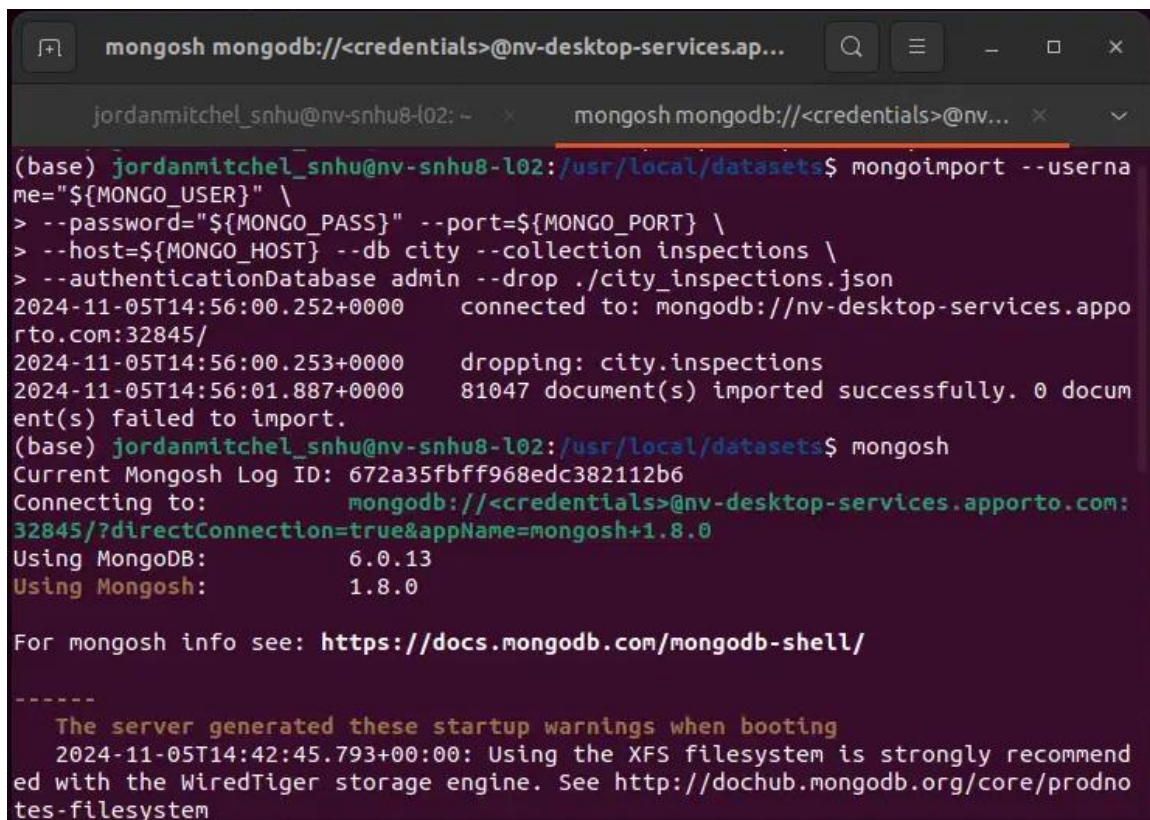
Professor Morris

Assignment: CRUD in MongoDB

1. Access the dataset for this assignment, *city_inspections.json*, in the */usr/local/datasets* directory in your Apporto environment. Using the *mongoimport* tool and documents found in the *city_inspections.json* file, load the database “city” into the “inspections” collection. Complete this task by typing the following commands in the Linux terminal to perform the import in the right directory:

Change into the Apporto directory with the data sets:

```
cd /usr/local/datasets/Use the mongo import utility to load the data set:mongoimport --
username="${MONGO_USER}" \
--password="${MONGO_PASS}" --port=${MONGO_PORT} \
--host=${MONGO_HOST} --db city --collection inspections \
--authenticationDatabase admin --drop ./city_inspections.json
```



