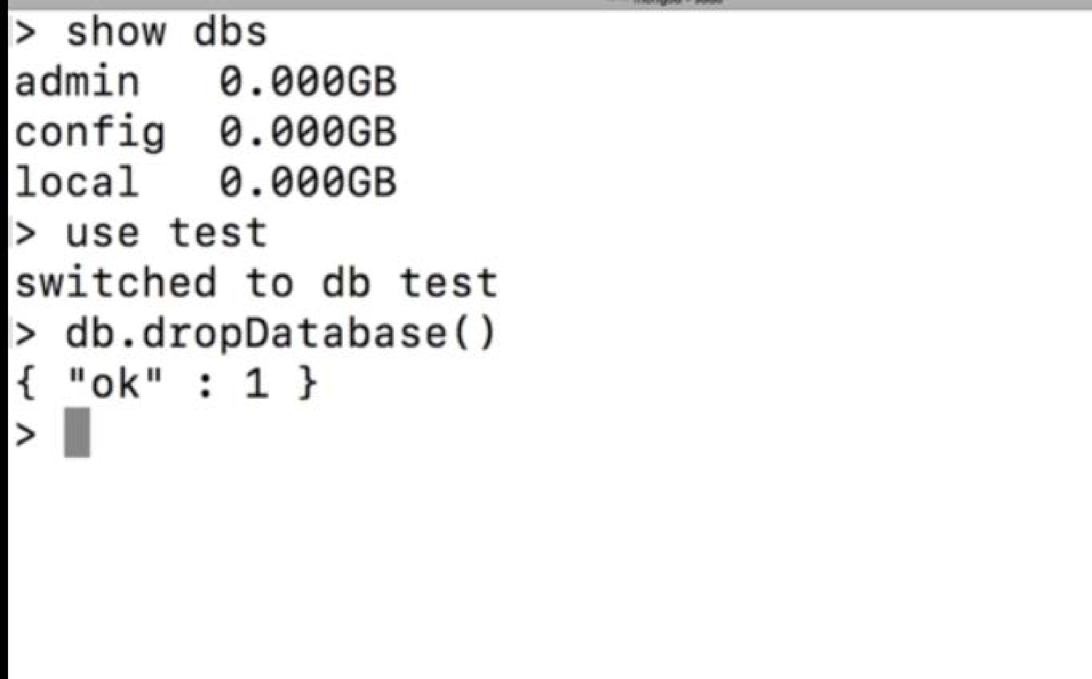
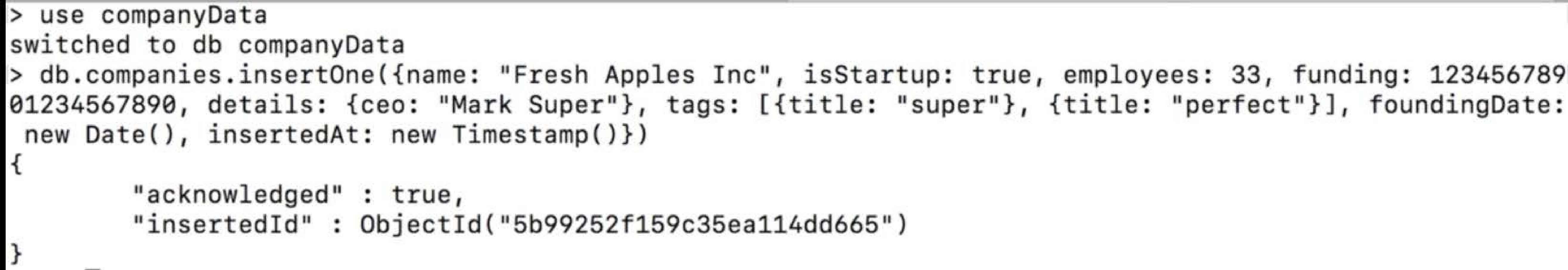
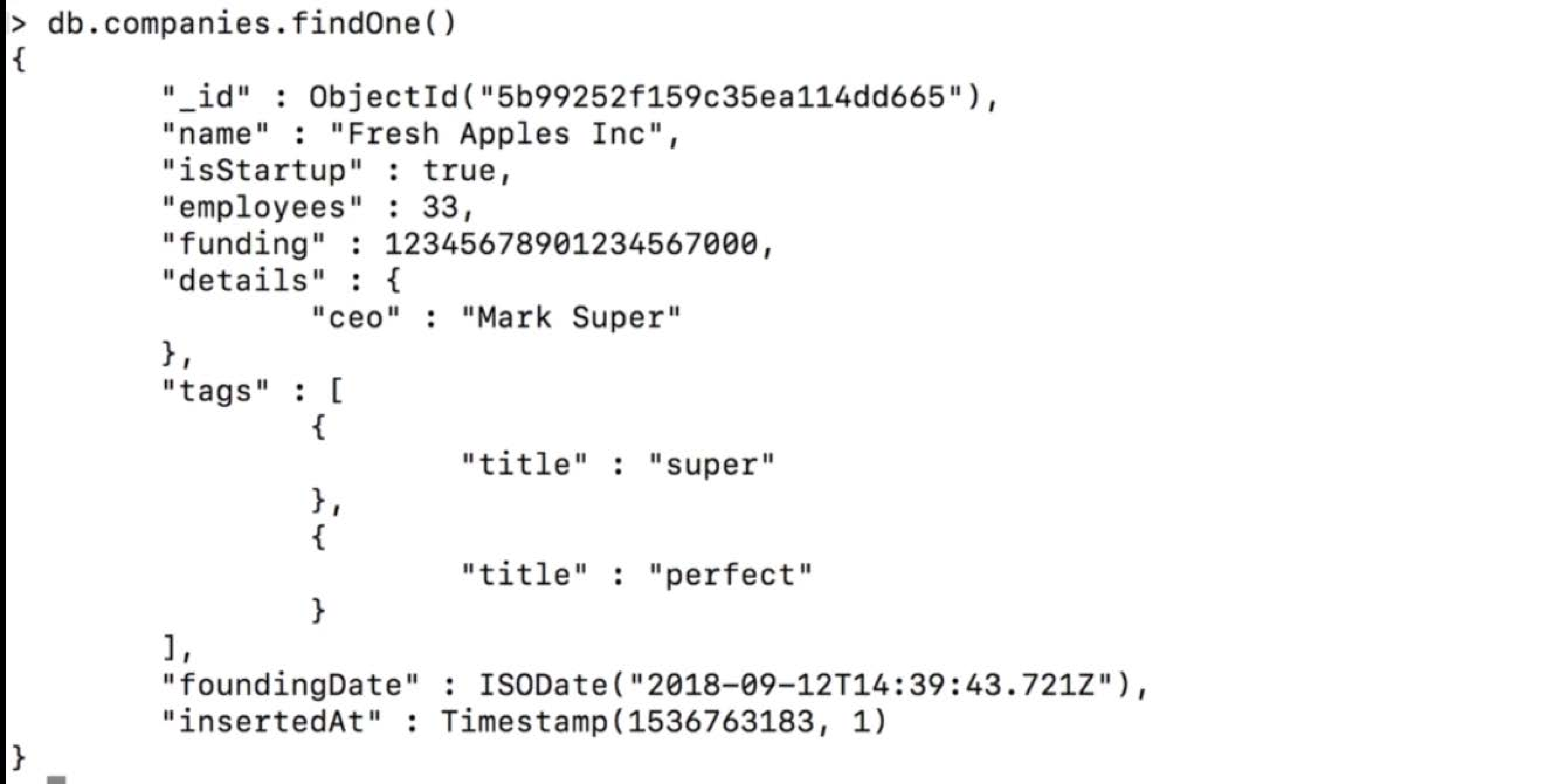


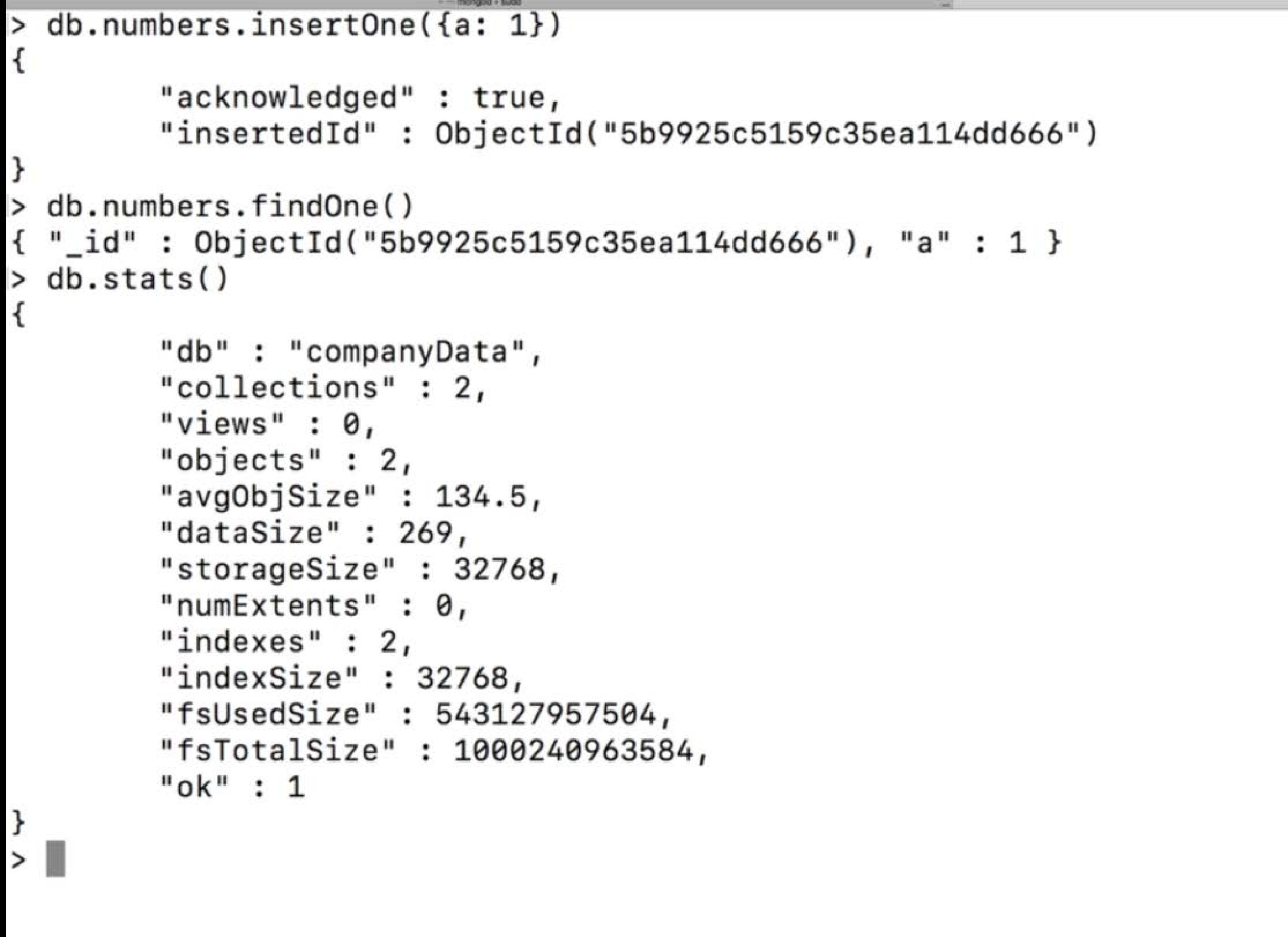
```
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
> use test
switched to db test
> db.dropDatabase()
{ "ok" : 1 }
>
```



