NoSQL-HandsOn4

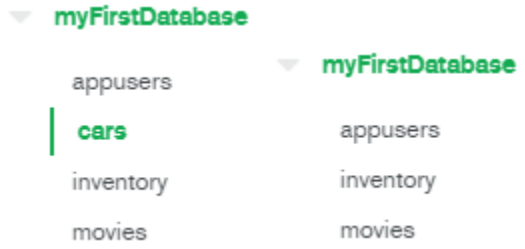
Alberta “Albi” Kovatcheva

Part 1

1. Start off by deleting the entire collection cars.
   Queries:
   db.cars.drop()

```
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.drop()
true
```

   Results: Before and after deletion of the ‘cars’ collection.



   Former ‘cars’ collection documented in appendix.

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

```
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"
  }
]);
```

Results:

```
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("60ef6a952e7b5a0e31ae136e"),
                ObjectId("60ef6a952e7b5a0e31ae136f"),
                ObjectId("60ef6a952e7b5a0e31ae1370"),
                ObjectId("60ef6a952e7b5a0e31ae1371"),
                ObjectId("60ef6a952e7b5a0e31ae1372"),
                ObjectId("60ef6a952e7b5a0e31ae1373"),
                ObjectId("60ef6a952e7b5a0e31ae1374"),
                ObjectId("60ef6a952e7b5a0e31ae1375"),
                ObjectId("60ef6a952e7b5a0e31ae1376"),
                ObjectId("60ef6a952e7b5a0e31ae1377"),
                ObjectId("60ef6a952e7b5a0e31ae1378")
        ]
}
```

3. Create an index on the price field.
   Query:
   db.cars.createIndex({ price: 1 })
   db.cars.getIndexes()

Results:

```
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.createIndex( { price: 1 } )
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "commitQuorum" : "votingMembers",
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1626303346, 7),
                "signature" : {
                        "hash" : BinData(0,"ITkYB7dLnbPkzCErio9MGP1R4kc="),
                        "keyId" : NumberLong("6929556022696607745")
                }
        },
        "operationTime" : Timestamp(1626303346, 7)
}
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.getIndexes();
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_"
        },
        {
                "v" : 2,
                "key" : {
                        "price" : 1
                },
                "name" : "price_1"
        }
]
```

4. Create an index on the used field for the cars collection.
   Queries:
   db.cars.createIndex({used: 1})
   db.cars.getIndexes()

   Results:

```
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.createIndex({used: 1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 2,
        "numIndexesAfter" : 3,
        "commitQuorum" : "votingMembers",
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1626303772, 7),
                "signature" : {
                        "hash" : BinData(0,"lki+UIwFHM4yWJRfHyxcnfSyze0="),
                        "keyId" : NumberLong("6929556022696607745")
                }
        },
        "operationTime" : Timestamp(1626303772, 7)
}
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.getIndexes()
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_"
        },
        {
                "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.

Queries:
db.cars.find({ year: { $lt : 2012 }}).pretty()
db.cars.deleteMany({ year: { $lt : 2012 }})

Results:

```
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.find({ year: { $lt : 2012 }}).pretty()
{
        "_id" : ObjectId("60ef6a952e7b5a0e31ae136e"),
        "make" : "Hyundai",
        "model" : "Santa Fe",
        "price" : 8000,
        "year" : 2003,
        "used" : true,
        "color" : "Black"
}
{
        "_id" : ObjectId("60ef6a952e7b5a0e31ae1374"),
        "make" : "Ford",
        "model" : "Escape",
        "price" : 7480,
        "year" : 2011,
        "used" : true,
        "color" : "Sterling Grey Metallic"
}
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.deleteMany({ year: { $lt : 2012 }})
{ "acknowledged" : true, "deletedCount" : 2 }
```

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

6. Delete the first document that is a BMW.
   Queries:
   db.cars.find({make: "BMW"}).pretty()
   db.cars.deleteOne({make: "BMW"})
   db.cars.find({make: "BMW"}).pretty()

   Results:

```
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.find({make: "BMW"}).pretty()
{
        "_id" : ObjectId("60ef6a952e7b5a0e31ae136f"),
        "make" : "BMW",
        "model" : "ALPINA B6 Gran Coupe",
        "price" : 124300,
        "year" : 2017,
        "used" : false,
        "color" : "Mediterranean Blue Metallic"
}
{
        "_id" : ObjectId("60ef6a952e7b5a0e31ae1373"),
        "make" : "BMW",
        "model" : "5 Series 535i Sedan",
        "price" : 18995,
        "year" : 2013,
        "used" : true,
        "color" : "Space Gray Metallic"
}
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.deleteOne({make: "BMW"})
{ "acknowledged" : true, "deletedCount" : 1 }
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.find({make: "BMW"}).pretty()
{
        "_id" : ObjectId("60ef6a952e7b5a0e31ae1373"),
        "make" : "BMW",
        "model" : "5 Series 535i Sedan",
        "price" : 18995,
        "year" : 2013,
        "used" : true,
        "color" : "Space Gray Metallic"
}
```

7. Drop the index created on the used cars created above.
   Queries:
   db.cars.dropIndex({used: 1})
   db.cars.getIndexes()

   Results:

NoSQL-HandsOn4

Alberta “Albi” Kovatcheva

```
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.dropIndex({used: 1})
{
        "nIndexesWas" : 3,
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1626307039, 1),
                "signature" : {
                        "hash" : BinData(0,"c5eNEQxF1PrNu179bSv5YCBgN0k="),
                        "keyId" : NumberLong("6929556022696607745")
                }
        },
        "operationTime" : Timestamp(1626307039, 1)
}
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.getIndexes()
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_"
        },
        {
                "v" : 2,
                "key" : {
                        "price" : 1
                },
                "name" : "price_1"
        }
]
```

NoSQL-HandsOn4

Alberta “Albi” Kovatcheva

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.

Queries:

