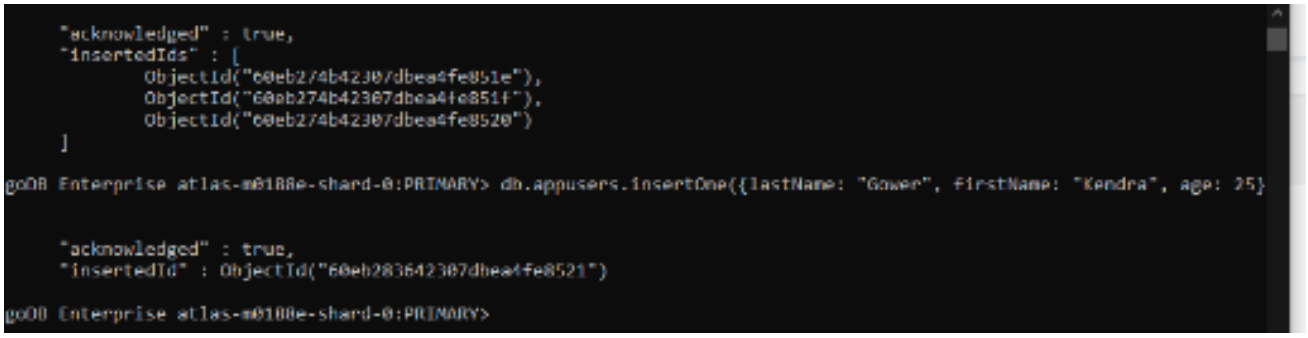
Part 1

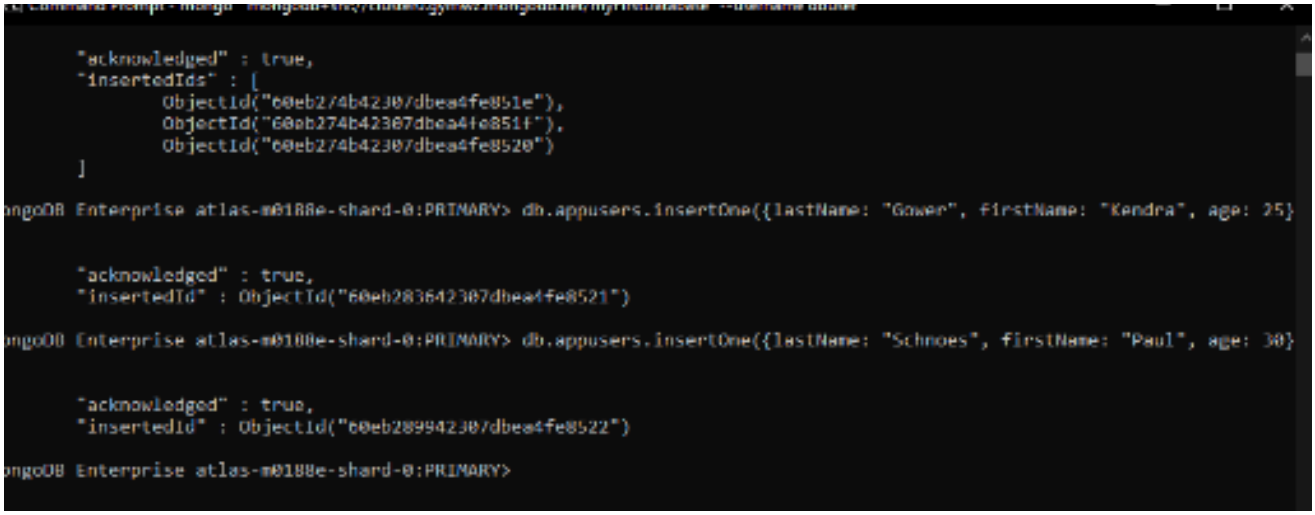
1. Create three more users of your choosing, using the `insertOne()` query, and add them to the `appusers` collection.
    - Include at least the `firstName`, `lastName`, and `age` fields. Feel free to also include any other fields you think would be useful data for an app user.

      Caution!