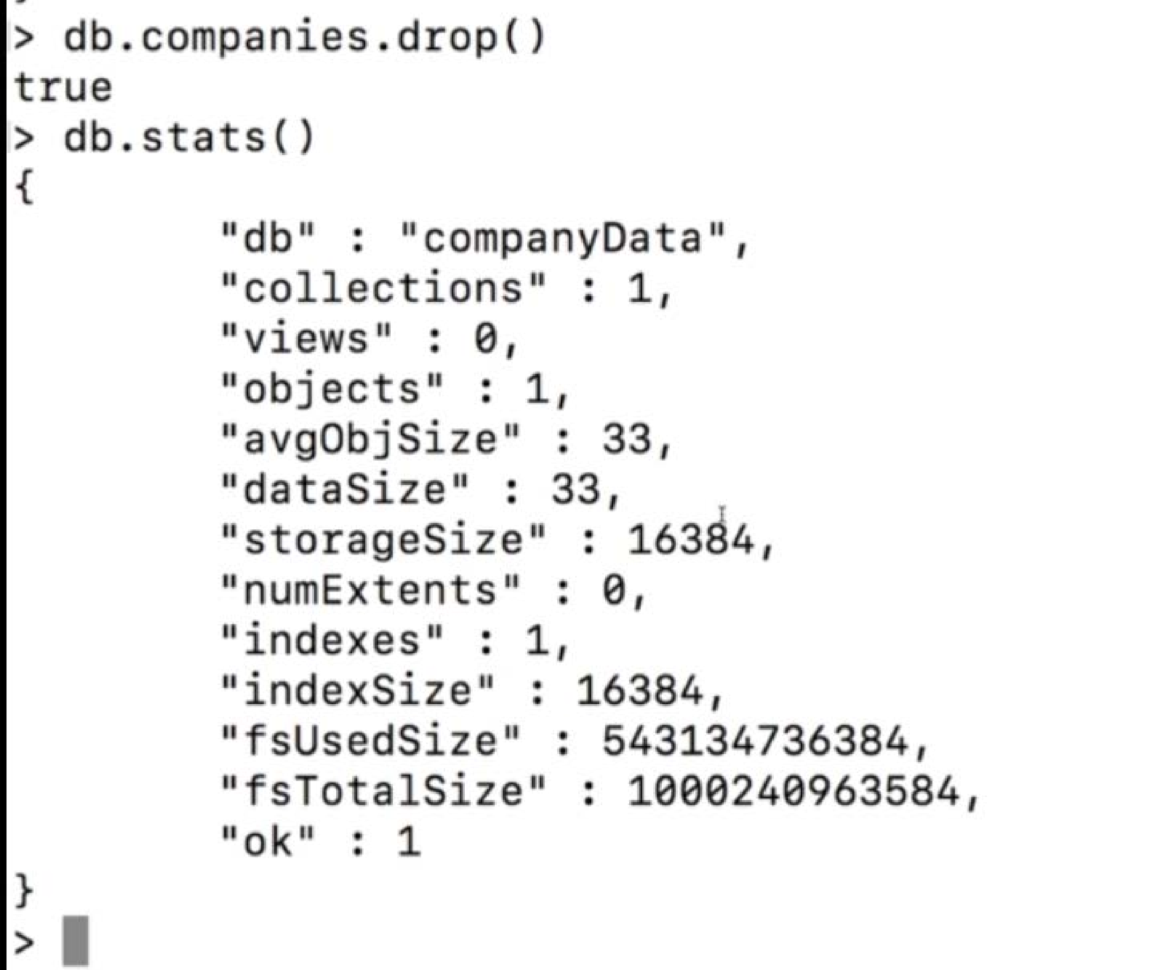
```
> use companyData
switched to db companyData
> db.companies.insertOne({name: "Fresh Apples Inc", isStartup: true, employees: 33, funding: 123456789
01234567890, details: {ceo: "Mark Super"}, tags: [{title: "super"}, {title: "perfect"}], foundingDate:
 new Date(), insertedAt: new Timestamp()})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5b99252f159c35ea114dd665")
}
```



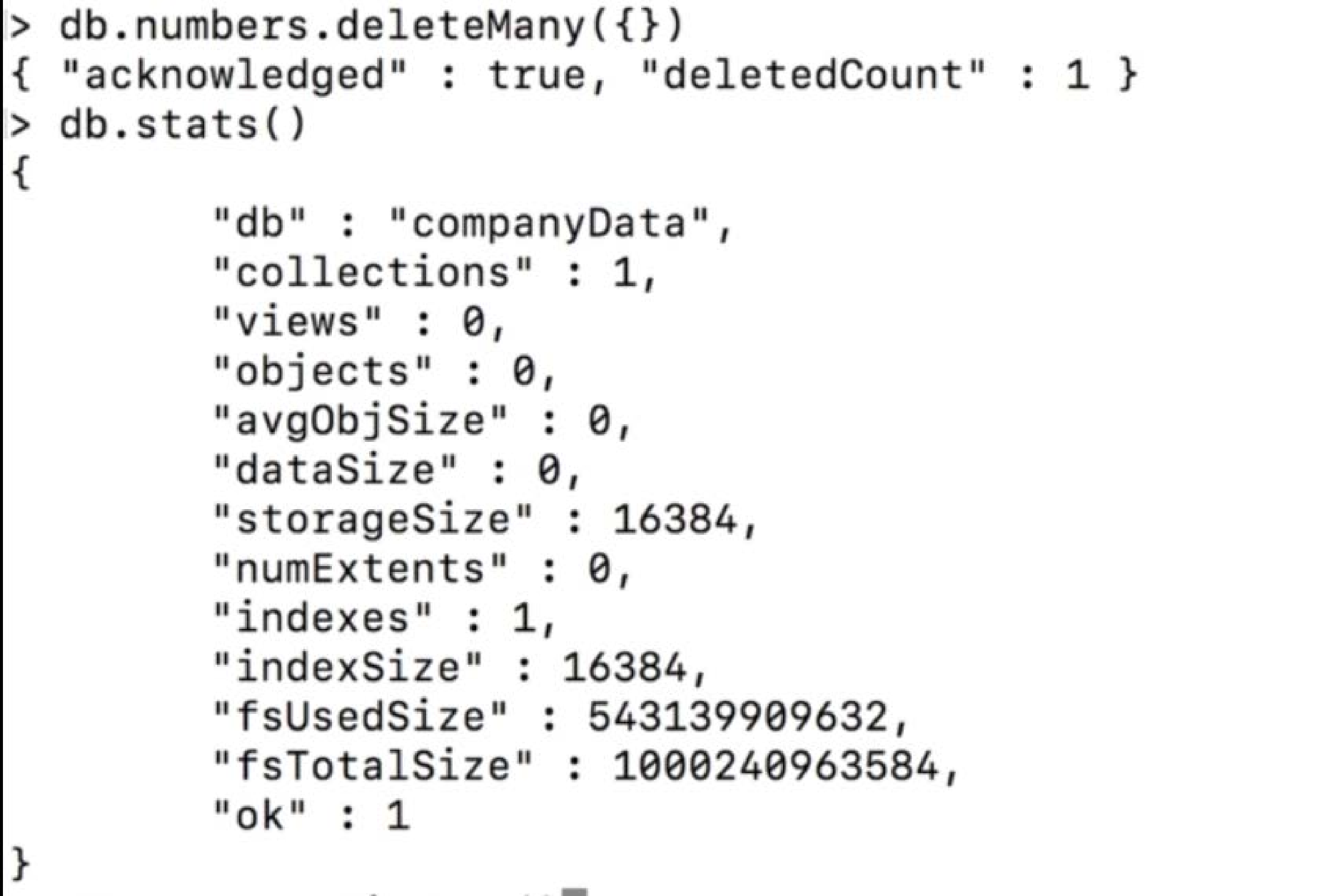
```
> db.companies.findOne()
{
        "_id" : ObjectId("5b99252f159c35ea114dd665"),
        "name" : "Fresh Apples Inc",
        "isStartup" : true,
        "employees" : 33,
        "funding" : 12345678901234567000,
        "details" : {
                "ceo" : "Mark Super"
        },
        "tags" : [
                {
                        "title" : "super"
                },
                {
                        "title" : "perfect"
                }
        ],
        "foundingDate" : ISODate("2018-09-12T14:39:43.721Z"),
        "insertedAt" : Timestamp(1536763183, 1)
}
```



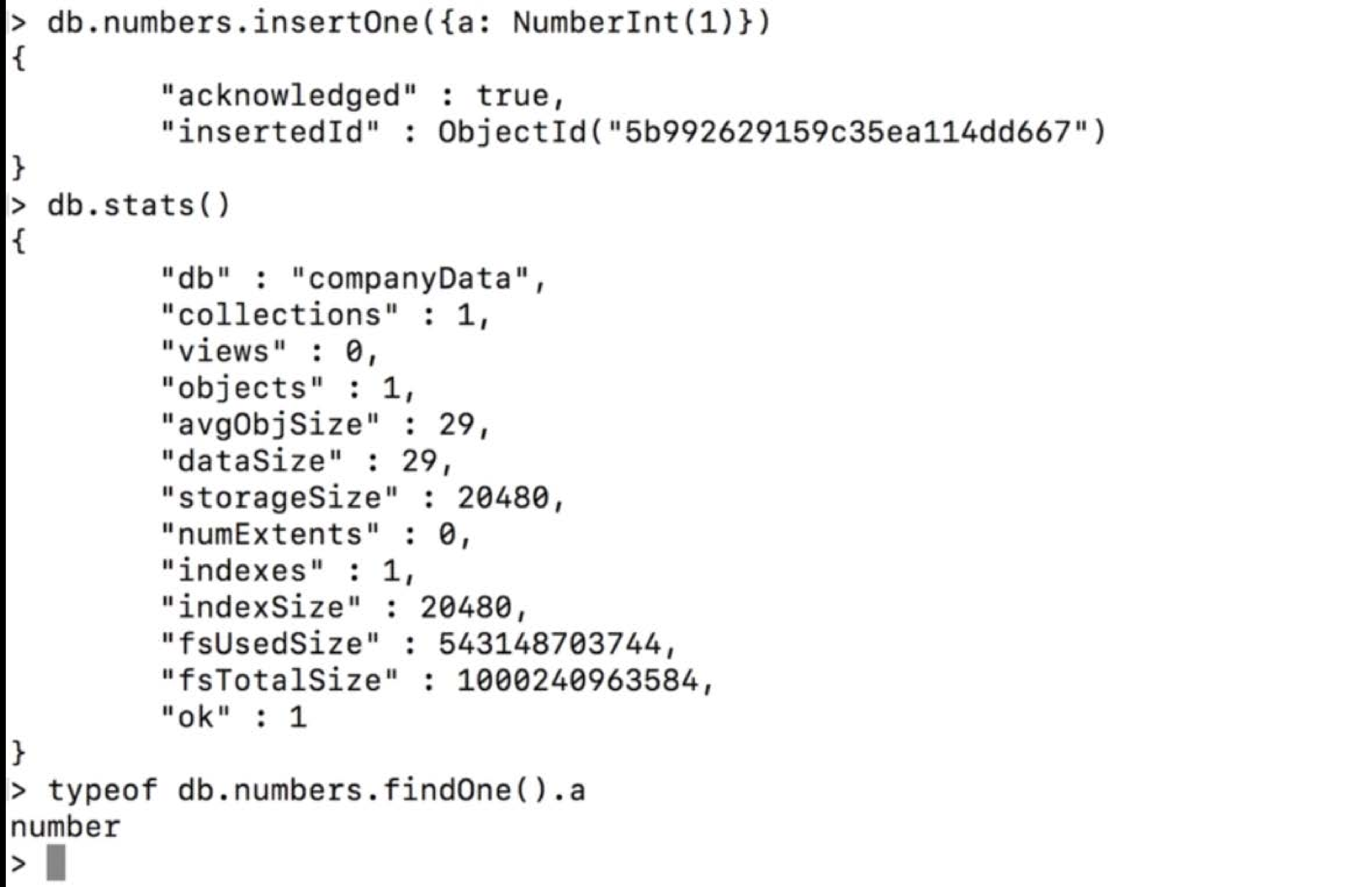
```
> db.numbers.insertOne({a: 1})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5b9925c5159c35ea114dd666")
}
> db.numbers.findOne()
{ "_id" : ObjectId("5b9925c5159c35ea114dd666"), "a" : 1 }
> db.stats()
{
        "db" : "companyData",
        "collections" : 2,
        "views" : 0,
        "objects" : 2,
        "avgObjSize" : 134.5,
        "dataSize" : 269,
        "storageSize" : 32768,
        "numExtents" : 0,
        "indexes" : 2,
        "indexSize" : 32768,
        "fsUsedSize" : 543127957504,
        "fsTotalSize" : 1000240963584,
        "ok" : 1
}
>
```