```
db.cars.createIndex({make: 1})
db.cars.dropIndex({price: 1})
db.cars.find({ make: "Volkswagen"}).pretty()
db.cars.deleteMany({ make: "Volkswagen"})
```

Results:

```
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.createIndex({make: 1})
{"t":{"$date":"2021-07-15T00:18:37.763Z"},"s":"W",  "c":"NETWORK",  "id":23019,   "ctx":"ReplicaSetMonitor-TaskExecutor","msg":"DNS resolution whil
e connecting to peer was slow","attr":{"peer":"cluster0-shard-00-00.bst2l.mongodb.net:27017","durationMillis":7791}}
{"t":{"$date":"2021-07-15T00:18:37.865Z"},"s":"W",  "c":"NETWORK",  "id":23019,   "ctx":"ReplicaSetMonitor-TaskExecutor","msg":"DNS resolution whil
e connecting to peer was slow","attr":{"peer":"cluster0-shard-00-01.bst2l.mongodb.net:27017","durationMillis":7860}}
{"t":{"$date":"2021-07-15T00:18:37.867Z"},"s":"W",  "c":"NETWORK",  "id":23019,   "ctx":"ReplicaSetMonitor-TaskExecutor","msg":"DNS resolution whil
e connecting to peer was slow","attr":{"peer":"cluster0-shard-00-02.bst2l.mongodb.net:27017","durationMillis":7790}}
{"t":{"$date":"2021-07-15T00:18:37.869Z"},"s":"W",  "c":"NETWORK",  "id":23019,   "ctx":"ReplicaSetMonitor-TaskExecutor","msg":"DNS resolution whil
e connecting to peer was slow","attr":{"peer":"cluster0-shard-00-00.bst2l.mongodb.net:27017","durationMillis":7233}}
{"t":{"$date":"2021-07-15T00:18:37.871Z"},"s":"W",  "c":"NETWORK",  "id":23019,   "ctx":"ReplicaSetMonitor-TaskExecutor","msg":"DNS resolution whil
e connecting to peer was slow","attr":{"peer":"cluster0-shard-00-01.bst2l.mongodb.net:27017","durationMillis":7235}}
{"t":{"$date":"2021-07-15T00:18:37.873Z"},"s":"W",  "c":"NETWORK",  "id":23019,   "ctx":"ReplicaSetMonitor-TaskExecutor","msg":"DNS resolution whil
e connecting to peer was slow","attr":{"peer":"cluster0-shard-00-02.bst2l.mongodb.net:27017","durationMillis":7237}}
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 2,
        "numIndexesAfter" : 3,
        "commitQuorum" : "votingMembers",
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1626308318, 9),
                "signature" : {
                        "hash" : BinData(0,"Be9h+MS1ax7ar8/d85qSag9wafw="),
                        "keyId" : NumberLong("6929556022696607745")
                }
        },
        "operationTime" : Timestamp(1626308318, 9)
}
```

```
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.dropIndex({price: 1})
{
        "nIndexesWas" : 3,
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1626308384, 1),
                "signature" : {
                        "hash" : BinData(0,"iXIaUl5ThAJ8Cilcu5xFYmzUyi8="),
                        "keyId" : NumberLong("6929556022696607745")
                }
        },
        "operationTime" : Timestamp(1626308384, 1)
}
```

NoSQL-HandsOn4

Alberta “Albi” Kovatcheva

```
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.find({ make: "Volkswagen"}).pretty()
{
        "_id" : ObjectId("60ef6a952e7b5a0e31ae1378"),
        "make" : "Volkswagen",
        "model" : "Jetta 1.4T S",
        "price" : 19495,
        "year" : 2018,
        "used" : false,
        "color" : "Silk Blue Metallic"
}
```

```
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.deleteMany({ make: "Volkswagen"})
{ "acknowledged" : true, "deletedCount" : 1 }
```

NoSQL-HandsOn4

Alberta “Albi” Kovatcheva

Appendix

1. Former “cars” collection.

```
MongoDB Enterprise atlas-y1r8u5-shard-0:PRIMARY> db.cars.find({}).pretty()
{
        "_id" : ObjectId("60ea154e2c25749797dfe7df"),
        "make" : "Toyota",
        "model" : "Corolla",
        "color" : "gold",
        "year" : 2006,
        "fourDoor" : true,
        "fourWheelDrive" : true
}
{
        "_id" : ObjectId("60ea154e2c25749797dfe7e0"),
        "make" : "Nissan",
        "model" : "Versa",
        "color" : "white",
        "year" : 2012,
        "fourDoor" : true,
        "fourWheelDrive" : true
}
{
        "_id" : ObjectId("60ea154e2c25749797dfe7e1"),
        "make" : "Tesla",
        "model" : "Roadster",
        "color" : "red",
        "year" : 2022,
        "fourDoor" : false,
        "fourWheelDrive" : true
}
{
        "_id" : ObjectId("60ea154e2c25749797dfe7e2"),
        "make" : "Lamborghini",
        "model" : "Urus",
        "color" : "yellow",
        "year" : 2021,
        "fourDoor" : true,
        "fourWheelDrive" : true
}
{
        "_id" : ObjectId("60ea154e2c25749797dfe7e3"),
        "make" : "Mercedes",
        "model" : "AMGG63",
        "color" : "white",
        "year" : 2019,
```

```
    "fourDoor" : true,
    "fourWheelDrive" : true
}
```