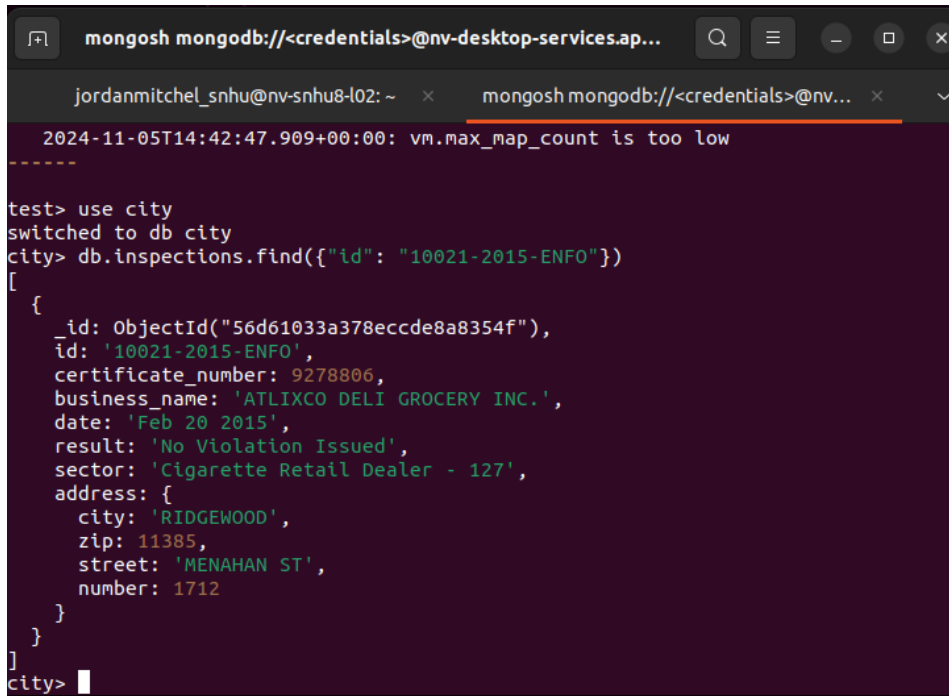
```
mongosh mongodb://<credentials>@nv-desktop-services.ap...
jordanmitchel_snhu@nv-snhu8-l02: ~ mongosh mongodb://<credentials>@nv...
(base) jordanmitchel_snhu@nv-snhu8-l02: /usr/local/datasets$ mongoimport --userna
me="${MONGO_USER}" \
> --password="${MONGO_PASS}" --port=${MONGO_PORT} \
> --host=${MONGO_HOST} --db city --collection inspections \
> --authenticationDatabase admin --drop ./city_inspections.json
2024-11-05T14:56:00.252+0000    connected to: mongodb://nv-desktop-services.appo
rto.com:32845/
2024-11-05T14:56:00.253+0000    dropping: city.inspections
2024-11-05T14:56:01.887+0000    81047 document(s) imported successfully. 0 docum
ent(s) failed to import.
(base) jordanmitchel_snhu@nv-snhu8-l02: /usr/local/datasets$ mongosh
Current Mongosh Log ID: 672a35fbff968edc382112b6
Connecting to:      mongodb://<credentials>@nv-desktop-services.apporto.com:
32845/?directConnection=true&appName=mongosh+1.8.0
Using MongoDB:      6.0.13
Using Mongosh:      1.8.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
The server generated these startup warnings when booting
2024-11-05T14:42:45.793+00:00: Using the XFS filesystem is strongly recommend
ed with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodno
tes-filesystem
```