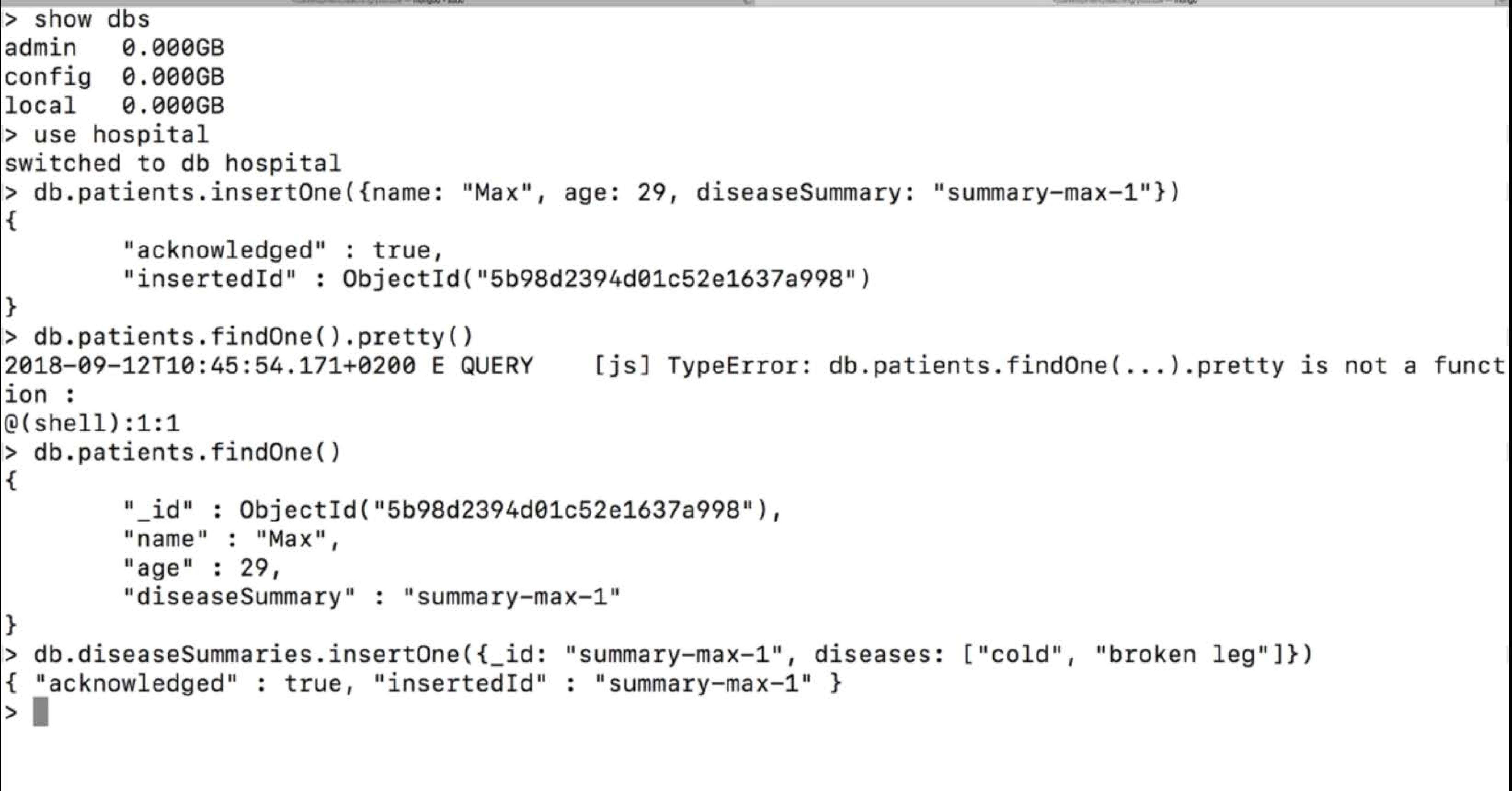
```
> db.companies.drop()
true
> db.stats()
{
        "db" : "companyData",
        "collections" : 1,
        "views" : 0,
        "objects" : 1,
        "avgObjSize" : 33,
        "dataSize" : 33,
        "storageSize" : 16384,
        "numExtents" : 0,
        "indexes" : 1,
        "indexSize" : 16384,
        "fsUsedSize" : 543134736384,
        "fsTotalSize" : 1000240963584,
        "ok" : 1
}
> 
```



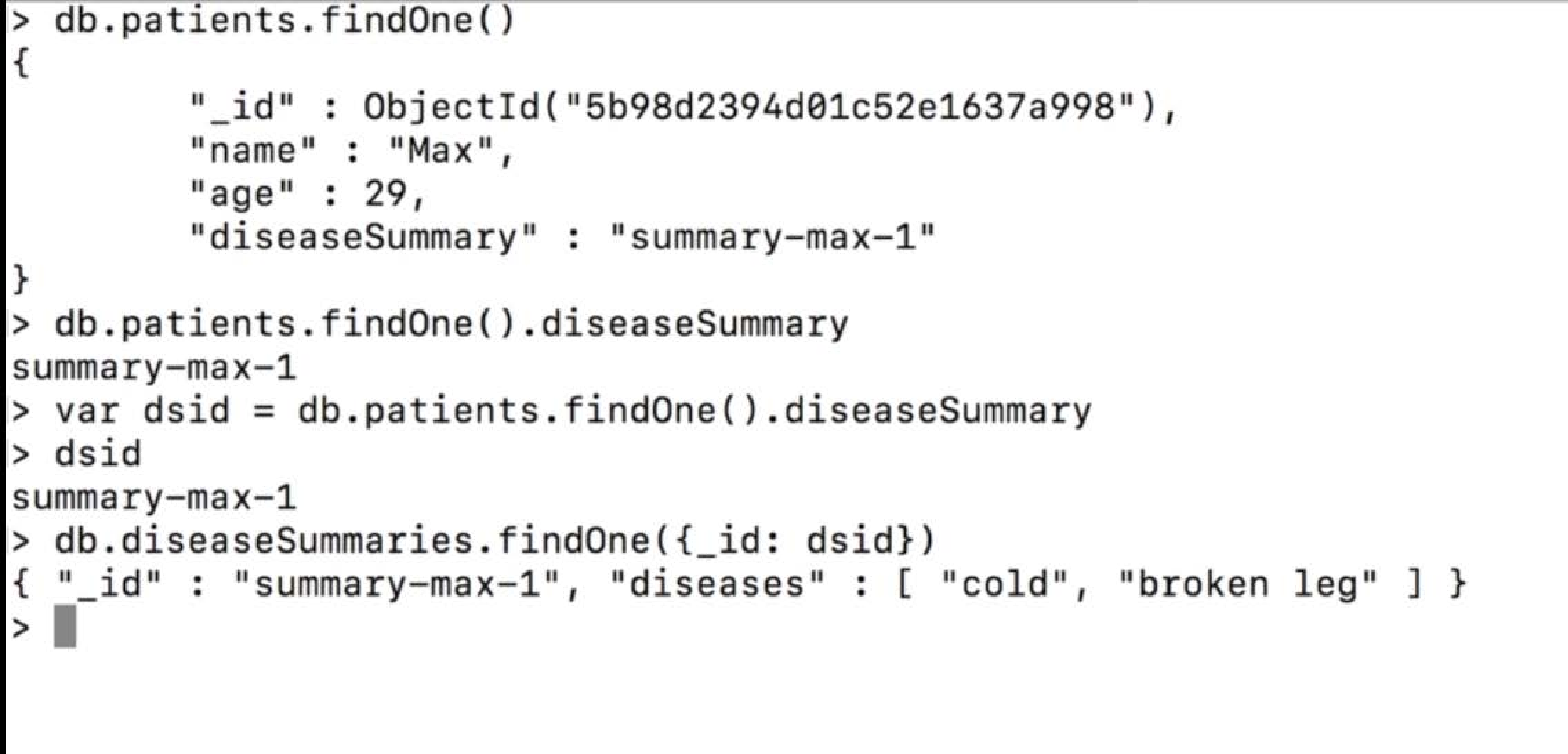
```
> db.numbers.deleteMany({})
{ "acknowledged" : true, "deletedCount" : 1 }
> db.stats()
{
        "db" : "companyData",
        "collections" : 1,
        "views" : 0,
        "objects" : 0,
        "avgObjSize" : 0,
        "dataSize" : 0,
        "storageSize" : 16384,
        "numExtents" : 0,
        "indexes" : 1,
        "indexSize" : 16384,
        "fsUsedSize" : 543139909632,
        "fsTotalSize" : 1000240963584,
        "ok" : 1
}
```



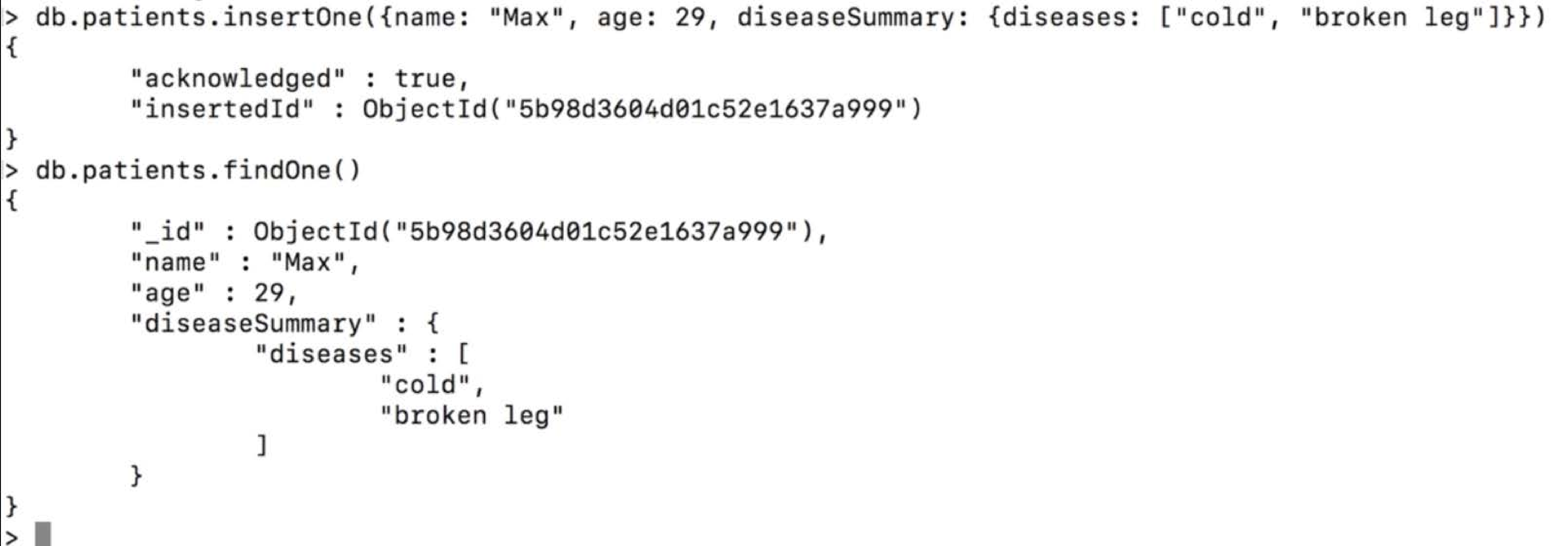
```
> db.numbers.insertOne({a: NumberInt(1)})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5b992629159c35ea114dd667")
}
> db.stats()
{
        "db" : "companyData",
        "collections" : 1,
        "views" : 0,
        "objects" : 1,
        "avgObjSize" : 29,
        "dataSize" : 29,
        "storageSize" : 20480,
        "numExtents" : 0,
        "indexes" : 1,
        "indexSize" : 20480,
        "fsUsedSize" : 543148703744,
        "fsTotalSize" : 1000240963584,
        "ok" : 1
}
> typeof db.numbers.findOne().a
number
> 
```



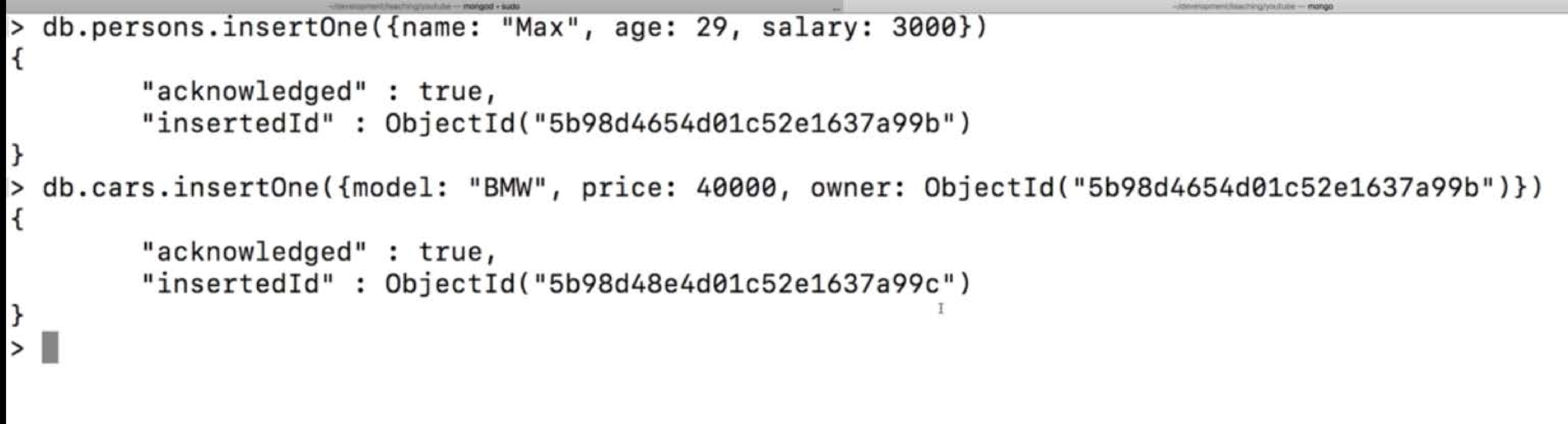
```
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
> use hospital
switched to db hospital
> db.patients.insertOne({name: "Max", age: 29, diseaseSummary: "summary-max-1"})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5b98d2394d01c52e1637a998")
}
> db.patients.findOne().pretty()
2018-09-12T10:45:54.171+0200 E QUERY    [js] TypeError: db.patients.findOne(...).pretty is not a funct
ion :
@(shell):1:1
> db.patients.findOne()
{
        "_id" : ObjectId("5b98d2394d01c52e1637a998"),
        "name" : "Max",
        "age" : 29,
        "diseaseSummary" : "summary-max-1"
}
> db.diseaseSummaries.insertOne({_id: "summary-max-1", diseases: ["cold", "broken leg"]})
{ "acknowledged" : true, "insertedId" : "summary-max-1" }
> 
```



