DS108-04-08 - NoSQL - Lesson 4 Hands-On

For your Lesson 4 Hands-On, you will be working with your new knowledge on NoSQL. This Hands-On will be graded, so be sure you complete all requirements.

> [!Caution] Caution! Do not submit your project until you have completed all requirements, as you will not be able to resubmit.

> [!info] To Submit Be sure to zip and submit your `NoSQL–HandsOn4` text document when finished! You will not be able to re-submit, so be sure the screenshots to each part are located within this document.

Requirements

This Hands-On is structured into *two* parts, and each part may ask you to run multiple queries. After each query, please take a screenshot and add it to a text document (or an equivalent) and name this file `NoSQL–HandsOn4`. This way, you will be able to submit your answers to each part all at once. Good luck! ___

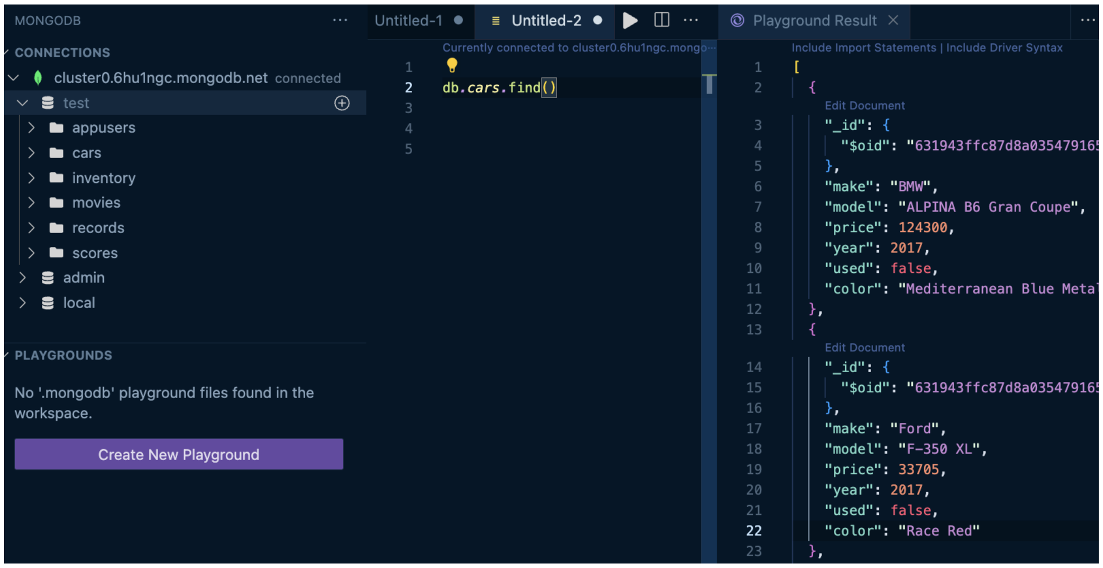
Part 1

Follow the below steps:

1.1 Delete the entire collection `cars`

- 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.

`db.cars.find();`

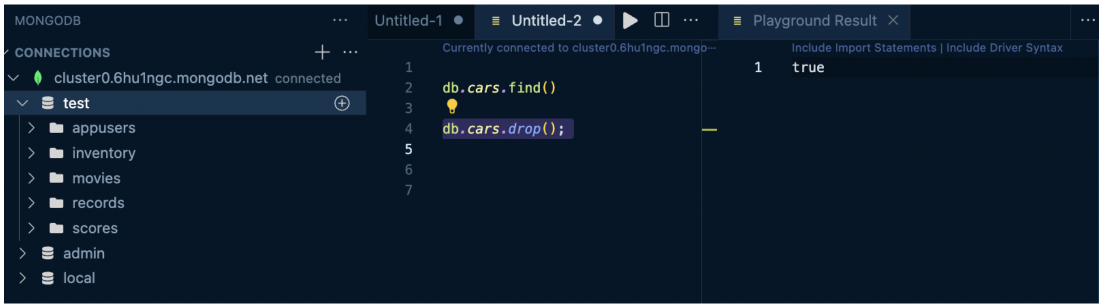


MONGODB
CONNECTIONS
cluster0.6hu1ngc.mongodb.net connected
test
appusers
cars
inventory
movies
records
scores
admin
local
PLAYGROUNDS
No '.mongodb' playground files found in the workspace.
Create New Playground
Untitled-1
Untitled-2
Playground Result
Currently connected to cluster0.6hu1ngc.mongo...
Include Import Statements | Include Driver Syntax
Edit Document

```
db.cars.find()
```

```
[
  {
    "_id": {
      "$oid": "631943ffc87d8a03547916
    },
    "make": "BMW",
    "model": "ALPINA B6 Gran Coupe",
    "price": 124300,
    "year": 2017,
    "used": false,
    "color": "Mediterranean Blue Meta
  },
  {
    "_id": {
      "$oid": "631943ffc87d8a03547916
    },
    "make": "Ford",
    "model": "F-350 XL",
    "price": 33705,
    "year": 2017,
    "used": false,
    "color": "Race Red"
  },
```

