

Part 1

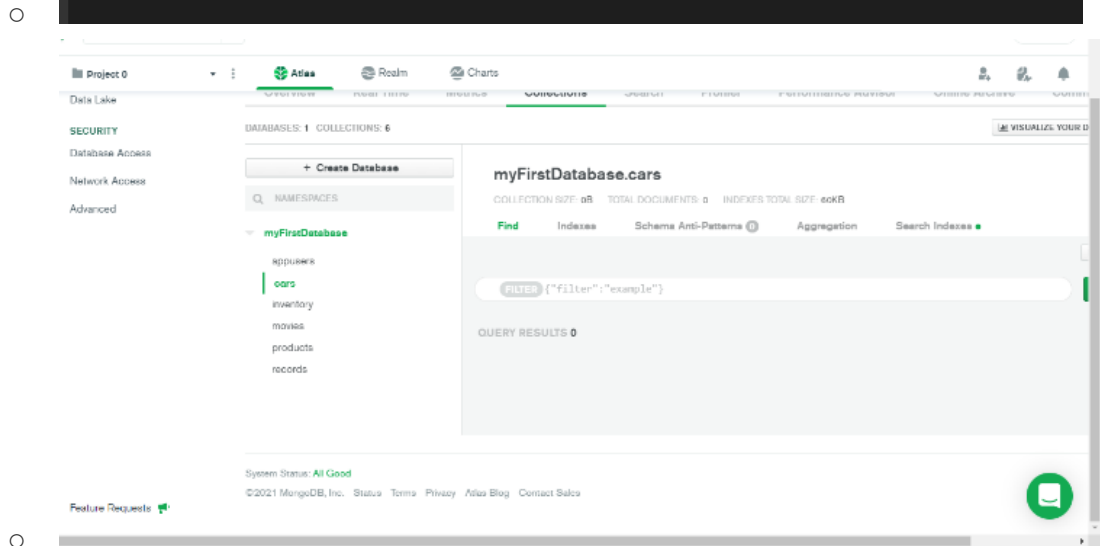
Follow the below steps:

1. Start off by deleting the entire collection cars.
 - Take a screenshot of the query as well as the list of your collections in Atlas to be sure this collection has been deleted.

```
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\music\Desktop> mongo "mongodb://cluster0-gmzc.mongodb.net/myFirstDatabase" --username admin
MongoDB shell version v4.4.6
Enter password:
Connecting to: mongodb://cluster0-shard-00-01-gmzc.mongodb.net:27011,cluster0-shard-00-02-gmzc.mongodb.net:27011,cluster0-shard-00-00-gmzc.mongodb.net:27011/myFirstDatabase?authSource=admin&compressor=disabled&gcpServiceName=mongodb&replicaSet=atlas-atl88e-shard-0&ssl=true
Implicit session session { "id" : UUID("95718912-56f8-45c1-8b8f-d4e88d90001") }
MongoDB server version: 4.4.6
Error while trying to show server startup warnings: user is not allowed to do action [getlog] on [admin.]
MongoDB Enterprise atlas-atl88e-shard-0:PRIMARY> db.cars.deleteMany({})
{"n":1}
MongoDB Enterprise atlas-atl88e-shard-0:PRIMARY>
```