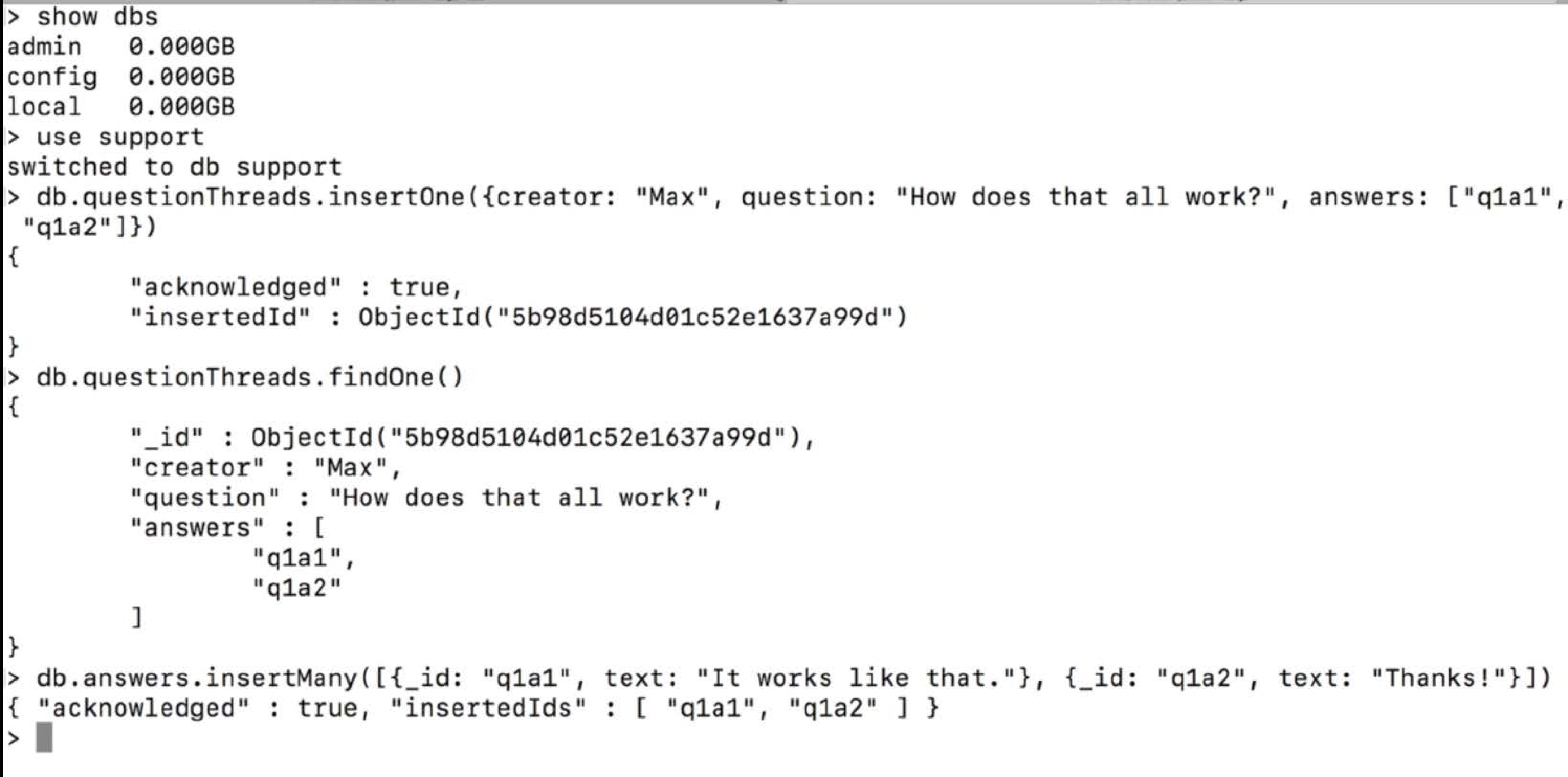
```
> db.patients.findOne()
{
        "_id" : ObjectId("5b98d2394d01c52e1637a998"),
        "name" : "Max",
        "age" : 29,
        "diseaseSummary" : "summary-max-1"
}
> db.patients.findOne().diseaseSummary
summary-max-1
> var dsid = db.patients.findOne().diseaseSummary
> dsid
summary-max-1
> db.diseaseSummaries.findOne({_id: dsid})
{ "_id" : "summary-max-1", "diseases" : [ "cold", "broken leg" ] }
> 
```

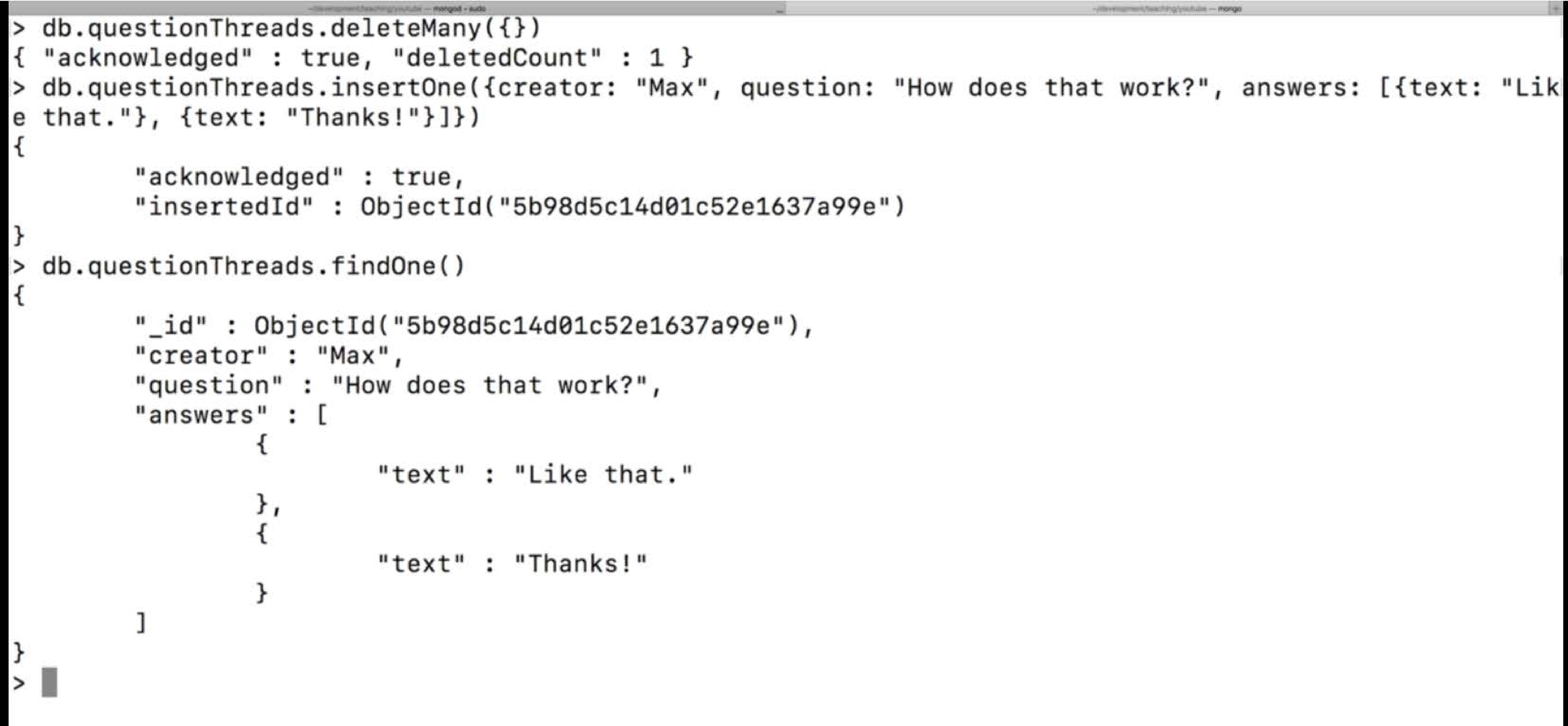


```
> db.patients.insertOne({name: "Max", age: 29, diseaseSummary: {diseases: ["cold", "broken leg"]}})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5b98d3604d01c52e1637a999")
}
> db.patients.findOne()
{
        "_id" : ObjectId("5b98d3604d01c52e1637a999"),
        "name" : "Max",
        "age" : 29,
        "diseaseSummary" : {
                "diseases" : [
                        "cold",
                        "broken leg"
                ]
        }
}
> 
```

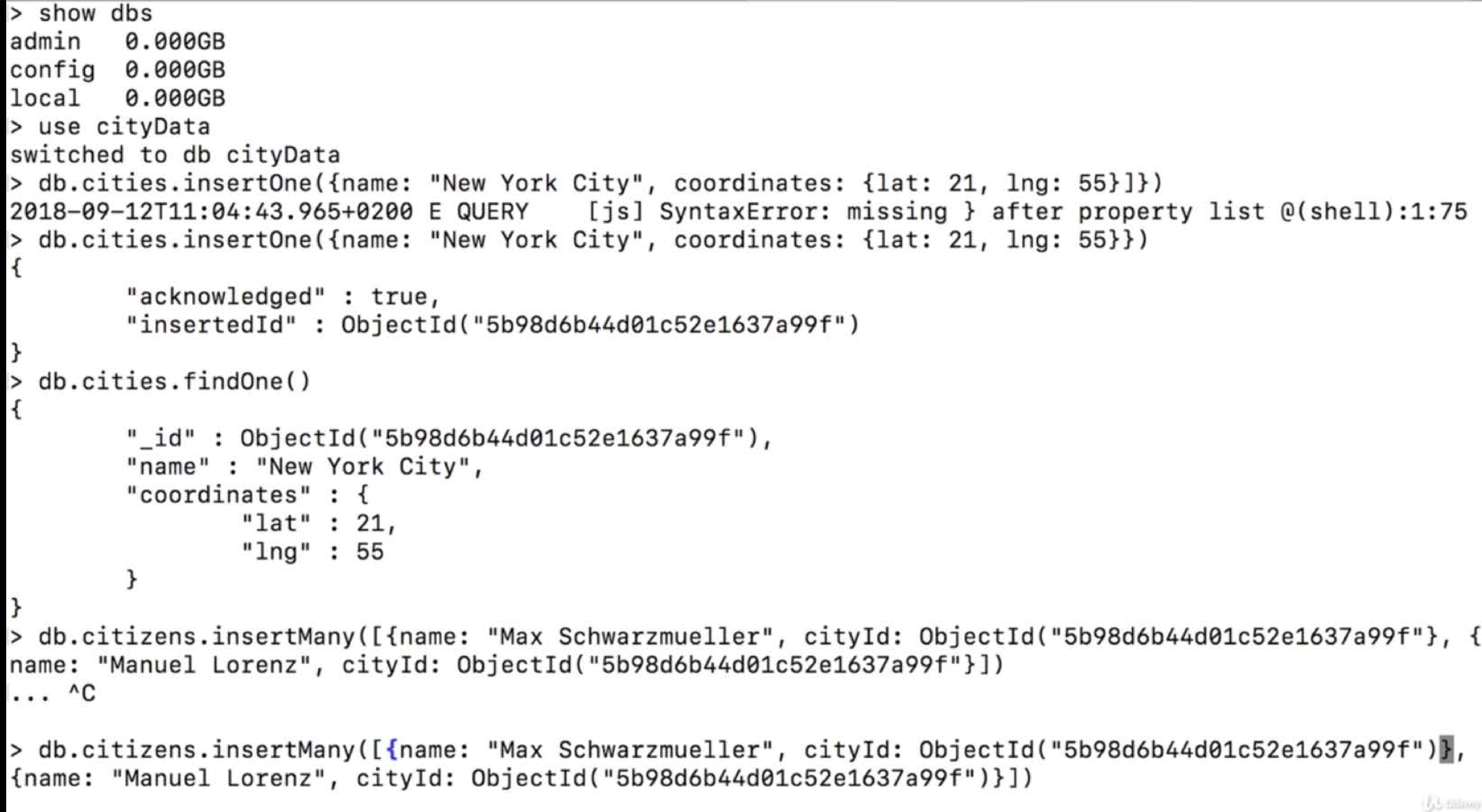


```
> db.persons.insertOne({name: "Max", age: 29, salary: 3000})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5b98d4654d01c52e1637a99b")
}
> db.cars.insertOne({model: "BMW", price: 40000, owner: ObjectId("5b98d4654d01c52e1637a99b")})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5b98d48e4d01c52e1637a99c")
}
> 
```


```
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
> use support
switched to db support
> db.questionThreads.insertOne({creator: "Max", question: "How does that all work?", answers: ["q1a1",
 "q1a2"]})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5b98d5104d01c52e1637a99d")
}
> db.questionThreads.findOne()
{
        "_id" : ObjectId("5b98d5104d01c52e1637a99d"),
        "creator" : "Max",
        "question" : "How does that all work?",
        "answers" : [
                "q1a1",
                "q1a2"
        ]
}
> db.answers.insertMany([{_id: "q1a1", text: "It works like that."}, {_id: "q1a2", text: "Thanks!"}])
{ "acknowledged" : true, "insertedIds" : [ "q1a1", "q1a2" ] }
> 
```