`db.cars.drop();`



MONGODB
CONNECTIONS
cluster0.6hu1ngc.mongodb.net connected
test
appusers
inventory
movies
records
scores
admin
local
Untitled-1
Untitled-2
Playground Result
Currently connected to cluster0.6hu1ngc.mongo...
Include Import Statements | Include Driver Syntax

```

db.cars.find()

db.cars.drop();
```

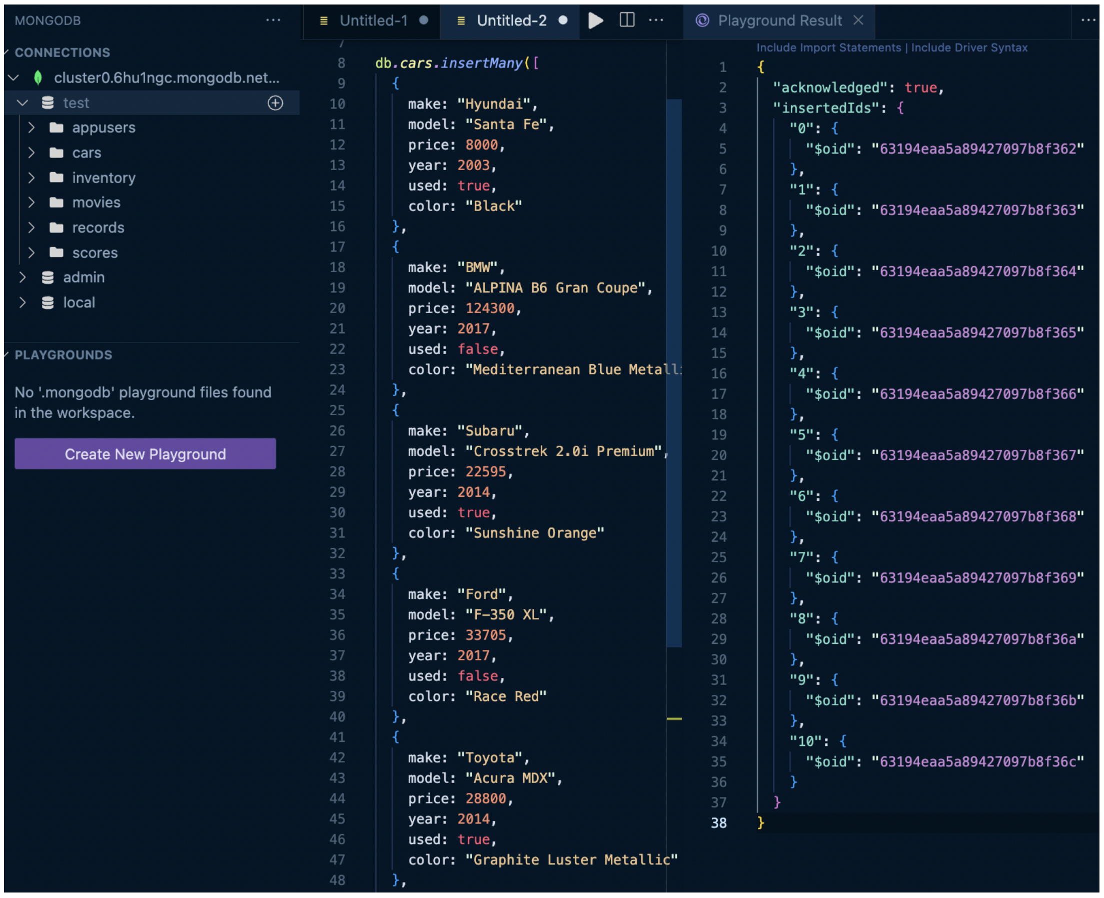
```
true
```

1.2 Run the query to recreate the `cars` collection

- 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"
  }
]);
```

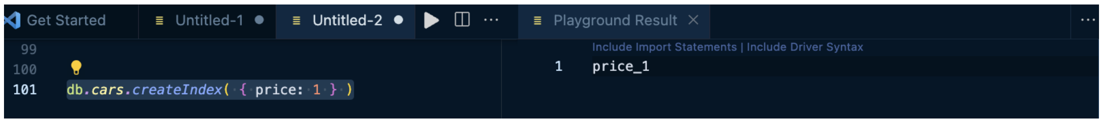