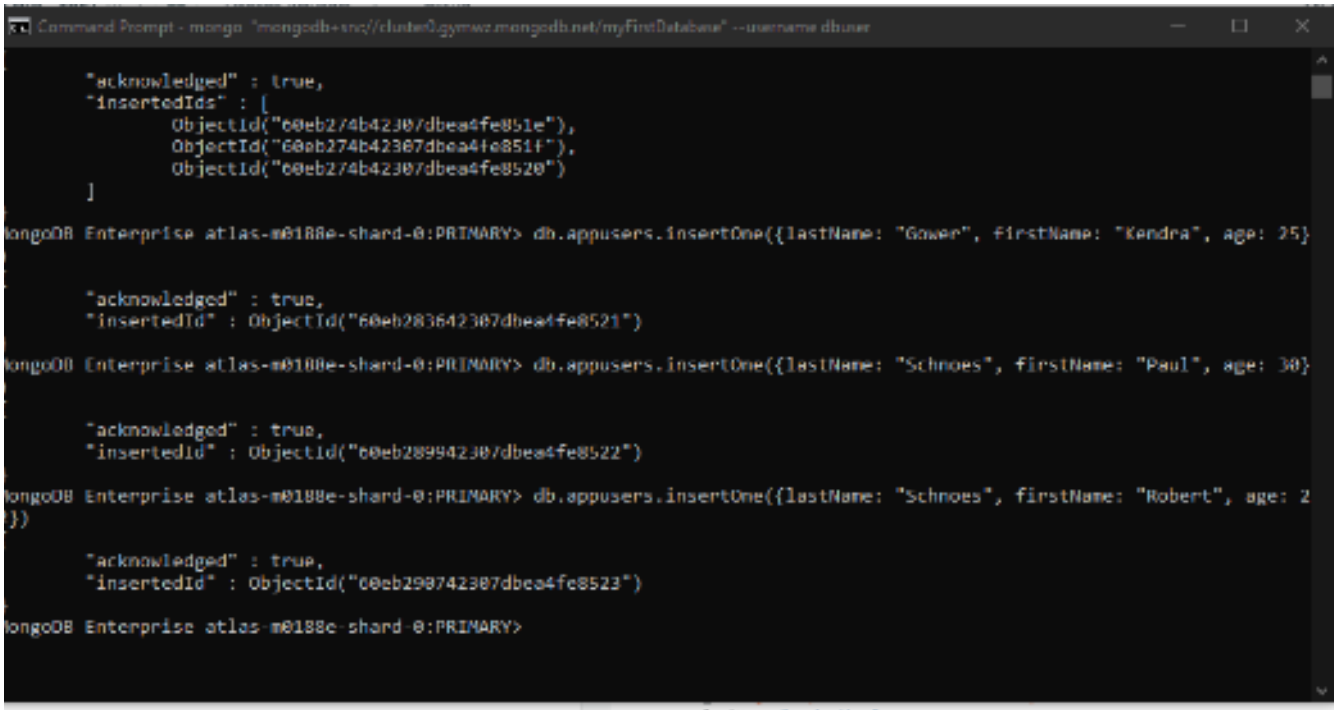
      Please take a screenshot of each query you run (since you are inserting each user one-by-one using `insertOne()`, you will run at least three queries) and save it to your `NoSQL-HandsOn1` text document.
2. Next, run a basic `find()` query to see all of your documents within your database through the Mongo Shell.



