

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
```

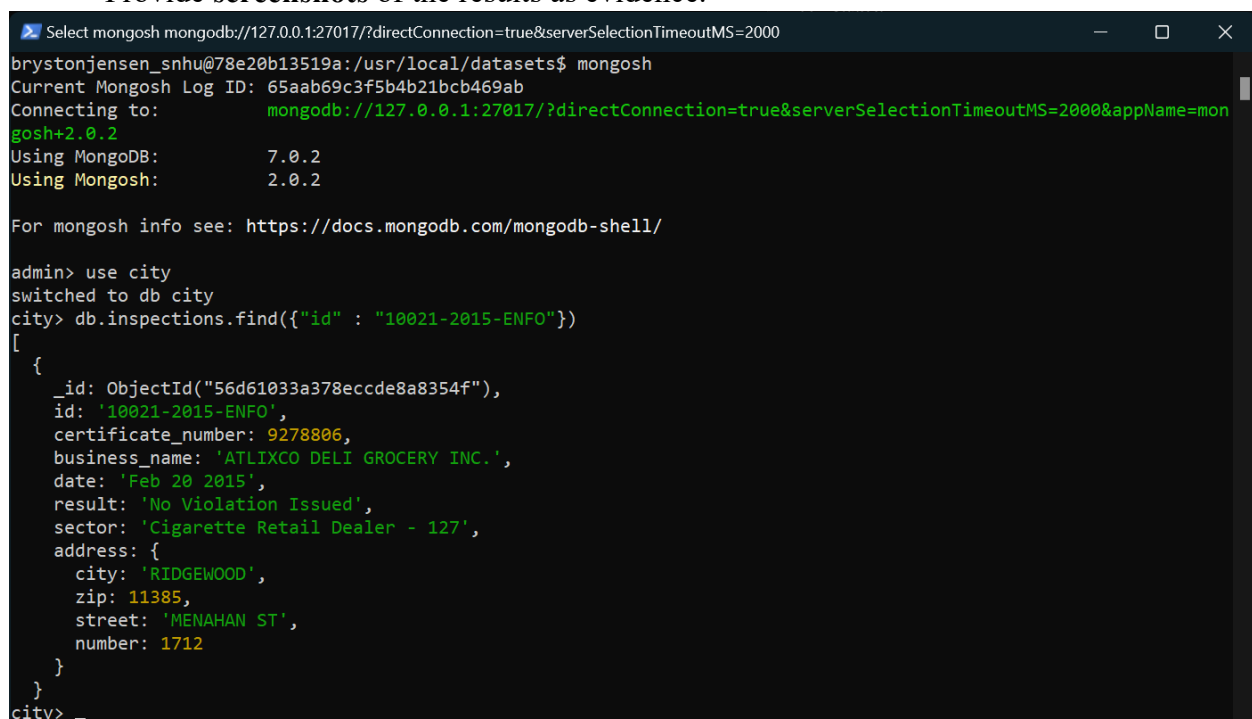
**Tip:** In any Linux systems, commands must be exact and use proper syntax and case sensitivity.

```
brystonjensen_snhu@78e20b13519a: /usr/local/datasets$ mongoimport --username="${MONGO_USER}" --password="${MONGO_PASS}" --port=${MONGO_PORT} --host=${MONGO_HOST} --db=city --collection inspections --authenticationDatabase admin --drop ./city_inspections.json
2024-01-19T17:50:51.555+0000    connected to: mongodb://localhost:27017/
2024-01-19T17:50:51.555+0000    dropping: city.inspections
2024-01-19T17:50:52.682+0000    81047 document(s) imported successfully. 0 document(s) failed to import.
```

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"})`

Provide **screenshots** of the results as evidence.



```
Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
brystonjensen_snhu@78e20b13519a: /usr/local/datasets$ mongosh
Current Mongosh Log ID: 65aab69c3f5b4b21bcb469ab
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.0.2
Using MongoDB:      7.0.2
Using Mongosh:       2.0.2