```
db.cars.insertMany([
  {
    make: "Hyundai",
    model: "Santa Fe",
    price: 8000,
    year: 2003,
    used: true,
    color: "Black"
  },
  {
    make: "BMW",
    model: "ALPINA B6 Gran Coupe",
    price: 124300,
    year: 2017,
    used: false,
    color: "Mediterranean Blue Metall
  },
  {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

```
{
  "acknowledged": true,
  "insertedIds": {
    "0": {
      "$oid": "63194eaa5a89427097b8f362"
    },
    "1": {
      "$oid": "63194eaa5a89427097b8f363"
    },
    "2": {
      "$oid": "63194eaa5a89427097b8f364"
    },
    "3": {
      "$oid": "63194eaa5a89427097b8f365"
    },
    "4": {
      "$oid": "63194eaa5a89427097b8f366"
    },
    "5": {
      "$oid": "63194eaa5a89427097b8f367"
    },
    "6": {
      "$oid": "63194eaa5a89427097b8f368"
    },
    "7": {
      "$oid": "63194eaa5a89427097b8f369"
    },
    "8": {
      "$oid": "63194eaa5a89427097b8f36a"
    },
    "9": {
      "$oid": "63194eaa5a89427097b8f36b"
    },
    "10": {
      "$oid": "63194eaa5a89427097b8f36c"
    }
  }
}
```

1.3 Create an index on the `price` field.

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



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

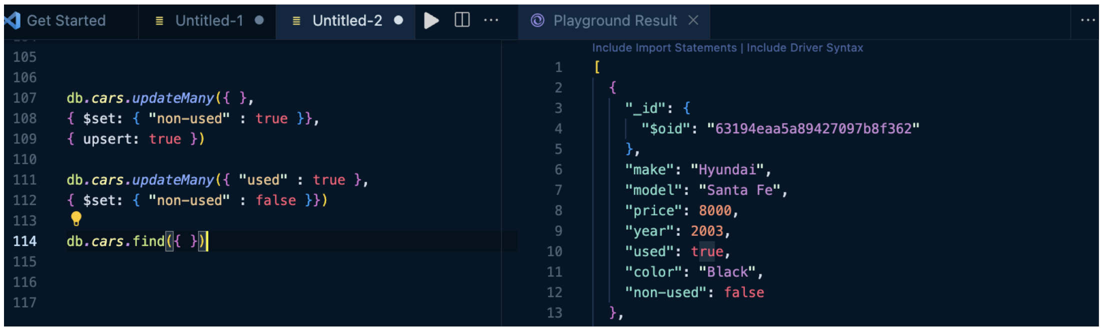
```
price_1
```

1.4 Create an index on the `non-used` field for the cars collection.

```
db.cars.updateMany({ },
{ $set: { "non-used" : true }},
{ upsert: true })
```

```
db.cars.updateMany({ "used" : true },
{ $set: { "non-used" : false }})
```

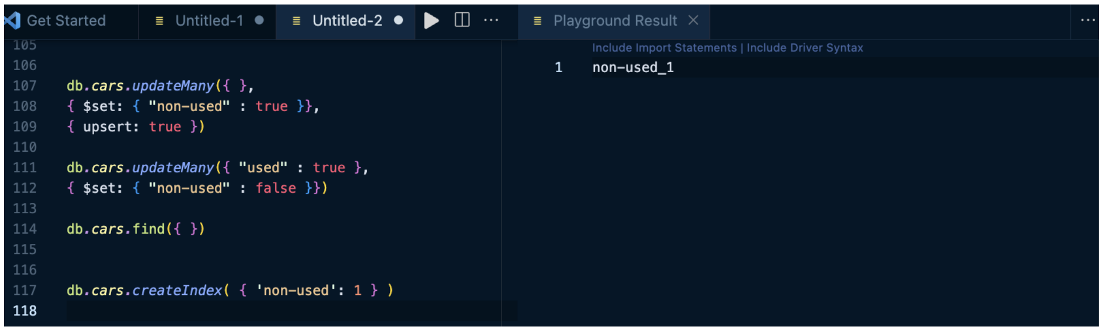
```
db.cars.find({ })
```



```
db.cars.updateMany({ },
{ $set: { "non-used" : true }},
{ upsert: true })

db.cars.updateMany({ "used" : true },
{ $set: { "non-used" : false }})

db.cars.find({ })
```

```
[
  {
    "_id": {
      "$oid": "63194eaa5a89427097b8f362"
    },
    "make": "Hyundai",
    "model": "Santa Fe",
    "price": 8000,
    "year": 2003,
    "used": true,
    "color": "Black",
    "non-used": false
  },
```