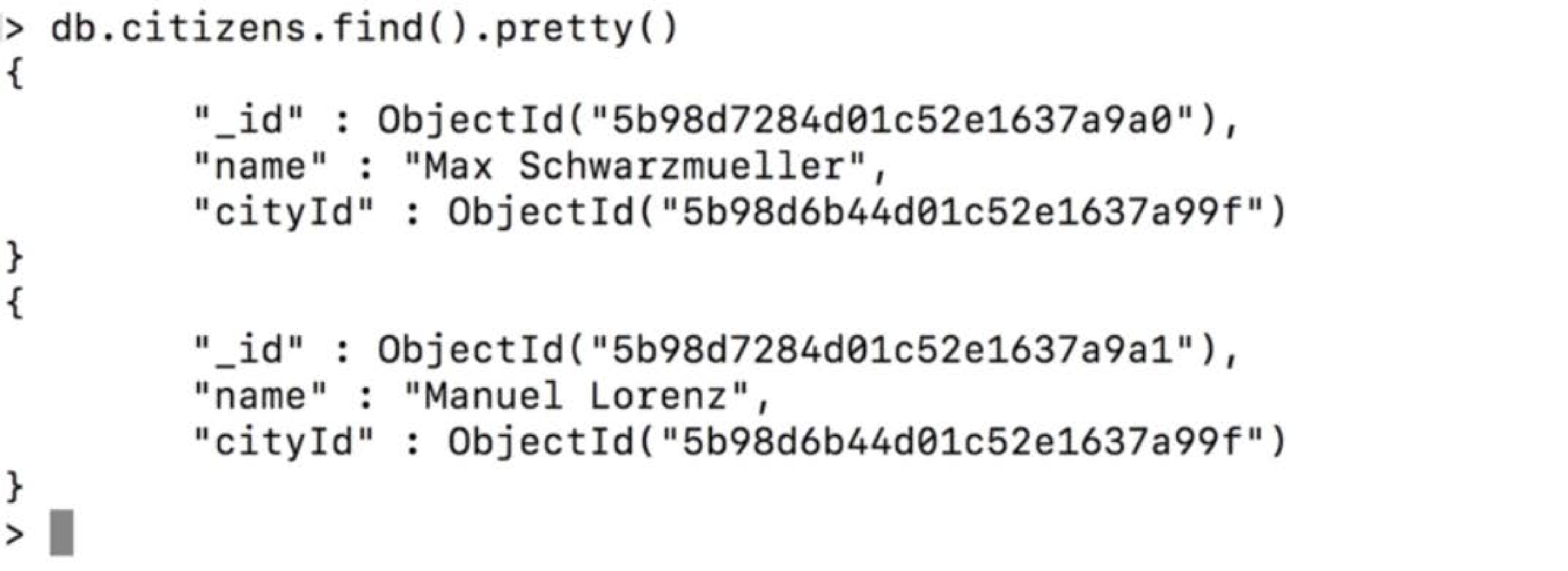


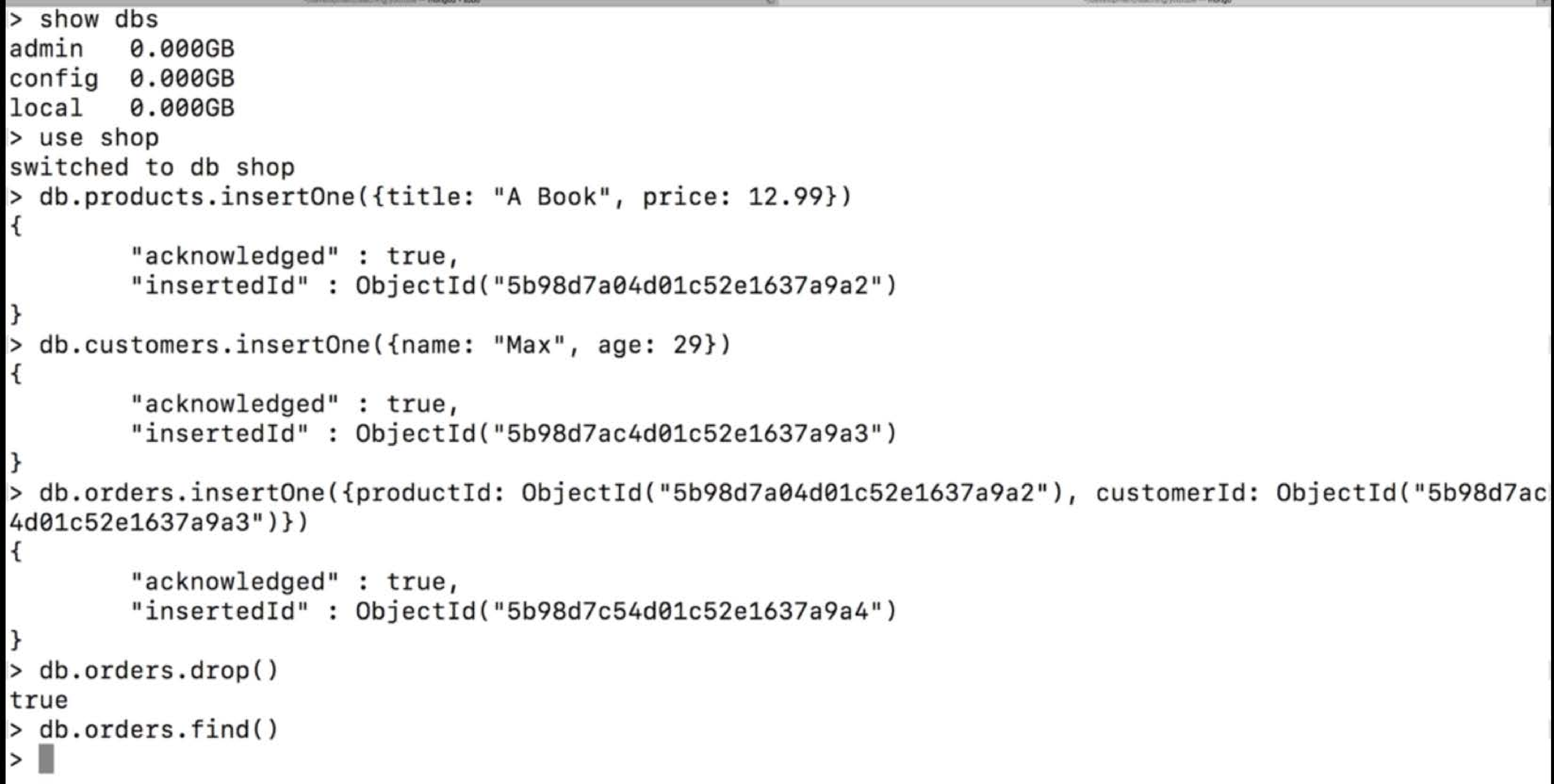
```
> db.questionThreads.deleteMany({})
{ "acknowledged" : true, "deletedCount" : 1 }
> db.questionThreads.insertOne({creator: "Max", question: "How does that work?", answers: [{text: "Lik
e that."}, {text: "Thanks!"}]})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5b98d5c14d01c52e1637a99e")
}
> db.questionThreads.findOne()
{
        "_id" : ObjectId("5b98d5c14d01c52e1637a99e"),
        "creator" : "Max",
        "question" : "How does that work?",
        "answers" : [
                {
                        "text" : "Like that."
                },
                {
                        "text" : "Thanks!"
                }
        ]
}
> 
```



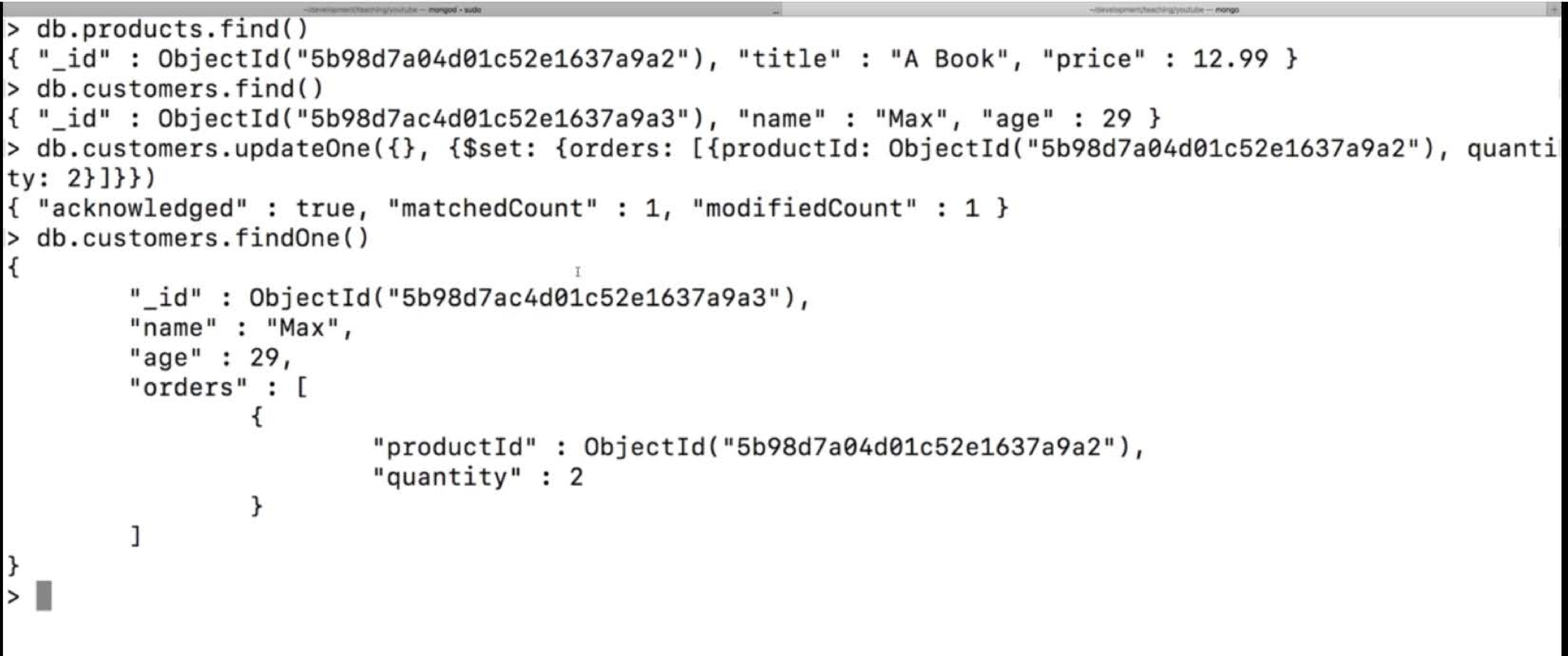
```
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
> use cityData
switched to db cityData
> db.cities.insertOne({name: "New York City", coordinates: {lat: 21, lng: 55}]})
2018-09-12T11:04:43.965+0200 E QUERY    [js] SyntaxError: missing } after property list @(shell):1:75
> db.cities.insertOne({name: "New York City", coordinates: {lat: 21, lng: 55}})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5b98d6b44d01c52e1637a99f")
}
> db.cities.findOne()
{
        "_id" : ObjectId("5b98d6b44d01c52e1637a99f"),
        "name" : "New York City",
        "coordinates" : {
                "lat" : 21,
                "lng" : 55
        }
}
> db.citizens.insertMany([{name: "Max Schwarzmueller", cityId: ObjectId("5b98d6b44d01c52e1637a99f"}, {
name: "Manuel Lorenz", cityId: ObjectId("5b98d6b44d01c52e1637a99f"}])
... ^C

> db.citizens.insertMany([{name: "Max Schwarzmueller", cityId: ObjectId("5b98d6b44d01c52e1637a99f")},
{name: "Manuel Lorenz", cityId: ObjectId("5b98d6b44d01c52e1637a99f")}])
```