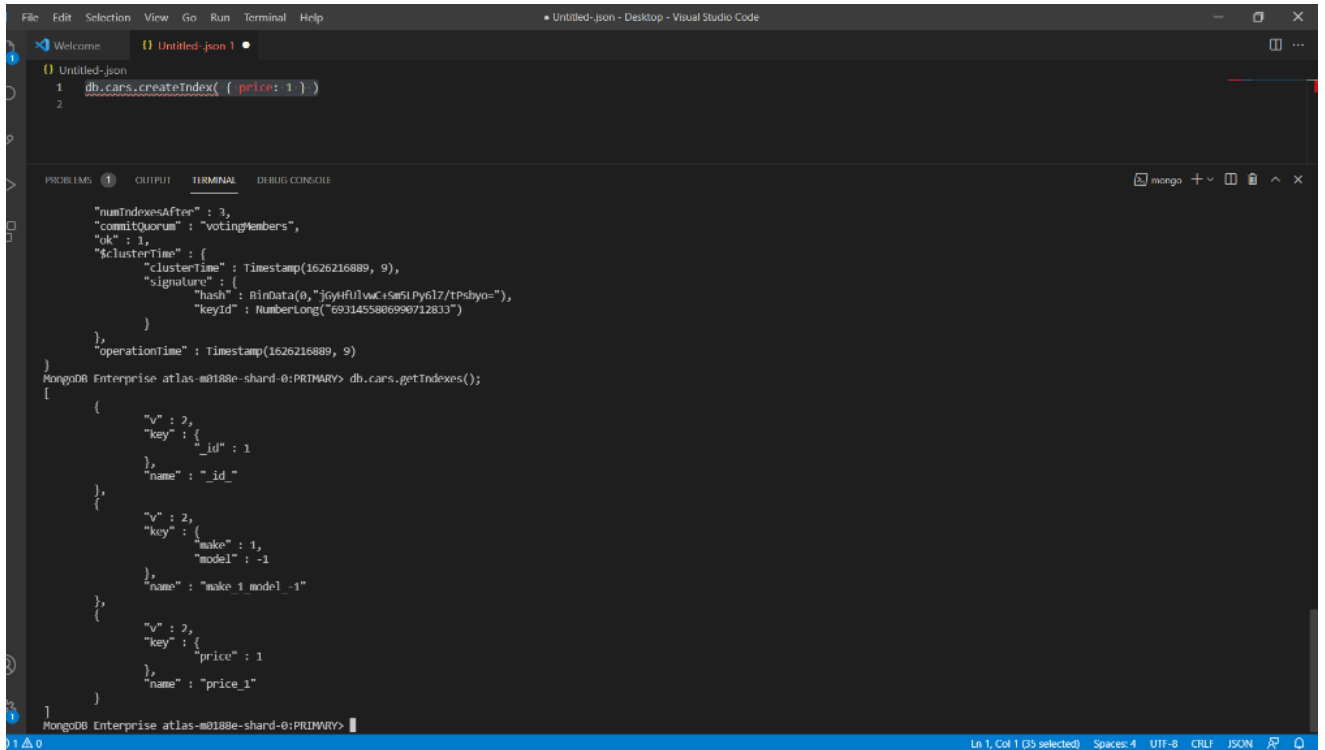


2. Next, run the following query to recreate the cars collection.
 - The following includes more cars than before.

```
db.cars.insertMany([
  {
    make: "Hyundai",
    model: "Santa Fe",
    price: 8000,
    year: 2003,
    used: true,
    color: "Black"
  },
  {
    make: "BMW",
    model: "ALPINA B6 Gran Coupe",
    price: 124300,
    year: 2017,
    used: false,
    color: "Mediterranean Blue Metallic"
  },
])
```

```
{
  make: "Subaru",
  model: "Crosstrek 2.0i Premium",
  price: 22595,
  year: 2014,
  used: true,
  color: "Sunshine Orange"
},
{
  make: "Ford",
  model: "F-350 XL",
  price: 33705,
  year: 2017,
  used: false,
  color: "Race Red"
},
{
  make: "Toyota",
  model: "Acura MDX",
  price: 28800,
  year: 2014,
  used: true,
  color: "Graphite Luster Metallic"
},
{
  make: "BMW",
  model: "5 Series 535i Sedan",
  price: 18995,
  year: 2013,
  used: true,
  color: "Space Gray Metallic"
},
{
  make: "Ford",
  model: "Escape",
  price: 7480,
  year: 2011,
  used: true,
  color: "Sterling Grey Metallic"
},
{
  make: "Subaru",
  model: "Impreza",
  price: 18495,
```

```
    year: 2018,  
    used: false,  
    color: "Crimson Red Pearl"  
  },  
  {  
    make: "Toyota",  
    model: "Yaris",  
    price: 15635,  
    year: 2018,  
    used: false,  
    color: "Super White"  
  },  
  {  
    make: "Honda",  
    model: "Civic LX",  
    price: 14999,  
    year: 2016,  
    used: true,  
    color: "Crystal Black Pearl"  
  },  
  {  
    make: "Volkswagen",  
    model: "Jetta 1.4T S",  
    price: 19495,  
    year: 2018,  
    used: false,  
    color: "Silk Blue Metallic"  
  }  
];
```

```
File Edit Selection View Go Run Terminal Help