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



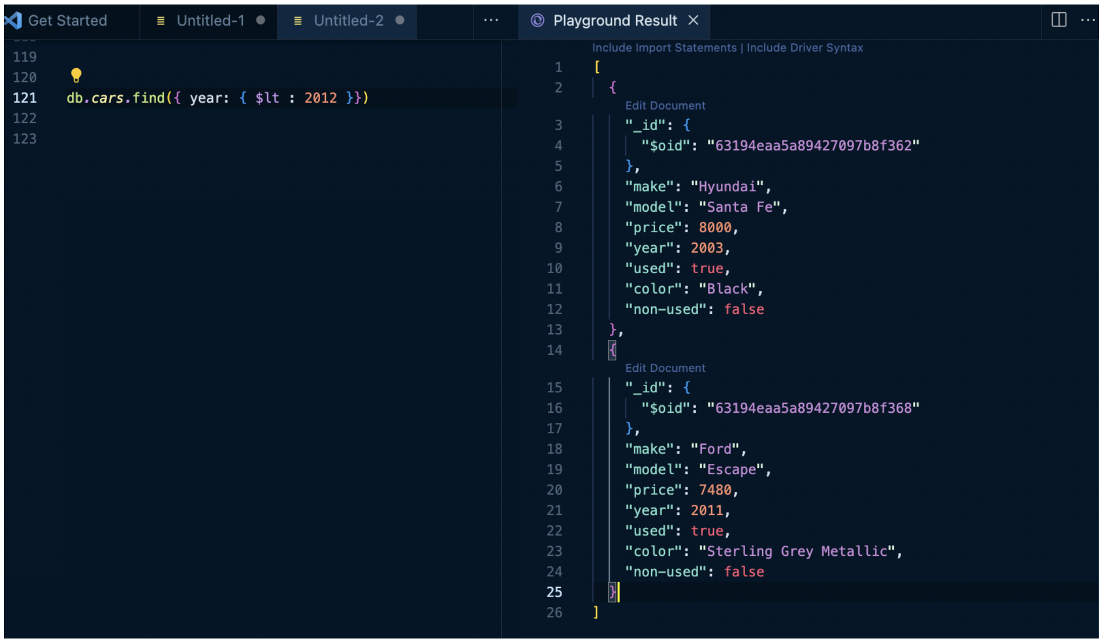
```
db.cars.updateMany({ },
{ $set: { "non-used" : true }},
{ upsert: true })

db.cars.updateMany({ "used" : true },
{ $set: { "non-used" : false }})

db.cars.find({ })


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

```
non-used_1
```

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.

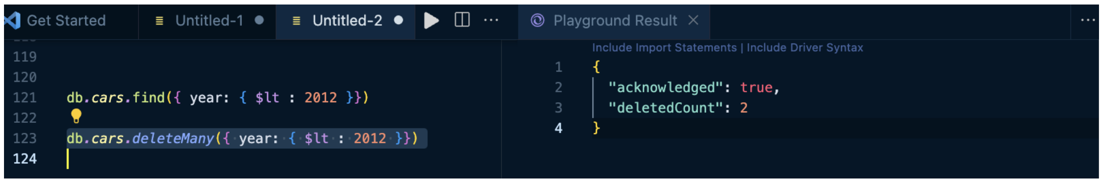
```
db.cars.find({ year: { $lt : 2012 }})
```



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