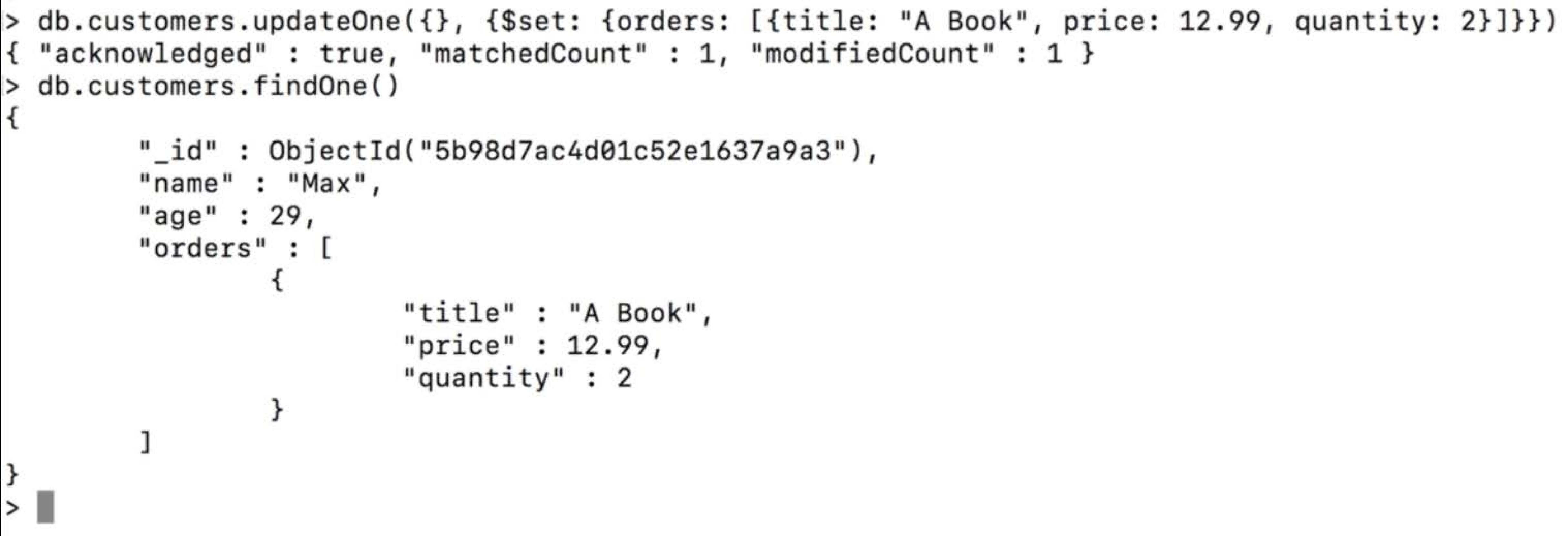

```
> db.citizens.find().pretty()
{
        "_id" : ObjectId("5b98d7284d01c52e1637a9a0"),
        "name" : "Max Schwarzmueller",
        "cityId" : ObjectId("5b98d6b44d01c52e1637a99f")
}
{
        "_id" : ObjectId("5b98d7284d01c52e1637a9a1"),
        "name" : "Manuel Lorenz",
        "cityId" : ObjectId("5b98d6b44d01c52e1637a99f")
}
>
```



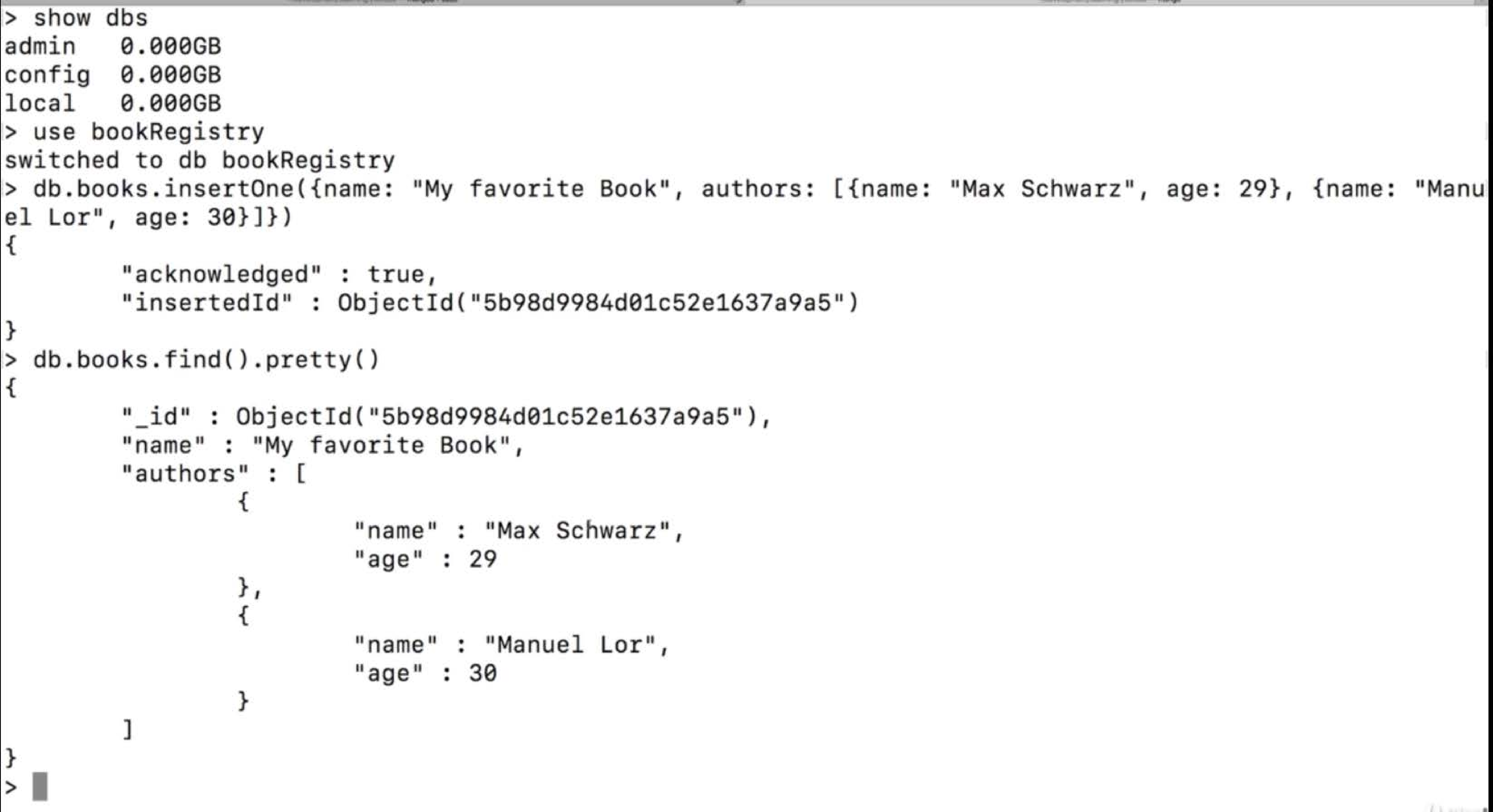
```
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
> use shop
switched to db shop
> db.products.insertOne({title: "A Book", price: 12.99})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5b98d7a04d01c52e1637a9a2")
}
> db.customers.insertOne({name: "Max", age: 29})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5b98d7ac4d01c52e1637a9a3")
}
> db.orders.insertOne({productId: ObjectId("5b98d7a04d01c52e1637a9a2"), customerId: ObjectId("5b98d7ac
4d01c52e1637a9a3")})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5b98d7c54d01c52e1637a9a4")
}
> db.orders.drop()
true
> db.orders.find()
> 
```



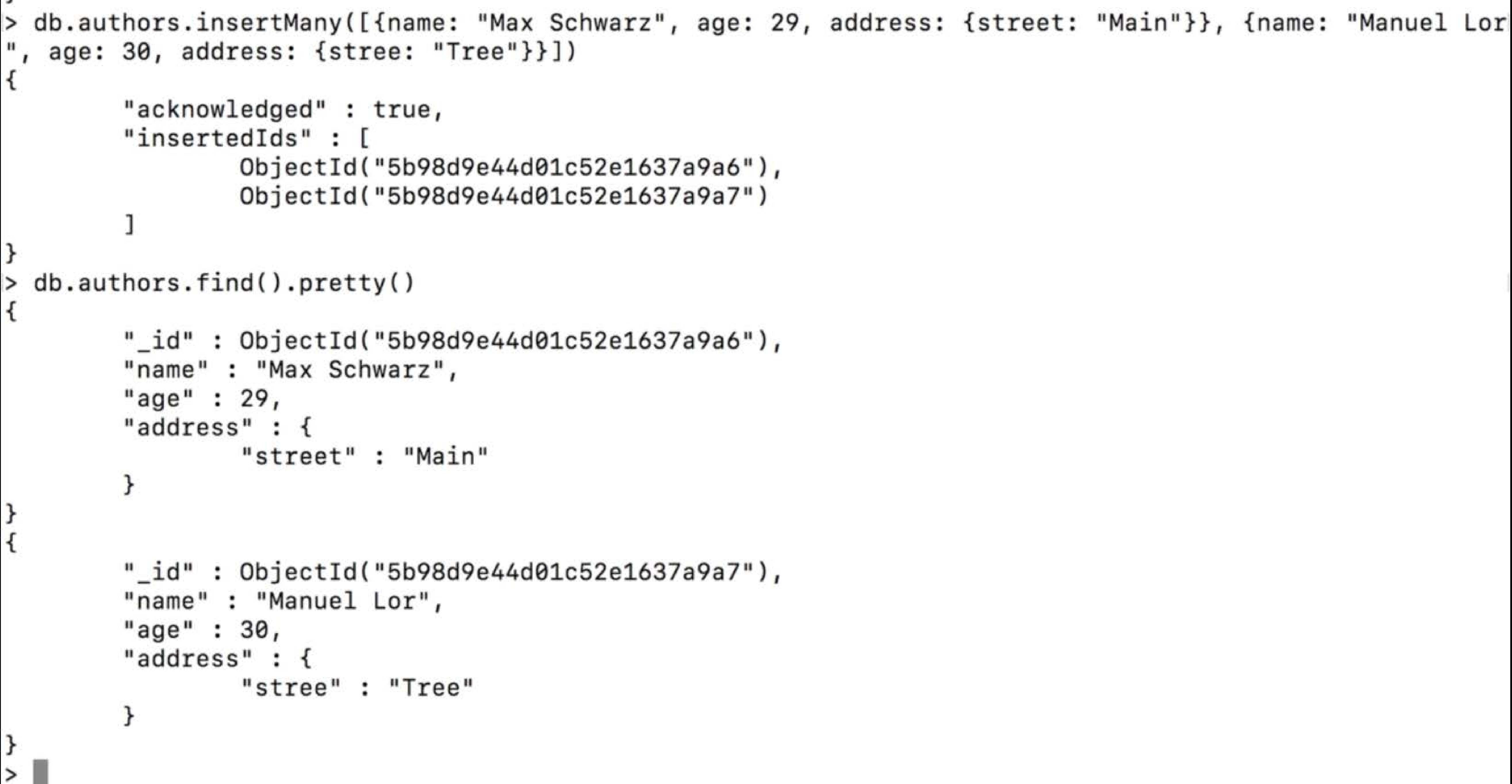
```
> db.products.find()
{ "_id" : ObjectId("5b98d7a04d01c52e1637a9a2"), "title" : "A Book", "price" : 12.99 }
> db.customers.find()
{ "_id" : ObjectId("5b98d7ac4d01c52e1637a9a3"), "name" : "Max", "age" : 29 }
> db.customers.updateOne({}, {$set: {orders: [{productId: ObjectId("5b98d7a04d01c52e1637a9a2"), quanti
ty: 2}]}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.customers.findOne()
{
        "_id" : ObjectId("5b98d7ac4d01c52e1637a9a3"),
        "name" : "Max",
        "age" : 29,
        "orders" : [
                {
                        "productId" : ObjectId("5b98d7a04d01c52e1637a9a2"),
                        "quantity" : 2
                }
        ]
}
> 
```



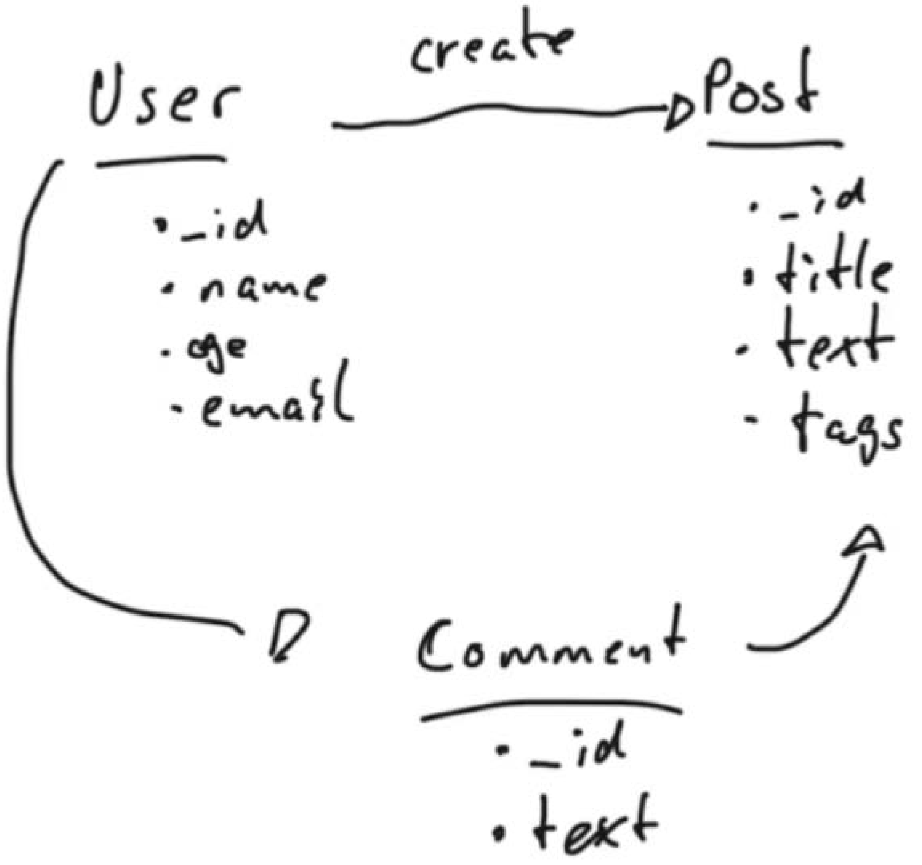
```
> db.customers.updateOne({}, {$set: {orders: [{title: "A Book", price: 12.99, quantity: 2}]}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.customers.findOne()
{
        "_id" : ObjectId("5b98d7ac4d01c52e1637a9a3"),
        "name" : "Max",
        "age" : 29,
        "orders" : [
                {
                        "title" : "A Book",
                        "price" : 12.99,
                        "quantity" : 2
                }
        ]
}
> 
```