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

admin> 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://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
city> db.inspections.find({"result" : "Out of Business"}, {"business_name":1}).limit(10)
[
  {
    _id: ObjectId("56d61033a378eccde8a835e9"),
    business_name: 'COPY SOLUTIONS INC'
  },
  {
    _id: ObjectId("56d61033a378eccde8a83628"),
    business_name: 'L.A. PAWNSHOP CORP.'
  },
  {
    _id: ObjectId("56d61033a378eccde8a8361e"),
    business_name: 'PARITA CARDS AND GIFTS INC.'
  },
  {
    _id: ObjectId("56d61033a378eccde8a83656"),
    business_name: 'GRAND ARMY CAR PARK, LLC'
  },
  {
    _id: ObjectId("56d61033a378eccde8a8366d"),
    business_name: 'CAR APPROVAL CORP.'
  },
  {
    _id: ObjectId("56d61033a378eccde8a8366f"),
    business_name: 'CHAMPION CONTRACTING NY CORP.'
  },
  {
    _id: ObjectId("56d61033a378eccde8a83694"),
    business_name: 'CITYLINE SPORTS INC'
  },
  {
    _id: ObjectId("56d61033a378eccde8a83698"),
    business_name: 'PUEBLA BEST FLOWERS, INC'
  },
  {
    _id: ObjectId("56d61033a378eccde8a836b6"),
    business_name: 'WALYED DELI GROCERY INC'
  },
  {
    _id: ObjectId("56d61033a378eccde8a836c1"),
    business_name: 'B & D HOLDING GROUP INC. BULLFROG REMODELING'
  }
]
city> 
```

- 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.

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

Be sure to insert the address as a subdocument and use the JavaScript function **Date()** for “Today’s date.”

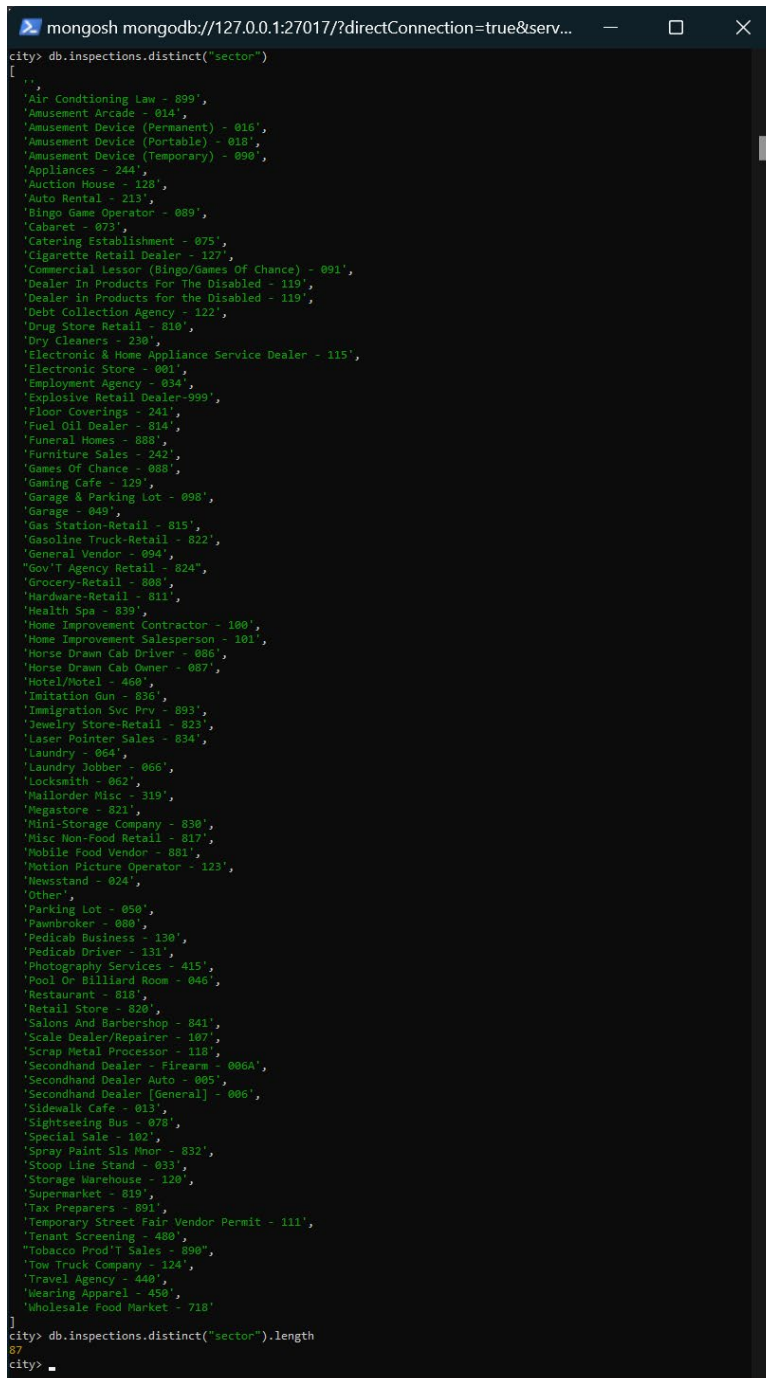
```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
city> db.inspections.insertOne({"id":"20032-2020-ACME", "certificate_number":"9998888", "business_name":"ACME Explosives", "date":new Date(), "result":"Business Padlocked", "sector":"Explosive Retail Dealer-999", "address":{"number":"1721", "street":"Boom Road", "city":"BRONX", "zip":"10463"}})
{
  acknowledged: true,
  insertedId: ObjectId("65aabd903f5b4b21bcb469ac")
}
city>
```

Verify your database creation and insertion using the **findOne()** function in the mongo shell. Provide a **screenshot** as evidence.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
city> db.inspections.findOne({"id":"20032-2020-ACME"})
{
  _id: ObjectId("65aabd903f5b4b21bcb469ac"),
  id: '20032-2020-ACME',
  certificate_number: '9998888',
  business_name: 'ACME Explosives',
  date: ISODate("2024-01-19T18:21:04.583Z"),
  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**.  
Verify by providing **screenshots** of the queries and results as evidence.

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



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serv...
city> db.inspections.distinct("sector")