```
[
  {
    "_id": {
      "$oid": "63194eaa5a89427097b8f362"
    },
    "make": "Hyundai",
    "model": "Santa Fe",
    "price": 8000,
    "year": 2003,
    "used": true,
    "color": "Black",
    "non-used": false
  },
  {
    "_id": {
      "$oid": "63194eaa5a89427097b8f368"
    },
    "make": "Ford",
    "model": "Escape",
    "price": 7480,
    "year": 2011,
    "used": true,
    "color": "Sterling Grey Metallic",
    "non-used": false
  }
]
```

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

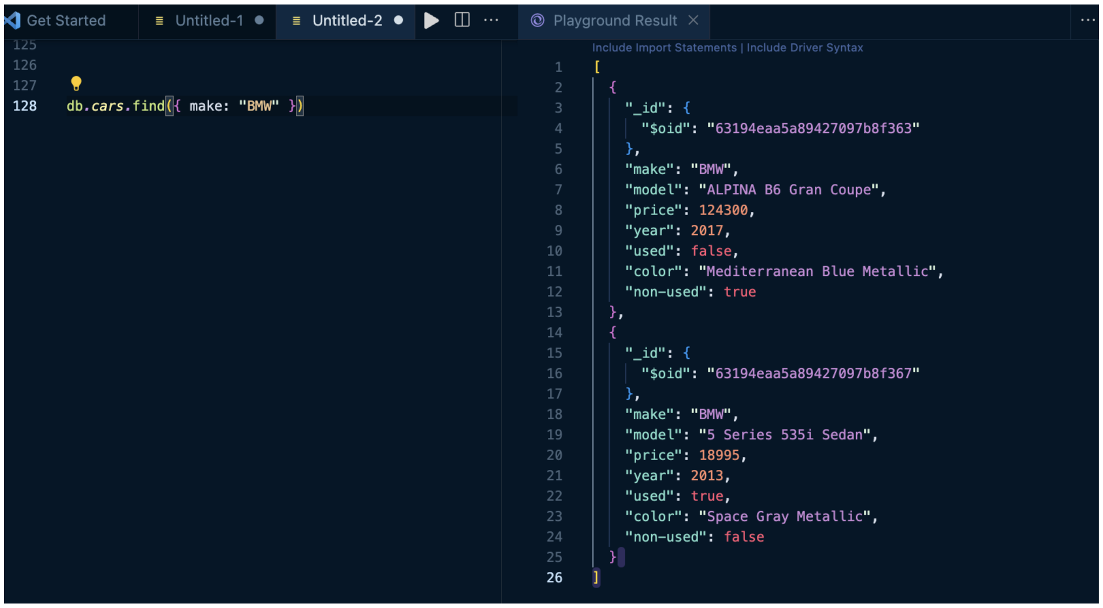


```
db.cars.find({ year: { $lt : 2012 }})

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

```
{
  "acknowledged": true,
  "deletedCount": 2
}
```

1.6 Delete the first document that is a `BMW`.

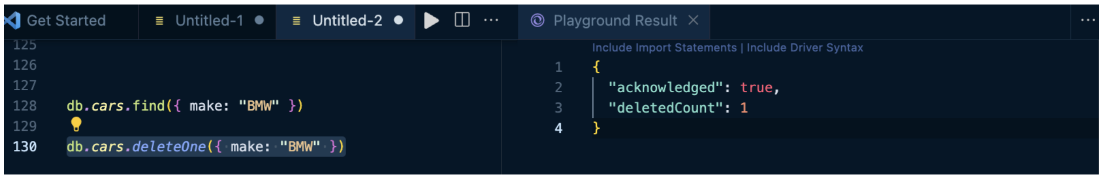
`db.cars.find({ make: "BMW" })`



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

```
[
  {
    "_id": {
      "$oid": "63194eaa5a89427097b8f363"
    },
    "make": "BMW",
    "model": "ALPINA B6 Gran Coupe",
    "price": 124300,
    "year": 2017,
    "used": false,
    "color": "Mediterranean Blue Metallic",
    "non-used": true
  },
  {
    "_id": {
      "$oid": "63194eaa5a89427097b8f367"
    },
    "make": "BMW",
    "model": "5 Series 535i Sedan",
    "price": 18995,
    "year": 2013,
    "used": true,
    "color": "Space Gray Metallic",
    "non-used": false
  }
]
```