2. Verify your load by “using” the “city” database, and issuing the following queries in the mongo shell:

a. `db.inspections.find({"id" : "10021-2015-ENFO"})`



The screenshot shows a terminal window with the MongoDB shell. The prompt is `test>`. The user enters `use city`, and the prompt changes to `city>`. The user then enters the query `db.inspections.find({"id" : "10021-2015-ENFO"})`. The output is a JSON array containing one document. The document has the following fields: `_id` (ObjectId), `id` ('10021-2015-ENFO'), `certificate_number` (9278806), `business_name` ('ATLIXCO DELI GROCERY INC.'), `date` ('Feb 20 2015'), `result` ('No Violation Issued'), `sector` ('Cigarette Retail Dealer - 127'), and `address` (an object with `city` 'RIDGEWOOD', `zip` 11385, `street` 'MENAHAN ST', and `number` 1712).

```
mongosh mongodb://<credentials>@nv-desktop-services.ap...
jordanmitchel_snhu@nv-snhu8-l02: ~ × mongosh mongodb://<credentials>@nv... ×
2024-11-05T14:42:47.909+00:00: vm.max_map_count is too low
-----
test> use city
switched to db city
city> db.inspections.find({"id" : "10021-2015-ENFO"})
[
  {
    _id: ObjectId("56d61033a378eccde8a8354f"),
    id: '10021-2015-ENFO',
    certificate_number: 9278806,
    business_name: 'ATLIXCO DELI GROCERY INC.',
    date: 'Feb 20 2015',
    result: 'No Violation Issued',
    sector: 'Cigarette Retail Dealer - 127',
    address: {
      city: 'RIDGEWOOD',
      zip: 11385,
      street: 'MENAHAN ST',
      number: 1712
    }
  }
]
city>
```

b. `db.inspections.find({"result":"Out of Business"},{"business_name":1}).limit(10)`

```

mongosh mongodb://<credentials>@nv-desktop-services.ap...
jordanmitchel_snhu@nv-snhu8-l02: ~ x mongosh mongodb://<credentials>@nv... x
city> db.inspections.find({"result": "Out of Business"}, {"business_name": 1}).limit(10)
[
  {
    _id: ObjectId("56d61033a378eccde8a835e9"),
    business_name: 'COPY SOLUTIONS INC'
  },
  {
    _id: ObjectId("56d61033a378eccde8a8361e"),
    business_name: 'PARITA CARDS AND GIFTS INC.'
  },
  {
    _id: ObjectId("56d61033a378eccde8a83628"),
    business_name: 'L.A. PAWNSHOP CORP.'
  },
  {
    _id: ObjectId("56d61033a378eccde8a83656"),
    business_name: 'GRAND ARMY CAR PARK, LLC'
  },
  {
    _id: ObjectId("56d61033a378eccde8a8366d"),
    business_name: 'CAR APPROVAL CORP.'
  },
  {

```

3. Using the appropriate commands in the mongo shell, insert a document to the database named “city” within the collection named “inspections.” Use the following key-value pairs as data for your document. Be sure to insert the address as a subdocument and use the JavaScript function Date() for “Today’s date.” Verify your database creation and insertion using the findOne() function in the mongo shell. Provide a screenshot as evidence.

Key	Value
id	"20032-2020-ACME"
certificate_number	9998888
business_name	"ACME Explosives"
date	Today's date
result	"Business Padlocked"
sector	"Explosive Retail Dealer-999"
address	number -> 1721 street -> Boom Road city -> BRONX zip -> 10463

```
mongosh mongodb://<credentials>@nv-desktop-services.apporto.com:32845/?directCo...
jordanmitchel_snhu@nv-snhu8-l02: ~  mongosh mongodb://<credentials>@nv-desktop-ser...
switched to db city
city> db.inspections.insertOne({
...   "id": "20032-2020-ACME",
...   "certificate_number": 9998888,
...   "business_name": "ACME Explosives",
...   "date": new Date("Nov 05 2024"),
...   "result": "Business Padlocked",
...   "sector": "Explosive Retail Dealer-999",
...   "address": {
...     "number": 1721,
...     "street": "Boom Road",
...     "city": "BRONX",
...     "zip": 10463,
...   }
... })
{
  acknowledged: true,
  insertedId: ObjectId("672bb5cc642c4fd78c05b5bf")
}
city> db.inspections.findOne({"id": "20032-2020-ACME"})
{
  _id: ObjectId("672bb5cc642c4fd78c05b5bf"),
  id: '20032-2020-ACME',
  certificate_number: 9998888,
  business_name: 'ACME Explosives',
  date: ISODate("2024-11-05T00:00:00.000Z"),
  result: 'Business Padlocked',
  sector: 'Explosive Retail Dealer-999',
  address: { number: 1721, street: 'Boom Road', city: 'BRONX', zip: 10463 }
}
city> 
```

4, Answer the following questions using MongoDB queries.