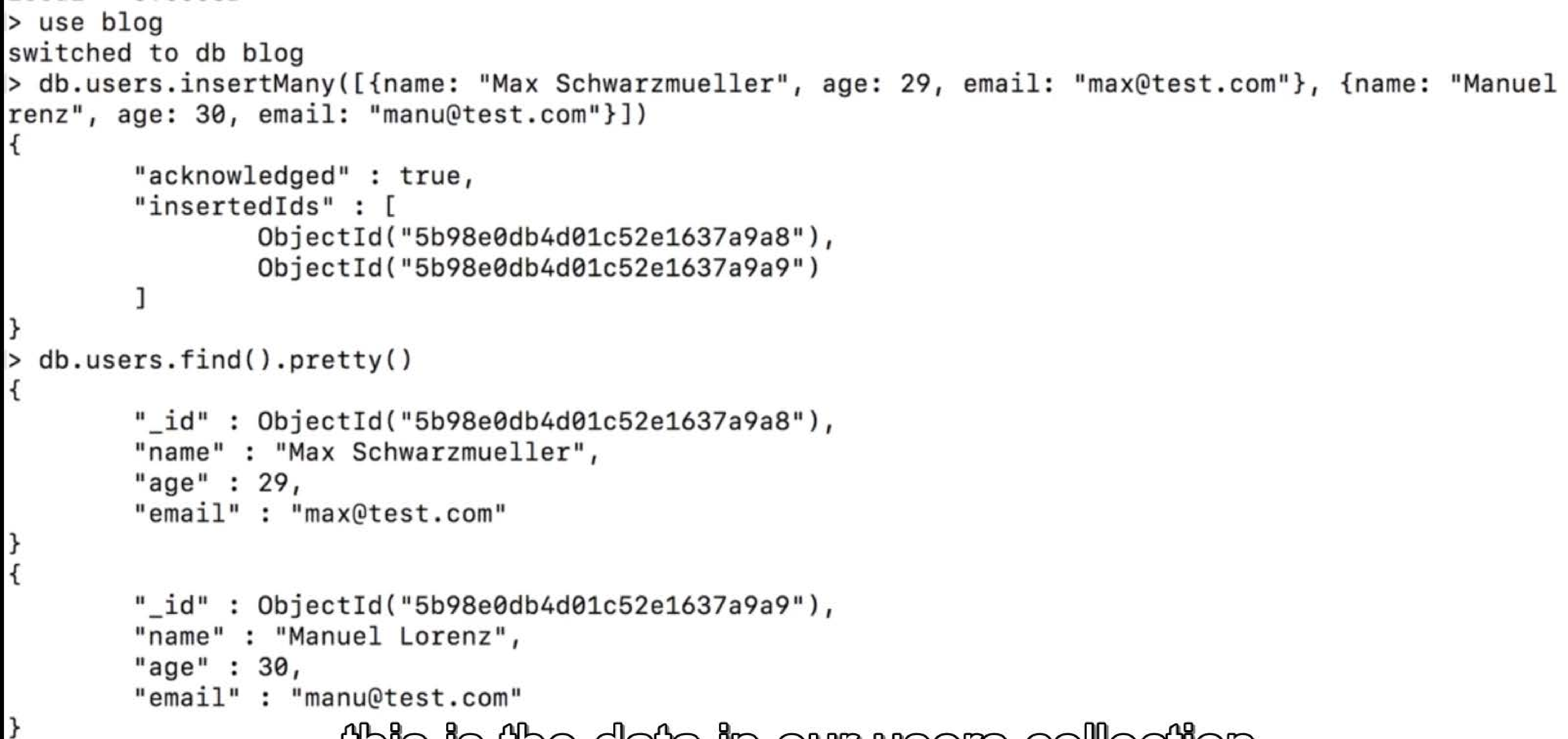


```
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
> use bookRegistry
switched to db bookRegistry
> db.books.insertOne({name: "My favorite Book", authors: [{name: "Max Schwarz", age: 29}, {name: "Manu
el Lor", age: 30}]})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5b98d9984d01c52e1637a9a5")
}
> db.books.find().pretty()
{
        "_id" : ObjectId("5b98d9984d01c52e1637a9a5"),
        "name" : "My favorite Book",
        "authors" : [
                {
                        "name" : "Max Schwarz",
                        "age" : 29
                },
                {
                        "name" : "Manuel Lor",
                        "age" : 30
                }
        ]
}
> 
```



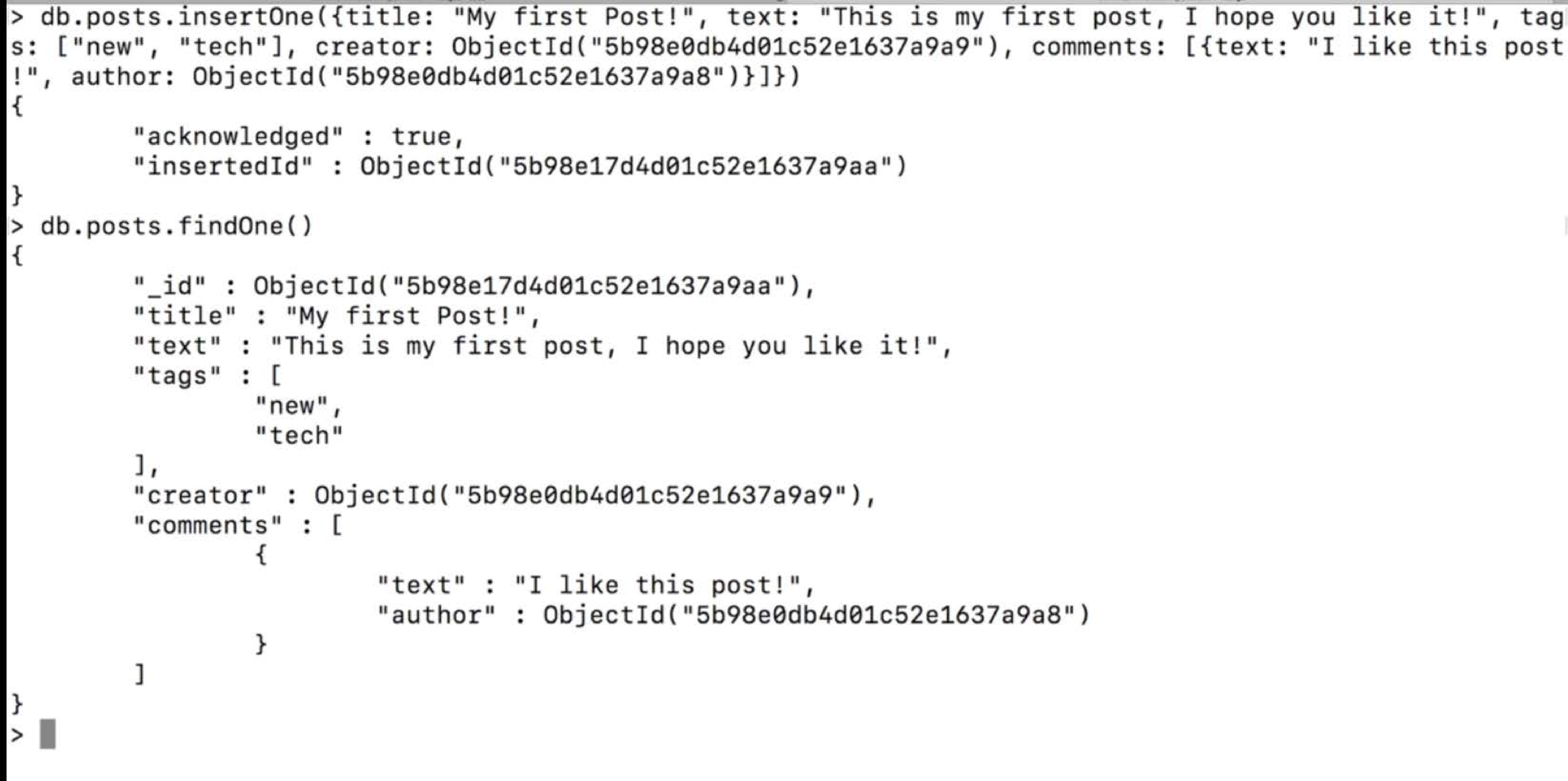
```
> db.authors.insertMany([{name: "Max Schwarz", age: 29, address: {street: "Main"}}, {name: "Manuel Lor
", age: 30, address: {stree: "Tree"}}])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("5b98d9e44d01c52e1637a9a6"),
                ObjectId("5b98d9e44d01c52e1637a9a7")
        ]
}
> db.authors.find().pretty()
{
        "_id" : ObjectId("5b98d9e44d01c52e1637a9a6"),
        "name" : "Max Schwarz",
        "age" : 29,
        "address" : {
                "street" : "Main"
        }
}
{
        "_id" : ObjectId("5b98d9e44d01c52e1637a9a7"),
        "name" : "Manuel Lor",
        "age" : 30,
        "address" : {
                "stree" : "Tree"
        }
}
>
```


create
User
• _id
• name
• age
• email
Post
• _id
• title
• text
• tags
Comment
• _id
• text

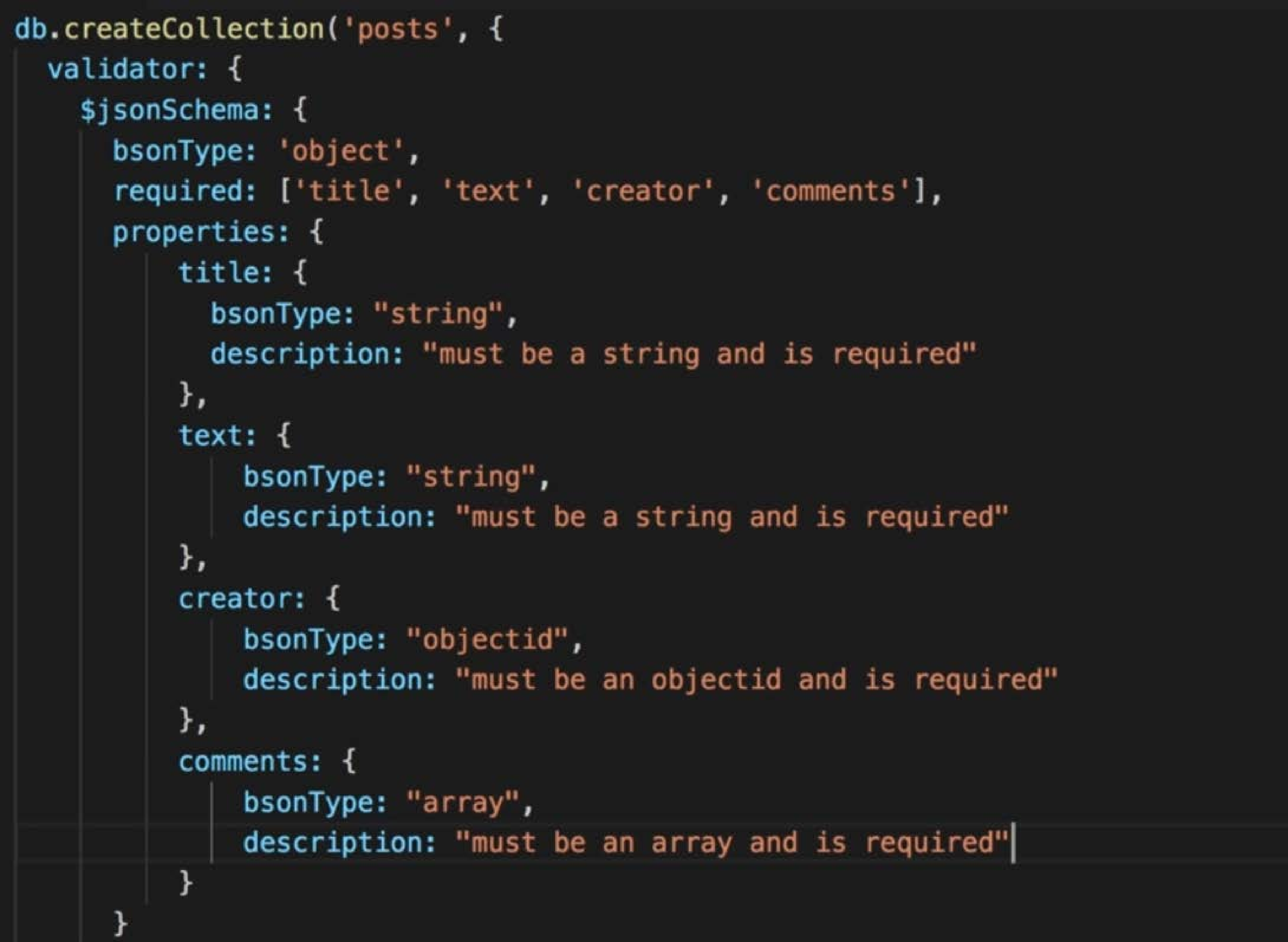


```
> use blog
switched to db blog
> db.users.insertMany([{name: "Max Schwarzmueller", age: 29, email: "max@test.com"}, {name: "Manuel
renz", age: 30, email: "manu@test.com"}])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("5b98e0db4d01c52e1637a9a8"),
                ObjectId("5b98e0db4d01c52e1637a9a9")
        ]
}
> db.users.find().pretty()
{
        "_id" : ObjectId("5b98e0db4d01c52e1637a9a8"),
        "name" : "Max Schwarzmueller",
        "age" : 29,
        "email" : "max@test.com"
}
{
        "_id" : ObjectId("5b98e0db4d01c52e1637a9a9"),
        "name" : "Manuel Lorenz",
        "age" : 30,
        "email" : "manu@test.com"
}
```