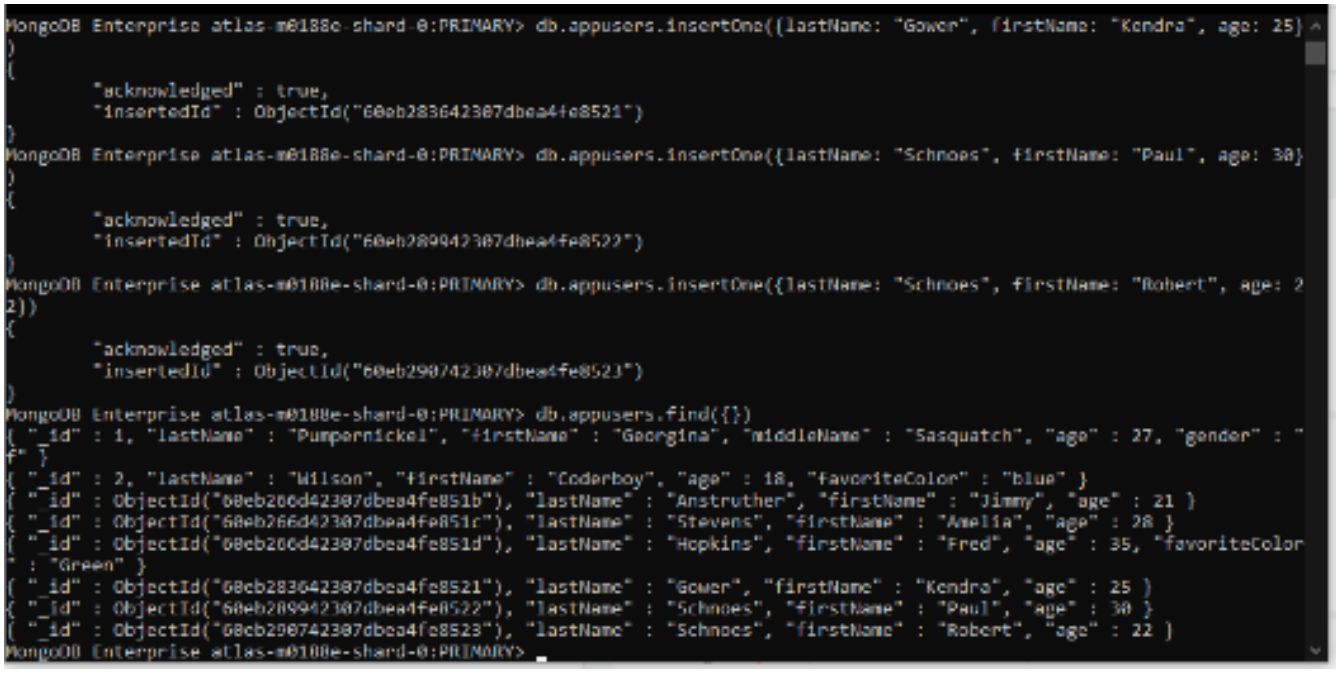
User 1: db.appusers.insertOne({lastName: "Gower", firstName: "Kendra", age: 25})



User 2: db.appusers.insertOne({lastName: "Schnoes", firstName: "Paul", age: 30})