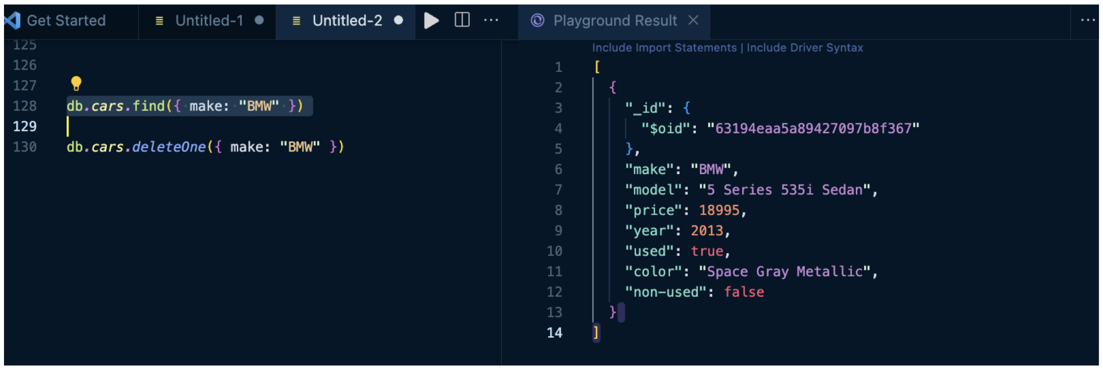
`db.cars.deleteOne({ make: "BMW" })`



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

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

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

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

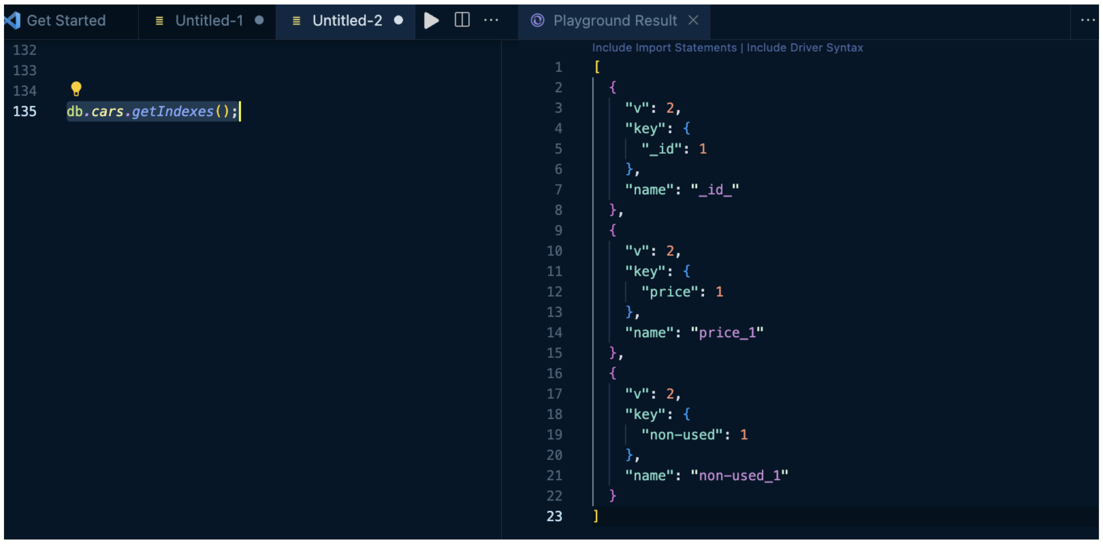


```
db.cars.find({ make: "BMW" })

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

```
[
  {
    "_id": {
      "$oid": "63194eaa5a89427097b8f367"
    },
    "make": "BMW",
    "model": "5 Series 535i Sedan",
    "price": 18995,
    "year": 2013,
    "used": true,
    "color": "Space Gray Metallic",
    "non-used": false
  }
]
```

1.7 Drop the index created on the `non-used` cars created above.