[
  '',
  'Air Conditioning Law - 899',
  'Amusement Arcade - 014',
  'Amusement Device (Permanent) - 016',
  'Amusement Device (Portable) - 018',
  'Amusement Device (Temporary) - 090',
  'Appliances - 244',
  'Auction House - 120',
  'Auto Rental - 213',
  'Bingo Game Operator - 089',
  'Cabaret - 073',
  'Catering Establishment - 075',
  'Cigarette Retail Dealer - 127',
  'Commercial Lessor (Bingo/Games Of Chance) - 091',
  'Dealer In Products For The Disabled - 119',
  'Dealer In Products For The Disabled - 119',
  'Debt Collection Agency - 122',
  'Drug Store Retail - 810',
  'Dry Cleaners - 230',
  'Electronic & Home Appliance Service Dealer - 115',
  'Electronic Store - 001',
  'Employment Agency - 034',
  'Explosive Retail Dealer-999',
  'Floor Coverings - 241',
  'Fuel Oil Dealer - 814',
  'Funeral Homes - 880',
  'Furniture Sales - 242',
  'Games Of Chance - 088',
  'Gaming Cafe - 129',
  'Garage & Parking Lot - 098',
  'Garage - 049',
  'Gas Station-Retail - 815',
  'Gasoline Truck-Retail - 822',
  'General Vendor - 094',
  'Gov'T Agency Retail - 824',
  'Grocery-Retail - 800',
  'Hardware-Retail - 811',
  'Health Spa - 839',
  'Home Improvement Contractor - 100',
  'Home Improvement Salesperson - 101',
  'Horse Drawn Cab Driver - 086',
  'Horse Drawn Cab Owner - 087',
  'Hotel/Hotel - 460',
  'Imitation Gun - 836',
  'Immigration Svc Pny - 893',
  'Jewelry Store-Retail - 823',
  'Laser Pointer Sales - 834',
  'Laundry - 064',
  'Laundry Jobber - 066',
  'Locksmith - 062',
  'Mailorder Misc - 319',
  'Magazine - 821',
  'Mini-Storage Company - 830',
  'Misc Non-Food Retail - 817',
  'Mobile Food Vendor - 881',
  'Motion Picture Operator - 123',
  'Newsstand - 024',
  'Other',
  'Parking Lot - 050',
  'Pawnbroker - 080',
  'Pedicab Business - 130',
  'Pedicab Driver - 131',
  'Photography Services - 415',
  'Pool Or Billiard Room - 046',
  'Restaurant - 818',
  'Retail Store - 820',
  'Salons And Barbershop - 841',
  'Scale Dealer/Repairer - 107',
  'Scrap Metal Processor - 118',
  'Secondhand Dealer - Firearm - 006A',
  'Secondhand Dealer Auto - 005',
  'Secondhand Dealer [General] - 006',
  'Sidewalk Cafe - 013',
  'Sightseeing Bus - 078',
  'Special Sale - 102',
  'Spray Paint Sls Mnor - 832',
  'Stoop Line Stand - 033',
  'Storage Warehouse - 120',
  'Supermarket - 810',
  'Tax Preparers - 891',
  'Temporary Street Fair Vendor Permit - 111',
  'Tenant Screening - 480',
  'Tobacco Prod'T Sales - 890',
  'Tow Truck Company - 124',
  'Travel Agency - 440',
  'Wearing Apparel - 450',
  'Wholesale Food Market - 718'
]
city> db.inspections.distinct("sector").length
87
city>
```

There is a list of 87 distinct values of “sector” in the inspections collection.