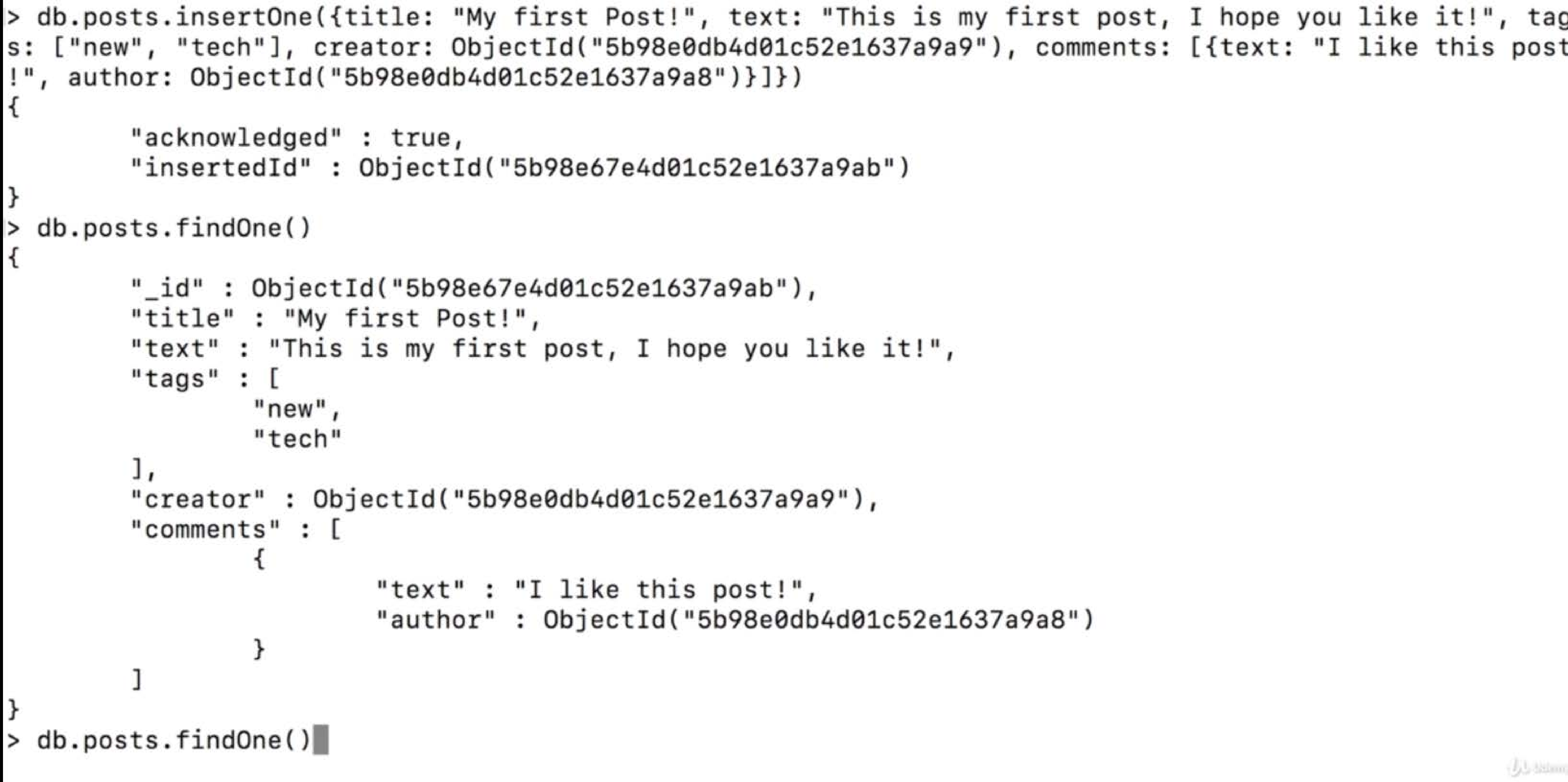
this is the data in our users collection



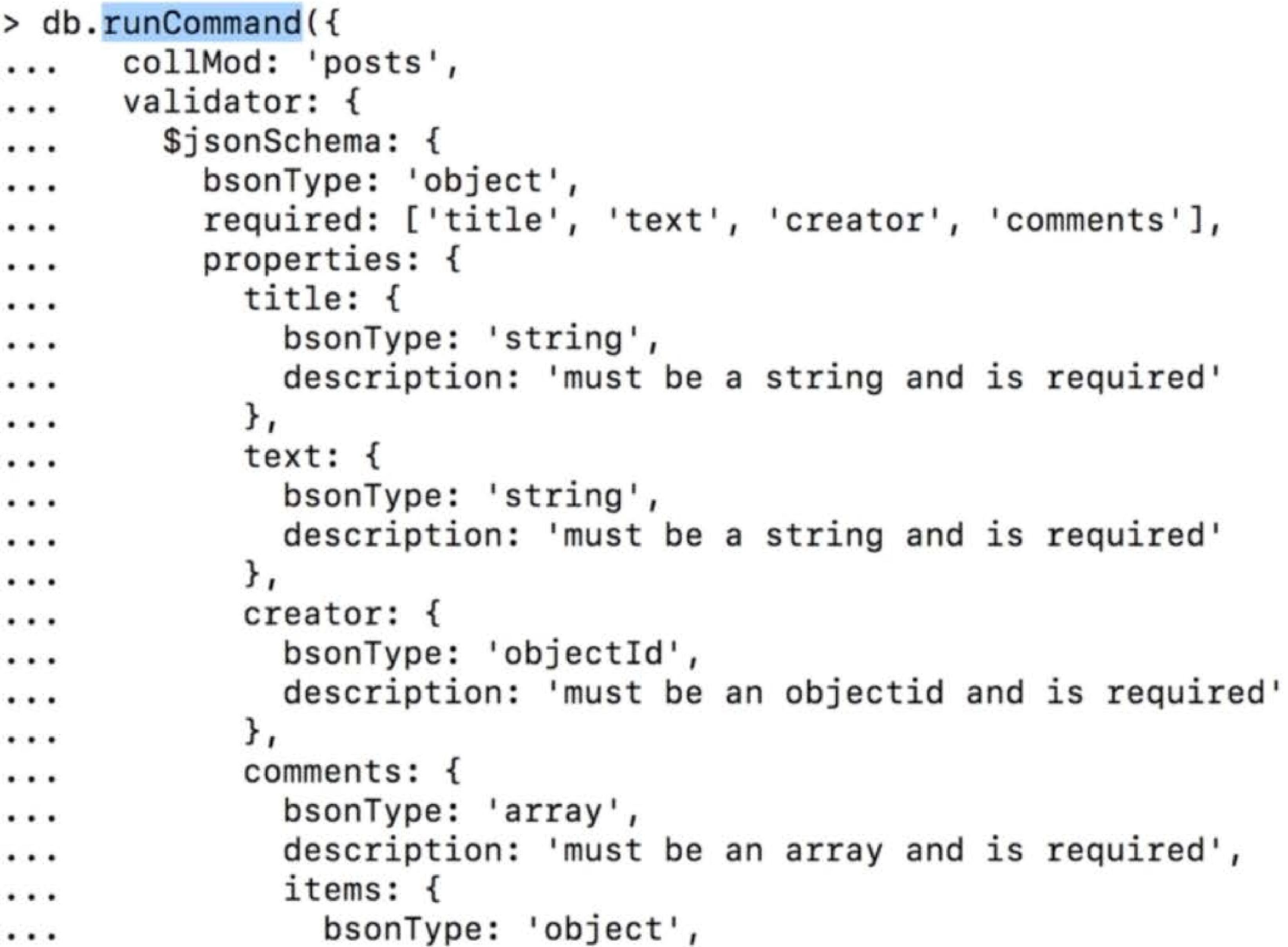
```
> db.posts.insertOne({title: "My first Post!", text: "This is my first post, I hope you like it!", tag
s: ["new", "tech"], creator: ObjectId("5b98e0db4d01c52e1637a9a9"), comments: [{text: "I like this post
!", author: ObjectId("5b98e0db4d01c52e1637a9a8")}]})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5b98e17d4d01c52e1637a9aa")
}
> db.posts.findOne()
{
        "_id" : ObjectId("5b98e17d4d01c52e1637a9aa"),
        "title" : "My first Post!",
        "text" : "This is my first post, I hope you like it!",
        "tags" : [
                "new",
                "tech"
        ],
        "creator" : ObjectId("5b98e0db4d01c52e1637a9a9"),
        "comments" : [
                {
                        "text" : "I like this post!",
                        "author" : ObjectId("5b98e0db4d01c52e1637a9a8")
                }
        ]
}
> 
```



```
db.createCollection('posts', {
  validator: {
    $jsonSchema: {
      bsonType: 'object',
      required: ['title', 'text', 'creator', 'comments'],
      properties: {
        title: {
          bsonType: "string",
          description: "must be a string and is required"
        },
        text: {
          bsonType: "string",
          description: "must be a string and is required"
        },
        creator: {
          bsonType: "objectid",
          description: "must be an objectid and is required"
        },
        comments: {
          bsonType: "array",
          description: "must be an array and is required"
        }
      }
```



```
> db.posts.insertOne({title: "My first Post!", text: "This is my first post, I hope you like it!", tag
s: ["new", "tech"], creator: ObjectId("5b98e0db4d01c52e1637a9a9"), comments: [{text: "I like this post
!", author: ObjectId("5b98e0db4d01c52e1637a9a8")}]})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5b98e67e4d01c52e1637a9ab")
}
> db.posts.findOne()
{
        "_id" : ObjectId("5b98e67e4d01c52e1637a9ab"),
        "title" : "My first Post!",
        "text" : "This is my first post, I hope you like it!",
        "tags" : [
                "new",
                "tech"
        ],
        "creator" : ObjectId("5b98e0db4d01c52e1637a9a9"),
        "comments" : [
                {
                        "text" : "I like this post!",
                        "author" : ObjectId("5b98e0db4d01c52e1637a9a8")
                }
        ]
}
> db.posts.findOne()
```



```
> db.runCommand({
...   collMod: 'posts',
...   validator: {
...     $jsonSchema: {
...       bsonType: 'object',
...       required: ['title', 'text', 'creator', 'comments'],
...       properties: {
...         title: {
...           bsonType: 'string',
...           description: 'must be a string and is required'
...         },
...         text: {
...           bsonType: 'string',
...           description: 'must be a string and is required'
...         },
...         creator: {
...           bsonType: 'objectId',
...           description: 'must be an objectid and is required'
...         },
...         comments: {
...           bsonType: 'array',
...           description: 'must be an array and is required',
...           items: {
...             bsonType: 'object',
...           bsonType: 'objectId',
...           description: 'must be an objectid and is required'
...         },
...         comments: {
...           bsonType: 'array',
...           description: 'must be an array and is required',
...           items: {
...             bsonType: 'object',
...             required: ['text', 'author'],
...             properties: {
...               text: {
...                 bsonType: 'string',
...                 description: 'must be a string and is required'
...               },
...               author: {
...                 bsonType: 'objectId',
...                 description: 'must be an objectid and is required'
...               }
...             }
...           }
...         }
...       }
...     }
...   },
...   validationAction: 'warn'
... });
{ "ok" : 1 }
> 
```