What is the distinct list of inspection “sector” in the current inspections collection? How many are in the list? Do not count by hand.

What is the difference in the date data type for the business named “AUSTIN 2012” versus your business document insertion of “Acme Explosives”?

How many businesses have a “Violation Issued”? (See Value column above.)

```
mongosh mongodb://<credentials>@nv-desktop-services.apporto.com:32845/?directC...
mongosh mongodb://<credentials>@nv-desktop-ser... x jordanmitchel_snhu@nv-snhu8-l02: ~ x v
tion=true&appName=mongosh+1.8.0
Using MongoDB: 6.0.13
Using Mongosh: 1.8.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
The server generated these startup warnings when booting
2024-11-05T14:42:45.793+00:00: Using the XFS filesystem is strongly recommended with the WiredTi
ger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2024-11-05T14:42:47.909+00:00: Failed to read /sys/kernel/mm/transparent_hugepage/defrag
2024-11-05T14:42:47.909+00:00: vm.max_map_count is too low
-----

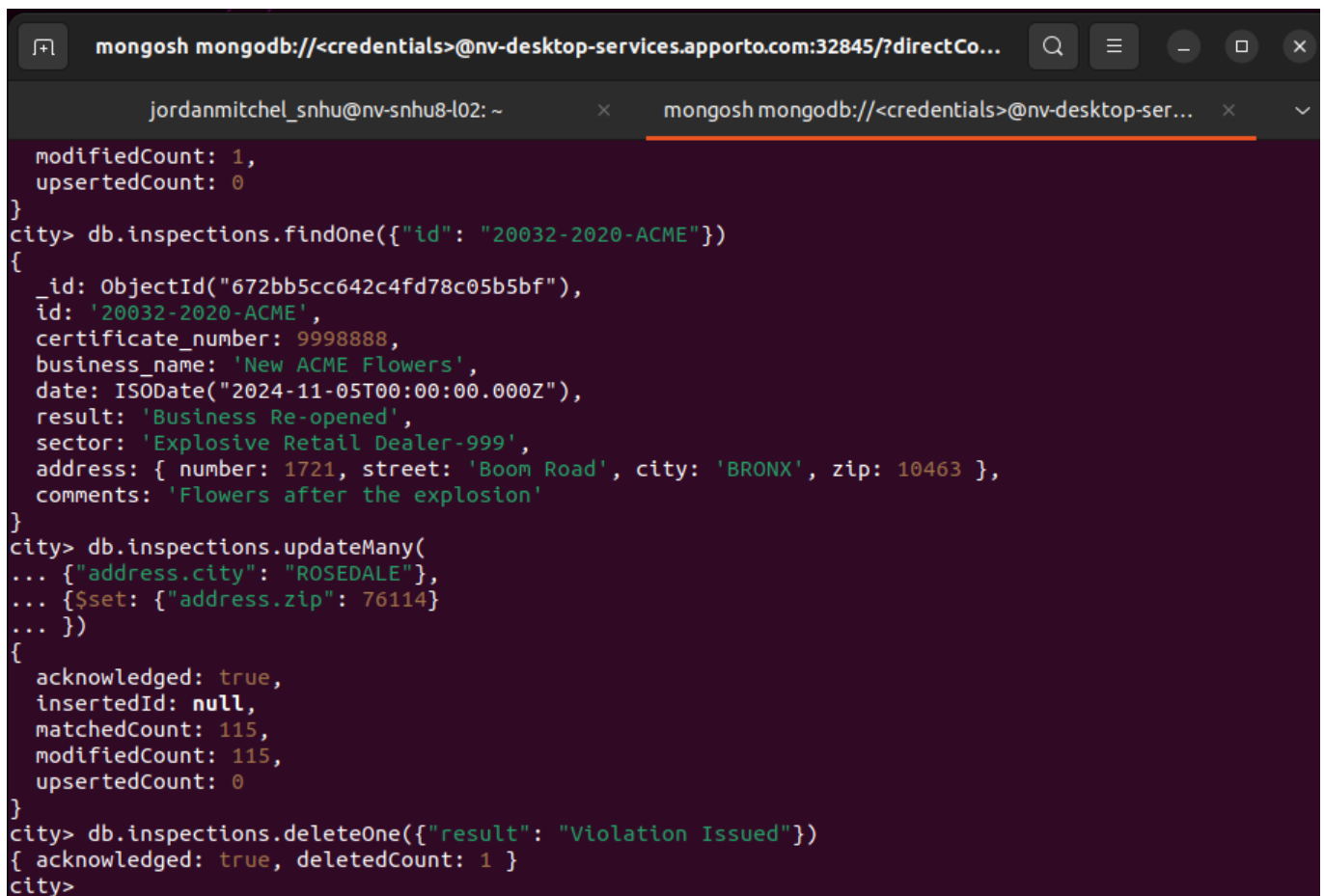
test> use city
switched to db city
city> db.inspections.distinct("sector").length
87
city> db.inspections.find({"business_name": "AUSTIN 2012"}, {"date": 1})
[ { _id: ObjectId("56d61035a378eccde8a971c3"), date: 'Jan 22 2016' } ]
city> db.inspections.find({"business_name": "ACME Explosives"}, {"date": 1})
[
  {
    _id: ObjectId("672a6717c727c0e65443285f"),
    date: ISODate("2024-11-05T00:00:00.000Z")
  }
]
city> db.inspections.countDocuments({"result": "Violation Issued"})
13823
city> 
```