User 3: db.appusers.insertOne({lastName: "Schnoes", firstName: "Robert", age: 22})



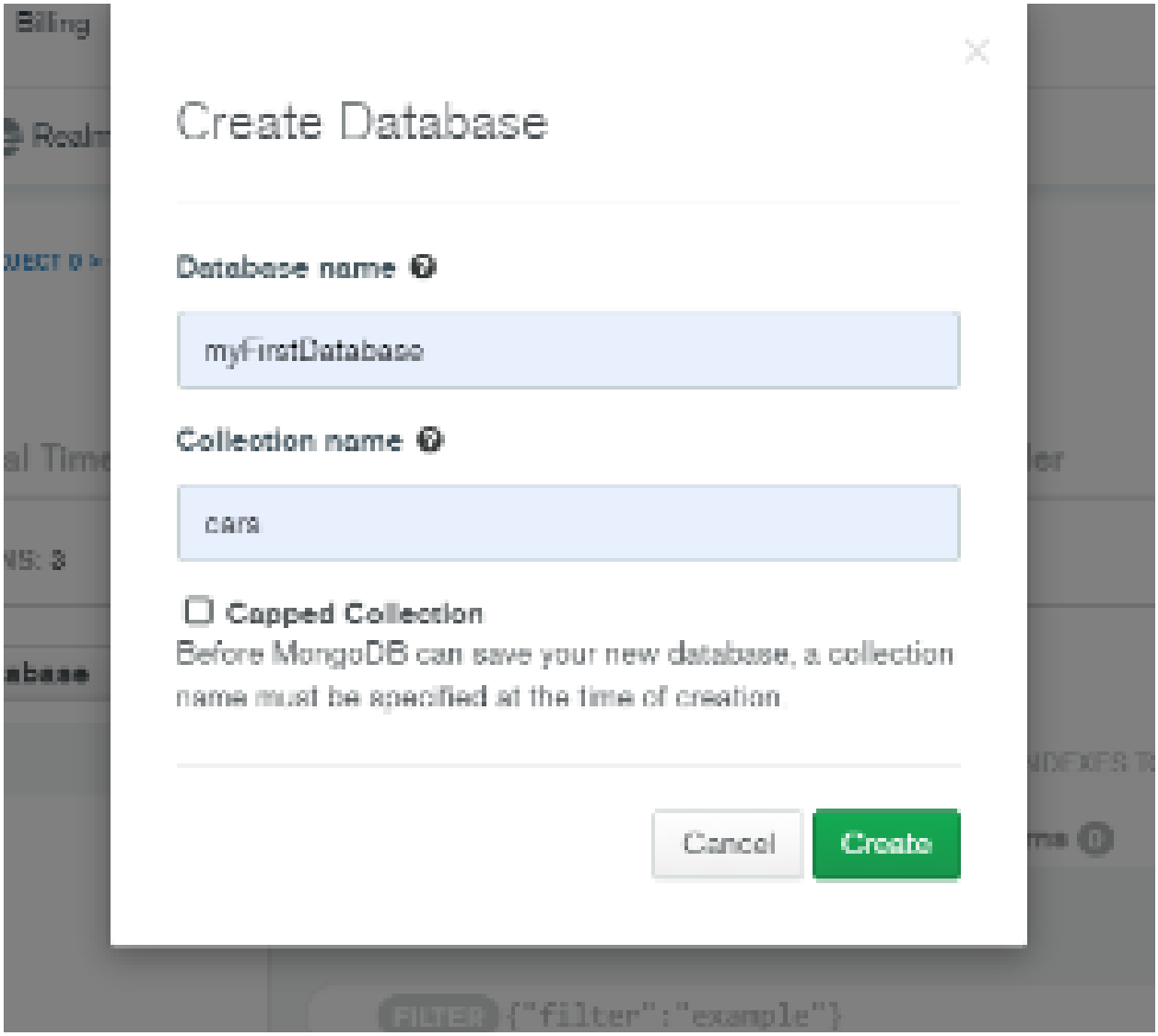
Find: db.appusers.find({})