```
db.cars.getIndexes();
```

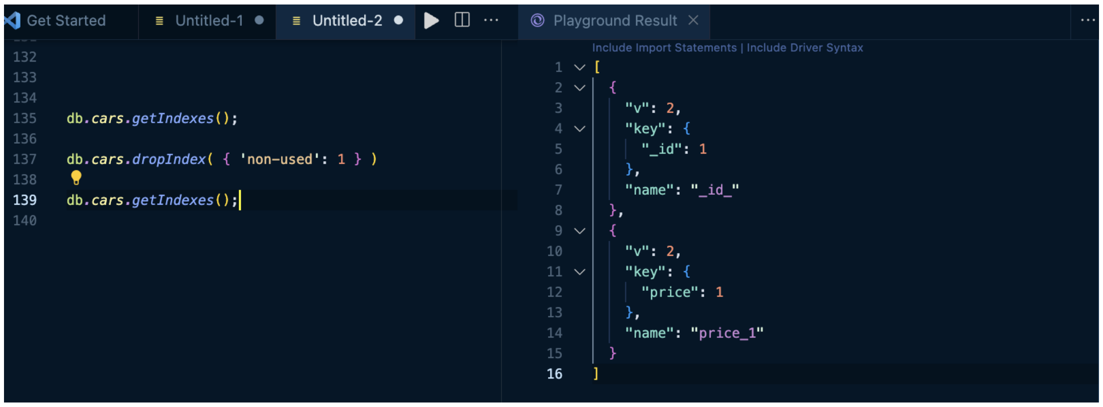


```
db.cars.getIndexes();
```

```
[
  {
    "v": 2,
    "key": {
      "_id": 1
    },
    "name": "_id_"
  },
  {
    "v": 2,
    "key": {
      "price": 1
    },
    "name": "price_1"
  },
  {
    "v": 2,
    "key": {
      "non-used": 1
    },
    "name": "non-used_1"
  }
]
```

```
db.cars.dropIndex( { 'non-used': 1 } )
```

```
db.cars.getIndexes();
```



```
db.cars.getIndexes();

db.cars.dropIndex( { 'non-used': 1 } )

db.cars.getIndexes();
```

```
[
  {
    "v": 2,
    "key": {
      "_id": 1
    },
    "name": "_id_"
  },
  {
    "v": 2,
    "key": {
      "price": 1
    },
    "name": "price_1"
  }
]
```

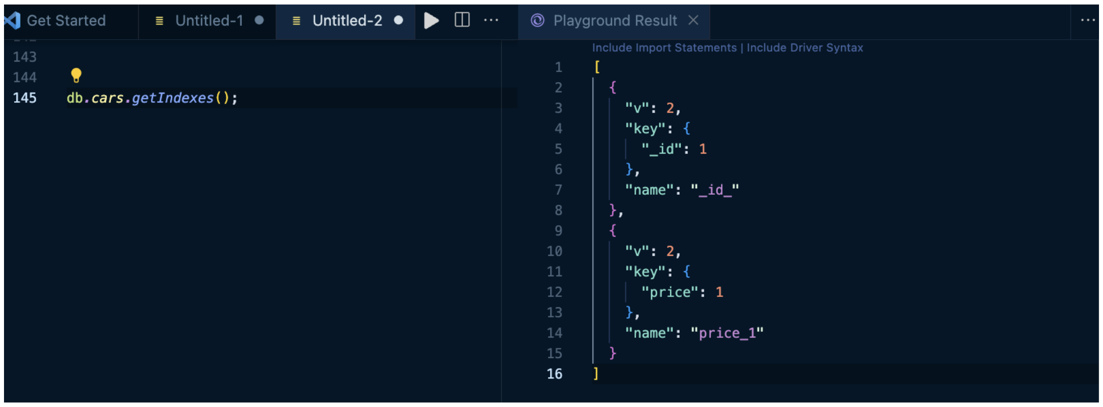
Part 2

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.

2.1 Delete the `price` index

`db.cars.getIndexes();`

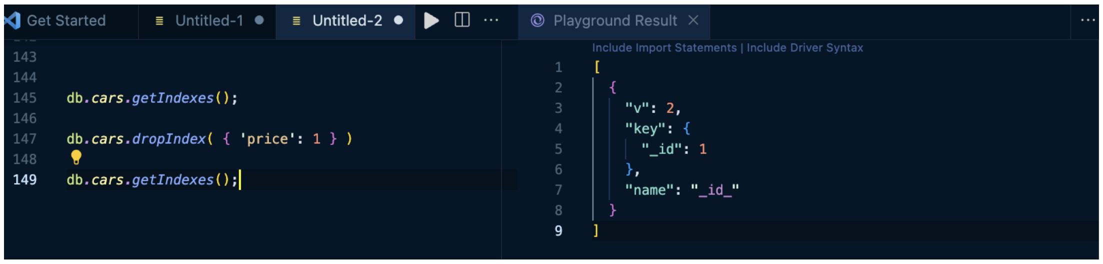