5. Using the appropriate command in the mongo shell, update the document with the ID “20032-2020-ACME” in the collection “inspections” in the database “city” with the information below.

```
mongosh mongodb://<credentials>@nv-desktop-services.apporto.com:32845/?directCo...
jordanmitchel_snhu@nv-snhu8-l02: ~
city> db.inspections.countDocuments({"result": "Violation Issued"})
13823
city> db.inspections.updateOne(
... {"id": "20032-2020-ACME"},
... {
... $set: {
... "business_name": "New ACME Flowers",
... "result": "Business Re-opened",
... "comments": "Flowers after the explosion"
... }
... })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
city> db.inspections.findOne({"id": "20032-2020-ACME"})
{
  _id: ObjectId("672bb5cc642c4fd78c05b5bf"),
  id: '20032-2020-ACME',
  certificate_number: 9998888,
  business_name: 'New ACME Flowers',
  date: ISODate("2024-11-05T00:00:00.000Z"),
  result: 'Business Re-opened',
  sector: 'Explosive Retail Dealer-999',
  address: { number: 1721, street: 'Boom Road', city: 'BRONX', zip: 10463 },
  comments: 'Flowers after the explosion'
}
city>
```

6. Using the database “city” with documents found in the “inspections” collection, perform the tasks listed below. Verify by providing **screenshots** of the results as evidence.

- a. **Update all the documents** that contain the key-value pair "city":"ROSEDALE" in the address subdocument by changing the zip code in the address subdocument to "76114".
- b. **Remove the first document** with the key-value pair "result":"Violation Issued."

A screenshot of a MongoDB shell session. The browser window title is 'mongosh mongodb://<credentials>@nv-desktop-services.apporto.com:32845/?directCo...'. The address bar shows 'jordanmitchel_snhu@nv-snhu8-l02: ~'. The terminal window shows the following commands and results:

```
modifiedCount: 1,
upsertedCount: 0
}
city> db.inspections.findOne({"id": "20032-2020-ACME"})
{
  _id: ObjectId("672bb5cc642c4fd78c05b5bf"),
  id: '20032-2020-ACME',
  certificate_number: 9998888,
  business_name: 'New ACME Flowers',
  date: ISODate("2024-11-05T00:00:00.000Z"),
  result: 'Business Re-opened',
  sector: 'Explosive Retail Dealer-999',
  address: { number: 1721, street: 'Boom Road', city: 'BRONX', zip: 10463 },
  comments: 'Flowers after the explosion'
}
city> db.inspections.updateMany(
... {"address.city": "ROSEDALE"},
... {$set: {"address.zip": 76114}}
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 115,
  modifiedCount: 115,
  upsertedCount: 0
}
city> db.inspections.deleteOne({"result": "Violation Issued"})
{ acknowledged: true, deletedCount: 1 }
city>
```