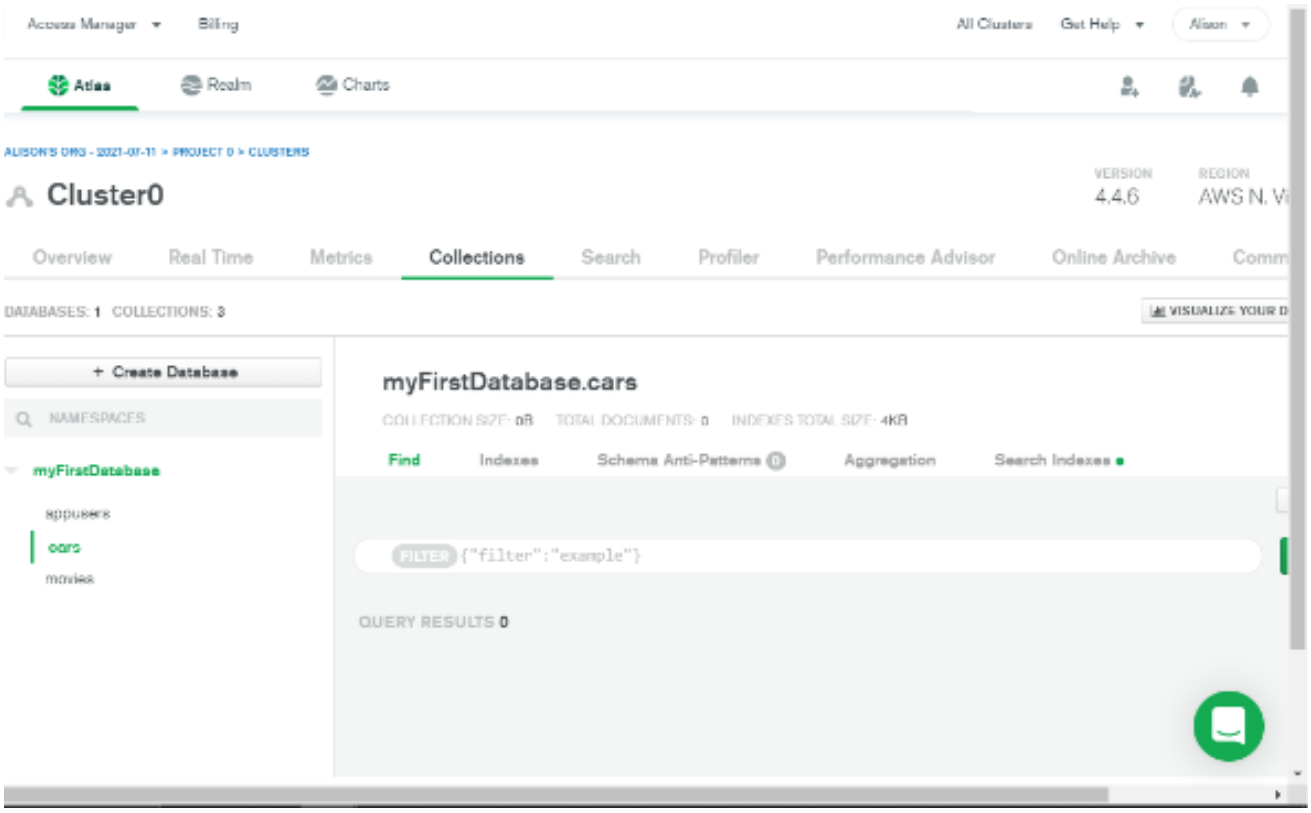
Part 2

1. Create a new collection within your database through Atlas.

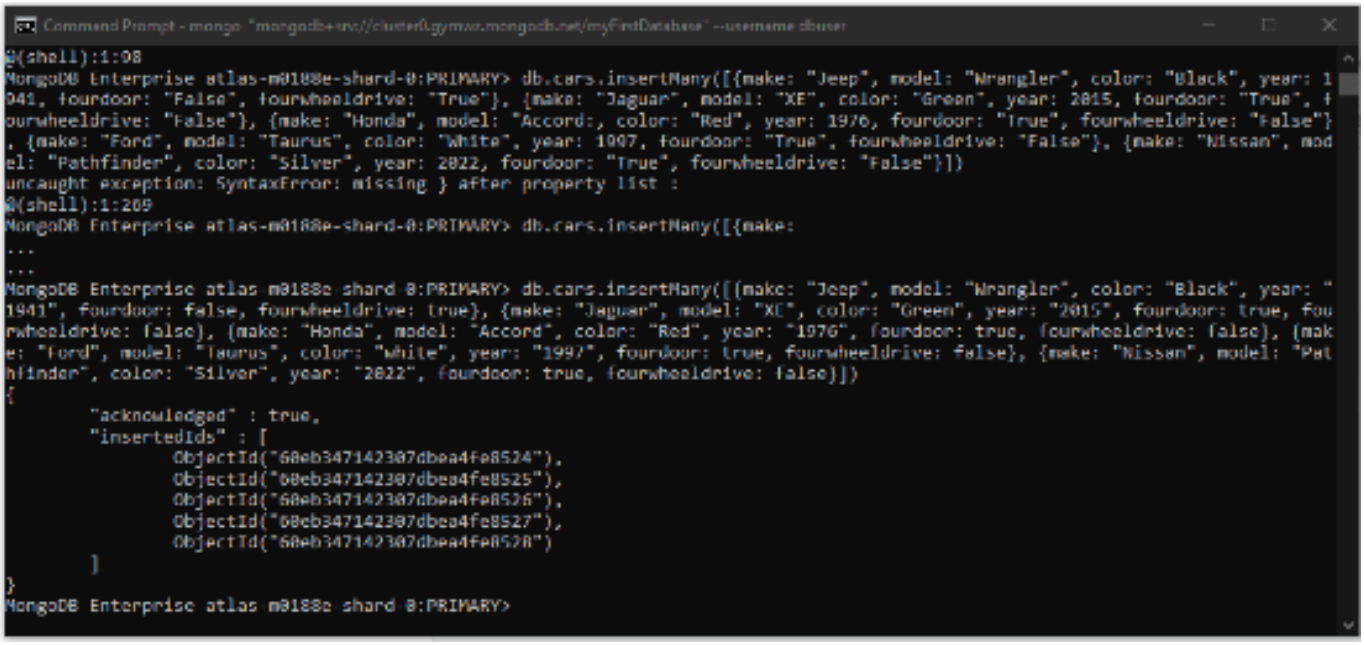
    - This new collection should be named `cars`.
2. Insert five cars into this collection using the `insertMany()` query.
    - Include the following fields: `make`, `model`, `color`, `year`, `fourDoor`, and `fourWheelDrive`.
        - The last two fields, `fourDoor` and `fourWheelDrive`, should be of type Boolean (i.e. true or false). `{ "sale": true }`
3. Once that is done, run a `find()` query to see your newly created documents.
4. Lastly, return to Atlas and view your collection and documents there.