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

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
city> db.inspections.findOne({business_name:"AUSTIN 2012"})
{
  _id: ObjectId("56d61035a378eccde8a971c3"),
  id: '6658-2016-ENFO',
  certificate_number: 9327364,
  business_name: 'AUSTIN 2012',
  date: 'Jan 22 2016',
  result: 'Out of Business',
  sector: 'Grocery-Retail - 808',
  address: { city: 'NEW YORK', zip: 10016, street: 'PARK AVE', number: 10 }
}
city>
```

The difference is that the date in this one is very simple, and possibly just entered as a string. For the one I created, it is created using the Date() function and has more data for it including the time it was created. I am just noticing though that the date seems to be in a weird format on the document I created. I am not sure how that happened.

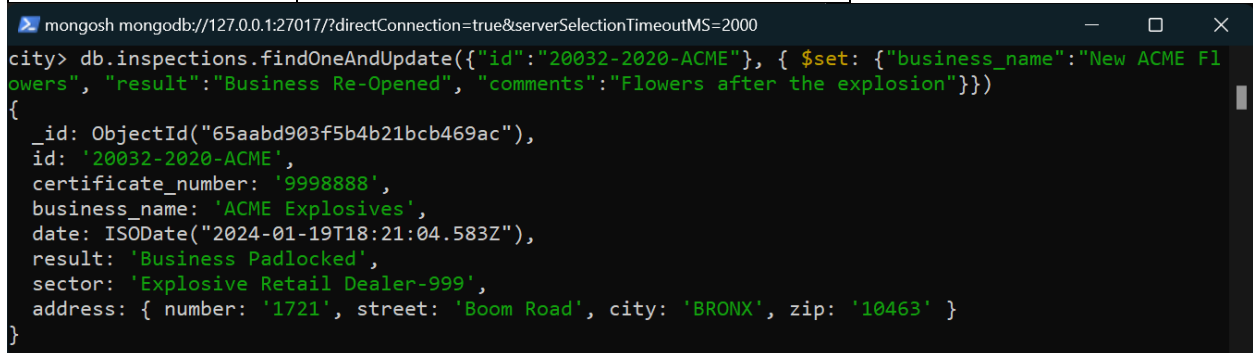
- c. How many businesses have a “Violation Issued”? (See **Value** column above.)

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
city> db.inspections.countDocuments({result:"Violation Issued"})
13823
city> _
```

There are 13,823 businesses that have a Violation Issued.

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.

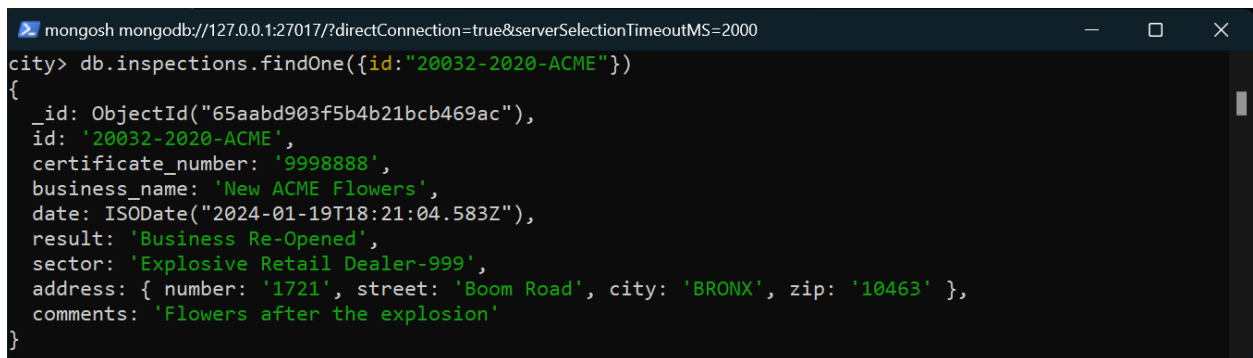
Key	Value
business_name	"New ACME Flowers"
Result	"Business Re-opened"
comments	"Flowers after the explosion"