```
db.cars.getIndexes();
```

```
[
  {
    "v": 2,
    "key": {
      "_id": 1
    },
    "name": "_id_"
  },
  {
    "v": 2,
    "key": {
      "price": 1
    },
    "name": "price_1"
  }
]
```

`db.cars.dropIndex( { 'non-used': 1 } )`

`db.cars.getIndexes();`



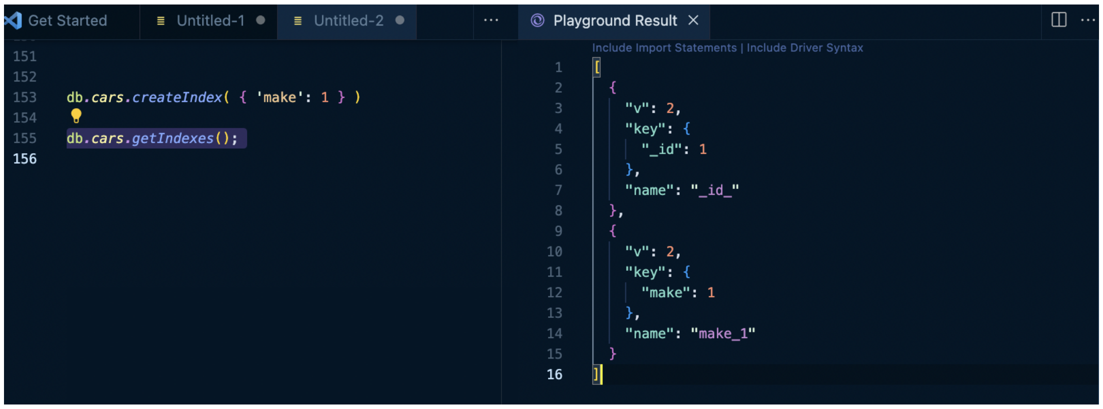
```
db.cars.getIndexes();

db.cars.dropIndex( { 'price': 1 } )

db.cars.getIndexes();
```

```
[
  {
    "v": 2,
    "key": {
      "_id": 1
    },
    "name": "_id_"
  }
]
```

2.2 Create a `make` index

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

```
db.cars.getIndexes();
```

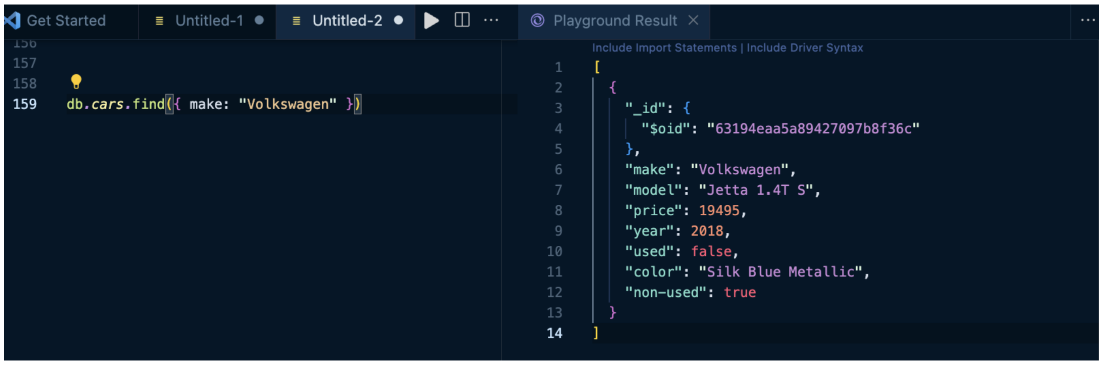


```
db.cars.createIndex( { 'make': 1 } )

db.cars.getIndexes();
```

```
[
  {
    "v": 2,
    "key": {
      "_id": 1
    },
    "name": "_id_"
  },
  {
    "v": 2,
    "key": {
      "make": 1
    },
    "name": "make_1"
  }
]
```

2.3 Delete documents with `make: "Volkswagen"`

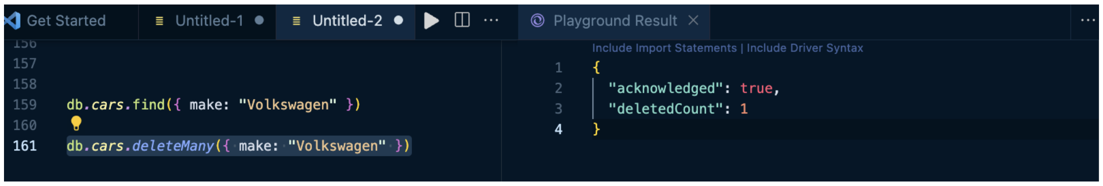
```
db.cars.find({ make: "Volkswagen" })
```



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

```
[
  {
    "_id": {
      "$oid": "63194eaa5a89427097b8f36c"
    },
    "make": "Volkswagen",
    "model": "Jetta 1.4T S",
    "price": 19495,
    "year": 2018,
    "used": false,
    "color": "Silk Blue Metallic",
    "non-used": true
  }
]
```

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



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

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

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