Create new collection through Atlas



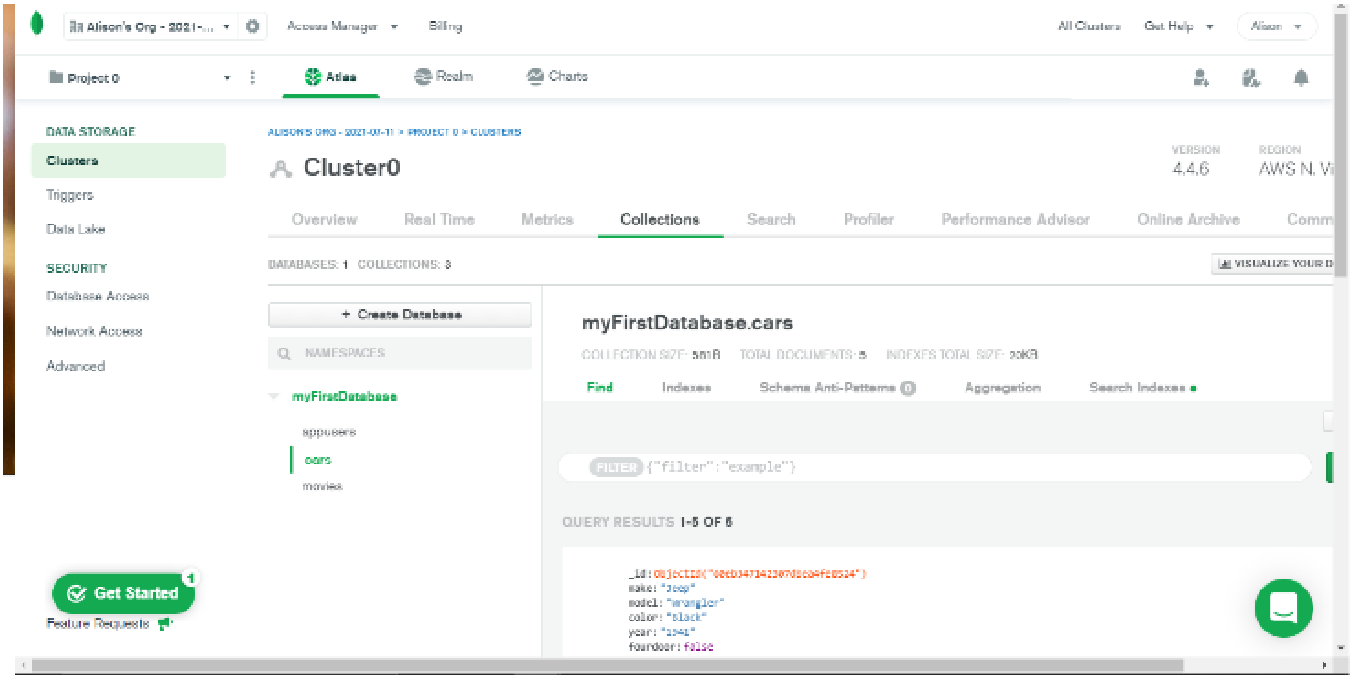


db.cars.insertMany([{make: "Jeep", model: "Wrangler", color: "Black", year: "1941", fourdoor: false, fourwheeldrive: true}, {make: "Jaguar", model: "XE", color: "Green", year: "2015", fourdoor: true, fourwheeldrive: false}, {make: "Honda", model: "Accord", color: "Red", year: "1976", fourdoor: true, fourwheeldrive: false}, {make: "Ford", model: "Taurus", color: "white", year: "1997", fourdoor: true, fourwheeldrive: false}, {make: "Nissan", model: "Pathfinder", color: "Silver", year: "2022", fourdoor: true, fourwheeldrive: false}])