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
city> db.inspections.findOneAndUpdate({"id":"20032-2020-ACME"}, { $set: {"business_name":"New ACME Flowers", "result":"Business Re-Opened", "comments":"Flowers after the explosion"}})
{
  _id: ObjectId("65aabd903f5b4b21bcb469ac"),
  id: '20032-2020-ACME',
  certificate_number: '9998888',
  business_name: 'ACME Explosives',
  date: ISODate("2024-01-19T18:21:04.583Z"),
  result: 'Business Padlocked',
  sector: 'Explosive Retail Dealer-999',
  address: { number: '1721', street: 'Boom Road', city: 'BRONX', zip: '10463' }
}
```

**findOneAndUpdate()** finds the first document that matches the search and updates it THEN displays the document BEFORE it was updated.

Verify your database update using the appropriate **find()** function in the mongo shell. Provide a screenshot as evidence.

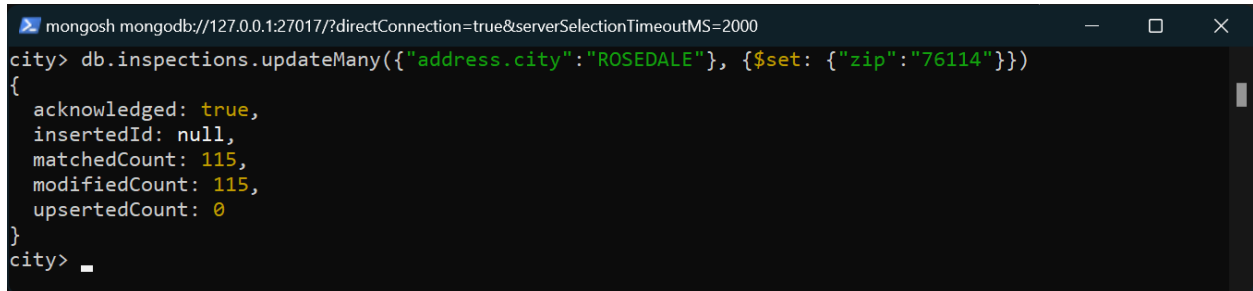


```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
city> db.inspections.findOne({"id":"20032-2020-ACME"})
{
  _id: ObjectId("65aabd903f5b4b21bcb469ac"),
  id: '20032-2020-ACME',
  certificate_number: '9998888',
  business_name: 'New ACME Flowers',
  date: ISODate("2024-01-19T18:21:04.583Z"),
  result: 'Business Re-Opened',
  sector: 'Explosive Retail Dealer-999',
  address: { number: '1721', street: 'Boom Road', city: 'BRONX', zip: '10463' },
  comments: 'Flowers after the explosion'
}
```

Displaying the updated document.

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.

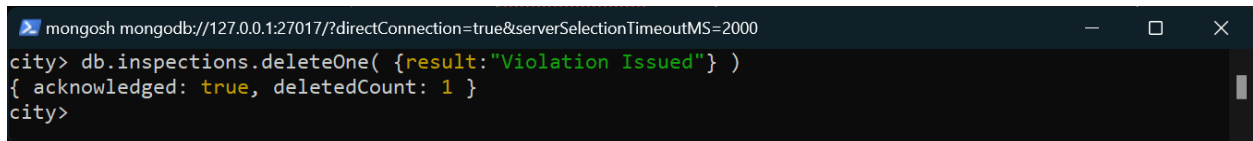
- a. **Update all the documents** that contain the key-value pair "city":"ROSEDALE" in the address subdocument by changing the zip code in the address subdocument to "76114".



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
city> db.inspections.updateMany({"address.city":"ROSEDALE"}, {$set: {"zip":"76114"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 115,
  modifiedCount: 115,
  upsertedCount: 0
}
city>
```

I updated 115 documents to the new zipcode.

- b. **Remove the first document** with the key-value pair "result":"Violation Issued."



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
city> db.inspections.deleteOne( {result:"Violation Issued"} )
{ acknowledged: true, deletedCount: 1 }
city>
```

The deleteOne() function removes the first document it finds that meets the criteria. As you can see, it only deleted one.