Untitled - json - Desktop - Visual Studio Code

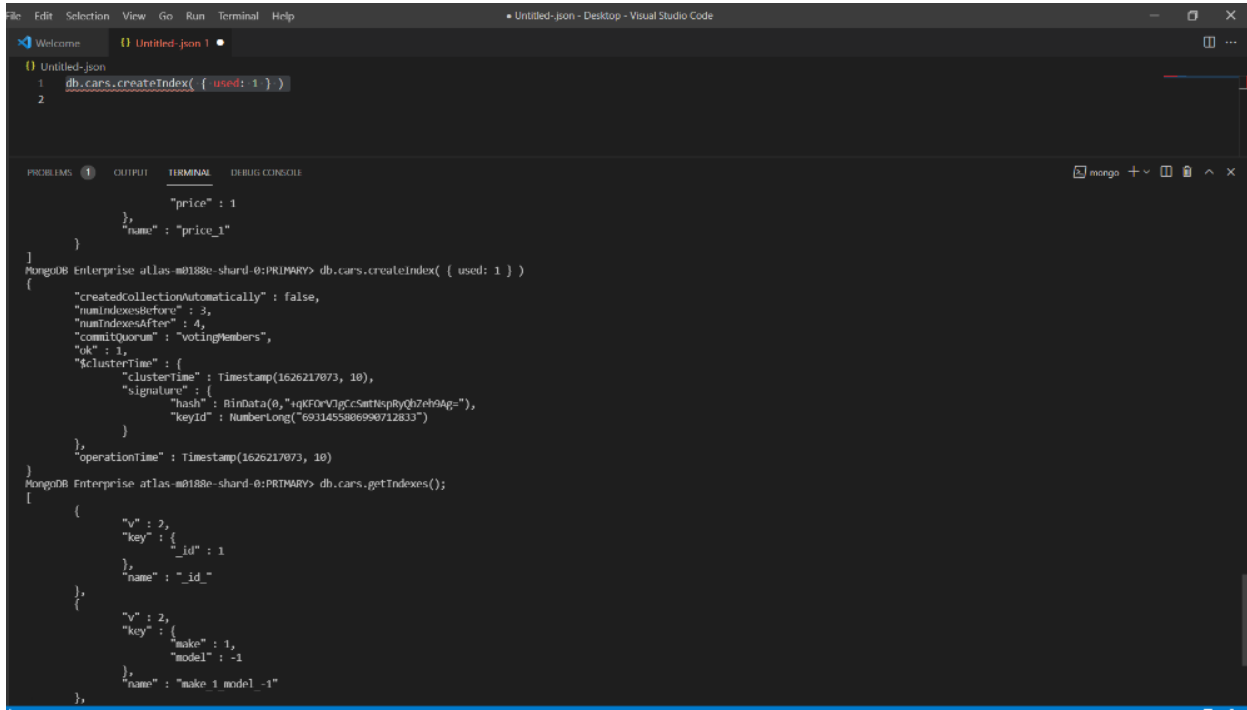
() Untitled-json
1 db.cars.createIndex( { price: 1 } )
2

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.getIndexes();
[
  {
    "v": 2,
    "key": {
      "_id": 1
    },
    "name": "_id_"
  },
  {
    "v": 2,
    "key": {
      "make": 1,
      "model": -1
    },
    "name": "make_1_model_-1"
  },
  {
    "v": 2,
    "key": {
      "price": 1
    },
    "name": "price_1"
  }
]
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY>
```

db.cars.getIndexes();

4. Create an index on the used field for the cars collection.

```
db.cars.createIndex( { used: 1 } )
```



```
File Edit Selection View Go Run Terminal Help
Untitled - json - Desktop - Visual Studio Code

() Untitled-json
1 db.cars.createIndex( { used: 1 } )
2

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.createIndex( { used: 1 } )
{
  "createdCollectionAutomatically": false,
  "numIndexesBefore": 3,
  "numIndexesAfter": 4,
  "commitQuorum": "votingMembers",
  "ok": 1,
  "$clusterTime": {
    "clusterTime": Timestamp(1626217073, 10),
    "signature": {
      "hash": BinData(0,"+qR0rV7pCcSmtNpbyQh7eh9Ag="),
      "keyId": NumberLong("6931455866990712833")
    }
  },
  "operationTime": Timestamp(1626217073, 10)
}
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.getIndexes();
[
  {
    "v": 2,
    "key": {
      "_id": 1
    },
    "name": "_id_"
  },
  {
    "v": 2,
    "key": {
      "make": 1,
      "model": -1
    },
    "name": "make_1_model_-1"
  },
  {
    "v": 2,
    "key": {
      "price": 1
    },
    "name": "price_1"
  },
  {
    "v": 2,
    "key": {
      "used": 1
    },
    "name": "used_1"
  }
]
```