Command Prompt - mongo "mongodb+srv://cluster0.gymvo.mongodb.net/myFirstDatabase" --username dbuser
@(shell):1:98
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.insertMany([{make: "Jeep", model: "Wrangler", color: "Black", year: 1941, fourdoor: "False", fourwheeldrive: "True"}, {make: "Jaguar", model: "XE", color: "Green", year: 2015, fourdoor: "True", fourwheeldrive: "False"}, {make: "Honda", model: "Accord", color: "Red", year: 1976, fourdoor: "True", fourwheeldrive: "False"}, {make: "Ford", model: "Taurus", color: "White", year: 1997, fourdoor: "True", fourwheeldrive: "False"}, {make: "Nissan", model: "Pathfinder", color: "Silver", year: 2022, fourdoor: "True", fourwheeldrive: "False"}])
uncaught exception: SyntaxError: missing } after property list :
@(shell):1:209
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.insertMany([{make:
...
...
MongoDB Enterprise atlas m0188e shard 0:PRIMARY> db.cars.insertMany([{make: "Jeep", model: "Wrangler", color: "Black", year: "1941", fourdoor: false, fourwheeldrive: true}, {make: "Jaguar", model: "XE", color: "Green", year: "2015", fourdoor: true, fourwheeldrive: false}, {make: "Honda", model: "Accord", color: "Red", year: "1976", fourdoor: true, fourwheeldrive: false}, {make: "Ford", model: "Taurus", color: "White", year: "1997", fourdoor: true, fourwheeldrive: false}, {make: "Nissan", model: "Pathfinder", color: "Silver", year: "2022", fourdoor: true, fourwheeldrive: false}])
{
"acknowledged" : true,
"insertedIds" : [
ObjectId("60eb347142307dbea4fe8524"),
ObjectId("60eb347142307dbea4fe8525"),
ObjectId("60eb347142307dbea4fe8526"),
ObjectId("60eb347142307dbea4fe8527"),
ObjectId("60eb347142307dbea4fe8528")
]
}
MongoDB Enterprise atlas m0188e shard 0:PRIMARY>


Command Prompt - mongo "mongodb+srv://cluster0.gymvo.mongodb.net/myFirstDatabase" --username dbuser
...
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.insertMany([{make: "Jeep", model: "Wrangler", color: "Black", year: "1941", fourdoor: false, fourwheeldrive: true}, {make: "Jaguar", model: "XE", color: "Green", year: "2015", fourdoor: true, fourwheeldrive: false}, {make: "Honda", model: "Accord", color: "Red", year: "1976", fourdoor: true, fourwheeldrive: false}, {make: "Ford", model: "Taurus", color: "White", year: "1997", fourdoor: true, fourwheeldrive: false}, {make: "Nissan", model: "Pathfinder", color: "Silver", year: "2022", fourdoor: true, fourwheeldrive: false}])
{
"acknowledged" : true,
"insertedIds" : [
ObjectId("60eb347142307dbea4fe8524"),
ObjectId("60eb347142307dbea4fe8525"),
ObjectId("60eb347142307dbea4fe8526"),
ObjectId("60eb347142307dbea4fe8527"),
ObjectId("60eb347142307dbea4fe8528")
]
}
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.find({})
{ "_id" : ObjectId("60eb347142307dbea4fe8524"), "make" : "Jeep", "model" : "Wrangler", "color" : "Black", "year" : "1041", "fourdoor" : false, "fourwheeldrive" : true }
{ "_id" : ObjectId("60eb347142307dbea4fe8525"), "make" : "Jaguar", "model" : "XE", "color" : "Green", "year" : "2015", "fourdoor" : true, "fourwheeldrive" : false }
{ "_id" : ObjectId("60eb347142307dbea4fe8526"), "make" : "Honda", "model" : "Accord", "color" : "Red", "year" : "1976", "fourdoor" : true, "fourwheeldrive" : false }
{ "_id" : ObjectId("60eb347142307dbea4fe8527"), "make" : "Ford", "model" : "Taurus", "color" : "White", "year" : "1997", "fourdoor" : true, "fourwheeldrive" : false }
{ "_id" : ObjectId("60eb347142307dbea4fe8528"), "make" : "Nissan", "model" : "Pathfinder", "color" : "Silver", "year" : "2022", "fourdoor" : true, "fourwheeldrive" : false }
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY>

db.cars.find({})


Alison's Org - 2021-...
Access Manager
Billing
All Clusters
Get Help
Alison
Project 0
Atlas
Realm
Charts
DATA STORAGE
Clusters
Triggers
Data Lake
SECURITY
Database Access
Network Access
Advanced
Cluster0
VERSION
4.4.6
REGION
Overview
Real Time
Metrics
Collections
Search
Profiler
Performance Advisor
Online Archive
DATABASES: 1 COLLECTIONS: 3
+ Create Database
NAMESPACES
myFirstDatabase
appusers
cars
movies
myFirstDatabase.cars
Find
Indexes
Schema Anti-Patterns
Aggregation
Search Indexes
FILTER
{"filter":"example"}
QUERY RESULTS 1-5 OF 5
Get Started
Feature Requests