5. Find and delete all documents with a year before 2012.

- Be sure to do a find with your filtering criteria first to be sure you're about to delete the correct documents.

```
db.cars.find({year : {$lt : 2012}})
```

```
db.cars.deleteMany({year : {$lt : 2012}})
```

```
    "name" : "used_1"
  }
}
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.find(year)
uncaught exception: ReferenceError: year is not defined :
@($shell):1:11
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.find(year)
uncaught exception: ReferenceError: year is not defined :
@($shell):1:15
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.find(year < 2012)
uncaught exception: SyntaxError: missing : after property id :
@($shell):1:19
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.find(year : {$lt : 2012})
uncaught exception: SyntaxError: missing : after property id :
@($shell):1:27
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.find(year : {$lt : 2012})
{ "_id" : ObjectId("60ee1954a0b35bf37c91670"), "make" : "Hyundai", "model" : "Santa Fe", "price" : 8000, "year" : 2003, "used" : true, "color" : "Black" }
{ "_id" : ObjectId("60ee1954a0b35bf37c91676"), "make" : "Ford", "model" : "Escape", "price" : 7400, "year" : 2011, "used" : true, "color" : "Sterling Grey Metallic" }
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.deleteMany({year : {$lt : 2012}})
{ "acknowledged" : true, "deletedCount" : 2 }
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY>
```

6. Delete the first document that is a BMW.

```
db.cars.deleteOne({make : "BMW"})
```

```
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.find({make : {$eq : "BMW"}})
...
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.find({make : "BMW"})
{ "_id" : ObjectId("60ee1954a0b35bf37c91671"), "make" : "BMW", "model" : "ALPINA B6 Gran Coupe", "price" : 124300, "year" : 2017, "used" : false, "color" : "Mediterranean Blue Metallic" }
{ "_id" : ObjectId("60ee1954a0b35bf37c91675"), "make" : "BMW", "model" : "5 Series 535i Sedan", "price" : 18995, "year" : 2013, "used" : true, "color" : "Space Gray Metallic" }
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.deleteOne({make : "BMW"})
{ "acknowledged" : true, "deletedCount" : 1 }
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.find({make : "BMW"})
{ "_id" : ObjectId("60ee1954a0b35bf37c91675"), "make" : "BMW", "model" : "5 Series 535i Sedan", "price" : 18995, "year" : 2013, "used" : true, "color" : "Space Gray Metallic" }
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY>
```

7. Drop the index created on the non-used cars created above.

```
db.cars.drop ( { used: 1 } )
```

```
    "name" : "used_1"
  }
}
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.drop ( { used: 1 } )
uncaught exception: SyntaxError: unexpected token: '{' :
@($shell):1:13
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.drop ( { used: 1 } )
true
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.drop ( { used: 1 } )
false
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY>
```

Part 2

Below is a real-life scenario. Please read this scenario and run the appropriate queries needed. You are currently working for a car dealership. They sell both used and new cars. The company would like to easily and efficiently search through their cars using the "make" of the car. Recently, they made the searching efficient using the price of the car, but that is no longer needed since they will now be using the make of the vehicles. Please reflect that in the database. Also, the company has decided to no longer sell Volkswagens and has already sold the last Volkswagen on the lot so they would like you to reflect that in the database as well.

Delete make Volkswagen

```
db.cars.deleteMany({make : "Volkswagen"})
```

```
{ "acknowledged" : true, "deletedCount" : 1 }
```

Search through make for cars

```
db.cars.createIndex( {make: 1 } )
```

```
@(shell):1:23
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY> db.cars.createIndex( {make: 1 } )
{
  "createdCollectionAutomatically" : true,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "commitQuorum" : "votingMembers",
  "ok" : 1,
  "$clusterTime" : {
    "clusterTime" : timestamp(1626218807, 5),
    "signature" : {
      "hash" : BinData(9,"wY/qpsV6IHQ2sM0cVQ99ZTWE+We="),
      "keyId" : NumberLong("6931455800990/12835")
    }
  },
  "operationTime" : timestamp(1626218807, 5)
}
MongoDB Enterprise atlas-m0188e-shard-0:PRIMARY>
```

Ln 6, Col 1 (35 selected) Spaces: 4 UTF-8 ORF JSON

NoSQL HandsOn4 - Google Do... x